THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace

Présentée et soutenue par :
Nicolas DROUGARD

le vendredi 18 décembre 2015

Titre :

Exploiting imprecise information sources in sequential decision making problems under uncertainty

Tirer profit de sources d'information imprécises pour la décision séquentielle dans l'incertain

---

École doctorale et discipline ou spécialité :

EDSYS : Systèmes embarqués

Unité de recherche :

Équipe d'accueil ISAE-ONERA CSDV

Directeur(s) de Thèse :

M. Didier DUBOIS (directeur de thèse)
M. Florent TEICHTEIL-KÖNIGSBUCH (co-directeur de thèse)

Jury :

M. Patrice PERNY - Président, Rapporteur
M. Didier DUBOIS - Directeur de thèse
M. Jean-Loup FARGES Ingénieur de recherche ONERA
M. Hector GEFFNER Professor - Rapporteur
M. Florent TEICHTEIL-KÖNIGSBUCH Research Engineer - Co-directeur de thèse
M. Bruno ZANUTTINI Assistant professor

To the Pentahedron, and especially to Madam Sarbel.

Thanks

My very first thanks goes to my PhD mentors Didier, Florent and Jean-Loup, for their trust and support throughout what was for me a great adventure: I did learn a ton during these three years thanks to your advice, your help, your availability and for sharing all this valuable knowledge with me.

I would like to strongly thank the committee for their work and time: Professors Hector Geffner, Patrice Perny and Bruno Zanuttini. Thank you for your reactions and advice. Bruno, thank you for these few days at the GREYC with the MAD team and for your enthusiasm.

This work is dedicated to my two sisters Anne and Marion, to my parents, my grandparents and family: I thank you for everything, I am very lucky to have you.

I will also thank all the amazing researchers I had an opportunity to meet or to work with: Alexandre Albore, Nahla Ben Amor, Magalie Babier, Philippe Bidau, Pierre Bisquert, Grégory Bonnet, Oliver Buffet, Maël Chiapino, Frédéric Dehais, Ludovic D'Estampes, Clément Farabet, Hélène Fargier, Myriam Foucras, Thibault Gateau, Nicolas Goix, Marek Grzes, Guillaume Infantes, Andrey Kolobov, Maxime Laborde, Charles Lesire, Éric Lombardi, Faouzy Lyazrhi, Rémi Munos, Florence Nicole, Sylvain Peyronnet, Sergio Pizziol, Caroline Ponzoni Carvalho Chanel, Nico Potyka, Henry Prade, Xavier Pucel, Régis Sabbadin, Scott Sanner, Jörg Shönfish, Catherine Tessier, Tiago Veiga, Guillaume Verges, Simon Vernhes, Gérard Verfaillie, Vincent Vidal, Paul Weng, etc.

I was really happy to share these three nice years with my very sympathetic doctoral student colleagues: Adrien (thank you for all these exciting discussions), my POMDP partner Caroline, my ski partners Elodie and Emmanuel, my swimming pool partner Jérémy, my thesis brother Jonathan, my pause partner Jorrit, "myself" (the last real engineer: he knows very well whom I am talking about), Sergio (I am really proud of our baby appearing in the third chapter), Simon (thank you for all your help in coding, in administrating and for your company during pauses), Alvaro, Clément, Gaetan, Guillaume, Hélène, Henry, Igor, Jan, Julia, Marie, Marine, Martin, Mathieu, Nicolas, Patrick, Pierre, Sylvain, Thibault, Yann, etc.

Finally, I would like to thank my friends because they were a great support during these three years, especially when things went wrong: Alexis, Anaïs, Angèle, Anne-Laure, Arthur, Benoît, Camille, Carole, Catherine, Charlotte, Clémence, Clémentine, Cyril, David, Elisa, Elsa, Emmanuel, Fanny, Frank, Gabriel, Guillaume, Isabelle, Jean, Jérémy, Jocelyn, Jonathan (my rubber duck, thank you for your great help during this thesis, and all these amazing projects), Henry, Ilyès, Julien, Kévin, Lauranne (thank you for having corrected most of the English mistakes of this thesis, for your patience, and obviously thank you for your beautiful smile and all the happiness you bring me), Lola, Louis, Lucille, Mael, Manon, Margaux, Marie, Marie-Laure, Marina, Marion, Mathilde, Maxime, Moustapha, Nicolas, Olivier (mdm), Oriane, Pablo, Paul, Quentin, Renaud, Robin, Sarah, Sophie, Thibault, Yann etc.

I also would like to thank my personal computer which has been a really valiant partner, my cat Ziza (for having slept for me during my few sleepless nights), Eclipse (for underlining errors in red), Linguee (for making this thesis, I hope, more understandable), and Seattle (for the magic of this place).

Toulouse, February 23, 2016.

Contents

List of Figures

Introduction

Context

Providing decisional autonomy to a robot requires among other things to compute a function which returns the symbols of the actions to be triggered at a given moment, considering the data from its sensors. The features of interest of the robot and its surroundings form a *system*. In general, for a given sequence of actions performed by the robot, the evolution of this system is not fixed for sure, but its behavior may be known by performing tests on the robot or using information from expert knowledge. As well, the raw or processed data from the robot sensors are not generally a deterministic outcome of the state of the system or of the taken actions: nevertheless, these data, called also *observations* of the system, depend on the robot's actions and system states. Relations between observations, system states and actions may be known through tests of the sensors in various situations, or by taking into account the description of the sensors, data processing, or any related expert information. For instance, in the case of a robot using Computer Vision (CV), the output of the image processing algorithm employed is considered as an observation of the system since it is the result of processed sensor data and the input of the decision model: here data are images from camera. For a given camera, and a given vision algorithm, the behavior of the observation is related to the action and to the system state during the process of taking images.

Thus, in order to make a robot autonomously fulfill a chosen *mission*, we are looking for a function returning actions conditioned on the sequence of system observations, and taking into account uncertainty about the system evolution and its observation. Such functions may be called *strategies*. The research domain associated to this kind of problem, *i.e.* strategy computation, is not restricted to robotics and is called *sequential decision making under uncertainty*: in the general case, the entity which has to act is called the *agent*. In this thesis, although provided results are mostly theoretical and general enough to tackle much more various applications, the problem of strategy computation is studied in the context of autonomous robotics, and the agent is the decisional part of the robot. Computing a strategy for a given robotic mission needs a proper framework: the best known model describes the state and observation behaviors using Probability Theory.

A probabilistic model for strategy computation

Markov Decision Processes (MDPs) define a useful formalism to express sequential decision problems under probabilistic uncertainty [9]. It is a framework well suited to compute strategies if the actual system state is known by the agent at each point in time. In the robotic context, this assumption means that the considered mission allows us to assume that the robot has full knowledge of the features of interest via its sensors. In this model, a system state is denoted by the letter s, and the finite set of all the possible states is $\mathcal{S}$. The finite set $\mathcal{A}$ consists of all possible actions $a \in \mathcal{A}$ available to the agent. The time is discretized into integers $t \in \mathbb{N}$ which represent time steps of the action sequence.

The state dynamics is assumed to be *Markovian*: at each time step t, the next system state $s_{t+1} \in \mathcal{S}$, only depends on the current one $s_t \in \mathcal{S}$ and the chosen action $a_t \in \mathcal{A}$. This

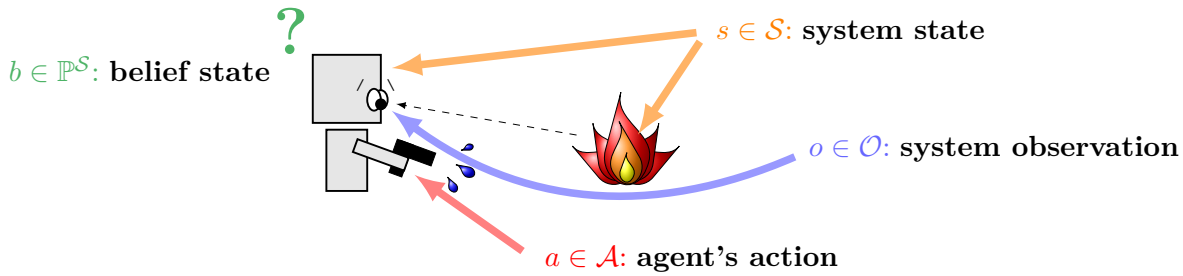


Figure 1 – *Use of a POMDP for the firefighter robot mission modeling: in this toy example, the mission of the robot is fire prevention. The* ***states of the system*** $s \in \mathcal{S}$ *encode for instance the robot location, the water jet orientation, the amount of water used, the fire location and its level on a scale between "minor fire" and "severe fire", etc. Using vision and heat sensors, the robot gets* ***observations*** $o \in \mathcal{O}$ *which are the raw or processed values from the sensors: the output of a classifier whose input is a image of the scene (see Figure 3 and 5), and which returns the fire level or location may be encoded in an observation. Finally, the* ***actions of the robot*** $a \in \mathcal{A}$ *are for instance the rotor activations impacting the rotation of the robot's wheels, the water pumping, the orientation of the water jet or sensors etc. The* ***reward function*** $r(s, a)$ *decreases with the fire level state, and is decreased by a cost proportional to the amount of water used: as an optimal strategy maximizes the mean of the sum of the rewards, the goal of the robot is thus to attack fires without wasting water. This mean can be computed knowing the probabilities describing the uncertain dynamic of the system. The robot actions* $a \in \mathcal{A}$ *have a probabilistic effect on the system, as described by the* ***transition function*** $\mathbf{p}(s' \mid s, a)$*: for instance, the activation of wheel rotors modifies the location of the robot, and the probability of each possible next locations, given the current system state, takes part in the definition of the POMDP. An other example is the action modifying the water jet orientation, which redefines the probability of the next fire level given the current system state. The robot actions* $a \in \mathcal{A}$ *and next states* $s' \in \mathcal{S}$ *may also impact the observations from the sensors, as defined by the* ***observation function*** $\mathbf{p}(o' \mid s', a)$*: for instance, the orientation of the vision sensor may modify the probability of fire detection or fire level evaluation, which are parts of the observations* $o' \in \mathcal{O}$*. Finally, the* ***belief state*** *is the conditional probability distribution of the current system state conditioned on all observations and actions up to the current time step: as observations and actions are the only data available to the robot, the belief state can be seen as the robot's guess.*

relation is described by a transition function $\mathbf{p}(s_{t+1} \mid s_t, a_t)$ which is defined as the probability distribution on the next system states s_{t+1} conditioned on each action: if the action $a_t \in \mathcal{A}$ is selected by the agent, and the current system state is $s_t \in \mathcal{S}$, the next state $s_{t+1} \in \mathcal{S}$ is reached with probability $\mathbf{p}(s_{t+1} \mid s_t, a_t)$.

The mission of the agent is described in terms of rewards: a reward function $r : (s, a) \mapsto r(s, a) \in \mathbb{R}$ is defined for each action $a \in \mathcal{A}$ and system state $s \in \mathcal{S}$, and models the goal of the agent. Each reward value $r(s, a) \in \mathbb{R}$ is a local incentive for the agent. The more rewards are gathered during one execution of the process, the better: a realization of a sequence of system states and actions is considered as well fulfilling the desired mission if encountered rewards $r(s_t, a_t)$ are high. Solving an infinite horizon MDP consists in computing an optimal strategy, *i.e.* a function prescribing actions $a \in \mathcal{A}$ to be taken over time, and maximizing the mean of the sum of rewards gathered during one execution: this mean is computed with respect to the probabilistic behavior of the system state encoded by transition functions $\mathbf{p}(s_{t+1} \mid s_t, a_t)$. For instance, a preferred strategy may be a function d defined on $\mathcal{S}$, as the current state is available to the agent, and with values in $\mathcal{A}$.

Such markovian strategies are proven to be optimal for some criteria like the one based on accumulated discounted rewards: indeed, a well-known criterion measuring the accuracy of

strategy d is the (infinite horizon) expected discounted total reward:

$$\mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t r(s_t, d_t)\right], \tag{1}$$

where $d_t = d(s_t) \in \mathcal{A}$ and $0 < \gamma < 1$ is a discount factor ensuring the convergence of the sum.

The assumption that the agent has a perfect knowledge of the system state is quite strong: in particular, in the case of robots realizing tasks with conventional sensors, the latter are usually unable to provide to the robot all the features of interest for the mission. Thus, a more flexible model has been built, taking into account *partial observability* of the system state by the agent.

Indeed a Partially Observable MDP (POMDP) [135] makes a step further into modeling flexibility, handling situations in which the agent does not know directly the current state of the system: it more finely models an agent acting under uncertainty in a partially hidden environment.

The set of system states $\mathcal{S}$, the set of actions $\mathcal{A}$, the transition function $\mathbf{p}(s_{t+1} \mid s_t, a_t)$ and the reward function $r(s, a)$ remain the same as for the MDP definition. In this model, since the current system state $s \in \mathcal{S}$ cannot be used as available information for the agent, the agent knowledge about the actual system state comes from observations $o \in \mathcal{O}$, where $\mathcal{O}$ is a finite set. The observation function $\mathbf{p}(o_{t+1} \mid s_{t+1}, a_t)$ gives for each action $a_t \in \mathcal{A}$ and reached system state $s_{t+1} \in \mathcal{S}$, the probability over possible observations $o_{t+1} \in \mathcal{O}$. Finally, the *initial belief state* $b_0(s)$ defines the *prior* probability distribution over the system state space $\mathcal{S}$. An example of usage of a POMDP is presented in Figure 1.

Solving a POMDP consists in computing a strategy which returns a proper action at each process step, according to all received observations and selected actions *i.e.* all of the data available to the agent: a criterion for the strategy may be also the expected discounted sum of rewards (I.2).

Most POMDP algorithms reason about the *belief state*, defined as the probability of the actual system state knowing all the system observations and agent actions from the beginning. This belief is updated at each time step using the Bayes rule and the new observation. At a given time step $t \in \mathbb{N}$, the belief state $b_t(s)$ is defined as the probability that the t^{th} state is $s \in \mathcal{S}$ conditioned on all past actions and observations, and with prior b_0: it estimates the actual system state using the available data, as the latter is not directly observable.

It can be easily recursively computed using Bayes rule: at time step t, if the belief state is b_t, chosen action $a_t \in \mathcal{A}$ and new observation $o_{t+1} \in \mathcal{O}$, next belief is

$$b_{t+1}(s') \propto \mathbf{p}(o_{t+1} \mid s', a_t) \cdot \sum_{s \in \mathcal{S}} \mathbf{p}(s' \mid s, a_t) \cdot b_t(s). \tag{2}$$

as illustrated by the Bayesian Network in Figure 2.

As successive beliefs are computed with the observations perceived by the agent, they are considered as visible by the agent. Let us denote by $\mathbb{P}^{\mathcal{S}}$ the infinite set of probability distributions over $\mathcal{S}$. An optimal strategy can be looked for as a function d defined on $\mathbb{P}^{\mathcal{S}}$ such that successive $d_t = d(b_t) \in \mathcal{A}$ maximize the expected reward (I.2): the agent decisions are then based on the belief state.

The POMDP framework is a flexible model for autonomous robotics, as illustrated by the firefighter example, see Figure 1: it allows to describe all the robotic and surrounding system, as well as the robot's mission, and it is commonly used in robotics [109, 100, 93, 32, 33]. It takes into account that the robot receives data from its sensors only, and thus has to figure out the actual system state using these data, named observations, in order to fulfill the mission. However the POMDP model raises some issues, in particular in the robotic context.

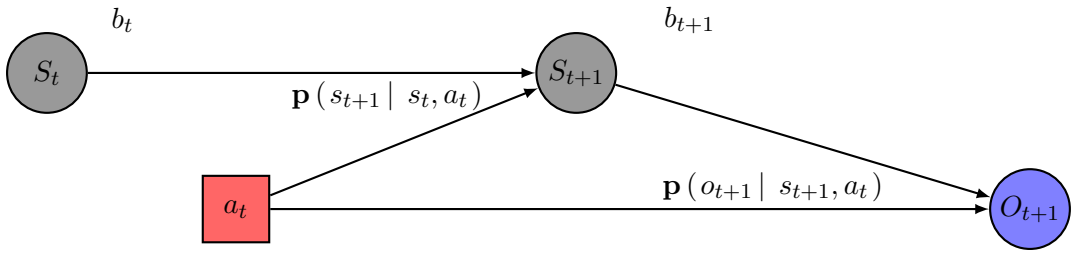


Figure 2 – *Bayesian Network illustrating the belief update: the states are the gray circular nodes, the action is the red square node, and the observation is the blue circular node. The random variable S_{t+1} representing the next state s_{t+1} depends on the current one s_t and on the current action a_t. The random variable O_{t+1} representing the next observation o_{t+1} depends on the next state s_{t+1} and on the current action a_t too. The belief state b_t (resp. b_{t+1}) is the probabilistic estimation of the current (resp. next) system state s_t (resp. s_{t+1}).*

Figure 3 – *Example of an observation method in a robotic context: the robot, here a drone, is equipped with a camera and uses a classifier computed from a image dataset (as NORB, see Figure 4): such a classifier is described in Figure 5. The classifier is generated before the mission (off-line) with a image dataset (see the right part of the illustration), and the classifier output is used during the mission (online) as an observation, for the agent (see the left part). Here, observations are thus generated by computer vision means.*

Practical issues of the POMDP framework

Complexity

Solving a POMDP *i.e.* computing an optimal strategy, is PSPACE-hard in finite horizon [103] and even undecidable in infinite horizon [92]. Moreover a space exponential in the problem description may be required for an explicit specification of such a strategy. See [95] for a more detailed complexity analysis of POMDPs.

This high complexity is well-known by POMDP practitioners: optimality can only be reached for tiny problems, or highly structured ones. Classical approaches try to solve this problem using Dynamic Programming and linear programming techniques [27]. Otherwise, only approximate solutions can be computed, and thus the strategy has no optimality guaranty. For instance, popular approaches such as point-based methods [108, 84, 136], grid-based ones [70, 25, 13] or Monte Carlo approaches [132], use approximate computations.

The *Predictive State Representation* approach [89], reasons directly at the level of the sequences of observations and actions, without referring to the belief states. It may be an interesting approach to simplify computations [77], but only few works address this problem. This approach is more motivated by research in learning [134, 18]. Moreover, it does not tackle the next POMDP's practical issues that will be highlighted, concerning modeling flaws appearing when using this framework: these issues are easily illustrated via robotic situations.

Parameter imprecision and computer vision

Consider now robots using visual perception, and whose observations come from computer vision algorithms based on statistical learning (see, for instance, Figure 3). In this situation,

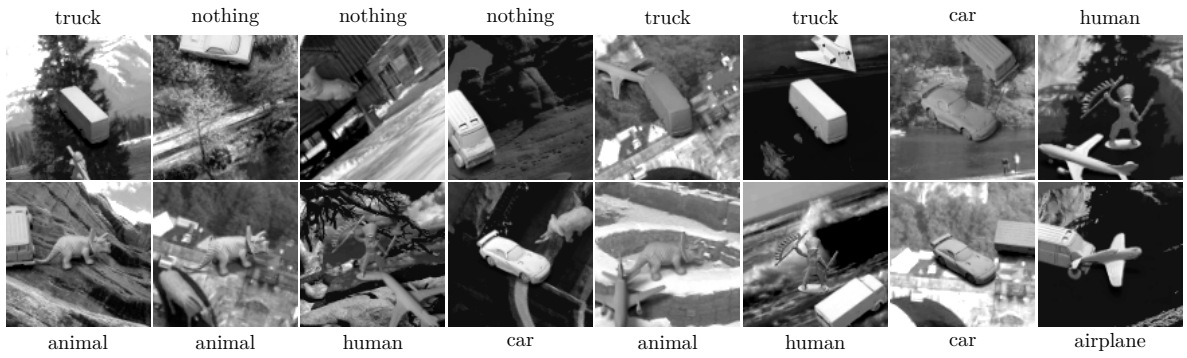


Figure 4 – *Example of image dataset for computer vision: the labeled image dataset NORB [87]. The size of NORB is higher than* 3.10^5*, and images from this dataset represents objects among the* 5 *classes: "animal", "car", "human", "nothing", "plane" and "truck". Each element of a labeled image dataset is composed of a image (e.g. a image showing a car) and a label corresponding to the class of the object represented by the image (in the previous example, the label is "car"). This dataset can be used for supervised learning to compute a classifier (see Figure 5). In order to be able to discern locations of targets, a image is labeled with the name of centered object ("nothing" if there is nothing in the image center).*

the robot uses a *classifier* to recognize objects in images: the classifier is supposed to return the name of the object actually in the image, and makes some mistakes with a low probability (see confusion matrix of Figure 6).

The classifier is computed using a *training dataset* of images (as NORB, see Figure 4, authors made it available at `http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/`). A powerful gradient-based learning algorithm meant to compute classifiers using image datasets is described in Figure 5: the associated framework is called Convolutional Nextwork [88]. Figures (4), (5) and (6) illustrate the example of a classifier computed for a UAV mission where features of interests (system states of the problem) are related with the presence (or absence) of animals, cars, humans, planes or trucks: the statistical problem of computing a classifier recognizing such objects in images is called *multiclass classification*.

As the classifier is learned based on a image dataset (see weights learned in Figure 5), its behavior, and thus its performances (*i.e.* how well it predicts objects in images) are inevitably dependent on the dataset. It is a problem if the image variability in the dataset is too low: in this case, the probabilistic behavior of the classifier will be dependent on these particular images, and the robotic system will have poor observation capabilities when the considered mission involves images too different from the ones in the dataset.

Some large image datasets with a high image variability exist (e.g. NORB, Figure 4, although variability could be ideally higher): note however that with such datasets, the vision performances are reduced, or good performances are, at least, harder to reach.

A confusion matrix can be computed (see Figure 6) using such a labeled image dataset, not used for the training, and called *testing dataset*: observation frequencies can be deduced from this matrix, normalizing rows into probabilities. A row corresponds to an object in the scene, and probabilities in this row are observation probabilities, *i.e.* each probability value is the frequency with which the classifier returns the name of the object of the corresponding column. These probabilities can be used to define the observation function $\mathbf{p}(o' \mid s', a)$ introduced above. This approach raises the issue of knowing if the testing dataset is representative enough of the actual mission. If not, these observation probabilities may not be reliable, and the POMDP badly defined: however, as shown by equation (2) the belief update needs a perfect knowledge of the observation probability distributions.

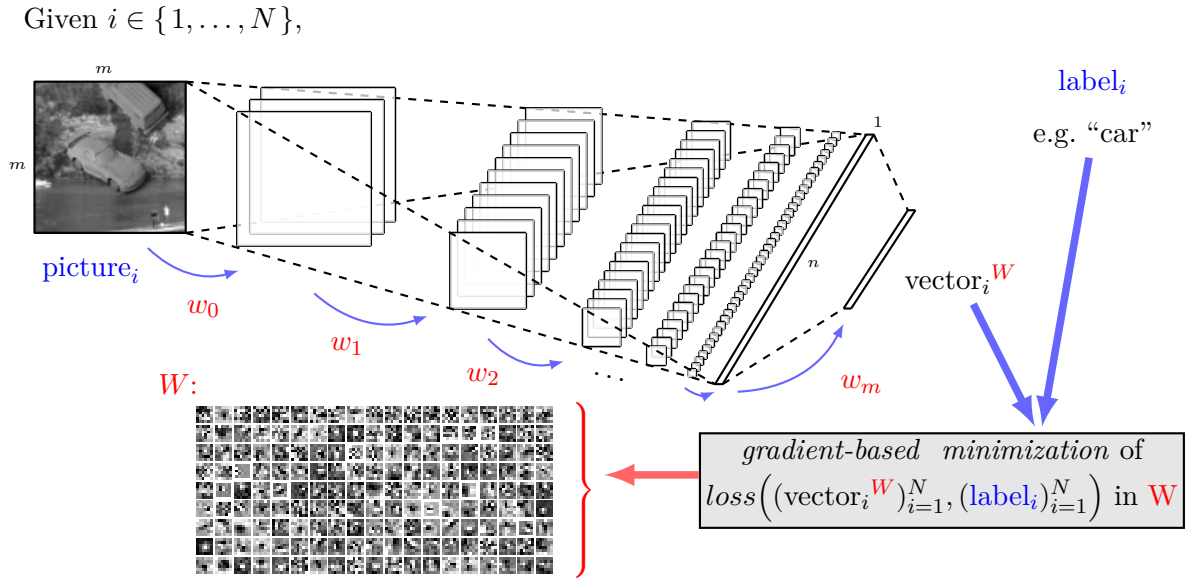


Figure 5 – *Example of classifier training for computer vision: the labeled image dataset NORB, see Figure 4, is used to train and test a classifier. The learning algorithm is based on Convolutional Networks [88, 128] using gradient methods [20, 86, 21]. The weights* $W = (w_0, \ldots, w_m)$ *are the parameters of a particular transformation (see the "bi-pyramid" of successive transformation stages) from the image to a vector representing a label among* $\{animal, car, human, nothing, plane, truck\}$*. These weights are learned in order to minimize a given loss function i.e. a proper criterion representing the error of the classifier over the dataset. A classical loss function is the Mean Squared Error (MSE). The environment Torch7 (based on lua and C languages, [36]) has been used to compute the displayed weights.*

More generally, observations of the agent may be outputs of image processing algorithms whose semantics (image correlation, object matching, class inference, preprocessing followed by classifiers such as the one computed in Figure 5 etc.) are so complex that probabilities of occurrence are hard to rigorously extract.

Finally, if the considered datasets are labeled more precisely (as NORB, which includes information such as the lighting condition, or the object scale), we can imagine that the computed observation probabilities (from the confusion matrix) were more reliable, or the vision performances upgraded (since separation when learning the classifier is easier). However, as more observation or states are involved in this case, the POMDP is harder to solve. Moreover, as the number of images per class is reduced (since there are more complex and numerous classes), a poorer confusion matrix is obtained when testing (in terms of confidence).

As a conclusion, the POMDP model supposes the knowledge of all probability distributions involved: unfortunately the actual frequencies used to compute them are not precisely known in practice. The imprecision about these probabilities, for instance the imprecision related to the behavior of outputs of the computer vision algorithms when dealing with images taken during the mission, has to be taken into account to make the robot autonomous under any circumstances. In general, the computation of the probability distributions of a POMDP needs enough tests for each possible system state and action, which is hard to perform: there are limited data to learn the model in practice.

Some variations of the POMDP framework have been built in order to take into account the imprecision about the probability distributions of the model, also called *parameter imprecision*.

Works handling parameter imprecision

Here, transition and observation functions, namely $\mathbf{p}(s' \mid s, a)$ and $\mathbf{p}(o' \mid s', a), \forall (s, s', o', a) \in \mathcal{S}^2 \times \mathcal{O} \times \mathcal{A}$, are known as *POMDP parameters*, or also *model parameters*. To the best of our knowledge, the first model meant to handle parameter imprecision has been named POMDPIP, for *POMDP with Imprecise Parameters* (POMDPIP) [76]. In this work, each POMDP parameter is replaced by a set of possible parameters. Figure 7 introduces the Imprecise Probability Theory that inspired this idea: in this theory, uncertainty and knowledge about it are represented by sets of probability distributions (named *credal sets*) to express the fact that the model is not fully known. Imprecision on the POMDP model is well expressed in this way: for instance, if confidence bounds on the computed probabilities are available, these bounds can be taken into account when modeling the problem, considering the sets of all likely probability distributions.

Starting from this modeling, a *second order belief* is introduced in this work: it is defined as a probability distribution over the possible model parameters. This approach thus confuses parameter imprecision and event frequency. Indeed, strategies which are looked for are the optimal strategies of a particular "averaged POMDP": this POMDP results from the computation of the average of the possible parameters with respect to the given second order belief. However, the "averaged parameters" of this new POMDP are potentially not even part of the credal set (e.g. with non-convex credal sets). Although POMDPIPs are well designed, imprecision is not satisfactorily handled when the problem is solved because of the use of a second order belief. Moreover, it is claimed that the actual second order belief is not known, and then is allowed to vary in order to make the computation of an approximated strategy easier: the analysis of performances only details how far is the approximated strategy from the "averaged POMDP".

Another work, named *Bounded Parameters POMDPs* (BPPOMDP) [96], deals with parameter imprecision: in this work, imprecision on each parameter, *i.e.* on each probability distribution defining the POMDP, is described by a lower and a upper bound on possible distributions. In other words, credal sets are defined with bounds. The BPPOMDP solving does not introduce any second order belief. However, solving BPPOMDPs is similar to solving POMDPIPs [76] in spirit since the flexibility provided by the parameter imprecision is used to make the computations as easy as possible: the criterion defining optimal strategies is not made explicit, but care is taken to simplify the representation of an approximate criterion in order to avoid memory issues.

The major issue with POMDPIP and BPPOMDP frameworks is that no appropriate criterion choice is made to manage parameter imprecision: for instance, a suitable approach would be to compute a cautious strategy regarding the imprecision *i.e.* taking into account the worst

animal	human	plane	truck	car	nothing		
3688	575	256	48	144	149	animal	75.885%
97	4180	81	20	225	257	human	86.008%
292	136	3906	237	202	87	plane	80.370%
95	1	44	4073	514	133	truck	83.807%
129	3	130	1283	3283	32	car	67.551%
154	283	36	63	61	4263	nothing	87.716%

Figure 6 – *Example of confusion matrix for multiclass classification: this matrix is computed with a testing dataset of images, different from the training dataset. Each row only considers the images of a given object, and numbers represent the answers of the classifier: for instance,* 3688 *images of animals are well recognized, but* 575 *are confused with a human. Average row correct is here* 80.223%. *Torch7 environment [36] has also been used to compute this matrix from the classifier and the testing dataset.*

case. Yet, the opposite approach would be to look for an optimistic strategy, considering the best case.

A more recent approach deals with the cautious point of view, and is thus named *robust POMDP* [101]. Inspired by the corresponding work in the fully observable case (called *uncertain MDP* [97]), this work uses the well-known *maximin*, or *worst case* criterion, which comes from *Game Theory*: in this framework, an optimal strategy maximizes the lowest criterion among all the parameters considered possible. If the parameter imprecision is not stationary, *i.e.* can change at each time step, the corresponding optimal strategy (in the maximin sense) can be easily computed using classical computations (called *Dynamic Programming* [8]). However, when the parameter imprecision is stationary, things become harder: the proposed computations lead to an approximately optimal strategy, as the handled criterion is a lower bound of the desired maximin criterion. Yet, the stationary assumption for the parameter imprecision seems more suitable according to the proposed POMDP definition: indeed, transition and observation functions are here defined as independent from the time step, mostly to make the use of infinite horizon criteria such as (I.2) easier.

Although the use of sets of probability distributions (credal sets) makes the model more consistent with the reality of the problem (parameters are imprecise in practice), taking them into account increases eventually the complexity of the optimal strategy computation (e.g. with a maximin criterion). This intuition is illustrated in Figure 7: Imprecise Probability Theory has the greatest expressiveness compared to the other theories [145], that is why computations with this modeling can be expected to be harder to perform. As explained above, POMDP solving is already a really hard task, thus using a framework leading to more complex computations does not seem to be a good approach. Moreover, with this framework, some restrictive assumptions are commonly needed such as the convexity of credal sets [97, 101]. Modeling a problem in an expressive manner may lead to the use of important approximations without control of the latter, as done with the POMDPIP and BPPOMDP frameworks. It may be a better approach to start with a simpler model which can be solved more easily in practice.

Finally, while there are fascinating questions about algorithms in this area, this thesis does not deal with Reinforcement Learning (RL) [47, 117]. We are interested in robotic applications whose very first executions have to proceed efficiently and safely: the basic reason is that robotic systems and their operations may be expensive, and executions are not allowed to break the robot. However, RL sequentially improves a strategy with potentially unsafe executions. However, RL approaches may improve the strategy in the long term using all the previous executions.

Another practical issue of the POMDP model may be also pointed out: it is about the definition of the belief state concerning the very first state of the system, and more generally about how agent knowledge is represented.

Agent ignorance modeling

The initial belief state b_0, or *prior* probability distribution over the system states, takes part in the definition of the POMDP problem. Given a system state $s \in \mathcal{S}$, $b_0(s)$ is the frequency of the event "the initial state is s". This quantity may be hard to properly compute, especially when the amount of available past experiments is limited: this reason has been already invoked above, leading to the imprecision of the transition and observation functions.

As an example, consider a robot that is for the first time in a room with an unknown exit location (initial belief state) and has to find the exit and reach it. In practice, no experience can be repeated in order to extract a frequency of the location of the exit. In this kind of situation, uncertainty is not due to a random fact, but to a lack of knowledge: no frequentist initial belief state can be used to define the model.

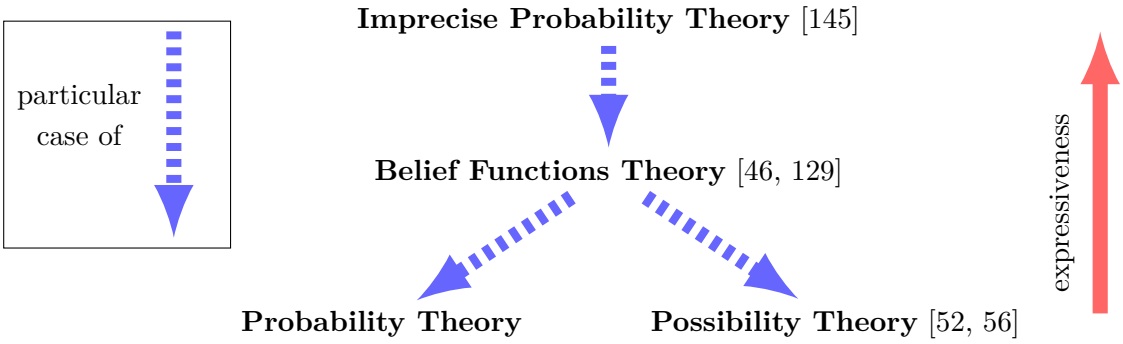


Figure 7 – *Most known uncertainty theories and their relations. Imprecise Probability Theory (IPT) considers sets of probability measures defined on the universe Ω in order to represent uncertainty of events and knowledge about it: this theory is the most general one. The Belief Functions Theory (BFT, or Dempster-Shafer Theory, or yet Theory of Evidence), which is a particular case of IPT, considers a mass function m defined on all the subsets of the universe, and which sums to 1. From this mass function, upper and lower bounds on possible probability measures can be defined. The upper bound is called plausibility measure: $\forall A \subset \Omega$, $Pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$. The lower bound is called the belief function: $\forall A \subset \Omega$, $bel(A) = \sum_{B \subseteq A} m(B)$. The set of probability distributions represented by m are all probability measures $\mathbb{P}$ such that $\forall A \subset \Omega$, $bel(A) \leqslant \mathbb{P}(A) \leqslant Pl(A)$. For instance, if the function m returns 1 for the universe, and 0 for all other subsets, it expresses the total ignorance about the actual probability distribution: it corresponds to the set of all the possible probability distributions in IPT. If the mass function is equal to zero for each non-singleton subsets, it simply corresponds to a probability distribution since $bel = Pl$. If the mass function is positive only on a sequence of nested subsets, it corresponds to a possibility distribution: bel is then called the* ***necessity measure****, and Pl the* ***possibility measure*** *(see Section I.2.1 of Chapter I). Probability and Possibility Theories are thus particular cases of BFT and a fortiori of IPT.*

In other cases, the agent may strongly believe that the exit is located in a wall as in the vast majority of rooms, but it still grants a very small probability p_ϵ to the fact that the exit may be a staircase in the middle of the room. Even if this is very unlikely to be the case, this second option must be taken into account in the belief state, otherwise Bayes rule (see Equation 2) cannot correctly update it if the exit is actually in the middle of the room. Eliciting p_ϵ without past experience is not obvious at all and does not rely on any rational reasons, yet it dramatically impacts the agent's strategy.

The initial system state may be deliberately stated as unknown by the agent with absolutely no probabilistic information: consider robotic missions for which a part of the system state, describing something that the robot is supposed to infer by itself, is initially fully unknown. In a robotic exploration context, the location or the nature of a target, or even the initial location of the robot may be defined as absent from the knowledge of the agent. Classical approaches initialize the belief state as a uniform probability distribution (*e.g.* over all robot/target possible locations, or over possible target natures), but it is a subjectivist answer [41, 64]. Indeed, all probabilities are the same because no event is more plausible than another: it corresponds to equal betting rates. However following belief updates (see Equation 2) will eventually mix up frequentist probability distributions, given by transition and observation functions, with this initial belief which is a subjective probability distribution: it does not always make sense and it is questionable in many cases. Thus, the use of POMDPs in these contexts, faces the difficulty to encode agent ignorance.

Since the knowledge of the agent about given features of the system may be initially partial (or even absent), it makes sense to pay attention to the evolution of this knowledge during the execution of the considered process. Some works inspired by the research in *Active Perception*, have been focusing on the information gathered by the agent in the POMDP framework [29, 31]. In these works, the *entropy* of the belief state is taken into account in the criterion, ensuring

the computed strategy to make the process reach belief states that have a lower entropy (*i.e.* are closer to a deterministic probability distribution) while always satisfying the requirement of an high expected total reward. This approach leads to really interesting results in practice.

Nevertheless, a tradeoff parameter is introduced making the entropy part of the criterion more or less important: tuning the latter may add additional computations. Moreover this approach does not distinguish between the actual frequentist behavior of the system state, and the lack of knowledge about the actual belief state (as a probability distribution) due to the imprecision of the initial belief state, or even of the transition and observation function: it just tends to make the belief as deterministic as possible, even if it is not possible due to the actual probability distributions of the model (e.g. if the transition probability distribution $\mathbf{p}(s_{t+1} \mid s_t, a_t)$ has an high entropy for each action $a_t \in \mathcal{A}$), and even if it is not needed (e.g. if each system state $s \in \mathcal{S}$ such that the belief state b_t is not zero, $b_t(s) > 0$, leads to an high reward).

However, it has been shown that the proposed criterion (including the entropy of the belief state) makes the agent estimate faster its real state, which is really useful in some practical problems. Other models have been introduced, making the reward function dependent on the belief state [4, 30]: this enables to make the agent behavior vary with respect to the probabilistic representation of its knowledge.

General problem

Previous sections presented some issues encountered in practice when using the POMDP framework to compute strategies, especially in the robotic context. The really high complexity of the problem of computing an optimal strategy is a first issue: robotic missions often result in high dimensional problems, preventing any algorithm to computed a sufficiently near optimal strategy, because of the prohibitive computation time or memory needed for this task. Second we highlighted the difficulty of defining the probability distributions describing the problem: for instance the observation function can be hard to define when the observations come from complex computer vision algorithms. Finally the problem of managing the knowledge and the ignorance of the agent has been discussed: there is no clear answer concerning the way to represent the initial lack of knowledge about the actual world for the agent. Also, how to handle and take into account the current agent knowledge is still open to question: the difficulty comes from the classical POMDP definition which only allows the use of frequentist probability distributions, while more expressive mathematical tools seem necessary.

These problems are the starting points of our work. Indeed, the latter consists in contributing to the problem of computing appropriate strategies for partially observable domains. The computed strategies have to make the robot fulfill the mission as well as possible, from the very first execution, *i.e.* strategy computations are performed before any real execution of the mission. In order to make the robot more powerful in the long term, RL algorithms may improve the prior strategies that we propose to compute in this work, taking into account each past execution of the robotic mission. However this idea is not considered here since we focus on fulfilling the robotic mission as well as possible, especially for the first executions of the process: RL algorithms are useless in this context since the dataset of recorded missions is not sufficiently large during the first mission executions.

The general challenge guiding this work is to proceed with strategy computations only using data and knowledge really available in practice, possibly making the strategy computation easier, instead of using the highly complex and hard to define POMDP framework. In other words, it consists in paying particular attention to the issues pointed out above: namely, the complexity of the strategy computation, the imprecision of the model, and the management of the knowledge of the agent.

Alternative uncertainty theories

Although the use of a more expressive framework may increase the complexity of strategy computations, previous discussions on the issues of parameter imprecision and agent knowledge suggest that we should consider alternative uncertainty theories: those described in Figure 7 may bring useful properties to deal with our problems. As presented above, the Imprecise Probability Theory (IPT) has been already explored to improve the POMDP framework [76, 96, 101]. However, the proper use of the credal sets defining the problem makes the computations harder.

A less expressive uncertainty theory is called the Belief Functions Theory (BFT, see Figure 7). Consider the universe Ω, which is a finite set whose elements represent the possible elementary events of a given situation. The BFT encodes uncertainty about the possible events with a mass function m. The latter is defined on each disjunctive subset of the universe *i.e.* $m : 2^{\Omega} \rightarrow [0, 1]$, and sums to 1 over these subsets: $\sum_{A \subseteq \Omega} m(A) = 1$. Two non-additive measures can be defined from this mass: as detailed in Figure 7 the belief measure *bel* and the plausibility measure *Pl* can be deduced from the mass function, and define the credal set $\mathcal{C}_m$ represented by this mass function: $\mathcal{C}_m = \{\, \mathbb{P} \text{ probability measure } \mid \forall A \subseteq \Omega, bel(A) \leqslant \mathbb{P}(A) \leqslant Pl(A) \,\}$. This shows that BFT is thus a particular case of IPT.

Recall that the frequentist probability of an elementary event $\omega \in \Omega$ can be computed in practice using the law of large numbers *i.e.* this probability can be defined as the number of times the considered event ω occurred according to a sufficiently large dataset of recorded samples, over the size of this dataset (the number of samples): in other words, the frequentist probability is the frequency of occurrence of the considered elementary event in the used dataset. For instance, when computing one of the probability distributions leading to the transition function in the POMDP framework, namely $s' \in \mathcal{S} \mapsto \mathbf{p}\,(s' \mid s, a)$ (given a system state $s \in \mathcal{S}$ and an action $a \in \mathcal{A}$), the finite universe Ω can be defined as the set of system states $\mathcal{S}$, and then an elementary event is a system state $s' \in \mathcal{S}$. Another example is the computation of one of the probability distributions leading to the observation function: $o' \in \mathcal{O} \mapsto \mathbf{p}\,(o' \mid s', a)$, for a given pair $(s', a) \in \mathcal{S} \times \mathcal{A}$. In this case, the finite universe Ω can be defined as the set of observations $\mathcal{O}$, and an observation $o' \in \mathcal{O}$ is an elementary event.

Suppose now that we need to compute, for some modeling purpose (e.g. to define the transition or the observation functions of a POMDP), a frequentist probability distribution describing a given situation (e.g. dynamics of the variables encoding a robotic mission): the frequentist probability of each elementary event has to be computed in the classical way (described just above). Consider now that some samples of the dataset have been imprecisely observed in practice. Usually, samples, *i.e.* instances of an elementary event, are present several times in the dataset: the number of instances of this elementary event, *i.e.* the number of times the event occurred, leads to the frequentist probability distribution after a normalization (dividing this number by the size of the dataset, as explained just above). In the case of imprecise samples in the dataset, a sample consists of a disjunctive set of elementary events instead of the actual elementary event. Indeed, this set describes the imprecision of the sample: the actual elementary event is not clear and could be, rationally speaking, any of the elementary events included in the returned set. The quantity $m(A)$ can then be defined as the frequency of occurrence of the set $A \subseteq \Omega$, as no better precision is available. As just formally presented, the use of a mass function can be needed to handle the imprecision of a given dataset. The mass function can be also the direct translation of the lack of knowledge about a given situation.

Indeed, the total ignorance about the actual probability distribution defining the model is encoded with a mass function returning 1 for Ω, and 0 for each other subset: in IPT, it corresponds to the credal set containing all the existing probability distributions. Note that a more restricted ignorance can be stated with a mass function returning 1 for $A \subset \Omega$, and

0 for each other subset: it means that the support of the (unknown) probability distribution describing the situation of interest is A. Note also that the BFT framework enable to encode complete knowledge about the probability distribution describing the situation: it is the case when the mass is positive on singletons only, *i.e.* $m(A) = 0$, $\forall A \subseteq \Omega$ such that $\#A \neq 1$. Thus, it generalizes Probability Theory as illustrated by Figure 7. The fully observable MDP framework has been already extented to this formalism [142]: to the best of our knowledge, no update of the POMDP framework using BFT has been proposed.

However, the use of BFT in the POMDP framework leads to an exponential growth of the size of the sets on which strategy computations are based: the set $2^{\mathcal{S}}$ (resp. $2^{\mathcal{O}}$) is considered with BFT, instead of $\mathcal{S}$ (resp. $\mathcal{O}$) with the Probability Theory, since BFT assigns a mass to each subset of the universe. This remark is not in favor of using BFT in our study, as the POMDP complexity is already a big issue. Moreover, the mass function m is not always easily specified in the modeling phase. Consider for example a probability distribution encoded in the observation function of a POMDP. The BFT counterpart of this object should be a mass function describing the uncertainty about the observations. The way of defining this mass function from a given classifier (for instance as the one described in Figure 5) and a given testing dataset of images (as NORB, see Figure 4) is not obvious. Indeed the imprecision of the probabilistic behavior of the resulting computer vision algorithm is not easily quantifiable: it should be translated into a mass function positive on sets containing at least two observations. However the sets to consider, as well as the mass value to fix on them, are not clearly established. Although BFT provides a simple way to encode both frequentist information and ignorance, the expressiveness of this theory seems hard to use when managing parameter imprecision in practice.

Thus, let us focus on a less expressive theory called *Possibility Theory* (see Section I.2.1 of Chapter I): as shown by Figure 7, this theory is a particular case of BFT, much like Probability Theory. All subsets of the finite universe Ω which have a positive mass are called *focal sets*. Each *possibility measure* is equivalent to a mass function for which all focal sets are nested: the more focal sets contain a given elementary event, the more the latter is plausible. The possibility measure is defined as the plausibility measure Pl (defined in Figure 7), and the belief measure *bel* is called in this case *necessity measure*. A complete review of the different uncertainty theories and their respective meanings is formulated in [61].

Possibility Theory

A possibility measure has been defined as the plausibility measure of a mass function m whose focal sets are nested (see Figure 8). Let us denote these nested subsets by F_i, $\forall i \in \{1, \dots, d\}$, with d a positive integer: $F_d \subset F_{d-1} \subset \dots \subset F_1 = \Omega$. Note that if $A \subseteq \Omega$ and for a given $i \in \{1, \dots, d\}$, $A \cap F_i \neq \emptyset$, then $A \cap F_j \neq \emptyset$ for each $j \in \{1, \dots, i\}$ since focal sets are nested. As defined in Figure 7, the plausibility measure is $Pl(A) = \sum_{B \cap A = \emptyset} m(B)$, $\forall A \subseteq \Omega$. Thus, it is easy to show that for each $A \subseteq \Omega$, $Pl(A) = \sum_{i=1}^{d_A} m(F_i)$, where $d_A = \max \Big\{ i \in \{1, \dots, d\} \Big| F_i \cap A \neq \emptyset \Big\}$. As $d_{A \cup B} = \max \Big\{ i \in \{1, \dots, d\} \Big| F_i \cap A \neq \emptyset \text{ or } F_i \cap B \neq \emptyset \Big\}$, this integer can be written $d_{A \cup B} = \max \{ d_A, d_B \}$, and then

$$Pl(A \cup B) = \sum_{i=1}^{d_{A \cup B}} m(F_i) = \max \left\{ \sum_{i=1}^{d_A} m(F_i), \sum_{i=1}^{d_B} m(F_i) \right\} = \max \Big\{ Pl(A), Pl(B) \Big\}.$$

Note also that $Pl(\Omega) = 1$ as the mass function sums to 1 over 2^{Ω}, and that $Pl(\emptyset) = 0$ as the mass function assigns zero to the empty set by definition.

The classical definition of a possibility measure, denoted by $\Pi : 2^{\Omega} \to [0, 1]$, corresponds in fact to the results presented just above:

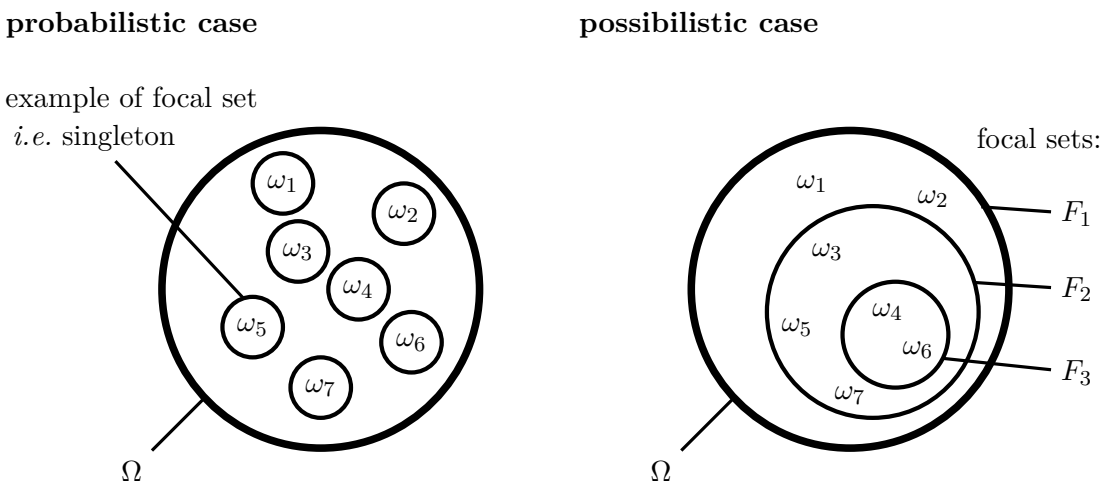


Figure 8 – *The focal sets of a mass function* $m : 2^\Omega \to [0,1]$ *are the subsets* $A \subseteq \Omega$ *of the universe* $\Omega = \{\omega_1, \ldots, \omega_7\}$ *such that the mass function is positive:* $m(A) > 0$. *As Probability Theory and Possibility Theory are particular cases of BFT (see Figure 7), each of the probability (resp. possibility) distributions is equivalent to a certain mass function. Thus, the left part of the illustration shows the focal sets from a probability measure* $\mathbb{P}$*: they are the singletons of* Ω *and* $m(\{\omega\}) = \mathbb{P}(\{\omega\})$. *The right part shows the focal sets from a possibility measure: they are nested subsets of* Ω*, denoted by* $F_3 \subset F_2 \subset F_1 = \Omega$.

- $\Pi(\Omega) = 1$;
- $\Pi(\emptyset) = 0$;
- $\forall A, B \subseteq \Omega$, $\Pi(A \cup B) = \max\Big\{\Pi(A), \Pi(B)\Big\}$.

It follows that this measure is entirely defined by the associated possibility distribution, *i.e.* the possibility measure of the singletons: $\forall \omega \in \Omega$, $\pi(\omega) = \Pi(\{\omega\})$. Indeed, the possibility value of a given event can be defined as the maximal possibility value of the elementary events constituting the event of interest: $\Pi(A) = \max_{\omega \in A} \pi(\omega)$. This is quite similar to the fact that a probability measure over the finite universe Ω, $\mathbb{P} : 2^\Omega \to [0,1]$, is equivalent to the associated probability distribution $\mathbf{p} : \omega \mapsto \mathbb{P}(\{\omega\})$, using the additivity of the probability measure.

The definition of this measure leads to the possibilistic normalization:

$$\max_{\omega \in \Omega} \pi(\omega) = \Pi(\Omega) = 1.$$

If $(\overline{\omega}, \underline{\omega}) \in \Omega^2$ are such that $\pi(\overline{\omega}) < \pi(\underline{\omega})$, the meaning is that $\overline{\omega}$ is less plausible than $\underline{\omega}$. Elementary events with possibility degree 0, *i.e.* $\omega \in \Omega$ such that $\pi(\omega) = 0$, are impossible (same meaning as $\mathbf{p}(\omega) = 0$, with $\mathbf{p} : \Omega \to [0,1]$ a probability distribution). Moreover, elementary events such that $\pi(\omega) = 1$ are entirely possible: here, this is not equivalent to $\mathbf{p}(\omega) = 1$ but rather a necessary condition for a probability value equal to one.

As Ω is a finite set, the number of possible values of a possibility distribution is finite (much like a probability distribution over a finite universe Ω): let us denote by $\{l_1, l_2, \ldots, 0\}$ with $1 = l_1 > l_2 > \ldots > 0$, the different values of the possibility distribution, *i.e.* $\{\pi(\omega) \mid \omega \in \Omega\}$.

On one hand, if the current possibilistic belief coincides with the distribution $\pi(\omega) = 1$, $\forall \omega \in \Omega$, all elementary events are totally possible, and it models therefore the total ignorance of the actual elementary event: this means that Possibility Theory can satisfactorily encode agent ignorance, one of the issues guiding our work. On the other hand, the full knowledge of the actual elementary event, say $\tilde{\omega} \in \Omega$, is encoded by a possibility distribution equal to the

classical indicator function of the singleton $\{\tilde{\omega}\}$, *i.e.* $\pi(\omega) = \mathbb{1}_{\{\tilde{\omega}\}}(\omega)$: the possibility degree of $\tilde{\omega}$ is 1, and all others have a possibility degree equal to 0.

Finally, between these two extrema, a possibilistic situation is described by a set of entirely possible elementary events, $\{\omega \in \Omega \text{ s.t. } \pi(\omega) = 1\}$, and successive sets of less plausible ones $\{\omega \in \mathcal{S} \text{ s.t. } \pi(\omega) = l_i\}$ down to the set of impossible states $\{\omega \in \Omega \text{ s.t. } \pi(\omega) = 0\}$.

As already explained (see above and Figure 7), a remnant from BFT is the existence of two measures in Possibility Theory: the first one is the possibility measure, denoted by Π, and called plausibility measure in BFT (Pl). The second one is the necessity measure, denoted by $\mathcal{N}$, and called belief measure in BFT (bel). Recall that, given a mass function m, the belief measure of the set $A \subseteq \Omega$ is $bel(A) = \sum_{B \subseteq A} m(B)$. Now, if the mass represents a situation which can be expressed with Possibility Theory, the focal sets $(F_i)_{i=1}^{d}$ of the mass function are nested: $F_d \subset F_{d-1} \subset \ldots \subset F_1 = \Omega$ with $d \in \mathbb{N}^*$ the number of focal sets. Note that if $F_i \subseteq A$ for a given $i \in \{1, \ldots, d\}$, $F_j \subseteq A$, $\forall j \geqslant i$, because focal sets are nested. Thus, given $A \subseteq \Omega$, a distinction is made between two cases.

The first case is when $F_d \not\subseteq A$: in this case $bel(A) = 0$ since no focal set is in A. Let us denote by $\overline{A}$ the complementary set of A: $A \cup \overline{A} = \Omega$ and $A \cap \overline{A} = \emptyset$. Now an obvious remark about sets is recalled: if A and B are two subsets of Ω, "$B \subseteq A$" $\Leftrightarrow$ "$B \cap \overline{A} = \emptyset$". Hence, in the first case, $\overline{A} \cap F_i \neq \emptyset$ for each $i \in \{1, \ldots, d\}$, and then the plausibility of $\overline{A}$ is $Pl(\overline{A}) = 1$ (as the sum of all focal sets). So, in this case, $Pl(\overline{A}) + bel(A) = 1$.

In the second case, *i.e.* $F_d \subseteq A$ (and thus $A \neq \emptyset$), it is easy to show that $bel(A) = \sum_{i=d^A}^{d} m(F_i)$, where $d^A = \min\left\{i \in \{1, \ldots, d\} \,\middle|\, F_i \subseteq A\right\}$. Thus, if $A \neq \Omega$, we can see that $\min\left\{i \in \{1, \ldots, d\} \,\middle|\, F_i \cap \overline{A} = \emptyset\right\} = 1 + \max\left\{i \in \{1, \ldots, d\} \,\middle|\, F_i \cap \overline{A} \neq \emptyset\right\}$ *i.e.* $d^A = 1 + d_{\overline{A}}$ (thanks to the obvious remark about sets), where d_A is defined above as $\max\left\{\{1, \ldots, d\} \,\middle|\, F_i \cap A \neq \emptyset\right\}$. As shown above, $Pl(\overline{A}) = \sum_{i=1}^{d_{\overline{A}}} m(F_i)$, thus if $A \neq \Omega$, $Pl(\overline{A}) + bel(A) = 1$. Finally, if $A = \Omega$, all the focal sets are included in A, so $bel(A) = 1$, and $\overline{A} = \emptyset$, so $Pl(\overline{A}) = 0$: $Pl(\overline{A}) + bel(A) = 1$ too.

That is why the necessity measure, which corresponds to the belief measure *bel* in BFT, can be defined from the possibility measure as follows: $\forall A \subseteq \Omega$,

$$\mathcal{N}(A) = 1 - \Pi(\overline{A}).$$

Note that we can differentiate Quantitative Possibility Theory [52] from Qualitative Possibility Theory [62]. The former uses the product when combining possibility distributions in order to compute joint distributions (much like in Probability Theory). The latter uses the min operator for this purpose (see Section I.2.2 of Chapter I). Thus, the qualitative theory uses only max and min operators. Consider some possibility distributions whose values are in the finite set $\{l_1, l_2, \ldots, 0\}$: as only max and min operators are allowed, computations do not generate any value not already included in the finite set $\{l_1, l_2, \ldots, 0\}$. This is not true for Quantitative Possibility Theory since the use of the product operator leads to the creation of new real values in $[0, 1]$.

Finally, in the qualitative framework, the possibility values assigned to each of the elementary events are not really taken into account in terms of real numbers since only qualitative comparisons are performed via the use of max and min operators. Hence, Qualitative Possibility Theory is classically defined with the introduction of a qualitative scale $\mathcal{L}$, which may be defined as $\{l_1, l_2, \ldots, 0\}$, or as any other totally ordered set indifferently. As they are qualitative data, we use the term *possibility degrees* instead of possibility values. The next section clarifies why the use of a qualitative framework is beneficial in terms of complexity and modeling to our study.

Note the similarities between Possibility and Probability Theories, replacing max by +, and min by $\times$ (in the qualitative case). Note also that Qualitative Possibility Theory defines the semiring $(\mathcal{L}, \max, \min)$, and Quantitative Possibility Theory leads to the semiring $([0,1]\max, \times)$. These algebraic structures are close to the well known tropical semirings [107], as the *max-plus* semiring $(\mathbb{R} \cup \{-\infty\} \max, +)$ and the *min-plus* one $(\mathbb{R} \cup \{+\infty\}, \min, +)$.

Qualitative Possibilistic POMDPs

A qualitative possibilistic counterpart of the POMDP framework has been proposed in [121]: this model is called Qualitative Possibilistic POMDP and denoted by π-POMDP. A π-POMDP is simply a POMDP with qualitative possibility distributions as parameters, instead of probability distributions. As the π-POMDP framework is qualitative, the counterpart of the reward function, called *preference function*, is a qualitative function: indeed the preference function returns values from the finite qualitative scale $\mathcal{L}$, and thus is non-additive.

As a particular case of BFT and *a fortiori* of IPT, Possibility Theory expresses a partial knowledge of the actual probability distribution, as presented above. A qualitative possibility distribution is even more imprecise since only qualitative information is given by such a distribution. This imprecision results in the measures presented above: possibility and necessity measures. As detailed in Section I.2.3, two qualitative criteria have been proposed in [125], counterparts of the probabilistic criterion (I.2): a pessimistic one, which is a qualitative counterpart of the maximin (worst-case) criterion in IPT. The other criterion is qualified as optimistic, as a qualitative counterpart of the best case criterion in IPT.

One of the most interesting property of π-POMDPs is the simplification of the strategy computation. Indeed, algorithms proposed for solving classical (probabilistic) POMDPs are often based on the set of the belief states called *belief space*. The belief space is infinite in the general case: each time step leads eventually to a finite number of new belief states, which makes this set countable. In order to get useful properties and means for the strategy computation, the set of all probability distributions over the system space $\mathcal{S}$ is considered, *i.e.* the continuous simplex $\mathbb{P}^{\mathcal{S}} = \{\mathbf{p} : \mathcal{S} \to [0,1] \mid \sum_{s\in\mathcal{S}} \mathbf{p}(s) = 1, \text{ and } \mathbf{p}(s) \geqslant 0, \forall s \in \mathcal{S}\}$. The infinite size of the belief space partly explains why probabilistic POMDPs are very hard to resolve. On the contrary, π-POMDPs have a finite belief space. Indeed, the number of qualitative possibility distributions over the system space $\mathcal{S}$ is lower than $\#\mathcal{L}^{\#\mathcal{S}}$, as the qualitative scale $\mathcal{L}$ is finite. The fully observable version of the π-POMDP model is called π-MDP: as explained in Section I.2.5 of Chapter I, any π-POMDP reduces to a π-MDP whose system space is the qualitative possibilistic belief space, and whose size is exponential in the number of states. In works [66, 69, 121], complexity of π-MDP appears to be lower than complexity of MDP, which is polynomial [103]: complexity of π-POMDP is then at worst exponential in the problem description, whereas POMDP solving may be undecidable [92].

In addition to simplifying the computations, the π-POMDP framework may be very interesting to our use case. Indeed, when we considered a robot using computer vision algorithms to get observations (see Figure 3), we previously highlighted the difficulty of properly defining the probabilistic observation function: probability values of the answers of the vision algorithms in the context of the robotic mission are imprecisely known and hard to define in practice. Finding qualitative estimates of their recognition performance is easier: the π-POMDP model only requires qualitative data, thus it allows to build the model without the use of information other than the one really available. For instance, the confusion matrix in Figure 6 may lead to a qualitative possibilistic observation function which only takes into account how answer frequencies are sorted: in the presence of a human (see second line), the most frequent answer is “human”, the second one is “nothing”, the third one is “car”, etc. Thus, the corresponding possibility distribution is such that, conditioned on the presence of a human, the possibility

degree of the answer "human" is higher than the possibility degree of the answer "nothing", which is higher than the possibility degree of the answer "car", etc. Instead of assigning frequencies which are not really reliable in practice, the qualitative possibilistic model naturally expresses imprecisions about the problem.

Finally, let us recall that the constant possibility distribution whose possibility degrees are all equal to 1 (maximal element of $\mathcal{L}$), represents total ignorance: this distribution can be used to define the initial belief state when it has to represent an agent which initially ignores a given situation. Thus, the π-POMDP framework allows for a formal modeling of the agent's lack of knowledge.

The use of the Qualitative Possibility Theory [62] is thus studied in this work, as it appears to be able to both simplify a POMDP, and to model parameter imprecision and ignorance related to robotic missions. This framework indeed simplifies computations, is able to encode the problem with only available data and formally models the lack of knowledge: thus this theory offers solutions to the three issues highlighted previously. However we note that, as a qualitative framework, it does not allow for frequentist information encoding.

To the best of our knowledge, a rather limited study of the π-POMDP model exists in the literature up to now: in fact, the work [121] seems to be the only one proposing both a definition of π-POMDPs and a toy example to illustrate this model. The fully observable version (π-MDP) has generated some more interest in the research community [123, 122, 147].

Description of our Study

This thesis contributes to determine to which extend Qualitative Possibility Theory can contribute to *planning under uncertainty in partially observable domains*, and more generally to *sequential uncertainty management*, in terms of computation simplification and modeling. It presents recent contributions in the use of this theory for planning under uncertainty and knowledge representation, with an almost systematic use of graphical models [81, 10, 19].

The end of this introduction describes how this thesis is structured. Indeed, each of the following sections corresponds to a chapter of our work and details its contents.

State of the art

As probabilistic POMDPs and qualitative possibilistic ones are the central objects of this thesis, the *first chapter* presents these models starting from low level definitions. Most of the classical results presented are accompanied by proofs making this thesis self-contained: this work is thus accessible to all researchers and students with basic mathematical competences, even if they are not familiar with alternative uncertainty theories, nor member of the *planning under uncertainty* community.

The first part of this chapter focuses on the classical (probabilistic) POMDP: it begins with the presentation of the less complex fully observable case, called MDP. The well-know POMDP is then set up, giving some formal results and proofs that are sometimes hard to find in the literature. This part ends with a review about some of the most successful POMDP algorithms including the ones used in the next chapters.

The second part is naturally devoted to the qualitative possibilistic counterpart of the POMDP. It begins with a presentation of the Possibility Theory, with a specific focus on the qualitative version. It includes the qualitative counterparts of conditioning, and averaging. Finally, as all the needed theoretical tools have been presented, the fully observable model (π-MDP) can be defined, followed by the partially observable one (π-POMDP). As noted above, to the best of our knowledge, only one ten-paged paper has already dealt with the π-POMDPs:

the second part of this chapter is thus the first work formally detailing the construction of this model.

Note that the notations and the terminology set up in this initial chapter, and thus used throughout this thesis, are those classically used in the POMDP framework: it should make our work more readable to the POMDP community or more generally to the probabilistic community. Moreover, dedicating this first chapter to a formal presentation of the probabilistic and the qualitative possibilistic models is meant to highlight their common structure.

Natural updates of the possibilistic model

The *second chapter* proposes several extensions of the work [121]. It begins with the presentation of some variants of the qualitative criteria presented in the first chapter: the latter are related to the aggregations of the preferences over time, and to the chosen approach *i.e.* pessimistic or optimistic.

A qualitative possibilistic version of the Mixed-Observable MDPs [100, 3], where some state variables are fully observable, is then built. It is named π-MOMDP and generalizes both π-MDP and π-POMDP. This contribution reduces dramatically the complexity of solving π-POMDPs, while catching finer information about the environment some state variables of which are fully observable. For instance, the battery level of a robot may be considered as directly observable information for decision making, leading to easier computations. More generally, the existence of visible variables is common in robotics [100].

Next, a qualitative criterion for missions with unbounded durations is proposed, along with an algorithm for computing the associated optimal strategy. This algorithm is used to compute a strategy for a target recognition mission: experimental results compare executions using this strategy to those using the strategy from a probabilistic algorithm, in situations where the probabilistic dynamic of the observations is actually not properly defined.

Note that this experiment is the first practical use of the π-POMDP framework to the best of our knowledge. It also highlights interesting behaviors of the qualitative possibilistic belief state: they are then clarified theoretically in the end of the chapter.

The main contributions of this chapter have been published in [49]. The experiments illustrate that these contributions are necessary in practice. However they are not sufficient to reach a competitive computation time or to deal with realistic robotic problems: the orientation of the next chapter stems from this observation.

Factorized models and symbolic algorithms

The size of the robotic problem dealt with in the previous chapter is small enough to allow the proposed algorithm to compute a strategy within a reasonable amount of time. The contributions of the *third chapter* are meant to make computations possible for larger structured planning problems.

The first part of this chapter proposes a definition for the *factored π-MOMDPs*: additional independence assumptions are provided to these processes – in a qualitative possibilistic sense, as defined in the first chapter. Large planning problems satisfying these assumptions can be solved more easily: building upon the probabilistic SPUDD algorithm [75], we conceived a possibilistic algorithm named PPUDD for solving factorized π-MOMDPs using *Algebraic Decision Diagrams* (ADD). The guess motivating this contribution is that computations between ADDs is less time and memory consuming when performed in the possibilistic framework than in the probabilistic one: qualitative operations should lead to smaller ADDs when sum and product are replaced with min and max operators because the formers produce ADDs with potentially more leaves.

Independence assumptions defining a factored π-MOMDP concern the variables encoding successive belief states. Moreover variables defining a factored π-MOMDP are those representing successive system states and observations. That is why, the following section of this chapter exhibits sufficient conditions on the system state and observation variables leading to the desired independence between belief state variables. A robotic example is used as an illustration of these conditions. As proofs use the graphical concept called *d-Separation* [141], these conditions also lead to the belief variable independence in the probabilistic MOMDP framework.

Performances of our solver PPUDD are next compared to those of its probabilistic counterparts, in terms of computation time, and with quality criteria measuring some mission achievement metrics. Finally, the last part of this chapter describes the results of PPUDD at the 2014 International Probabilistic Planning Competition[1]. We participated in the competition in order to test the performances of our algorithm against probabilistic ones, in terms of expected reward gathering, with various probabilistic planning problems.

Some contributions of this chapter have been published in [50]. The various planning problems of the competition also highlight some issues of our algorithm when used to find an approximate strategy for a probabilistic problem in order to benefit from qualitative computations which are simpler. Although modeling points are part of these issues, the next chapter shows that a possibilistic approach is very useful where only qualitative data are available. The approach proposed in the final chapter takes into account the highlighted modeling issues. Moreover, time can be entirely consumed before actual computations of PPUDD begin, when loading ADDs encoding some high dimensional planning problems: memory issues are also observed in practice. MDP algorithms using state space search [79, 83] do not face these issues due to ADD-encoding of MDP instances. They can be used to solve the problem resulting from the last chapter. The latter focuses on the belief state management, and proposes a hybrid POMDP technically resulting in a classical probabilistic MDP.

A qualitative possibilistic process for human-machine interaction modeling

The *fourth chapter* deals with situations where probability distributions are clearly not available: the managed problem comes from the work [110] and is a joint work with the author, Sergio Pizziol, working in the field of *Human-Machine Interactions* (HMI). In systems representing human behavior in certain situations, for instance interacting with a control panel of an aerial vehicle, a sufficient amount of statistical data is lacking, especially concerning the human operator's state of mind: only expert knowledge can be used in practice as no frequentist information about the problem is available.

In this chapter, used processes are similar to those studied in the previous chapters: the only difference is that actions are not chosen anymore, but used as observations to infer the state of the human-machine system. This process is called *qualitative possibilistic Hidden Markov Process* (π-HMP).

This process is used as a tool to produce diagnosis for HMI systems: machine state and human actions are called *occurrences*, and the possible transitions of the system are called *effects*. After the model construction describing the problem in terms of occurrences and effects from expert qualitative data and the machine model, the human assessment of the situation can be estimated. The proposed model can also detect human assessment errors, and produce a diagnosis for them.

[1] `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`

Probabilistic-possibilistic approach: a hybrid perspective

The last chapter of this thesis persists to show the possible improvement of the POMDP framework using the qualitative possibility theory, especially for the belief state management. It also takes into account the issues highlighted in the third chapter.

Indeed, the *fifth chapter* argues for a hydrid POMDP with both probabilistic and possibilistic settings coherently mixed up. A new translation from POMDPs into Fully Observable MDPs is described here. Unlike the classical translation presented in the first chapter, the resulting problem state space is finite, making probabilistic MDP solvers able to solve this simplified version of the initial partially observable problem. Indeed, this approach encodes agent beliefs with possibility distributions over states, leading to an MDP whose state space is a finite set of epistemic states.

Additional simplifications of the computations are described for *factored POMDPs* [133, 148, 143, 94], *i.e.* POMDPs with a particular independence structure. These last contributions have been published in [48].

State of the Art

I

The main topic of this thesis is Partially Observable Markov Decision Processes (POMDPs). The practical use of this model has been criticized in Introduction, however it sums up accurately the principal features of a robotic system. As we ambition to make this thesis mathematically self-contained the POMDP model is built in Section I.1 from low level objects of Probability Theory. The main ways to compute strategies from this model are then summarized in Section I.1.11. Next, Possibility Theory is presented in order to introduce Qualitative Possibilistic Markov Decision Processes (π-MDPs) and Partially Observable ones (π-POMDPs) which are the starting point of this work.

I.1 From Markov Chains to Partially Observable Markov Decision Processes

The first theoretical object behind the POMDP model, as its name suggests, is the *Markov Chain*. In order to present it, let us look back one century ago.

I.1.1 Markov Chains

In the early years of the twentieth century Andreï Markov, a doctoral student of Pafnouti Tchebychev, set up Markov Chains. Studying successive letters in the words of novels, he had the idea to define this kind of sequence of random variables: indeed, each letter depended primarily on the previous one. As usual, a random variable is a measurable function defined on a set Ω equipped with a σ-algebra $\mathcal{F}$ and a probability measure $\mathbb{P}$.

Definition I.1.1 (*Markov Chain*)

Let $\mathcal{S}$ be a countable set called set of states and $(S_t)_{t\in\mathbb{N}}$ a sequence of random variables whose values are in $\mathcal{S}$. The sequence $(S_t)_{t\in\mathbb{N}}$ is a Markov Chain if $\forall t \geqslant 1, \forall (s_0, s_1, \ldots, s_{t+1}) \in \mathcal{S}^{t+2}$

$$\mathbb{P}\left(S_{t+1} = s_{t+1} \mid S_0 = s_0, S_1 = s_1, \ldots, S_t = s_t\right) = \mathbb{P}\left(S_{t+1} = s_{t+1} \mid S_t = s_t\right) \tag{I.1}$$

i.e. $\forall t \geqslant 1$, the random variable S_{t+1} is independent from all previous variables $\{S_i \mid i \leqslant t-1\}$ conditional on the random variable S_t: the value of S_0, or its probability distribution is given, and the probability distribution of S_{t+1} only depends on the value of S_t and on the time step $t \geqslant 0$ (see Figure I.1).

Figure I.1 describes the Bayesian Network [105, 104] of a Markov Chain. In a Bayesian Network, or directed acyclic graphical model, the variables are represented by nodes. The absence of an arrow between two random variables (nodes) represents an assumption about the conditional independence of the variables. Let S' be a random variable: the set of variables from which an arrow starts and leads to S' is called the set of *parents* of S', and denoted by $parents(S') =$

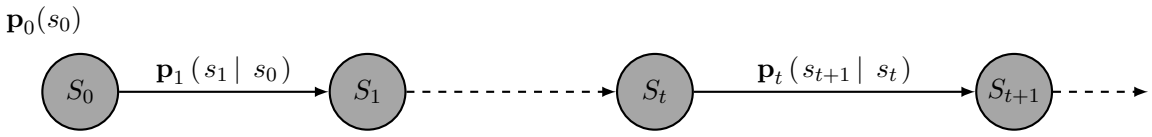


Figure I.1 – *Bayesian Network of a Markov Chain: each node (black circle) represents a random variable. Each variable S_{t+1} has only one parent S_t:* $\forall t \geqslant 1 \; S_{t+1} \perp\!\!\!\perp \{S_0, \ldots, S_{t-1}\} | S_t$.

$\{S \mid S \to S'\}$. If $S \in parents(S')$, we say that S' is a *children* of S: the set of the children of S is denoted by $children(S)$. The set of the *descendants* of a random variable S is the smallest set of variables $descend(S)$ which contains all the children of S and such that $\forall S' \in descend(S)$, $children(S') \subset descend(S)$ *i.e.* all the children of a descendant is a descendant. The assumption taken through a Bayesian Network is that each variable S is independent from its *non-descendants* $nondescend(S) = \left\{\tilde{S} \notin descend(S) \cup \{S\} \cup parents(S)\right\}$ conditional on its parents $parents(S)$, denoted by:

$$S \perp\!\!\!\perp nondescend(S) \mid parents(S).$$

In other words, S only depends on its direct parents. Figure I.1 is also called Dynamic Bayesian Networks (DBNs) [42] since the arrows model also successive time steps.

Here, the *Markov Property i.e.* Equation I.1 of Definition I.1.1, implies that $\forall t \in \mathbb{N}$, the variable S_{t+1} has only one arrow pointing to it from S_t *i.e.* the variable S_{t+1} has only one parent in the Network, which is S_t.

As $\mathcal{S}$ is countable, let us number its elements, the *states*: $\mathcal{S} = \left\{s^{(1)}, s^{(2)}, \ldots\right\}$. The probabilistic dynamics of a Markov Chain can be represented by a sequence of stochastic matrices[1] $(M_t)_{t\in\mathbb{N}}$ defined by $(M_t)_{i,j} = \mathbb{P}\left(S_{t+1} = s^{(j)} \,\middle|\, S_t = s^{(i)}\right)$. Note that the model is entirely defined given this sequence of matrices and the distribution of the first random variable S_0: $\mathbf{p}_0(s) = \mathbb{P}(S_0 = s)$. For a better readability, $\mathbb{P}(S_{t+1} = s' \mid S_t = s)$ is denoted by $\mathbf{p}_t(s' \mid s)$, and called the *transition probability distribution.*

An important result about the Markov Chains is the following:

Property I.1.1

Let $(S_t)_{t\in\mathbb{N}}$ be a Markov Chain whose values are in the countable set of states $\mathcal{S}$.
$\forall t \in \mathbb{N}$, $\forall f : \mathcal{S} \to \mathbb{R}$ bounded, $\forall (s_0, \ldots, s_t) \in \mathcal{S}^{t+1}$,

$$\begin{aligned}\mathbb{E}\left[f(S_{t+1}) \mid S_0 = s_0, \ldots, S_t = s_t\right] &= \sum_{s'\in\mathcal{S}} f(s') \cdot \mathbb{P}\left(S_{t+1} = s' \mid S_t = s_t\right) \\ &= \mathbb{E}\left[f(S_{t+1}) \mid S_t = s_t\right].\end{aligned}$$

The proof is given in Annex A.2.

This result will help rigorously set up Markov Decision Processes, presented in the next section.

I.1.2 Markov Decision Processes

MDPs [115] were proposed was built to model *systems* subject to a probabilistic uncertainty dependent on *actions* over *time*. In its classical formulation, the finite set $\mathcal{S}$ defines the possible *states of the system* $s \in \mathcal{S}$. Here, for the needs of the POMDP model building in a following section, $\mathcal{S}$ is defined as a countable set of states, as stated earlier. The set of the non-negative

[1] A stochastic matrix is a matrix whose values are non-negative real numbers and whose rows sum to one.

integers $\mathbb{N}$ models the *time*, or *stages of the process*. Possible *actions*, denoted by a, are chosen from a finite set $\mathcal{A}$. In order to get a clear overview of what information is used to decide each action, the *agent* is defined as the entity responsible for decision making, *i.e.* she/he must choose the successive actions given the current information about the system. In our case, a system state $s \in \mathcal{S}$ consists of the features of both a robot and its environment, needed to describe the robotic mission. The agent is in this case the decision making part of the robot who determines the robot's actions to be exectuted given the successive states of the system.

If the sequence of chosen actions $a \in \mathcal{A}$ is known, an MDP is a Markov Chain: the system state of an MDP at stage $t+1$, represented by the variable S_{t+1}, only depends, in the probabilistic meaning of the term, on the previous system state variable S_t and on the time step $t \geqslant 0$. However, the agent influences the probabilistic system dynamics by selecting the actions $a \in \mathcal{A}$ at each time step $t \in \mathbb{N}$.

In the MDP framework, it is assumed that the agent is perfectly informed of the current system state $s_t \in \mathcal{S}$ at each time step. Its decision can then be *a priori* based on the current system state and all previous ones: Theorem 1 shows however that, for the used criterion, it is equivalent to consider that each decision is taken based on the current state only. A stochastic matrix M^d can be defined for each *decision rule* $\left\{ \begin{array}{lll} d: & \mathcal{S} \to & \mathcal{A} \\ & s \mapsto & d(s) \end{array} \right.$ describing transition probability distributions of the Markov Chain $(S_t)_{t\in\mathbb{N}}$ if the decision rule d is used: $\left(M^d\right)_{i,j}$ is the probability of reaching the state $s^{(j)}$ from $s^{(i)}$ when the action $d(s^{(i)})$ is selected by the agent. Consider a sequence of decision rules $(d_t)_{t\in\mathbb{N}}$: such a sequence is called *strategy* and defines a sequence of stochastic matrices $(M_t)_{t\in\mathbb{N}}$ with $M_t = M^{d_t}$. Each strategy thus defines the parameters of a Markov Chain entirely.

A reward $r_t(s,a) \in \mathbb{R}$ is associated to each triple $(s,a,t) \in \mathcal{S}\times\mathcal{A}\times\{0,\ldots,H-1\}$ to model the importance of going through the state s and selecting action a at time t. The function $r : \mathcal{S}\times\mathcal{A}\times\{0,\ldots,H-1\} \to \mathbb{R}$ is assumed to be bounded: this assumption is not necessary if $\mathcal{S}$ is finite, as $r_t(s,a) \leqslant \max\limits_{s'\in\mathcal{S}} \max\limits_{a'\in\mathcal{A}} \max\limits_{0\leqslant t'\leqslant H-1} \{ r_{t'}(s',a') \}$, however, $\mathcal{S}$ is here countable and thus may be infinite. The value of a system finishing in state s is described by the terminal reward $R(s)$, defined for each state s. The function R is assumed to be bounded on $\mathcal{S}$.

For instance in the Navigation problem of the International Probabilistic Planning Competition, [127], a robot in a grid has to reach a goal: if the state s encodes a robot location which is not the goal, $r_t(s,a) = -1$, and $r_t(s,a) = 0$ otherwise. If the designers of the model want the robot to be in a sleep mode at the end of the mission, terminal reward could have been defined as $R(s) = 1$ if the system state s encodes the sleep mode, and $R(s) = 0$ otherwise. If some of the moving actions of the robot were more energy demanding than others, for all $s \in \mathcal{S}$ the functions $a \mapsto r_t(s,a)$ may be non-constant, etc.

Solving the optimal control problem for an MDP with horizon $H \in \mathbb{N}$, *i.e.* for a process whose number of stages is H, consists in finding a strategy maximizing the expectation of the sum of the rewards, or *expected total reward*:

$$V_H\Big(s, (d_t)_{t=0}^{H-1}\Big) := \mathbb{E}\left[\sum_{t=0}^{H-1} r_t\Big(S_t, d_t(S_t)\Big) + R(S_H) \,\middle|\, S_0 = s \right]. \tag{I.2}$$

The set of strategies of horizon $H \in \mathbb{N}$ *i.e.* the sequence of H decision rules d_t numbered from 0 to $H-1$ is denoted by $\mathcal{D}_H$. The criterion V_H is a function of the initial state $s \in \mathcal{S}$, the horizon size $H \in \mathbb{N}$ and the strategy $(d_t)_{t=0}^{H-1} \in \mathcal{D}_H$: this function is called *value function*.

I.1.3 Dynamic Programming

In 1949, the Research ANd Development (RAND) corporation hired Richard Bellman to work on multi-stage decision processes. Richard Bellman found a really efficient method called *Dynamic Programming* [8] to solve a class of problems by decomposing them into several simpler subproblems. The optimal control of MDPs, *i.e.* the computation of a strategy maximizing the expected total reward, is part of this class, and is classically performed using Dynamic Programming [115].

The notation (d) is used from now on to represent a strategy $(d_t)_{t=0}^{H-1} \in \mathcal{D}_H$. Theorem 1 highlights the opportunity to use Dynamic Programming in order to compute the optimal value function:

$$V_H^*(s) = \sup_{(d)\in\mathcal{D}_H} V_H\Big(s,(d)\Big) \tag{I.3}$$

and a strategy $(d^*) = (d_t^*)_{t=0}^{H-1} \in \mathcal{D}_H$ such that $\forall s \in \mathcal{S}$, $V_H^*(s) = V_H\Big(s,(d^*)\Big)$. As shown in the proof of the following Theorem 1, this strategy exists, and it is thus possible to rewrite the optimal value function I.3, as follows: $V_H^*(s) = \max_{(d)\in\mathcal{D}_H} V_H(s,(d))$.

At step $0 \leqslant t < H$, the probability that the system state becomes $s' \in \mathcal{S}$ from state $s \in \mathcal{S}$ when the agent selects action $a \in \mathcal{A}$ is denoted by $\mathbf{p}_t(s' \mid s,a) = \mathbb{P}(S_{t+1} = s' \mid S_t = s, a)$. However, we assume that $\forall t \in \{0,\dots,H-1\}$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, the support of the transition probability distributions $\mathbf{p}_t(.\mid s,a)$ is finite, *i.e.* $\exists \mathcal{S}_{s,a,t} \subset \mathcal{S}$, such that $\#(\mathcal{S}_{s,a,t}) < +\infty$, and $\forall s' \in \mathcal{S} - \mathcal{S}_{s,a,t}$, $\mathbf{p}_t(s' \mid s,a) = 0$. For $i \in \{0,\dots,H-1\}$, the partial value function V_i is defined as the value function for the process starting at time step $t = H - i$, and $V_i^*(s') = \max_{(d_t)_{t=H-i}^{H-1}\in\mathcal{D}_i} V_i(s')$.

Theorem 1

The following Dynamic Programming equations for an horizon $H \in \mathbb{N}$ computes successive functions V_i^* for each $\forall 0 \leqslant i \leqslant N$ and an optimal strategy (d^*): $\forall s \in \mathcal{S}$,

$$\begin{aligned} V_0^*(s) &= R(s), \\ \text{and } \forall 1 \leqslant i \leqslant H, \quad V_i^*(s) &= \max_{a\in\mathcal{A}} \left\{ r_{H-i}(s,a) + \sum_{s'\in\mathcal{S}_{s,a,t}} \boldsymbol{p}_{H-i}(s' \mid s,a)\, V_{i-1}^*(s') \right\}. \end{aligned}$$

As well, $\forall 1 \leqslant i \leqslant H, \forall s \in \mathcal{S}$,

$$d_{H-i}^*(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \left\{ r_{H-i}(s,a) + \sum_{s'\in\mathcal{S}_{s,a,t}} \boldsymbol{p}_{H-i}(s' \mid s,a)\, V_{i-1}^*(s') \right\}.$$

The proof of this theorem is given in Annex A.3 and uses Property I.1.1.

If the state space $\mathcal{S}$ is finite, Algorithm 1 solves the optimal control problem of a Markov Chain, *i.e.* the MDP problem: computations are performed recursively, using the Dynamic Programming equations.

I.1.4 Infinite Horizon MDP

The previous section was devoted to present the finite horizon version of the MDP framework. In some situations, designers of the MDP model do not know when the process will end, or they want the agent to control the system forever: the MDP problem has to be expressed whatever the horizon H, or, more generally, for an infinite horizon. For instance, when the

Algorithm 1: Dynamic Programming Algorithm for finite state space MDP

$V_0^* \leftarrow R$;
for $i \in \{1, \ldots, H\}$ **do**
  for $s \in \mathcal{S}$ **do**
    $V_i^*(s) \leftarrow \max_{a \in \mathcal{A}} \left\{ r_{H-i}(s,a) + \sum_{s' \in \mathcal{S}} \mathbf{p}_{H-i}(s' \mid s,a)\, V_{i-1}^*(s') \right\}$;
    $d_{H-i}^*(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \left\{ r_{H-i}(s,a) + \sum_{s' \in \mathcal{S}} \mathbf{p}_{H-i}(s' \mid s,a)\, V_{i-1}^*(s') \right\}$;
return V_H^*, $(d^*)_{t=0}^{H-1}$;

MDP models a robotic mission, it sounds better not to bound the time of the process in order to let the mission be completed, in case of delay.

An infinite horizon MDP is defined by the 6-tuple $\langle \mathcal{S}, \mathcal{A}, T, r, s_0, \gamma \rangle$:

- $\mathcal{S}$ is the countable set of potential states of the system, *i.e.* states of the agent and its environment;
- $\mathcal{A}$ is the finite set of actions which can be selected by the agent;
- T is the set of transition probability distributions: $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, T contains the probability distribution over the reached state $s' \in \mathcal{S}$ from system state s and selecting action a, *i.e.* the function $s' \mapsto \mathbf{p}(s' \mid s,a) \in T$ defined on $\mathcal{S}$. Note that in this formulation, the probability distribution is stationary *i.e.* it does not depend on the stage of the process: $\forall t \in \mathbb{N}$, $\mathbf{p}_t(s' \mid s,a) = \mathbf{p}(s' \mid s,a)$. As previously, we assume that $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, the support of the transition probability distributions $\mathbf{p}(. \mid s,a)$ is finite, *i.e.* $\exists \mathcal{S}_{s,a} \subset \mathcal{S}$, such that $\#(\mathcal{S}_{s,a}) < +\infty$, and $\forall s' \in \mathcal{S} - \mathcal{S}_{s,a}$, $\mathbf{p}(s' \mid s,a) = 0$;
- $r(s,a)$ the reward obtained when the agent selects action $a \in \mathcal{A}$ and the system is in state $s \in \mathcal{S}$. This function is assumed to be bounded on $\mathcal{S} \times \mathcal{A}$. Note that the reward is stationary too here: $\forall s \in \mathcal{S}, a \in \mathcal{A}$, $\forall t \in \mathbb{N}$, $r_t(s,a) = r(s,a)$;
- the initial state $s_0 \in \mathcal{S}$, is the state where the process begins;
- the discount factor γ, a real number such that $0 < \gamma < 1$.

Consider a *plan i.e.* a sequence of actions indexed by the stage of the process $t \in \mathbb{N}$: $(a_t)_{t \in \mathbb{N}}$. The system is initially in state s_0. Next, at each step of the process ($t = 0, 1, \ldots$), the system is in state $s_t \in \mathcal{S}$, the agent selects action $a_t \in \mathcal{A}$ and the system reaches a state $s_{t+1} \in \mathcal{S}$ according to the transition probability distribution $\mathbf{p}(. \mid s_t, a_t) = \mathbb{P}(S_{t+1} = s' \mid S_t = s_t, a_t)$. Finally, the agent gets the reward $r(s_t, a_t) \in \mathbb{R}$ which is aggregated to the previous ones with a sum. Figure I.2 graphically sums up the stationary MDP model: the presented figure is an Influence Diagram, *i.e.* it represents the relations between successive variables, just like a Dynamic Bayesian Network, but also including actions and rewards.

As the horizon is infinite, the set of infinite sequences of decision rules, or the set of strategies, is denoted by $\mathcal{D}_\infty = \{ (d_t)_{t \in \mathbb{N}} \mid \forall t \in \mathbb{N}, d_t : \mathcal{S} \to \mathcal{A} \}$. The *discounted reward* at time step t is $\gamma^t \cdot r(s_t, a_t)$, with $0 < \gamma < 1$. The real number γ makes the total sum of discounted rewards converge: $\sum_{t=0}^{+\infty} \gamma^t r(s_t, d_t(s_t))$ whatever the trajectory $(s_t)_{t \in \mathbb{N}} \in \mathcal{S}^{\mathbb{N}}$ and the strategy $(d_t)_{t \in \mathbb{N}} \in \mathcal{D}_\infty$. Indeed, as the reward function r is bounded, $\exists c > 0$ such that $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$,

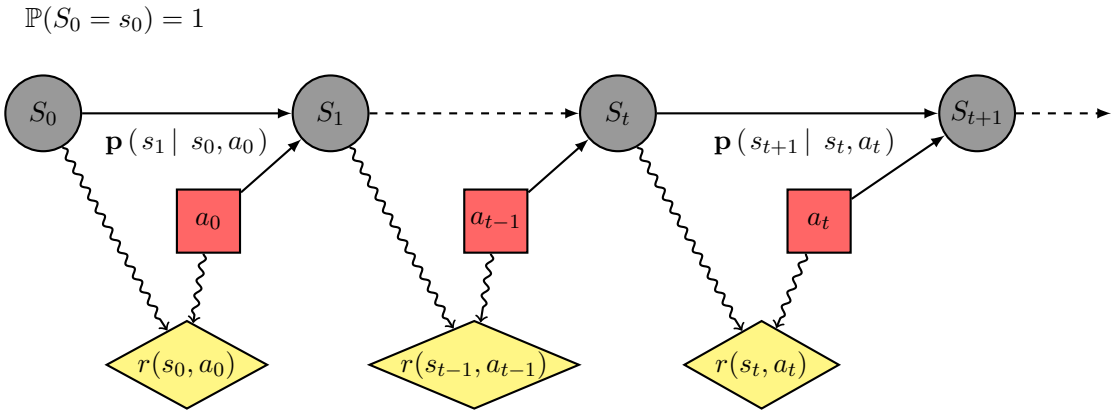


Figure I.2 – *Influence Diagram of an MDP: black circles represent successive system states, red squares are selected actions, and yellow diamonds are the rewards. The Bayesian Network resulting from removing rewards and wavy arrows asserts that* $\forall t \geqslant 1$, $S_{t+1} \perp\!\!\!\perp \{S_0, \dots, S_{t-1}\} \,|\, \{S_t, A_t\}$, *where* A_t *represents the action at time step* t *seen as a random variable.*

$r(s,a) < c$. Thus $\sum_{t=0}^{+\infty} \gamma^t r\Big(s_t, d_t(s_t)\Big) \leqslant c \sum_{t=0}^{+\infty} \gamma^t \leqslant \dfrac{c}{1-\gamma} < +\infty$. The discount factor γ can be defined as the probability that the process goes on, *i.e.* does not terminate, after each stage $t \in \mathbb{N}$: at each time step t, whatever the current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, the process has a probability $1-\gamma$ to reach an artificial absorbing and reward-free state s_{end} modeling the end of the process, *i.e.* $\mathbb{P}(S_{t+1} = s_{end} \mid S_t = s, a) = 1-\gamma$, with $\forall a \in \mathcal{A}$, $r(s_{end}, a) = 0$ and $\mathbb{P}(S_{t+1} = s_{end} \mid S_t = s_{end}, a) = 1$. If such a terminal state s_{end} does not make sense in the particular context, the discount factor γ models only the fact that long term rewards are less important than short term ones.

Thus, once a problem has been defined in terms of an MDP, solving the problem consists in computing a strategy maximizing the expected sum of the discounted rewards. This quantity, called as previously the value function, is denoted by V and depends on the initial state and the strategy $(d) \in \mathcal{D}_\infty$:

$$V\Big(s, (d)\Big) := \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\Big(S_t, d_t(S_t)\Big) \,\middle|\, S_0 = s\right]. \tag{I.4}$$

Note that, for a given strategy, we denote by $V^d : \mathcal{S} \to \mathbb{R}$ the function such that $V^d(s) = V\Big(s, (d)\Big)$.

I.1.5 The Value Iteration algorithm

The Value Iteration algorithm is a solver for Infinite Horizon MDPs: it is based on the Bellman equation, which is derived in Annex A.4. Let $(d)_{t\in\mathbb{N}} \in \mathcal{D}_\infty$:

Definition I.1.2 (*Bellman Equation*)

$$V^d(s) = r\Big(s, d_0(s)\Big) + \gamma \sum_{s' \in \mathcal{S}_{s,d_0(s)}} \boldsymbol{p}\Big(s' \Big| s, d_0(s)\Big) V^{d^+}(s'). \tag{I.5}$$

where $\forall t \in \mathbb{N}$, $\forall s \in \mathcal{S}$, $d_t^+(s) = d_{t+1}(s)$.

The set of bounded functions defined on $\mathcal{S}$ and with real values, is denoted by $\mathcal{F}_B(\mathcal{S},\mathbb{R})$. The Bellman operator is defined as

$$\begin{array}{rccccc} \mathcal{B}^d: & \mathcal{F}_B(\mathcal{S},\mathbb{R}) & \longrightarrow & \mathcal{F}_B(\mathcal{S},\mathbb{R}) & & \\ & V & \longmapsto & s & \longmapsto & r\Big(s,d_0(s)\Big)+\gamma\sum_{s'\in\mathcal{S}_{s,d(s)}}\mathbf{p}\Big(s'\Big|s,d_0(s)\Big)V(s'). \end{array}$$

The Bellman equation I.5 can then be written

$$V^d=\mathcal{B}^dV^{d^+}. \tag{I.6}$$

Consider the sup norm on $\mathcal{F}_B(\mathcal{S},\mathbb{R})$: $\|V\|_\infty=\sup_{s\in\mathcal{S}}|V(s)|$.

Theorem 2

$(\mathcal{F}_B(\mathcal{S},\mathbb{R}),\|.\|_\infty)$ is a Banach space.

The proof is given in Annex A.5.

Property I.1.2

$\mathcal{B}^d$ is a contracting operator of $(\mathcal{F}_B(\mathcal{S},\mathbb{R}),\|.\|_\infty)$ whose contraction is γ.

The proof is given in Annex A.6.

If the strategy (d) is stationary, *i.e.* $\forall t\in\mathbb{N}$, $\forall s\in\mathcal{S}$, $d_t(s)=d(s)$, then $V^d=V^{d^+}$, and the Bellman equation I.6 becomes $V^d=\mathcal{B}^dV^d$. The Fixed-Point Theorem assures that the equation $V=\mathcal{B}^dV$ has a unique solution, and it is thus the value function $V^d\in\mathcal{F}_B(\mathcal{S},\mathbb{R})$. This equation is a characteristic equation of the value function V^d for a given stationary strategy (d). It is worth mentioning that it is also a linear invertible system whose analytical solution is known.

In order to simplify the next formulae, the operator $\mathcal{B}^a$, for $a\in\mathcal{A}$, is introduced:

$$\begin{array}{rccccc} \mathcal{B}^a: & \mathcal{F}_B(\mathcal{S},\mathbb{R}) & \longrightarrow & \mathcal{F}_B(\mathcal{S},\mathbb{R}) & & \\ & V & \longmapsto & s & \longmapsto & r(s,a)+\gamma\sum_{s'\in\mathcal{S}_{s,a}}\mathbf{p}\left(s'\mid s,a\right)V(s'). \end{array}$$

Let us also introduce now the Dynamic Programming operator $\mathcal{B}^*$: $\forall V\in\mathcal{F}_B(\mathcal{S},\mathbb{R})$, $\forall s\in\mathcal{S}$,

$$(\mathcal{B}^*V)(s)=\max_{a\in\mathcal{A}}\mathcal{B}^aV(s)$$

Property I.1.3

The Dynamic programming operator $\mathcal{B}^*$ is also a contracting operator of $\mathcal{F}_B(\mathcal{S},\mathbb{R})$ whose contraction is γ.

The proof is given in Annex A.7.

The Fixed-Point Theorem assures that the Dynamic Programming equation

$$V=\mathcal{B}^*V \tag{I.7}$$

has a unique solution in $\mathcal{F}_B(\mathcal{S},\mathbb{R})$, denoted by V^*.

Using this assertion, the following theorem assures that V^* is the optimal value function:

Theorem 3

The solution V^* of the dynamic programming equation (I.7) is equal to the optimal value function

$$V^*(s)=\sup_{d\in\mathcal{D}_\infty}V^d(s)=\sup_{d\in\mathcal{D}_\infty}\mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t\cdot r\Big(S_t,d_t(S_t)\Big)\;\middle|\;S_0=s\right].$$

Let us define the decision rule $d^* : s \mapsto a^* \in \operatorname{argmax}_{a \in A}(\mathcal{B}^a V^*)(s)$*. The associated stationary strategy, i.e. the strategy* $(d_t^*)_{t\in\mathbb{N}} \in \mathcal{D}_\infty$ *such that* $\forall t \in \mathbb{N}$, $\forall s \in \mathcal{S}$, $d_t^*(s) = d^*(s)$ *is optimal for the criterion (I.4).*

The proof is given in Annex A.8. Note that the d^* maximizes the criterion also among all history-dependent strategies, *i.e.* strategies depending on all the previous system states. Indeed, it can be shown as in the proof of Theorem 1 (*i.e.* using Property I.1.1) that the value function (I.4) from a given time step only depends on the current system state and not on the previous ones: thus there is no need to adapt the returned action with respect to the previous system states.

If the state space $\mathcal{S}$ is finite, the *Value Iteration Algorithm* 2, directly derived from the Fixed-Point Theorem, leads to the computation of the optimal value function. This theorem assures indeed that $\forall V^0 \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})$, $(\mathcal{B}^*)^n V^0 \longrightarrow V^*$, in the sense of the norm $\|.\|_\infty$, when $n \to +\infty$.

Algorithm 2: Value Iteration Algorithm for MDP

$N \leftarrow$ number of iterations;
$V \leftarrow V^0$;
$i \leftarrow 1$;
while $i \leqslant N$ **do**
  for $s \in \mathcal{S}$ **do**
    $V'(s) \leftarrow \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathbf{p}\left(s' \mid s,a\right) V(s') \right\} \quad \Big(= (\mathcal{B}^* V)(s) \Big)$;
  $V \leftarrow V'$;
  $i \leftarrow i + 1$;
for $s \in \mathcal{S}$ **do**
  $d(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathbf{p}\left(s' \mid s,a\right) V(s') \right\}$;
return V, d

Error analysis

Let us denote by $(V^i)_{i\geqslant 0}$ the successive functions computed by Algorithm 2 presented thereafter. The optimal value function is denoted by V^*. First, the following theorem informs us about the convergence of V^N:

Theorem 4

$$\left\|V^N - V^*\right\|_\infty \leqslant \frac{\gamma^N}{1-\gamma} \cdot \left\|V^0 - V^1\right\|_\infty . \tag{I.8}$$

The proof is given in Annex A.9.

However, we are more interested in an error bound of V^d, where (d) is the strategy returned by the algorithm. The control of the strategy error is given by the next theorem:

Theorem 5

$$\left\|V^d - V^*\right\|_\infty \leqslant \frac{2\cdot\gamma^N}{1-\gamma} \left\|V^1 - V^0\right\|_\infty .$$

The proof is given in Annex A.10.

The number of iteration N can be set up using this bound[2]: if the desired maximal error is $\varepsilon > 0$, N has to be greater than $\log\left(\frac{\varepsilon(1-\gamma)}{2\|V^1-V^0\|_\infty}\right) / \Big(\log(\gamma)\Big)$.

Another classical way to solve an MDP is the use of the *Policy Iteration algorithm*, also called Howard's algorithm [115]: this algorithm converges in a finite number of iterations but implies to solve at each iteration the linear system (I.6) with $d^+ = d$.

I.1.6 Partially Observable Markov Decision Process

POMDPs generalize MDPs allowing the agent to misperceive, or partially observe, with a given probability, the current state of the system: the system state $s \in \mathcal{S}$ is not given as input to the agent. The latter has to figure it out using the observations $o \in \mathcal{O}$ of the system received at each time step.

Definition

A POMDP, in the infinite horizon settings, is defined by a 8-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, r, b_0, \gamma >$:

- $\mathcal{S}$ the **finite** set of system states: the current state s_t is not given as input to the agent;
- $\mathcal{A}$ the finite set of actions which can be selected by the agent;
- $\mathcal{O}$ the finite set of observations that the agent may receive: the current observation $o_t \in \mathcal{O}$ is given as input to the agent;
- T the set of transition probability distributions of the system state: $\forall s \in \mathcal{S}, a \in \mathcal{A}$, probability distributions $\mathbf{p}(.\mid s, a) \in T$ are defined on $\mathcal{S}$;
- O, the set of observation probability distributions, *i.e.* the set of probability distributions over $\mathcal{O}$, defining the probability that the agent observes o', after selecting action a, if the system has reached the state s': $\forall s' \in \mathcal{S}$, $a \in \mathcal{A}$, probability distributions $\mathbf{p}(.\mid s', a) \in O$ are defined over the set of observations $\mathcal{O}$;
- $r(s, a)$ the reward gathered by the agent in the situation where it selects action $a \in \mathcal{A}$ when the system is in state $s \in \mathcal{S}$.
- b_0, a probability distribution defining the uncertainty on the initial state represented by the random variable S_0: $S_0 \sim b_0$, *i.e.* $\forall s \in \mathcal{S}$, $b_0(s) = \mathbb{P}(S_0 = s)$. This probability distribution is called *initial belief state* about the system state. It is an *epistemic* probability distribution, because it estimates the actual initial system state, since it is not available for the agent.

Note that the set of system states $\mathcal{S}$ is assumed to be finite here: this is a sufficient assumption for the robotic mission features we would like to model, and a classical way to define POMDPs. Previously, in the Fully Observable case (MDP, see Section I.1.2), the space $\mathcal{S}$ was assumed countable, which is a less restrictive assumption (if $\mathcal{S}$ is countable, it may be infinite): the main results about the POMDP resolution comes from the results of Section I.1.2 about (Fully Observable) MDPs with countable state space $\mathcal{S}$. The finite set of the observations $\mathcal{O}$ and the set of the observation probability distributions O, are both new in the Partially Observable model in comparison with the Fully Observable MDP model. However, the transition

[2]The standard VI algorithm uses the distance between two successive value functions for the stopping criterion (*while* loop): this distance is related to the distance between the current value function and the optimal one using another result of Banach's contraction theorem (see [115]).

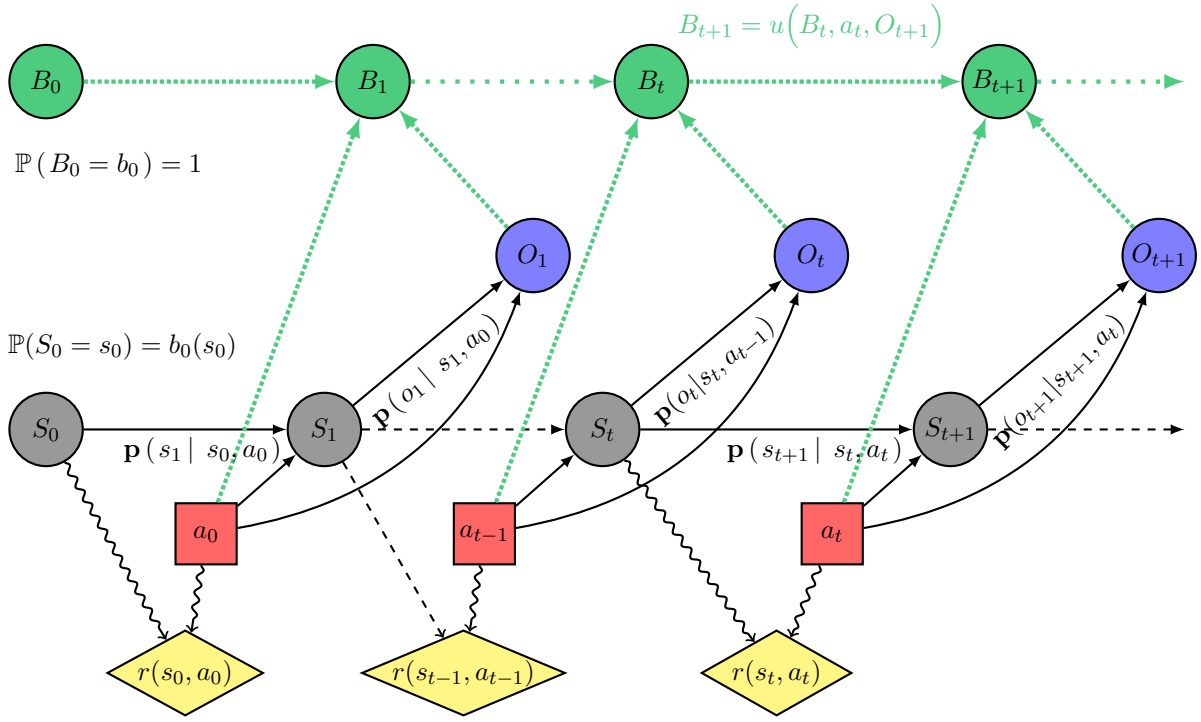


Figure I.3 – *Influence Diagram of a POMDP and its belief updating process: black circles represent successive system states S_t, blue ones represent successive observations O_t, red squares are selected actions a_t, and yellow diamonds are the associated rewards. Green circles at the top of the figure are the successive belief states B_t constituting the belief updating process, computed using the update $B_{t+1} = u(B_t, a_t, O_{t+1})$. Just like the wavy lines leading to rewards, the green dotted lines represent a deterministic influence. The Bayesian Network resulting from removing belief states and rewards, asserts that $\forall t \geqslant 1$, $S_{t+1} \perp\!\!\!\perp \{S_0, S_1, O_1, \ldots, S_{t-1}, O_{t-1}\} \,|\, \{S_t, A_t\}$, where A_t represents the action at time step t seen as a random variable. As well, $\forall t \geqslant 1$, O_t is independent from all other variables conditional on $\{S_t, A_t\}$.*

probability distributions T, the finite set of actions $\mathcal{A}$ and the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, were already introduced in the section about MDPs.

Consider a plan $(a_t)_{t\in\mathbb{N}} \in \mathcal{A}^{\mathbb{N}}$. Initially, the system is in state s_0 with probability b_0. Next, just like the MDP model, at each time step $(t = 0, 1, \ldots)$ the system is in state $s_t \in \mathcal{S}$, the agent selects action $a_t \in \mathcal{A}$ and the system state changes, reaching $s_{t+1} \in \mathcal{S}$ according to the probability distribution $\mathbf{p}(.\mid s_t, a_t)$. The agent gets the reward $r(s_t, a_t)$ and also finally gets the observation o_{t+1} with probability $\mathbf{p}(o_{t+1} \mid s_{t+1}, a_t) = \mathbb{P}(O_{t+1} = o_{t+1} \mid S_{t+1} = s_{t+1}, a_t)$: for each time step $t \geqslant 1$, the observation random variable O_t is such that $O_t \sim \mathbf{p}(.\mid s_t, a_{t-1})$, where $\mathbf{p}(.\mid s_t, a_{t-1}) \in O$ is the probability distribution over the observations. The bottom of Figure I.3 fully sums up the definition of this process: the *belief updating process* in green at the top of this figure is defined in the next section.

I.1.7 The belief updating process

The independence assumptions highlighted by the Bayesian Network (as a first step, the black straight-lined arrows and associated nodes, and finally, the dashed green ones too) of Figure I.3, assert that the probability distributions defining the POMDP fully describe the uncertainty related to the system: at time step $t \geqslant 1$, the probability that the t first observations are $(o_1, \ldots, o_t)$ and the $t+1$ first system states are $(s_0, \ldots, s_t)$ conditional on the t first selected

actions $(a_0, \ldots, a_{t-1})$ is

$$\begin{aligned}\mathbb{P}\left(S_0 = s_0, \ldots, S_t = s_t, O_1 = o_1, \ldots, O_t = o_t \mid a_0, \ldots, a_{t-1}\right) \\ = b_0(s_0) \cdot \prod_{i=1}^{t-1} \mathbf{p}\left(s_{i+1} \mid s_i, a_i\right) \cdot \prod_{i=1}^{t-1} \mathbf{p}\left(o_{i+1} \mid s_{i+1}, a_i\right).\end{aligned}$$

Let us define the belief updating process which is classically a basis for the strategy computation: at time step $t \in \mathbb{N}$, if the observation sequence from the beginning is $(o_1, \ldots, o_t) \in \mathcal{O}^t$ given as input to the agent, and if the successive selected actions are $(a_0, \ldots, a_{t-1}) \in \mathcal{A}^t$, the *belief state* at time step t is the probability distribution of the system state conditioned on the observation and action sequences.

Definition I.1.3 (*Belief State and Information*)

The ***belief state*** *at time step t is the function $b_t : \mathcal{S} \to \mathbb{R}$ defined as*

$$b_t(s) = \mathbb{P}\left(S_t = s \mid O_1 = o_1, \ldots, O_t = o_t, a_0, \ldots, a_{t-1}\right) = \mathbb{P}\left(S_t = s \mid I_t = i_t\right), \tag{I.9}$$

where $i_t = \{o_1, \ldots, o_t, a_0, \ldots, a_{t-1}\}$, is the ***information*** *gathered by the agent at time step t. The random variable version of i_t is denoted by $I_t = \{O_1, \ldots, O_t, a_0, \ldots, a_{t-1}\}$.*

The belief state at time step t is thus the *a posteriori* probability distribution of the system state, given the initial probability distribution b_0 and the probability distributions O and T defining the POMDP, and conditional on the information i_t. The belief process, which is the sequence of belief states, can be computed recursively using Bayes rule.

Theorem 6

If the belief state at time step t is b_t, the selected action is $a_t \in \mathcal{A}$, and the next observation is o_{t+1}, the next belief state b_{t+1} is computed as follows:

$$b_{t+1}(s') = \frac{\displaystyle\sum_{s \in \mathcal{S}} \boldsymbol{p}\left(o_{t+1} \mid s', a_t\right) \cdot \boldsymbol{p}\left(s' \mid s, a_t\right) \cdot b_t(s)}{\displaystyle\sum_{\tilde{s} \in \mathcal{S}} \sum_{\tilde{s}' \in \mathcal{S}} \boldsymbol{p}\left(o_{t+1} \mid \tilde{s}', a_t\right) \cdot \boldsymbol{p}\left(\tilde{s}' \mid \tilde{s}, a_t\right) \cdot b_t(\tilde{s})}. \tag{I.10}$$

This formula is called the ***belief update****, and since the belief state b_{t+1} is shown to be a function of b_t, a_t and o_{t+1}, we denote it by*

$$b_{t+1} = u(b_t, a_t, o_{t+1}).$$

The proof is given in Annex A.11.

The belief update formula (I.10) is more simply written as

$$b_{t+1}(s') \propto \sum_{s \in \mathcal{S}} \mathbf{p}\left(o_{t+1} \mid s', a_t\right) \cdot \mathbf{p}\left(s' \mid s, a_t\right) \cdot b_t(s)$$

as $\displaystyle\sum_{\tilde{s} \in \mathcal{S}} \sum_{\tilde{s}' \in \mathcal{S}} \mathbf{p}\left(o_{t+1} \mid \tilde{s}', a_t\right) \cdot \mathbf{p}\left(\tilde{s}' \mid \tilde{s}, a_t\right) \cdot b_t(\tilde{s})$ is nothing more than a normalization constant.

Note that b_0 may be seen as the actual value of a random variable B_0, like the value s_0 of the random variable S_0 in the MDP section I.1.4: $\mathbb{P}(B_0 = b_0) = 1$. Now, if B_t is a random variable representing the belief state at time t, $B_{t+1} = u\Big(B_t, a_t, O_t\Big)$ is a random variable as a function of random variables. In the top of Figure I.3 the belief process is represented by the belief state variables $(B_t)_{t \geqslant 0}$: this figure highlights the links between the belief process and the POMDP process (green dotted arrows). More formaly the belief state variable B_t is

$B_t(s) = \mathbb{P}(S_t = s \mid I_t) = \mathbb{E}\left[\mathbb{1}_{\{S_t=s\}} \,\middle|\, I_t\right]$, $\forall s \in \mathcal{S}$, *i.e.* the conditional expectation[3] of the characteristic function of the set $\{S_t = s\} \subseteq \Omega$.

I.1.8 A belief dependent value function

As the agent has only access to the information $i_t = \{o_1, \ldots, o_t, a_0, \ldots, a_{t-1}\}$ at time step $t \geqslant 1$, the sequence of decision rules $(d_t)_{t\in\mathbb{N}}$ is such that $\forall t \in \mathbb{N}$, $d_t : i_t \mapsto a \in \mathcal{A}$. Let us present the criterion, or value function, which has to be maximized by choosing the right strategy: for an initial belief state b_0, which defines the probability distribution of S_0, the value function for an infinite horizon and a strategy based on the current information $(d_t)_{t\in\mathbb{N}}$, is the expected discounted total reward:

$$V^d(b_0) = \mathbb{E}\left[\sum_{t\geqslant 0} \gamma^t \cdot r\Big(S_t, d_t(I_t)\Big)\right]. \tag{I.11}$$

The following work leads to a formulation of the value function where the belief update process $(B_t)_{t\in\mathbb{N}}$ appears. It considers as well the action sequence as a sequence of random variables: $(A_t)_{t\in\mathbb{N}}$. It covers then the case $A_t = d_t(I_t)$ proposed in the value function definition (I.11). The information random variable I_t becomes then $\{O_1, \ldots, O_t, A_1, \ldots A_t\}$. Using Fubini theorem and the linearity of the probabilistic expectation,

$$\begin{aligned}
V^d(b_0) &= \sum_{t\geqslant 0} \gamma^t \cdot \mathbb{E}\left[r(S_t, A_t)\right] \\
&= \sum_{t\geqslant 0} \gamma^t \cdot \mathbb{E}\Big[\mathbb{E}\left[r(S_t, A_t) \mid I_t\right]\Big] \text{ as a consequence of Definition A.1} \\
&= \sum_{t\geqslant 0} \gamma^t \cdot \mathbb{E}\left[\mathbb{E}\left[\sum_{s\in\mathcal{S}} r(s, A_t) \cdot \mathbb{1}_{\{S_t=s\}} \,\middle|\, I_t\right]\right] \\
&= \sum_{t\geqslant 0} \gamma^t \cdot \mathbb{E}\left[\sum_{s\in\mathcal{S}} r(s, A_t) \cdot \mathbb{E}\left[\mathbb{1}_{\{S_t=s\}} \,\middle|\, I_t\right]\right] \qquad \text{(I.12)} \\
&= \sum_{t\geqslant 0} \gamma^t \cdot \mathbb{E}\left[\sum_{s\in\mathcal{S}} r(s, A_t) \cdot B_t(s)\right] \text{ by definition of } B_t \\
&= \sum_{t\geqslant 0} \gamma^t \cdot \mathbb{E}\left[r(B_t, A_t)\right] \qquad \text{(I.13)} \\
&= \mathbb{E}\left[\sum_{t\geqslant 0} \gamma^t \cdot r(B_t, A_t)\right]
\end{aligned}$$

where the belief reward function $r : (b, a) \mapsto r(b, a) := \sum_s r(s, a) \cdot b(s)$ is introduced in line (I.13). Line I.12 uses the linearity of the conditional expectation and Property A.1 (as a function of I_t, $A_t = d_t(I_t)$ is $\sigma(I_t)$-measurable, see Property A.3).

Another way to see that the expectation of the discounted reward is equal to the expected discounted belief reward is to compute it conditionally on the action sequence:

Theorem 7

$$\mathbb{E}\left[r(S_t, A_t) \mid \widehat{A}_t = \widehat{a}_t\right] = \mathbb{E}\left[r(B_t, A_t) \mid \widehat{A}_t = \widehat{a}_t\right]. \tag{I.14}$$

using the notations $\widehat{A}_t = \{A_0, \ldots, A_t\}$ *and* $\widehat{a}_t = \{a_0, \ldots, a_t\}$.

The proof is given in Annex A.12.

[3] The general definition of the probabilistic conditional expectation (as a random variable) is given in Annex, Definition A.1.

I.1.9 A POMDP as a belief-MDP

In this section a POMDP is expressed as a MDP, whose states are the belief states: the resulting MDP is denoted by $\langle \tilde{S}, \mathcal{A}, \tilde{T}, \tilde{r}, \tilde{s_0}, \gamma \rangle$. The state space $\tilde{\mathcal{S}}$ is the set of all reachable belief states from b_0, denoted by $\mathbb{P}^{\mathcal{S}}_{b_0}$. This set is countable: indeed, as $\mathcal{A}$ and $\mathcal{O}$ are finite, each reachable belief state has a finite number of possible successors. As there is only one initial belief state, each set of belief states generated for each time step is finite. Numbering them is thus as easy as numbering each belief state of each successive finite set, following the time index t. The set of all probability distributions is denoted by $\mathbb{P}^{\mathcal{S}}$: thus, $\mathbb{P}^{\mathcal{S}}_{b_0} \subset \mathbb{P}^{\mathcal{S}}$.

Let b_t be a given belief state, *i.e.* a probability distribution in $\mathbb{P}^{\mathcal{S}}_{b_0}$. The sequence $(B_t)_{t\in\mathbb{N}}$ is a sequence of random variables: as highlighted by the belief update (I.10), if $B_t = b_t$, and the selected action is a_t, the value of the next variable B_{t+1} is a deterministic function of the observation O_{t+1}.

Before defining the belief-MDP, the belief process is shown to be a Markov process:

Theorem 8

$\forall a \in \mathcal{A}, \forall b' \in \mathbb{P}^{\mathcal{S}}_{b_0}$,

$$\mathbb{P}\left(B_{t+1} = b' \mid I_t = i_t, a\right) = \mathbb{P}\left(B_{t+1} = b' \mid B_t = b^{i_t}_{b_0}, a\right), \tag{I.15}$$

where $b^{i_t}_{b_0}$ is the current belief state if the initial belief is b_0 and the information gathered is i_t.

The proof is given in Annex A.13.

As highlighted by equation (21) in the proof, if $B_t = b_t$, the probability that the next belief B_{t+1} is b_{t+1}, is the sum of all probabilities of the observations o' such that $u(b_t, a_t, o') = b_{t+1}$, *i.e.* of the observations leading to the belief state b_{t+1}: it defines the transition probability distributions of the belief process, *i.e.* elements of $\tilde{T}$, as follows: $\forall t \geqslant 0$,

$$\begin{aligned}
\mathbf{p}\left(b_{t+1} \mid b_t, a_t\right) &= \mathbb{P}\left(B_{t+1} = b_{t+1} \mid B_t = b_t, a_t\right) \\
&= \sum_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ u(b_t, a_t, o') = b_{t+1}}} \mathbb{P}\left(O_{t+1} = o' \mid B_t = b_t, a_t\right) \\
&= \sum_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ u(b_t, a_t, o') = b_{t+1}}} \sum_{(s,s') \in \mathcal{S}^2} \mathbf{p}\left(o' \mid s', a_t\right) \cdot \mathbf{p}\left(s' \mid s, a_t\right) \cdot b_t(s) \\
&= \sum_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ u(b_t, a_t, o') = b_{t+1}}} \mathbf{p}\left(o' \mid b_t, a_t\right).
\end{aligned} \tag{I.16}$$

where $\sum_{(s,s')\in\mathcal{S}^2} \mathbf{p}\left(o' \mid s', a_t\right) \cdot \mathbf{p}\left(s' \mid s, a_t\right) \cdot b_t(s)$ is denoted by $\mathbf{p}\left(o' \mid b_t, a_t\right)$.

Finally, the reward associated with the belief state b_t is defined as previously

$$\begin{array}{rccc}
\tilde{r}: & \mathbb{P}^{\mathcal{S}}_{b_0} \times \mathcal{A} & \to & \mathbb{R} \\
& (b, a) & \mapsto & \sum_{s\in\mathcal{S}} r(s) \cdot b(s),
\end{array} \tag{I.17}$$

and the initial state denoted by $\tilde{s_0}$ is the belief state b_0.

As $B_0 = b_0$, we can write that $\mathbb{P}(B_0 = b_0) = 1$, and then previous section demonstrates that

$$\mathbb{E}\left[\sum_{t\geqslant 0} \gamma^t \cdot r\Big(S_t, d_t(I_t)\Big)\right] = \mathbb{E}\left[\sum_{t\geqslant 0} \gamma^t \cdot r\Big(B_t, d_t(I_t)\Big) \,\middle|\, B_0 = b_0\right].$$

The right part of the equation is the value function of the MDP $\langle \tilde{S}, \mathcal{A}, \tilde{T}, \tilde{r}, \tilde{s_0}, \gamma \rangle$. As highlighted by Dynamic Programming exposed in the sections I.1.3 and I.1.4, the criterion may

vary for two belief states, but not if the information varies always leading to the same belief state: action sequence has to be chosen as a sequence of functions of the current belief state, *i.e.* a maximizing action sequence $(A_t)_{t\geqslant 0}$ is given by a strategy $(d_t)_{t\geqslant 0}$: $A_t = d_t(B_t)$. As this criterion can be computed directly with the belief-MDP model $\langle \tilde{S}, \mathcal{A}, \tilde{T}, \tilde{r}, \tilde{s_0}, \gamma\rangle$, no information is lost in focusing our efforts in solving the belief-MDP instead of the initial POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, r, b_0, \gamma\rangle$ whose criterion is the left part of the equation.

The translation of a POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, b_0, \gamma\rangle$ into the so-called belief-MDP $\langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{T}, \tilde{r}, \tilde{s_0}, \gamma\rangle$ is summed up here:

- $\tilde{\mathcal{S}} = \mathbb{P}^{\mathcal{S}}_{b_0}$, the set of all reachable belief states from the initial one b_0;
- $\tilde{T}$ contains all transition probability distributions for belief states: $\forall a \in \mathcal{A}$, $\forall b \in \mathbb{P}^{\mathcal{S}}_{b_0}$, the belief transition probability distribution defined by equation (I.16), $\mathbf{p}(\cdot \mid b, a)$ is in $\tilde{T}$;
- the reward function $\tilde{r}$ is defined by the equation (I.13),
- the initial state $\tilde{s_0} = b_0$.

Note that the action set $\mathcal{A}$, as well as the discount factor γ remain the same. Note also that this belief-MDP fulfills the conditions defined in Section I.1.4. First, $\tilde{\mathcal{S}} = \mathbb{P}^{\mathcal{S}}_{b_0}$ is countable. Second, the successors of b_t for action $a \in \mathcal{A}$ form the set $\left\{ u(b_t, a, o') \in \mathbb{P}^{\mathcal{S}}_{b_0} \;\middle|\; o' \in \mathcal{O} \right\}$, which is finite as $\mathcal{O}$ is finite. For each $b \in \tilde{\mathcal{S}}$ and $a \in \mathcal{A}$, a finite number of belief states $b' \in \tilde{\mathcal{S}}$ is such that $\mathbf{p}(b' \mid b, a) > 0$, *i.e.* $\forall b \in \tilde{\mathcal{S}}$, $\forall a \in \mathcal{A}$, the support of the transition probability distribution $\mathbf{p}(\cdot \mid b, a)$ is finite: $\exists \tilde{\mathcal{S}}_{b,a} \subset \tilde{\mathcal{S}}$, such that $\#(\tilde{\mathcal{S}}_{b,a}) < +\infty$, and $\forall b' \in \tilde{\mathcal{S}} - \tilde{\mathcal{S}}_{b,a}$, $\mathbf{p}(b' \mid b, a) = 0$, making the transition function $\tilde{T}$ of the MDP $\langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{T}, \tilde{r}, \tilde{s_0}, \gamma\rangle$ satisfying the condition stated in Section I.1.4.

Bellman Equation I.5 stated in page 32 can be rewritten in the context of POMDPs: given a strategy $(d_t)_{t\geqslant 0}$, $\forall b \in \tilde{\mathcal{S}}$,

$$\begin{aligned} V^d(b) &= \left(\mathcal{B}^d V^{d^+}\right)(b) \\ &= \tilde{r}\Big(b, d_0(b)\Big) + \gamma \cdot \sum_{b' \in \tilde{\mathcal{S}}_{b,d_0(b)}} \mathbf{p}\Big(b' \Big| b, d_0(b)\Big) \cdot V^{d^+}(b') \\ &= \sum_{s\in\mathcal{S}} \tilde{r}\Big(s, d_0(b)\Big) \cdot b(s) + \gamma \cdot \sum_{o'\in O} \mathbf{p}\Big(o' \Big| b, d_0(b)\Big) \cdot V^{d+}\Big(u\Big(b, d_0(b), o'\Big)\Big) \end{aligned}$$

using transition formula (I.16), and where u is the update function defined in Theorem 6.

The Dynamic Programming operator is obtained adding $\max_{a\in\mathcal{A}}$ at the beginning, and replacing $d_0(b)$ by $a \in \mathcal{A}$ (see the Dynamic Programming equation I.7):

$$\mathcal{B}^* V(b) = \max_{a\in\mathcal{A}} \left\{ \sum_{s\in\mathcal{S}} r(s,a) \cdot b(s) + \gamma \cdot \sum_{o'\in\mathcal{O}} \mathbf{p}(o' \mid b, a) \cdot V\Big(u(b, a, o')\Big) \right\}. \tag{I.18}$$

The Dynamic Programming Equation $V^* = \mathcal{B}^* V^*$ characterizes

$$\sup_{d\in\mathcal{D}_\infty} V^d(b) = \sup_{d\in\mathcal{D}_\infty} \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t \cdot \tilde{r}\Big(B_t, d_t(B_t)\Big) \;\middle|\; B_0 = b \right].$$

Given the fact that some optimal MDP strategies are Markovian for additive criteria (*i.e.* only depends on the time step and the current system state), some optimal POMDP policies only depend on the current belief state, or equivalently on the full history of observations.

I.1.10 Solving a POMDP

Given an action $a \in \mathcal{A}$, the reward $\tilde{r}$ of a belief state $b \in \mathbb{P}^{\mathcal{S}}$ is a linear function of the belief state b: $\tilde{r}(b,a) = \sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) = \langle \tilde{r}_a, b\rangle_{\mathbb{R}^{\mathcal{S}}}$, where $\tilde{r}_a$ is the vector from $\mathbb{R}^{\mathcal{S}}$ such that for each index $s \in \mathcal{S}$, the value is $r(a,s) \in \mathbb{R}$. The function $(x,y) \mapsto \langle x,y\rangle_{\mathbb{R}^{\mathcal{S}}}$ is the classical scalar product of x and y in the vector space $\mathbb{R}^{\mathcal{S}}$.

For each belief state $b \in \mathbb{P}^{\mathcal{S}}$, the value $V^0(b) = \max_{a\in\mathcal{A}} \tilde{r}\,(b,a)$, is the optimal expected reward for an horizon 0 (only one decision step), starting with the belief state b. This function is PieceWise Linear and Convex (PWLC), *i.e.* there exists a finite set of vectors $\Gamma \subset \mathbb{R}^{\mathcal{S}}$ such that $V^0(b) = \max_{\alpha\in\Gamma}\langle \alpha, b\rangle_{\mathbb{R}^{\mathcal{S}}}$. As shown by R D. Smallwood and E J. Sondik [135] successive $V^i = \mathcal{B}^* V^{i-1}$ are PWLC, *i.e.* $V^i(b) = \max_{\alpha\in\Gamma}\langle \alpha, b\rangle_{\mathbb{R}^{\mathcal{S}}}$, where $\alpha \in \mathbb{R}^{\mathcal{S}}$ is called "α-vector":

Theorem 9

PWLC functions remain PWLC after the application of the operator $\mathcal{B}^*$. More specifically, if a function $V : \mathbb{P}^{\mathcal{S}} \to \mathbb{R}$ is PWLC, i.e. such that

$$V(b) = \max_{\alpha\in\Gamma}\langle \alpha, b\rangle_{\mathbb{R}^{\mathcal{S}}} = \max_{\alpha\in\Gamma}\sum_{s\in\mathcal{S}} \alpha(s)\cdot b(s) \text{ with } \Gamma \subset \mathbb{R}^{\mathcal{S}}, \#\Gamma < +\infty,$$

then, $\forall b \in \mathbb{P}^{\mathcal{S}}$,

$$\Big(B^*V\Big)(b) = \max_{\alpha'\in\Gamma'}\langle \alpha', b\rangle_{\mathbb{R}^{\mathcal{S}}} = \max_{a\in\mathcal{A}}\max_{(\alpha_o)\in\Gamma^{\mathcal{O}}}\langle \alpha_{a,(\alpha_o)}, b\rangle_{\mathbb{R}^{\mathcal{S}}}, \tag{I.19}$$

with the new α-vectors $\alpha_{a,(\alpha_o)} \in \Gamma'$ defined as

$$\alpha_{a,(\alpha_o)}(s) = r(s,a) + \gamma\cdot\sum_{o'\in\mathcal{O},s'\in\mathcal{S}} \boldsymbol{p}\left(o' \mid s',a\right)\cdot\boldsymbol{p}\left(s' \mid s,a\right)\cdot\alpha_{o'}(s), \tag{I.20}$$

where $(\alpha_o) \in \Gamma^{\mathcal{O}}$ is the notation for a vector such that the coordinates with index $o \in \mathcal{O}$ are α-vector $\alpha_o \in \Gamma$.

The proof is given in Annex A.14. Figures I.4 and I.5 represent examples of PWLC value functions

This result inspired many POMDP solvers. Indeed, it makes possible to compute the optimal value function and the associated strategy in finite horizon settings, and to approach them for an infinite horizon POMDP, through a few modifications of the Value Iteration algorithm, Algorithm 2: while the state space is infinite, the value function is summed up in a set of α-vectors. Let us start with a PWLC function $V^0(b) = \max_{\alpha\in\Gamma^0}\sum_{s\in\mathcal{S}}\alpha(s)\cdot b(s)$, where Γ^0 is the initial set of α-vectors. In order to perform the same computations as in the finite horizon case, it is possible to start with the *reward vectors* as initial α-vectors: $\Gamma^0 = \left\{\alpha \in \mathbb{R}^{\mathcal{S}} \;\middle|\; \forall s\in\mathcal{S}, \alpha(s) = r(s,a) \text{ with } a\in\mathcal{A}\right\}$. The function encoded by Γ^0 is in this case the optimal initial reward for an agent whose belief state is b: $V^0(b) = \max_{\alpha\in\Gamma^0}\sum_{s\in\mathcal{S}}\alpha(s)\cdot b(s) = \max_{a\in\mathcal{A}}\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) = \max_{a\in\mathcal{A}}\mathbb{E}_{S\sim b}\left[r(S,a)\right]$. Iterations of the operator $\mathcal{B}^*$ starting from V^0 create a sequence of functions $(V^i)_{i\in\mathbb{N}}$ whose theoretical limit is $\sup_{(d)\in\mathcal{D}_\infty} V^d$ as shown in Section I.1.4: here, each function of the sequence can be summed up with a finite set of alpha vectors, which makes computations possible in practice.

Given a number of iterations N for a specified error bound $\varepsilon > 0$ (see the error analysis in Section I.1.5), Algorithm 3 leads to a value function in the form of a set of α-vectors Γ_ε:

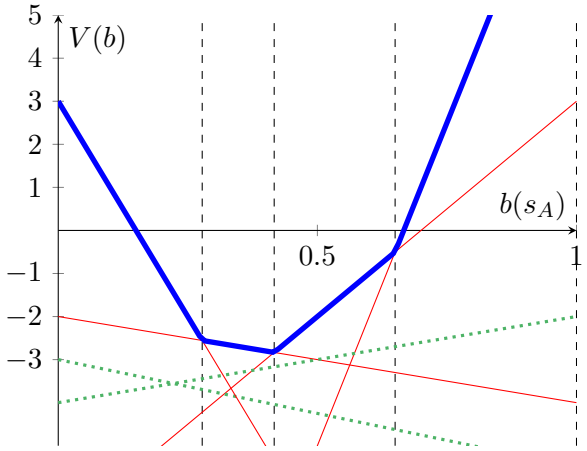


Figure I.4 – *Example of useful alpha vectors (red), value function PWLC (thick blue), and useless (bad) alpha vectors (dotted green) of a POMDP at a given iteration: the state space is* $\mathcal{S} = \{s_A, s_B\}$*: x-axis represents* $b(s_A)$ *and y-axis represents* $V(b)$*. As* $\#\mathcal{S} = 2$*,* $b(s_B) = 1 - b(s_A)$*.*

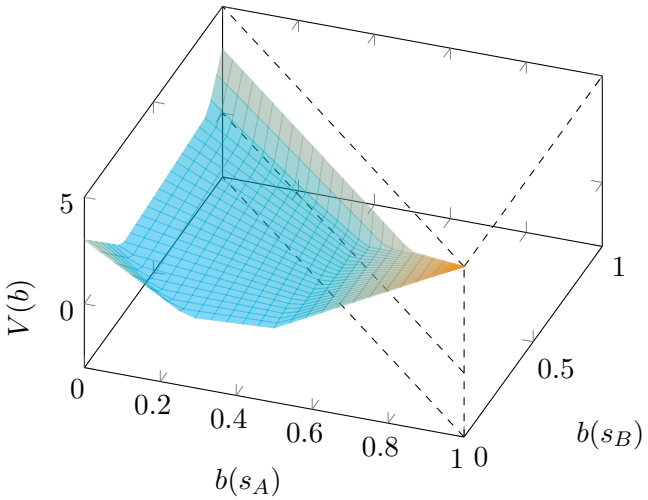


Figure I.5 – *Value function PWLC at a given iteration when the state space is* $\mathcal{S} = \{s_A, s_B, s_C\}$*: x-axis represents* $b(s_A)$*, y-axis represents* $b(s_B)$ *and z-axis represents* $V(b)$*. As* $\#\mathcal{S} = 3$*,* $b(s_C) = 1 - b(s_A) - b(s_B)$*.* $V(b)$ *is the maximum of a set of hyperplans.*

$V^{\varepsilon}(b) = \max_{\alpha\in\Gamma_\varepsilon} \sum_{s\in\mathcal{S}} \alpha(s)\cdot b(s)$. The associated strategy, approximately optimal, is

$$\begin{aligned} d^{\varepsilon}(b) &\in \operatorname*{argmax}_{a\in\mathcal{A}} \max_{(\alpha_o)\in(\Gamma_\varepsilon)^{\mathcal{O}}} \langle \alpha_{a,(\alpha_o)}, b\rangle_{\mathbb{R}^{\mathcal{S}}} \\ &\in \operatorname*{argmax}_{a\in\mathcal{A}} \max_{(\alpha_o)\in\Gamma_\varepsilon^{\mathcal{O}}} \sum_{s\in\mathcal{S}} r(s,a) + \gamma\cdot \sum_{o'\in\mathcal{O}}\sum_{s'\in\mathcal{S}} \mathbf{p}\left(o' \mid s', a\right)\cdot \mathbf{p}\left(s' \mid s, a\right)\cdot \alpha_{o'}(s'). \end{aligned}$$

At execution, if the agent has a belief state b, and the α-vector $\alpha_{a,(\alpha_o)}$ is such that $\forall \alpha\in\Gamma_\varepsilon$, $\langle \alpha_{a,(\alpha_o)}, b\rangle_{\mathbb{R}^{\mathcal{S}}} \geqslant \langle \alpha, b\rangle_{\mathbb{R}^{\mathcal{S}}}$, then action a is approximately optimal (with error ε).

Algorithm 3: Value Iteration Algorithm for POMDPs

```
Γ ← Γ^0;
i ← 1;
while i ⩽ N do
    for a ∈ A, (α_{o'}) ∈ Γ^O do
        for s ∈ S do
            α(s) ← r(s,a) + γ · Σ_{o'∈O} Σ_{s'∈S} p(o' | s', a) · p(s' | s, a) · α_{o'}(s');
        Γ' ← {Γ', α};
    Γ ← Γ';
    i++;
return Γ;
```

This algorithm is really naive since the number of α-vectors increases as a double exponential with iterations: if $\forall n\in\mathbb{N}$, Γ^n is the set of α-vectors at the end of iteration n, and $g_n = \#\Gamma^n$, then $g_{n+1} = \#\mathcal{A}\cdot g_n^{\#\mathcal{O}}$. Thus, as $\#\mathcal{A}$ is the initial number of α-vectors, $g_n = (\#\mathcal{A})^{\sum_{i=0}^{n-1}(\#\mathcal{O})^i}\cdot(\#\mathcal{A})^{(\#\mathcal{O})^n}$

A first improvement consists in removing, at each iteration $i = 1,\dots,N$, dominated α-vectors $\alpha_{bad}\in\Gamma^i$, *i.e.* α-vectors such that $\max_\alpha \sum_{s\in\mathcal{S}}\alpha(s)\cdot b(s) > \sum_{s\in\mathcal{S}}\alpha_{bad}(s)\cdot b(s)$, $\forall b\in\mathbb{P}^{\mathcal{S}}$: Cassandra's algorithms use linear programs to prune these dominated α-vectors [28, 27].

Whereas the resolution of finite state MDPs (MDPs with $\#\mathcal{S}<\infty$) is a P-complete problem [103], solving a finite horizon POMDP is PSPACE-hard [103], and solving an infinite horizon POMDP is undecidable [92]. These theoretical complexities are a faithful representation of the difficulty of solving POMDPs in practice. Algorithm 3 or Cassandra's improvements [28, 27] solve only really small POMDPs, *i.e.* POMDPs with a few system states and observations. For instance in the case of robotic mission problems, the number of system states may be quite big, as well as the number of observations, and classical computations are intractable: other computation methods are necessary to compute efficiently a satisfactory strategy. The next section is devoted to the presentation of the most notorious algorithms producing approximate stategies within reasonable time.

I.1.11 Upper and lower bounds on a value function

This section is meant to sum up the main good recipes to approximate POMDP optimal strategies, as strategy computation is a difficult task in practice and theoretically intractable [103, 92].

First consider $\widehat{V}$ and $\tilde{V}$ two functions mapping the set of all probability distributions $\mathbb{P}^{\mathcal{S}}$ to $\mathbb{R}$. Suppose that $\forall b\in\mathbb{P}^{\mathcal{S}}$, $\widehat{V}(b)\leqslant\tilde{V}(b)$. Then, $\forall b\in\mathbb{P}^{\mathcal{S}}$,

$$\left(\mathcal{B}^*\widehat{V}\right)(b) = \max_{a\in\mathcal{A}}\left\{\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \gamma\cdot\sum_{o'\in\mathcal{O}}\mathbf{p}\left(o'\mid b,a\right)\cdot\widehat{V}\left(u(b,a,o')\right)\right\} \leqslant \left(\mathcal{B}^*\tilde{V}\right)(b).$$

Thus, if $\forall b \in \mathbb{P}^{\mathcal{S}}$, $V(b) \leqslant V^*(b)$, *i.e.* if V is a lower bound of the optimal value function V^*, $\forall b \in \mathbb{P}^{\mathcal{S}}$, $\left(\mathcal{B}^*V\right)(b) \leqslant \left(\mathcal{B}^*V^*\right)(b) = V^*(b)$. As well, if V is an upper bound of the optimal value function, $\forall b \in \mathbb{P}^{\mathcal{S}}$, $\left(\mathcal{B}^*V\right)(b) \geqslant \left(\mathcal{B}^*V^*\right)(b) = V^*(b)$. This means that the iterations of the Dynamic Programming operator $\mathcal{B}^*$ on a lower (resp. upper) bound of V^* return a lower (resp. upper) bound of V^*.

Defining $r_{min} = \min_{s\in\mathcal{S},a\in\mathcal{A}} r(s,a)$, the constant function $\underline{V_0}(b) = \sum_{t\geqslant 0} \gamma^t \cdot r_{min} = \frac{r_{min}}{1-\gamma}$, $\forall b \in \mathbb{P}^{\mathcal{S}}$, is an example of initial lower bound of the optimal value function: the worst reward is obtained at each time step. The only α-vector representing this function is $\alpha_0(s) = \frac{r_{min}}{1-\gamma}$, $\forall s \in \mathcal{S}$.

Let us start from an initial set of α-vectors denoted by $\underline{\Gamma_0}$, defining a lower bound of the optimal value function, *i.e.* such that $\underline{V_0}(b) = \max_{\alpha\in\underline{\Gamma_0}} \sum_{s\in\mathcal{S}} \alpha(s)\cdot b(s) \leqslant V^*(b)$: for instance $\underline{\Gamma_0} = \{\alpha_0\}$. The α-vectors $\alpha \in \underline{\Gamma_1}$ computed using the current α-vector set Γ_0 and the equation (I.20) of Theorem 9, take part in the definition of $\underline{V_1}(b) = \left(\mathcal{B}^*\underline{V_0}\right)(b)$. As noted above, $\underline{V_1}$ is also a lower bound: $\underline{V_1}(b) = \max_{\alpha\in\underline{\Gamma_1}} \sum_{s\in\mathcal{S}} \alpha(s)\cdot b(s) = \left(\mathcal{B}^*\underline{V_0}\right)(b) \leqslant V^*(b)$, $\forall b \in \mathbb{P}^{\mathcal{S}}$. Thus, as $\underline{V_0}$ and $\underline{V_1}$ are lower bounds of V^*, $\max\left\{\underline{V_0}, \underline{V_1}\right\}$ too, *i.e.* $\max_{\alpha\in\underline{\Gamma_0}\cup\underline{\Gamma_1}} \sum_{s\in\mathcal{S}} \alpha(s)\cdot b(s)$ is a lower bound of $V^*(b)$, and the best available at the moment. Hence, it is sufficient to maintain a set of α-vectors $\underline{\Gamma}$: the α-vectors computed from $\underline{\Gamma}$ may be added to $\underline{\Gamma}$, and the dominated ones may be removed. Because of the convergence of $\left((\mathcal{B}^*)^n\underline{V_0}\right)_{n\in\mathbb{N}}$ towards V^*, the computation of new α-vectors tends to improve the lower bound.

The presented machinery of incremental improvement of the lower bound of V^* using new α-vectors cannot be adapted to compute an upper bound: starting from an upper bound $\overline{V_0}$, if the computed α-vectors represent a function $\overline{V_1}$ (also an upper bound of V^*) which tends to be closer to V^* than $\overline{V_0}$ (because of the convergence), then $\max\left\{\overline{V_0}, \overline{V_1}\right\}$ is not a better upper bound: it is actually the worst one. However, it is convenient to consider the maximum over all computed alpha vectors in practice, that is why $\min\left\{\overline{V_0}, \overline{V_1}\right\}$ cannot be considered.

Another method must be used to maintain and improve an upper bound of the value function: an upper bound $\overline{V}_0$ of the optimal value function is only computed over a set of $n > 0$ belief states, leading to the belief-value mappings $\{b_i, \overline{v}_i\}_{i=1}^n$, where $\overline{V}_0(b_i) = \overline{v}_i$ and $(b_i, \overline{v}_i) \in \mathbb{P}^{\mathcal{S}} \times \mathbb{R}$. As the limit of a sequence of convex functions is convex, the optimal value function V^* is known to be convex. As the mappings are such that $\overline{v}_i \geqslant V^*(b_i)$, and as V^* is convex, any convex combination of the belief states $(b_i)_{i=1}^n$ has an optimal value lower or equal to the same convex combination of the upper bound values $(\overline{v_i})_{i=1}^n$. Indeed, if $(w_i)_{i=1}^n$ are convex coefficients, *i.e.* $\sum_{i=1}^n w_i = 1$ and $\forall i \in \{1,\ldots,n\}$, $w_i \geqslant 0$, then the optimal value function at $\sum_{i=1}^n w_i \cdot b_i$ is bounded by the convex combination of $(\overline{v_i})_{i=1}^n$ with respect to $(w_i)_{i=1}^n$: $V^*(\sum_{i=1}^n w_i \cdot b_i) \leqslant \sum_{i=1}^n w_i \cdot V^*(b_i) \leqslant \sum_{i=1}^n w_i \cdot \overline{v_i}$. In order to get an upper bound of V^* defined on a larger set of belief states, the belief-value mappings $\{b_i, \overline{v_i}\}_{i=1}^n$ may be completed with the pairs (b, v) such that $b \in \mathbb{P}^{\mathcal{S}}$ is in the convex hull of $(b_i)_{i=1}^n$, and v is the least value of $\sum_{i=1}^n w_i \cdot \overline{v_i}$, where $(w_i)_{i=1}^n$ are convex coefficients such that $b = \sum_{i=1}^n w_i \cdot b_i$. These coefficients defining the interpolation of the belief-value mappings can be computed using linear programming or with approximate computations [73, 113]. Finally, consider $b_j \in (b_i)_{i=1}^n$ such that $\forall a \in \mathcal{A}$, $\forall o' \in \mathcal{O}$, $u(b_j, a, o')$ is in $(b_i)_{i=1}^n$. The upper bound value $\overline{v_j}$ associated to b_j can be replaced with $(\mathcal{B}^*\overline{V}_0)(b_j)$, computed using the equation (I.18) defining the Dynamic Programming operator $\mathcal{B}^*$:

$$(\mathcal{B}^*\overline{V}_0)(b_j) = \max_{a\in\mathcal{A}} \Big\{ \sum_{s\in\mathcal{S}} r(s,a)\cdot b_j(s) + \gamma \cdot \sum_{o'\in\mathcal{O}} \mathbf{p}\left(o' \mid b_j, a\right) \overline{V}_0\left(u(b_j,a,o')\right) \Big\},$$

where $\overline{V}_0\left(u(b_j,a,o')\right) = \overline{v_k}$ with $u(b_j,a,o') = b_k \in (b_i)_{i=1}^n$. This value update replaces then the

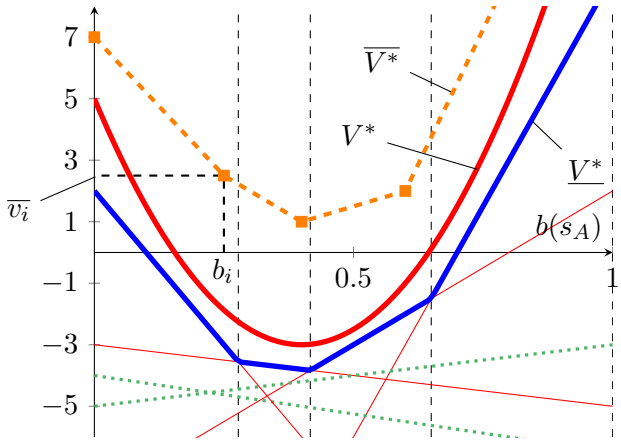


Figure I.6 – *Bounds of the optimal value function V^*. The latter is represented by the regular thick red line. The lower bound $\underline{V^*}$ is the piecewise linear function represented by the thick blue line. Useful α-vectors are represented by the thin red lines, and dominated ones are represented by dotted green lines. The upper bound $\overline{V^*}$ is represented by the piecewise linear dashed orange line: the squares represent the belief-value mappings.*

value $\overline{v_j}$ and leads to an improved upper bound of the optimal value function in b_j. A famous and simple method to compute an initial upper bound of V^* is called the Q_{MDP} method [90]: it consists in computing an optimal value function for the underlying MDP *i.e.* for the MDP built ignoring the observations and the observation probabilities, and considering that the state is fully observable. An MDP strategy is searched among a more general set of functions of the data available at execution, than a POMDP strategy. First, the reward is defined on the system states and the actions, which are directly available during execution in fully observable settings. Second, as the uncertainty over the observations is conditional on the system state, the observation random variables O_t can be written as measurable functions of the state and action variables S_t and A_{t-1}: the functions from the actions and the observations consist thus of a subset of the functions from the system states and the actions. We can conclude that, the optimal value function of the POMDP starting from the deterministic belief state $b_0^A(s) = \mathbb{1}_{\{s=s_A\}}$ with $s_A \in \mathcal{S}$, namely

$$V^*(b_0^A) = \max_{\substack{(d_t)_{t\geqslant 0} \text{ s.t.} \\ d_t:\mathcal{I}_t\to\mathcal{A}}} \mathbb{E}_{S_0\sim b_0^A}\left[\sum_{t\geqslant 0}\gamma^t\cdot r\Big(S_t, d(I_t)\Big)\right],$$

is less than or equal to the optimal value function for the associated MDP starting from s_A:

$$V^*_{MDP}(s_A) = \max_{\substack{(d_t)_{t\geqslant 0} \text{ s.t.} \\ d_t:\mathcal{S}\to\mathcal{A}}} \mathbb{E}\left[\sum_{t\geqslant 0}\gamma^t\cdot r\Big(S_t, d(S_t)\Big)\,\middle|\, S_0 = s_A\right].$$

Indeed, the maximum operator of the latter is performed on a set of functions including the one used to maximize the POMDP value function at b_0^A (first equation). Hence, an initial upper bound of the POMDP optimal value function V^* can be computed for the deterministic belief states *i.e.* the belief state $b \in \mathbb{P}^{\mathcal{S}}$ such that $b(s) = \mathbb{1}_{\{s=s_A\}}$ with $s_A \in \mathcal{S}$: $\overline{V_0}(b) \geqslant V^*_{MDP}(s_A)$. The convex hull of these deterministic belief states is the full set of probability distributions $\mathbb{P}^{\mathcal{S}}$. Thus, using the previous interpolation trick, an upper bound of V^* is available in the full belief space $\mathbb{P}^{\mathcal{S}}$. Other methods to compute bounds for V^* are available for instance in [72]. Figure I.6 illustrates the described bounds.

I.1.12 POMDP solvers

Some recent POMDP solvers are said to be *point-based* as they maintain a set of belief states $(b_i)_{i=1}^n$ (the *belief points*) with associated α-vectors $(\alpha_i)_{i=1}^n$. The current approximation V of the optimal value function is described by $(\alpha_i)_{i=1}^n$: $\forall b \in \mathbb{P}^{\mathcal{S}}$, $V(b) = \max_{i=1}^n \sum_{s\in\mathcal{S}} \alpha_i(s) \cdot b(s)$. Each belief state of $(b_i)_{i=1}^n$ is such that $V(b_i) = \sum_{s\in\mathcal{S}} b_i(s) \cdot \alpha_i(s)$. The point-based Bellman backup of a belief state b_i is the replacement of its α-vector α_i by one of the new α-vectors (Equation I.20) which maximize equation (I.19) with the belief state b_i, *i.e.* replacing it with $\alpha' \in \operatorname{argmax}_{\alpha'\in\Gamma'} \langle \alpha', b_i \rangle_{\mathbb{R}^{\mathcal{S}}}$.

The first point-based algorithm was PBVI (*Point-Based Value Iteration*) [108] which starts with a single belief point b_0. At a given iteration, the point-based Bellman backup of each belief state of the current set $(b_i)_{i=1}^n$ is computed and lead to a new set of α-vectors $(\upsilon'_i)_{i=1}^n$. This operation is repeated until the convergence of the α-vectors. Next, for each belief state in $(b_i)_{i=1}^n$, one successor is selected: the most distant one from $(b_i)_{i=1}^n$, with respect to a given distance metric. These successors are added to the set, which becomes $(b_i)_{i=1}^{2n}$, and the next iteration begins.

Another point-based POMDP solver is the *Perseus* solver [137]: this solver starts running trials of random exploration of the belief space, sampling $a \in \mathcal{A}$ and observation $o' \in \mathcal{O}$ at each time step of a trial to compute the next belief state $b' = u(b, a, o')$ from the current one $b \in \mathbb{P}^{\mathcal{S}}$. The set of all reached belief states is $(b_i)_{i=1}^n$ and does not evolve anymore. A lower bound PWLC $\underline{V}$ is used to approach the value function: the associated set of α-vectors is denoted by $\underline{\Gamma}$. At each iteration, $\mathbb{B}$ is initialized as a copy of $(b_i)_{i=1}^n$, and $\Gamma = \emptyset$. While $\mathbb{B} \neq \emptyset$, an arbirary belief state b is selected in $\mathbb{B}$, and the associated new α-vector α' is computed with the point-based Bellman backup on b and using $\underline{\Gamma}$. If $\langle \alpha', b \rangle_{\mathbb{R}^{\mathcal{S}}} \geqslant \underline{V}(b)$, all the belief states whose value is improved by α', *i.e.* the belief states $\tilde{b} \in \mathbb{B}$ such that $\langle \alpha', \tilde{b} \rangle_{\mathbb{R}^{\mathcal{S}}} \geqslant \underline{V}(\tilde{b})$, are removed from $\mathbb{B}$. The new α' is added to Γ. Otherwise, if $\langle \alpha', b \rangle_{\mathbb{R}^{\mathcal{S}}} < \underline{V}(b)$, b is removed from $\mathbb{B}$ and an α-vector from $\underline{\Gamma}$ such that $\tilde{\alpha} \in \operatorname{argmax}_{\alpha\in\underline{\Gamma}} \langle \alpha, b \rangle_{\mathbb{R}^{\mathcal{S}}}$ is added to Γ. When $\mathbb{B} = \emptyset$, $\underline{\Gamma}$ is set to Γ, and a new iteration begins.

The HSVI solver (*Heuristic Search Value Iteration*) [136] is also a point-based algorithm. This solver maintains both an upper and a lower bound on V^*: $\overline{V}$ and $\underline{V}$. It takes into account the fact that the error of the approximation of V^* is less important for much later successors of $b_0 \in \mathbb{P}^{\mathcal{S}}$, due to the discount factor γ: given an error $\varepsilon > 0$, a sequence of belief states $(b_t)_{t\geqslant 0}$ is generated from b_0 until $\overline{V}(b_t) - \underline{V}(b_t) < \frac{\varepsilon}{\gamma^t}$. The generation of the belief sequence is done selecting actions according to the upper bound: if the current belief state is b, the chosen action is in $\operatorname{argmax}_{a\in\mathcal{A}} r(s,a) + \gamma \sum_{o'\in\mathcal{O}} \overline{V}\Big(u(b,a,o')\Big)$. This trick tends to force the improvement of the upper bound: if the selected action is not optimal, as successors for this action are selected, the computations will focus on these belief states, and will decrease (improve) the upper bound for them. As well, the observation selected $o' \in \mathcal{O}$ is such that the value $\overline{V}\Big(u(b,a,o')\Big) - \underline{V}\Big(u(b,a,o')\Big) \cdot \mathbf{p}\,(o' \mid b, a)$ is the greatest: the probable belief states for which $\overline{V}$ and $\underline{V}$ are poor bounds are preferred, in order to focus the computational efforts on the belief space subsets where the bounds are the worst. Then, both bounds are updated starting with the last reached belief state backward down to time step t_0: this order makes these updates more efficient. The scheme starting with a belief sequence generation, and updating the bounds on it, is repeated until $\overline{V}(b_0) - \underline{V}(b_0) < \varepsilon$.

The solver SARSOP [84], inspired by HSVI and FSVI [130], refines the generation method of the belief sequence. First of all, the belief space is clustered using a simple learning technique: the features are for instance the initial upper bound $\underline{V_0}$ and the entropy of belief states. This discretization is used to maintain an estimation of the optimal value function denoted by $\widehat{V}$: this estimation is constant over each cluster, equal to the average of the estimated

optimal value of the belief states in this cluster. Let b be a belief state reached during a generation of a belief sequence: L_1 is a real number such that $L_1 \leqslant V^*(b)$. Let $L_2 = \Big(\mathcal{B}^*\underline{V}\Big)(b)$. Thus, the lower bound of the optimal value function $\underline{V}$ on b is likely to be improved by selecting the next belief state $b' = u(b,a,o')$ (where $a \in \mathcal{A}$ and $o' \in \mathcal{O}$ are selected as in HSVI) if the optimal value for b' is likely to be large enough for it: that is, if $r(b,a) + \gamma \cdot \Big(\mathbf{p}\,(\,o' \mid\, b,a\,) \cdot \widehat{V}(b') + \sum_{\tilde{o}\neq o'} \mathbf{p}\,(\,\tilde{o} \mid\, b,a\,) \cdot \underline{V}\Big(u(b,a,\tilde{o})\Big)\Big)$ is greater than $L = \max\{\,L_1, L_2\,\}$. In this case, if L_1' is such that $L = r(b,a) + \gamma \cdot \Big(\mathbf{p}\,(\,o' \mid\, b,a\,) \cdot L_1' + \sum_{\tilde{o}\neq o'} \mathbf{p}\,(\,\tilde{o} \mid\, b,a\,) \cdot \underline{V}\Big(u(b,a,\tilde{o})\Big)\Big)$, the condition $\widehat{V}\Big(u(b,a,\tilde{o})\Big) \geqslant L_1'$ is a good indicator that the selection of the belief state b' is likely to improve the current lower bound at b: if $\widehat{V}(b') \geqslant L_1'$ (and if, as in HSVI, $\overline{V}(b') - \underline{V}(b') \leqslant \frac{\varepsilon}{\gamma^t}$), the belief state b' is selected and the same test is performed on its successor knowing that $V^*(b')$ is likely to be greater than L_1'. In this way, the selected belief sequences may be longer than in HSVI, as sequence generation continues if the next belief states are likely to improve optimal value function estimation. Next, when the generation of the sequence ends, the bound updates start from the last belief state down to b_0, following the generated belief state sequence in the reverse order (as with HSVI). Finally, SARSOP proposes also another major computational simplification: all the previous belief sequence generations can be summed up in a tree representing the different transitions $b \to u(b,a,o')$ and which is recorded in the memory. Let us define $\underline{Q}(b,a) = r(b,a) + \gamma \cdot \sum_{o'\in\mathcal{O}} \mathbf{p}\,(\,o' \mid\, b,a\,) \cdot \underline{V}\Big(u\,(\,b,a,o'\,)\,\Big)$, and $\overline{Q}(b,a)$ with the same formula replacing $\underline{V}$ by $\overline{V}$. If b is a belief state, a an action, and $\exists a' \in \mathcal{A}$, $\exists b' \in \mathbb{P}^{\mathcal{S}}_{b_0}$ such that $\overline{Q}(b,a) \leqslant \underline{Q}(b',a')$, then, all the successors of b when selecting action a are removed from the tree, and the associated α-vectors deleted. Indeed, less stored α-vectors and belief states speed up the POMDP resolution, as lots of useless computations are avoided: the computations are focused on the belief states which seem to be in $\mathbb{P}^{\mathcal{S}}_{b_0,*}$. The set $\mathbb{P}^{\mathcal{S}}_{b_0,*}$ is the subset of $\mathbb{P}^{\mathcal{S}}_{b_0}$ containing the belief states reached with an optimal strategy.

While the α-vector (or Sondic's) representation is used by a large proportion of POMDP solvers, *grid-based* POMDP solvers, which does not use it, are also popular algorithms [12, 91, 25, 13]. These solvers are based on a discretization of the belief space $\mathbb{P}^{\mathcal{S}}$, which leads to an MDP over the finite set of discretized belief states: computed strategies map any cluster of belief states to an action $a \in \mathcal{A}$.

Another POMDP solver based on a discretization of the belief space is RTDP-bel [70]. The discretization is only used to store a finite number of values during the computations. The approximation maintains the approximate optimal value function as a piecewise constant function: two belief states in the same discretization group have the same value. The algorithm operates in the same way as RTDP (*Real Time Dynamic Programming*) [7], a *Goal-MDP* solver which converges to the optimal value function without considering all the system states. A Goal-MDP [11] is an MDP all reward values of which are negative; it includes also a subset of the system state $\mathcal{G} \subset \mathcal{S}$ called *set of goals*. The system states in $\mathcal{G}$ are absorbing and cost-free: $\forall (s,a) \in \mathcal{G} \times \mathcal{A}$, $r(s,a) = 0$ and $\mathbf{p}\,(\,s \mid\, s,a\,) = 1$. The criterion of a Goal-MDP is the expected (undiscounted) total reward, *i.e.* the expectation of the sum of the rewards over time steps without factors γ^t: indeed, the sum of costs is guaranteed to converge provided that there exist proper strategies (*i.e.* reaching goals with probability 1) and no positive-cost cycles in the MDP graph. As well, a *Goal-POMDP* is a POMDP with only negative rewards and a set of goals $\mathcal{G}$ which are absorbing, cost-free and fully observable system states *i.e.* $\mathcal{O}$ contains $\mathcal{G}$ too, and $\forall s' \in \mathcal{G}$, $\forall a \in \mathcal{A}$, $\forall t \geqslant 0$, $\mathbb{P}\,(\,O_{t+1} = s' \mid\, S_{t+1} = s', a\,) = 1$. In fact RTDP-bel is a Goal-POMDP solver. It initializes the value function V with a (piecewise constant) upper bound called *admissible heuristic*. Then, the repetition of trials starting from

b_0 improves the approximation of the optimal value function and the associated strategy. At a given stage of a trial, the current belief state is denoted by b. The Q-value is $Q(b,a) = \Big\{ r(b,a) + \sum_{o' \in \mathcal{O}} \mathbf{p}\,(o' \mid b,a) \cdot V\Big(u(b,a,o')\Big) \Big\}$. An action $a \in \operatorname{argmax}_{\tilde{a} \in \mathcal{A}} Q(b,\tilde{a})$ is selected, and $V(b)$ is updated to $\max_{a \in \mathcal{A}} Q(b,a)$. Then s is sampled from b, as well as s' according to $\mathbf{p}\,(s' \mid s,a)$, and o' using $\mathbf{p}\,(o' \mid s',a)$. If $b' = u(b,a,o')$ is such that $\forall s \in \mathcal{S} \setminus \mathcal{G}$, $b'(s) = 0$, then another trial begins. Otherwise, the next stage of the trial considers b'. As any classical (discounted) POMDP (Section I.1.6) can be translated into a Goal-POMDP [15], RTDP-bel can solve any POMDP.

Finally, POMCP [132] is the partially observable counterpart of the MDP solver UCT [80]. The latter is based on the UCB (Upper Confidence Bound) strategy for stochastic bandits [5] and is an instance of MCTS (Monte-Carlo Tree Search) [34]. A decision tree whose nodes are the reached states and arrows are the actions, is built during simulations to maintain at each node the counts $N(s,a)$ of the visits to the couple (s,a), for each action $a \in \mathcal{A}$ and for each reachable system state. The estimation V of the optimal value function is computed by Monte-Carlo simulations. The Q-value is $Q(s,a) = \{\, r(s,a) + \gamma \cdot \sum_{s' \in \mathcal{S}} \mathbf{p}\,(s' \mid s,a) \cdot V(s') \,\}$. The UCB-inspired exploration-exploitation strategy is to select $a \in \operatorname{argmax}_{a \in \mathcal{A}} \left\{ Q(s,a) + c \cdot \sqrt{\frac{logN(s)}{N(s,a)}} \right\}$, where $N(s) = \sum_{a \in \mathcal{A}} N(s,a)$ and $c > 0$ is the relative ratio between exploration to exploitation: the more c is small, the more actions with high values are selected (exploitation), the more $c > 0$ is large, the more the actions are selected with about the same rate, without paying attention to the estimated values. The term $c \cdot \sqrt{\frac{logN(s)}{N(s,a)}}$ is meant to force the actions rarely tried before to be selected (exploration). In the POMCP case, during computations, the belief state is approximated by an unweighted particule filter $B_t(s) \approx \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}_{S_i = s}$ where $\forall i = 1, \ldots, K$, $S_i \sim B_t$, and the nodes of the tree represent possible successive pieces of information $i_t = \{\, a_0, o_1, \ldots, a_{t-1}, o_t \,\}$, instead of the system states like in UCT.

The algorithms presented in this section are part of the state of the art POMDP solvers, and some of them will be used in the next chapters in order to conduct comparisons with our work. Now, the second main subject of this thesis is presented: Possibility Theory, and the qualitative possibilistic counterpart of the Partially Observable Markov Decision Processes.

I.2 Qualitative Possibilistic MDPs

This section presents the uncertainty framework studied in this thesis: namely, the model π-POMDP. First of all, the Possibility Theory is presented, with a particular emphasis on the qualitative part of the theory. Qualitative conditioning is then presented, as well as notions of independence. Finally, the qualitative possibilistic counterpart of the MDPs called *Qualitative Possibilistic MDPs*, or π-MDPs are defined, followed by the *Qualitative Possibilistic POMDPs*, or π-POMDPs.

I.2.1 Possibility Theory

The "fuzzy sets" introduced by Lotfi Zadeh [153], have been studied by Didier Dubois [39] and Henri Prade and their contributions have led to the foundation of Possibility Theory [57].

As in Probability Theory, this theory is based on the definition of an uncertainty measure, called *possibility measure*. Unlike the probability measure $\mathbb{P}$ which is a classical measure (Definition I.2.1), the *possibility measure*, denoted by Π, is a *fuzzy measure*, or *capacity*. For simplicity, a fuzzy measure is not supposed to be additive, but just *monotone*, as highlighted by Definition I.2.2. In this thesis, Possibility Theory will only concern finite sets such as $\mathcal{S}$ and $\mathcal{O}$, that is why definitions only concern finite sets.

Definition I.2.1 (*Measure*)

A classical measure $\mathbb{M}$ *on the finite set* Ω *is a function from* 2^Ω *to* $\mathbb{R}^+$ *such that*

- $\mathbb{M}(\emptyset) = 0$ *(null empty set);*
- $\forall A, B \subseteq \Omega$ *such that* $A \cap B = \emptyset$, $\mathbb{M}(A \cup B) = \mathbb{M}(A) + \mathbb{M}(B)$ *(additivity).*

Definition I.2.2 (*Fuzzy Measure*)

A fuzzy measure $\mathfrak{M}$ *on the finite set* Ω *is a function from* 2^Ω *to* $\mathbb{R}^+$ *such that*

- $\mathfrak{M}(\emptyset) = 0$ *(null empty set);*
- $\forall A, B \subseteq \Omega$ *such that* $A \subseteq B$, $\mathfrak{M}(A) \leqslant \mathfrak{M}(B)$ *(monotonicity).*

Note that a classical measure is a particular case of fuzzy measure since classical measures are monotone: if $A \subset B$,

$$\mathbb{M}(B) = \mathbb{M}\Big((A \cap B) \cup (\overline{A} \cap B)\Big) = \mathbb{M}(A) + \mathbb{M}(\overline{A} \cap B) \geqslant \mathbb{M}(A),$$

where $\overline{A}$ is the complementary set of A in Ω *i.e.* $\overline{A} \cap A = \emptyset$, and $\overline{A} \cup A = \Omega$.

A possibility measure is also a particular case of fuzzy measure:

Definition I.2.3 (*Possibility Measure*)

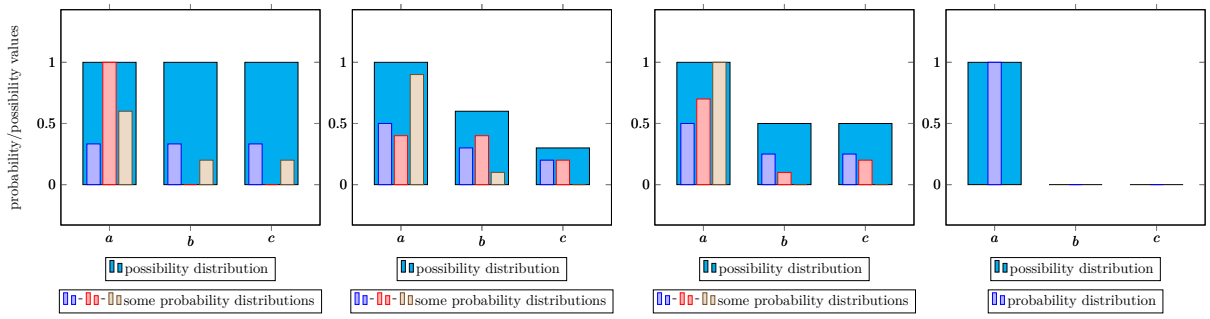
A possibility measure on the finite set Ω *is a fuzzy measure such that*

- $\Pi(\Omega) = 1$ *(normalization);*
- $\forall A, B \subset \Omega$, $\Pi\{A \cup B\} = \max\{\Pi(A), \Pi(B)\}$ *(maxitivity).*

Probability Theory models the uncertainty due to the variability of events: in practice, used probabilities are estimated frequencies of events stated as the actual variability model of events. Another view of this theory is De Finetti's one [41]: the probability value of an event is an exchangeable bet, *i.e.* the value in $[0, 1]$ that a given person is willing to give for the bet winning 1 if the event is true. However this person takes into account in her/his choice of value that the bet can be reversed just before verifying if the event is true. Indeed, she/he may be asked to get the chosen value, and to give 1 if the event is true. As the probability values depend on a natural person, who wants to guess the actual probability distribution as well as possible with respect to information she/he knows about the event, they are called *subjective probabilities*: that is why the theory based on this definition is called Subjective Probability Theory.

Possibility Theory is devoted to uncertainty due to a lack of knowledge or imprecision about an event. Quantitative Possibility Theory can be seen as a special case of imprecise probabilities *i.e.* a possibility measure Π represents a set of probability measures defined on Ω, denoted by $\mathcal{P}^\Pi$, and each probability measure $\mathbb{P} \in \mathcal{P}^\Pi$ is a guess about the actual probabilistic model. The set $\mathcal{P}^\Pi$ is the set of each probability measure $\mathbb{P}$ such that $\forall A \subset \Omega$, $\mathbb{P}(A) \leqslant \Pi(A)$. In this case, possibility measures are thus "inflated" probability measures, in order to model that frequencies are not well known, as illustrated iin Figure I.7. The possibility measure is then $\forall A \subset \Omega$, $\Pi(A) = \max_{\mathbb{P} \in \mathcal{P}^\Pi} \mathbb{P}(A)$.

If the set of probability distributions consists of all the probability distributions defined on Ω, *i.e.* the probabilistic model is completely unknown, and then $\forall A \subset \Omega$, $\Pi(A) = \max_{\mathbb{P} \in \mathcal{P}^\Pi} \mathbb{P}(A) = 1$: the ignorant possibility measure is equal to 1 for each set $A \subset \Omega$, as



(a) *ignorant distribution* (b) *three different degrees* (c) *two different degrees* (d) *determinism*
Figure I.7 – *Example of quantitative possibility distributions (thick blue), and some of the associated probability distributions (thin) i.e. some of the probability measures encoded by the possibility distribution. Indeed, a quantitative possibility distribution is a upper bound on possible probability distributions, defining thus a particular crecal set. Distributions are defined on* $\Omega = \{a, b, c\}$.

illustrated in Figure I.7a. On the contrary, if the actual elementary event is known to be ω_A, $\forall \omega \in \Omega$ such that $\omega \neq \omega_A$, $\Pi(\{\omega\}) = \mathbb{P}(\{\omega\}) = 0$ and $\Pi(\{\omega_A\}) = \mathbb{P}(\{\omega_A\}) = 1$, as illustrated in Figure I.7d. It is worth noting that there exist some sets of probability distributions which are not represented by any quantitative possibility distribution: for instance, the set of probability distributions on $\Omega = \{\omega_A, \omega_B\}$, $\{\mathbb{P} \mid \mathbb{P}(\omega_A) \geqslant 0.1, \mathbb{P}(\omega_B) \geqslant 0.1\}$.

Unlike quantitative ones, Qualitative Possibility Theory uses possibility measures whose values are defined in any ordered scale. This theory allows us to reason under a lack of quantitative information: the only information given by a qualitative possibility measure is the rank between events *i.e.* $\forall A, B \subseteq \Omega$, the information "event A is less plausible than event B", which is written $\Pi(A) \leqslant \Pi(B)$. Hence qualitative possibility measures Π are often defined as functions $2^\Omega \to \mathcal{L}$, where $\mathcal{L}$ is a finite set called *possibility scale* and equipped with a total order. In this work, and more specifically for the following three chapters of this thesis, the possibility scale is defined as $\mathcal{L} = \left\{0, \frac{1}{k}, \ldots, 1\right\}$ to simplify notations.

The structure of Possibility Theory is easily understood using the terminology of *fuzzy sets*. A classical set A of elements of Ω can be defined through a characteristic (or membership) function

$$\mathbb{1}_A : \begin{cases} \Omega & \to & \{0,1\} \\ \omega & \mapsto & \begin{cases} 1 & \text{if } \omega \in A \\ 0 & \text{otherwise.} \end{cases} \end{cases}$$

In practice, some problems may need to be more flexible about the membership of elements using *membership degrees*: a *fuzzy set* $\mathfrak{A}$ is defined by a characteristic function $\mathbb{1}_{\mathfrak{A}} : \Omega \to \mathcal{L}$ whose range of values is not only $\{0,1\}$ but may be in a totally ordered set $\mathcal{L}$: $\mathbb{1}_{\mathfrak{A}}(\omega) \in \mathcal{L}$ is the *membership degree* of $\omega \in \Omega$. If $\mathbb{1}_{\mathfrak{A}}(\omega) = 0$, then $\omega \notin \mathfrak{A}$. If $\mathbb{1}_{\mathfrak{A}}(\omega) = 1$, then $\omega \in \mathfrak{A}$. And, finally, if $\mathbb{1}_{\mathfrak{A}}(\omega) = \lambda \in \mathcal{L} \setminus \{0,1\}$, then $\omega \in \mathfrak{A}$ with membership degree λ.

Let us define the *possibility distribution* as $\pi(\omega) = \Pi(\{\omega\})$: according to Definition I.2.3, the possibility measure is entirely defined by the distribution π. Consider that the set Ω is the set of states $\mathcal{S}$. Let S be a variable representing a state of the problem, and whose values are in $\mathcal{S}$. Consider an expert description of the actual value of S, given for instance in natural langage: "the state is near state $s_A \in \mathcal{S}$ (in some sense) and is not s_B". This description given by the expert knowledge leads to a fuzzy set $\mathfrak{T}$: $\mathbb{1}_{\mathfrak{T}}(s)$ is the degree of membership of $s \in \mathcal{S}$, *i.e.* the degree of how well s respects the description. For instance, in the previous example, $\mathbb{1}_{\mathfrak{T}}(s_B) = 0$, and $\mathbb{1}_{\mathfrak{T}}(s) > \mathbb{1}_{\mathfrak{T}}(s')$ if s is "closer" (in some sense) to s_A than s'. In summary, the function $\mathbb{1}_{\mathfrak{T}} : \mathcal{S} \to \mathcal{L}$ gives the degree of similarity of a prototype, which corresponds to the expert description. It is assumed that at least one state $s \in \mathcal{S}$ is fully consistent with the prototype: $\mathbb{1}_{\mathfrak{T}}(s) = 1$. Consider again the variable S: its actual value is unknown, but the

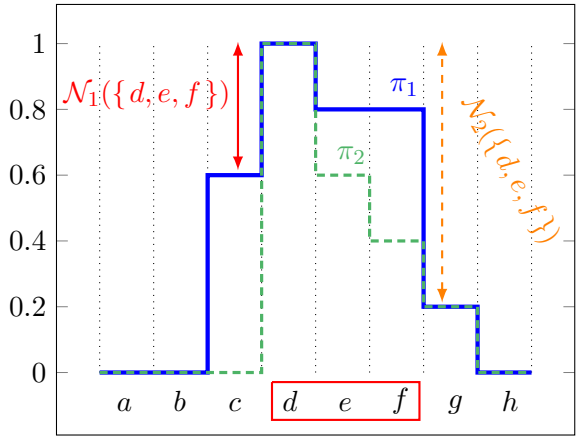


Figure I.8 – *Example of two possibility distributions over* $\Omega = \{a, b, c, d, e, f, g, h\}$*:* π_1 *(solid blue line) and* π_2 *(dashed green one), with* π_2 *more specific than* π_1*. The necessity measure* $\mathcal{N}_1$ *associated with* π_1 *is assessed on the event* $\{d, e, f\} \subset \Omega$*: the necessity degree is equal to* $0.4 = 1 - 0.6$*, as illustrated with solid red arrows. The necessity measure* $\mathcal{N}_2$ *associated with* π_2 *is assessed on the same event: the necessity degree is equal to* $0.8 = 1 - 0.2$*, as illustrated with dashed orange arrows.*

fuzzy set $\mathfrak{T}$ associated to the expert description leads to the possibility distribution of this variable: the possibility distribution π is defined as the membership function of the fuzzy set $\mathfrak{T}$: $\forall s \in \mathcal{S}$, $\pi(s) = \mathbb{1}_{\mathfrak{T}}(s)$. A possibility distribution of the variable $S \in \mathcal{S}$ is a characteristic function of a fuzzy set based on $\mathcal{S}$ where the elementary events $\{S = s\}$ are mutually exclusive: $\forall s_A \neq s_B \in \mathcal{S}$, $\Pi(S = s_A, S = s_B) = 0$.

Now the possibility degree of the event $S \in \{s_A, s_B\}$, *i.e.* of the event "the state is s_A or s_B" is $\Pi(\{s_A, s_B\}) = \max\{\mathbb{1}_{\mathfrak{T}}(s_A), \mathbb{1}_{\mathfrak{T}}(s_B)\} = \max_{s \in \{s_A, s_B\}} \pi(s)$: this is a maximum, as an extension of the logical "or" ($\vee$), usually defined on $\{0, 1\}$ or $\{\top, \bot\}$, and here defined on $\mathcal{L}$. This is easily generalized for more than two elementary events: $\forall A \subseteq \mathcal{S}$, $\Pi(A) = \max_{s \in A} \pi(s)$, which is also a consequence of Definition I.2.3. The possibility measure evaluates an event $A \subseteq \mathcal{S}$ by the most plausible elementary event in the event A. Hence, the normalization condition of Definition I.2.3, becomes $\max_{s \in \mathcal{S}} \pi(s) = 1$, which looks more like the probabilistic normalization $\sum_{s \in \mathcal{S}} \mathbf{p}(s) = 1$. As a conclusion, a possibility degree $\pi(s)$ can be seen as a "non-surprise" degree, since $1 - \pi(s)$ is considered as the "surprise" degree of the event $\{S = s\}$: the more an event is surprising, the more the complementary event is necessary.

Hence, for each possibility measure, a dual measure called *necessity* can be defined: the necessity degree of an event increases if the possibility degree of the opposite event decreases.

Definition I.2.4 (*Necessity Measure associated to* Π)

The necessity measure associated to Π *is the fuzzy measure* $\mathcal{N} : 2^{\Omega} \to [0, 1]$ *such that* $\forall A \subset \Omega$*,*

$$\mathcal{N}(A) = 1 - \Pi(\overline{A}),$$

where $\overline{A}$ *is the complementary event of* A *in* Ω*.*

Note that, as $\mathcal{N}(\emptyset) = 1 - \Pi(\Omega) = 0$, and as $\forall A, B$ subsets of Ω such that $A \subseteq B$, $\mathcal{N}(A) = 1 - \Pi(\overline{A}) \leqslant 1 - \Pi(\overline{B}) = \mathcal{N}(B)$, the necessity is indeed a fuzzy measure.

Note also that if an event $A \subseteq \Omega$ is not entirely possible ($\Pi(A) < 1$), then this event is not necessary at all ($\mathcal{N}(A) = 0$). As well, if the necessity degree of the event $A \subseteq \Omega$ is positive ($\mathcal{N}(A) > 0$), then it is entirely possible $\Pi(A) = 1$. Indeed, $\Pi(A \cup \overline{A}) = \Pi(\Omega) = 1$ and $\Pi(A \cup \overline{A}) = \max\left\{\Pi(A), \Pi(\overline{A})\right\}$. Thus $\max\left\{\Pi(A), \Pi(\overline{A})\right\} = 1$. Then, if $\Pi(A) < 1$, $\Pi(\overline{A}) = 1$ and $\mathcal{N}(A) = 0$. As well, if $\mathcal{N}(A) > 0$, $\Pi(\overline{A}) < 1$ and then $\Pi(A) = 1$.

Consider now $A, B \subseteq \Omega$: whereas $\Pi\{A \cup B\} = \max\{\Pi(A), \Pi(B)\}$,

$$\mathcal{N}(A \cap B) = \min\{\mathcal{N}(A), \mathcal{N}(B)\}.$$

Indeed,

$$\begin{aligned}
\mathcal{N}(A \cap B) &= 1 - \Pi(\overline{A \cap B}) \\
&= 1 - \Pi(\overline{A} \cup \overline{B}) \\
&= 1 - \max\left\{\Pi(\overline{A}), \Pi(\overline{B})\right\} \\
&= \min\left\{1 - \Pi(\overline{A}), 1 - \Pi(\overline{B})\right\} \\
&= \min\{\mathcal{N}(A), \mathcal{N}(B)\}.
\end{aligned}$$

The total ignorance is modeled by a possibility distribution π such that $\forall \omega \in \Omega$, $\pi(\omega) = 1$ *i.e.* any elementary event is possible. In this case, $\forall A \subseteq \Omega$, $A \neq \Omega$, $\mathcal{N}(A) = 1 - \Pi(\overline{A}) = 1 - \max_{\omega \in \overline{A}} \Pi(\omega) = 0$: apart from the universe Ω, no event is necessary.

On the contrary, the perfect knowledge that the actual elementary event is $\omega_A \in \Omega$ is modeled by a possibility distribution π such that $\pi(\omega_A) = 1$ and $\pi(\omega) = 0$, $\forall \omega \neq \omega_A$. The necessity of the singleton $\{\omega_A\}$ is also equal to one: $\mathcal{N}(\{\omega_A\}) = 1 - \Pi(\overline{\{\omega_A\}}) = 1$. The elementary event ω_A is necessary, and all the other have a null necessity degree: if $\omega_B \neq \omega_A$, $\mathcal{N}(\{\omega_B\}) = 1 - \Pi(\overline{\{\omega_B\}}) = 1 - \pi(\omega_A) = 0$.

These two particular cases give the intuition to formalize the knowledge, or the information, provided by a possibility distribution. This idea is led by the word *specificity*:

Definition I.2.5 (*Specificity*)

A possibility distribution π_2 is more specific (i.e. more informative) than another possibility distribution π_1, if $\forall \omega \in \Omega$,

$$\pi_2(\omega) \leqslant \pi_1(\omega).$$

Both notions of specificity and necessity are illustrated in Figure I.8.

The main concepts of Possibility Theory have been presented. Possibilistic planning models studied in this thesis are based on *conditional possibility distributions* and some independence assumptions, that is why the next section focuses on these notions.

I.2.2 Qualitative Conditioning and Possibilistic Independence

In practice, two kinds of independence between variables can be distinguished. The first one is the *decompositional independence*: two variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ are said independent in this sense, if the joint distribution of these two variables can be decomposed into two marginal distributions (one for each variable) without losing any information provided by the joint distribution. In the probabilistic framework, variables X and Y are independent in the decompositional sense, if $\forall E \subset \mathcal{X}, F \subset \mathcal{Y}$,

$$\mathbb{P}(X \in E, Y \in F) = \mathbb{P}(X \in E) \cdot \mathbb{P}(Y \in F). \tag{I.21}$$

The second kind of independence is called the *causal independence*: a variable X is independent of a variable Y in the causal sense, if the distribution of X is not modified when something about Y is learned *i.e.* there is no causality from Y towards X, or yet, Y does not influence X. Note that this independence relation is not necessarily symetric: "Y does not influence X" does not imply that "X does not influence Y". In terms of probability measures, the causal independence of X from Y can be written $\forall E \subset \mathcal{X}, F \subset \mathcal{Y}$,

$$\mathbb{P}(X \in E \mid Y \in F) = \mathbb{P}(X \in E). \tag{I.22}$$

In Probability Theory, both equations (I.21) and (I.22) are equivalent, if probabilities are positive, and then the causal independence is symetric: in this theory, decompositional and causal independence are equal and called simply independence.

In Possibility Theory, Zadeh [152] introduced the *non-interactivity independence*, or NI-independence, a decompositional independence inspired from Fuzzy Logic: as the fuzzy generalization of the "and" ($\wedge$) operator is the minimum (min), if events A and B do not interact together, the degree of truth of $A \cap B$ (*i.e.* of the event "A and B occur"), or its possibility degree, is $\Pi(A \cap B) = \min\{\Pi(A), \Pi(B)\}$. The analogy is also possible with the framework of fuzzy sets, as the fuzzy "intersection" ($\cap$) is represented by the minimum between the membership functions. This leads to the definition of the NI-independence for variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, replacing the event A (resp. B) by $\{X \in E\} \subseteq \Omega$ with $E \subseteq \mathcal{X}$ (resp. $\{Y \in F\} \subseteq \Omega$ with $F \subseteq \mathcal{Y}$):

Definition I.2.6 (*Non Interactivity Independence*)

Two events $A, B \subset \Omega$ are NI-independent if

$$\Pi(A \cap B) = \min\{\Pi(A), \Pi(B)\}.$$

Then, two variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ are NI-independent if $\forall E \subset \mathcal{X}, F \subset \mathcal{Y}$,

$$\Pi(X \in E, Y \in F) = \min\Big\{\Pi(X \in E), \Pi(Y \in F)\Big\}.$$

Finally, in terms of possibility distributions, it simply asserts that the joint one is equal to the minimum between the marginal ones: $\forall x \in \mathcal{X}$, $y \in \mathcal{Y}$,

$$\pi(x, y) = \min\{\pi(x), \pi(y)\},$$

where $\pi(x) = \Pi(\{X = x\})$, $\pi(y) = \Pi(\{Y = y\})$, and the joint possibility distribution $\pi(x, y)$ is $\Pi(\{X = x\} \cap \{Y = y\})$.

Note that, as Π is a fuzzy measure, Π is monotone, and then $\forall A, B \subset \Omega$, $\Pi(A \cap B) \leqslant \Pi(A)$ and $\Pi(A \cap B) \leqslant \Pi(B)$: thus, the inequality $\Pi(A \cap B) \leqslant \min\{\Pi(A), \Pi(B)\}$ is always true, with an equality when events are NI-independent. Figure I.9 illustrates a joint possibility distribution $\pi(x, y)$ over $\mathcal{X} \times \mathcal{Y}$ whose corresponding variables are not NI-independent, whereas Figure I.10 represents a similar distribution whose variables are NI-independent. Note that, if a joint distribution $\pi(x, y)$ is given, $\pi(x)$ can be computed from it by marginalization using the max operator over $\mathcal{Y}$:

$$\begin{aligned}
\pi(x) &= \Pi\Big(\{X = x\}\Big) \\
&= \Pi\Big(\cup_{y \in \mathcal{Y}} \{X = x\} \cap \{Y = y\}\Big) \\
&= \max_{y \in \mathcal{Y}} \Pi\Big(\{X = x\} \cap \{Y = y\}\Big) \\
&= \max_{y \in \mathcal{Y}} \pi(x, y).
\end{aligned}$$

The NI-independence leads to a first qualitative possibilistic conditioning, introduced by Hisdal [74]. Indeed, the conditional possibility degree of an event $A \subset \Omega$ given an event $B \subset \Omega$, namely $\Pi(A|B)$, can be obtained from $\Pi(B)$ and $\Pi(A \cap B)$, as a solution of the following equation:

$$\Pi(A \cap B) = \min\Big\{\Pi(A|B), \Pi(B)\Big\}. \tag{I.23}$$

Intuitively, once conditioned on event B, event A (or rather "$A|B$") does not interact with the event B anymore. Moreover Equation I.23 is close to the probabilistic equation $\mathbb{P}(A \cap B) =$

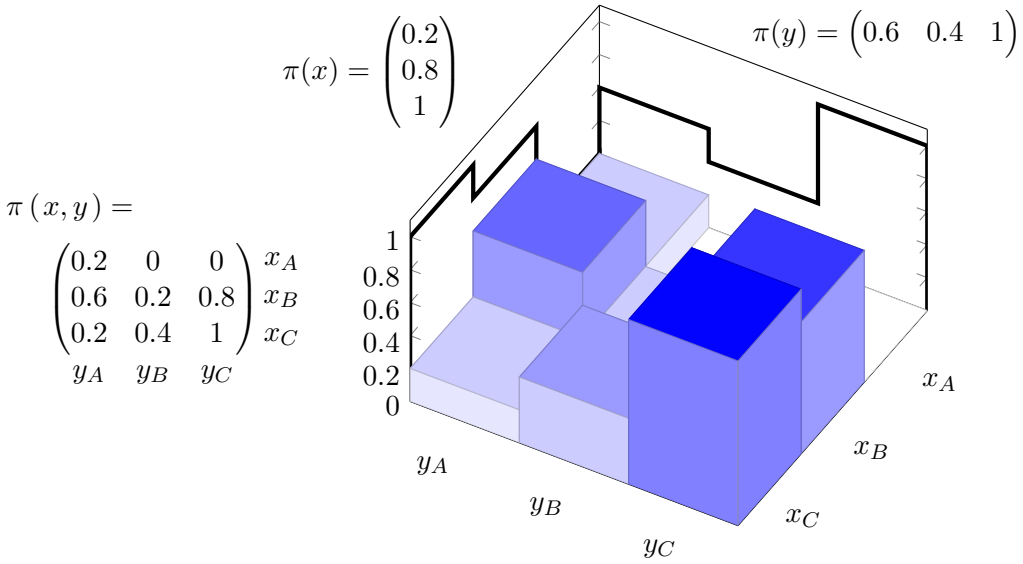


Figure I.9 – *Example of a joint distribution over* $\mathcal{X} \times \mathcal{Y} = \{x_A, x_B, x_C\} \times \{y_A, y_B, y_C\}$ *without any independence.*

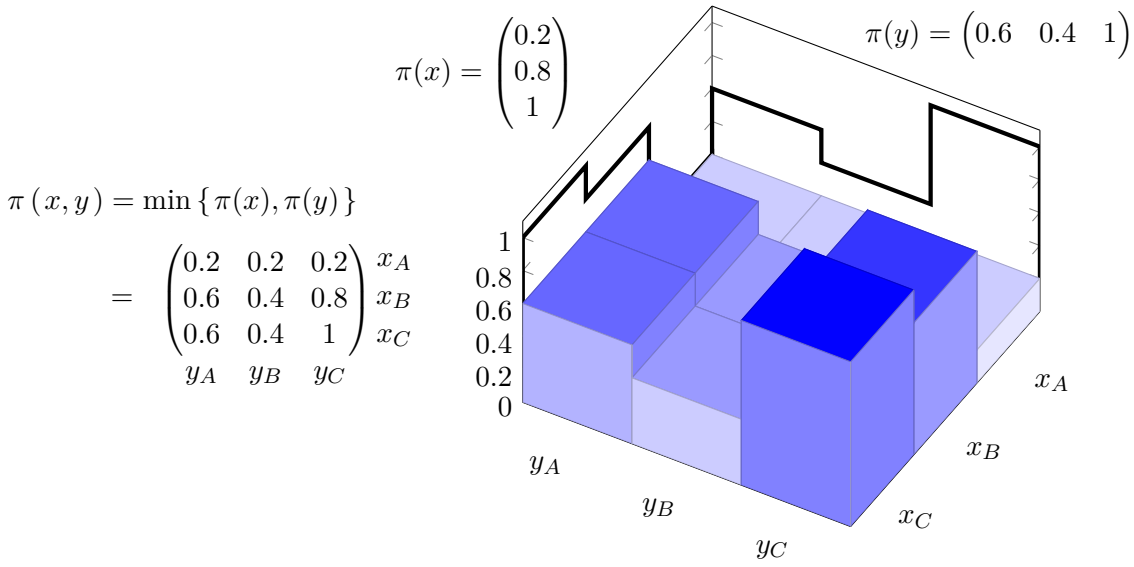


Figure I.10 – *Example of a joint distribution over* $\mathcal{X} \times \mathcal{Y} = \{x_A, x_B, x_C\} \times \{y_A, y_B, y_C\}$ *when* X *and* Y *are NI-independent (no M-independence).*

$\mathbb{P}(A|B) \cdot \mathbb{P}(B)$ which comes from the definition of the probabilistic conditioning. Possibility Theory looks very similar to Probability Theory observing that the addition ($+$) in Probability Theory becomes a maximum (max) in Possibility Theory, and the multiplication ($\times$) becomes a minimum (min). However, Quantitative Possibility Theory keeps the multiplication operator ($\times$) for conditioning and computing joint possibility measures: in this theory, the classical conditioning is then equivalent to the Dempster rule of conditioning [46], an evidentialist (BFT, see Introduction) extension of the well-known Bayes rule.

The classical qualitative possibilistic conditioning [58], counterpart of Bayes rule, is the least specific solution of Equation I.23:

Definition I.2.7 (*Qualitative Possibilistic Conditioning*)

$\forall A, B \subset \Omega$, *such that* $\Pi(B) > 0$,

$$\Pi\left(A \mid B\right) = \begin{cases} 1 & \text{if } \Pi(B) = \Pi(A \cap B), \\ \Pi(A \cap B) & \text{otherwise.} \end{cases} \tag{I.24}$$

The conditional possibility distributions of the variable $X \in \mathcal{X}$ *knowing variable* $Y \in \mathcal{Y}$ *is thus,* $\forall x \in \mathcal{X}$, $\forall y \in \mathcal{Y}$ *such that* $\pi(y) > 0$,

$$\pi\left(x \mid y\right) = \begin{cases} 1 & \text{if } \pi(y) = \pi(x, y), \\ \pi(x, y) & \text{otherwise.} \end{cases} \tag{I.25}$$

where $\pi\left(x \mid y\right) = \Pi\left(X = x \mid Y = y\right)$.

The meaning of the conditioning (I.24) can be explained as follows: when the event A contains the elementary event which has the higest possibility degree in B, *i.e.* if $\Pi(B) = \Pi(A \cap B)$, then $\Pi\left(A \mid B\right) = 1$. Indeed, as conditioning on B assumes that the new set of possible elementary events (universe) is B, A has the maximal possibility degree among the new universe B: the possibility measure conditioned on B is normalized, setting to 1 the possibility degree of most plausible events in B. As qualitative possibility distributions only define a ranking between the events, the possibility degree of events A such that $\Pi(A \cap B) < \Pi(B)$ (second case of Equation I.24) are simply set to $\Pi(A \cap B)$. As $\min\left\{\Pi\left(A \mid B\right), \Pi(B)\right\} = \Pi\left(A \mid B\right)$, Equation I.24 is true thanks to this choice.

Defining the qualitative possibilistic conditioning as previously (I.2.7), the non-interactivity independence (Definition I.2.6) between variables X and Y corresponds to the fact that $\forall x, y \in \mathcal{X} \times \mathcal{Y}$ either the knowledge that the variable X is equal to x does not decrease the possibility degree of $Y = y$, or the knowledge that the variable Y is equal to y does not decrease the possibility degree of $X = x$.

$$\pi(x) = \pi(x \mid y) = \begin{pmatrix} 0.2 \\ 1 \\ 1 \end{pmatrix} \qquad \pi(y) = \begin{pmatrix} 0.6 & 0.4 & 1 \end{pmatrix}$$

$$\pi(x, y) = \min\{\pi(x), \pi(y)\} = \begin{array}{c} \\ = \\ \\ \end{array} \begin{array}{c} \begin{pmatrix} 0.2 & 0.2 & 0.2 \\ 0.6 & 0.4 & 1 \\ 0.6 & 0.4 & 1 \end{pmatrix} \begin{matrix} x_A \\ x_B \\ x_C \end{matrix} \\ \begin{matrix} y_A & y_B & y_C \end{matrix} \end{array}$$

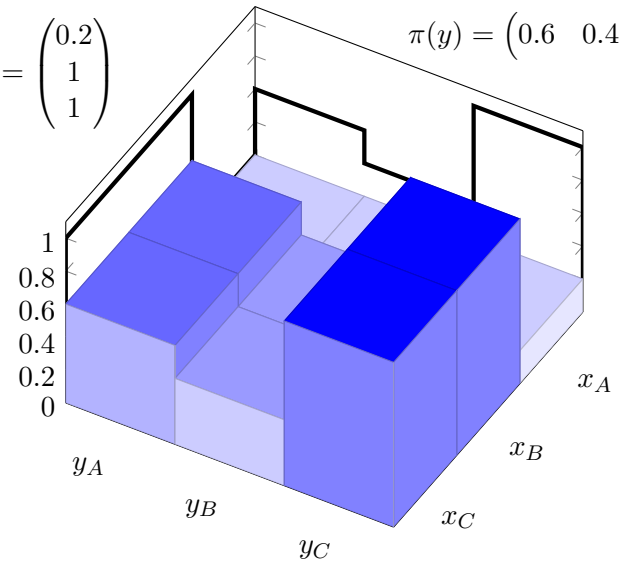


Figure I.11 – *Example of a joint distribution over* $\mathcal{X} \times \mathcal{Y} = \{x_A, x_B, x_C\} \times \{y_A, y_B, y_C\}$ *when* X *is M-independent from* Y.

Theorem 10

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad \pi(x, y) = \min\{\pi(x), \pi(y)\}$$

(non interactivity independence)

$$\Updownarrow$$

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad \pi(x) \leqslant \pi(x \mid y) \text{ or } \pi(y) \leqslant \pi(y \mid x)$$

(either the knowledge of Y *does not bring any knowledge about* X*, or the knowledge of* X *doesn't bring any knowledge about* Y*.)*

The proof is given in Annex A.15.

Another independence called the *Min-based independence*, or M-independence, comes from the conditioning (I.2.7):

Definition I.2.8 (*Min-based "Causal" Independence*)

The event $A \subset \Omega$ *is M-independent from the event* $B \subset \Omega$ *if*

$$\Pi(A \mid B) = \Pi(A).$$

As well, the variable X *is M-independent from the variable* Y *if their distributions are such that*

$$\pi(x \mid y) = \pi(x).$$

Note that, using Theorem 10, it follows that the M-independence implies the NI-independence: it can be also observed replacing $\Pi(A \mid B)$ by $\Pi(A)$ in the equation I.23, as both possibility degrees are equal.

This independence is causal, and not symetric. Figure I.11 displays an example of a joint distribution $\pi(x, y)$ such that $\pi(x \mid y) = \pi(x)$. However $\pi(y \mid x) \neq \pi(y)$: indeed, if $X = x_A$,

Y is fully unknown, *i.e.* $\pi(y \mid x_A) = \begin{pmatrix}1 & 1 & 1\end{pmatrix}$, if $X = x_B$, $\pi(y \mid x_B) = \begin{pmatrix}0.6 & 0.4 & 1\end{pmatrix}$, and the same for $X = x_C$. This illustrates the fact that the M-independence is not symetric.

The symetrized version of the M-independence is the *Symetric Min-Based Independence* or MS-independence:

Definition I.2.9 (*Symetric Min-Based Independence*)

Variables X and Y are said to be MS-independent if X is M-independent from Y and Y is M-independent from X.

However, this symetric independence is too restrictive: one of the variable has to be entirely unknown, as highlighted by the following theorem.

Theorem 11

If X and Y are MS-independent, then $\forall x \in \mathcal{X}$, $\Pi(X = x) = 1$ or $\forall y \in \mathcal{Y}$, $\Pi(Y = y) = 1$.

The proof is given in Annex A.16.

Finally, we present a second qualitative possibilistic conditioning, proposed in [40], based on the previous one (Definition I.2.7): this one ensures that the posterior distribution is not less specific than the prior.

Definition I.2.10 (*Alternative Qualitative Possibilistic Conditioning*)

This alternative conditioning is denoted by $\pi(x\|y)$ and is a modified version of the classical one (Definition I.2.7): $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\pi(x\|y) = \begin{cases} \pi(x) & \text{if } \pi(x' \mid y) \geqslant \pi(x'), \forall x' \in \mathcal{X}, \\ \pi(x \mid y) & \text{otherwise.} \end{cases}$$

A more general presentation of the various possibilistic conditionings and independences and their consequences on possibilistic graphical models is available in the thesis of N.Ben Amor [10].

We just introduced that the operators min and max will be present in most of equations below as the studied models are purely qualitative. Some properties about these operators, which are used in the following sections, end this one.

Property I.2.1

Consider the functions $f : \Omega \to \mathcal{L}$ and $\lambda \in \mathcal{L}$:

$$\max_{\omega\in\Omega} \{ 1 - f(\omega) \} = 1 - \min_{\omega\in\Omega} f(\omega), \tag{I.26}$$

$$\min_{\omega\in\Omega} \{ 1 - f(\omega) \} = 1 - \max_{\omega\in\Omega} f(\omega), \tag{I.27}$$

$$\min_{\omega\in\Omega} \min \{ f(\omega), \lambda \} = \min \left\{ \min_{\omega\in\Omega} f(\omega), \lambda \right\}, \tag{I.28}$$

$$\min_{\omega\in\Omega} \max \{ f(\omega), \lambda \} = \max \left\{ \min_{\omega\in\Omega} f(\omega), \lambda \right\}, \tag{I.29}$$

$$\max_{\omega\in\Omega} \min \{ f(\omega), \lambda \} = \min \left\{ \max_{\omega\in\Omega} f(\omega), \lambda \right\} \tag{I.30}$$

$$\text{and } \operatorname*{argmax}_{\omega\in\Omega} f(\omega) \subseteq \operatorname{argmax}_{\omega\in\Omega} \min \{ f(\omega), \lambda \}, \tag{I.31}$$

$$\max_{\omega\in\Omega} \max \{ f(\omega), \lambda \} = \max \left\{ \max_{\omega\in\Omega} f(\omega), \lambda \right\}, \tag{I.32}$$

$$\text{and } \operatorname*{argmax}_{\omega\in\Omega} f(\omega) \subseteq \operatorname{argmax}_{\omega\in\Omega} \max \{ f(\omega), \lambda \}. \tag{I.33}$$

Let us introduce now $g : \Omega \to \mathcal{L}$ and suppose that $\exists \omega^* \in \Omega$ such that $g(\omega^*) = 0$:

$$\min_{\omega\in\Omega} \max \left\{ \min \{ \lambda, f(\omega) \}, g(\omega) \right\} = \min_{\omega\in\Omega} \min \left\{ \lambda, \max \{ f(\omega), g(\omega) \} \right\}. \tag{I.34}$$

As well, if we introduce $h : \Omega \to \mathcal{L}$ and suppose that $\exists \omega^* \in \Omega$ such that $h(\omega^*) = 1$,

$$\max_{\omega\in\Omega} \min \left\{ \max \{ \lambda, f(\omega) \}, h(\omega) \right\} = \max_{\omega\in\Omega} \max \left\{ \lambda, \min \{ f(\omega), h(\omega) \} \right\}. \tag{I.35}$$

The proof is given in Annex A.17.

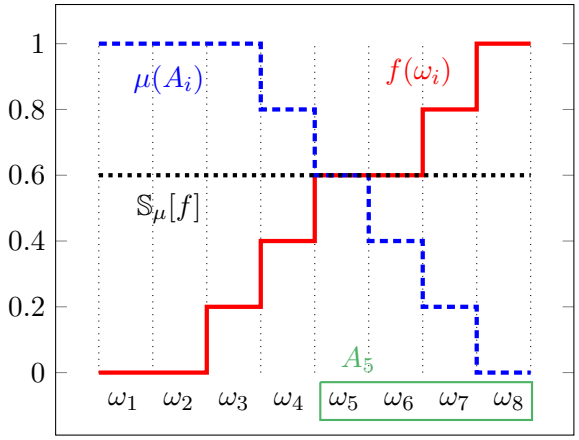


Figure I.12 – *Illustration of the result of the Sugeno Integral: the x-axis represents the set $\Omega = \{\omega_1, \ldots, \omega_{\#\Omega}\}$, where $\forall i \in \{1, \ldots, \#\Omega - 1\}$, $f(\omega_i) \leqslant f(\omega_{i+1})$. The y-axis is $\mathcal{L}$. The red line represents the degrees $f(\omega_i)$, the dashed blue one represents the degrees $\mu(A_i)$ with $A_i = \{\omega_i, \ldots, \omega_{\#\Omega}\}$, and the black dotted one is the result of the Sugeno Integral.*

I.2.3 Qualitative Criteria

As detailed in the first section of this chapter, the criterion used to quantify the quality of a strategy for a given (PO)MDP is the expectation of the reward, *i.e.* the integral of the reward function with respect to the probability measure. The concept of Integral has been extended to the framework of fuzzy measures. When the measure is quantitative, the extension is called the *Choquet Integral* [35]. In the case of a qualitative measure, the resulting objet is the *Sugeno Integral* [139]. As the planning framework that will be studied is qualitative, we define here the Sugeno Integral:

Definition I.2.11 (*Sugeno Integral*)

The Sugeno Integral of a function $f : \Omega \to \mathcal{L}$ with respect to the capacity (fuzzy measure) $\mu : 2^\Omega \to \mathcal{L}$ is

$$\mathbb{S}_\mu [f] = \max_{i=1}^{\#\Omega} \min \{ f(\omega_i), \mu(A_i) \} \tag{I.36}$$

$$= \min_{i=1}^{\#\Omega} \max \{ f(\omega_i), \mu(A_{i+1}) \} \tag{I.37}$$

where $f(\omega_1) \leqslant \ldots \leqslant f(\omega_{\#\Omega})$, $A_i = \{\omega_i, \omega_{i+1}, \ldots, \omega_{\#\Omega}\}$ and $A_{\#\Omega+1} = \emptyset$.

The proof of the equality is given in Annex A.18.

As illustrated in Figure I.12, the Sugeno integral of $f : \Omega \to \mathcal{L}$ with respect to the fuzzy measure μ is the highest degree $\lambda \in \mathcal{L}$ such that the measure μ of $\{\omega \mid f(\omega) \geqslant \lambda\}$ is higher or equal to λ. As an example, the h-index (or Hirsh index) is the Sugeno integral of the function $paper \mapsto \#citations$ with respect to the counting measure.

The Sugeno integral with respect to a possibility measure, and the one with respect to a necessity measure, lead to two criteria presented below. Before that, the following theorem rewrites both integral in a more simple way.

Theorem 12 (*Sugeno Integrals with respect to Possibility and Necessity measures*)

$$\mathbb{S}_\Pi[f] = \max_{i=1}^{\#\Omega} \min \{ f(\omega_i), \pi(\omega_i) \}, \tag{I.38}$$

$$\mathbb{S}_\mathcal{N}[f] = \min_{i=1}^{\#\Omega} \max \{ f(\omega_i), 1 - \pi(\omega_i) \}. \tag{I.39}$$

are rewritings of the Sugeno integrals with respect to possibility and necessity measures.

The proof is given in Annex A.19.

These integrals can be seen as the possibilistic expectations of the variable $f : \Omega \to \mathcal{L}$. Let us introduce the variable $S : \Omega \to \mathcal{S}$ whose possibility distribution is $\pi(s) = \Pi(\{S = s\}) = \Pi(\{\omega \in \Omega \mid S(\omega) = s\}) = \max_{\{\omega \in \Omega \mid S(\omega) = s\}} \pi(\omega)$. We can note for instance that the Sugeno Integral of the (classical) characteristic function of the event $\{S \in A\}$ with $A \subseteq \mathcal{S}$, namely

$\mathbb{1}_{\{\,S\in A\,\}}(\omega) = \begin{cases} 1 & \text{if } S(\omega)\in A \\ 0 & \text{otherwise} \end{cases}$, is equal to the possibility degree of this event:

$$\begin{aligned}
\mathbb{S}_\Pi\left[\mathbb{1}_{\{\,S\in A\,\}}\right] &= \max_{\omega\in\Omega} \min\left\{\mathbb{1}_{\{\,S\in A\,\}}(\omega), \pi(\omega)\right\} \\
&= \max_{s\in\mathcal{S}} \max_{\{\,\omega\in\Omega \mid S(\omega)=s\,\}} \min\{\,\mathbb{1}_A(s), \pi(\omega)\,\} \\
&= \max_{s\in\mathcal{S}} \min\{\,\mathbb{1}_A(s), \pi(s)\,\} \qquad \text{(I.40)} \\
&= \max_{s\in\mathcal{S}} \begin{cases} \pi(s) & \text{if } s\in A \\ 0 & \text{otherwise} \end{cases} \\
&= \max_{s\in A} \pi(s) = \Pi(S\in A). \qquad \text{(I.41)}
\end{aligned}$$

where line (I.40) comes from equation (I.30) of Property (I.2.1). In the same way,

$$\begin{aligned}
\mathbb{S}_\mathcal{N}\left[\mathbb{1}_{\{\,S\in A\,\}}\right] &= \min_{s\in\mathcal{S}} \max\{\,\mathbb{1}_A(s), 1-\pi(s)\,\} \\
&= \min_{s\in\mathcal{S}} \{\,1-\min\{\,1-\mathbb{1}_A(s), \pi(s)\,\}\,\} \qquad \text{(I.42)} \\
&= 1-\max_{s\in\mathcal{S}} \min\{\,\mathbb{1}_{\overline{A}}(s), \pi(s)\,\} \qquad \text{(I.43)} \\
&= 1-\max_{s\in\overline{A}} \pi(s) = 1-\Pi\left(\left\{S\in\overline{A}\right\}\right) = \mathcal{N}\left(\{\,S\in A\,\}\right).
\end{aligned}$$

where lines (I.42) and (I.43) come from equations (I.26) and (I.27) of Property (I.2.1). These remarks are the counterparts of the probabilistic equality $\mathbb{E}\left[\mathbb{1}_{\{\,S\in A\,\}}\right] = \mathbb{P}(\{\,S\in A\,\})$.

Qualitative Possibilistic Decision Criteria, *i.e.* functions $\mathcal{A}\to\mathcal{L}$ measuring the accuracy of actions given a possibilistic and a preference model, have been proposed in [125, 62, 59], based on Sugeno integrals (I.38) and (I.39). Let us recall that the set $\mathcal{S}$ (resp. $\mathcal{A}$) is as previously the finite set of system states s (resp. of actions a). The variable representing the system state is $S\in\mathcal{S}$. Let $(\pi_a)_{a\in\mathcal{A}}$ be a family of possibility distributions over $\mathcal{S}$, *i.e.* $\forall a\in\mathcal{A}$, $\pi_a(s) = \Pi_a(\{\,S=s\,\})$ is the possibility degree of the situation $\{\,S=s\,\}\subset\Omega$ after selecting action $a\in\mathcal{A}$. Let function $\rho:\mathcal{S}\to\mathcal{L}$ be the preference function, defining the preference degree of each system state $s\in\mathcal{S}$.

Definition I.2.12 (*Qualitative Decision Criteria*)

Let π_a be the possibility distribution describing the uncertainty about the system state given that action $a\in\mathcal{A}$ has been selected, and $\rho(s)$ the preference of the system state $s\in\mathcal{S}$. Using formula (I.38) with $f=\rho(S)$, the Sugeno integral of the preference with respect to the possibility measure Π_a leads to an optimistic criteria for $a\in\mathcal{A}$:

$$\mathbb{S}_{\Pi_a}[\rho(S)] = \max_{s\in\mathcal{S}} \min\{\,\rho(s), \pi_a(s)\,\}. \qquad \text{(I.44)}$$

As well, using formula (I.39) with $f=\rho(S)$, the Sugeno integral of the preference with respect to the necessity measure associated to Π_a, $\mathcal{N}_a$, leads to a cautious criteria for $a\in\mathcal{A}$:

$$\mathbb{S}_{\mathcal{N}_a}[\rho(S)] = \min_{s\in\mathcal{S}} \max\{\,\rho(s), 1-\pi_a(s)\,\}. \qquad \text{(I.45)}$$

These criteria can be understood best with the fuzzy sets vision: a possibility distribution $\pi_a:\mathcal{S}\to\mathcal{L}$ is the characteristic (or membership) function of the fuzzy set of the possible system states after selecting $a\in\mathcal{A}$, denoted by $\mathfrak{T}^a$ *i.e.* $\pi(s) = \mathbb{1}_{\mathfrak{T}^a}(s)$. The preference function $\rho:\mathcal{S}\to\mathcal{L}$ can also be viewed as the characteristic function of the fuzzy set denoted by $\mathfrak{R}$

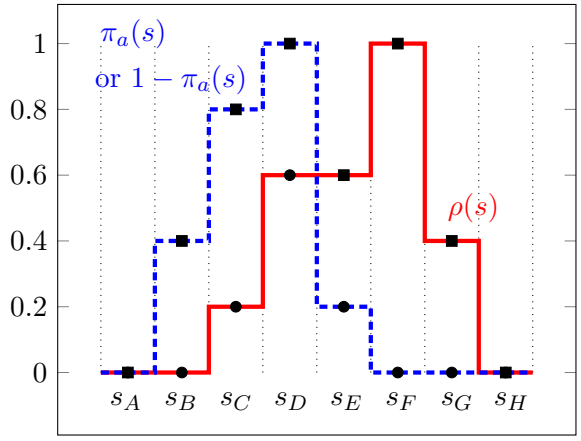


Figure I.13 – *Illustration of the qualitative criteria. The solid red line is* $\rho(s) = \mathbb{1}_{\mathfrak{R}}(s)$*. The dashed blue line may represent* $\pi_a(s) = \mathbb{1}_{\mathfrak{T}^a}(s)$*, and the black circles represent the characteristic function of the intersection* $\mathbb{1}_{\mathfrak{T}^a \cap \mathfrak{R}}(s) = \min\{\rho(s), \pi_a(s)\}$*. Actions maximizing the optimistic criterion (I.44), maximizes the highest membership degree* $\max_{s\in\mathcal{S}} \mathbb{1}_{\mathfrak{T}^a \cap \mathfrak{R}}(s)$*, which is here equal to* 0.6*, the degree of* s_D*. The dashed blue line may also represent* $1-\pi_a(s) = \mathbb{1}_{\overline{\mathfrak{T}^a}}(s)$*, and the black squares represent the characteristic function of the union* $\mathbb{1}_{\overline{\mathfrak{T}^a} \cup \mathfrak{R}}(s) = \max\{\rho(s), 1-\pi_a(s)\}$*. Actions maximizing the pessimistic criterion (I.45), maximizes the lowest membership degree* $\min_{s\in\mathcal{S}} \mathbb{1}_{\overline{\mathfrak{T}^a} \cup \mathfrak{R}}(s)$*, which is here equal to* 0*, as* s_A *and* s_H *are totally possible and unpleasant.*

representing the preferred system states: $\rho(s) = \mathbb{1}_{\mathfrak{R}}(s)$. Finally, the characteristic function of the fuzzy set of plausible and preferred states after selecting action $a \in \mathcal{A}$, *i.e.* $\mathfrak{T}^a \cap \mathfrak{R}$, is $\mathbb{1}_{\mathfrak{T}^a \cap \mathfrak{R}} = \min\{\mathbb{1}_{\mathfrak{T}^a}, \mathbb{1}_{\mathfrak{R}}\} = \min\{\pi_a, \rho\}$.

An action $a \in \mathcal{A}$ maximizing the optimistic criterion (I.44), is thus an action that maximizes the highest membership degree of $\mathfrak{T}^a \cap \mathfrak{R}$, *i.e.* of the fuzzy set of possible and preferred states. This criterion is optimistic because it maximizes the degree of the best situation, but does not ensure that unwanted states are avoided by the system.

The characteristic function of the complementary set of $\mathfrak{T}^a$, denoted by $\overline{\mathfrak{T}^a}$ is $\mathbb{1}_{\overline{\mathfrak{T}^a}} = 1 - \mathbb{1}_{\mathfrak{T}^a} = 1 - \pi_a$: the fuzzy set of implausible system states. An action $a \in \mathcal{A}$ maximizing the pessimistic criterion (I.45), is thus an action that maximizes the lowest membership degree of $\overline{\mathfrak{T}^a} \cup \mathfrak{R}$, *i.e.* of the fuzzy set of the system states which are implausible or preferred. An action that maximizes the lowest degree of this fuzzy set, tries to make all the system states either implausible or preferred, *i.e.* to ensure that if any system state is plausible it is preferred: it maximizes the "degree" of the inclusion $\mathfrak{T} \subseteq \mathfrak{R}$.

Note that, for a function $f : \mathcal{A} \to \mathcal{L}$, $\max_{a\in A} f(a) = 1 - \min_{a\in A} f(a)$, which can be shown as equation (I.27) of Property I.2.1. Thus,

$$\operatorname*{argmax}_{a\in A} \left\{ \min_{s\in\mathcal{S}} \max\{\rho(s), 1-\pi_a(s)\} \right\} = \operatorname*{argmin}_{a\in A} \left\{ 1 - \min_{s\in\mathcal{S}} \max\{\rho(s), 1-\pi_a(s)\} \right\}.$$

As $1 - \min_{s\in\mathcal{S}} \max\{\rho(s), 1-\pi_a(s)\} = \max_{s\in\mathcal{S}} \left\{ 1 - \max\{\rho(s), 1-\pi_a(s)\} \right\} = \max_{s\in\mathcal{S}} \min\{1 - \rho(s), \pi_a(s)\}$ (see equations (I.26) and (I.27) of Property I.2.1), the action $a \in \mathcal{A}$ minimizes the highest membership degree of the fuzzy set $\mathfrak{T}^a \cap \overline{\mathfrak{R}}$, *i.e.* the fuzzy set of plausible and unwanted system states: the pessimistic criterion tries to keep down all membership degrees of this set.

The pessimistic criterion (I.45) tends to avoid unwanted system states, whereas the optimistic criterion (I.44) wants to make it possible that the system reaches preferred ones. Figure (I.13) illustrates the result of the criteria for a given action $a \in \mathcal{A}$.

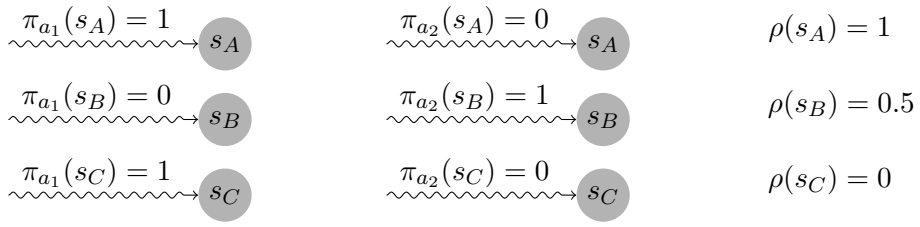


Figure I.14 – *Illustration of the example of Section I.2.3 about qualitative criteria. Action a_1 maximizes the optimistic criterion (I.44), which can lead to the best state (s_A), but also the worst (s_C). On the contrary, action a_2 maximizes the pessimistic criterion (I.45) since the worst state is not reachable with this action.*

The following toy example, illustrated in Figure I.14 is meant to present the two criteria in practice: let the set of system states $\mathcal{S}$ be $\{ s_A, s_B, s_C \}$ and the set of actions $\mathcal{A} = \{ a_1, a_2 \}$. The preference model and the uncertainty model are described respectively by ρ and $(\pi_a)_{a \in \mathcal{A}}$:

- $1 = \rho(s_A) > \rho(s_B) > \rho(s_C) = 0$;
- selecting action a_1, $\pi_{a_1}(s_A) = \pi_{a_1}(s_C) = 1$, and $\pi(s_B) = 0$;
- selecting action a_2, $\pi_{a_2}(s_A) = \pi(s_C) = 0$, and $\pi(s_B) = 1$, *i.e.* the system is in state s_B deterministically.

As $\min \{ \rho(s), \pi_{a_1}(s) \} = \begin{cases} 1 & \text{if } s = s_A, \\ 0 & \text{otherwise.} \end{cases}$, the optimistic criterion is equal to $S_{\Pi_{a_1}}[r(S)] = 1$ for a_1. Now, as $\min \{ \rho(s), \pi_{a_2}(s) \} = \begin{cases} \rho(s_B) & \text{if } s = s_B, \\ 0 & \text{otherwise.} \end{cases}$, the optimistic criterion is equal to $S_{\Pi_{a_2}}[r(S)] = \rho(s_B)$ for a_2. Thus, as $\rho(s_B) < 1$, a_1 maximizes the optimistic criterion (I.44).

The optimistic criterion is maximized by action a_1, because with this action, the best system state, s_A, is entirely possible. However, this action makes also the worst system state, s_C entirely possible: state s_A is not necessary at all: $\mathcal{N}(\{ s_A \}) = 1 - \pi(\{ s_B, s_C \}) = 1 - \max(\pi(s_B), \pi(s_C)) = 0$. A more cautious action is a_2, whose preference of the reached state (s_B) is lower than 1, but certain.

As expected, the action a_2 maximizes the cautious criterion (I.45): $\max \{ \rho(s), 1 - \pi_{a_1}(s) \} = \begin{cases} 1 & \text{if } s = s_A \text{ or } s_B, \\ 0 & \text{otherwise.} \end{cases}$ and then the cautious criterion is equal to 0 for a_1. It is greater with a_2: $\max \{ \rho(s), \pi_{a_2}(s) \} = \begin{cases} 1 & \text{if } s = s_A \text{ or } s_C, \\ \rho(s_B) & \text{otherwise.} \end{cases}$ and thus the criterion is equal to $\rho(s_B) > 0$ for a_2. This choice is more cautious since $\mathcal{N}_{a_2}(\{ s_B \}) = 1 - \Pi(\{ s_A, s_C \}) = 1$, *i.e.* the preference of the state will be $\rho(s_B)$ with certainty.

This section ends with a remark about this qualitative framework: possibility degrees are compared to preference degrees in the presented criteria. This requires a *commensurability* assumption, *i.e.* these comparisons must mean something. When values of ρ are in $\{ 0, 1 \}$, this assumption is not necessary as system states with a preference of 1 are the goals, and other states are not. How to model problems in practice with these settings will be detailed in the experimental parts.

I.2.4 π-MDPs

This model, presented in [123, 122, 121], is a qualitative possibilistic version of the probabilistic MDPs detailed in Section I.1.2, based on the criteria (optimistic and pessimistic) presented

in Section I.2.3: this version is called *Qualitative Possibilistic Markov Decision Process*, or π-MDP.

The finite set of system states, describing the agent and its environment, remains denoted by $\mathcal{S}$, as seen in Section I.1.2 where probabilistic MDPs are presented. The finite set of action is always $\mathcal{A}$ and $\mathcal{L}$ is the possibility scale $\left\{0, \frac{1}{k}, \ldots, 1\right\}$, with $k \geqslant 2$.

As in the probabilistic case, this model considers that successive system states, represented by the sequence of variables $(S_t)_{t\in\mathbb{N}}$ with $S_t \in \mathcal{S}\ \forall t \geqslant 0$, is Markovian. In this qualitative possibilistic framework, it means that the sequence $(S_t)_{t\in\mathbb{N}}$ is such that $\forall t \geqslant 0, \forall (s_0, s_1, \ldots, s_{t+1}) \in \mathcal{S}^{t+2}$ and for each action sequence of action $(a_t)_{t\geqslant 0} \in \mathcal{A}^{\mathbb{N}}$, S_{t+1} is M-independent (see Definition I.2.8) from variables $\{S_0, \ldots, S_{t-1}\}$, conditioned on $\{S_t = s\}$ and a_t:

$$\Pi\Big(\ S_{t+1} = s_{t+1} \ \Big|\ S_t = s_t, a_t\ \Big) = \Pi\Big(\ S_{t+1} = s_{t+1} \ \Big|\ S_t = s_t, S_{t-1} = s_{t-1}, \ldots, S_0 = s_0, (a_t)_{t\geqslant 0}\ \Big). \tag{I.46}$$

Using this property, dynamics of the system are fully described with possibilistic transitions $\pi_t(s' \mid s, a) = \Pi\Big(\ S_{t+1} = s' \ \Big|\ S_t = s, a\ \Big) \in \mathcal{L}$: $\forall t \geqslant 0$, $(s, s') \in \mathcal{S}^2$ and $a \in \mathcal{A}$, $\pi_t(s' \mid s, a)$ is the possibility degree, that at time step t, the system reaches the state s' when the agent selects action a, conditioned to the fact that the current state is s. Finally a π-MDP is entirely defined with the sequence of preference functions $(\rho_t)_{t=0}^{H-1}$, where $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\rho_t(s, a)$ is the preference degree when the system state is s and the agent selects action a at time step t. The final preference function, Ψ, gives for each system state $s \in \mathcal{S}$, the preference degree if $S_H = s$: $\Psi(s)$. With the previous notations for the preference functions, $\Psi(s) = \rho_H(s, a)$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}$. Figure I.2 in the MDP section (Section I.1.2) is a good representation of a π-MDP, replacing rewards by preferences, and probability distributions by possibility distributions.

In order to easily derive the MDP criteria from the qualitative possibilistic criteria (I.44) and (I.45), let us introduce, for an horizon $H \geqslant 0$, a *H-length trajectory* $\mathcal{T} = (s_1, \ldots, s_H)$, and $\mathcal{T}_H = \mathcal{S}^H$ the set of such trajectories. A decision rule is denoted by $\delta : \mathcal{S} \to \mathcal{L}$, and a H-length strategy is a sequence of decision rules δ_t: $(\delta_t)_{t=0}^{H-1}$. The set of all the H-length strategies is denoted by Δ_H. In [120], for a given H-length strategy $(\delta) \in \Delta_H$, a given sequence of system states $\mathcal{T} = (s_1, \ldots, s_H) \in \mathcal{T}_H$, and a given initial state $s_0 \in \mathcal{S}$, the *preference of an H-length trajectory from s_0* is defined as the lowest preference degree along s_0 and the trajectory:

$$\rho\Big(\mathcal{T}, (\delta)\Big) = \min\left\{ \min_{t=0}^{H-1} \rho\Big(s_t, \delta_t(s_t)\Big), \Psi(s_H) \right\}.$$

This is the possibilistic counterpart of the sum $\sum_{t=0}^{H-1} r_t\Big(S_t, d_t(S_t)\Big) + R(S_H)$, reward aggregation of the probabilistic framework. Note that any preference aggregation of the qualitative possibilistic framework has to result in a degree $\lambda \in \mathcal{L}$.

Using the Markov property of this system state process, for a given initial system state $s_0 \in \mathcal{S}$, a horizon $H \in \mathbb{N}$, and a strategy $(\delta_t)_{t=0}^{H-1}$, the possibility degree of the trajectory $\mathcal{T} = (s_1, \ldots, s_H)$ is

$$\Pi\left(S_H = s_h, S_{H-1} = s_{h-1}, \ldots, S_1 = s_1 \middle| S_0 = s_0, (\delta_t)_{t=0}^{H-1}\right) = \min_{t=0}^{H-1} \pi_{t+1}\Big(s_{t+1} \Big| s_t, \delta_t(s_t)\Big) \tag{I.47}$$

denoted by $\pi\Big(\mathcal{T} \Big| s_0, (\delta)\Big)$.

The Sugeno integral of the preference of the trajectory with respect to this distribution is denoted by

$$\mathbb{S}_\Pi\left[\rho\Big(\mathcal{T}, (\delta)\Big) \ \middle|\ S_0 = s_0, (\delta)\right] = \mathbb{S}_\Pi\left[\min\left\{ \min_{t=0}^{H-1} \rho\Big(S_t, \delta_t(S_t)\Big), \Psi(S_H) \right\} \ \middle|\ S_0 = s_0, (\delta)\right]$$

and defines the optimistic criterion defining optimal strategies, *i.e.* an optimistic value function:

$$\overline{U_H}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big)=\max_{\mathcal{T}\in\mathcal{T}_H}\min\Big\{\rho\Big(\mathcal{T},(\delta)\Big),\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\Big\}. \tag{I.48}$$

It is equivalent to the optimistic qualitative possibilistic criterion (I.44), however, the expectation is over trajectories $\mathcal{T}_H$, and the preference depends on the strategy. The *optimal optimistic strategy* $\overline{\delta^*}$ is the strategy maximizing the optimistic value function (I.48), and the *optimal optimistic value function* is the maximal optimistic value function among strategies $(\delta)\in\Delta_H$:

Definition I.2.13 (*Optimal Optimistic Value Function and Strategy*)

$\forall s\in\mathcal{S}$,

$$\overline{U_H^*}(s)=\max_{(\delta)\in\Delta_H}\Big\{\overline{U_H}\Big(s,(\delta)\Big)\Big\} \qquad \textit{(optimal optimistic value function)}, \tag{I.49}$$

$$\overline{\delta^*}(s)\in\operatorname*{argmax}_{(\delta)\in\Delta_H}\Big\{\overline{U_H}\Big(s,(\delta)\Big)\Big\} \qquad \textit{(optimal optimistic strategy)}. \tag{I.50}$$

As well, the pessimistic qualitative possibilistic criterion (I.45) leads to a cautious criterion for strategies: the pessimistic value function is the Sugeno integral of the preference trajectory with respect to the necessity measure which comes from the possibility distribution over trajectories $\mathcal{T}_H$ (I.47) with the strategy $(\delta)\in\Delta_H$:

$$\underline{U_H}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big)=\min_{\mathcal{T}\in\mathcal{T}_H}\max\Big\{\rho\Big(\mathcal{T},(\delta)\Big),1-\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\Big\}. \tag{I.51}$$

denoted by $\mathbb{S}_{\mathcal{N}}\Big[\rho\Big(\mathcal{T},(\delta)\Big)\ \Big|\ S_0=s,(\delta)\Big]$. As previously for the optimistic case, the *optimal cautious strategy* $\underline{\delta^*}$ is the strategy maximizing the pessimistic value function (I.51), and the *optimal pessimistic value function* is the maximal pessimistic value function among strategies $(\delta)\in\Delta_H$:

Definition I.2.14 (*Optimal Pessimistic Value Function and Strategy*)

$\forall s\in\mathcal{S}$,

$$\underline{U_H^*}(s)=\max_{(\delta)\in\Delta_H}\Big\{\underline{U_H}\Big(s,(\delta)\Big)\Big\} \qquad \textit{(optimal pessimistic value function)}, \tag{I.52}$$

$$\underline{\delta^*}(s)\in\operatorname*{argmax}_{(\delta)\in\Delta_H}\Big\{\underline{U_H}\Big(s,(\delta)\Big)\Big\} \qquad \textit{(optimal pessimistic strategy)}. \tag{I.53}$$

As for the probabilistic MDPs (see Section I.1.3 Theorem 1), optimal value functions and strategies can be computed with Dynamic Programming:

Theorem 13 (*Dynamic Programming for π-MDPs*)

The optimal optimistic criterion and an associated optimal strategy can be computed as follows: $\forall s\in\mathcal{S}$,

$$\overline{U_0^*}(s)=\Psi(s), \qquad \textit{and, } \forall 1\leqslant i\leqslant H,$$

$$\overline{U_i^*}(s)=\max_{a\in\mathcal{A}}\min\Big\{\rho_{H-i}(s,a),\max_{s'\in\mathcal{S}}\min\Big\{\pi_{H-i}\left(s'\mid s,a\right),\overline{U_{i-1}^*}(s')\Big\}\Big\}. \tag{I.54}$$

$$\overline{\delta^*_{H-i}}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min\left\{\pi_{H-i}\left(s' \mid s,a\right), \overline{U^*_{i-1}}(s')\right\}\right\}. \quad \text{(I.55)}$$

As well, the optimal pessimistic criterion and an associated optimal strategy can be computed as follows: $\forall s \in \mathcal{S}$,

$$\underline{U^*_0}(s) = \Psi(s), \qquad \text{and, } \forall 1 \leqslant i \leqslant H,$$
$$\underline{U^*_i}(s) = \max_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max\left\{1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U^*_{i-1}}(s')\right\}\right\}. \quad \text{(I.56)}$$

$$\underline{\delta^*_{H-i}}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max\left\{1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U^*_{i-1}}(s')\right\}\right\}. \quad \text{(I.57)}$$

The proof is given in Annex A.20.

In this theorem, the horizon i is the opposite modulo H of the stage of the process t: during execution, $\delta_t = \delta_{H-i}$ is used at time step t, *i.e.* when it remains i steps. These Dynamic Programming formulae lead to the optimistic algorithm, Algorithm 4 and the pessimistic one Algorithm 5, the qualitative possibilistic counterpart of Algorithm 1 for finite horizon probabilistic MDPs.

Algorithm 4: Dynamic Programming Algorithm for Optimistic π-MDP

$\overline{U^*_0} \leftarrow \Psi$;
for $i \in \{1,\ldots,H\}$ **do**
  for $s \in \mathcal{S}$ **do**
    $\overline{U^*_i}(s) \leftarrow \max_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min\left\{\pi_{H-i}\left(s' \mid s,a\right), \overline{U^*_{i-1}}(s')\right\}\right\}$;
    $\overline{\delta^*_{H-i}}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min\left\{\pi_{H-i}\left(s' \mid s,a\right), \overline{U^*_{i-1}}(s')\right\}\right\}$;
return $\overline{U^*_H}$, $\overline{\delta^*}$;

Algorithm 5: Dynamic Programming Algorithm for Pessimistic π-MDP

$\underline{U^*_0} \leftarrow \Psi$;
for $i \in \{1,\ldots,H\}$ **do**
  for $s \in \mathcal{S}$ **do**
    $\underline{U^*_i}(s) \leftarrow \max_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max\left\{1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U^*_{i-1}}(s')\right\}\right\}$;
    $\underline{\delta^*_{H-i}}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min\left\{\rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max\left\{1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U^*_{i-1}}(s')\right\}\right\}$;
return $\underline{U^*_H}$, $\underline{\delta^*}$;

Note that a broader class of MDP models, including both probabilistic and qualitative possibilistic MDPs presented above, is called *Algebraic MDPs* [106].

The next section is devoted to the presentation of the qualitative possibilistic counterpart of the POMDPs denoted by π-POMDPs: the π-POMDP model is the partially observable version of the π-MDP one. This model has been presented first in [121] in pessimistic settings. The algorithm to solve it has been also presented in case no intermediate preference degree

is involved *i.e.* in case where preference functions ρ_t are not used: in these settings only the terminal preference function Ψ models the goal of the mission: an optimistic strategy maximizes the plausibility of strategies which end with a good preference, and a cautious strategy minimizes the plausibility of strategies ending in unwanted states. As the preference of a system state trajectory $\mathcal{T} = (s_1, \ldots, s_H)$ is simply $\rho(\mathcal{T}) = \Psi(s_H)$, while the preference of a π-MDP trajectory is $\left\{ \min_{t=0}^{H-1} \rho(s_t, a_t), \Psi(s_H) \right\}$, it it sufficient to consider a classical π-MDP such that $\rho_t(s, a) = 1$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$ and $\forall t \in \{0, \ldots, H-1\}$. The π-MDP criteria are simplified as follows:

Definition I.2.15 (*Criteria for π-MDP with Terminal Preference Only*)

The optimistic (resp. pessimistic) criterion is the Sugeno integral of the last state preference $\Psi(S_H)$ with respect to the possibility measure (resp. necessity measure) associated to the possibility distribution

$$\Pi\left(S_H = s_H \mid S_0 = s_0, (\delta_t)_{t=0}^{H-1}\right) = \max_{(s_1,\ldots,s_{H-1}) \in \mathcal{S}^{H-1}} \min_{t=0}^{H-1} \pi\Big(s_{t+1} \Big| s_t, \delta_t(s_t)\Big)$$

denoted by $\pi\Big(s_H \Big| s_0, (\delta)\Big)$: these integrals may be denoted by $\mathbb{S}_\Pi\left[\Psi\Big(S_H\Big) \;\Big|\; S_0 = s_0, (\delta)\right]$ (resp. $\mathbb{S}_\mathcal{N}\left[\Psi\Big(S_H\Big) \;\Big|\; S_0 = s_0, (\delta)\right]$).

Optimistic criterion:

$$\overline{U_H}\Big(s_0, (\delta_t)_{t=0}^{H-1}\Big) = \max_{s_H \in \mathcal{S}} \min\left\{\Psi(s_H), \pi\Big(s_H \Big| s_0, (\delta)\Big)\right\}. \tag{I.58}$$

Pessimistic criterion:

$$\underline{U_H}\Big(s_0, (\delta_t)_{t=0}^{H-1}\Big) = \min_{s_H \in \mathcal{S}} \max\left\{\Psi(s_H), 1 - \pi\Big(s_H \Big| s_0, (\delta)\Big)\right\}. \tag{I.59}$$

In this case the π-MDP focuses on the preference over terminal states, whatever the intermediate ones.

I.2.5 π-POMDPs

The qualitative possibilistic POMDP (π-POMDP) model has been first presented in [121]. As explained in Section I.1.6 which presents the classical probabilistic POMDP model, in partially observable settings the system state is not given anymore as input to the agent: the agent has to infer it using the observations $o \in \mathcal{O}$ received at each time step, represented by the observation process $(O_t)_{t\in\mathbb{N}}$. The uncertainty about successive observation variables O_t only depends on the current action and the reached state: if the agent selected action $a \in \mathcal{A}$ at time step t, and the system has reached state $s' \in \mathcal{S}$ at time step $t+1$, the observation $o' \in \mathcal{O}$ is received with possibility degree $\pi_t(o' \mid s', a) = \Pi(O_{t+1} = o' \mid S_{t+1} = s', a)$: conditional to the next system state s' and the current action a, the next observation variable is M-independent (see Definition I.2.9) from all other variables. Figure I.3 of Section I.1.6 illustrates just as well the dynamic and the structure of a π-POMDP: however, the rewards r and R have to be replaced by preferences ρ and Ψ, and transition (resp. observation) probability distributions $\mathbf{p}$ have to be replaced by the possibility distribution $\pi_t(s' \mid s, a)$ (resp. $\pi_t(o' \mid s', a)$).

As with the probabilistic model, the computation of strategies is performed by translating of the π-POMDP into a fully observable π-MDP. The state space of the later is the set of possible *qualitative possibilistic belief states* $\beta : \mathcal{S} \to \mathcal{L}$ describing the knowledge about the actual system state, *i.e.* the set of all the possibility distributions over $\mathcal{S}$. This set is denoted

by $\Pi_{\mathcal{L}}^{\mathcal{S}} = \{\, \pi : \mathcal{S} \to \mathcal{L} \mid \max_{s \in \mathcal{S}} \pi(s) = 1 \,\}$. Note first that the number of possible possibilistic beliefs about the actual system state is

$$(\#\mathcal{L})^{\#\mathcal{S}} - (\#\mathcal{L} - 1)^{\#\mathcal{S}}\,. \tag{I.60}$$

Indeed, there are $\#\mathcal{L}^{\#\mathcal{S}}$ different functions from $\mathcal{S}$ to $\mathcal{L}$, and $(\#\mathcal{L} - 1)^{\#\mathcal{S}}$ non-normalized ones *i.e.* functions $f : \mathcal{S} \to \mathcal{L}$ such that $\max_{s \in \mathcal{S}} f(s) < 1$. The number of possibility distributions over $\mathcal{S}$ is the number of normalized functions from $\mathcal{S}$ to $\mathcal{L}$, that is the total number of functions minus the number of non-normalized ones.

First of all, let us formally defined a π-POMDP as the 7-uple $< \mathcal{S}, \mathcal{A}, \mathcal{O}, T^{\pi}, O^{\pi}, \Psi, \beta_0 >$:

- $\mathcal{S}$, a finite set of hidden system states;
- $\mathcal{A}$ a finite set of actions;
- $\mathcal{O}$ a finite set of observations;
- T^{π} the set of transition possibility distributions containing for each time step $t \in \mathbb{N}$, each current system state $s \in \mathcal{S}$ and each current action $a \in \mathcal{A}$, the possibility distributions over the next system state $s' \in \mathcal{S}$, $\pi_t\,(\,s' \mid\, s, a\,)$;
- O^{π}, the set of observation possibility distributions: for each time step $t \in \mathbb{N}$, each current action $a \in \mathcal{A}$, each next state s', the possibility distribution over the next observation $o' \in \mathcal{O}$, $\pi_t\,(\,o' \mid\, s', a\,)$ is part of O^{π}.
- Ψ the preference function, defining for each state $s \in \mathcal{S}$, the preference assigned to the situations where the system terminates in state s.
- β_0, the possibilistic *initial belief state*, is the possibility distribution defining the uncertainty about the initial state: $\forall s \in \mathcal{S}$, $\beta_0(s) = \Pi(S_0 = s)$.

At each time step the current qualitative possibilistic belief state is computed from these objects: the possibilistic counterpart of the probabilistic belief defined in Definition I.1.3 of Section I.1.6. The initial belief state $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ is part of the definition of a π-POMDP. At a time step $t \geqslant 1$, the belief state is the possibility distribution over the current system state, conditional to all the data available to the agent.

Definition I.2.16 (*Qualitative Possibilistic Belief state*)

$$\beta_t(s) = \Pi\,(\,S_t = s \mid\, O_1 = o_1, \ldots, O_t = o_t, a_0, \ldots, a_{t-1}\,) = \Pi\,(\,S_t = s \mid\, I_t = i_t\,) \tag{I.61}$$

where $i_t = \{\, o_1, \ldots, o_t, a_0, \ldots, a_{t-1} \,\}$ is the information available to the agent at time t ($i_0 = \{\,\} = \emptyset$), and I_t the variable version (as in the probabilistic POMDP presentation).

The possibilistic belief updating process consists in the sequence of belief states, which can be computed recursively:

Theorem 14 (*Qualitative Possibilistic Belief Update*)

If the belief state at time step t is β_t, the selected action is $a_t \in \mathcal{A}$, and the next observation is o_{t+1}, the next belief state β_{t+1} is computed as follows:

$$\beta_{t+1}(s') = \begin{cases} 1 & \text{if } \pi_t\,(\,s', o_{t+1} \mid\, \beta_t, a_t\,) = \pi_t\,(\,o_{t+1} \mid\, \beta_t, a_t\,)\,, \\ \pi_t\,(\,s', o_{t+1} \mid\, \beta_t, a_t\,) & \text{otherwise.} \end{cases} \tag{I.62}$$

where the joint distribution over system state variable S_{t+1} and observation variable O_{t+1} conditional on the current information, is denoted by $\pi_t\,(\,s', o' \mid\, \beta_t, a_t\,) =$

$\min\left\{\pi_t\left(o' \mid s', a_t\right), \max_{s\in\mathcal{S}} \min\left\{\pi_t\left(s' \mid s, a_t\right), \beta_t(s)\right\}\right\}$. *The notation* $\pi\left(o' \mid \beta_t, a_t\right)$ *is also used for* $\max_{s'\in\mathcal{S}} \pi_t\left(s', o' \mid \beta_t, a_t\right)$.

This formula is called the ***possibilistic belief update****, and since the belief state* β_{t+1} *is shown to be a function of* β_t, a_t *and* o_{t+1}, *we denote it by*

$$\beta_{t+1} = \nu(\beta_t, a_t, o_{t+1}),$$

with ν *called the belief update function.*

The proof is given in Annex A.21.

The possibilistic belief update (I.62) is denoted by

$$\beta_{t+1}(s') \propto^{\pi} \pi_t\left(s', o_{t+1} \mid \beta_t, a_t\right)$$

as it only consists in normalizing the function $s' \mapsto \pi\left(s', o_{t+1} \mid \beta_t, a_t\right)$ in a possibilistic sense ($\max_s \pi(s) = 1$).

We denote by B_t^{π} the belief state when considered as a variable, *i.e.* B_0^{π} is deterministic equal to β_0 (but S_0 is uncertain with possibility distribution β_0) and $B_{t+1}^{\pi} = \nu(B_t^{\pi}, a_t, O_{t+1})$ where O_{t+1} is the observation variable at time step $t+1$.

To make things clear, the π-POMDP model is defined here only with a terminal preference function Ψ, and no intermediate ones ρ_t. The next chapter will address formally the model with intermediate preference degree: the pessimistic one presented in [121] and an optimistic one. The criteria, or optimistic and pessimistic value functions of the π-POMDP model with terminal preference only, are thus similar to criteria (I.58) and (I.59). Note that the optimistic criterion has not been presented yet to the best of our knowledge, and is proposed now in parallel with the pessimistic one [121].

Definition I.2.17 (*π-POMDP Criteria with Terminal Preference Only*)

This is the same criteria as in the fully observable case (terminal preference case, Definition I.2.15): however, these criteria depends here on the initial belief state.

The optimistic π-POMDP criterion, or ***optimistic value function****, is the Sugeno integral of the terminal preference with respect to the possibility measure of the system process for a given strategy* $(\delta_t)_{t=0}^{H-1}$*: the strategy which is looked for is a sequence of function of the available information* i_t, *i.e.* $(\delta) = (\delta_t)_{t=0}^{H-1}$ *with* $\delta_t : i_t \mapsto \delta(i_t) \in \mathcal{A}$.

$$\overline{U_H}\Big(\beta_0, (\delta)_{t=0}^{H-1}\Big) = \max_{s_H\in\mathcal{S}} \min\Big\{\Psi(s_H), \pi\Big(s_H\Big|\beta_0, (\delta)\Big)\Big\}. \tag{I.63}$$

As well, the π-POMDP ***pessimistic value function*** *is the Sugeno integral of the terminal preference with respect to the necessity measure of the system process given such a strategy:*

$$\underline{U_H}\Big(\beta_0, (\delta)_{t=0}^{H-1}\Big) = \min_{s_H\in\mathcal{S}} \max\Big\{\Psi(s_H), 1 - \pi\Big(s_H\Big|\beta_0, (\delta)\Big)\Big\}. \tag{I.64}$$

where

$$\begin{aligned}\pi\Big(s_H\Big|\beta_0, (\delta)\Big) &= \Pi\Big(S_H = s_H\Big|(\delta)\Big)\\ &= \max_{(s_0,\ldots,s_{H-1})\in\mathcal{S}^H} \min\Big\{\min_{t=0}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t, \delta(i_t)\Big), \beta_0(s_0)\Big\}\end{aligned}$$

is the possibility distribution over the last system state given the strategy. Thus the optimistic criterion may be denoted by $\mathbb{S}_{\Pi}\Big[\Psi(S_H)\Big|\beta_0, (\delta)\Big]$ *and the pessimistic one*

$\mathbb{S}_{\mathcal{N}}\Big[\Psi(S_H)\Big|\beta_0,(\delta)\Big]$, *as they are optimistic and pessimistic Sugeno integrals based on the distribution* $\pi\Big(s_H\Big|\beta_0,(\delta)\Big)$.

As in the probabilistic framework, Section I.1.8, these criteria can be rewritten based on a belief-dependent preference. Consider as previously, a strategy $(\delta_t)_{t=0}^{H-1}$ based on the current information $i_0=\emptyset$, $i_1=\{a_0,o_1\}$, $i_2=\{a_0,o_1,a_1,o_2\}$, *etc*: for each time step $t\geqslant 0$, $\delta_t : i_t \mapsto \delta_t(i_t)\in\mathcal{A}$. Recall that ν is the belief update function defined in Theorem 14. We denote by $\widehat{O}_t=\Big(O_i\Big)_{i=1}^{t}$ the successive observations until time step t seen as variables, and I_t the associated information. The belief state at time step $t+1$, seen as a variable, can be written $B_{t+1}^{\pi}=\nu^{\delta_t,\widehat{O}_{t+1}}(B_t^{\pi})$, $\forall t\geqslant 0$, with $(\delta_t)_{t=0}^{H-1}$ such a strategy, and the notation $\nu^{\delta_t,\widehat{O}_{t+1}}:\beta\mapsto\nu\Big(\beta,\delta_t(I_t),O_{t+1}\Big)$. Thus,

$$B_H^{\pi}=\left(\bigcirc_{t=0}^{H-1}\nu^{\delta_t,\widehat{O}_{t+1}}\right)(B_0^{\pi}),$$

where $\bigcirc$ is the function composition operator. Then, knowing that $\widehat{O}_H=\widehat{o}_H$ with $\widehat{o}_H=\{o_1,\ldots,o_H\}\in\mathcal{O}^H$ a sequence of observations, B_H^{π} is known: it is denoted by $\beta_{\beta_0}^{\delta,\widehat{o}_H}$, and is called the belief state generated by the observation sequence $\widehat{o}_H$ and the strategy $(\delta_t)_{t=0}^{H-1}$.

Theorem 15 (π-*POMDP Criteria Rewritings – Terminal Preference Case*)

Let $\widehat{o}_H=\{o_1,\ldots,o_H\}$ a sequence of observations, and $(\delta)=(\delta)_{t=0}^{H-1}$ be a strategy such that δ_{t+1} is a function of the information $i_{t+1}=\{\delta_t(i_t),o_{t+1}\}$. The possibility distribution over the possible sequences of observations is denoted by

$$\pi\Big(\widehat{o}_H\Big|(\delta),\beta_0\Big)=\Pi\left(\widehat{O}_H=\widehat{o}_H\;\middle|\;(\delta),\beta_0\right)$$

$$=\max_{(s_0,\ldots,s_H)\in\mathcal{S}^{H+1}}\min\left\{\pi_t\Big(o_{t+1}\Big|s_{t+1},\delta_t(i_t)\Big),\pi_t\Big(s_{t+1}\Big|s_t,\delta_t(i_t)\Big),\beta_0(s_0)\right\}.$$

The optimistic π-POMDP criterion is equal to the Sugeno integral of the belief-based optimistic preference $\overline{\Psi}(B_H^{\pi})=\max_{s\in\mathcal{S}}\min\{\Psi(s),B_H^{\pi}(s)\}$ with respect to the possibility measure over the observation sequences. That is, denoting by $\beta_{\beta_0}^{\delta,\widehat{o}_H}$ the belief state generated by the observation sequence $\widehat{o}_H$ and the strategy $(\delta_t)_{t=0}^{H-1}$, the ***optimistic criterion can be rewritten as***

$$\begin{aligned}\overline{U_H}\Big(\beta_0,(\delta)_{t=0}^{H-1}\Big)&=\max_{\widehat{o}_H}\min\left\{\overline{\Psi}(\beta_{\beta_0}^{\delta,\widehat{o}_H}),\pi\Big(\widehat{o}_H\Big|(\delta),\beta_0\Big)\right\}\\&=\max_{\widehat{o}_H}\min\left\{\max_{s\in\mathcal{S}}\min\left\{\Psi(s),\beta_{\beta_0}^{\delta,\widehat{o}_H}(s)\right\},\pi\Big(\widehat{o}_H\Big|(\delta),\beta_0\Big)\right\}.\end{aligned}\tag{I.65}$$

Likewise, the pessimistic criterion is equal to the Sugeno integral of the belief-based pessimistic preference $\underline{\Psi}(B_H^{\pi})=\min_{s\in\mathcal{S}}\max\{\Psi(s),1-B_H^{\pi}(s)\}$, with respect to the necessity measure over the observation sequences. The π-POMDP ***pessimistic criterion can be rewritten as***

$$\begin{aligned}\underline{U_H}\Big(\beta_0,(\delta)_{t=0}^{H-1}\Big)&=\min_{\widehat{o}_H}\max\left\{\underline{\Psi}(\beta_{\beta_0}^{\delta,\widehat{o}_H}),1-\pi\Big(\widehat{o}_H\Big|(\delta),\beta_0\Big)\right\}\\&=\min_{\widehat{o}_H}\max\left\{\min_{s\in\mathcal{S}}\max\left\{\Psi(s),1-\beta_{\beta_0}^{\delta,\widehat{o}_H}(s)\right\},1-\pi\Big(\widehat{o}_H\Big|(\delta),\beta_0\Big)\right\}.\end{aligned}\tag{I.66}$$

The proof is given in Annex A.22.

Written as a Sugeno integral, the optimistic criterion $\mathbb{S}_\Pi\Big[\Psi(S_H)\Big|\beta_0,(\delta)\Big]$, which is based, as in Definition I.2.17, on $\pi\Big(s_H\Big|\beta_0,(\delta)\Big)$ (the possibility distribution over the last system state s_H), becomes in the previous theorem

$$\mathbb{S}_\Pi\Big[\max_{s\in\mathcal{S}}\min\{\,\Psi(s),B_H^\pi(s)\,\}\,\Big|\beta_0,(\delta)\Big]=\mathbb{S}_\Pi\Big[\overline{\Psi}(B_H^\pi)\Big|\beta_0,(\delta)\Big].$$

These two equal Sugeno integrals are based on the possibility distribution over the observation sequence $\pi\Big(\widehat{o}_H\Big|(\delta),\beta_0\Big)$. As well, in Theorem 15, the pessimistic criterion $\mathbb{S}_\mathcal{N}\Big[\Psi(S_H)\Big|\beta_0,(\delta)\Big]$ is rewritten

$$\mathbb{S}_\mathcal{N}\Big[\min_{s\in\mathcal{S}}\max\{\,\Psi(s),1-B_H^\pi(s)\,\}\,\Big|\beta_0,(\delta)\Big]=\mathbb{S}_\mathcal{N}\Big[\underline{\Psi}(B_H^\pi)\Big|\beta_0,(\delta)\Big].$$

Note that the belief-based preferences $\overline{\Psi}$ and $\underline{\Psi}$, are the π-POMDP counterparts of the POMDP belief-based reward $r(b,a)=\sum_{s\in\mathcal{S}}r(s,a)\cdot b_t(s)$. As well, these rewritings are the possibilistic counterpart of the rewriting $\mathbb{E}\Big[r\Big(S_t,d_t(i_t)\Big)\Big]=\mathbb{E}\Big[\sum_{s\in\mathcal{S}}B_t(s)\cdot r\Big(s,d_t(i_t)\Big)\Big]$.

This theorem assures us that the criteria can be expressed as Sugeno integrals of a function of the possibilistic belief state B_H^π: this result leads to the definition of π-MDPs whose states are the qualitative possibilistic belief states: these π-MDPs are denoted by $\langle\tilde{\mathcal{S}}^\pi,\mathcal{A},\tilde{T}^\pi,\tilde{\Psi}\rangle$. The state space $\tilde{\mathcal{S}}^\pi$ is the finite set of possibilistic belief states $\Pi_\mathcal{L}^\mathcal{S}=\{\,\beta\mid\ \beta:\mathcal{S}\to\mathcal{L},\max_{s\in\mathcal{S}}\beta(s)=1\,\}$.

Let β_t a given qualitative possibilistic belief, *i.e.* a possibility distribution in $\Pi_\mathcal{L}^\mathcal{S}$. The sequence of variables $(B_t^\pi)_{t\in\mathbb{N}}$ is the sequence of the belief functions seen as random variables. As highlighted by the possibilistic belief update (I.62), if $B_t^\pi=\beta_t$, and the selected action is a_t, the value of the next variable B_{t+1}^π is a deterministic function of the observation O_{t+1}.

A belief π-MDP is defined since the qualitative possibilistic belief updating process is shown to be a possibilistic Markov process *i.e.* $\forall a\in\mathcal{A},\forall\beta'\in\Pi_\mathcal{L}^\mathcal{S}$, B_{t+1}^π is M-independent from all previous variables conditional to the current belief B_t^π and the selected action $a_t\in\mathcal{A}$:

Theorem 16

The qualitative possibilistic belief updating process is a Markov process, i.e.

$$\Pi\left(B_{t+1}^\pi=\beta'\mid\ I_t=i_t,a_t\right)=\Pi\left(B_{t+1}^\pi=\beta'\mid\ B_t^\pi=\beta_{b_0}^{i_t},a_t\right),\tag{I.67}$$

where $\beta_{b_0}^{i_t}$ is the qualitative belief state reached starting with β_0 and with the information $i_t=\{\,a_0,o_1,a_1,o_2,\ldots,a_{t-1},o_t\,\}$.

The proof is given in Annex A.23.

As highlighted by the equation (44) in the proof, if $B_t^\pi=\beta$ and the selected action is $a\in\mathcal{A}$, the possibility degree that the next belief B_{t+1}^π is $\beta'\in\Pi_\mathcal{L}^\mathcal{S}$, is the maximum of all the possibility degrees of observations o' such that $\nu\left(\beta,a,o'\right)=\beta'$: it defines the transition possibility distributions of the belief process, *i.e.* elements of $\tilde{T}$, as follows: $\forall t\geqslant 0$,

$$\pi_t\left(\beta'\mid\ \beta,a\right)=\max_{\substack{o'\in\mathcal{O}\ \text{s.t.}\\ \nu(\beta,a,o')=\beta'}}\pi_t\left(o'\mid\ \beta,a\right),\tag{I.68}$$

where $\pi_t\left(o'\mid\ \beta,a\right)=\max_{(s,s')\in\mathcal{S}^2}\min\Big\{\pi_t\left(o'\mid\ s',a_t\right),\pi_t\left(s'\mid\ s,a_t\right),\beta(s)\Big\}$, is the possibility degree of observing o' conditional on all the previous information.

Finally, the preference functions associated with the possibilistic belief β_H are defined as highlighted by Theorem I.65: for an optimistic π-POMDP, the preference function is, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$,

$$\overline{\Psi}(\beta) = \max_{s \in \mathcal{S}} \min \{ \Psi(s), \beta(s) \} \tag{I.69}$$

and for a pessimistic one, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$,

$$\underline{\Psi}(\beta) = \min_{s \in \mathcal{S}} \max \{ \Psi(s), 1 - \beta(s) \}. \tag{I.70}$$

As for each belief $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, the possibility and necessity measure conditional on the information i don't vary if $i \in \left\{ i \mid \beta_{\beta_0}^i = \beta \right\}$, *i.e.* if the information i leads to β (see for instance the proof of Theorem 16), it is sufficient to look for a belief-based strategy $(\delta_t)_{t=0}^{H-1}$, such that $\forall t \geqslant 0$, $\delta_t : \beta_t \mapsto \delta_t(\beta_t) \in \mathcal{A}$.

The π-MDP $\langle \tilde{\mathcal{S}^\pi}, \mathcal{A}, \tilde{T^\pi}, \overline{\Psi} \text{ or } \underline{\Psi} \rangle$ built from of a π-POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T^\pi, O^\pi, \beta_0 \rangle$ is finally:

- $\tilde{\mathcal{S}^\pi} = \Pi_{\mathcal{L}}^{\mathcal{S}}$, the set of all qualitative possibilistic beliefs;
- $\tilde{T^\pi}$ contains all transition possibility distributions of the possibilistic beliefs: $\forall a \in \mathcal{A}$, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, the belief transition possibility distribution defined by the equation (I.68), $\pi_t(. \mid \beta, a)$ is in $\tilde{T^\pi}$;
- preference functions $\overline{\Psi}$ if the computed criterion is optimistic, see equation (I.69), or $\underline{\Psi}$ if it is pessimistic, equation (I.70).

Note now that, using the belief state transition definition (I.68), and the equation (I.30) of Property I.2.1, for each function from the belief space to $\mathcal{L}$, $U : \Pi_{\mathcal{L}}^{\mathcal{S}} \to \mathcal{L}$,

$$\begin{aligned}
\max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min \{ \pi_t(\beta' \mid \beta, a), U(\beta') \} &= \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min \left\{ \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o') = \beta'}} \pi_t(o' \mid \beta, a), U(\beta') \right\} \\
&= \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o') = \beta'}} \min \{ \pi_t(o' \mid \beta, a), U(\beta') \} \\
&= \max_{o' \in \mathcal{O}} \min \left\{ \pi_t(o' \mid \beta, a), U\Big(\nu(\beta, a, o')\Big) \right\},
\end{aligned}$$

This observation leads to Algorithm 6 which is the π-MDP algorithm (4) with terminal preference criteria (I.58), applied to the π-MDP $\langle \tilde{\mathcal{S}^\pi}, \mathcal{A}, \tilde{T^\pi}, \overline{\Psi} \rangle$.

Algorithm 6: Dynamic Programming Algorithm for Optimistic π-POMDP with Terminal Preference Only

$\overline{U_0^*} \leftarrow \overline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\overline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ \pi_t(o' \mid \beta, a), \overline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\}$;
    $\overline{\delta_{H-i}}(\beta) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ \pi_t(o' \mid \beta, a), \overline{U_{i-1}^*}(\nu(\beta, a, o')) \right\}$;
return $\overline{U_H^*}$, $(\overline{\delta^*})$;

As well, the equations (I.26) and (I.27) of Property I.2.1 leads to

$$\min_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \max \{ 1 - \pi_t(\beta' \mid \beta, a), U(\beta') \} = 1 - \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min \{ \pi_t(\beta' \mid \beta, a), 1 - U(\beta') \},$$

for each function $U : \Pi_{\mathcal{L}}^{\mathcal{S}} \to \mathcal{L}$. Thus, using the belief state transition definition (I.68),

$$\min_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \max \left\{ 1 - \pi_t \left(\beta' \mid \beta, a \right), U(\beta') \right\} = \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t \left(o' \mid \beta, a \right), U\Big(\nu(\beta, a, o') \Big) \right\}.$$

It leads to Algorithm 7, which is the π-MDP algorithm (5), with terminal preference criteria (I.59), applied to the π-MDP $\langle \tilde{\mathcal{S}}^{\pi}, \mathcal{A}, \tilde{T}^{\pi}, \underline{\Psi} \rangle$.

Algorithm 7: Dynamic Programming Algorithm for Pessimistic π-POMDP with Terminal Preference Only

$\underline{U_0^*} \leftarrow \underline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\underline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t \left(o' \mid \beta, a \right), \underline{U_{i-1}^*}\Big(\nu(\beta, a, o') \Big) \right\}$;
    $\underline{\delta_{H-i}}(\beta) \in \operatorname*{argmax}_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t \left(o' \mid \beta, a \right), \underline{U_{i-1}^*} \left(\nu(\beta, a, o') \right) \right\}$;
return $\underline{U_H^*}$, $(\underline{\delta^*})$;

In this first chapter, probabilistic and qualitative possibilistic POMDPs have been built one after the other shedding some light on the similarities between both models: with probabilistic POMDPs, system state dynamics and observation uncertainty are described with probabilities $\mathbf{p}\left(s' \mid s, a \right) \in \mathbb{R}$ and $\mathbf{p}\left(o' \mid s', a \right) \in \mathbb{R}$ while they are defined by possibility distributions $\pi\left(s' \mid s, a \right) \in \mathcal{L} = \left\{ 0, \frac{1}{k}, \ldots, 1 \right\}$ (with $k \geqslant 1$) and $\pi\left(o' \mid s', a \right) \in \mathcal{L}$ in the π-POMDP framework. Moreover, the probabilistic framework measures the benefit from passing through a system state $s \in \mathcal{S}$ and using action $a \in \mathcal{A}$ with the additive reward functions $r(s, a) \in \mathbb{R}$ (and $R(s)$ for the last state in case of finite-horizon problem); the possibilistic framework uses qualitative preferences $\rho(s, a) \in \mathcal{L}$ and $\Psi(s) \in \mathcal{L}$. Thus, the probabilistic criterion (the value function) for a given strategy is the expectation of the rewards written $\mathbb{E}\left[rewards\Big((S_t)_{t \geqslant 0} \Big) \right] \in \mathbb{R}$, and the possibilistic framework has two criteria (value functions) which are Sugeno integrals of preferences written $\mathbb{S}_{\Pi}\left[preferences\Big((S_t)_{t \geqslant 0} \Big) \right] \in \mathcal{L}$ for the optimistic one, and $\mathbb{S}_{\mathcal{N}}\left[preferences\Big((S_t)_{t \geqslant 0} \Big) \right] \in \mathcal{L}$ for the pessimistic one. A POMDP (resp. π-POMDP), is redefined in terms of fully observable MDP (resp. π-MDP) where the system states are the belief states $b_t \in \mathbb{P}_{b_0}^{\mathcal{S}}$ (resp. $\beta_t \in \Pi_{\mathcal{L}}^{\mathcal{S}}$), *i.e.* probability (resp. possibility) distributions over the system states of the initial POMDP: the belief-based reward has to be defined $r(b, a) = \mathbb{E}_{S \sim b}\left[r(S, a) \right] = \sum_{s \in \mathcal{S}} r(s, a) \cdot b(s)$ in the probabilistic case. In the possibilistic case, the belief-based preference can be written $\overline{\rho}(b, a) = \mathbb{S}_{\Pi, S \sim \beta}\left[\rho(S, a) \right] = \max_{s \in \mathcal{S}} \min \left\{ \rho(s, a), \beta(s) \right\}$ for the optimistic criterion, and $\underline{\rho}(b, a) = \mathbb{S}_{\mathcal{N}, S \sim \beta}\left[\rho(S, a) \right] = \min_{s \in \mathcal{S}} \max \left\{ \rho(s, a), 1 - \beta(s) \right\}$ for the pessimistic one.

The next chapter proposes some improvements of the qualitative possibilistic model: first, criteria are discussed, concerning the preference aggregation, and the impact of the choice of the (optimistic or pessimistic) criterion. Next, the Mixed-Observability property is defined: as for the probabilistic model, the complexity of solving π-POMDPs having this property is reduced. Finally the infinite horizon problem is formally defined and the proposed solving algorithm is shown to return an optimal strategy for a given criterion.

II Updates and Practical Study of the Qualitative Possibilistic Partially Observable Markov Decision Processes

The end of the previous chapter presented the π-POMDPs, a qualitative possibilistic counterpart of the classical probabilistic POMDPs. Recall that, in the qualitative possibilistic framework, the set of belief states is finite, $\#\Pi_{\mathcal{L}}^{\mathcal{S}} < +\infty$ (see Equation I.60) while the set of belief states is infinite in the probabilistic framework $\mathbb{P}_{b_0}^{\mathcal{S}}$: for this reason, π-POMDPs can be seen as a simpler model for sequential decision making under uncertainty, than the probabilistic one. This is a good point as solving probabilistic POMDPs is at least a PSPACE problem [103, 92]. Moreover, Qualitative Possibility Theory allows to model total ignorance as noted in Introduction: it is a motivation for the study of this model. The distribution $\forall s \in \mathcal{S}$, $\beta(s) = 1$ means that all states are plausible according to the belief state β, *i.e.* the agent considers that all system states are possible. Finally, possibility distributions only sort events and represent their plausibility only roughly: no quantitative value is assigned to them. If the probability distributions defining the POMDP are not known in practice, as in the robotic vision example given in Introduction, a qualitative description of the problem is more suitable.

The pessimistic version of the π-POMDP model has been previously defined in [121]. This section is then devoted to the update of this promising model: first, optimistic and pessimistic models with intermediate preferences are built and discussed. This discussion leads to other criteria. Next, the *Mixed-Observability* property [100, 3] is defined, describing the systems some state variables of which are fully observable: it is shown that this property, if correctly taken into account, dramatically reduces the complexity of solving π-POMDPs. Finally, a value iteration algorithm for the infinite horizon π-MDPs (Fully and Partially Observable) is next proposed and the optimality of the returned strategy (for a specified criterion) is shown assuming the existence of a "stay" action in some goal states. Experimental work finally illustrates the performance of the strategies computed from different criteria. It is also shown that strategies computed from π-POMDPs can outperform probabilistic POMDP strategies for a target recognition problem where the agent's observations are imprecise.

II.1 Intermediate Preferences in π-POMDPs

In the previous chapter, π-MDPs criteria taking into account intermediate preference degrees have been defined; after that, the particular case of terminal preferences only has been presented. The global preference degree of a trajectory $\mathcal{T} = (s_0, \ldots, s_H) \in \mathcal{S}^{H+1}$ has been defined

there as the minimum of all the preference degrees of the encountered states:

$$\rho\Big(\mathcal{T},(a_t)_{t=0}^{H-1}\Big)=\min\left\{\min_{t=0}^{H-1}\rho_t\Big(s_t,a_t\Big),\Psi(s_H)\right\}. \tag{II.1}$$

Another global preference based on the max operator can be also proposed. Let us explicitly define the global preferences for system states: they are the possibilistic counterparts of the sum $\sum_{t=0}^{H-1} r_t\Big(s_t,a_t\Big)+R(s_H)$ in the probabilistic model, given in the equation (I.2).

Definition II.1.1 (*Global Preferences on System State Trajectory*)

Let $(A_t)_{t=0}^{H-1}$ be a sequence of action variables modeling the successive agent decisions. Global preferences over system state trajectories $(S_t)_{t=0}^{H}$ are denoted as follows:

- *the **maximum-based** one, not very demanding,*

$$\overline{\mathcal{G}}\Big((S_t)_{t=0}^{H},(A_t)_{t=0}^{H-1}\Big)=\max\left\{\max_{t=0}^{H-1}\left\{\rho_t\Big(S_t,A_t\Big)\right\},\Psi(S_H)\right\}, \tag{II.2}$$

- *the classical **minimum-based** one, which has been used until now,*

$$\underline{\mathcal{G}}\Big((S_t)_{t=0}^{H},(A_t)_{t=0}^{H-1}\Big)=\min\left\{\min_{t=0}^{H-1}\left\{\rho_t\Big(S_t,A_t\Big)\right\},\Psi(S_H)\right\}, \tag{II.3}$$

i.e. with previous notations, if the system state trajectory is denoted by $\mathcal{T}=(s_0,\ldots,s_H)\in\mathcal{S}^{H+1}$ and the strategy $(\delta)=(\delta_t)_{t=0}^{H-1}$ with $\forall t\in\{1,\ldots,H-1\}$, $\delta:\mathcal{S}\to\mathcal{A}$,

- $\overline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big)=\overline{\mathcal{G}}\Big((s_t)_{t=0}^{H},\Big(\delta_t(s_t)\Big)_{t=0}^{H-1}\Big)$, *and*
- $\underline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big)=\underline{\mathcal{G}}\Big((s_t)_{t=0}^{H},\Big(\delta_t(s_t)\Big)_{t=0}^{H-1}\Big)=\rho\Big(\mathcal{T},(a)_{t=0}^{H-1}\Big)$, *see the equation (II.1).*

Note that other aggregation methods giving their results in $\mathcal{L}$ are possible: for instance median value or majority value. However those aggregation methods do not have features allowing an easy use of dynamic programming. Note also that a π-MDP with an optimistic criterion (or optimistic value function, see the equation (I.48) of Section I.2.4) and a maximum-based global preference degree $\overline{\mathcal{G}}$ has not been defined yet (all π-MDPs seen previously had a minimum-based global preference $\underline{\mathcal{G}}$), and we propose it now:

Definition II.1.2 (*Optimistic π-MDP Criterion – Maximum-based Global Preference*)

*Let us denote as previously the initial state $s_0\in\mathcal{S}$, an H-length trajectory $\mathcal{T}=(s_t)_{t=1}^{H}$, and $\mathcal{T}_H=\mathcal{S}^H$ the set of such trajectories. The **value function** of the Maximum-based Optimistic π-MDP is*

$$\overline{\overline{U_H}}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big)=\max_{\mathcal{T}\in\mathcal{T}_H}\min\left\{\overline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big),\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\right\},$$

where, $\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)$ is the possibility degree of the trajectory $\mathcal{T}$ given the strategy $(\delta)=(\delta_t)_{t=0}^{H-1}$, defined as in Section I.2.4, see the equation (I.47). As the Sugeno integral of the global reward with respect to the possibility measure, it can then be denoted by

$$\overline{\overline{U}}_H\Big(s,(\delta)\Big)=\mathbb{S}_\Pi\left[\overline{\mathcal{G}}\Big((S_t)_{t=0}^{H},\Big(\delta_t(S_t)\Big)_{t=0}^{H-1}\Big)\;\middle|\;S_0=s,(\delta)\right].$$

This optimistic π-MDP also can be solved using Dynamic Programming (DP), just like the π-MDPs presented in the previous chapter (see Section I.2.4), and as described by the next theorem:

Theorem 17 (*DP for optimistic π-MDPs with Maximum-based Global Preference*)

The optimal optimistic criterion with the maximum-based global preference $\overline{\mathcal{G}}$, denoted by $\overline{\overline{U_H^*}}$, and an associated optimal strategy $(\overline{\overline{\delta^*}})_{t=0}^{H-1}$, can be computed as follows:
$\forall s \in \mathcal{S}$,

$$\begin{aligned}
&\overline{\overline{U_0^*}}(s) = \Psi(s), \qquad \text{and, } \forall 1 \leqslant i \leqslant H, \\
&\overline{\overline{U_i^*}}(s) = \max_{a\in\mathcal{A}} \max \left\{ \rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min \left\{ \pi_{H-i}\left(s' \mid s,a\right), \overline{\overline{U_{i-1}^*}}(s') \right\} \right\}. \qquad \text{(II.4)}
\end{aligned}$$

$$\overline{\overline{\delta_{H-i}^*}}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \max \left\{ \rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min \left\{ \pi_{H-i}\left(s' \mid s,a\right), \overline{\overline{U_{i-1}^*}}(s') \right\} \right\}. \qquad \text{(II.5)}$$

This theorem can be proved in exactly the same way as the proof of Theorem 13 (see Annex A.20), however the equation (I.35) of Property I.2.1 has to be used.

The optimistic and pessimistic π-MDPs with a minimum-based global preference have been presented in the previous chapter. The optimistic π-MDP with a maximum-based global preference has been defined just above. Note that a last π-MDP may be pessimistic with a maximum-based global preference but is not defined here.

Let us recall that, in the partially observable case, the strategy $(\delta_t)_{t=0}^{H-1}$ is *a priori* an *information-based* one *i.e.* it is such that, for the time step $t \in \{0,\ldots,H-1\}$, δ_t maps the current information $i_t = \{a_0,\ldots,a_{t-1},o_1,\ldots,o_t\} \in \mathcal{A}^t \times \mathcal{O}^t$ to an action $a_t \in \mathcal{A}$. As shown below, two π-POMDP criteria with global preferences, *i.e.* with intermediate preferences, allow to translate the partially observable processes into fully observable ones called belief π-MDPs, as in the case of a terminal preference only (see Theorem 15):

- **the optimistic criterion with maximum-based global preference**, partially observable version of the Definition II.1.2,

$$\overline{U_H}\Big(\beta_0,(\delta)\Big) = \max_{\mathcal{T}\in\mathcal{T}_H} \min \left\{ \overline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big), \pi\Big(\mathcal{T}\Big|\beta_0,(\delta)\Big) \right\}. \qquad \text{(II.6)}$$

  In this formula, $\overline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big)$ is the maximum-based global preference, defined by the equation (II.2), where the strategy consists in a function of the current information i_t. The possibility degree $\pi\Big(\mathcal{T}\Big|\beta_0,(\delta)\Big) = \min\Big\{\min_{t=0}^{H-1}\pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big),\beta(s_0)\Big\}$ is the possibility degree of the trajectory $\mathcal{T}$ given the strategy (δ) and the initial belief state $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$. This criterion can thus be denoted by $\mathbb{S}_{\Pi}\left[\overline{\mathcal{G}}\Big((S_t)_{t=0}^H,\Big(\delta_t(I_t)\Big)_{t=0}^{H-1}\Big) \;\Big|\; \beta_0,(\delta)\right]$, where $I_t = \{O_t,\delta_{t-1}(I_{t-1}),I_{t-1}\}$ is the variable representing the current information.

- **the pessimistic criterion with minimum-based global preference**, partially observable version of the equation I.45 of Section I.2.4,

$$\underline{U_H}\Big(\beta_0,(\delta)\Big) = \min_{\mathcal{T}\in\mathcal{T}_H} \max \left\{ \underline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big), 1-\pi\Big(\mathcal{T}\Big|\beta_0,(\delta)\Big) \right\}. \qquad \text{(II.7)}$$

  Here, $\underline{\mathcal{G}}\Big(\mathcal{T},(\delta)\Big)$ is the minimum-based global preference, see the equation (II.3), with an information-based strategy (δ). The possibility degree of the trajectory is denoted by $\pi\Big(\mathcal{T}\Big|\beta_0,(\delta)\Big)$, and criterion may be denoted by $\mathbb{S}_{\mathcal{N}}\left[\underline{\mathcal{G}}\Big((S_t)_{t=0}^H,\Big(\delta_t(I_t)\Big)_{t=0}^{H-1}\Big) \;\Big|\; \beta_0,(\delta)\right]$.

Note that, in Section I.1.8, the probabilistic criterion is rewritten as a sum of rewards defined on the belief states: this is possible because of the linearity of the probabilistic expectation. In order to propose a global preference degree such as preferences (II.2) and (II.3) for the π-POMDPs, some properties of the Sugeno Integral are needed. These properties are the counterparts of the linearity of the probabilistic expectation:

Property II.1.1 (*Maxitivity and Minitivity of the possibilistic Sugeno integrals*)

Let f and g two functions from Ω to $\mathcal{L}$. Then,

$$\mathbb{S}_\Pi\left[\max\left\{f,g\right\}\right] = \max\left\{\mathcal{S}_\Pi\left[f\right],\mathcal{S}_\Pi\left[g\right]\right\}, \tag{II.8}$$

$$\mathbb{S}_\mathcal{N}\left[\min\left\{f,g\right\}\right] = \min\left\{\mathcal{S}_\mathcal{N}\left[f\right],\mathcal{S}_\mathcal{N}\left[g\right]\right\}. \tag{II.9}$$

where the Sugeno integrals $\mathbb{S}_\Pi$ and $\mathbb{S}_\mathcal{N}$ are defined in Section I.2.3 (see Theorem 12).

The proof is given in Annex B.2.

These properties offer rewritings of the Sugeno integrals (with respect to the possibility and necessity measures) of the global system state preferences $\overline{\mathcal{G}}$ and $\underline{\mathcal{G}}$, as Sugeno integrals of the global belief state preference. As the presented π-POMDP value functions, *i.e.* criteria (II.6) and (II.7), are Sugeno integrals of the global system state preference, they can be rewritten in the form of belief-dependent value functions similar to the one of Section I.1.8 for probabilistic POMDPs:

Theorem 18 (*Rewritings of the π-POMDP Value Functions*)

Recall that the sequence of variables representing the successive belief states is denoted by $(B_t^\pi)_{t=0}^{H-1}$, and the sequence of action variables by A_t (it includes the case $A_t = \delta_t(I_t)$). The following equalities are true:

$$\mathbb{S}_\Pi\left[\overline{\mathcal{G}}\Big((S_t)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right] = \mathbb{S}_\Pi\left[\overline{\mathcal{G}}\Big((B_t^\pi)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right], \tag{II.10}$$

$$\mathbb{S}_\mathcal{N}\left[\underline{\mathcal{G}}\Big((S_t)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right] = \mathbb{S}_\mathcal{N}\left[\underline{\mathcal{G}}\Big((B_t^\pi)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right]. \tag{II.11}$$

where the global preference degrees of a belief state trajectory $(B_t^\pi)_{t=0}^H$ are:

$$\overline{\mathcal{G}}\Big((B_t^\pi)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big) = \max\left\{\max_{t=0}^{H-1}\left\{\overline{\rho_t}\Big(B_t^\pi,A_t\Big)\right\},\overline{\Psi}(B_H^\pi)\right\}$$

$$\underline{\mathcal{G}}\Big((B_t^\pi)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big) = \min\left\{\min_{t=0}^{H-1}\left\{\underline{\rho_t}\Big(B_t^\pi,A_t\Big)\right\},\underline{\Psi}(B_H^\pi)\right\}.$$

The global preference degrees of a belief state trajectory $(B_t^\pi)_{t=0}^H$ are defined as functions of the intermediate preference degrees, denoted by $\overline{\rho_t}$ and $\underline{\rho_t}$:

$$\overline{\rho_t}(B_t^\pi,A_t) = \max_{s\in\mathcal{S}}\min\left\{\rho_t(s,A_t),B_t^\pi(s)\right\}$$

$$\underline{\rho_t}(B_t^\pi,A_t) = \min_{s\in\mathcal{S}}\max\left\{\rho_t(s,A_t),1-B_t^\pi(s)\right\}.$$

Finally, the terminal preference degrees are defined as in Theorem 15 in the previous chapter: $\overline{\Psi}(B_H^\pi) = \max_{s\in\mathcal{S}}\min\left\{\Psi(s),B_H^\pi(s)\right\}$, $\underline{\Psi}(B_H^\pi) = \min_{s\in\mathcal{S}}\max\left\{\Psi(s),1-B_H^\pi(s)\right\}$.

The proof is given in Annex B.3 and uses Theorem I.65 and Property II.1.1.

Two equivalent belief π-MDPs can be then defined from the π-POMDP criteria (II.6) and (II.7): as explained in Section I.2.5, their state space is $\tilde{\mathcal{S}}^\pi = \Pi_\mathcal{L}^\mathcal{S}$, *i.e.* the set of all belief states $\left\{\beta \mid \max_{s\in\mathcal{S}}\beta(s)=1\right\}$. The set of the transition possibility distribu-

tion, denoted by $\tilde{T}^{\pi}$, contains $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, $\forall t \in \{0, \ldots, H-1\}$, the possibility distribution $\forall \beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, $\pi_t(\beta' \mid \beta, a) = \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \pi_t(o' \mid \beta, a)$, where $\pi_t(o' \mid \beta, a)$ is a notation for $\max_{(s,s') \in \mathcal{S}^2} \min\left\{\pi_t(o' \mid s', a_t), \pi_t(s' \mid s, a_t), \beta(s)\right\}$, and $\nu : \Pi_{\mathcal{L}}^{\mathcal{S}} \times \mathcal{A} \times \mathcal{O} \to \Pi_{\mathcal{L}}^{\mathcal{S}}$ is the belief update function (see Theorem 14). Finally,

- for the optimistic π-POMDP with maximum-based global preference $\overline{\mathcal{G}}$, the preference function of the resulting π-MDP is $\forall t \in \{1, \ldots, H-1\}$, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, $\forall a \in \mathcal{A}$,

$$\overline{\rho_t}(\beta, a) = \max_{s \in \mathcal{S}} \min\{\rho_t(s, a), \beta(s)\}$$

  and terminal preference function is $\overline{\Psi}(\beta) = \max_{s \in \mathcal{S}} \min\{\Psi(s), \beta(s)\}$. The resulting π-MDP criterion is the one with the maximum-based global preference $\overline{\mathcal{G}}$ (see Definition II.1.2).

- for the pessimistic π-POMDP with minimum-based global preference $\mathcal{G}$, the preference function of the resulting π-MDP is $\forall t \in \{1, \ldots, H-1\}$, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, $\forall a \in \mathcal{A}$,

$$\underline{\rho_t}(\beta, a) = \min_{s \in \mathcal{S}} \max\{\rho_t(s, a), 1 - \beta(s)\}$$

  and terminal preference function is $\underline{\Psi}(\beta) = \min_{s \in \mathcal{S}} \max\{\Psi(s), 1 - \beta(s)\}$. Finally, the resulting π-MDP criterion is the one with the minimum-based global preference $\underline{\mathcal{G}}$ (see the equation (I.51) of Section I.2.4).

Note that line II.8 of Property II.1.1 justifies the rewriting of the optimistic π-POMDP (π-POMDP with criterion II.6) if the global preference degree is maximum-based. However, such a rewriting is impossible with the minimum-based global preference. Indeed, in order to keep a criterion based on the minimum (II.1), the following equality should be true: $\mathbb{S}_{\Pi}[\min\{f, g\}] = \min\{\mathbb{S}_{\Pi}[f], \mathbb{S}_{\Pi}[g]\}$. However, the following counterexample confirms that this equality is not true in general: consider $\Omega = \{\omega_1, \omega_2\}$, $f : \Omega \to \mathcal{L}$ and $g : \Omega \to \mathcal{L}$ such that $f(\omega_1) = 1$, $f(\omega_2) = 0$, and $g = 1 - f$. Consider the total ignorance possibility distribution: $\pi(\omega_1) = \pi(\omega_2) = 1$. As $\min\{f(\omega), g(\omega)\} = 0$, $\forall \omega \in \Omega$,

$$\mathbb{S}_{\Pi}[\min\{f, g\}] = \max_{\omega \in \Omega} \min\{f(\omega), g(\omega), \pi(\omega)\} = 0,$$

whereas $\mathbb{S}_{\Pi}[f] = \max_{\omega \in \Omega} \min\{f(\omega), \pi(\omega)\} = \max\{1, 0\} = 1$ and and $\mathbb{S}_{\Pi}[g] = \max\{0, 1\} = 1$ as well, thus

$$\min\{\mathbb{S}_{\Pi}[f], \mathbb{S}_{\Pi}[g]\} = 1.$$

It can be shown that $\mathbb{S}_{\mathcal{N}}[\max\{f, g\}] = \max\{\mathbb{S}_{\mathcal{N}}[f], \mathbb{S}_{\mathcal{N}}[f]\}$ is not true in general, with the same counterexample.

The rewritings of Theorem 18 lead to the Dynamic Programming (DP) algorithms (8) and (9): Algorithm (8) corresponds to the DP scheme of Theorem 17, and Algorithm 9 is the π-MDP algorithm (5):

- Algorithm 8 computes an optimal strategy for the π-MDP $\langle \tilde{\mathcal{S}}^{\pi}, \mathcal{A}, \tilde{T}^{\pi}, (\overline{\rho_t})_{t=0}^{H-1}, \overline{\Psi} \rangle$ with the optimistic criterion (using the Sugeno integral $\mathbb{S}_{\Pi}$) and a maximum-based global preference ($\overline{\mathcal{G}}$, see Definition II.1.1), also optimal for the π-POMDP $\langle \mathcal{S}, \mathcal{A}, T^{\pi}, (\rho_t)_{t=0}^{H-1}, \Psi \rangle$ with the optimistic criterion (using $\mathbb{S}_{\Pi}$), and maximum-based global preference ($\overline{\mathcal{G}}$).

- Algorithm 9 computes an optimal strategy for the π-MDP $\langle \tilde{\mathcal{S}}^{\pi}, \mathcal{A}, \tilde{T}^{\pi}, (\underline{\rho_t})_{t=0}^{H-1}, \underline{\Psi} \rangle$, with the pessimistic criterion (using the Sugeno integral $\mathbb{S}_{\mathcal{N}}$) and the classical global mininum-based preference ($\underline{\mathcal{G}}$, see the equation (II.1) or Definition II.1.1), also optimal for the π-POMDP $\langle \mathcal{S}, \mathcal{A}, T^{\pi}, (\rho_t)_{t=0}^{H-1}, \Psi \rangle$, with the pessimistic criterion ($\mathbb{S}_{\mathcal{N}}$) and the classical global mininum-based preference ($\underline{\mathcal{G}}$).

Algorithm 8: DP Algorithm for Optimistic π-POMDP with intermediate preferences

$\overline{U_0^*} \leftarrow \overline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\overline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \max \left\{ \overline{\rho_t}(\beta, a), \max_{o' \in \mathcal{O}} \min \left\{ \pi_t\left(o' \mid b_t, a\right), \overline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\} \right\}$;
    $\overline{\delta_{H-i}}(\beta) \in \operatorname{argmax}_{a \in \mathcal{A}} \max \left\{ \overline{\rho_t}(\beta, a), \max_{o' \in \mathcal{O}} \min \left\{ \pi_t\left(o' \mid b_t, a\right), \overline{U_{i-1}^*}\left(\nu(\beta, a, o')\right) \right\} \right\}$;
return $\overline{U_H^*}$, $(\overline{\delta^*})$;

Algorithm 9: DP Algorithm for Pessimistic π-POMDP with intermediate preferences

$\underline{U_0^*} \leftarrow \underline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\underline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \min \left\{ \underline{\rho_t}(\beta, a), \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t\left(o' \mid b_t, a\right), \underline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\} \right\}$;
    $\underline{\delta_{H-i}}(\beta) \in$
    $\operatorname{argmax}_{a \in \mathcal{A}} \min \left\{ \underline{\rho_t}(\beta, a), \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t\left(o' \mid b_t, a\right), \underline{U_{i-1}^*}\left(\nu(\beta, a, o')\right) \right\} \right\}$;
return $\underline{U_H^*}$, $(\underline{\delta^*})$;

II.1.1 Discussion

The maximum-based global preference $\overline{\mathcal{G}}$ (II.2) cares about the fact that at least one encountered state has a high preference. As explained just above, this global preference has been introduced in order to enable both the definition of belief-based preferences ($\overline{\rho}$ and $\overline{\Psi}$ as in the Terminal Preference case Section I.2.5), and a global preference for belief trajectories (the maximum of the belief-based preference degrees too), when the optimistic criterion is used (*i.e.* using $\mathbb{S}_\Pi$). Moreover, defining $\forall t \in \{0, \ldots, H-1\}$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\rho_t(s, a) = 0$, the use of the maximum-based global preference goes back to the case of the π-MDP with terminal preference only, *i.e.* to use the criterion (I.58) or (I.59).

The minimum-based global preference $\underline{\mathcal{G}}$ (II.3) has been used until this chapter and is the one proposed in [123, 122, 121]: using this global preference, a trajectory has a high preference degree if all the states of this trajectory have a high preference degree. It allows also, when the pessimistic criterion is used (*i.e.* using $\mathbb{S}_\mathcal{N}$), the definition of belief-based preferences ($\underline{\rho}$, $\underline{\Psi}$), and a global preference for belief state trajectories (also the minimum of the belief-based preference degrees). As noted before, if all intermediate preferences are set to 1, *i.e.* $\forall t \in \{0, \ldots, H-1\}$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\rho_t(s, a) = 1$, the use of the minimum-based global preference goes back to use a π-MDP with terminal preference only.

These global preferences suffer from the *drowning effect* [53]. Indeed, using the minimum-based global preference, if one of the encountered state has a low preference degree, the trajectory has a low preference degree no matter the other preference states. There is the same issue with the maximum-based global preference: no distinction will be made between a trajectory with only high preferences, and a trajectory with only one state with a high preference. As it will be presented and tested in this thesis, the *lexi* approaches, or other criteria [146] can solve these difficulties.

Finally, it can be noted that, in the optimistic case, a π-POMDP is satisfied by a total

ignorant belief state: for instance, if $\forall s \in \mathcal{S}$, $\beta(s) = 1$, $\overline{\Psi}(\beta) = \max_{s\in\mathcal{S}} \min\{\Psi(s), \beta(s)\} = \max_{s\in\mathcal{S}} \Psi(s)$. Thus, an uninformative belief state leads to the best preference. A criterion mixing a pessimistic belief-based preference $\overline{\Psi}$ and an optimistic one over the belief preferences can be introduced, even if it means that this criterion is not related anymore on any preference over system states:

Definition II.1.3 (*Mixed Optimistic-Pessimistic Criterion, Terminal Preference Case*)

Given a strategy $(\delta) = (\delta_t)_{t=0}^{H-1}$ and an initial belief state $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, the mixed π-POMDP value function can be defined by one of the three following equivalent formulae:

$$\begin{aligned} U\Big(\beta_0, (\delta)\Big) &= \max_{\beta_H \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min\left\{ \min_{s\in\mathcal{S}} \max\{\Psi(s), 1-\beta_H(s)\}, \pi\Big(\beta_H \Big| \beta_0, (\delta)\Big)\right\} \qquad \text{(II.12)}\\ &= \max_{\beta_H \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min\left\{ \underline{\Psi}(S_H), \pi\Big(\beta_H \Big| \beta_0, (\delta)\Big)\right\}\\ &= \mathbb{S}_\Pi\Big[\underline{\Psi}(S_H)\Big|\beta_0, (\delta)\Big]. \end{aligned}$$

In a nutshell, the general form of the Dynamic Programming equation of a π-POMDP is,

$$\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}, \quad \widehat{U}_0^*(\beta) = \tilde{\Psi}(\beta),$$

and, $\forall i \in \{1, \ldots, H\}$, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$,

$$\widehat{U_i^*}(\beta) = \max_{a\in\mathcal{A}} \widehat{\mathrm{M}}\left\{\tilde{\rho}_t(\beta, a), \widehat{\mathbb{S}}\Big(\pi_t(o' \mid \beta, a), \widehat{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big)\Big)\right\}, \qquad \text{(II.13)}$$

$$\widehat{\delta}_{H-i}^*(\beta) \in \operatorname*{argmax}_{a\in\mathcal{A}} \widehat{\mathrm{M}}\left\{\tilde{\rho}_t(\beta, a), \widehat{\mathbb{S}}\Big(\pi_t(o' \mid \beta, a), \widehat{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big)\Big)\right\} \qquad \text{(II.14)}$$

For instance, in the case of the optimistic π-POMDP value function (*i.e.* based on $\mathbb{S}_\Pi$) with a maximum-based global preference ($\overline{\mathcal{G}}$),

- $\widehat{U_i^*}$ is denoted by $\overline{U_i^*}$ as the optimistic criterion,
- $\tilde{\Psi}(\beta) = \overline{\Psi}(\beta) = \max_{s\in\mathcal{S}} \min\{\Psi(s), \beta(s)\}$,
- $\tilde{\rho}_t(\beta, a) = \overline{\rho_t}(\beta) = \max_{s\in\mathcal{S}} \min\{\rho_t(s, a), \beta(s)\}$,
- $\widehat{M}$ is the maximum operator,
- $\widehat{\mathbb{S}}\Big(\pi_t(o' \mid \beta, a), \widehat{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big)\Big) = \max_{o'\in\mathcal{O}} \min\Big\{\pi_t(o' \mid \beta, a), \overline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big)\Big\}$,

see Algorithm 8.

Another example is the case of the pessimistic π-POMDP value function (*i.e.* based on $\mathbb{S}_{\mathcal{N}}$) with a minimum-based global preference ($\underline{\mathcal{G}}$),

- $\widehat{U_i^*}$ is denoted by $\underline{U_i^*}$ as the pessimistic criterion,
- $\tilde{\Psi}(\beta) = \underline{\Psi}(\beta) = \min_{s\in\mathcal{S}} \max\{\Psi(s), 1-\beta(s)\}$,
- $\tilde{\rho}_t(\beta, a) = \underline{\rho_t}(\beta) = \min_{s\in\mathcal{S}} \max\{\rho_t(s, a), 1-\beta(s)\}$,
- $\widehat{M}$ is the minimum operator,

- $\widehat{\mathbb{S}}\Big(\pi_t\left(o' \mid \beta, a\right), \widehat{U^*_{i-1}}\Big(\nu(\beta,a,o')\Big)\Big) = \min_{o'\in\mathcal{O}} \max\Big\{1-\pi_t\left(o' \mid \beta, a\right), \underline{U^*_{i-1}}\Big(\nu(\beta,a,o')\Big)\Big\}$,

see Algorithm 9.

Finally, in the case of the mixed optimistic-pessimistic π-POMDP value function,

- $\widehat{U^*_i}$ is denoted by U^*_i,
- $\tilde{\Psi}(\beta) = \underline{\Psi}(\beta) = \min_{s\in\mathcal{S}} \max\{\Psi(s), 1-\beta(s)\}$,
- $\tilde{\rho_t}(\beta,a) = \underline{\rho_t}(\beta) = \min_{s\in\mathcal{S}} \max\{\rho_t(s,a), 1-\beta(s)\}$,
- $\widehat{M}$ has not been defined, as the case of intermediate preferences was not considered for this criterion: it may be defined as the minimum or the maximum operator,
- $\widehat{\mathbb{S}}\Big(\pi_t\left(o' \mid \beta, a\right), \widehat{U^*_{i-1}}\Big(\nu(\beta,a,o')\Big)\Big) = \max_{o'\in\mathcal{O}} \min\Big\{\pi_t\left(o' \mid \beta, a\right), U^*_{i-1}\Big(\nu(\beta,a,o')\Big)\Big\}$,

see just above Definition II.1.3 for the case of terminal preference only. The new mixed optimistic-pessimistic criterion of Definition II.1.3 will be the most frequently used in this thesis, mainly because it can be translated into optimistic π-MDP which produce good approximate strategies for probabilistic problem [122], and because it is adapted to the ubounded execution criterion presented below.

Let us recall now that $\tilde{\mathcal{S}}^\pi = \Pi^{\mathcal{S}}_{\mathcal{L}}$ is a finite set of cardinality $\#\tilde{\mathcal{S}}^\pi = \#\mathcal{L}^{\#\mathcal{S}} - (\#\mathcal{L}-1)^{\#\mathcal{S}}$ (the total number of $\#\mathcal{S}$-size vectors valued in $\mathcal{L}$, minus $(\#\mathcal{L}-1)^{\#\mathcal{S}}$ non-normalized distributions). For concrete problems, the state space can be dramatically large: $\#\tilde{\mathcal{S}}^\pi$ explodes and computations become intractable like in standard probabilistic POMDPs. The next section presents a way to exploit a specific structure of the problem that is very common in practice.

II.2 Mixed-Observability and π-MOMDPs

The complexity issue of π-POMDP solving is due to the fact that the size of the belief state space $\Pi^{\mathcal{S}}_{\mathcal{L}}$ exponentially grows with the size of the state space $\mathcal{S}$, see the equation (I.60) of Section I.2.5. However, in practice, states are rarely completely hidden. Using mixed-observability

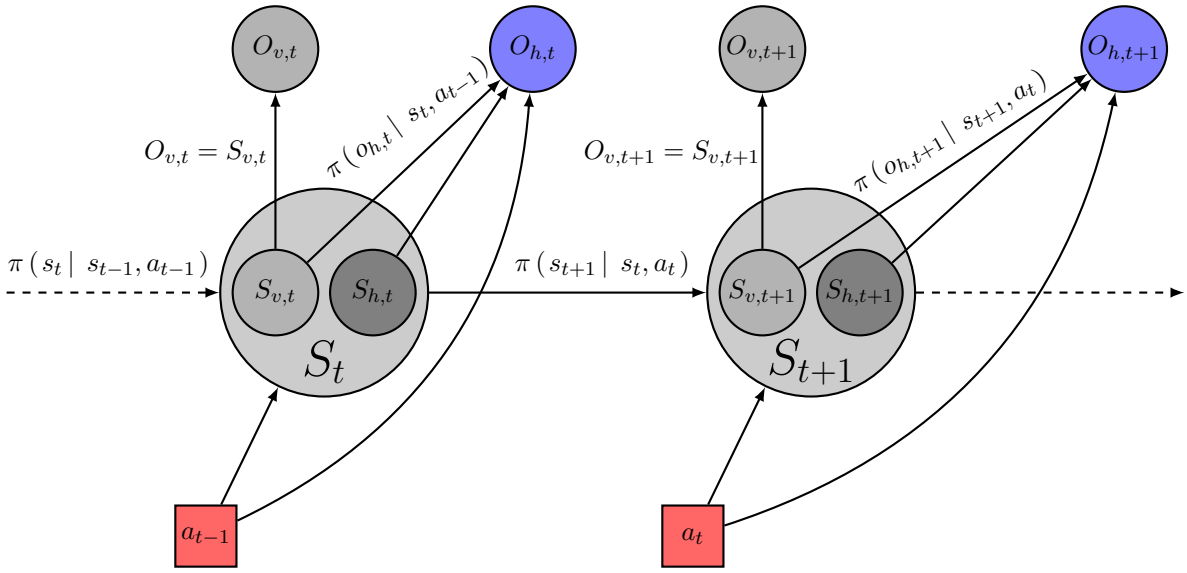


Figure II.1 – *Dynamic Bayesian Network of a π-MOMDP: at time step t, the system state is described by variable $S_t = (S_{v,t}, S_{h,t})$. The received observation is $O_t = (O_{v,t}, O_{h,t})$ with $O_{v,t} = S_{v,t}$, and $O_{h,t}$ depending on S_t and action a_t.*

can be a solution: inspired by a similar recent work in probabilistic POMDPs [100, 3], we present in this section a structured modeling that takes into account situations where the agent directly observes some part of the state. a π-POMDP which models such a situation respects the *mixed-observable property*. Belief states are then used only for the partially observed components and the size of the belief state space is substantially reduced. Thus, this model generalizes both π-MDPs and π-POMDPs.

Like in [3], we assume that the state space $\mathcal{S}$ of a Qualitative Possibilistic Mixed-Observable MDP (π-MOMDP) can be written as a Cartesian product of a visible state space $\mathcal{S}_v$ and a hidden one $\mathcal{S}_h$: $\mathcal{S} = \mathcal{S}_v \times \mathcal{S}_h$. Let $s = (s_v, s_h)$ be a state of the system. The component s_v is directly observed by the agent and s_h is only partially observed through the observations of the set $\mathcal{O}_h$: we denote by $\pi_t\left(o'_h \mid s', a\right)$, the possibility distribution over the future observation $o'_h \in \mathcal{O}_h$ at time step t, knowing the future state $s' \in \mathcal{S}$ and the current action $a \in \mathcal{A}$. Figure II.1 illustrates the structure of this Mixed-Observable model.

The visible state space is integrated to the observation space: $\mathcal{O}_v = \mathcal{S}_v$ and $\mathcal{O} = \mathcal{O}_v \times \mathcal{O}_h$. Then, knowing that the current visible component of the state is s_v, the agent *necessarily* observes $o_v = s_v$ (if $o'_v \neq s'_v$, $\pi_t\left(o'_v \mid s'_v, a\right) = 0$, $\forall a \in \mathcal{A}$). Formally, seen as a π-POMDP, its observation possibility distribution can be written as:

$$\begin{aligned}
\pi_t\left(o' \mid s', a\right) &= \pi_t\left(o'_v, o'_h \mid s'_v, s'_h, a\right) \\
&= \min\left\{\pi_t\left(o'_h \mid s'_v, s'_h, a\right), \pi_t\left(o'_v \mid s'_v\right)\right\} \\
&= \begin{cases} \pi_t\left(o'_h \mid s', a\right) & \text{if } o'_v = s'_v \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{II.15}$$

since $\forall a \in \mathcal{A}$, $\pi_t\left(o'_v \mid s'_v, a\right) = 1$ if $s'_v = o'_v$ and 0 otherwise. The following theorem, based on this equality enables the belief over hidden states to be defined.

Theorem 19 (*Nature of Reachable Belief States*)

Each reachable belief state of a π-MOMDP can be written as an element of $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ where $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ is the set of possibility distributions over $\mathcal{S}_h$: any reachable $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ can be written as (s_v, β_h) with $\beta_h(s_h) = \max_{\overline{s}_v \in \mathcal{S}_v} \beta(\overline{s}_v, s_h)$ and $s_v = \operatorname{argmax}_{\overline{s}_v \in \mathcal{S}_v} \beta(\overline{s}_v, s_h)$.

The proof is given in Annex B.4

As all reachable belief states are in $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ when the mixed-observability property holds, the next theorem rewrites the belief update function for the belief states $\beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ over the hidden system states $s_h \in \mathcal{S}_h$.

Theorem 20 (*Belief Update for a π-MOMDP*)

If a problem can be modeled by a π-MOMDP

$$\Big\langle S_v \times S_h, \mathcal{A}, \mathcal{O}_h, T^{\pi}, O^{\pi}, (\rho_t)_{t=0}^{H-1}, \Psi, \beta_0 = (s_{v,0}, \beta_{h,0}) \Big\rangle,$$

a new belief update function ν_h can be defined: if, at time step t, the current visible state is $s_{v,t} \in \mathcal{S}_v$, the current belief state about the hidden system state is $\beta_{h,t} \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, the selected action is $a_t \in \mathcal{A}$, the next visible state is $s_{v,t+1} \in \mathcal{S}_v$ and the next observation is $o_{h,t+1} \in \mathcal{O}_h$, then the next belief state about the hidden system state is

$$\beta_{h,t+1}(s'_h) = \begin{cases} 1 & \text{if } \begin{array}{l} \pi_t\left(s'_h, s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t\right) \\ \quad = \pi_t\left(s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t\right) \end{array} \\ \pi_t\left(s'_h, s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t\right) & \text{otherwise} \end{cases}, \tag{II.16}$$

where

$$\pi_t\left(s'_v, s'_h, o'_h \mid s_v, \beta_h, a\right) = \min\left\{\pi_t\left(o'_h \mid s', a\right), \max_{s_h \in \mathcal{S}_h} \min\left\{\pi_t\left(s' \mid s_v, s_h, a\right), \beta_h(s_h)\right\}\right\}$$

is the joint possibility distribution over hidden system states $s'_h \in \mathcal{S}_h$ *and visible objects (visible system state and observation)* $s'_v \in \mathcal{S}_v$ *and* $o'_h \in \mathcal{O}_h$*. The notation* $\pi_t\left(s'_v, o'_h \mid s_v, \beta_h, a\right)$ *is for the possibility degree of the visible objects* $\max_{s'_h \in \mathcal{S}_h} \pi_t\left(s'_h, s'_v, o'_h \mid s_v, \beta_h, a\right)$ *(using the notation* $s' = (s'_v, s'_h) \in \mathcal{S} = \mathcal{S}_v \times \mathcal{S}_h$*). This belief update is denoted by*

$$\beta'_h = \nu_h\left(s_v, \beta_h, a, s'_v, o'_h\right).$$

The proof is given in Annex B.5.

The state space of the belief π-MDP resulting from a π-MOMDP can be restricted to the product space $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, *i.e.* a finer belief π-MDP than those presented previously, benefiting from the Mixed-Observability, can be defined: $\langle \tilde{S}^\pi, \tilde{T}^\pi, \mathcal{A}, (\tilde{\rho}_t)_{t=0}^{H-1}, \tilde{\Psi}\rangle$, where

- the state space of the belief π-MDP is defined as $\tilde{S}^\pi = \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$,
- a transition possibility distribution in $\tilde{T}^\pi$ is such that $\forall\{0, \ldots, H-1\}$, $\forall a \in \mathcal{A}$, $\forall\Big[(s_v, \beta_h), (s'_v, \beta'_h)\Big] \in \left(\tilde{S}^\pi\right)^2$,

$$\pi_t\Big((s'_v, \beta'_h)\Big|(s_v, \beta_h), a\Big) = \max_{\substack{o'_h \in \mathcal{O}_h \text{ s.t.} \\ \nu_h(s_v, \beta_h, a, s'_v, o'_h) = \beta'_h}} \pi_t\left(s'_v, o'_h \mid s_v, \beta_h, a\right),$$

  where $\pi_t\left(s'_v, o'_h \mid s_v, \beta_h, a\right)$ is defined just above,

- If the belief-based preferences are optimistic *i.e.* $\tilde{\rho}_t = \overline{\rho_t}$ and $\tilde{\Psi} = \overline{\Psi}$, then $\forall t \in \{0, \ldots, H-1\}$, $\forall s_v \in \mathcal{S}_v$, $\forall \beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, $\forall a \in \mathcal{A}$, the preference functions can be rewritten

$$\overline{\rho_t}(s_v, \beta_h) = \max_{s_h \in \mathcal{S}_h} \min\left\{\rho_t(s_v, s_h), \beta_h(s_h)\right\},$$

  and

$$\overline{\Psi}(s_v, \beta_h) = \max_{s_h \in \mathcal{S}_h} \min\left\{\Psi(s_v, s_h), \beta_h(s_h)\right\}.$$

  Indeed, $\beta(\overline{s_v}, s_h) = 0$ if $\overline{s_v}$ is not the actual visible state s_v, thus, for instance

$$\begin{aligned}\overline{\Psi}(\beta) &= \max_{s \in \mathcal{S}} \min\left\{\Psi(s), \beta(s)\right\} \\ &= \max_{s_h \in \mathcal{S}_h} \min\left\{\Psi(s_v, s_h), \beta(s_h, s_v)\right\}.\end{aligned}$$

- If the belief-based preferences are pessimistic *i.e.* $\tilde{\rho}_t = \underline{\rho_t}$ and $\tilde{\Psi} = \underline{\Psi}$, then $\forall t \in \{0, \ldots, H-1\}$, $\forall s_v \in \mathcal{S}_v$, $\forall \beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, $\forall a \in \mathcal{A}$, the preference functions can be rewritten

$$\underline{\rho_t}(s_v, \beta_h) = \min_{s_h \in \mathcal{S}_h} \max\left\{\rho_t(s_v, s_h), 1 - \beta_h(s_h)\right\},$$

  and

$$\underline{\Psi}(s_v, \beta_h) = \min_{s_h \in \mathcal{S}_h} \max\left\{\Psi(s_v, s_h), 1 - \beta_h(s_h)\right\}.$$

  Indeed, $1 - \beta(\overline{s_v}, s_h) = 1$ if $\overline{s_v}$ is not the actual visible state s_v, thus, for instance

$$\begin{aligned}\underline{\Psi}(\beta) &= \min_{s \in \mathcal{S}} \max\left\{\Psi(s), 1 - \beta(s)\right\} \\ &= \min_{s_h \in \mathcal{S}_h} \max\left\{\Psi(s_v, s_h), 1 - \beta(s_h, s_v)\right\}.\end{aligned}$$

Theorem 21 (*Dynamic Programming Equation of a π-MOMDP*)

The dynamic programming equation becomes:
$\forall (s_v, \beta_h) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$,

$$\widehat{U_0^*}(s_v, \beta_h) = \tilde{\Psi}(s_v, \beta_h),$$

and, $\forall i \in \{1, \ldots, H\}$, $\forall (s_v, \beta_h) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$,

$$\widehat{U_i^*}(s_v, \beta_h) = \max_{a \in \mathcal{A}} \widehat{M}\left\{\tilde{\rho}_t(s_v, \beta_h, a), \widehat{\mathbb{S}}\left(\pi_t\left(s'_v, o'_h \mid s_v, \beta_h, a\right), \widehat{U_{i-1}^*}\left(\nu_h(s_v, \beta_h, a, s'_v, o'_h)\right)\right)\right\},$$

$$\widehat{\delta_i^*}(s_v, \beta_h) \in \operatorname*{argmax}_{a \in \mathcal{A}} \widehat{M}\left\{\tilde{\rho}_t(s_v, \beta_h, a), \widehat{\mathbb{S}}\left(\pi_t\left(s'_v, o'_h \mid s_v, \beta_h, a\right), \widehat{U_{i-1}^*}\left(\nu_h(s_v, \beta_h, a, s'_v, o'_h)\right)\right)\right\}, \tag{II.17}$$

where ν_h is the new belief update function (Theorem 20), and the notations come from the general Dynamic Programming Equation II.13.

The proof is given in Annex B.6.

A standard algorithm would have computed $\widehat{U_i^*}(\beta)$ for each $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ while this new dynamic programming equation leads to an algorithm which computes it only for all $(s_v, \beta_h) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, since only this kind of belief states can be encountered. The size of the new belief space is

$$\#(\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}) = \#\mathcal{S}_v \times \left(\#\mathcal{L}^{\#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_h}\right),$$

which is exponentially smaller than the size of standard π-POMDPs' belief space:

$$\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \#\mathcal{L}^{\#\mathcal{S}_v \times \#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_v \times \#\mathcal{S}_h}.$$

An even finer belief π-MDP could be defined on the set of reachable belief states starting from the initial belief state β_0: $\Pi_{\mathcal{L},\beta_0}^{\mathcal{S}}$ which is a subset of $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$.

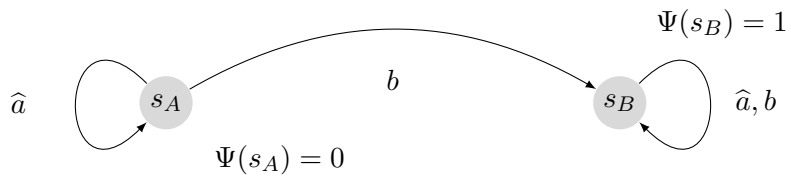
II.3 Infinite Horizon Settings

A finite strategy for possibilistic MOMDPs can now be computed for larger problems using the dynamic programming equation of Theorem 21 and selecting maximizing actions for each state $(s_v, \beta_h) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ (see the equation II.17), as done in the equation (II.14) for each $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$. However, for many problems in practice, it is difficult to determine a horizon size H. The goal of this section is to present an algorithm to solve optimistic π-MOMDPs with terminal preference only, under infinite horizon: it is the first proved algorithm to solve such π-(MO)MDPs.

II.3.1 The π-MDP case

Previous work, [121, 123], on solving π-MDPs proposed a Value Iteration algorithm that was proved to compute optimal value functions, but not necessarily optimal strategies for some problems with cycles. There is a similar issue in *undiscounted* probabilistic MDPs where the greedy strategy at convergence of Value Iteration does not need to be optimal [115]. It is not surprising that we are facing the same issue in π-MDPs since the possibilistic dynamic programming operator does not rely on algebraic products so that it cannot be contracted by some *discount factor* $0 < \gamma < 1$.

Figure II.2 – *Deterministic example showing the limits of previous algorithms*



For infinite horizon problems, the optimistic π-MDP model has to undergo a little change. The dynamics is defined as stationary, as in the probabilistic case, see Section I.1.4: $\forall t \geqslant 0$, $\forall (s, s') \in \mathcal{S}^2$, $\forall \in \mathcal{A}$,

$$\pi_t \left(s' \mid s, a \right) = \pi \left(s' \mid s, a \right).$$

The case of terminal preference only, is considered: starting from the general optimistic π-MDP (with intermediate preferences and minimum-based global preference, see Section I.2.4) the preference functions can be set equivalently to $\forall t \geqslant 0$, $\forall s \in \mathcal{S}$, $\forall \in \mathcal{A}$,

$$\rho_t(s, a) = 1,$$

and thus, only the terminal preference function Ψ has an effect on the criterion, and has to be defined for the instantiation of the π-MDP.

Algorithm 10: Optimistic π-MDP VI Algorithm – Terminal Preference Only

for $s \in \mathcal{S}$ **do**
  $\overline{U^*}(s) \leftarrow 0$;
  $\overline{U^c}(s) \leftarrow \Psi(s)$;
  $\overline{\delta^*}(s) \leftarrow \widehat{a}$;
while $\overline{U^*} \neq \overline{U^c}$ **do**
  $\overline{U^*} = \overline{U^c}$;
  for $s \in \mathcal{S}$ **do**
    $\overline{U^c}(s) \leftarrow \max\limits_{a \in \mathcal{A}} \max\limits_{s' \in \mathcal{S}} \min \left\{ \pi \left(s' \mid s, a \right), \overline{U^*}(s') \right\}$;
    if $\overline{U^c}(s) > \overline{U^*}(s)$ **then**
      $\overline{\delta^*}(s) \in \operatorname{argmax}\limits_{a \in \mathcal{A}} \max\limits_{s' \in \mathcal{S}} \min \left\{ \pi \left(s' \mid s, a \right), \overline{U^*}(s') \right\}$;
return $\overline{U^*}$, $\overline{\delta^*}$;

To the best of our knowledge, we propose here the first Value Iteration algorithm for π-MDPs, that provably returns an optimal strategy, and that is different from the one of [123]. Indeed, in the deterministic example of Figure II.2, action $\widehat{a}$, which is clearly suboptimal in state s_A, was found to be optimal in s_A with this algorithm: however it is clear that since $\pi \left(s_B \mid s_A, b \right) = 1$ and $\Psi(s_B) = 1$, $\overline{U_1^*}(s_A) = 1$. Obviously, $\overline{U_1^*}(s_B) = 1$ and since $\pi \left(s_A \mid s_A, \widehat{a} \right) = 1$, $\max_{s' \in \mathcal{S}} \min \left\{ \pi \left(s' \mid s_A, a \right), \overline{U_1^*}(s') \right\} = 1 \ \forall a \in \{ \widehat{a}, b \} = \mathcal{A}$, *i.e.* all actions are optimal in s_A. The "if" condition of Algorithm 10 permits to select the optimal action b during the first step. This condition and the initialization, which were not present in previous algorithms of the literature, are needed to prove the optimality of the strategy. The proof, which is quite lengthy and intricate, is presented in Annex C. This sound algorithm for π-MDPs will then be extended to π-MOMDPs in the next section.

As mentioned in [121], we assume the existence of an action "stay", denoted by $\widehat{a}$, which lets the system in the same state with necessity 1. This action is the possibilistic counterpart of

the discount parameter γ in the probabilistic model, as it guarantees convergence of the Value Iteration algorithm. However, we will see that action $\widehat{a}$ is finally used only on some particular satisfactory states. Note that a similar assumption is used to compute optimal strategies in the framework of deterministic processes (classical planning) whose horizon is not specified [85].

We denote by $\widehat{\delta}$ the decision rule such that $\forall s \in \mathcal{S}$, $\widehat{\delta}(s) = \widehat{a}$. The set of all the finite strategies is $\Delta = \cup_{i \geqslant 1} \Delta_i$, and $\#\delta$ is the size of a strategy (δ) in terms of decision epochs. We can now define the optimistic criterion for an infinite horizon: if $(\delta) \in \Delta$,

$$\overline{U}\Big(s_0, (\delta)\Big) = \max_{\mathcal{T} \in \mathcal{T}_{\#\delta}} \min \Big\{ \pi\Big(\mathcal{T}\Big| s_0, (\delta)\Big), \Psi(s_{\#\delta}) \Big\}, \tag{II.18}$$

where $\mathcal{T} = (s_1, \ldots, s_{\#\delta})$ is a trajectory of system states, $\mathcal{T}_{\#\delta}$ the set of such trajectories, and

$$\pi\Big(\mathcal{T}\Big| s_0, (\delta)\Big) = \min_{i=0}^{\#\delta - 1} \pi\Big(s_{i+1}\Big| s_i, \delta_i(s_i)\Big).$$

Theorem 22 (*Optimality of the VI Algorithm for Optimistic π-MDPs*)

If there exists an action $\widehat{a}$ such that, for each $s \in \mathcal{S}$, $\pi(s' \mid s, \widehat{a}) = 1$ if $s' = s$ and 0 otherwise, then Algorithm 10 computes the maximum optimistic criterion and an optimal strategy, i.e. maximizing the criterion (II.18), which is stationary (i.e. which does not depend on the stage of the process t).

The proof is given in Annex C. Note that, as in the probabilistic case (see Section I.1.5), the computed optimal strategy is stationary *i.e.* does not depend on the time step of the process.

Let s be a state such that $\overline{\delta^*}(s) = \widehat{a}$, where $\overline{\delta^*}$ is the returned strategy. By looking at Algorithm 10, it can be noted that $\overline{U^*}(s)$ always remains equal to $\Psi(s)$ during the iterations of the algorithm after the first entry in the while loop. Thus, $\forall s' \in \mathcal{S}$, either $\forall a \in \mathcal{A}$, $\Psi(s) \geqslant \pi(s' \mid s, a)$, or $\Psi(s) \geqslant \overline{U^*}(s')$. If the problem is non trivial, it means that s is a goal ($\Psi(s) > 0$) and that degrees of possibility of transitions to better goals are less than the degree of preference for s.

II.3.2 Value Iteration for π-MOMDPs

We are now ready to propose the Value Iteration algorithm for π-MOMDPs which reduces to a belief π-MDP which is optimistic and with terminal preference only (whose Value Iteration algorithm has been presented in the previous section). This Value Iteration algorithm is, for instance, devoted to π-MOMDPs with the π-POMDP mixed optimistic-pessimistic criterion, see Definition II.1.3: the terminal preference function is then $\underline{\Psi}(s_v, \beta_h) = \min_{s_h \in \mathcal{S}_h} \max\{\Psi(s_v, s_h), 1 - \beta_h(s_h)\}$. Another example of appropriate π-MOMDP with the optimistic π-POMDP criterion (see Algorithm 8, removing intermediate preferences): the terminal preference function is in this case $\overline{\Psi}(s_v, \beta_h) = \max_{s_h \in \mathcal{S}_h} \min\{\Psi(s_v, s_h), \beta_h(s_h)\}$.

Note that Algorithm 11 has the same structure as Algorithm 10. Note as well that a π-MOMDP is a π-MDP over $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$. Recall that the transition possibility distribution is

$$\pi_t\Big((s'_v, \beta'_h)\Big|(s_v, \beta_h), a\Big) = \max_{\substack{o'_h \in \mathcal{O}_h \text{ s.t.} \\ \nu_h(s_v, \beta_h, a, s'_v, o'_h) = \beta'_h}} \pi_t(s'_v, o'_h \mid s_v, \beta_h, a).$$

To satisfy the assumption of Theorem 22, it suffices to ensure that $\pi_t\Big((s'_v, \beta'_h)\Big|(s_v, \beta_h), a\Big) = 1$ if $s'_v = s_v$ and $\beta'_h = \beta_h$, and 0 otherwise. This property is verified if the two following conditions hold: $\pi(s'_v, s'_h \mid s_v, s_h, \widehat{a}) = 1$ if $(s'_v, s'_h) = (s_v, s_h)$, and 0 otherwise, and there exists

Algorithm 11: π-MOMDP Value Iteration Algorithm

for $s_v \in \mathcal{S}_v$ **and** $\beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ **do**
  $\overline{U^*}(s_v, \beta_h) \leftarrow 0$;
  $\overline{U^c}(s_v, \beta_h) \leftarrow \tilde{\Psi}(s_v, \beta_h)$;
  $\overline{\delta^*}(s_v, \beta_h) \leftarrow \widehat{a}$;
while $\overline{U^*} \neq \overline{U^c}$ **do**
  $\overline{U^*} \leftarrow \overline{U^c}$;
  for $s_v \in \mathcal{S}_v$ **and** $\beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ **do**
    $\overline{U^c}(s_v, \beta_h) \leftarrow \max_{a \in \mathcal{A}} \max_{s'_v \in \mathcal{S}} \max_{o'_h \in \mathcal{O}_h} \min \left\{ \pi\left(s'_v, o'_h \middle| s_v, \beta_h, a\right), \overline{U^*}\left(s'_v, \nu_h(s_v, \beta_h, s'_v, o'_h)\right) \right\}$;
    if $\overline{U^c}(s_v, \beta_h) > \overline{U^*}(s_v, \beta_h)$ **then**
      $\overline{\delta^*}(s_v, \beta_h) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{s'_v \in \mathcal{S}} \max_{o'_h \in \mathcal{O}_h} \min \left\{ \pi\left(s'_v, o'_h \middle| s_v, \beta_h, a\right), \overline{U^*}\left(s'_v, \nu_h(s_v, \beta_h, s'_v, o'_h)\right) \right\}$;
return u^*, δ^* ;

an observation "nothing" $\widehat{o_h}$ that is required for each state when $\widehat{a}$ is chosen *i.e.* $\forall (s'_v, s'_h) \in \mathcal{S}$, $\pi\left(o'_h \mid s'_v, s'_h, \widehat{a}\right) = 1$ if $o'_h = \widehat{o_h}$ and 0 otherwise. Indeed, it means that $\widehat{o_h}$ is received for sure when $\widehat{a}$ is selected: no information is provided by this observation, and the belief does not evolve.

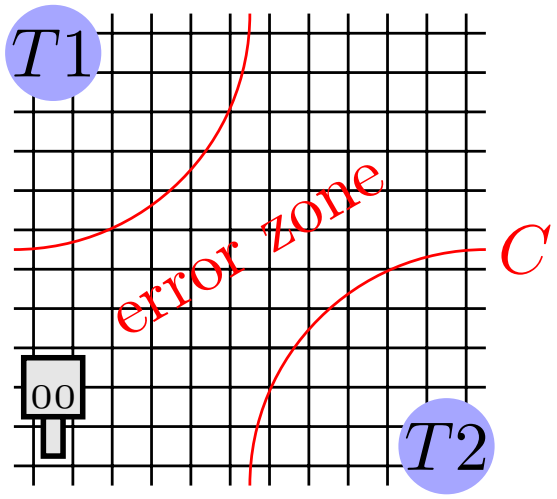
II.4 Results on a Robotic Mission and Possibilistic Belief State Behaviour

This section is devoted to the use of strategies computed by Algorithm 11 in the context of a concrete robotic problem. Consider a robot over a grid of size $g \times g$, with $g > 1$. It always perfectly knows its location on the grid $(x, y) \in \{1, \ldots, g\}^2$, which forms the visible state space $\mathcal{S}_v$. It starts at location $s_{v,0} = (1, 1)$. Two targets are located at $(x_1, y_1) = (1, g)$ ("target 1") and $(x_2, y_2) = (g, 1)$ ("target 2") on the grid, and the robot perfectly knows their positions. One of the targets is A, the other B and the robot's mission is to identify and reach target A as soon as possible. The robot does not know which target is A: the two situations, "target 1 is A" ($A1$) and "target 2 is A" ($A2$), constitute the hidden state space $\mathcal{S}_h$. The moves of the robot are deterministic and its actions $\mathcal{A}$ consist in moving in the four directions plus the action "stay". At each stage of the process, the robot analyzes pictures of each target and gets then an observation of the targets' natures: the two targets (oAA) can be observed as A, or target 1 (oAB), or target 2 (oBA) or no target (oBB).

In the probabilistic framework, the probability of having a good observation of target $i \in \{1, 2\}$, is not really known but approximated by $\mathbf{p}\left(good_i \mid x, y\right) = \frac{1}{2}\left[1 + \exp\left(-\frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{D}\right)\right]$ where $(x, y) = s_v \in \{1, \ldots, g\}^2$ is the location of the robot, (x_i, y_i) the position of target i, and D a normalization constant. We suppose that the observations of both targets are independent: then, for instance, the probability $\mathbf{p}\left(oAB \mid (x, y), A1\right)$ is equal to $\mathbf{p}\left(good_1 \mid (x, y)\right) \cdot \mathbf{p}\left(good_2 \mid (x, y)\right)$, $\mathbf{p}\left(oAA \mid (x, y), A1\right)$ to $\mathbf{p}\left(good_1 \mid (x, y)\right) \cdot [1 - \mathbf{p}\left(good_2 \mid (x, y)\right)]$, and so on. Each step of the process before reaching a target costs 1, reaching target A is rewarded by 100, and -100 for B. The probabilistic strategy was computed in mixed-observability settings with APPL[1] based on SARSOP [100, 84] (see

[1]The used software is available at `http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/`

Figure II.3 – *Illustration of a robotic mission, first experiment on π-MOMDPs*



Section I.1.11), using a precision of 0.046 (the memory limit is reached for higher precisions) and a discount factor $\gamma = 0.99$. This problem cannot be solved with the exact algorithm for MOMDPs [3] because it consumes the entire RAM after 15 iterations.

In the framework of Qualitative Possibility Theory, it is considered always possible to observe the good target: $\pi(good \mid x, y) = 1$. Secondly, the farther the robot is from target i, the more likely it is to badly observe it (e.g. observe A instead of B), which is a reasonable assumption if the actual probabilistic observation model is imprecisely known: $\pi(bad_i \mid x, y) = \frac{(x-x_i)^2+(y-y_i)^2}{2(g-1)^2}$, and thus, $\mathcal{L}$ can be defined by $\left\{0, \frac{1}{2(g-1)}, \ldots, 1\right\}$, or any other scale preserving the fact that the possibility degree of misperceiving increases with the distance from the considered target. The observation of a target is also considered as NI-independent from the the observation of the other target (see Definition I.2.6 of Section I.2.2). Thus for instance, $\pi(oAB \mid (x,y), A1) = 1$, $\pi(oAA \mid (x,y), A1) = \pi(bad_2 \mid x, y)$, $\pi(oBA \mid (x,y), A1) = \min\{\pi(bad_1 \mid x, y), \pi(bad_2 \mid x, y)\}$, etc. Note that the construction of this model with a probability-possibility transformation [63] would have been equivalent. The terminal preference function Ψ is equal to 0 for all the system's states and to 1 for states $[(x_1, y_1), A1]$ and $[(x_2, y_2), A2]$ where (x_i, y_i) is the position of target i. As mentioned in [121], the computed strategy guarantees a shortest path to a goal state. The strategy then aims at reducing the mission duration. The mixed optimistic-pessimistic criterion, see Definition II.1.3, is used here to compute the strategy.

Standard π-POMDPs, which do not exploit mixed-observability contrary to our π-MOMDP model, could not solve even very small 3×3 grids. Indeed, for this problem, $\#\mathcal{L} \geqslant 5$, $\#\mathcal{S}_v = 9$, and $\#\mathcal{S}_h = 2$. Thus, $\#\mathcal{S} = \#\mathcal{S}_v \times \#\mathcal{S}_h = 18$ and the number of belief states is then $\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \mathcal{L}^{\#\mathcal{S}} - (\mathcal{L}-1)^{\#\mathcal{S}} \geqslant 5^{18} - 4^{18} \geqslant 3.7.10^{12}$ instead of 81 states with a π-MOMDP. Therefore, the following experimental results could **not** be conducted with standard π-POMDPs, which indeed justifies our present work on π-MOMDPs.

In order to compare the performances of the probabilistic and possibilistic models, we compare the average of their total (undiscounted) rewards at execution, *i.e.* a reward-based criterion really close to the probabilistic criterion (the same with $\gamma = 1$): since the situation (the nature of the targets) is fully known by the agent when the robot is at a target's location, it can not end up choosing target B. If k is the number of time steps needed to identify and reach the correct target, then the total reward is $100 - k$.

We consider now that, in reality (thus here for the simulations), and contrary to what is described by the model, the image processing algorithms badly perform when the robot is far away from targets, *i.e.*, if $\forall i \in \{1, 2\}$, $\sqrt{(x-x_i)^2 + (y-y_i)^2} > C$, with C a positive constant,

then $\mathbf{p}(good_i \mid x, y) = 1 - P_{bad} < \frac{1}{2}$. In all other cases, we assume that the probabilistic model is the good one. Figure II.3 illustrates this problem, and indicates the zone where the robot misperceives calling it "error zone". For the following numerical experiments, we used 10^4 simulations to compute the statistical mean of the total reward at execution. The grid was 10×10, $D = 10$ and $C = 4$.

Figure II.4.a shows that the probabilistic model is more affected by the introduced error than the possibilistic one: it shows the total reward at execution of each model as a function of P_{bad}, the probability of badly observing targets when the robot's location is such that $\sqrt{(x - x_i)^2 + (y - x_i)^2} > C$, *i.e.* when the robot is in the "error zone". This is due to the fact that the possibilistic update of the belief state does not take into account new observations when the robot has already obtained a more reliable one, whereas the probabilistic model modifies the current belief at each step. Indeed, as there are only two hidden states ($A1$ and $A2$) that we now denote by s_h^1 and s_h^2, if $\beta_h(s_h^1) < 1$, then $\beta_h(s_h^2) = 1$ (possibilistic normalization). As the hidden state does not change during the mission, the joint possibility distribution over the hidden state and the observation is the minimum of the possibility distribution over the system state (described by the current belief state) and the observation possibility degree: e.g. for s_h^1, the joint distribution is $\min\{\pi(o_h \mid s_v, s_h^1, a), \beta_h(s_h^1)\}$, with $o_h \in \{oAA, oAB, oBA, oBB\}$. It implies that the joint possibility of s_h^1 and the observation o_h, is smaller than $\beta_h(s_h^1)$. The possibilistic counterpart of the belief update equation, see the equation (I.62) or the equation (II.16) for Mixed-Observability settings, ensures that the next belief is either more skeptic about s_h^1 if the observation is more reliable and confirms the prior belief ($\pi(o_h \mid s_v, s_h^1, a)$ is smaller than $\beta_h(s_h^1)$); or changes to the opposite belief if the observation is more reliable and contradicts the prior belief ($\pi(o_h \mid s_v, s_h^2, a)$ is smaller than both $\beta_h(s_h^1)$ and $\pi(o_h \mid s_v, s_h^1, a)$); or yet simply remains unchanged if the observation is not more informative than the current belief.

The following theorem gives sufficient conditions leading to an informative possibilistic belief update *i.e.* which make the resulting belief state more specific (see Definition I.2.5 of Section I.2.1) than the previous one: a belief state $\beta_1 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ is said more specific than a belief state $\beta_2 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ if $\forall s \in \mathcal{S}$, $\beta_1(s) \leqslant \beta_2(s)$. In order to get a total order on $\Pi_{\mathcal{L}}^{\mathcal{S}}$, the ranking relation $\preceq$ is defined to sort belief states with respect to their specificity:

$$\beta_1 \preceq \beta_2 \Leftrightarrow \sum_{s \in \mathcal{S}} \beta_1(s) \leqslant \sum_{s \in \mathcal{S}} \beta_2(s).$$

Note that if β_1 is more specific than β_2, then $\beta_1 \preceq \beta_2$.

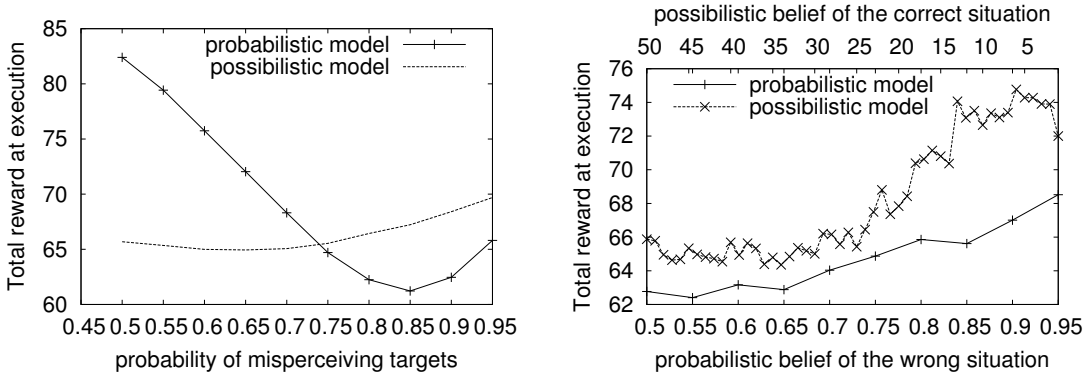
Theorem 23 (*Conditions for an increasing specificity of the belief states*)

Let $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ be the initial belief state modeling the total ignorance i.e. $\forall s \in \mathcal{S}$, $\beta_0(s) = 1$. If the transition function $\pi(s' \mid s, a)$ is deterministic, and if the observations are not informative, i.e. $\forall s' \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\forall o' \in \mathcal{O}$, $\pi(o' \mid s', a) = 1$, then $\beta_{t+1} \preceq \beta_t$, where $\beta_{t+1} \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ is the result of an update (I.62) of the belief state $\beta_t \in \Pi_{\mathcal{L}}^{\mathcal{S}}$,

The result $\beta_{t+1} \preceq \beta_t$ remains true if, for each action $a \in \mathcal{A}$, the transition possibility distributions $\pi(s' \mid s, a) = \mathbb{1}_{\{s=s'\}}$ (i.e. is equal to 1 if $s' = s$ and 0 otherwise), and $\forall o' \in \mathcal{O}$, $\forall a \in \mathcal{A}$, $\forall s', \tilde{s} \in \mathcal{S}$, $\pi(o' \mid s', a) \neq \pi(o' \mid \tilde{s}, a)$.

The proof is given in Annex B.7. Note that this theorem was devoted to π-POMDP. However, the same result holds for π-MOMDPs, replacing $s \in \mathcal{S}$ by $s_h \in \mathcal{S}_h$ and $o \in \mathcal{O}$ by $o_h \in \mathcal{O}_h$. Note also that the conditioning presented in Definition I.2.10 of Section I.2.2 leads to another belief update: with this one, if β_{t+1} is the update of the belief state β_t, β_{t+1} is not less specific than β_t. However, it does not ensure that $\beta_{t+1} \preceq \beta_t$.

The probabilistic belief update does not have these capabilities to directly change to the opposite belief and to disregard less reliable observations: the robot then proceed towards the



(a) Varying P_{bad}. (b) Varying β_0: here $\mathcal{L} = \{0, 1, \ldots, 50\}$.

Figure II.4 – *Comparison of the total reward gathered at execution for possibilistic and probabilistic models.*

wrong target because it is initially far away and thus badly observes targets (without knowing it). When it is close to this target, it gets good observations and gradually modifies its belief which becomes true enough to convince it to go towards the right target. However it has to cross a remote area away from targets: this yet gradually modifies its belief, which becomes wrong, and the robot finds itself in the same initial situation: it loses thus a lot of time to get out of this loop. We can observe that the total reward increases for high probabilities of misperceiving P_{bad}: this is because this high error leads the robot to reach the wrong target faster, thus to entirely know that the true target is the other one.

Now if we set $P_{bad} = 0.8$ and evaluate the total reward at execution for different wrong initial belief states, we get Figure II.4.b with the same parameters: we compare here the possibilistic model and the probabilistic one when the initial belief state is strongly oriented towards the wrong hidden states (i.e. the agent strongly believes that target 1 is B whereas it is A in reality). Note that the possibilistic belief state of the good target decreases when the necessity of the bad one increases. This figure shows that the possibilistic model yields higher rewards at execution if the initial belief state is wrong and the observation function is imprecise [2].

II.5 Conclusion

We have proposed a Value Iteration algorithm for possibilistic MDPs, which can produce optimal stationary strategies in infinite horizon contrary to previous methods. We have provided a complete proof of convergence that relies on the existence of intermediate "stay" actions that vanish for non goal states in the final optimal strategy. Finally, we have extended this algorithm to a new Mixed-Observable possibilistic MDP model, whose complexity is exponentially smaller than possibilistic POMDPs, so that we could compare π-MOMDPs with their probabilistic counterparts on realistic robotic problems. Our experimental results show that possibilistic strategies can outperform probabilistic ones when the probabilities of the observation function are not precisely known, and thus defined quite differently from the actual ones.

A value iteration algorithm for the pessimistic π-MDPs can be easily written on the basis of the optimistic value iteration algorithm, Algorithm 10. However, the optimality of the returned

[2] The implementation of the solver, as well as a generator of descriptions of such recognition problems (expressed in the RDDL language [126]) which is the input of the solver, are available on the repository `https://github.com/drougui/ppudd` : executions can be simulated using the possibilistic optimal strategy which is the output of the solver.

strategy seems hard to prove, essentially because it is not enough to construct a maximizing trajectory, as the proof in Annex C does. The works [147, 114] may be useful materials to help us to get results about pessimistic π-MDP in infinite horizon settings.

Note that, if some probabilistic information is really known by the designers of the model, the π-POMDP is a qualitative approximation of the probabilistic POMDP: in such cases the quantitative information which is available is not taken into account, in order to simplify the problem resolution. Indeed, this model only implies maximum and minimum operators. Note also that if the model has been built from expert qualitative information about the plausible behaviour of the system, an arbitrary probabilistic POMDP offers no more guarantee than the possibilistic one which only uses really available description of the problem.

Finally, as highlighted by the experiment, while the π-POMDPs are based on a computationally simpler uncertainty model than the probabilistic POMDP, the possibilistic belief updating process may has an interesting behaviour. Under some sufficient assumptions given by Theorem 23, the belief state is not modified by less reliable information than the previously gathered information, but is able to change to a quite opposite belief state if an information item which suggests it and which is more reliable is received. More complex problems have to be studied to get a better overview of this behavior in a wider set of situations. However, π-POMDPs with a large system state (or π-MOMDPs with large $\mathcal{S}_h$) cannot be solved with reasonable computation times by algorithms developed until now. The next chapter presents and uses another problem structure, the possibilistic counterpart of *factored POMDPs*, which leads to easier computations of optimal possibilistic strategies and making various experiments: the developed solvers use Algebraic Decision Diagrams avoiding some useless computations and making handled data more compact.

III Development of Symbolic Algorithms to Solve π-POMDPs

In this chapter, we propose the study of factored π-MOMDP models in order to solve large structured planning problems under qualitative uncertainty, or considered as qualitative approximations of probabilistic problems. Building upon the *Stochastic Planning Using Decision Diagrams* (SPUDD) algorithm for solving factored probabilistic MDPs, we have desined a symbolic algorithm named PPUDD for solving factored π-MOMDPs. Whereas the number of leaves of SPUDD's decision diagrams may be as large as the state space since their values are real numbers aggregated through additions and multiplications, the number of leaves of PPUDD ones is bounded by the number of elements in $\mathcal{L}$ because their values always remain in the finite scale $\mathcal{L}$ via min and max operations only. Finally, we present a sound transformation from factored mixed-observable possibilistic problems with both hidden and visible state variables to fully observable ones, on which PPUDD is run. Our experiments show that PPUDD's computation time is much smaller than SPUDD, Symbolic-HSVI and APPL for possibilistic and probabilistic versions of the same benchmarks under either total or mixed-observability, while still providing high-quality strategies. The performance of the strategies computed by PPUDD has been tested in the International Probabilistic Planning Competition (IPPC 2014) whose results are exposed here.

III.1 Introduction

As explained at the end of the previous chapter, starting from a probabilistic MOMDP [100, 3], the use of Possibility Theory instead of Probability Theory leads to an approximation of the initial probabilistic model [122]: probabilities and rewards are replaced by qualitative statements that lie in a finite scale (as opposed to continuous ranges in the probabilistic framework), which results in simpler computations. Possibilities and probabilities have similar behaviors for problems with low entropy probability distributions [55]. However, the decision resulting from both models can be completely different in practice. Consider for example a situation in which three actions a_A, a_B and a_C lead to three different sets of system states:

- action a_A leads to $\mathcal{S}_A = \{ s_A^1, s_A^2 \}$, with $\mathbf{p}\left(s_A^1 \mid a_A\right) = \mathbf{p}\left(s_A^2 \mid a_A\right) = 0.5$, $r(s_A^1, a_A) = 1$ and $r(s_A^2, a_A) = 0$;
- action a_B leads to $\mathcal{S}_B = \{ s_B^1 \}$ ($\mathbf{p}\left(s_B^1 \mid a_B\right) = 1$) with $r(s_B^1, a_B) = 0.5$;
- action a_C leads to $\mathcal{S}_C = \{ s_C^1, \ldots, s_C^7 \}$ with $\mathbf{p}\left(s_C^1 \mid a_C\right) = 0.4$, $\mathbf{p}\left(s_C^2 \mid a_C\right) = \ldots = \mathbf{p}\left(s_C^7 \mid a_C\right) = 0.1$, $r(s_C^1, a_C) = 0$ and $r(s_C^2, a_C) = \ldots = r(s_C^7, a_C) = 1$.

Actions a_A and a_B lead to the same average gain: 0.5. However, action a_C leads to a better one: 0.6. Now, let us define a possibilistic model respecting the ranking of event plausibilities:

- $\pi\left(s_A^1 \mid a_A\right) = \pi\left(s_A^2 \mid a_A\right) = 1$, $\rho(s_A^1, a_A) = 1$ and $\rho(s_A^2, a_A) = 0$;

- $\pi\left(s_B^1 \mid a_B\right) = 1$ and $\rho(s_B^1, a_B) = \frac{3}{4}$;
- $\pi\left(s_C^1 \mid a_C\right) = \frac{1}{2}$, $\pi\left(s_C^2 \mid a_C\right) = \ldots = \pi\left(s_C^7 \mid a_C\right) = \frac{1}{4}$, $\rho(s_C^1, a_C) = 0$ and $\rho(s_C^2, a_C) = \ldots = \rho(s_C^7, a_C) = 1$.

The action a_A is chosen by the optimistic approach. Indeed, it is entirely possible to reach s_A^1 ($\pi\left(s_A^1 \mid a_A\right) = 1$) whose preference is 1. The action a_B is chosen by the pessimistic approach since the preference $\frac{3}{4}$ is reached with certainty with this action: action a_A leads potentially to s_A^1 with the preference $\rho(s_A^2, a_A) = 0$, and a_C leads potentially to s_C^1 with $\rho(s_C^1, a_C) = 0$. Thus, in some cases, the three approaches, (the optimistic and pessimistic possibilistic approaches, and the probabilistic approach) may select three different actions.

Recall that the possibilistic approach benefits from computations on *finite* belief state spaces, whereas probabilistic MOMDPs require *infinite* ones. It means that the same algorithmic techniques can be used to solve π-MDPs, π-POMDPs or π-MOMDPs. What is lost in precision of the uncertainty model is saved in computational complexity. Problems where the uncertainty model is imprecisely known are also are naturally well-modeled by π-MOMDPs [49]: e.g., occurrence frequencies of observations resulting from a complex image processing algorithm depend on many environmental factors like light conditions so that they are generally not precisely known.

Previously presented works on π-(MO)MDPs do not totally take advantage of the problem structure, i.e. visible or hidden parts of the state can be themselves factorized into many state variables, which are flattened by current possibilistic approaches. In probabilistic settings, factored MDPs and Symbolic Dynamic Programming (SDP) frameworks [23, 75] have been extensively studied in order to reason directly at the level of state variables rather than state space in extension. A more recent work is for instance [116]. However, factored probabilistic MOMDPs have not yet been proposed to the best of our knowledge, probably because of the intricacy of reasoning with a mixture of a finite state subspace and an infinite belief state subspace due to the probabilistic model – contrary to the possibilistic case where both subspaces are finite. The famous algorithm SPUDD [75] solves factored probabilistic MDPs by using symbolic functional representations of value functions and strategies in the form of Algebraic Decision Diagrams (ADDs) [6], which compactly encode real-valued functions of Boolean variables: ADDs are directed acyclic graphs whose nodes represent state variables and leaves are the function's values. Instead of updating state values individually at each iteration of the algorithm, they are aggregated within ADDs and operations are symbolically and directly performed on ADDs over many states at once. However, SPUDD suffers from the manipulation of potentially huge ADDs in the worst case: for instance, expectation involves additions and multiplications of real values (probabilities and rewards), creating other values in-between, in such a way that the number of ADD leaves may equal the size of the state space, *i.e.* exponential in the number of state variables. Therefore, the work presented here is motivated by the simple observation that **symbolic operations with possibilistic MDPs would necessarily limit the size of ADDs**: indeed, this formalism operates over a *finite* possibilistic scale $\mathcal{L}$ with only max and min operations involved, which implies that all manipulated values remain in the finite scale $\mathcal{L}$, which is generally far smaller than the number of states.

Figure III.1 shows that ADDs used in the possibilistic settings have a limited number of nodes since the number of their leaves are at most equal to the cardinality of the possibilistic finite scale $\mathcal{L}$: the maximal size (maximal number of nodes) of an ADD whose leaves are in $\mathcal{L}$, is represented as a function of $\#\mathcal{L}$, in the case of 8 and 10 variables. The largest ADD in quantitative settings with 8 (resp. 10) variables have $2^9 - 1$ (resp. $2^{11} - 1$) nodes, as represented also in Figure III.1. Operating ADDs in the possibilistic framework would behave much like manipulating Binary Decision Diagrams (BDDs) [26], which are generally more compact and thus more efficient than ADDs.

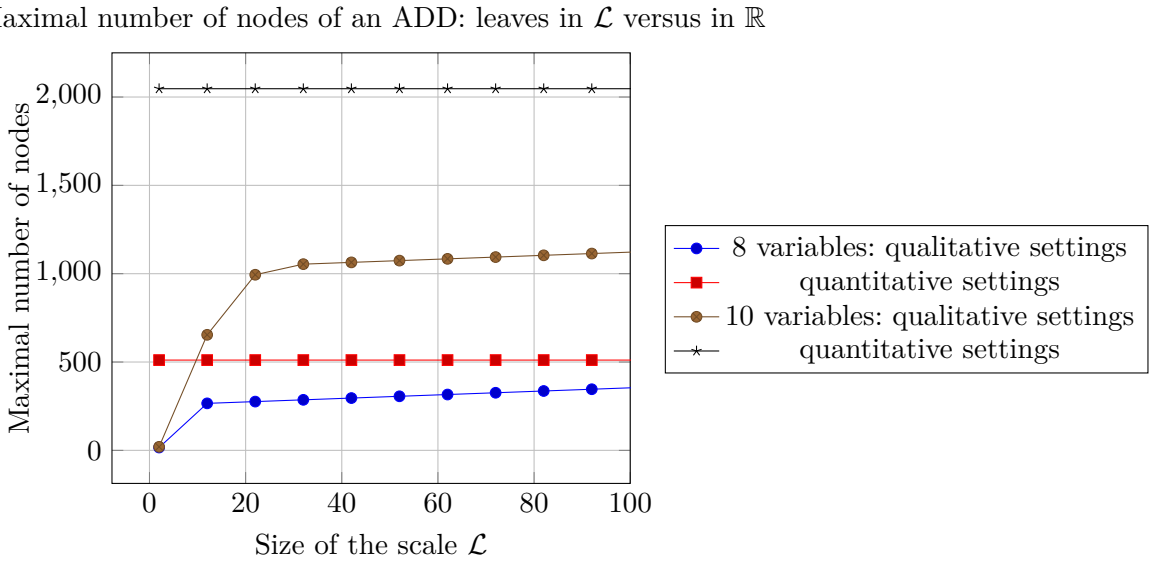


Figure III.1 – *The maximal size (total number of nodes) of an ADD whose values are in $\mathcal{L}$, i.e. in qualitative settings, is limited: the upper bound is represented in blue and brown lines with circles, as a function of the size of $\mathcal{L}$. When the leaves of the ADD are in $\mathbb{R}$, the number of its nodes is potentially exponential in the number of variables: the upper bound is represented with red squares and black stars (as a constant function of the size of $\mathcal{L}$).*

In this chapter we present a Symbolic Dynamic Programming algorithm for solving factored π-MOMDPs named Possibilistic Planning Using Decision Diagram (PPUDD). This contribution alone is insufficient, since it relies on a belief state variable whose number of values is exponential in the size of the state space. Therefore, our second contribution is a theorem to factorize the belief state itself in many variables under some assumptions about dependence relationships between state and observation variables of a π-MOMDP, which makes our algorithm more tractable while still exact and optimal. We note that our idea of factorizing the belief state under mixed-observability is sufficiently general to be reused in probabilistic models. Then, we experimentally assess our approach on possibilistic and probabilistic versions of the same benchmarks: PPUDD against SPUDD and APRICODD [138] under total observability to demonstrate that the generality of our approximate approach does not penalize performances on restrictive submodels; PPUDD against symbolic HSVI [133] (a symbolic version of HSVI, see Section I.1.11) and APPL [84, 100] (already used in previous chapter, and based on SARSOP, see Section I.1.11) under mixed-observability. These promising results were a motivation to take part in the International Probabilistic Planning Competition 2014 (IPPC): results of PPUDD on the fully observable track of IPPC 2014 are then provided and discussed. A general practical solver for solving π-MOMDP using ADDs and available on the repository `https://github.com/drougui/ppudd`) is finally detailed: performances of this solver on the problems of the Fully Observable track of IPPC are also presented.

III.2 Solving factored π-MOMDPs using symbolic dynamic programming

Factored MDPs [75] have been used to efficiently solve structured sequential decision problems under probabilistic uncertainty, by symbolically reasoning on functions of states via decision diagrams rather than on individual states. Inspired by this work on factored MDPs this section sets up a symbolic resolution of factored π-MOMDPs, which assumes that the visible state space $\mathcal{S}_v$, the hidden one $\mathcal{S}_h$ and the set of observations $\mathcal{O}_h$ are both Cartesian products of

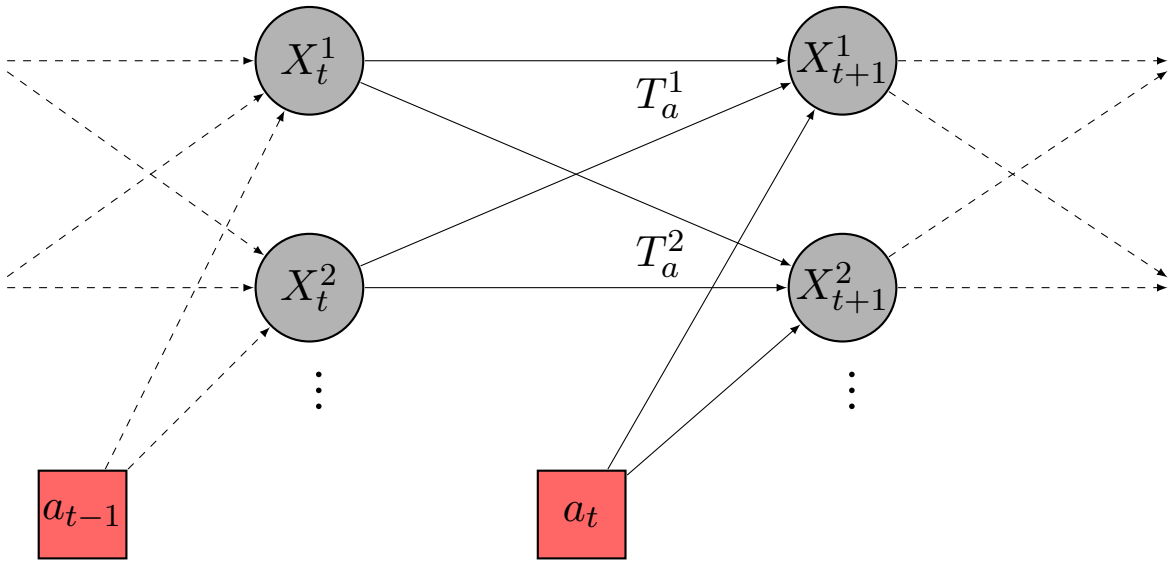


Figure III.2 – *Dynamic Bayesian Network of a factored (π-)MDP: in the possibilistic (resp. probabilistic) framework T_a^i is the transition possibility (resp. probability) distribution of next variable X_{t+1}^i conditional to the selected action $a \in \mathcal{A}$ and its parents $parents(X_{t+1}^i) \subseteq \{X_t^1, \dots, X_t^n\}$ (i.e. $parents(X_{t+1}^i)$ is a subset of the current state variables) where $n \geqslant 1$ is the number of variables describing the state space.*

finite sets described by variables. It boils down to solving a finite-space belief π-MDP whose state space is in the form of $\mathcal{S}_v^1 \times \cdots \times \mathcal{S}_v^m \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, where each of those spaces is finite. We will see in the next section how $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ can be further factorized thanks to the factorization of $\mathcal{S}_h$ and $\mathcal{O}_h$. While probabilistic belief factorization in [24, 131] is approximate, the one presented here relies on some assumptions but is exact. For now, as finite spaces of size K can be themselves factorized into $\lceil \log_2 K \rceil$ binary-variable spaces (see [75]), we can assume that we are reasoning about a factorized belief state π-MDP whose state space is denoted by $\mathcal{X}$ and fully described by variables $(X^1, \dots, X^n)$, with $n \in \mathbb{N}^*$ and $\forall i$, $X^i \in \{\top, \bot\}$: $\mathcal{X} = \{\top, \bot\}^n$.

Recall that Dynamic Bayesian Networks (DBNs) [42] already used in Section I.1 (for instance taking part of the influence diagrams in Figure I.2 and Figure I.3) and in the previous chapter (Figure II.1 illustrating the Mixed-Observable structure) are a useful graphical representation of studied processes. A DBN representing the structure of a factored π-MDP is depicted in Figure III.2: the state variables at a given time step $t \geqslant 0$ are denoted by $X_t = (X_t^i)_{i=1}^n$ (current variables), and $(X_{t+1}^i)_{i=1}^n$ are the state variables at step $t+1$ (next variables). In DBN semantics $parents(X_{t+1}^i)$ is the set of state variables on which the next state variable X_{t+1}^i "depends", *i.e.* a variable Y, represented by a node in the DBN, is in $parents(X_{t+1}^i)$ if and only if there is an arrow from Y to X_{t+1}^j. We assume that $parents(X_{t+1}^i) \subseteq \{X_t^1, \dots, X_t^n\}$, *i.e.* parents of the next state variable X_{t+1}^i are a part of the current state variables $\{X_t^1, \dots, X_t^n\}$: there cannot be any arrow between state variables of the same time step. Methods are discussed in the literature to circumvent this restrictive assumption [22].

As recalled in Section I.1.1, in probabilistic settings, the absence of an arrow in a Bayesian Network represents an independence assumption. Let us consider a set of variables $\{Y_1, \dots, Y_p\}$, with $p > 1$, and such that $\forall i \in \{1, \dots, p\}$, $Y_i \in \mathcal{Y}_i$ where $\mathcal{Y}_i$ is a finite set. For each $i \in \{1, \dots, p\}$, the set of the parents of variables Y_i can be denoted by $parents(Y_i) = \left\{Y_{pa_i(1)}, \dots, Y_{pa_i(p_i)}\right\} \subseteq \{Y_1, \dots, Y_p\}$, where $pa_i : \{1, \dots, p_i\} \to \{1, \dots, p\}$ is an increasing function. (note that $p_i \leqslant p$). Let us recall that $children(Y_i)$ is the set of all the variables $Y_j \in \{Y_1, \dots, Y_p\}$ such that there is an arrow from Y_i to Y_j. The set $descend(Y_i)$ is the set of the descendants of variable Y_i: $descend(Y_i)$ is defined as the smallest set containing $children(Y_i)$,

and such that $\forall Y_j \in descend(Y_i)$, $children(Y_j) \subset descend(Y_i)$. The set of non-descendants $nondescend(Y_i) = \left\{ Y_j \in \{ Y_1, \ldots, Y_p \} \;\middle|\; Y_j \notin \{ Y_i \} \cup descend(Y_i) \cup parents(Y_i) \right\}$ can be also denoted by $\left\{ Y_{nd_i(1)}, \ldots, Y_{nd_i(d_i)} \right\} \subset \{ Y_1, \ldots, Y_p \}$ where $nd_i : \{ 1, \ldots, d_i \} \to \{ 1, \ldots, n \}$ is an increasing function (and $d_i \leqslant p$). Hence, a DBN makes the assumption that Y_i is independent from its non-descendants conditional on its parents: $Y_i \perp\!\!\!\perp nondescend(Y_i) \mid parents(Y_i)$. In the probabilistic case, it can be written $\forall (y_1, \ldots, y_p) \in \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_p$, $\forall i \in \{ 1, \ldots p \}$,

$$\mathbb{P}\left(Y_i = y_i \;\middle|\; Y_{pa_i(1)} = y_{pa_i(1)}, \ldots, Y_{pa_i(p_i)} = y_{pa_i(p_i)}, Y_{nd_i(1)} = y_{nd_i(1)}, \ldots, Y_{nd_i(d_i)} = y_{nd_i(d_i)} \right)$$

$$= \mathbb{P}\left(Y_i = y_i \;\middle|\; Y_{pa_i(1)} = y_{pa_i(1)}, \ldots, Y_{pa_i(p_i)} = y_{pa_i(p_i)} \right). \tag{III.1}$$

This equation holds with a subset of the non-descendants $N \subset nondescend(Y_i)$: it suffices to consider the probability distribution over the set $nondescend(Y_i) \setminus N$, and compute the probabilistic expectation of each parts of the equation with respect to it. The upper part is the probability distribution conditional on the parents of Y_i and on C, and the lower part remains the same.

Using this formula, if we consider the state variables $(X_t^i)_{t \in \{ 0, \ldots, H \}}^{i \in \{ 1, \ldots, n \}}$ and actions $(a_t)_{t=0}^{H-1}$ of a factored MDP, denoting by X_t the variables $(X_t^i)_{i=1}^n$, we get from the DBN of Figure III.2 that $\forall t \in \{ 0, \ldots, H-1 \}$, $\forall (x_0, \ldots, x_{t+1}) \in \{ \top, \bot \}^{t+2}$, $\forall (a_0, \ldots, a_t) \in \mathcal{A}^{t+1}$,

$$\mathbb{P}\left(X_{t+1} = x_{t+1} \mid X_0 = x_0, \ldots, X_t = x_t, a_0, \ldots, a_{t-1} \right) = \mathbb{P}\left(X_{t+1} = x_{t+1} \mid X_t = x_t, a_t \right),$$

which is nothing more than the Markov property, justifying the definition of the transition probability distributions $\mathbf{p}(x'|x,a)$, $\forall (x, x') \in \{ \top, \bot \}^{2n}$ and $\forall a \in \mathcal{A}$.

Section I.2.2 explains that Qualitative Possibility Theory admits more than one independence definition: for instance, the non-interactivity independence (NI-independence, see Definition I.2.6) and the causal or min-based independence (M-independence, see Definition I.2.9) are two useful independence definitions for this chapter. If a DBN models M-dependences (causal dependences) between variables, then Equation III.1 holds replacing the probability measure $\mathbb{P}$ by the possibility measure Π:

$$\Pi\left(Y_i = y_i \;\middle|\; Y_{pa_i(1)} = y_{pa_i(1)}, \ldots, Y_{pa_i(p_i)} = y_{pa_i(p_i)}, Y_{nd_i(1)} = y_{nd_i(1)}, \ldots, Y_{nd_i(d_i)} = y_{nd_i(d_i)} \right)$$

$$= \Pi\left(Y_i = y_i \;\middle|\; Y_{pa_i(1)} = y_{pa_i(1)}, \ldots, Y_{pa_i(p_i)} = y_{pa_i(p_i)} \right), \tag{III.2}$$

which also holds considering only a subset of the descendants. The possibilistic Markov property is also deduced from the DBN of Figure III.2 the same result for π-MDPs: $\forall t \in \{ 0, \ldots, H-1 \}$, $\forall (x_0, \ldots, x_{t+1}) \in \{ \top, \bot \}^{t+2}$, $\forall (a_0, \ldots, a_t) \in \mathcal{A}^{t+1}$,

$$\Pi\left(X_{t+1} = x_{t+1} \mid X_0 = x_0, \ldots, X_t = x_t, a_0, \ldots, a_{t-1} \right) = \Pi\left(X_{t+1} = x_{t+1} \mid X_t = x_t, a_t \right),$$

which has been already defined in Section I.2.4, see Equation I.46.

Consider again variables $(Y)_{i=1}^n$. Let A and B be two disjoint subsets of $\{ 1, \ldots, n \}$. In order to simplify notations, if the probability $\mathbb{P}\left((Y_i = y_i)_{i \in A} \mid (Y_j = y_j)_{j \in B} \right)$ is considered $\forall (y_i)_{i \in A} \in \bigtimes_{i \in A} \mathcal{Y}_i$ and $\forall (y_i)_{i \in B} \in \bigtimes_{i \in B} \mathcal{Y}_i$, the probability can be denoted by $\mathbb{P}\left((Y_i)_{i \in A} \mid (Y_j)_{j \in B} \right)$. For instance, Equation III.1 can be rewritten

$$\mathbb{P}\left(Y_i \;\middle|\; parents(Y_i), nondescend(Y_i) \right) = \mathbb{P}\left(Y_i \;\middle|\; parents(Y_i) \right).$$

Let $(i_k)_{k=1}^{s}$ be an increasing sequence of indices with $s \leqslant p$, defining the set of variables $\{ Y_{i_1}, \ldots, Y_{i_k} \}$. The parents of variables $\{ Y_{i_1}, \ldots, Y_{i_k} \}$ are

$$parents\Big((Y_{i_k})_{k=1}^{s}\Big) := \bigcup_{k=1}^{s} parents(Y_{i_k}).$$

Suppose that the set of variables $\{ Y_{i_1}, \ldots, Y_{i_k} \}$ is such that $parents\Big((Y_{i_k})_{k=1}^{s}\Big) \cap \{ Y_{i_1}, \ldots, Y_{i_k} \} = \emptyset$, *i.e.* the parents of variables $\{ Y_{i_1}, \ldots, Y_{i_k} \}$ are not in $\{ Y_{i_1}, \ldots, Y_{i_k} \}$. Then, using the equation (III.1), we can write

$$\begin{aligned}
&\mathbb{P}\left((Y_{i_k})_{k=1}^{s} \;\middle|\; parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right) \\
&\qquad = \mathbb{P}\left(Y_{i_1} \;\middle|\; parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right) \cdot \mathbb{P}\left(Y_{i_2} \;\middle|\; Y_{i_1}, parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right) \cdot \\
&\qquad\qquad \ldots \cdot \mathbb{P}\left(Y_{i_s} \;\middle|\; Y_{i_1}, \ldots, Y_{i_{s-1}}, parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right) \\
&\qquad = \prod_{k=1}^{s} \mathbb{P}\Big(Y_{i_k} \;\Big|\; parents(Y_{i_k}) \Big).
\end{aligned}$$

As already noted, the variables of the factored MDP model depicted by Figure III.2, are such that $\forall t \geqslant 0, \forall i \in \{1, \ldots, n\}$, $parents(X_{t+1}^{i}) \subseteq \{ X_t^1, \ldots, X_t^n \}$: thus, using the previous general equation (with variables $(Y_i)_{i=1}^{p}$),

$$\begin{aligned}
\mathbb{P}\left(X_{t+1} \mid\; X_t, a_t \right) &= \mathbb{P}\left(X_{t+1}^1, \ldots, X_{t+1}^n \;\middle|\; X_t, a_t \right) \\
&= \prod_{i=1}^{n} \mathbb{P}\Big(X_{t+1}^{i} \;\Big|parents(X_{t+1}^{i}), a_t \Big).
\end{aligned}$$

It shows that the transition probability distributions $\mathbf{p}\left(x' \mid\; x, a \right)$ can be computed from the simpler distributions $\mathbb{P}\left(X_{t+1}^{i} \mid\; parents(X_{t+1}^{i}), a_t \right)$, $\forall i \in \{1, \ldots, n\}$. These transition distributions are denoted by $\mathbf{p}\left(X_i' \mid\; parents(X_i'), a \right)$, and by T_a^i in Figure III.2.

If we consider that the DBN involving variables $(Y_i)_{i=1}^{k}$ represents the M-dependences, Equation III.2 holds, and using the definition of the qualitative possibilistic conditioning (Definition I.2.7), we can also write

$$\begin{aligned}
&\Pi\left((Y_{i_k})_{k=1}^{s} \;\middle|\; parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right) \\
&\qquad = \min\left\{ \Pi\left(Y_{i_1} \;\middle|\; parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right), \Pi\left(Y_{i_2} \;\middle|\; Y_{i_1}, parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right), \right. \\
&\qquad\qquad \left. \ldots, \Pi\left(Y_{i_s} \;\middle|\; Y_{i_1}, \ldots, Y_{i_{s-1}}, parents\Big((Y_{i_k})_{k=1}^{s}\Big) \right) \right\} \\
&\qquad = \min_{k=1}^{s} \Pi\Big(Y_{i_k} \;\Big|\; parents(Y_{i_k}) \Big),
\end{aligned}$$

under the same assumption: $parents\Big((Y_{i_k})_{k=1}^{s}\Big) \cap \{ Y_{i_1}, \ldots, Y_{i_k} \} = \emptyset$. Thus, as we consider that the DBN in Figure III.2 represents the M-dependences,

$$\begin{aligned}
\Pi\left(X_{t+1} \mid\; X_t, a_t \right) &= \Pi\left(X_{t+1}^1, \ldots, X_{t+1}^n \;\middle|\; X_t, a_t \right) \\
&= \min_{i=1}^{n} \Pi\Big(X_{t+1}^{i} \;\Big|parents(X_{t+1}^{i}), a_t \Big).
\end{aligned}$$

The transition possibility distributions $\pi\left(x' \mid x, a\right)$ can be computed from the simpler distributions $\Pi\left(X_{t+1}^i \mid parents(X_{t+1}^i), a_t\right)$, $\forall i \in \{1, \ldots, n\}$, denoted by $\pi\left(X_{t+1}^i \mid parents(X_{t+1}^i), a\right)$, and by T_a^i in Figure III.2.

With the π-MOMDP notations, assumptions of the Bayesian Network in Figure III.2 allow us to compute the joint possibility transition as $\pi\left(s'_v, \beta'_h \mid s_v, \beta_h, a\right) = \pi\left(X' \mid X, a\right) = \min_{i=1}^{n} \pi\left(X'_i \mid parents(X'_i), a\right)$, where, given a time step t, primed variables are variables concerning time step $t+1$ (next variables), and non-primed variables are current variables (at time step t): for instance, X'_i is the notation for X_{t+1}^i, and X_i the one for X_t^i. Thus, a factored π-MOMDP can be defined with transition functions $T_a^i = \pi\left(X'_i \mid parents(X'_i), a\right)$ for each action a and variable X'_i (if transitions are assumed stationary).

Each transition function can be compactly encoded in an Algebraic Decision Diagram (ADD) [6]. An ADD, as illustrated in Figure III.3a, is a directed acyclic graph which compactly represents a real-valued function of binary variables, whose identical sub-graphs are merged and zero-valued leaves are not memorized. The following notations are used to make it explicit that we are working with symbolic functions encoded as ADDs:

- $\boxed{\min}\{f, g\}$ where f and g are 2 ADDs;
- $\boxed{\max}_{X_i} f = \boxed{\max}\left\{f^{X_i=0}, f^{X_i=1}\right\}$,

which can be easily computed because ADDs are constructed on the basis of the Shannon expansion: $f = \overline{X_i} \cdot f^{X_i=0} + X_i \cdot f^{X_i=1}$ where $f^{X_i=1}$ and $f^{X_i=0}$ are sub-ADDs representing the positive and negative Shannon cofactors (see Fig. III.3a).

The optimistic possibilistic update of dynamic programming, *i.e.* line 8 of the Value Iteration Algorithm 10 in the previous chapter, (or line 8 of the VI Algorithm 11 for π-MOMDPs) can be rewritten in a symbolic form, so that states are now globally updated at once instead of individually: denoting by $X = (X_1, \ldots, X_n)$ the current variable and $X' = (X'_1, \ldots, X'_n)$ the next one, the possibilistic Q-value of an action $a \in \mathcal{A}$ is $\overline{q^a} = \overline{q^a}(X) = \boxed{\max}_{X'} \boxed{\min}\left\{\pi\left(X' \mid X, a\right), \overline{U^*}(X')\right\}$. The computation of this ADD ($\overline{q^a}$) can be decomposed into independent computations using the following proposition:

Property III.2.1 (*Possibilistic regression of the Value Function*)

Consider the current value function $\overline{U^*} : \{\top, \bot\}^n \to \mathcal{L}$. For a given action $a \in \mathcal{A}$, let us define:

- $\overline{q_0^a} = \overline{U^*}(X'_1, \cdots, X'_n)$,
- $\overline{q_i^a} = \max_{X'_i \in \{\top, \bot\}} \min\left\{\pi\left(X'_i \mid parents(X'_i), a\right), \overline{q_{i-1}^a}\right\}$.

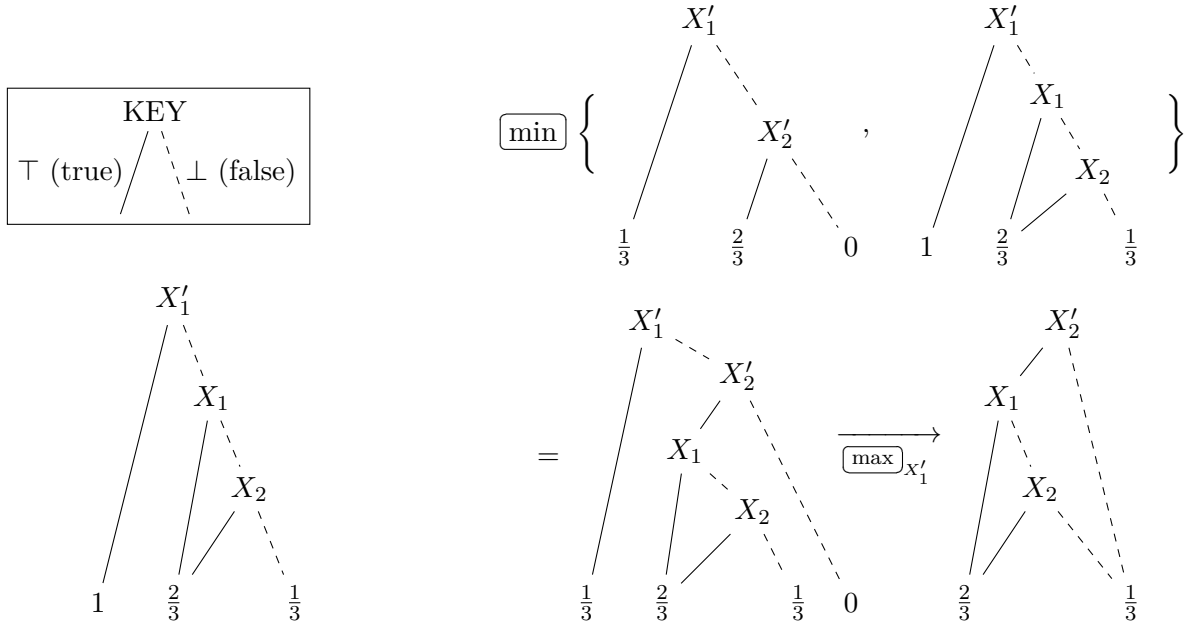
Then, the possibilistic Q-value of action a is: $\overline{q^a} = \overline{q_n^a}$, which depends on variables $X_1, \ldots, X_n$, and the next value function is $\overline{U^*}(X_1, \ldots, X_n) = \max_{a \in \mathcal{A}} \overline{q_n^a}(X_1, \ldots, X_n)$.

The proof is given in Annex D.1. Note that the same trick can be used to compute pessimistic value functions, using the equation (I.29) of Property I.2.1:

- $\underline{q_0^a} = \underline{U^*}(X'_1, \cdots, X'_n)$,
- $\underline{q_i^a} = \min_{X'_i \in \{\top, \bot\}} \max\left\{1 - \pi\left(X'_i \mid parents(X'_i), a\right), \underline{q_{i-1}^a}\right\}$,

and next value function is $\underline{U^*}(X_1, \ldots, X_n) = \max_{a \in \mathcal{A}} \underline{q_n^a}(X_1, \ldots, X_n)$.

The Q-value of action a, represented as an ADD, can be then iteratively regressed over successive post-action state variables $X'_i, 1 \leqslant i \leqslant n$. Figure III.3b illustrates the possibilistic regression of the Q-value of an action for the first state variable X_1 and leads to the intuition



(a) *ADD encoding T_1^a of Fig. III.2*

(b) *Symbolic regression of the current Q-value ADD combined with the transition ADD of Figure III.3a*

Figure III.3 – *Algebraic Decision Diagrams for PPUDD*

that ADDs should be far smaller in practice under possibilistic settings, since their leaves lie in $\mathcal{L}$ instead of $\mathbb{R}$, thus yielding more sub-graph simplifications.

Algorithm 12: PPUDD (infinite horizon resolution)

1 $\overline{U^*} \leftarrow 0$; $\overline{U^c} \leftarrow \Psi$; $\overline{\delta} \leftarrow \widehat{a}$;
2 **while** $\overline{U^*} \neq \overline{U^c}$ **do**
3 $\overline{U^*} \leftarrow \overline{U^c}$;
4 **for** $a \in \mathcal{A}$ **do**
5 $\overline{q^a} \leftarrow$ swap each X_i variable in $\overline{U^*}$ with X_i' ;
6 **for** $1 \leqslant i \leqslant n$ **do**
7 $\overline{q^a} \leftarrow \boxed{\min}\left\{\overline{q^a}, \pi\left(X_i' \,\middle|\, parents(X_i'), a\right)\right\}$;
8 $\overline{q^a} \leftarrow \boxed{\max}_{X_i'} \overline{q^a}$;
9 $\overline{U^c} \leftarrow \boxed{\max}\left\{\overline{q^a}, \overline{U^c}\right\}$;
10 update $\overline{\delta}$ to a where $\overline{q^a} = \overline{U^c}$ and $\overline{U^c} > \overline{U^*}$;
11 **return** $\overline{U^*}$, $\overline{\delta^*}$;

Algorithm 12 is a symbolic version of the π-MOMDP Value Iteration Algorithm (Algorithm 11 in the previous chapter), which relies on the regression scheme defined in Proposition III.2.1. Inspired by SPUDD [75], PPUDD means *Possibilistic Planning Using Decision Diagrams*. As for SPUDD, it needs to swap unprimed state variables to primed ones in the ADD encoding the current value function before computing the Q-value of an action a (see Line 5 of Algorithm 12 and Figure III.3b). This operation is required to differentiate the next state represented by primed variables from the current one when operating on ADDs. Lines 4-9 apply Proposition III.2.1 and correspond to Line 8 of Algorithm 10.

We mentioned at the beginning of this section that belief state space $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ could be described by $\lceil \log_2 K \rceil$ binary variables where $K = \#\mathcal{L}^{\#\mathcal{S}_h} - (\#\mathcal{L}-1)^{\#\mathcal{S}_h}$. However, this K can be very large so we propose in the next section a method to exploit the factorization of $\mathcal{S}_h$ and $\mathcal{O}_h$ in order to factorize $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ itself into small belief subvariables, which will decompose the possibilistic transition ADD into an aggregation of smaller ADDs. Note that PPUDD can solve π-MOMDPs even if this belief factorization is not feasible, but it will manipulate larger ADDs.

The previous chapter highlight that a pre-treatment is required to translate a π-MOMDP into a π-MDP whose state space is $\mathcal{X}$. We can then reason on the state space accessible to the agent $\mathcal{X} = S_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ and solve the π-MOMDP as a π-MDP. Next section links the structured properties of a π-MOMDP, concerning dependencies of original variables (visible, hidden and observation ones), to the factorization of the treated problem *i.e.* of the resulting π-MDP on $S_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ (concerning dependencies of visible and belief variables).

Finally, we note that we could have used *complete action diagrams* (CADs) introduced in [138], which directly encode the transition matrix of each action as a single ADD. On one hand, CADs are simpler to manipulate than a set of transition ADDs for each state variable, and can deal with correlated action effects. On the other hand, they require operating larger ADDs while preventing intermediate simplifications that are yet offered by reasoning about separate state variables as we do (Lines 4-9 of Algorithm 12) or SPUDD does [75].

III.3 π-MOMDP BELIEF FACTORIZATION

Factorizing the belief variable requires three structural assumptions on the π-MOMDP's DBN, which are illustrated by the Rocksample benchmark [136].

III.3.1 Motivating example.

A rover navigating in a $g \times g$ grid has to collect scientific samples from interesting ("good") rocks among R ones and then to reach the exit. It knows the locations of the R rocks $(x_i, y_i)_{i=1}^{R}$ but not which ones are actually of interest (called "good" rocks). However, sampling a rock is expensive: the rover is equipped with a noisy long-range sensor that can be used to determine if a rock is "good" or not ("bad"). When a rock is sampled, it becomes (or stays) "bad" (no more interesting). At the end of the mission, the rover has to reach the exit location at the right side of the grid:

- $\mathcal{S}_v$ consists of all the possible locations of the rover in addition to the exit ($\#\mathcal{S}_v = g^2+1$);
- $\mathcal{S}_h$ consists of all the possible natures of the rocks: $\mathcal{S}_h = \mathcal{S}_h^1 \times \ldots \times \mathcal{S}_h^R$, with $\forall 1 \leqslant i \leqslant R$, $\mathcal{S}_h^i = \{ good, bad \}$;
- $\mathcal{A}$ contains the (deterministic) moves in the 4 directions $(a_{north}, a_{east}, a_{south}, a_{west})$, checking rock i, (a_{check_i}) $\forall 1 \leqslant i \leqslant R$, and sampling the current rock, (a_{sample});
- $\mathcal{O} = \{ o_{good}, o_{bad} \}$ are the possible sensor's answers for the current rock.

The rationale behind observation dynamics is the following: the more the rover is close to the checked rock, the better it observes its nature. In the original probabilistic model, the probability of a correct observation equals $\frac{1}{2}\left(1 + e^{-c\sqrt{(x_r-x_i)^2+(y_r-y_i)^2}}\right)$ with $c > 0$. a constant (the smaller is c, the more effective is the sensor). The rover gets the reward $+10$ (resp. -10) for each good (resp. bad) sampled rock, and $+10$ when it reaches the exit.

In the possibilistic model, the observation function is approximated using a critical distance $d > 0$ beyond which checking a rock is uninformative: $\pi(o'_i \mid s'_i, a, s_v) = 1 \ \forall o'_i \in \mathcal{O}_i$. The possibility degree of erroneous observation becomes zero if the robot stands at the checked

rock, and least non zero possibility degree otherwise. Finally, as possibilistic semantics does not allow sums of rewards, an additional visible state variable $s_v^2 \in \{1, \ldots, R\}$ which counts the number of checked rocks is introduced. The qualitative dislike of sampling is modeled as $\Psi(s) = \frac{R+2-s_v^2}{R+2}$ if the location is terminal and zero otherwise. The location of the rover is finally denoted by $s_v^1 \in \mathcal{S}_v^1$ and the visible state is then $s_v = (s_v^1, s_v^2) \in \mathcal{S}_v^1 \times \mathcal{S}_v^2 = \mathcal{S}_v$.

Observations $\{o_{good}, o_{bad}\}$ for the current rock can be equivalently modeled as a Cartesian product of observations $\{o_{good_1}, o_{bad_1}\} \times \cdots \times \{o_{good_R}, o_{bad_R}\}$ for each rock. By using this equivalent modeling, state and observation spaces are both respectively factorized as $\mathcal{S}_v^1 \times \ldots \times \mathcal{S}_v^m \times \mathcal{S}_h^1 \times \ldots \times \mathcal{S}_h^l$ and $\mathcal{O} = \mathcal{O}^1 \times \ldots \times \mathcal{O}^l$, and we can now map each observation variable $O^j \in \mathcal{O}^j$ to its hidden state variable $S_h^j \in \mathcal{S}_h^j$. It allows us to reason about the DBN of Figure III.4, which expresses three important assumptions that will help us factorize the belief state itself:

1. all state variables $S_v^1, S_v^2, \ldots, S_v^m, S_h^1, S_h^2, \ldots, S_h^l$ are post-action independent variables, and the next visible variables do not depend on current hidden ones. Thus, there is no arrow between two state variables at the same time step, as $S_{v,t}^2$ and $S_{h,t}^1$, nor arrow from a current hidden variable to a next visible one, as $S_{h,t}^1$ and $S_{v,t+1}^1$;

2. a hidden variable does not depend on previous other hidden variables: the nature of a rock is independent from the previous nature of other rocks. For instance, there is no arrow from $S_{h,t}^1$ to $S_{h,t+1}^2$;

3. an observation variable is available for each hidden state variable, and depends on it. It does not depend on other hidden state variables nor current visible ones, but on previous visible state variables and action: for instance, there is no arrow between $S_{h,t+1}^1$ and O_{t+1}^2, nor between $S_{v,t+1}^1$ and O_{t+1}^1.

Each observation variable is indeed only related to the nature of the corresponding rock. The observation quality yet depends on the rover's location *i.e.* a current visible state variable, not allowed by the DBN: fortunately, as moves are deterministic, we avoid this issue considering observations depend on previous location and action.

III.3.2 Consequences of the factorization assumptions

In this section, we formally demonstrate how the three previous independence assumptions can be used to factorize $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ as the Cartesian product $\bigtimes_{j=1}^{l} \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$: indeed, the belief state β_h about the hidden states $s_h \in \mathcal{S}_h$ can be represented with marginal belief states $\beta_h^j \in \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ about hidden states $s^j \in \mathcal{S}_h^j$, $\forall j \in \{1, \ldots, l\}$.

To this end, we will use the *d-Separation criterion* [141] in order to show some independence between variables from the DBN. As explained in Section III.2, a DBN can be drawn from independence relations. Let us denote by $X \perp\!\!\!\perp Y \mid Z$ the assertion "X is independent from Y conditional on Z": recall that for a given definition of the used independence relation, e.g. probabilistic, non interactivity (NI), or minimum based (M, causal) independence, the DBN is drawn such that for each node (variable) X, $X \perp\!\!\!\perp nondescend(X) \mid parents(X)$. If the used independence relation obeys the *semi-graphoids* axioms [105, 144], the graphical criterion called d-Separation can be used to identify some independences between variables of the DBN.

This criterion is for instance used in probabilistic settings in [150]. Recall that the M-independence is not symetric (see Section I.2.2), and thus does not obey the axioms of semi-graphoids. However, the NI-independence leads to a semi-graphoid, as proved in [68].

Let us recall that M-independence implies NI-independence (see Theorem 10 of Section I.2.2). The DBN of Figure III.4 represents M-independences between variables: thus the DBN representing NI-independences (which is not drawn in this work) has potentially less arrows, *i.e.* assumes potentially more independences than the DBN representing the M-independences. Assuming that the DBN of Figure III.4 represents NI-independences is a relaxation *i.e.* we potentially forget some NI-independence assumptions by doing this assumption. However, all NI-independences proved using d-Separation criterion on the DBN are true.

First of all, the DBN of Figure III.4 representing the M-independence assumptions, some possibility distributions can be defined from the fact that each node is M-independent from its non-descendants conditional on its parents: given a time step $t \geqslant 0$, an action $a_t \in \mathcal{A}$, and a current state $s = (s_{v,t}, s_{h,t}) = (s^1_{v,t}, \ldots, s^m_{v,t}, s^1_{h,t}, \ldots, s^l_{h,t}) \in \mathcal{S}$,

- $\forall i \in \{1, \ldots, m\}$, the transition possibility distribution over the i^{th} visible state variable $s^i_{v,t+1} \in \mathcal{S}^i_v$:

$$\pi\left(s^i_{v,t+1} \;\middle|\; s_{v,t}, a_t\right) = \Pi\left(S^i_{v,t+1} = s^i_{v,t+1} \;\middle|\; S_{v,t} = s_{v,t}, a_t\right); \tag{III.3}$$

- $\forall j \in \{1, \ldots, l\}$, the transition possibility distribution over the j^{th} hidden state variable

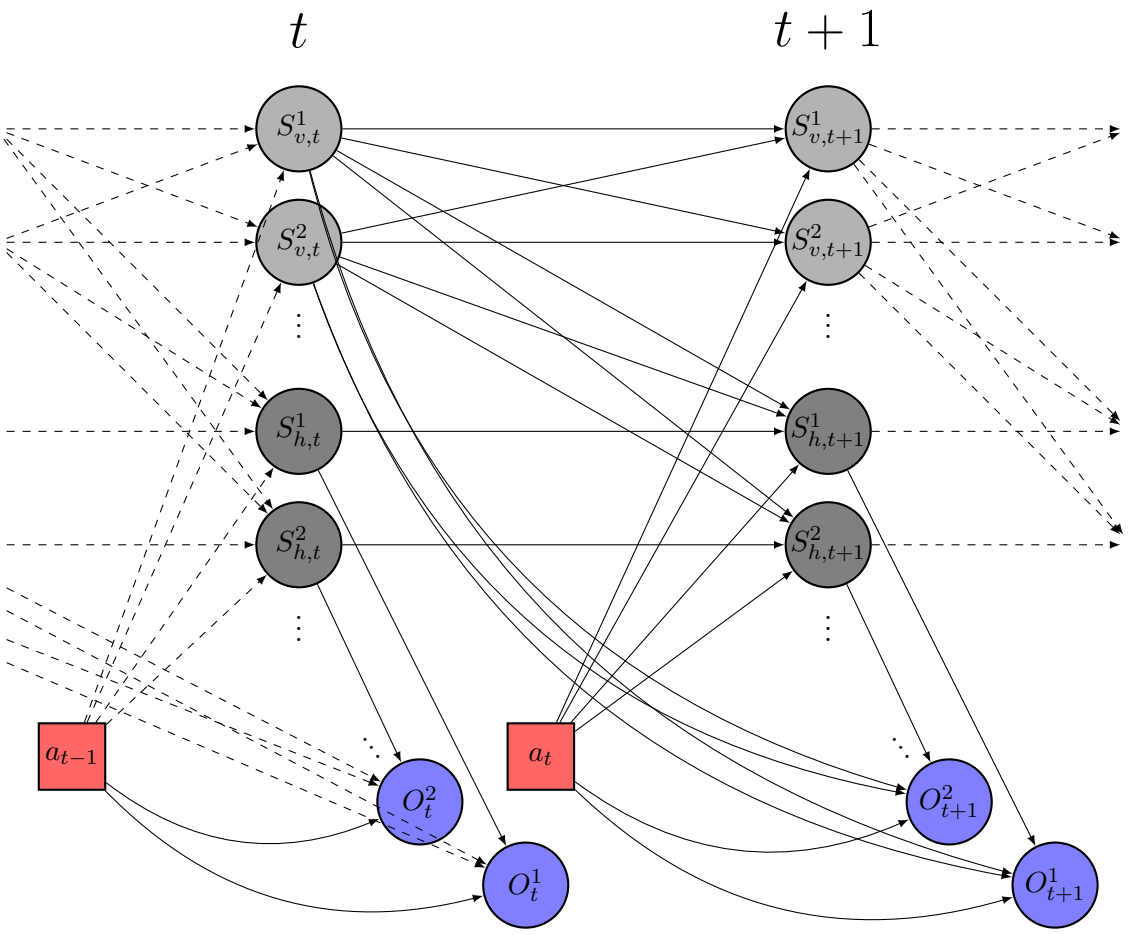


Figure III.4 – *DBN summing up independence assumptions of a π-MOMDP leading to marginal beliefs and a π-MDP with a factored transition function i.e. a factored belief π-MDP. Parents of a visible state variable are the previous visible state variables. Parents of a hidden state variable are the previous visible state variables and the corresponding previous hidden state variable. Finally, parents of an observation variable are the previous visible state variables, and the corresponding current hidden state variable.*

$s^j_{h,t+1} \in \mathcal{S}^i_h$:

$$\pi\left(s^j_{h,t+1} \,\middle|\, s_{v,t}, s_{h,t}, a_t\right) = \Pi\left(S^i_{h,t+1} = s^i_{h,t+1} \,\middle|\, S_{h,t} = s_{h,t}, a_t\right); \tag{III.4}$$

- $\forall j \in \{1, \ldots, l\}$, the observation possibility distribution over the j^{th} observation variable $o^j \in \mathcal{O}^i$:

$$\pi\left(o^j_{t+1} \,\middle|\, s_{v,t}, s_{h,t+1}, a_t\right) = \Pi\left(O^j_{t+1} = o^j_{t+1} \,\middle|\, S_{v,t} = s_{v,t}, S_{h,t+1} = s_{h,t+1}, a_t\right). \tag{III.5}$$

With these distributions, the dynamics of the process of a π-MOMDP respecting the assumptions of Figure III.4 is entirely defined.

Let us define the information i_t known by the agent at time step $t \geqslant 1$ when the model is a (π-)MOMDP: $i_0 = \{s_{v,0}\}$, and for each time step $t \geqslant 1$, $i_t = \{o_t, s_{v,t}, a_{t-1}, i_{t-1}\}$: the corresponding variable is denoted by I_t. The next theorem shows that the current belief can be decomposed into marginal belief states dependent on the current information i_t.

Theorem 24 (*Independence of hidden state variables and marginal belief states*)

Consider a π-MOMDP described by the DBN of Figure III.4. If initial hidden variables $S^1_{h,0}, \ldots, S^l_{h,0}$ are NI-independent, then at each time step $t > 0$ the belief over hidden states can be written as

$$\beta_{h,t} = \min_{j=1}^{l} \beta^j_{h,t}$$

with $\forall s \in \mathcal{S}^j_h$, $\beta^j_{h,t}(s) = \Pi\left(S^j_{h,t} = s \,\middle|\, I_t = i_t\right)$ the belief state concerning hidden states of the set $\mathcal{S}^j_h$.

The proof is given in Annex D.2.

Thanks to the previous theorem, the state space accessible to the agent can now be rewritten as $\mathcal{S}^1_v \times \ldots \times \mathcal{S}^m_v \times \Pi_{\mathcal{L}}^{\mathcal{S}^1_h} \times \cdots \times \Pi_{\mathcal{L}}^{\mathcal{S}^l_h}$ with $\Pi_{\mathcal{L}}^{\mathcal{S}^j_h} \subsetneq \mathcal{L}^{\mathcal{S}^j_h}$. The size of $\Pi_{\mathcal{L}}^{\mathcal{S}^j_h}$ is $\#\mathcal{L}^{\#\mathcal{S}^j_h} - (\#\mathcal{L}-1)^{\#\mathcal{S}^j_h}$ (see Equation I.60). If all state variables are binary, $\#\Pi_{\mathcal{L}}^{\mathcal{S}^j_h} = 2\#\mathcal{L} - 1$ for all $1 \leqslant j \leqslant l$, so that $\#\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h} = 2^m(2\#\mathcal{L} - 1)^l$: contrary to probabilistic settings, **hidden state variables and visible ones have a similar impact on the solving complexity**, i.e. both singly-exponential in the number of state variables. In the general case, by noting $\kappa = \max\{\max_{1\leqslant i\leqslant m} \#S_{v,i}, \max_{1\leqslant j\leqslant l} \#S_{h,j}\}$, there are $\mathcal{O}(\kappa^m(\#\mathcal{L})^{(\kappa-1)l})$ flattened belief states, which is indeed exponential in the arity of state variables too.

In Section I.2.5 about π-POMDP, the belief state variable at time step $t \geqslant 0$ is denoted by B^π_t, and its possible values are $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$. Now that $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ has been factorized, we can consider the *marginal belief state variables* $B^{\pi,j}_{h,t}$, $\forall j \in \{1, \ldots, l\}$, whose possible values are $\beta^j_h \in \Pi_{\mathcal{L}}^{\mathcal{S}^j_h}$, *i.e.* belief states concerning hidden states $s^j \in \mathcal{S}^j_h$. We now want to show that successive variables $S^1_{v,t}, \ldots, S^m_{v,t}, B^{\pi,1}_{h,t}, \ldots, B^{\pi,l}_{h,t}$ respect the assumptions of the DBN of Figure III.2, *i.e.* are independent post-action variables, as successive variables $X^1_t, \ldots, X^n_t$. This result is based on Lemma III.3.1, which shows how marginal belief state are actually updated.

Lemma III.3.1 (*Update of the marginal belief states*)

At time $t \geqslant 0$, if the system is in the visible state $s_{v,t} = (s^1_{v,t}, \ldots, s^m_{v,t}) \in \mathcal{S}_v$, in the belief state over the j^{th} hidden state $\beta^j_{h,t} \in \Pi_{\mathcal{L}}^{\mathcal{S}^j_h}$, and if the agent selects action $a_t \in \mathcal{A}$ and then gets observation $o^j_{t+1} \in \mathcal{O}^j$, the update of the belief state about hidden system states

$s^j \in \mathcal{S}_h^j$ *is:*

$$\beta_{h,t+1}^j(s_{t+1}^j) = \begin{cases} 1 & \text{if } \pi\left(o_{t+1}^j, s_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right) = \pi\left(o_{t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right) \\ \pi\left(o_{t+1}^j, s_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right) & \text{otherwise.} \end{cases} \tag{III.6}$$

where $\pi\left(o_{t+1}^j, s_{h,t+1}^j \;\middle|\; s_v, \beta_h^j, a\right)$ *is the notation for*

$$\max_{s^j \in \mathcal{S}_h^j} \min\left\{\pi\left(o_{t+1}^j \;\middle|\; s_v, s_{t+1}^j, a\right), \pi\left(s_{t+1}^j \;\middle|\; s_v, s^j, a\right), \beta_h^j(s^j)\right\},$$

using distributions (III.4) and (III.5), and

$$\pi\left(o_{t+1}^j \;\middle|\; s_v, \beta_h^j, a\right) = \max_{s_{t+1}^j \in \mathcal{S}_h^j} \pi\left(o_{t+1}^j, s_{t+1}^j \;\middle|\; s_v, \beta_h^j, a\right).$$

The proof is given in Annex D.3. The associated **belief update function** is ν^j:

$$\beta_{h,t+1}^j = \nu^j(s_{v,t}, \beta_{h,t}^j, a_t, o_{t+1}^j),$$

which can be denoted by

$$\beta_{h,t+1}^j(s_{h,t+1}^j) \propto^\pi \pi\left(o_{t+1}^j, s_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)$$

as it consists of the possibilistic normalization of the joint possibility distribution over the j^{th} hidden state variable and the j^{th} observation.

Hence, the possibility degree that the marginal belief state variables $B_{h,t+1}^{\pi,j}$ is $\beta_{h,t+1}^j \in \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ conditional on $B_{h,t}^{\pi,j} = \beta_{h,t}^j$ and the action $a_t \in \mathcal{A}$, can be computed:

$$\Pi\left(B_{h,t+1}^{\pi,j} = \beta_{h,t+1}^j \;\middle|\; S_{v,t} = s_{v,t}, B_{h,t}^{\pi,j} = \beta_{h,t}^j, a_t\right) = \max_{\substack{o^j \in \mathcal{O}^j \text{ s.t.} \\ \nu^j(s_{v,t}, \beta_{h,t}^j, a_t, o^j) = \beta_{h,t+1}^j}} \pi\left(o^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right) \tag{III.7}$$

defining the transition possibility distribution of marginal belief states $\pi\left(\beta_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)$.

Finally, Theorem 25 relies on Lemma III.3.1 to ensure independence of all post-action variables of the belief π-MDP resulting from the factorization, conditional on the current state: this allows us to write the possibilistic transition function of the belief-state π-MDP in a factorized form:

Theorem 25 (*Factored expression of the transition possibility distribution*)

If independence assumptions of a π-MOMDP are described by the DBN of Figure III.4, then $\forall \beta_{h,t} = (\beta_{h,t}^1, \ldots, \beta_{h,t}^l) \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}, \beta_{h,t+1} = (\beta_{h,t+1}^1, \ldots, \beta_{h,t+1}^l) \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, $\forall (s_{v,t}, s_{v,t+1}) \in (\mathcal{S}_v)^2$, $\forall a_t \in \mathcal{A}$,

$$\pi\left(s_{v,t+1}, \beta_{h,t+1} \;\middle|\; s_{v,t}, \beta_{h,t}, a\right)$$

$$= \min\left\{\min_{i=1}^{m} \pi\left(s_{v,t+1}^i \;\middle|\; s_{v,t}, a_t\right), \min_{j=1}^{l} \pi\left(\beta_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)\right\},$$

where the transition possibility distributions of visible state variables is given in the equation (III.3) and the one of marginal belief state variables in the equation (III.7).

The proof is given in Annex D.4. Using this result, such a factorized expression of the transition possibility distribution allows to compute the value function with $n = m + l$ stages, as

described in the previous section: the π-MOMDP is indeed a factored π-MDP since variables $(S_v^1,\ldots,S_v^m,B_h^{\pi,1},\ldots,B_h^{\pi,l})$, can play the role of variables $X_1,\ldots,X_n$ in Algorithm 12.

Note that Probability Theory does not make any difference between causal independence (M-independence in Possibility Theory, see Definition I.2.8) and decompositional independence (NI-independence in Possibility Theory, see Definition I.2.6). Moreover, probabilistic independence relation obey the axioms of semi-graphoids: so previous independence results due to d-Separation are also true in probabilistic settings. If independence assumptions between variables of a probabilistic MOMDP [100, 3] are described by the DBN of Figure III.4, then a similar factorization result can be deduced:

Theorem 26 (*Factored expression of the transition probability distribution*)

Consider a probabilistic MOMDP with independence assumptions described by the DBN of Figure III.4. The following probability distributions define entirely dynamics of variables:

- *the transition probability distributions of visible state variables:* $\forall i \in \{1,\ldots,m\}$,
$$\boldsymbol{p}\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right) = \mathbb{P}\left(S_{v,t+1}^i = s_{v,t+1}^i \,\middle|\, S_{v,t} = s_{v,t}, a_t\right);$$
- *the transition probability distributions of hidden state variables:* $\forall j \in \{1,\ldots,l\}$,
$$\boldsymbol{p}\left(s_{h,t+1}^j \,\middle|\, s_{v,t}, s_{h,t}^j, a_t\right) = \mathbb{P}\left(S_{h,t+1}^j = s_{h,t+1}^j \,\middle|\, S_{v,t} = s_{v,t}, S_{h,t}^j = s_{h,t}^j, a_t\right);$$
- *the observation probability distributions:* $\forall j \in \{1,\ldots,l\}$,
$$\boldsymbol{p}\left(o_{t+1}^j \,\middle|\, s_{v,t}, s_{h,t+1}^j, a_t\right) = \mathbb{P}\left(O_{t+1}^j = o_{t+1}^j \,\middle|\, S_{v,t} = s_{v,t}, S_{h,t+1}^j = s_{h,t+1}^j, a_t\right).$$

Let us define probabilistic marginal belief states: $\forall j \in \{1,\ldots,l\}$, $\forall s \in \mathcal{S}_h^j$,
$$b_{h,t}^j(s) = \mathbb{P}\left(S_{h,t}^j = s \,\middle|\, I_t = i_t\right).$$

Given such a marginal belief state $b_{h,t}^j$ *and a visible state* $s_{v,t} \in \mathcal{S}_v$*, if the action* $a_t \in \mathcal{A}$ *is selected and the observation* $o_{t+1}^j \in \mathcal{O}^j$ *is received, then the next belief state about the* j^{th} *hidden state variable is* $\forall s_{h,t+1}^j \in \mathcal{S}_h^j$,
$$b_{h,t+1}^j(s_{h,t+1}^j) \propto \boldsymbol{p}\left(o_{t+1}^j \,\middle|\, s_{v,t}, s_{h,t+1}^j, a_t\right) \cdot \sum_{s_{h,t}^j \in \mathcal{S}_h^j} \boldsymbol{p}\left(s_{h,t+1}^j \,\middle|\, s_{v,t}, s_{h,t}^j, a_t\right) \cdot b(s_{h,t}^j),$$
denoted by $b_{h,t+1}^j(s_{h,t+1}^j) = u^j(s_{v,t}, b_{h,t}^j, a_t, o_{t+1}^j)$.

The transition probability distribution can be written as follows: $\forall b_{h,t} = (b_{h,t}^1,\ldots,b_{h,t}^l) \in \mathbb{P}^{\mathcal{S}_h}, b_{h,t+1} = (b_{h,t+1}^1,\ldots,b_{h,t+1}^l) \in \mathbb{P}^{\mathcal{S}_h}$, $\forall (s_{v,t}, s_{v,t+1}) \in (\mathcal{S}_v)^2$, $\forall a_t \in \mathcal{A}$,
$$\boldsymbol{p}\left(s_{v,t+1}, b_{h,t+1} \,\middle|\, s_{v,t}, b_{h,t}, a\right) = \prod_{i=1}^{m} \boldsymbol{p}\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right) \cdot \prod_{j=1}^{l} \boldsymbol{p}\left(b_{h,t+1}^j \,\middle|\, s_{v,t}, b_{h,t}^j, a_t\right),$$
where
$$\boldsymbol{p}\left(b_{h,t+1}^j \,\middle|\, s_{v,t}, b_{h,t}^j, a_t\right) = \sum_{\substack{o^j \in \mathcal{O}^j \text{ s.t.} \\ u^j(s_{v,t}, \beta_{h,t}^j, a_t, o^j) = \beta_{h,t+1}^j}} \boldsymbol{p}\left(o_{t+1}^j \,\middle|\, b_{h,t}^j, a_t\right),$$
and
$$\boldsymbol{p}\left(o_{t+1}^j \,\middle|\, b_{h,t}^j, a_t\right) = \sum_{s_{h,t+1}^j \in \mathcal{S}_h^j} \boldsymbol{p}\left(o_{t+1}^j \,\middle|\, s_{v,t}, s_{h,t+1}^j, a_t\right) \cdot \sum_{s_{h,t}^j \in \mathcal{S}_h^j} \boldsymbol{p}\left(s_{h,t+1}^j \,\middle|\, s_{v,t}, s_{h,t}^j, a_t\right) \cdot b(s_{h,t}^j).$$

The MDP built from such a probabilistic MOMDP is thus a factored MDP.

Previous theorems allow to express the transition distribution of the (π-)MDP resulting from a (π-)MOMDP with distributions which concern less variables. The value function update is then divided into $n = m + l$ stages in the possibilistic case, as depicted by the *for* loop of Algorithm 12. Qualitative possibilistic MOMDPs can however also be solved using ADDs even if the independence assumptions do not hold: in this case, one global transition distribution, encoded as a big ADD concerning all variables $(S_v^1, \ldots, S_v^m, B_h^\pi)$ is used, and the number of potential values $\beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ of the global belief state variable B_h^π increases exponentially with the number of hidden states: $\#\Pi_{\mathcal{L}}^{\mathcal{S}_h} = (\#\mathcal{L})^{\#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_h}$ (see Equation I.60). Nevertheless, if the factorization of the transition distribution is possible, handled ADDs have less nodes and computations should be faster. These results are used in the next section to compute more efficiently optimal strategies of π-MOMDPs.

III.4 Experimental results

The main expected advantages of using factored π-(MO)MDPs over their probabilistic counterparts are:

1. values of ADDs are in the finite scale $\mathcal{L}$ rather than $\mathbb{R}$, so that the number of their leaves is at most $\#\mathcal{L} \ll 2^n$ (probabilistic models' ADDs can have up to 2^n leaves, where n is the number of variables involved in the ADD);

2. π-MOMDPs boil down to factored *finite*-state belief π-MDPs that can be solved by PPUDD assuming some independence assumptions on the underlying DBNs;

3. π-MOMDPs are in the same complexity class as π-MDPs *if all hidden state variables are binary* (in probabilistic models, partially-observable problems are always in a higher complexity class).

Of course, we have to pay a price: namely, possibilistic models can be seen as approximations of probabilistic ones (except if probabilities in the model are not precisely known and uncertainty of the problem is better described in a qualitative form). Yet, many state-of-the-art probabilistic algorithms are approximate, e.g. MDP solver PROST [79] (based on UCT algorithm [80]) and POMDP solvers described in Section I.1.11. Our PPUDD algorithm, however, is exact.

In the case of the infinite horizon probabilistic MOMDPs, it is sufficient to look for a stationary strategy *i.e.* an optimal strategy can be defined as a function $d^* : \mathcal{S}_v \times \mathbb{P}^{\mathcal{S}_h} \to \mathcal{A}$ maximizing the value function given an initial visible state $s_{v,0} \in \mathcal{S}_v$ and an initial belief state $b_{h,0} \in \mathbb{P}^{\mathcal{S}_h}$:

$$V(s_{v,0}, b_{h,0}, d^*) = \sup_{d:\ \mathcal{S}_v \times \mathbb{P}^{\mathcal{S}_h} \to \mathcal{A}} V(s_{v,0}, b_{h,0}, d),$$

where $V(s_{v,0}, b_{h,0}, d) = \mathbb{E}\Big[\sum_{t \geqslant 0} \gamma^t \cdot r\Big(S_t, d(S_{v,t}, B_{h,t})\Big) \;\Big|\; S_{v,0} = s_{v,0}, B_{h,0} = b_{h,0}\Big]$, and the action defining probabilistic dynamics at time step t is $d(S_{v,t}, B_{h,t})$ if the strategy is d (see Section I.1.6 and [100, 3]).

Consider now qualitative possibilistic MOMDPs [49]: first, the optimistic criterion of an infinite horizon π-MDP can be written

$$\max_{t \geqslant 0} \max_{\mathcal{T} \in \mathcal{T}_t} \min \left\{ \pi\left(\mathcal{T} \mid\ s_0, (\delta)\right), \Psi(s_t) \right\},$$

using Equation II.18 in the previous chapter and Lemma C.1 of Annex C. In this formula, the set of t-length trajectories $(s_1, \ldots, s_t) \in \mathcal{S}^t$ is denoted by $\mathcal{T}_t$ and recall that $\Psi : \mathcal{S} \to \mathcal{L}$ is the

terminal preference function. Let us now denote by x_t the couple of visible and belief states $(s_{v,t}, \beta_{h,t}) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, available to the agent at time step $t \geqslant 0$: $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ is denoted by $\mathcal{X}$. Theorem 22 assures that at least one stationary strategy is optimal if a "stay" action, which maintains the system in the same state and denoted by $\widehat{a} \in \mathcal{A}$, is available: recall that this action is only used in some goal states in the optimal strategy. The qualitative infinite horizon criterion for a mixed optimistic-pessimistic π-MOMDP (see the sections II.1.1 and II.2) can thus be written

$$\overline{U}\Big(x_0, (\delta)\Big) = \max_{t \geqslant 0} \max_{\mathcal{T} \in \mathcal{T}_t} \min \left\{ \pi\Big(\mathcal{T} \;\Big|\; x_0, (\delta)\Big), \underline{\Psi}(x_t) \right\}, \tag{III.8}$$

where $\mathcal{T}_t = (x_1, \ldots, x_t)$ is a trajectory of couples $x_i = (s_{v,i}, \beta_{h,i})$, $\mathcal{T}_t$ the set of such trajectories, and

$$\pi\Big(\mathcal{T} \;\Big|\; x_0, (\delta)\Big) = \min_{i=0}^{t-1} \pi\Big(x_{i+1} \;\Big|\; x_i, \delta(x_i)\Big)$$

the possibility degree of such trajectories, with the transition possibility distribution $\pi\Big(x_{i+1}\Big|x_i, \delta(x_i)\Big)$ defined in Section II.2. Recall also that the pessimistic terminal preference is denoted by $\underline{\Psi}(x) = \min_{s_h \in \mathcal{S}_h} \max\{\Psi(s_v, s_h), 1 - \beta_h(s_h)\}$ with $x = (s_v, \beta_h)$. An optimal strategy δ^* is thus such that

$$\overline{U}(s_{v,0}, \beta_0, \delta^*) = \sup_{\delta:\ \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h} \to \mathcal{A}} \overline{U}(s_{v,0}, \beta_0, \delta),$$

see Section II.3.2 of the previous chapter.

In the probabilistic case, as the belief state $B_{h,t}$ is a deterministic function of the current available information I_t, a strategy of an infinite probabilistic MOMDP can be defined as a sequence of functions based on the current information (and thus non stationary): $(d_t)_{t \geqslant 0}$, with d_t a function of the visible state s_v and the information $i_t = \{s_{v,0}, a_0, o_1, \ldots, s_{v,t-1}, a_{t-1}, o_t\}$. In this way, the strategy only depends on the variables of the initial problem (and not on the current belief state). As in the probabilistic case, the qualitative possibilistic belief state $B_{h,t}^{\pi}$ is fully specified by the current available information I_t. The strategy of an infinite horizon π-MOMDP can thus be defined as a sequence of functions based on the current information: $(\delta_t)_{t \geqslant 0}$, with δ_t a function of the visible state $s_{v,t}$ and the information i_t. This being so, the strategy does not depend on the current possibilistic belief state, but on the variables of the π-MOMDP.

At execution, the agent knows the successive visible system states and the available information: taking into account the previous paragraph, he/she may thus use the optimal strategy provided by either the probabilistic model using d^*, or the possibilistic one using δ^*. Obviously, if the performance of the strategy is measured using the probabilistic criterion V, *i.e.* evaluating the average of rewards obtained using the given strategy during many trials, the optimal possibilistic strategy δ^* is less efficient than d^*:

$$V(s_v, b_h, d^*) \geqslant V(s_v, b_h, \delta^*).$$

However, as dimensions of considered problems may be high enough to make the probabilistic computations intractable or to make probabilistic solvers require too many computation time resources, strategies returned by probabilistic solvers are approximations of d^* which may be less efficient than δ^* even in the probabilistic sense. Note also that

$$\overline{U}(s_v, \beta_h, \delta^*) \geqslant \overline{U}(s_v, \beta_h, d^*),$$

but this criterion will not be used during the experiments as it is not a standard performance measure when planning under uncertainty. Qualitative criteria can however be good performance measure of strategies in practice if the probabilities of the model are in fact given arbitrarily from a qualitative evaluation of variables dynamics.

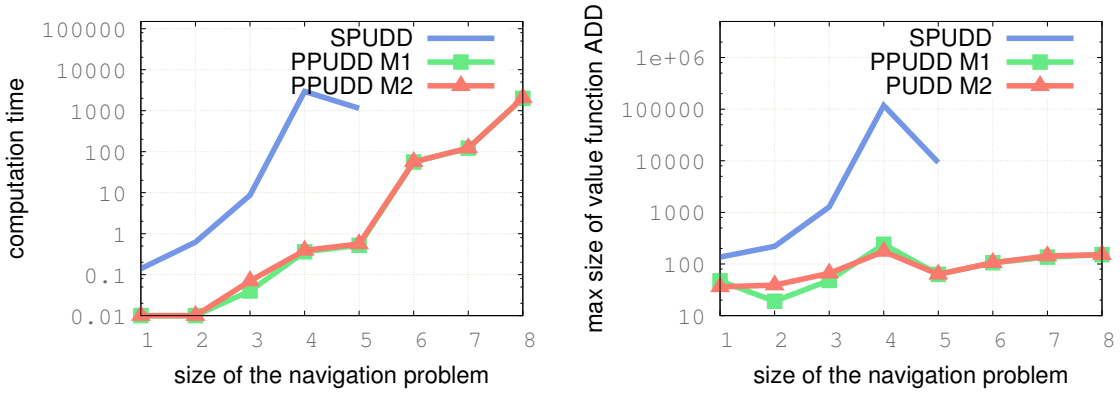
In this section, we compare our possibilistic approach against probabilistic solvers in order to answer the following question: what is the efficiency/quality tradeoff achieved by reasoning about an approximate model (π-MOMDP) but with an exact efficient algorithm (PPUDD)? Despite radically different methods, possibilistic strategies and probabilistic ones are both able to return an action for each possible visible state variable and current information: they are thus directly comparable and statistically evaluated under identical settings *i.e.* using transition and reward functions defined by the probabilistic model (criterion V).

It has been shown in [122] that the optimistic π-MDP criterion (see Equation I.48 of Section I.2.4) leads to better strategies than the pessimistic one (see Equation I.51) when the goal is to approximate an optimal strategy of a probabilistic fully observable MDP. Moreover, we proposed an algorithm for infinite horizon π-MDPs which has been proved to produce an optimal strategy with the optimistic criterion, see Section II.3 of previous chapter: Algorithm 10. This algorithm is also devoted to problems without intermediate preferences. The optimistic criterion, as well as the case of terminal preference only (see Definition I.2.15), are thus preferred in the following experimentations. In the case of mixed-observability, the mixed optimistic-pessimistic criterion (see Definition II.1.3) is a good choice as the π-MOMDP boils down to an optimistic π-MDP. Moreover, with this criterion, the more a belief state is specific, the higher is its preference, unlike the purely optimistic ones.

III.4.1 Robotic missions

We first assessed PPUDD performances on totally observable factored problems since PPUDD is also the first algorithm to solve factored π-MDPs (by inclusion in π-MOMDPs). To this end, we compared PPUDD against SPUDD on the *Navigation problem* used in the International Probabilistic Planning Competition 2011 [127]. In this domain, a robot navigates on a grid where it must reach some goal location most reliably. It can apply actions going north, east, south, west and stay: all these actions cost 1 except on the goal. When moving, it can suddenly disappear with some probability defined as a Bernoulli distribution, so that a good policy tries to reach the goal by avoiding situations where it may disappear. This probabilistic model is approximated by two possibilistic ones where: the preference of reaching the goal is 1; in the first model (M1) the highest probability of each Bernoulli distribution is replaced by 1 (for possibility normalization reasons) and the same value for the lowest probability is kept; for the second model (M2), the probability of disappearing is replaced by 1 and the other one is kept. Figure III.5a shows that SPUDD runs out of memory from the 6^{th} problem, and PPUDD computation's time outperforms SPUDD by many orders of magnitude for the two models. Intuitively, this result comes from the fact that PPUDD's ADDs should be smaller because their leaf values are in the finite scale $\mathcal{L}$ rather than $\mathbb{R}$, which is indeed demonstrated in Figure III.5b. Performances were evaluated with two relevant criteria: frequency of runs where the policy reaches the goal (see Figure III.5c), and average length of execution runs that reach the goal (see Figure III.5d), that are both functions of the problem's instance. As expected, model (M2) is more cautious than model (M1) and gets a better reached goal frequency (similar to SPUDD's one for the instances it can solve). The later is more optimistic and gets a better average length of execution runs than model (M2) due to its dangerous behavior. For fairness reasons, we also compared ourselves against APRICODD [138], which is an approximate algorithm for factored MDPs: however parameters impacting the approximation are hard to tune (either huge computation times, or zero qualities) and it is largely outperformed by PPUDD in both time and quality whatever the parameters (curves are not shown since uninformative).

Finally, we compared PPUDD on the *Rocksample problem* (RS), described in Section



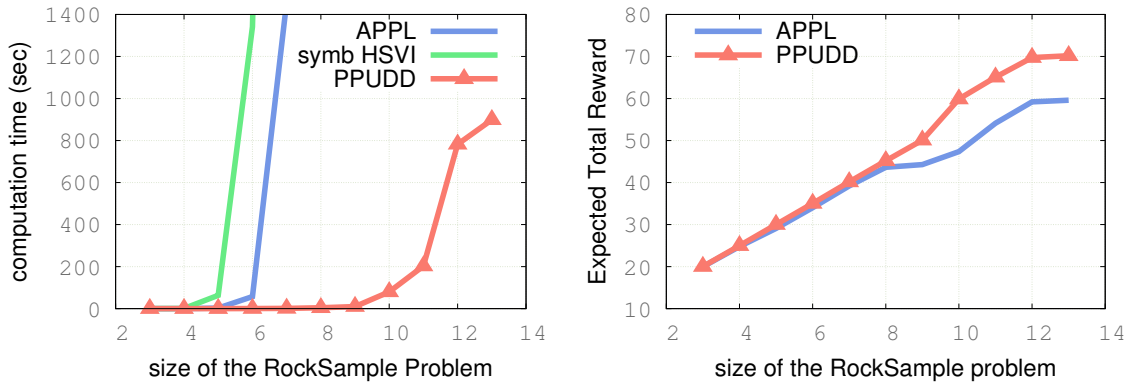
(a) *Computation time (sec)*

(b) *Size of ADD value function*

(c) *Reached goal frequency*

(d) *Time to reach the goal*

Figure III.5 – *PPUDD vs. SPUDD on the Navigation problem: the x-axis represents indexes of problem instances, increasing with the problem sizes.*



(a) *Computation time (sec)* (b) *Expected total reward*

Figure III.6 – *PPUDD vs. APPL and symb HSVI on the RockSample problem: the x-axis represents indexes of problem instances, increasing with the problem sizes.*

III.3.1, against a recent probabilistic MOMDP planner, APPL [100], and a POMDP planner using ADDs, symbolic HSVI [133]. Both algorithms are approximate and anytime, so we decided to stop them when they reached a precision of 1. Figure III.6a, where problem instances increase with grid size and number of rocks, shows that APPL runs out of memory at the 8^{th} problem instance, symbolic HSVI at the 7^{th} one, while PPUDD outperforms them by many orders of magnitude. Instead of precision, computation time of APPL can be fixed at PPUDD's computation time in order to compare their expected total rewards (using probabilistic model's rewards) after they consumed the same CPU time. Surprisingly, Figure III.6b shows that rewards gathered are higher with PPUDD than with APPL. The reason is that APPL is in fact an approximate probabilistic planner, which shows that our approach consisting in exactly solving an approximate model can outperform algorithms that approximately solve an exact model. Moreover, exact POMDP planners are unable to scale to problems of the size of the RockSample ones. Finally, it is worth noting that probabilities of the observation model, which represent uncertainties of sensor outputs, may be difficult to precisely know in practice, in which case possibilistic models may be more physically accurate. In fact for this example the policy produced by PPUDD is the best to get all possible rewards: this is essentially because the rover can be sure of a rock's nature checking when it is on it.

These results assured us that it was not unreasonable to present PPUDD in the International Probabilistic Planning Competition 2014, even though the computation of strategies for probabilistic problems is not the initial vocation of this solver. The next section describes the competition context as well as the presented versions of PPUDD, and discusses the results of the different competitor solvers.

III.4.2 International Probabilistic Planning Competition 2014

The fully observable track of the International Probabilistic Planning Competition (IPPC) allows to fairly compare performances of MDP solvers. The competitors' solvers have to compute strategies for some problems which are not known in advance. Given one of these problems, solvers have a limited amount of time to send actions to the competition server which simulates the evolution of the system state: successive states are sampled by the competition server using the transition probability distributions of the MDP defining the problem, and sent to a given competitor's solver. For each system state received, the solver has to send back the action it has computed. These data exchanges are conducted during few trials of finite horizon and the

score of the solver for the considered problem is the average (over trials) of the undiscounted and finite sum of rewards along the trajectory generated by the trial.

Materials about this competition are available at the official web page of the competition `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`. Problems are grouped in *domains*, which are MDPs whose a finite number of parameters are undefined: the problem, or MDP, used in practice during the competition is an *instance* of a domain, *i.e.* a domain whose parameters have been set. In this competition, 8 domains have been proposed, called respectively *Academic advising*, *Crossing traffic*, *Elevators*, *Skill teaching*, *Tamarisk*, *Traffic*, *Triangle tireworld* and *Wildfire*. Three possible encodings of the instances of these domains are proposed, *i.e.* three different languages can be used to describe the instances: the first is the *Planning Domain Definition Language* (PPDDL, [151]); the second is a *LISP*-like language introduced with symbolic algorithms such as SPUDD which defines explicitely transition probability distributions and reward function as ADDs (see `http://users.cecs.anu.edu.au/~ssanner/IPPC_2011/`); finally, the third is the *Relational Dynamic Influence Diagram Language* (RDDL, [126]) which is simpler and more expressive than the previous ones. The competition consists in evaluating the solver over 10 instances per domain with 30 runs per instance and 18 minutes per instance: it takes 24 hours in total.

In order to ensure that everyone has the same computational power, each competitor solver is set up in a remote server whose RAM is 7.5Gb with 2 cores.The client and server for the competition are available in the open source *RDDLSim* software, which is available online at `http://code.google.com/p/rddlsim/`. Four solvers have been proposed for this competition:

- *PROST* [79], based on *Upper Confidence bound applied to Trees* (UCT, [80]) and using directly RDDL encoding;
- *GOURMAND* [83, 82], based on *Labeled Real Time Dynamic Programming* (LRTDP, [14]) using PPDDL encoding;
- *symbolic LRTDP*, using ADDs and LISP-like encoding [45];
- our algorithm PPUDD, using LISP-like encoding too.

As the score given to solvers only depends on the 40 first stages of the process, the presented version of PPUDD consists of the Algorithm 12 with the "while condition" $\overline{U^*} \neq \overline{U^c}$ at line 2 replaced by the condition "iteration $\leqslant$ 40". It also incrementally augments the planning horizon while maintaining a mask stored in form of a Binary Decision Diagram (BDD, *i.e.* an ADD with leaves in $\{0, 1\}$) representing the states reachable from the initial state: the computation of the current value function is then restricted to the reachable states only. While PPUDD is an offline algorithm, we proposed also *AnyTime PPUDD* (ATPPUDD) which is an anytime version which learns computation times of Bellman backups while dispatching the computational effort accordingly over the remaining planning horizon much like GOURMAND does in the probabilistic world (see [83]).

When encoded with the LISP-like format, problems of the competition, *i.e.* instances of each domains, are described as factored MDPs with boolean system state variables: for each action $a \in \mathcal{A}$ and for each next boolean system state variable X'_i, one ADD representing the corresponding transition probability distribution $\mathbf{p}\Big(X'_i \;\Big|\; parents(X'_i), a\Big)$ is given. In order to define the π-MDP which will be solved by PPUDD, we simply normalize these distributions in the possibilistic sense: we set to 1 the possibility degree of an assignment of X'_i when its probability value is maximal, and to the probability value otherwise. For instance, for a given assignment of the previous variables $parents(X'_i)$, if the probability value of the assignment (or event) $X'_i = 1$ is 0.7 (and thus probability 0.3 that $X'_i = 0$), then the possibility degree of $X'_i = 1$ is set to 1, and the one of $X'_i = 0$ is set to 0.3.

In terms of ADDs, it can be computed as follows: let us first recall the notation $\mathbf{p}\Big(X_i' \;\Big|\; parents(X_i'), a\Big)^{X_i'=0}$, used to represent the subtree of the ADD $\mathbf{p}\Big(X_i' \;\Big|\; parents(X_i'), a\Big)$ setting X_i' to false (*i.e.* to 0). As well, $\mathbf{p}\Big(X_i' \;\Big|\; parents(X_i'), a\Big)^{X_i'=1}$ is the subtree of the same ADD, setting X_i' to true (*i.e.* to 1). Let us denote by $\mathbb{1}_{\mathbf{p}_\top > \mathbf{p}_\bot}$ the BDD equal to 1 for each variable assignment such that

$$\mathbf{p}\Big(X_i' \;\Big|\; parents(X_i'), a\Big)^{X_i'=0} < \mathbf{p}\Big(X_i' \;\Big|\; parents(X_i'), a\Big)^{X_i'=1},$$

and equal to 0 for other assignments. The BDD always equal to 1 is denoted by $\mathbb{1}$. The BDD $\mathbb{1}_{\mathbf{p}=0.5}$ is equal to 1 for variable assignments such that the probability of the event $X_i' = 1$ (or $X_i' = 0$) is equal to 0.5, and this BDD is equal to 0 otherwise. We can also denote by $\mathbb{1}_{\mathbf{p}_\top < \mathbf{p}_\bot}$ the BDD which is equal to 1 for assignments of variables in $parents(X_i')$ such that the probability of event $X_i' = 1$ is lower than the probability of event $X_i' = 0$: this BDD can be computed from previous BDDs, $\mathbb{1} \boxed{-} \mathbb{1}_{\mathbf{p}_\top > \mathbf{p}_\bot} \boxed{-} \mathbb{1}_{\mathbf{p}=0.5}$, where $\boxed{-}$ is the minus operator, applied to trees. The possibility transition distribution for the i^{th} variable is

$$\pi\Big(X_i' \;\Big|\; parents(X_i'), a\Big) = \boxed{\max}\Bigg\{ \mathbb{1}_{\mathbf{p}=0.5}, \boxed{\min}\Big\{\mathbb{1}_{\mathbf{p}_\top > \mathbf{p}_\bot}, \mathbf{p}\Big(X_i' \Big| parents(X_i'), a\Big)\Big\}, \boxed{\min}\Big\{\mathbb{1}_{\mathbf{p}_\top < \mathbf{p}_\bot}, \mathbf{p}\Big(X_i' \Big| parents(X_i'), a\Big)\Big\} \Bigg\}$$

As well, for each action $a \in \mathcal{A}$, an ADD representing the reward function for this action is provided and denoted by $r(X_1, \ldots, X_n, a)$. Let us define then for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\Psi(s, a) = \frac{r(s, a) - \min_{s,a} r(s, a)}{\max_{s,a} r(s, a) - \min_{s,a} r(s, a)} \in [0, 1].$$

The terminal preference function is set to $\Psi(s) = \max_{a \in \mathcal{A}} \Psi(s, a)$, and the strategy is initialized by $\delta^*(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \Psi(s, a)$ at the beginning of the algorithm.

Note that possibility and preference degrees are not in a scale $\mathcal{L}$ as previously defined (*i.e.* $\Big\{0, \frac{1}{k}, \frac{2}{k} \ldots, 1\Big\}$ for some $k \geqslant 1$). Indeed, possibility degrees comes from probability values, and preferences are normalized rewards. However, only max and min operators are used, so it has no impact on the qualitative results of the computations.

The library used to perform computations with ADDs is the *CU Decision Diagram Package* (CUDD, `http://vlsi.colorado.edu/~fabio/CUDD/`), and the described versions of PPUDD are available at the adress `https://github.com/drougui/ppudd`.

Following figures illustrates the results of IPPC 2014: the score is given in function of the instance index, which generally increases with the difficulty (or the size) of the associated problem.

Figure III.7 presents the scores obtained by each solver for each of the 10 instances of the *Academic advising* domains, *i.e.* the average over 30 trials of the sum of the encountered rewards. Performances of our algorithms are close to the best ones. However, an unexplained and unwanted bug occurred with ATPPUDD for the 2^{nd} instance, as only 3 runs have been performed by this solver. For the other instances, PPUDD and ATPPUDD produce strategies with performances like PROST and GOURMAND, and better than Symbolic LRTDP. This has been less true for the *Crossing traffic* problem, whose results are also described by Figure

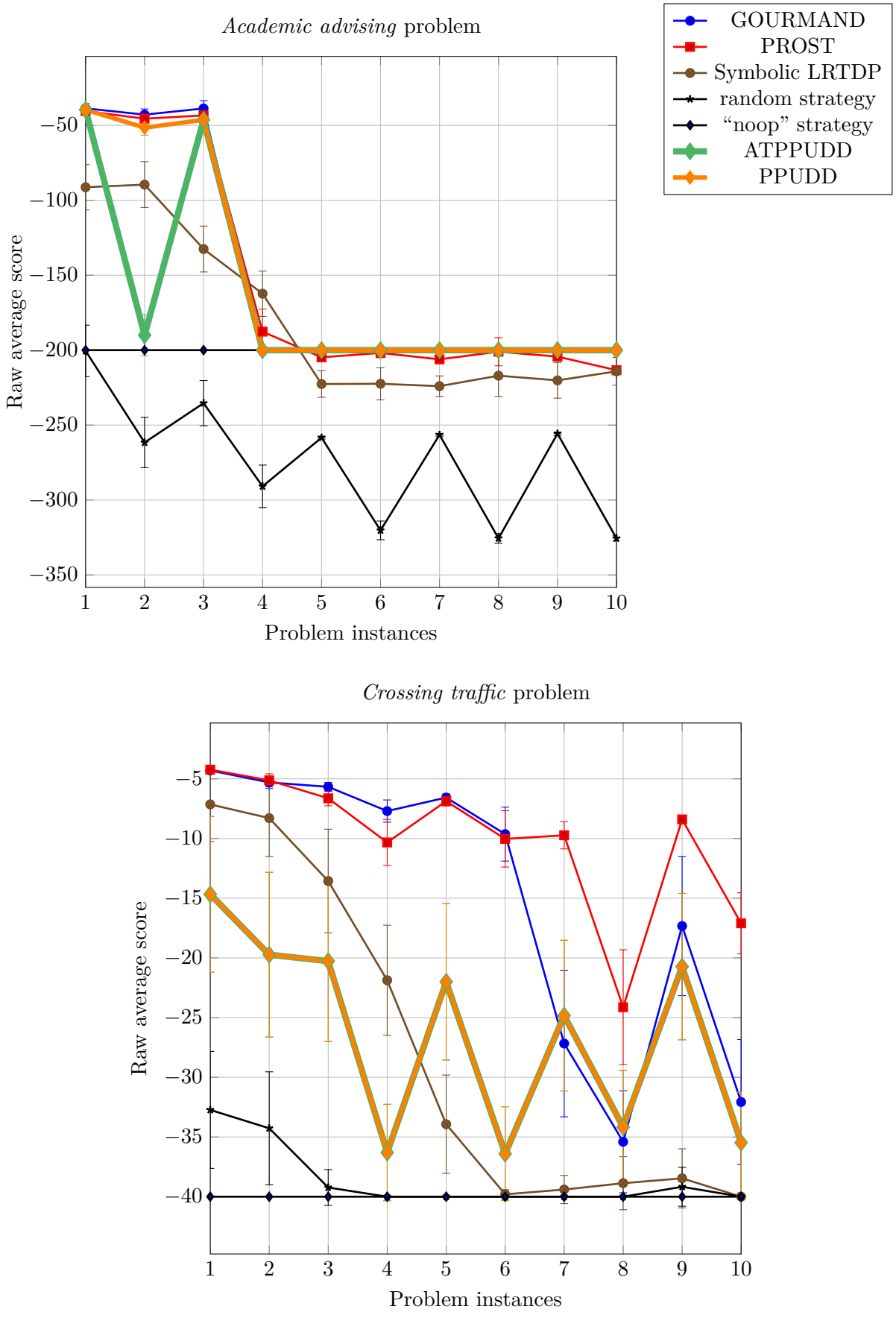


Figure III.7 – *Results of the International Probabilistic Planning Competition – Fully Observable track*

III.7. This problem models a robot which has to reach a goal which is across an highway with a lot of cars. These cars arrive randomly and move left. As the possibility degree of the fact that no car arrive is set to 1 by our naive MDP to π-MDP translation, the optimistic criterion leads to decide to cross the street, even if an unseen car may arrive (with a probability < 0.5 but bit enough to be cautious). This explain the poor quality of the produced strategies for this domain. Note however that, for the 6 last instances (most difficult problems) our approach leads to better strategies than the probabilistic solver Symbolic LRTDP.

In the *Elevators* problem, people arrive randomly and have to be transported to the correct building stage: as the frequentist information is lost using the possibilistic approach and seems important in this problem (people do not want to wait once arrived), scores of our algorithms are poorer than the ones of PROST and GOURMAND. The toy example at the beginning of the introduction of this chapter illustrates that the possibilistic approaches can select actions probabilistically clearly suboptimal when the probability values are at the heart of the problem. PPUDD and ATPPUDD are however better than Symbolic LRTDP, as shown by Figure III.8, and than doing nothing ("noop strategy") or choosing actions randomly ("random strategy"). PPUDD and ATPPUDD have quite good behaviours with the *Skill teaching* problem as illustrated by the same figure. Moreover, ATPPUDD leads to better results for the last three instances: as these instances are the *Skill teaching* problems with the largest system space, the anytime version, which manages the computation time, produces strategies with better performances than PPUDD, which classically solve the associated π-MDP, but cannot complete computations and lead to a poorer strategy.

With respect to other solvers, possibilistic solvers have good results with the *Tamarisk* domain, as shown in Figure III.9.. However, some instances (e.g. the 6^{th}, the 8^{th} and the 10^{th}) are not even run as the ADD instantiation takes too long. Symbolic LRTP faces the same issue as it uses also the LISP-like encoding of the problem. We think that this is an issue specific to the competition, as each problem has to be equivalently translated into three different languages (RDDL, PPDDL and LISP-like), which produces sometimes artificially complex encodings of the problems. The *Traffic* domain is really hard to solve by PPUDD and ATPPUDD (see Figure III.9). Actually the least scores are obtained with this domain, and even the random and the noop strategies are better strategies. Note that we did not implement any "watchdog" returning random actions when the computed strategy is less effective than the random one. However, this kind of gadget is essential to improve results for such large and risky problem. As mentioned above for the *Crossing traffic* problem, the optimistic criterion may lead to dangerous actions, as it does here. Moreover, as this problem involves frequentist information (car arrivals) an high suboptimality of the strategy produced by the possibilistic approach is confirmed for this kind of problems (see the *Elevator* problems). Finally, the *Traffic* problem is known to be one of the hardest domain, so ADD instantiation takes long, as well as computations, which are then not proceeded enough to produce satisfying results.

Finally, the two last domains, whose results are described in Figure III.10, are called *Triangle Tireworld* and *Wildfire*. Firt, ATPPUDD faces an unexplained bug for each instance of the *Triangle Tireworld* domain: no trial is performed from the 5^{th} instance, and maximum 2 trials are performed for other instances (which explains the poor score for each instance). As already mentioned for the *Tamarisk* domain, ADD instantiation takes too long for the last instances, and no trial is performed for the last 4 instances with PPUDD too: Symbolic LRTDP faces the same issue. The last domain, called *Wildfire*, leads to highly frequentist problems: it involves random fire starts. That is why PPUDD and ATPPUDD strategies are not really efficient, but not too distant from Symbolic LRTDP solver's results.

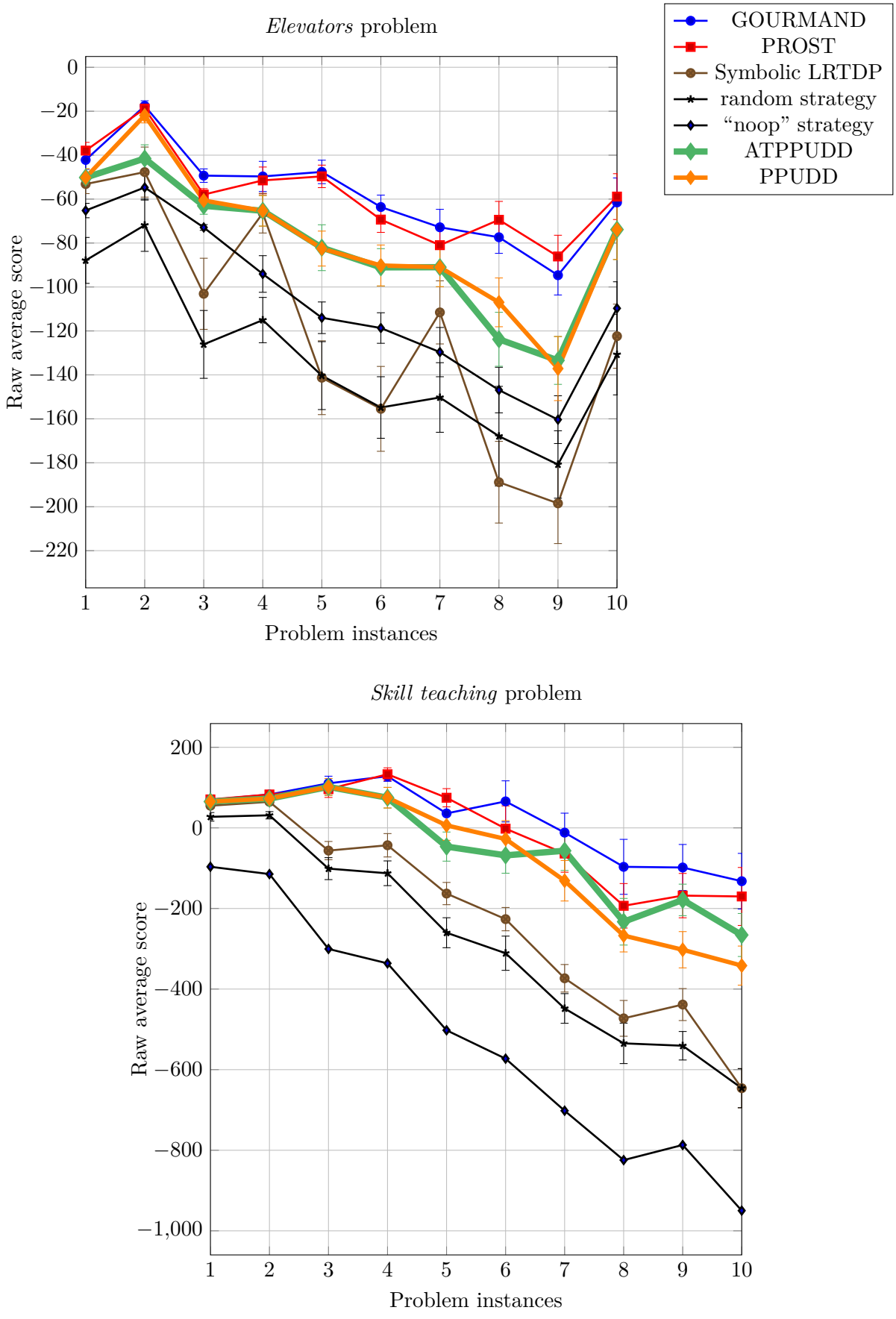


Figure III.8 – *Results of the International Probabilistic Planning Competition – Fully Observable track*

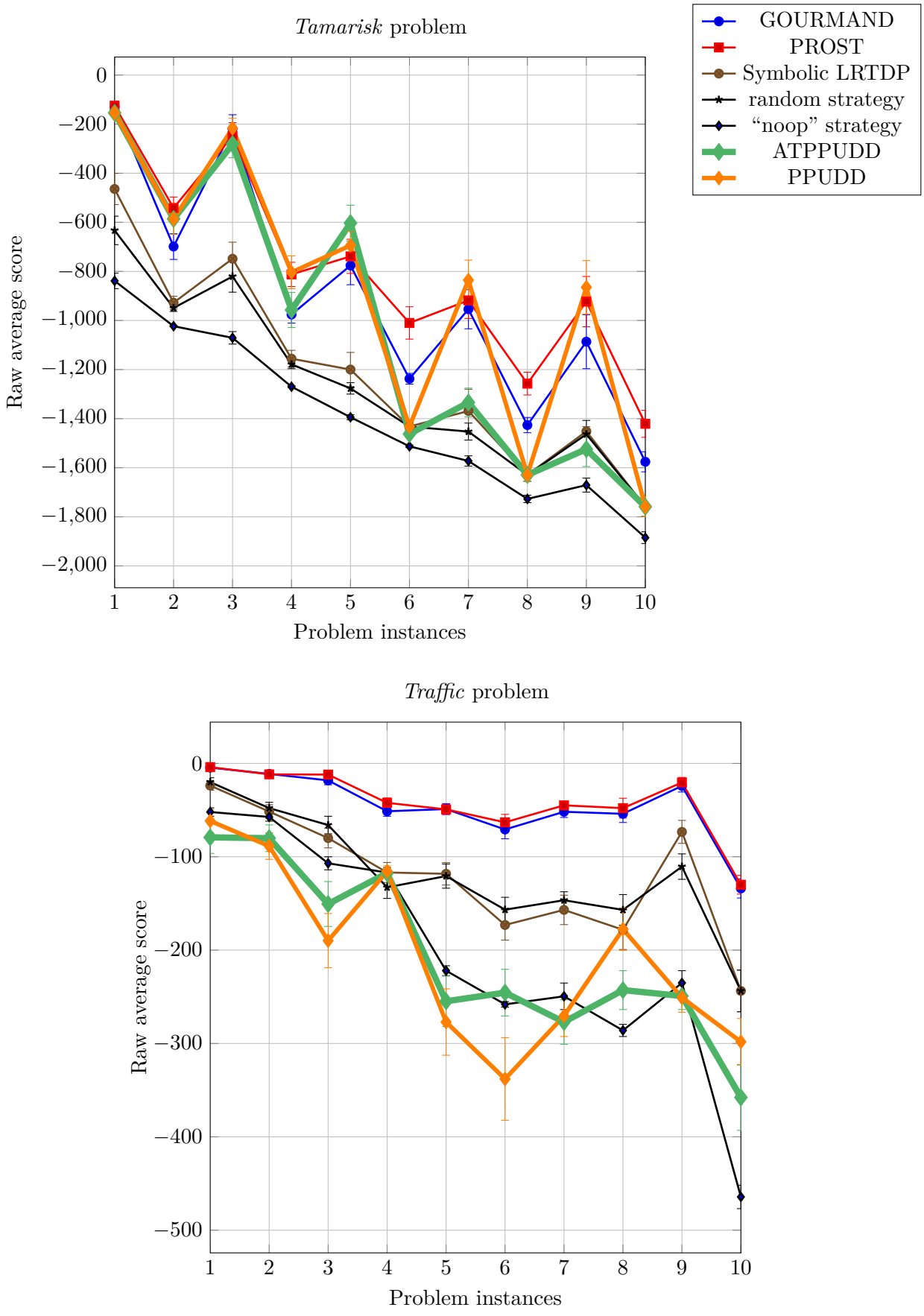
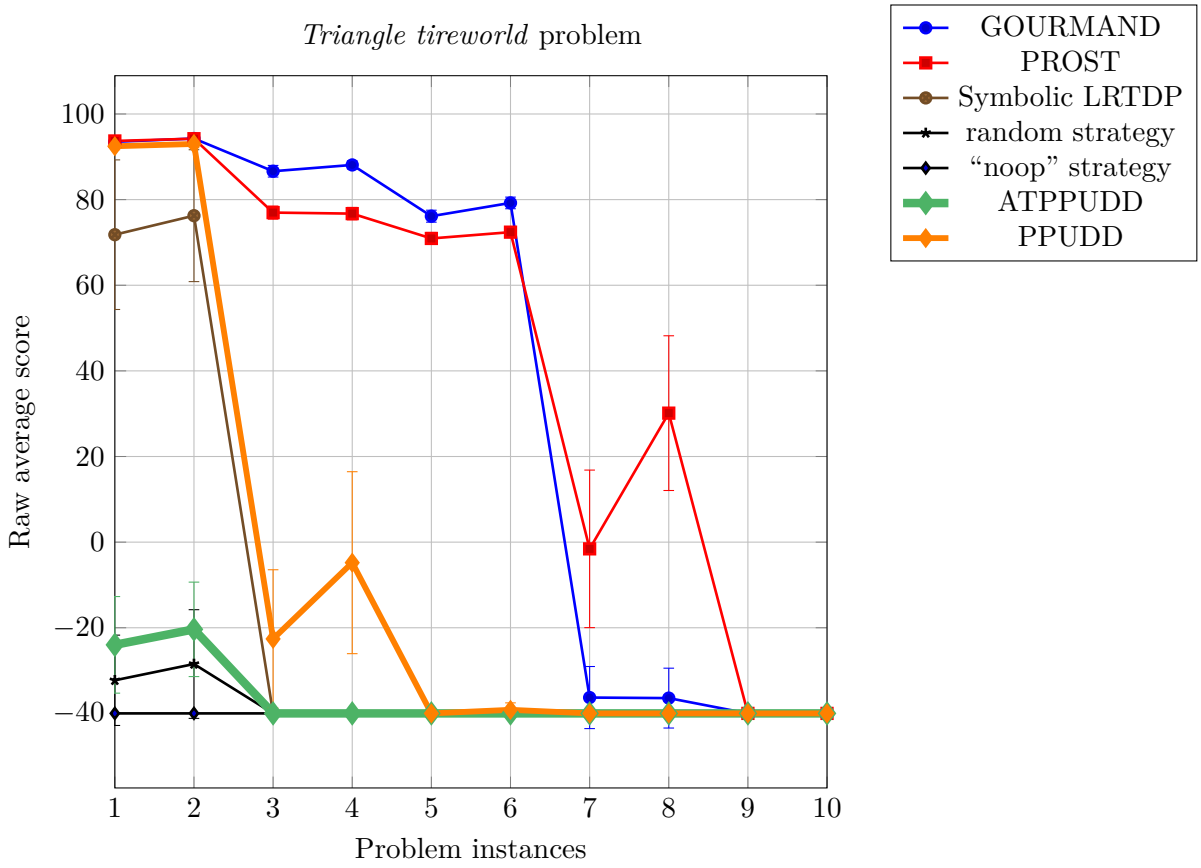


Figure III.9 – *Results of the International Probabilistic Planning Competition – Fully Observable track*



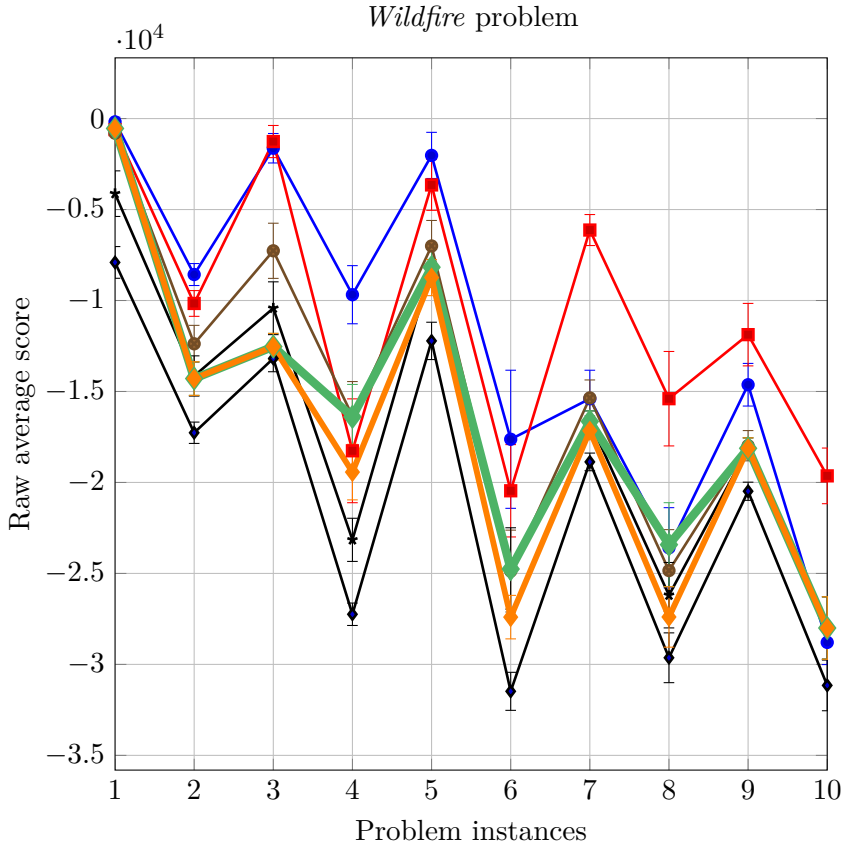


Figure III.10 – *Results of the International Probabilistic Planning Competition – Fully Observable track*

III.5 Conclusion

We presented PPUDD, the first algorithm to the best of our knowledge that solves factored possibilistic qualitative (MO)MDPs with symbolic calculations. In our opinion, possibilistic models are a good tradeoff between non-deterministic ones, whose uncertainties are not at all quantified yielding a very approximate model, and probabilistic ones, where uncertainties are fully specified, sometimes arbitrarily in practice. The resolution of planning problems using the framework of non-determinism is called *contingent/conformant planning* studied for instance in [1, 16]. By the way, in the introduction of this thesis, we might also add "non-determinism" as a particular case of Possibility Theory in Figure 7, as values of associated non additive measures are 0 or 1 instead of a more flexible scale $\mathcal{L}$.

Moreover, π-MOMDPs reason about finite values in a qualitative scale $\mathcal{L}$ whereas probabilistic MOMDPs deal with values in $\mathbb{R}$, which implies larger ADDs for symbolic algorithms. Also, the former reduce to finite-state belief π-MDPs contrary to the latter that yield *continuous*-state belief MDPs of significantly higher complexity. Our experimental results highlight the point that using an exact algorithm (PPUDD) for an approximate model (π-MDPs) can bring significantly faster computations than reasoning about complex exact models, while providing better strategies than approximate algorithms (APPL) for exact models. In the future, we would like to developp a probabilistic algorithm using the generalization of our possibilistic belief factorization theory to probabilistic settings (see Theorem 26): related but sightly different results have been proposed for probabilistic POMDPs [131, 112]. These results also does not concern the case of mixed observability.

This chapter finally presents the results of our possibilistic approach during IPPC 2014: the highlighted bottleneck of our possibilistic algorithms resides on the translation from probabilities to possibilities: the naive automated translation presented before the description of the results leads to poor policies in benchmarks with complex dynamics and reward structures. Another issue is the size of the input LISP-like encoded domains whose ADD instantiation before optimization takes a very long time or does not even fit into memory for many difficult benchmarks: this difficulty is shared with the Symbolic LRTDP solver. However, there is almost no discretization of the initial probability values defining the MDP in order to produce the possibility degrees during the instantiation of the ADD defining the π-MDP: the maximal difference between two possibility degrees is set to 10^{-3}. Stronger discretizations have not been tested yet, and could improve scores of our solvers for problems with such memory issues. Modeling issues have been also highlighted, namely the fact that some problems request a cautious behaviour, not provided by the use of the optimistic criterion (see Definition III.8) used during the competition. Moreover, as illustrated in introduction, these experiments show that problems with high entropy events are outperformed by probabilistic approaches since the possibilistic approach does not take into account the frequentist information about the problem. The use of *lexi*-approaches, as used in the following chapter, may be a possibilistic stratagem to get around this issue. Note finally that the partially observable version of PPUDD (with the generation of a mask of reachable belief states, avoiding useless computations on unreachable beliefs) is also available on the repository `https://github.com/drougui/ppudd`.

The next chapter, Chapter IV, deals with *Human-Machine Interaction* (HMI) problems: the uncertainty dynamics of the system are in this context typically not known in terms of probability values, and the qualitative possibilistic approach is shown to be a natural approach to produce efficient diagnosis of human errors.

Finally, the last chapter, Chapter V, takes into account the remarks made using the results of IPPC14: an approach using Probability and Possibility Theory in order to benefit from both approaches in the resolution of factored POMDPs is presented: quantitative information of the problem is kept to avoid the highlighted modeling issues, and the belief state is handled in

a possibilistic way, in order to get a smart discretization of it and to benefit from a finite and factorized belief state spaces. This approach leads to a factored probabilistic MDP which can be solved for instance by GOURMAND or PROST (which does not use the memory constraining LISP-like encoding).

IV Application of Qualitative Possibilistic Hidden Markov Processes for Diagnosis in Human-Machine Interaction

The work developed in this chapter is quite independent from the main theme of this thesis, namely the problem of *Decision Making under Uncertainty*. This work, performed in collaboration with Sergio Pizziol and extending the work of his PhD thesis [110], contributes to modelling human-machine interactions. It is part of this thesis since it is a significant example of the need of qualitative models (such as those presented in previous chapters) in some practical situations.

We formalize here a framework providing an estimate of the human assessment of the machine state, an automated detection of human operator attentional errors, and finally an estimate of the most plausible causes of these errors. A qualitative possibilistic approach is used to deal with uncertainty about the human operator assessment.

The human-machine context is first introduced to point out the need for a new modelling method for human attentional error. Then a human error model is derived from the machine logic, using expert assumptions on human errors and their plausibility. A human-machine interaction model results from the combination of the machine logic and the error model. Using the Possibility Theory, an analysis model estimating the human assessment is built on the interaction model, summed up in a *Possibilistic Hidden Markov Processes* (π-HMPs). The possibilistic analysis is first performed on a toy example. Finally the soundness of the approach is shown through simulation tests with pilots performing a flight mission.

IV.1 Introduction

In human-machine interaction studies, the problem of the correct human assessment of the machine state has been widely discussed. The main issue is that a human operator with a wrong assessment of the machine state is likely to perform *erroneous actions*, *i.e.* actions whose outcome is different from what is intended [78].

Many different approaches have been proposed to deal with this issue: among them, mental models and situation awareness [65, 119], formal inference rules [99], or *error models* for the human misinterpretation of the machine feedbacks [118]. Those approaches are based on a deterministic model for the human error, suited for error dynamic analysis but not for error detection. Moreover, they do not benefit from the flexibility provided by uncertainty representations.

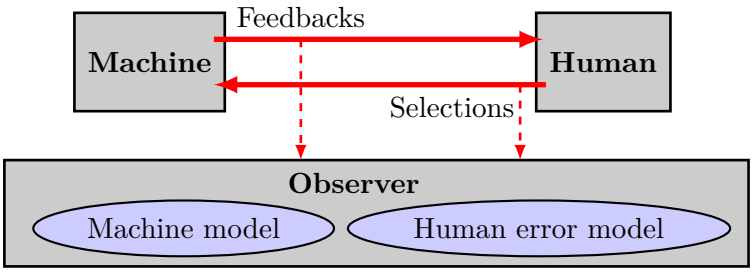


Figure IV.1 – *The three actors involved in the study. The red arrows represent information flows.*

The human assessment of the machine state is not observable during the interaction with the machine: nevertheless it may be estimated via uncertainty modelling for example using Probability Theory [98]. However probability values can be difficult to define in practice because of a lack of quantitative information related to the human operator's behaviour. A method to build an interaction model relying on less informative data is needed. This chapter proposes a new method building such a model with qualitative expert data and using the machine logic.

The human assessment of the machine state (shortly *assessment*) is mainly based on the *feedbacks* provided by the machine. *Feedbacks* are pieces of information the machine sends (via visual or aural alerts/signs) in order to inform the human operator about its current state. Some formal approaches have been proposed in order to estimate whether the human receives enough feedbacks [102, 37]. Nevertheless even if enough feedbacks are provided, the problem of their correct reception remains. If the human does not perceive some machine feedbacks during the interaction, the human assessment may be incorrect, leading to erroneous actions based on such a wrong assessment: the main objective of this work is to formalise an analysis model detecting assessment errors through human assessment estimation.

Three actors are involved in this framework: a machine, a human operator acting on the machine, and an observer analysing the human-machine interaction. Human operator actions on the machine are called *selections*.

The observer knows the machine logic and contemplates possible human assessment errors. Moreover, during the human machine interaction, it observes a sequence of data generated by the machine and the human operator: these data are called *observable occurrences*, and consist in each successive feedback (from the machine) and selection (from the human operator), as illustrated Figure IV.1. Machine state changes corresponding to those feedbacks and selections can also be considered as part of the observable data: indeed, the observer perfectly knows the initial machine state as well as the deterministic model of the machine, so the current machine state is easily determined.

The analysis model, proposed here, is meant to help the observer in the analysis of the human-machine interaction: using successive observable occurrences, as well as a machine model and a human error model, it provides an estimate of the human operator assessment of the machine state, a detection of human assessment errors, and an explanation for those errors.

The system designers could later modify the machine logic to make it take into account the assessment errors detection and diagnosis performed by the observer using the analysis model. For instance they could provide new specific feedbacks meant to correct the human operator assessment [44, 43]. Note that those applications are out of the scope of this work. This work focuses on a method to set up a human-machine interaction model from the machine model and on the definition of the analysis model; experiments on a flight simulator are also

provided, showing that this approach is reponsive in practice.

Some key concepts are detailed in the next section, as the definition of the machine model describing the machine logic. The human assessment error model (shortly *error model*) is then defined starting from the machine model. Next the human-machine interaction model is presented, resulting from the combination of the machine logic and the error model. It exhaustively defines all the human assessment transitions considered as possible by the observer. Later some working assumptions are given: assumed by the observer and expressed in natural language, they involve the plausibility of these human assessment transitions. An analysis model can then be set up based on these assumptions.

The analysis model presented in this work uses the Qualitative Possibility Theory as it is well suited to encode qualitative expert knowledge: the possibilistic analysis model is formally described, in the form of a Hidden Markov Process. It provides a human assessment estimation, assessment error detection and explanation of the error.

A detailed example of this approach is provided Section IV.4 analysing interactions for a simple three-state machine. In the last section, the method is tried through tests with pilots performing a flight simulator mission.

IV.2 Framework for human-machine interactions modelling including assessment errors

Hereafter we call *state* the machine state represented by the notation $s \in \mathcal{S}$. The actual human assessment of the state is represented by $h \in \mathcal{S}$: the equality $h = s^* \in \mathcal{S}$ means that the human operator thinks that the state of the machine is s^*. Note that this work is based on the simplifying assumption that the human operator is certain about the state of the machine: the representation of the human knowledge is limited to a unique machine state $h = s^*$. This unique state can also be seen as the most plausible one from the human operator's point of view, *i.e.* the one on which she/he bases her/his selections. Remember that a selection is a human operator action on the human-machine interface.

If no assessment error arises, assessment h coincides with actual state s. However the sending of feedbacks does not guarantee the correct receipt of the information, in particular for the *automated state changes* [67], *i.e.* state changes that are not fired by a selection. So assessment errors may occur.

The actual assessment h is not observable since the observer has no access to the human assessment of the situation: the main contribution of this chapter is then to provide a possibilistic estimation for it. Successive *observable occurrences* (shortly *occurrences*) *i.e.* each successive feedback (from the machine) and selection (from the human operator), are represented by the variable v, and are used to update this estimation. Occurrences can be divided into three categories:

- human selections on the machine interface;
- automated machine state changes with relevant feedback sending;
- the initialization, representing the beginning of the interaction process.

The observer knowing the initial state and machine model is able to deduce the actual state at each occurrence (so the machine state is considered as observable as well). Next section details how this machine logic is described, starting point for a human error model derivation.

IV.2.1 Machine model

The machine logic is summarized through a *logic table* representation [67]. As this representation has been developed to describe the deterministic behaviour of the machine, the logic table takes into account the machine state s, but not the human assessment h. Machine state transitions are represented as triplets (previous state s, current occurrence v', current state s'). These machine state transitions are summarized in pairs of (*situation*, *behaviour*):

Definition IV.2.1 (*Situation*)

A situation *is defined as the conjunction between a proposition about current occurrence $\mathcal{P}(v')$ and a proposition concerning the previous state $\mathcal{P}(s)$: $\mathcal{P}(v') \wedge \mathcal{P}(s)$. In practice, the proposition about the current occurrence is a disjunction of occurrences.*

Moreover the machine state is described by a tuple of state variables: $s = (s^1, s^2, \ldots, s^n)$. The proposition about the previous state is a Conjunctive Normal Form (CNF) of these state variables, i.e. a logic conjunction between disjunctions of assignments of the same state variable.

For instance consider a set of possible occurrences $\{ v_A, v_B, v_C \}$, and a set of states described by variables $(s^1, s^2) \in \{ s^1_A, s^1_B \} \times \{ s^2_A, s^2_B, s^2_C \}$. An example of proposition about the current occurrence might be $\mathcal{P}(v') = (v' = [v_A \vee v_C])$. Current occurrences and previous state variables are either defined explicitly or take the parametric value "no matter which occurrence/assignment" denoted by "$[**]$". Proposition $\mathcal{P}(s) = (s^1 = [**]) \wedge (s^2 = [s^2_A \vee s^2_B])$ is an example of CNF, or proposition about the current state. The situation is finally fully described with $\mathcal{P}(v') \wedge \mathcal{P}(s)$. In this example, the situation is expressed in natural language as: "occurrence is either v_A or v_C, variable s^1 takes any value, and variable s^2 is either s^2_A or s^2_B".

Definition IV.2.2 (*Behaviour*)

A behaviour*, which is the result of a situation, is a proposition $\mathcal{P}(s')$ defined as a logic conjunction between assignments of the different state variables.*

These assignments are either defined explicitly, or take the parametric value "same assignment as the corresponding previous state variable assignment" denoted by "$[*]$".

An example of proposition describing a behaviour for the previous situation example might be $\mathcal{P}(s') = (s'^1 = [*]) \wedge (s'^2 = [s^2_C])$. In this example, the behaviour is expressed as: "variable s'^1 assignment is the same as s^1, and variable s'^2 assignment is s^2_C".

A complete set of (situation, behaviour) pairs can be summed up in a logic table.

Definition IV.2.3 (*Logic table*)

The set of pairs (situation, behaviour) is represented in an explicit way with a table called logic table*. The first column of the table contains the occurrence variable notation v and state variables names, the second column contains possible occurrences, and possible state variables assignments. Pairs (situation, behaviour) are represented in the next columns (1, 2, 3, etc). In those columns, boxes containing* 1 *mean that the current occurrence variable or state variable is equal to the current line value (assignment). If for some* situation *all the boxes corresponding to the occurrence variable or a state variable are empty, the occurrence variable or state variables take value $[**]$ (no matter which occurrence/machine state). Note that this is equivalent to fill those boxes with many* 1*: the only purpose of this convention is the table readability. If for some* behaviours *all the boxes for a state variable are empty, the variable takes value $[*]$ (same as the previous state).*

As a toy example let us consider the case of a machine whose state can be represented with one binary variable $s \in \{ s_A, s_B \}$. The set of possible occurrences is $\{ v_A, v_B, v_C \}$. Table IV.1 gives the logic table of the following (situation, behaviour) pairs:

		1	2	3
SITUATION				
v'	v_A	1		
	v_B		1	
	v_C	1		
s	s_A	1	1	
	s_B			1
BEHAVIOUR				
s'	s_A			
	s_B		1	

Table IV.1 – *Example of logic table for a machine with one binary machine state variables and three possible occurrences: each pair (situation, behaviour) is described by a column.*

1. • situation: $(v' = [v_A \vee v_C]) \wedge (s = [s_A])$;
   • behaviour: $(s' = [*])$.
2. • situation: $(v' = [v_B]) \wedge (s = [s_A])$;
   • behaviour: $(s' = [s_B])$.
3. • situation: $(v' = [**]) \wedge (s = [s_B])$;
   • behaviour: $(s' = [*])$.

The total number of columns is equal to 3, as the number of pairs (situation, behaviour) describing the state machine. Column 1 of table IV.1 has to be read: "if $v' = v_A \vee v_B$ and $s = s_A$, then machine state remains the same".

As the machine logic depiction has been presented, a human error model can be now deduced, leading to a full human-machine interaction model.

IV.2.2 Derivation of an error model

Classically, human-machine interaction models are based on the machine logic, and, for each occurrence, only on the expected (or feared) consequence on the human assessment of the machine state [118, 111]. The presented interaction model provides a more expressive representation of the assessment dynamics: many consequences on the human assessment of the machine state, called *effects* and denoted by variable $e \in E$, are considered possible for each occurrence. Nevertheless, they may be defined as more or less plausible by experts. In other words the term *effect* means "the (non observable) effect (of an observable occurrence) on the human assessment of the machine state". For instance one possible effect is the correct human perception and interpretation of a feedback. Other possible effects, for the same observable occurrence (*i.e.* the feedback sending) could be that the feedback goes unperceived or misinterpreted.

An effect can be formally defined as the result of a partial function $f_e : S \times V \times S \to E$ of previous assessment h, current occurrence v' and current assessment h': $e = f_e(h, v', h')$. Indeed that function defines the effect of the occurrence v' on the assessment dynamics, *i.e.* on the transition from h to h'. Partial function f_e is not defined for all triplets $(h, v', h') \in S \times V \times S$. Indeed, in the context of occurrence v' some assessment transitions are not assumed to be possible by experts: if h cannot become h', $f_e(h, v', h')$ is not defined, and no effect is associated with this transition. Effects of v' are thus each $f_e\,(h, v', h')$, $\forall (h, h') \in \mathcal{S}^2$ when defined.

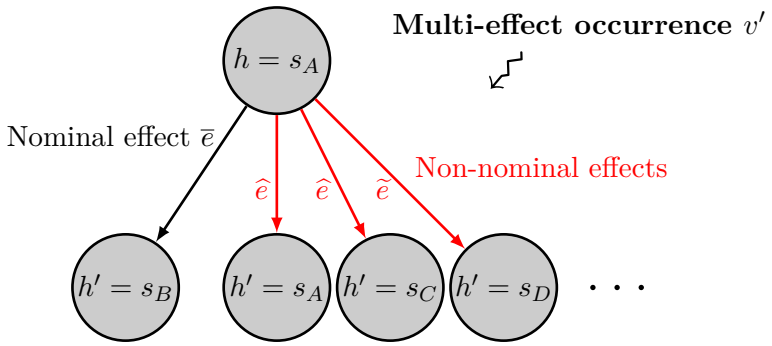


Figure IV.2 – *Nominal effect and non-nominal effects of the occurrence v' on the assessment $h \in \mathcal{S}$ which becomes $h' \in \mathcal{S}$.*

For a given occurrence, the effect when no assessment error arises *i.e.* when the human assessment transition is equal to the actual machine state transition, is the *nominal effect*. Nominal effects are then already defined given the logic table, replacing machine state variables s and s' with human assessment variables h and h'.

Non-nominal effects could also take place and correspond to assessment errors. Nevertheless for some occurrences only nominal effects are taken into account (*i.e.* the observer does not foresees any human assessment error for those occurrences). Occurrences with more than one possible effect for at least one possible assessment h are called *multi-effect* occurrences, and are illustrated in Figure IV.2. Formally if v' and h are such that $\exists (h'_A, h'_B) \in \mathcal{S}^2$ and $h'_A \neq h'_B$ for which $f_e(h, v', h'_A)$ and $f_e(h, v', h'_B)$ are defined, and $f_e(h, v', h'_A) \neq f_e(h, v', h'_B)$, v' is a *multi-effect* occurrence.

The error model is completed once all non-nominal effects have been defined: the logic table can be enhanced adding non-nominal effects for some occurrences. The way those non-nominal effects are described is the same as for the nominal effects: by pairs (situation, behaviour) represented by columns of the logic table. Referring to the example of the logic table given above (see Table IV.1), the expert knowledge could for instance assert that occurrence v_C could possibly make the human believe that if the machine state is initially s_A it finally changes to s_B. This potential effect is described by column 4 in Table IV.2. This new column does not replace the nominal case that is unaltered and still described by column 1. For readability columns corresponding to non-nominal effects are written in red. Occurrence v_C is thus a multi-effect occurrence. A logic table that includes assessment errors as Table IV.2 is called an *enhanced logic table*. The expert knowledge may again enhance the error model, stating that occurrence v_A could also lead to the same kind of human assessment error (see column 5).

Remember that effects e concern the human assessment dynamics h, which is of course not observable: actual effects are thus not observable, as a result of $f_e(h, v', h')$. Effects can however be sorted according to their plausibility, as presented right now.

IV.2.3 Effect plausibility

In this study nominal effects are considered as generally more plausible than the corresponding non-nominal ones *i.e.* than the corresponding human assessment errors starting from the same previous assessment h, and under the same occurrence v': the human operator is thus assumed to know the machine logic and to have a quite good perception of the feedbacks.

Experts, after the enumeration of the potential non-nominal effects, have also to sort all effects according to their plausibility, dividing them into categories: for instance, effects whose plausibility is normal $\overline{e}$ (shortly *normal effects*), effects whose plausibility is less than normal but not unusual $\widehat{e}$ (shortly *less than normal effects*), effects whose plausibility is unusual $\underline{e}$ (shortly *unusual effects*), or even effects whose plausibility is very rare $\tilde{e}$ (shortly *very rare*

columns		1	2	3	4	5
SITUATION						
v'	v_A	1				1
	v_B		1			
	v_C	1			1	
h	s_A	1	1		1	
	s_B			1		1
BEHAVIOUR						
h'	s_A					1
	s_B		1		1	
EFFECT		$\overline{e}$	$\tilde{e}$	$\overline{e}$	$\widehat{e}$	$\underline{e}$
POSSIBILITY		1	ε	1	λ	δ

Table IV.2 – *Enhanced logic table of the logic table IV.1: occurrences v_A and v_C have both a non-nominal effect, described respectively by columns 5 and 4. Each column represent a pair (situation, behaviour), and effect row represents the effect plausibility label. Last row assigns a possibility degree to each effect (see section IV.3.1).*

effects). The line "effect" is thus added to Table IV.2 specifying effects plausibility. For instance, nominal effect described by column 1 is normal according to the expert ($\overline{e}$), but nominal effect described by column 2 is very rare ($\tilde{e}$).

We have made the assumption that for each assessment h, there always exists at least one occurrence v' with a normal effect. Thus, for each current human assessment h, it exists one possible occurrence and next human assessment considered as normal *i.e.*

$$\exists v', h' \text{ such that } f_e(h, v', h') = \overline{e}. \tag{IV.1}$$

This remark is essential and forms a rule imposed in practice when filling the effect row of the enhanced enhanced logic table. It means that for each human assessment, at least one next step of the human-machine interaction is normal. Moreover, this property suits the possibilistic analysis model construction, as recalled section IV.3.

In the next section, the analysis model is defined from a given effect plausibility ranking, using a plausibility measure on the human-machine system dynamics: more details about the chosen measure are given in section IV.3. The last part of this section (IV.2.5) will derive the effect ranking from a set of expert rules, leading to a fully defined human-machine interaction model.

Note that an effect is *normal* if its plausibility is considered as normal (by the expert, or the system designer). Nominal effects and normal effects must not be confused (see Figure IV.3): the words *normal, unusual* are used to define the plausibility of effects *i.e.* to sort them, in order to fully define the interaction model, as just explainded in this section IV.2.3 and performed from general assumptions in section IV.2.5. On the other hand, effects are said *Nominal* if they represent a human assessment transition without error, *i.e.* a human assessment transition corresponding to the machine state transition (ideal human understanding). Thus, effects are said *non-nominal* if they represent assessment errors, and are added to the logic table using expert knowledge.

In the following section IV.2.4 the human-machine interaction system dynamic is detailed.

IV.2.4 System dynamics: trajectories and exceptions

After the manifestation of $m \geqslant 0$ occurrences, the sequence of machine states is called the $(m+1)$-length *state trajectory* and is denoted by $\mathcal{S}_m = (s_0, s_1, \ldots, s_m)$. This trajectory is considered

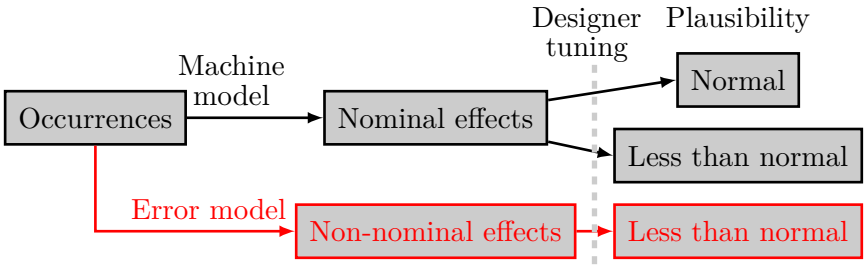


Figure IV.3 – *Nominal effects, non-nominal ones defining the error model, and plausibility evaluation.*

as observable, as well as the $(m+1)$-length *occurrence trajectory* $\mathcal{V}_m = (v_0, v_1, \ldots, v_m)$. In other words the observer defined in introduction section IV.1 and Figure IV.1, thanks to their knowledge of the machine logic, is able to provide $\forall 0 \leqslant t \leqslant m$,

- the occurrence at stage t, v_t (e.g. selection, or automated state change),
- and actual state of the machine s_t, deduced from the machine logic.

However the actual *effect trajectory* $(e_0, e_1, \ldots, e_m)$ and *assessment trajectory* $(h_0, h_1, \ldots, h_m)$ are not observable. They may be estimated using the possibilistic analysis model described in section IV.3. Remember that each occurrence may have many effects. So a $(m+1)$-length occurrence trajectory corresponds to many possible $(m+1)$-length effect trajectories: $\forall 0 \leqslant t \leqslant m$, e_t is an effect[1] of the occurrence v_t. Each time a multi-effect occurrence is fired, several *assessment trajectories* are possible, one or more for each possible effect (see Figure IV.2). The set of possible effects trajectories is denoted by $\mathcal{E}_m$ and the set of possible assessments trajectories $\mathcal{H}_m$.

In practice, possible effects and assessments trajectories are stored together in the form of *non-observable trajectories*: $(e_0, h_0, e_1, h_1, \ldots, h_{m-1}, e_m, h_m)$ with $\forall 0 \leqslant t \leqslant m$, $e_t = f_e(h_{t-1}, v_t, h_t)$ if f_e is defined for this triplet, and removing h_{-1} for $t = 0$. The firing of a new occurrence v_{m+1} updates the set of non-observable trajectories adding possible effects and assessments. Each non-observable trajectory ending with $h \in \mathcal{S}$ at stage m, is completed with each pair $(f_e(h, v_{m+1}, h'), h')$ such that $f_e(h, v_{m+1}, h')$ is defined, *i.e.* stored in the enhanced logic table. Because several multi-effect occurrences may be fired the number of possible non-observable trajectories may increase significantly.

Initially the most plausible non-observable trajectory is the one that includes only nominal effects (*i.e.* no assessment errors). The corresponding assessment trajectory (removing effects from the non-observable trajectory) is called the *objective assessment trajectory* and is equal to the machine state trajectory. After the firing of an occurrence whose effects are all considered at the most as unusual in the actual situation, the objective assessment trajectory is no longer considered as normal. This situation is called an *exception* and the occurrence that led to the exception a *triggering occurrence*. Typical triggering occurrences are selections considered as erroneous in the particular context.

Let us describe the mechanism and the behaviour of the wanted analysis model which will be set up in section IV.3.1. If an exception is detected by the analysis model, the model itself verifies if there is a non-nominal effect (*i.e.* an assessment error) in the past history that, if considered as the actual effect, could lead to a situation in which the firing of the triggering occurrence is not unusual, but instead normal. For instance the analysis models may verify if there is a human feedback misinterpretation that could explain a human selection otherwise considered as erroneous. This non-nominal effect is called *exception explanation* and the assessment trajectory embedding this exception explanation is considered as the new most

[1] Remember that effects are not observables.

plausible one. The formerly most plausible assessment trajectory is no longer coherent with the actual observations of the system. Therefore, its plausibility is decreased. On the other hand, if no exception explanation is found, the plausibility of the assessment trajectories remains unchanged. The *possibilistic Bayes rule*, also presented in the section IV.3, formally defines how to implement these concepts.

The following section IV.2.5 presents the expert rules chosen to complete the human-machine interaction model: these assumptions about effects are used in applications presented in last sections IV.4 and IV.5.

IV.2.5 Working assumptions

In order to define in practice non-nominal effects modelling human operator assessment errors, as well as their plausibility, two methods are possible: designing them one by one *by hand* with the help of experts of the domain, or deriving them *mechanically* from a limited number of general assumptions formulated by the experts. Nevertheless a mixed approach is also possible: for instance designers could start with the *mechanical* assumption-based method and successively suppress or add *by hand* some non-nominal effects, or even rank *by hand* the plausibility of some effects.

Hereafter some assumptions for the mechanical approach about possible effects and their plausibilities are defined. Defining a generic set of assumptions is out of the scope of this work. Note that these rules have been defined by experts for the experiment presented in section IV.5, and may be unsuitable for other applications. They are defined here to set an example of interaction model (including the ranking of effects), before the building of the possibilistic analysis model.

Here is the list of the chosen working assumptions:

- the human knowledge of the machine behaviour is correct;
- the human should perceive feedbacks, but she/he can possibly miss them;
- the human knowledge of the initial state is uncertain, but is likely to coincide with the real one;
- selections that do not change the machine state are considered as *slips*, *i.e.* unmeant selections;
- the missing of a feedback is more likely to happen than a slip or a mistake: a *mistake* is a selection or a lack of selection defined as erroneous by the system designer.
- the missing of $n < n_{max}$ feedbacks is more likely to happen than missing $n+1$ feedbacks, or a slip.

The first assumption implies that the human is sufficiently trained to use the machine and knows its behaviour: nominal effects are then generally considered as more plausible than the corresponding non-nominal ones. as previously announced.

According to the next two assumptions, the interaction model takes into account two multi-effect occurrences. The first one is the feedback sending: feedbacks should be perceived (nominal effect) but could go unseen (non-nominal effect). The second one is the initial human appreciation of the machine state: the initial assessment should be correct (nominal effect), but could be wrong (non-nominal effect).

The fourth assumption means that selections which do not change the machine state are not voluntary. The last assumption states that the more feedbacks are lost, the less the situation is plausible.

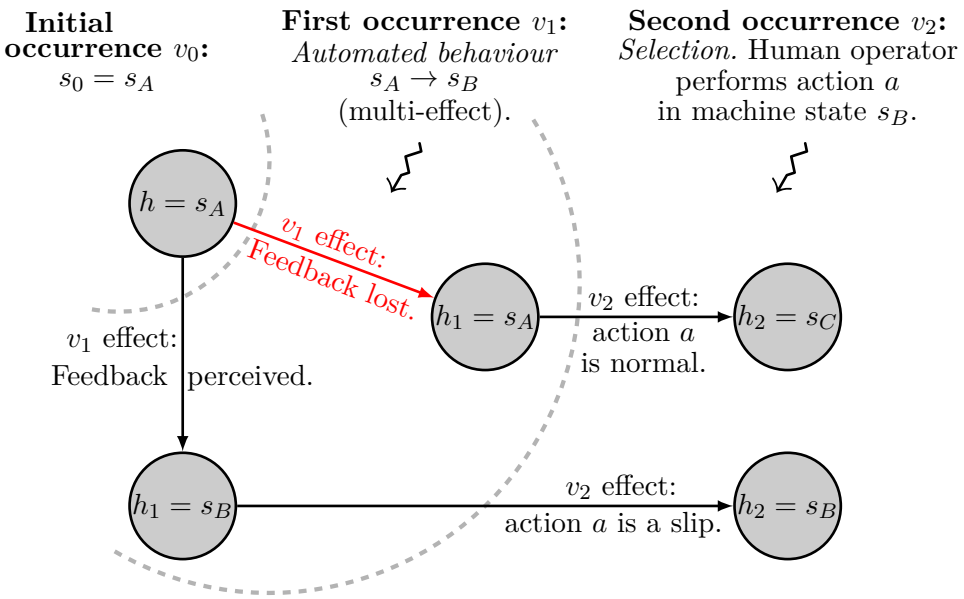


Figure IV.4 – *Loss of feedback as an exception explanation.*

For this interaction model, the following occurrences and corresponding effects are thus used:

- the execution of selections which have one nominal effect; depending on the current assessment they are classified as normal selections, or as slips/mistakes which are unusual;
- an *automated state change* with two possible effects: the reception and well interpretation of the relevant feedback (nominal), which is normal, and the loss of the feedback (non-nominal), which is not normal but more likely to happen than an unusual effect;
- the state initialization with two possible effects: correct initialization (nominal), which is normal, and wrong initialization (non-nominal), which is unusual.

The interaction model is so fully defined. The analysis model detailed in the next section is based on the Qualitative Possibility Theory (see Section I.2.1) providing a formal way to sort effects according to their plausibility. Starting from a human-machine interaction model, built as presented in the current section, the analysis model leads to a possibilistic estimation of the human assessment at each step of the process.

Before detailing the analysis model let us show with an example the desired comportment of the model for an automated state change followed by an action that is a slip in the actual state of the machine (see Figure IV.4).

Let us start with the automated state change with machine initial state $s_A \in \mathcal{S}$ and final state $s_B \in \mathcal{S}$. The correct reception of the relevant feedback has to be considered as the most plausible effect. The objective assessment trajectory ($h_0 = s_A$, $h_1 = s_B$) has to be normal and so the most plausible one. The second trajectory ($h_0 = s_A$, $h_1 = s_A$) corresponds to the loss of feedback and it has to be less than normal but more than unusual.

Later on the human performs a selection (action a) that does not modify the machine state. This selection is a slip in the actual state of the machine: s_B. Therefore it is a slip also for a human operator for which $h_1 = s_B$. The objective assessment trajectory becomes ($h_0 = s_A$, $h_1 = s_B$, $h_2 = s_B$), and its plausibility should be reduced to that of a slip (unusual).

If the human assessment has not been updated in step 1 due to a loss of feedback (second trajectory) the assessment is still $h_1 = s_A$. Suppose that in this machine state s_A action a is

totally normal. The second trajectory becomes ($h_0 = s_A$, $h_1 = s_A$, $h_2 = s_C$) and its plausibility should remain unchanged (less than normal but more than unusual).

An exception should be detected and the analysis model should identify the exception explanation in the loss of the relevant feedback.

IV.3 Human assessment estimation, error detection and diagnostic

Possibility Theory is well suited to encode qualitative expert knowledge such as the working assumptions presented in the previous section. This section begins with a short presentation of this theory. Later on the first step of the possibilistic analysis model is presented: starting from the interaction model, natural language knowledge is expressed in terms of possibility degrees: basically, what was "plausible" becomes "possible", what was "less normal" is defined as "less possible", what was "unusual" becomes "far less possible", and what was "very rare" becomes "almost impossible". This section ends with formal computations leading to successive human assessment estimations, the error assessment detection and the exception explanation.

Here the problem deals with uncertainty related to the effects of each occurrence, *i.e.* the human assessment transitions: in situations where no wide enough experiments dataset are available, the corresponding uncertainty cannot be modelled precisely with frequencies leading to transition probabilities. Possibility Theory allows the definition of a model using only the available information about the system, which is however enough rich to build a useful model.

IV.3.1 Possibilistic analysis model

In order to perform the possibilistic analysis the interaction model has to be fully defined, *i.e.* all the effects have to be defined and sorted according to their plausibility. The effect ranking can be encoded defining an appropriate qualitative scale $\mathcal{L}$ and assigning possibility degrees from this scale to the effects. The example of table IV.2 defines the qualitative scale $\mathcal{L}$ as $\{0, \varepsilon, \delta, \lambda, 1\}$ with $0 < \varepsilon < \delta < \lambda < 1$, and assigns possibility degree 1 to normal effects, $\pi(\overline{e}) = 1$, degree λ to less normal effects, $\pi(\widehat{e}) = \lambda$, degree δ to unusual effects $\pi(\underline{e}) = \delta$, and degree ε to very rare effects, $\pi(\tilde{e}) = \varepsilon$. Of course, if a transition is impossible, *i.e.* if function f_e is not defined, corresponding possibility degree is 0. This last modelling step leads to the full definition of a possibilistic hidden Markov process whose states are the successive human assessments, sound framework for human assessment estimation.

The interaction model used in this work has been set up in section IV.2.5, providing definition of occurrence effects and the ranking of those effects. The possibility degrees have still to be assigned to them. Occurrence effects possibility degrees are in the qualitative scale $\mathcal{L} = \{0, \varepsilon, \lambda, 1\}$ with $0 < \varepsilon < \lambda < 1$. The following notations defining classes of effects are useful to assign possibility degrees:

- $e_{0c} \doteq$ the correct initialization – nominal effect of an "initialization" occurrence, $\pi(e_{0c}) = 1$;
- $e_{0w} \doteq$ a wrong initialization – non-nominal effect of an "initialization" occurrence, $\pi(e_{0w}) = \varepsilon$;
- $e_f \doteq$ the correct reception of a feedback – nominal effect of an "automated behaviour" occurrence: $\pi(e_f) = 1$;
- $e_l \doteq$ a missed feedback – non-nominal effect of an "automated behaviour" occurrence: $\pi(e_l) = \lambda$;

- $e_n \doteq$ a generic occurrence effect considered as normal, other than e_{0c} and e_f (for instance a normal selection): $\pi(e_n) = 1$;
- $e_s \doteq$ the arising of a slip – nominal effect of a selection occurrence, $\pi(e_s) = \varepsilon$;
- $e_m \doteq$ a mistake – nominal effect of a selection or "absence of selection" occurrence, $\pi(e_m) = \varepsilon$.

The human assessment of the initial state is uncertain, but is likely to coincide with the real one, *i.e.* a correct initialization is more plausible than a wrong one: $1 = \pi(e_{0c}) > \pi(e_{0w}) = \varepsilon$. Moreover, the good reception of a feedback is more plausible than a loss of one of them, which is more likely to happen than a slip or a mistake: $\pi(e_f) = 1 > \pi(e_l) = \lambda > \pi(e_s) = \pi(e_m) = \varepsilon$.

Before starting the human assessment estimation, it is important to understand the link between possibility degrees of effects, and possibilistic system dynamics. An effect encodes the manifestation of an occurrence v' and the transition from current human assessment $h \in \mathcal{S}$ to the next one $h' \in \mathcal{S}$: an effect is then plausible when occurrence v' and assessment h' are plausible knowing previous one h. That leads to the following equation defining the joint possibility distribution over next assessment and occurrence:

$$\pi\left(v', h' \mid h\right) = \pi\left(f_e\left(h, v', h'\right)\right) \tag{IV.2}$$

i.e. the possibility degree of the effect of an occurrence v' is the joint possibility degree of the this occurrence (v') and the next assessment (h') associated with the effect, knowing current assessment (h). For the initialization occurrence (beginning of the human-machine interaction), as no previous assessment is available, equation IV.2 becomes simply: $\pi\left(v_0, h_0\right) = \pi\left(f_e\left(v_0, h_0\right)\right)$. Note that initialization occurrence is artificially added to define initial uncertainty: it is thus considered as a totally normal occurrence: $\pi(v_0) = 1$. Then $\pi\left(v_0, h_0\right) = \min\left\{\pi(v_0), \pi\left(h_0 \mid v_0\right)\right\} = \pi\left(h_0 \mid v_0\right)$, *i.e.*

$$\pi\left(h_0 \mid v_0\right) = \pi\left(f_e\left(v_0, h_0\right)\right). \tag{IV.3}$$

Recall that the possibility degree 0 is of course assigned to each triplets (h, v', h') for which f_e is not defined: no such effects have been declared possible by experts.

As explained around equation IV.1 for each assessment h, there exists an occurrence v' and a human assessment h' entirely possible: there is always a couple (v', h') such as $\pi\left(v', h' \mid h\right) = 1$. Thus $\pi\left(v', h' \mid h\right)$ defines actually a joint possibility distribution, as possibilistic normalization is naturally ensured.

IV.3.2 Human assessment estimation

Initial possibility distribution over human assessment h is denoted by $\pi_0\left(h\right) = \pi\left(h_0 = h \mid v_0\right)$ which depends on initial occurrence v_0 defining initial machine state s_0. It encodes the initial estimation of the human assessment about the machine state, using assumptions of the interaction model: knowing the initial machine state, as part of data available for the observer, a positive possibility degree is assigned to each potential human operator initial assessment of the machine state, using equation IV.3.

Given occurrence v_{t+1}, the possibilistic dynamics of human belief (assessment), *i.e.* the possibility degree of each assessment transition, is summed up in the transition function $(h, h') \mapsto T_{t+1}\left(h, h'\right) = \pi\left(h_{t+1} = h' \mid h_t = h, v_{t+1}\right)$.

Definition IV.3.1 (*Transition function*)

As $\pi\left(v_{t+1}, h' \mid h\right)$ is given by the interaction model, using equation IV.2, and as v_{t+1} is

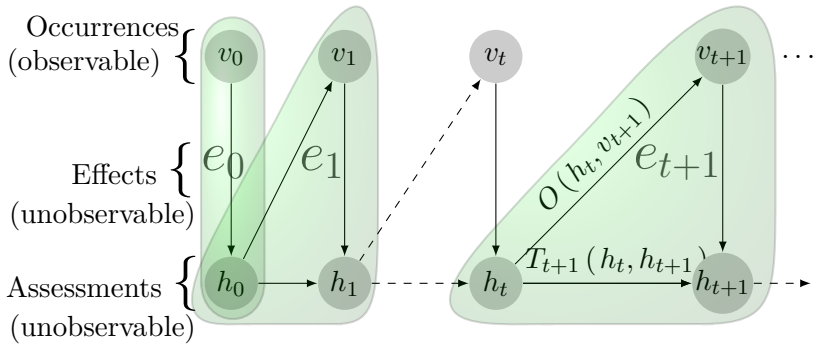


Figure IV.5 – *Dynamic Bayesian Network of the problem: relations between occurrences (v_t), and corresponding effects (e_t) on assessments evolution (h_t).*

known, T_{t+1} is computed with normalization

$$T_{t+1}(h, h') = \begin{cases} 1 \text{ if } h' \in \underset{h' \in \mathcal{S}}{\operatorname{argmax}}\, \pi(v_{t+1}, h' \mid h) \\ \pi(v_{t+1}, h' \mid h) \text{ otherwise,} \end{cases}$$

which comes directly from possibilistic conditionning I.2.7.

The occurrences sequence $(v_t)_{t\in\mathbb{N}}$ only concerns actual facts (*e.g.* human operator selection, automated behaviour, ...) that are available (fully observable): T_t is then in fact a transition possibility distribution of the non-stationary possibilistic Markov Chain $(h_t)_{t\in\mathbb{N}} \in \mathcal{S}^{\mathbb{N}}$ with initial possibility distribution π_0. This formalism is the same as the one used in the previous chapter for planning under uncertainty [49]. Here, unlike in planning problems, no action has to be chosen in this framework: however hidden state (here human assessment) is inferred in the same way. As successive human assessments constitute states of the Markov process, and are not directly observable, the analysis model is in fact summed up in a possibilistic hidden Markov process illustrated by Figure IV.5.

At occurrence step t, the *possibilistic estimation of the human assessment* is defined by the following possibility distribution:

Definition IV.3.2 (*Possibilistic estimation of the human assessment*)

$$\pi_t(h) = \pi(h = h_t \mid v_0, \ldots, v_t) \tag{IV.4}$$

As illustrated in Figure IV.5, some occurrences (as selections), depends on the previous belief (assessment) of the human operator *i.e.* possibility degree of occurrence v_{t+1} depends on h_t, and this dependence is defined bu the observation function $(h, v') \mapsto O(h, v') = \pi(v_{t+1} = v' \mid h_t = h)$: this information is used to update the current estimation π_t.

Definition IV.3.3 (*Observation function*)

As $\pi(v_{t+1}, h' \mid h)$ is given by equation IV.2, observation function is given by marginalization $O(h, v_{t+1}) = \max_{h' \in \mathcal{S}} \pi(v_{t+1}, h' \mid h)$

The set of assessments h such that $\pi_t(h) = 1$ is denoted by H_t^* (human assessments of the machine state that are totally possible). At step t the next occurrence v_{t+1} can contradict estimation of the assessment: an exception arises when the possibility degree of this occurrence v_{t+1} knowing one of the most plausible $h \in H_t^*$ is less than the same possibility degree knowing another human assessment $\tilde{h} \notin H_t^*$, $O(h, v_{t+1}) \leqslant O(\tilde{h}, v_{t+1})$, and less than current estimation of the latter $O(h, v_{t+1}) \leqslant \pi_t(\tilde{h})$. More generally, information given by next occurrence v_{t+1} is used to update estimation, using next theorem.

Theorem 27

Human assessment estimation update, $\pi'_t(h) = \pi(h_t = h \mid v_0, v_1, \ldots, v_{t+1})$*, can be computed as follow:*

$$\pi'_t(h) = \begin{cases} 1 & \text{if } h \in \underset{\mathcal{S}}{\operatorname{argmax}} \min\{O(h, v_{t+1}), \pi_t(h)\}, \\ \min\{O(h, v_{t+1}), \pi_t(h)\} & \text{otherwise.} \end{cases} \quad \text{(IV.5)}$$

This equation does not modify assessment estimation ($\pi_t \equiv \pi'_t$) *if the following sufficient condition holds:* $\forall h \in H_t^*$, $O(h, v_{t+1}) = 1$ *and* $\forall h \notin H_t^*$, $\pi_t(h) \leqslant O(h, v_{t+1})$.

Proof: Qualitative possibilistic conditionning I.2.7 provides equation IV.5, observation function playing the role of $\pi(y_{obs} \mid x)$, occurrence v_{t+1} the observation role, and assessment h the role of the state. Now, if $\forall h \in H_t^*$, $O(h, v_{t+1}) = 1$, as $\pi_t(h) = 1$, the first case occurs and $\pi'_t(h) = \pi_t(h) = 1$. The same equation holds $\forall h \notin H_t^*$: as $\pi_t(h) < 1$, the second case ("otherwise") occurs, and as $\pi_t(h) \leqslant O(h, v_{t+1})$, $\pi'_t(h) = \min\{\pi_t(h), O(h, v_{t+1})\} = \pi_t(h)$. ■

It is now possible to formally define an exception: **an exception is detected at step** $t+1$ **when update IV.5 makes the possibility degree of an assessment** h **different from the actual state** s_t **and such that** $\pi_t(h) < 1$ **become** $\pi'_t(h) = 1$, *i.e.* when the occurrence underlying the estimation update leading to π'_t, contradicts the previous estimation π_t and make an assessment different from the actual machine state becomes totally possible.

Once π'_t is computed using observation function $O(h, v_{t+1})$, estimation π_{t+1} is easily deducted using π' and transition function T_{t+1}:

Theorem 28

Assume $\pi'_t(h) = \pi(h_t = h \mid v_0, \ldots, v_{t+1})$ *is available: next possibilistic estimation of the human assessment* $\pi_{t+1}(h) = \pi(h_{t+1} = h \mid v_0, \ldots, v_{t+1})$ *is computed as follow, propagating assessment estimation over one step of the possibilistic Markov process:*

$$\pi_{t+1}(h') = \max_{h \in \mathcal{S}} \min\{T_{t+1}(h, h'), \pi'_t(h)\} \quad \text{(IV.6)}$$

Proof: As $T_{t+1}(h, h') = \pi(h_{t+1} = h' \mid h_t = h, v_{t+1})$, then $\pi_{t+1}(h') = \max_{h \in \mathcal{S}} \pi(h_{t+1} = h', h_t = h \mid v_0, \ldots, v_{t+1})$

$$\begin{aligned} &= \max_{h \in \mathcal{S}} \min\{T_{t+1}(h, h'), \pi(h_t = h \mid v_0, \ldots, v_{t+1})\} \\ &= \max_{h \in \mathcal{S}} \min\{T_{t+1}(h, h'), \pi'_t(h)\}. \end{aligned}$$

■

IV.3.3 Exception explanation

As all possible non-observable trajectories are recorded as described in section IV.2.4, set of possible effects (respectively assessments) trajectories at step m, $\mathcal{E}_m$, (respectively $\mathcal{H}_m$) is available removing assessments (respectively effects): they are used in case of exception to provide, if existing, an exception explanation. The explanation search uses operator leximin [51]. Operator leximin is a function similar to the minimum that may discriminate trajectories whose minimum possibility degree is the same. The idea is to compare effects trajectories at first via the simple minimum of effect possibility degrees, *i.e.* for a possible effects trajectory $(e_0, \ldots, e_m) \in \mathcal{E}_m$, via $\min_{t=0}^{m} \pi(e_t)$.

Using equality IV.2, it appears that the minimum of effects possibility degrees corresponds to the joint possibility degree of the observed occurrences trajectory $(v_0, \ldots, v_m) \in \mathcal{V}_m$ and

the assessments trajectory $(h_0, \ldots, h_m) \in \mathcal{H}_m$ such that, $\forall 0 \leqslant t \leqslant m$, $e_t = f_e(h_t, v_{t+1}, h_{t+1})$ (removing "h_{-1}" for $t = 0$): $\min_{t=0}^{m} \pi(e_t) = \min_{t=0}^{m} \pi\left(v_{t+1}, h_{t+1} \mid h_t\right)$ (removing "$|h_{-1}$" for $t = 0$) which is equal to $\pi(h_0, \ldots, h_m, v_0, \ldots, v_m)$.

When an exception occurs, effects trajectories which maximize this quantity are the most plausible explainations, and associated assessment trajectories can inform the observer about what the human operator thought. If more than one effects trajectory maximize this quantity, leximin operator can help to discriminate them. It counts in each trajectory the multiplicity of effects which have the minimum possibility degree and chose as lexi-minimal the effects trajectory with largest multiplicity (and then the most plausible ones are the effects trajectories with lowest multiplicity). If some trajectories have the same number of effects having the minimal possibility degree, leximin operator remove these effects, and counts multiplicity of the new minimal possibility degree, etc.

Definition IV.3.4 (*Leximin*)

Consider finite sequences of elements from a totally ordered space Π. *A sequence* $(\pi_1, \ldots, \pi_m) \in \Pi^m$ *is lower than a sequence* $(\tilde{\pi}_1, \ldots, \tilde{\pi}_m) \in \Pi^m$ *in* leximin *sense, i.e.* $(\pi_1, \ldots, \pi_m) >^{\text{leximin}} (\tilde{\pi}_1, \ldots, \tilde{\pi}_m)$ *if and only if, assuming that elements* $\pi \in \Pi$ *are classified in increasing order in both sequences,* $\exists 1 \leqslant i \leqslant m$ *such that* $(\pi_1, \ldots, \pi_i) = (\tilde{\pi}_1, \ldots, \tilde{\pi}_i)$ *and* $\pi_{i+1} > \tilde{\pi}_{i+1}$.

For a deeper investigation of the leximin operator, see [51] which explains how to find most credible trajectories in max leximin sence, *i.e.* $\underset{\mathcal{E}_m}{\operatorname{argmax}} \underset{t=0}{\overset{m}{\operatorname{leximin}}} \pi(e_t)$, using dynamic programming.

Finally, as stated when defining our particular interaction model in section IV.2.5, not perceiving $n < n_{max}$ feedbacks is more likely to happen than not perceiving $n + 1$ feedbacks. Moreover not perceiving $n < n_{max}$ feedbacks (denoted by e_l^n) is more likely to happen than slips, *i.e.* $\forall n < n_{max}$

$$\pi({e_l}^n) > \pi(e_s) \text{ and } \pi({e_l}^n) > \pi({e_l}^{n+1}).$$

This condition is taken into account for exception explanation as leximin naturally encodes it. However, assessment estimation as presented here, suffers from *drowning effect* due to the min operator [51]. It is possible to get around this issue redefining $\mathcal{L}$ and min operator: $\mathcal{L} = \{0, \varepsilon, \lambda_{n_{max}}, \ldots, \lambda_2, \lambda_1, 1\}$, and $\min\{\lambda_t, \lambda_j\} = \lambda_{t+j}$, with $\lambda_{t+j} = \lambda_{n_{max}}$ if $i + j \geqslant n_{max}$ (min operator remains the classical one for other values). The use of the leximin operator for h may solve this issue as well (but more discriminating).

In the following sections a mock-up example and a real case example are detailed, illustrating and validating the possibilistic analysis model based on the particular interaction model presented in section IV.2.5.

IV.4 Interacting with a three-state machine

This example is meant to detail the estimation of assessment h as well as the error detection and explaination over few occurrences: human selections and automated state changes. Let us consider a machine with only one state variable with three values L, M and H (for respectively "low", "medium" and "high"), two possible selections *selU* and *selD* (for respectively "up" and "down"), and three automated state changes: *acL* (machine state becomes low), *acM* (machine state becomes medium) and *acH* (machine state becomes high). The interaction model is described in the enhanced logic table IV.3: an additional row "DETAILS" appears, containing for each column representing a non-nominal effect, a reference to the column number of the corresponding nominal effect. A description of each effect is given as well in the lower part of

columns		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SITUATION																
v'																
Selection	$selU$	1	1	1												
	$selD$				1	1	1									
Autom. change	acL							1			1	1				
	acM								1				1	1		
	acH									1					1	1
h	L	1			1								1		1	
	M		1			1					1					1
	H			1			1					1		1		
BEHAVIOUR																
h'	L				1	1		1					1		1	
	M	1					1		1		1					1
	H		1	1						1		1		1		
EFFECT		e_n	e_n	e_s	e_s	e_n	e_n	e_f	e_f	e_f	e_l	e_l	e_l	e_l	e_l	e_l
POSSIBILITY		1	1	ε	ε	1	1	1	1	1	λ	λ	λ	λ	λ	λ
DETAILS	from										7	7	8	8	9	9
		Selection Up, from Low to Medium	Selection Up, from Medium to High	Selection Up, no effect because already High	Selection Down, no effect because already Low	Selection Down, from Medium to Low	Selection Down, from High to Medium	Automated behaviour to Low	Automated behaviour to Medium	Automated behaviour to High	Automated behaviour from Medium to Low, but unseen	Automated behaviour from High to Low, but unseen	Automated behaviour from Low to Medium, but unseen	Automated behaviour from High to Medium, but unseen	Automated behaviour from Low to High, but unseen	Automated behaviour from Medium to High, but unseen

Table IV.3 – *Enhanced logic table for the three-state machine: the last row provides a natural language description for each effect.*

the table. Note that columns 7, 8 and 9 represents each three nominal effects: for instance, column 7 represents transitions from $h = L$ (Low), from $h = M$ (Medium) and from $h = H$ (High), to $h' = L$ (Low). Note that the three effects of column 7 lead to the same assessment L (Low). The same is true for columns 8 and 9.

In this section the human assessment estimation evolution is shown for two scenarios consisting of three successive occurrences.

IV.4.1 Two successive selections

Initialization

The initial human assessment of the machine state is denoted by $h_0 \in \mathcal{S} = \{L, M, H\}$. The initial machine state, denoted by $s_0 \in \mathcal{S}$, is equal to M.

First occurrence v_0 encodes initialization $\{s_0 = M\}$ with two possible effects on human assessment (stated by experts in section IV.2.5): a correct initialization $e_{0c} = f_e(v_0, M)$ and a wrong one $e_{0w} = f_e(v_0, L) = f_e(v_0, H)$, which is less plausible. Then, using equation IV.3,

possibility distribution over h_0 is

$$\begin{aligned}\pi_0(h) &= \pi(h_0 = h \mid v_0) \\ &= \pi(f_e(v_0, h)) \\ &= \begin{cases} \pi(e_{0c}) = 1 & \text{if } h = M, \\ \pi(e_{0w}) = \varepsilon & \text{otherwise.} \end{cases}\end{aligned}$$

As three assessments are possible ($\forall h \in \mathcal{S}, \pi_0(h) > 0$), it leads to three possible non-observable trajectories at this initialization step: a good initialization (e_{0c}, M) and two wrong initializations, (e_{0w}, L) and (e_{0w}, H).

Execution of a first *up* selection

After the execution of an *up* selection, machine state becomes $s_1 = H$: the occurrence is then $v_1 = selU$.

If $h_0 = M$ this occurrence v_1 has only one possible effect considered as normal e_n and making the assessment become $h_1 = H$ (see column 2 of table IV.3). Also if $h_0 = L$, occurrence v_1 has the same effect e_n and the new assessment is $h_1 = M$ (see column 1).

On the other hand if $h_0 = H$, occurrence v_1 has only one possible effect considered as unusual: in fact, as the new assessment is unchanged by the *up* selection ($h_1 = H$), this effect is a slip e_s (see column 3).

These effects, encoded by function $(h, h') \mapsto f_e(h, v_1, h')$, can be represented by a matrix whose rows are indexed by current assessments $h = L$, M and then H, and columns by next assessment $h' = L$, M and then H: $\begin{pmatrix} \emptyset & e_n & \emptyset \\ \emptyset & \emptyset & e_n \\ \emptyset & \emptyset & e_s \end{pmatrix}$ where $\emptyset$ is put where $f_e(h, v', h')$ is not defined. Possibility distribution can then be represented as well using equation IV.2: $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \varepsilon \end{pmatrix}$.

Using the definition of the observation function IV.3.3, by maximizing the previous matrix over column index h' (marginalization), $O(h, v_1) = \pi(v_1 \mid h_0 = h) = \begin{pmatrix} 1 & 1 & \varepsilon \end{pmatrix}$ represented by a vector indexed by variable h_0 with assignment L, M and then H. Indeed, this selection effect is considered totally possible except when human operator thinks machine state is H: in this case it has no reason to select *up* as machine state is already the highest (H), and this occurrence is considered as a slip (e_s with $\pi(e_s) = \varepsilon$). As $\pi_0(h) = \begin{pmatrix} \varepsilon & 1 & \varepsilon \end{pmatrix}$ (indexed by h_0), update of π_0 is not necessary, as sufficient condition stated by theorem 27 is satisfied.

Possibilistic transition function, computed from previous matrix using definition IV.3.1, can be expressed by the following matrix (h indexes rows with L, M and then H; h' indexes columns in the same way), and increases deterministically the assessment: $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$.

Next human assessment is represented by variable h_1. Its estimation (definition IV.4) is given by propagation equation IV.6: using a representation with matrices and the "max-min" matrix product $\otimes$ which replaces sum and product of the classical matrix product $\times$ by respectively max and min, this equation becomes

$$\begin{aligned}\pi_1(h_1) &= \max_{h \in \mathcal{S}} \min \{T_1(h, h_1), \pi_0(h)\} \\ &= \begin{pmatrix} \varepsilon & 1 & \varepsilon \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & \varepsilon & 1 \end{pmatrix}\end{aligned}$$

where $h' = h_1$ indexes last vector with assignments L, M and then H. Finally, as T_1 encodes a deterministic transition, only three non-observable trajectories are possible at this step of the process: (e_{0w}, L, e_n, M), (e_{0c}, M, e_n, H) and (e_{0w}, H, e_s, H).

Execution of a second *up* selection

After the execution of a second *up* selection, machine state remains unchanged and then $s_2 = H$. The second occurrence is thus $v_2 = selU$. In this paragraph, vectors are indexed with h from L to H. Estimation update has now to be computed.

As $O(h, v_2) = \begin{pmatrix} 1 & 1 & \varepsilon \end{pmatrix}$, $\min\{\pi_1(h), O(h, v_2)\} = \begin{pmatrix} 0 & \varepsilon & \varepsilon \end{pmatrix}$, and finally, update IV.5 asserts that $\pi_1'(h) = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$. As assessment $h = M$ becomes entirely possible, this update **leads to an exception** because actual machine state is $s_2 = H$. This selection contradicts previous estimation π_1 since the human operator has no reason to select *up* if their assessment of the machine state is H.

As $v_2 = v_1$, transition function $T_2 = T_1$. Propagation equation IV.6 leads to the estimation of the next assessment h_2 from $\pi_1'(h)$, represented by the vector $\begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$, where h indexes this vector with L, M and then H. Using matrices representation of T_2 and "max-min" matrix product $\otimes$,

$$\pi_2(h') = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

where h indexes rows of the matrix, $h' = h_2$ indexes columns and last vector. After the two occurrences v_1 and v_2 the possibilistic model returns a certitude for the human assessment: the human operator assessment for the machine state is H. Deterministic transition function conserves three possible non-observable trajectories: $(e_{0w}, L, e_n, M, e_n, H)$, $(e_{0c}, M, e_n, H, e_s, H)$ and $(e_{0w}, H, e_s, H, e_s, H)$. Associated effects trajectories have the same possibility degree: $\min\{\pi(e_0), \pi(e_1), \pi(e_2)\} = \varepsilon$. However the most credible trajectories can be found using leximin operator: (e_{0w}, e_n, e_n), $(e_{0c}, e_n, e_s) \in \operatorname{argmax}_{\mathcal{E}_3} \text{leximin}\{\pi(e_0), \pi(e_1), \pi(e_2)\}$, *i.e.* the trajectory with a correct initialization and the trajectory beginning with $h_0 = L$ are the most plausible ones. The exception explanations are then a slip at the end, or a wrong initialization.

IV.4.2 Automated state change followed by a selection

This scenario shows the effects of the automated state changes on the human assessments.

Initialization and automated state change

Starting from the same machine state as in the previous scenario, the same initial occurrence $v_0 = \{s_0 = M\}$ occurs, and $\pi_0(h) = \begin{pmatrix} \varepsilon & 1 & \varepsilon \end{pmatrix}$. The same non-observable trajectories are recorded: (e_{0w}, L), (e_{0c}, M) and (e_{0w}, H).

Then machine state automatically goes to H: $v_1 = acH$. Occurrence v_1 can produce two effects: e_f if feedback of this automated state change is well received by human operator, which is totally possible $\pi(e_f) = 1$; and e_l if it is missed, with possibility degree $\pi(e_l) = \lambda$ (see columns 9, 14 and 15 of table IV.3). Variable h' (next assessment) indexing columns, and h (current one) indexing rows, with values L, M and then H, $f_e(h, v_1, h')$ can be written $\begin{pmatrix} e_l & \emptyset & e_f \\ \emptyset & e_l & e_f \\ \emptyset & \emptyset & e_f \end{pmatrix}$, and $\pi(f_e(h, v_1, h')) = \begin{pmatrix} \lambda & 0 & 1 \\ 0 & \lambda & 1 \\ 0 & 0 & 1 \end{pmatrix}$.

Using marginalization IV.3.3, yields the observation function $O(h, v_1) = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ *i.e.* this automated behabiour is entirely possible whatever the human operator assessment. Then estimation update is not necessary as sufficient condition of theorem 27 is satisfied.

Using normalization IV.3.1, transition function is then $T_1(h, h') = \pi(h_1 = h' \mid h_0 = h, v_1) = \begin{pmatrix} \lambda & 0 & 1 \\ 0 & \lambda & 1 \\ 0 & 0 & 1 \end{pmatrix}$. Finally, using equation IV.6 and "max-min" matrix product $\otimes$, $\pi_1(h') = \begin{pmatrix} \varepsilon & 1 & \varepsilon \end{pmatrix} \otimes \begin{pmatrix} \lambda & 0 & 1 \\ 0 & \lambda & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \varepsilon & \lambda & 1 \end{pmatrix}$. For each initial trajectory, except for the one beginning

with $h_0 = H$, next effect can be either e_f or e_l: (e_{0w}, L, e_f, H), (e_{0w}, L, e_l, L), (e_{0c}, M, e_f, H), (e_{0c}, M, e_l, M) and (e_{0c}, H, e_f, H).

Execution of an up selection

Human operator has no reason to execute this selection if their assessment of the machine state is H (estimated as the most plausible assessment). This occurrence $v_2 = selU$ corrects estimation using theorem 27: as previously, $O(h, v_2) = \begin{pmatrix} 1 & 1 & \varepsilon \end{pmatrix}$, indexed by h. Then $\min\{\pi_1(h), O(h, v_2)\} = \begin{pmatrix} \varepsilon & \lambda & \varepsilon \end{pmatrix}$. Finally update 27 concludes that $\pi_1'(h) = \begin{pmatrix} \varepsilon & 1 & \varepsilon \end{pmatrix}$. As $h_1 = H$ is no more the most plausible assessment, while machine state $s_1 = H$, it leads to an exception. Estimation of h_2 is computed thanks to deterministic transition produced by $v_2 = selU$, already used in the previous scenario: $\pi_2(h) = \begin{pmatrix} \varepsilon & 1 & \varepsilon \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & \varepsilon & 1 \end{pmatrix}$, where $\otimes$ is yet the previously defined "max-min" matrix product.

As transition function T_2 is deterministic, number of trajectories remains 5:

- $(e_{0w}, L, e_f, H, e_s, H)$,
- $(e_{0w}, L, e_l, L, e_n, M)$,
- $(e_{0c}, M, e_f, H, e_s, H)$,
- $(e_{0c}, M, e_l, M, e_n, H)$,
- $(e_{0c}, H, e_f, H, e_s, H)$.

Computing $\text{argmax}_{\mathcal{E}_3} \text{leximin}\{\pi(e_0), \pi(e_1), \pi(e_2)\}$, the situation with a correct initialization, followed by a missed feedback and by a normal selection is the best guess of the analysis model: $(e_{0c}, M, e_l, M, e_n, H)$ explains thus the exception.

The current section was meant to get an intuition of the mechanism of the analysis model in estimating successive human assessments, and in detecting and explaining assessment errors. In practice, machines logics are much more complex: next section presents the results of our analysis model when facing a realistic human-machine system.

IV.5 Interacting with flight control and guidance

In this section a real case application is presented: the AutoPilot (AP) of a flight simulator. An experience has been conduced with ten general aviation pilots in this flight simulator. The experience was originally meant to test the soundness of a method to detect dangerous situations called human-machine conflicts, in which the pilot actions are not coherent with the actual state of the machine. Those conflicts are the consequence of human attentional errors. For that reason the data collected in that experience is a valuable source of attentional errors in a realistic setting. This dataset is used in this section to test the possibilistic analysis model for the detection of human attentional errors. For more details about the used dataset, see [111].

As for the three-state machine example presented in section IV.4, the first step is the definition of the logic of the automation and the definition of some human assessment errors.

IV.5.1 System description

The definition of human assessment errors is easily automatized starting from the logic of the automation and using rules stated section IV.2.5. Hereafter are detailed state variables of the machine, possible occurrences and effects.

State variables: s_i

s_1 = AP state (On/Off);

s_2 = AutoTHRust (ATHR) state (On/Off);

s_3 = Airspeed (Underspeed, Normal, Near overspeed, Overspeed). "Underspeed/Overspeed" means that the airspeed is smaller/greater than minimum/maximum speed. Minimum and maximum speed are calculated by the autopilot and they depend on the flight envelope. "Near overspeed" means that the speed is between the maximum speed and the maximum speed minus five knots. "Normal" means that the speed is between the minimum speed and maximum speed minus five knots;

s_4 = Control stick (Actioning, Not actioning);

s_5 = Throttle lever (Actioning, Not actioning).

Occurrences: v

- initialisation:

  v_A = *Initialization* $\doteq$ Start of the experiment and creation of all the initial assessment trajectories;

- selections:

  v_B = *AP button* $\doteq$ Autopilot engagement/disengagement button pressed. The Autopilot is the part of the automation that if switched on is in charge for the control of the pitch, the roll and the yaw of the aircraft, *i.e.* of its attitude;

  v_C = *ATHR button* $\doteq$ Autothrust engagement/disengagement button pressed. The Autothrust is the part of the automation that if switched on is in charge of the control of the thrust;

  v_D = *Control Stick On* $\doteq$ Control stick activation;

  v_E = *Control Stick Off* $\doteq$ Control stick deactivation;

  v_F = *Throttle lever On* $\doteq$ Throttle lever activation;

  v_G = *Throttle lever Off* $\doteq$Throttle lever deactivation;

- automated state changes:

  v_H = *Speed Low* $\doteq$ Airspeed takes value Underspeed and consequent AP disconnection;

  v_I = *Speed Normal* $\doteq$ Airspeed takes value Normal;

  v_J = *Near overspeed* $\doteq$ Airspeed takes value Near overspeed and consequent vertical speed constraint;

  v_K = *Overspeed* $\doteq$ Airspeed takes value Overspeed and consequent AP disconnection;

  v_L = *Trajectory divergence* $\doteq$ divergence between pilot selected trajectory and autopilot executed trajectory that is greater than 250 feet.

Effects: e

- Initialization effects:

  The correct initialization (nominal effect): e_{0c}, with $\pi(e_{0c}) = 1$;

  The wrong initializations (non-nominal effect): e_{0wi}, with $\pi(e_{0wi}) = \varepsilon$. Note that there are many possible initializations: as many as the cardinality of the cartesian product of the state variables. One of those is the correct initialization, and all the others are wrong. For computational issues, in this work, a limited number of possible wrong initializations is taken into account: the initialization may be wrong for only one variable at a time. The cardinality of the initial set of possible assessments is reduced to the number of variables. This reduced initialization set of possible assessments has shown to be rich enough to provide proper detections and explanations for the analysed dataset.

- Automated state changes effects:
  nominal effects, e_f with $\pi(e_f) = 1$

  e_{f1} = Pilot perception of Airspeed change to Underspeed;

  e_{f2} = Pilot perception of Airspeed change to Normal;

  e_{f3} = Pilot perception of Airspeed change to Near overspeed;

  e_{f4} = Pilot perception of Airspeed change to Overspeed;

  e_{f5} = Trajectory divergence greater than 250 feet when the AP is off (the pilot is in charge of the flight level).

  non-nominal effects (lost feedbacks), e_l with $\pi(e_l) = \lambda$

  e_{l1} = Airspeed takes value Underspeed (missed feedback);

  e_{l2} = Airspeed takes value Normal (missed feedback);

  e_{l3} = Airspeed takes value Near overspeed (missed feedback);

  e_{l4} = Airspeed takes value Overspeed (missed feedback);

  e_{l5} = Airspeed takes value Overspeed but just AP disconnection perceived;

  e_{l6} = Airspeed takes value Underspeed but just AP disconnection perceived;

  e_{l7} = Trajectory divergence greater than 250 feet when AP is on (missed feedback).

- Selection effects:
  normal effects, e_n, with $\pi(e_n) = 1$

  e_{n1} = AP/ATHR connection/disconnection in nominal condition;

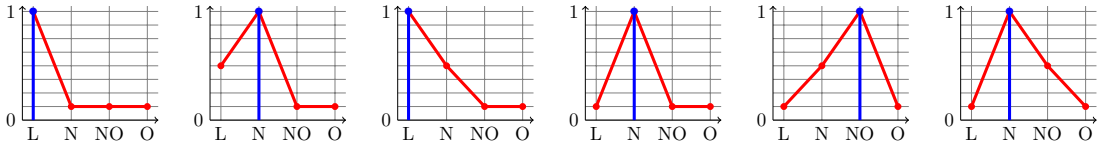
  e_{n2} = Control stick/Throttle lever activation/deactivation in nominal condition.

  slips and mistakes: e_s, with $\pi(e_s) = \epsilon$

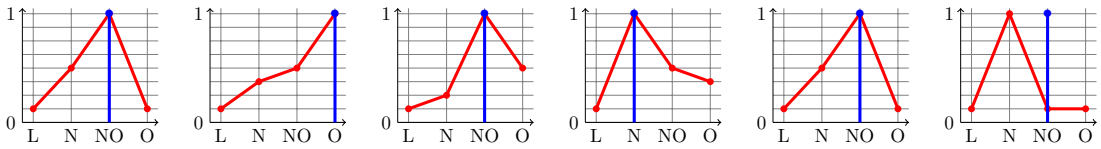
  e_{s1} = AP connection during overspeed/underspeed (slip);

  e_{s2} =Control stick/Throttle lever activation when AP/ATHR On (slip);

  e_{s3} =Trajectory divergence consciously greater than 250 feet and increasing because of AP on (mistake). This effect is defined as a mistake by the designer by hand, so its possibility degree is not automatically generated starting from the logic table and the general assumptions. By that we mean that *the pilot may not voluntarily be aware of the increasing trajectory divergence and that the AP is on (so it is the cause of the divergence) without taking actions, passively accepting that their requests are not executed.*



(a) *Initial* (b) *After 12 occurrences* (c) *After 17 occurrences* (d) *After 22 occurrences* (e) *After 24 occurrences* (f) *After 27 occurrences*



(g) *After 28 occurrences* (h) *After 29 occurrences* (i) *After 30 occurrences* (j) *After 31 occurrences* (k) *After 33 occurrences* (l) *After 34 occurrences*

Figure IV.6 – *Experiment 1: Possibilistic estimation on human assessment of Airspeed (solid red curve), and actual machine state (blue bar). "L" is "Low speed", "N" is "Normal speed", "NO" is "Near Overspeed" and "O" is Overspeed.*

IV.5.2 Experiments

Data generated during the experience have been pre-processed to generate sequences of occurrences (among which are selections). Those occurrence sequences (one sequence for each pilot running the experiment in the flight simulator) have been then processed by our analysis model to automatically detect exceptions, *i.e.* when the objective assessment trajectory is no longer considered as normal. In those cases the exception is analysed: the exception explanation (if any is found) is specified by the model using the description of the relevant human assessment error, *i.e.* the less possible effect of the trajectory explanation, and the triggering occurrence is specified as well. That in textual form[2]:

- if an exception explanation is found it is labelled as an explaned exception: `Exception description: 'Triggering occurrence' because 'exception explanation'`;
- if no exception explanation is found it is labelled as a simple exception: `Exception description: 'Triggering occurrence'`.

Hereafter the analysis performed for two participants of the experience is presented.

Example 1

The sequence of occurrences generated from the data recorded during the experience with the first participant is shown hereafter. A total of 57 occurrences (among which are some selections) have been generated:

'Initialization', 'Control Stick Off', 'Throttle lever On', 'Throttle lever Off', 'Throttle lever On', 'Throttle lever Off', 'Throttle lever On', 'Throttle lever Off', 'Throttle lever On', 'Throttle lever Off', 'Control Stick On', 'Speed Normal', 'Control Stick Off', 'Control Stick On', 'ATHR button', 'Control Stick Off', 'SpeedLow', 'Speed Normal', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'AP button', 'Control Stick Off', 'Near overspeed', 'Control Stick On', 'Control Stick Off', 'Speed Normal', 'Near overspeed', 'Overspeed', 'Near

[2]The message that is automatically generated by our model could later be used to compose specific feedbacks meant to correct the human state assessment. By the way the definition of those feedbacks is out of the scope of this work. Note that the real time version of the algorithm is totally feasible: the computing time for a 15-mn mission is less than one minute.

overspeed', 'Speed Normal', 'AP button', 'Near overspeed', 'Trajectory divergence', 'AP button', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'Control Stick Off', 'Control Stick On', 'Control Stick Off', 'AP button', 'Trajectory divergence', 'AP button', 'Control Stick On', 'Speed Normal', 'Near overspeed', 'Control Stick Off'

The initial machine state is:

s_1 = AP state: Off;

s_2 = ATHR state: On;

s_3 = Airspeed: Underspeed;

s_4 = Control stick: Actioning;

s_5 = Throttle lever: Not actioning.

After the firing of the 25th occurrence, 119.5 seconds from the beginning of the experiment, the analysis model detects an exception:

`Exception description: 'Control stick On when AP on!'.`

After the firing of the 34th occurrence, 772.6 seconds from the beginning of the experiment, the analysis model detects an exception and explains it:

`Exception description: 'Vertical speed divergence > 250 ft, unnoticed'`
`because 'Speed becomes >Vmax-5, but unseen!'.`

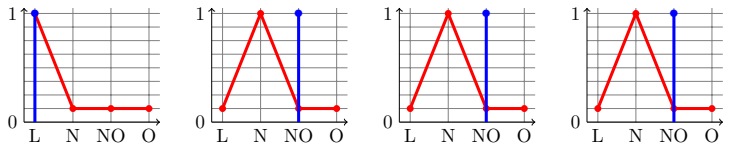
After the firing of the 51st occurrence, 886.1 seconds from the beginning of the experiment, an exception is explained again:

`Exception description: 'Vertical speed divergence > 250 ft, unnoticed'`
`because 'Speed becomes >Vmax-5, but unseen!'.`

It is worth noting that the experimenters [111] reported two human automation conflicts corresponding to the second and third exceptions, and that their findings on those conflicts causes (based on the observation of the data, the video recording and the interview of the pilot) is in agree with the exception explanation provided by the analysis model here.

After the execution of the 57 occurrences, 196 assessment trajectories are considered as possible (with different possibility degrees). The computation time is 30 seconds. Some possibilistic estimations of the human assessment of the machine state variable "Airspeed" are represented in Figure IV.6: the possibility distribution over the human assessment h of the airspeed, $\pi_t(h)$, is indicated by the red curve. This possibilistic evaluation is qualitative, nevertheless in the graphic representation, quantitative values are arbitrarily assigned to the possibility degrees (respecting qualitative ordering) to plot them. The actual machine state s is stated by the blue bar with value 1 on the y-axis: if no exception arises, most possible assessment h should be the actual state (blue bar).

Remember that after the firing of 34 occurrences an exception is detected by the analysis model, which is graphically highlighted in Figure IV.6l: most possible human assessment is no more the real machine state.



(a) *Initial* (b) *After 18 occurrences* (c) *After 46 occurrences* (d) *After 49 occurrences*

Figure IV.7 – *Experiment 2: Possibilistic estimation on human assessment of speed (solid red curve), actual machine state (blue bar).*

Example 2

A sequence of occurrences has been generated from the data recorded during experience with a second participant. A total of 85 occurrences (among which are some human selections) have been generated. Hereafter the beginning of the sequence:

'Initialization', 'Throttle lever On', 'Throttle lever Off', 'ATHR button', 'Throttle lever On' ...

Initial machine state variables are the same as previously, except of control stick, which is here initially actioning.

This second example was chosen because of the second occurrence, 30.2 seconds from the beginning of the experiment. The analysis model detects the following exception:

```
Exception description: 'Throttle lever On when ATHR on!' because 'Wrong
    state initialization'
```

Initially the ATHR is on: operating the throttle lever has no result (this action is considered as a slip). The model explains this slip as the result of a wrong initial assessment: if the pilot initial assessment of the ATHR state was Off, that could explain the execution of this action as nominal. It is worth noting that after giving up this useless action ('Throttle lever Off') the participant deactivated the ATHR ('ATHR button') and she/he started again operating the throttle lever ('Throttle lever On'), probably because she/he had a wrong initial situation assessment (as the found exception explanation) and she/he understood their assessment error.

The analysis model detected also three times the same exception as for the previous participant:

```
Exception description: 'Vertical speed divergence > 250 ft, unnoticed'
    because 'Speed becomes >Vmax-5, but unseen!'
```

Figure IV.7 shows the estimation of the human assessment of the speed, initially, and when these exceptions occur.

IV.6 Conclusion

This chapter proposes a model for the human-machine interaction based on a machine model and expert knowledge on an human assessment error model. The human-machine interaction is modelled as a possibilistic hidden Markov process. Qualitative Possibility Theory has been chosen because it is well suited to handle uncertainty defined by expert knowledge. The proposed possibilistic analysis model provides an estimation of the human assessment of the machine state and detects assessment errors. The analysis model is able to provide also an explanation (diagnosis) when an assessment error is detected.

This process of detection/identification could be used in real time applications in order to inform the human operator of their assessment errors. It can help to make them correct their situation awareness and prevent the execution of other errors.

This work is based on the simplifying assumption that the human operator is certain about the state of the machine: a possible extension of this model may be to drop this assumption using a more refined representation of the human state assessment, as a set of machine states, or an uncertainty measure over the machine states.

This article proposes a model for the human-machine interaction based on a machine model and expert knowledge on an human assessment error model. The human-machine interaction is modelled as a possibilistic hidden Markov process. Qualitative Possibility Theory has been chosen because it is well suited to handle uncertainty defined by expert knowledge. The proposed possibilistic analysis model provides an estimation of the human assessment of the machine state and detects assessment errors. The analysis model is able to provide also an explanation (diagnosis) when an assessment error is detected.

This process of detection/identification could be used in real time applications in order to inform the human operator of their assessment errors. It can help to make them correct their situation awareness and prevent the execution of other errors.

This work is based on the simplifying assumption that the human operator is certain about the state of the machine: a possible extension of this model may be to drop this assumption using a more refined representation of the human state assessment, as a set of machine states, or an uncertainty measure over the machine states.

V A Hybrid Model: Planning in Partially Observable Domains with Fuzzy Epistemic States and Probabilistic Dynamics

While the previous chapters dealt with purely qualitative possibilistic models, this one proposes to use the strength of both probabilistic and possibilistic approaches, in order to solve fully defined factored POMDPs, or partially defined ones. This idea comes from the analysis of the results of the experiments of Chapter III: qualitative modeling may lead to poor strategies for risky problems, or when the frequentist information defining the POMDP is at the heart of planning the problem. Here, a new translation from Partially Observable MDP into Fully Observable MDP is described. Unlike the classical translation (see Section I.1.9), the resulting problem state space is finite, making MDP solvers able to solve this simplified version of the initial partially observable problem: this approach encodes agent beliefs with fuzzy measures over states, leading to an MDP whose system state space is a finite set of epistemic states. The translation is described in a formal manner with semantic arguments. Then actual computations of this transformation are detailed, in order to highly benefits from the factorized structure of the initial POMDP in the the final MDP problem size reduction and structure. Finally size reduction and tractability of the resulting MDP is illustrated on a simple POMDP problem.

V.1 Introduction

The approach proposed here simplifies the belief space of a POMDP problem before solving it. The transformation described leads to a fully observable MDP on a finite number of epistemic states, *i.e.* a problem modeling an agent acting under uncertainty in a fully observable environment [115]. As such a finite state space MDP problem is P-complete [103] this transformation qualifies as a simplification, and any MDP solver can return a policy for this translated POMDP.

More than only a simplification of the initial POMDP problem, the theoretical framework used here for belief states representation formally models an agent's knowledge about the system state. Indeed the proposed translation defines the belief states as possibility distributions over system states $s \in \mathcal{S}$ which represents the fuzzy set of possible system states, as done with π-POMDP models.

The major originality of this work comes from the finiteness of the scale $\mathcal{L}$: indeed it follows that the number of possible belief states over the system state is, as well, finite (smaller than $\#(\mathcal{L}^{\mathcal{S}}) = (\#\mathcal{L})^{\#\mathcal{S}}$, see Equation I.60). What very clearly distinguishes this approach from the classical one is that the classical translation leads to an infinite set of belief states (the

continuous set of all probability distributions over $\mathcal{S}$, or the sequence of reachable belief states from an initial one, see Section I.1.9). The translation described here leads to an MDP whose system state space is the set of possible possibilistic belief states, or *epistemic states*, that is why its state space is finite.

In addition to POMDP simplification and knowledge modelling, this qualitative possibilistic framework offers some interesting properties: the possibilistic counterpart of the Bayes rule leads to a special belief state behaviour. Indeed the agent can possibly change their mind radically and rapidly, as described in Section II.4, experimental section of Chapter II.Moreover, under some conditions, the increased specificity of the belief state distribution is enforced, *i.e.* the knowledge about the current state is non decreasing with time steps (see Section II.4). Finally, in order to fully define the resulting MDP, the translation has to attach a reward function to its states: as a possibilistic belief state distributions constitute the new (epistemic) states of the problem, the definition of the rewards uses the Choquet integral adapted to fuzzy measures. This integral is used with the dual measure of the possibility measure defined by the belief state. The dual measure of a possibility measure, called the *necessity measure* (see Definition I.2.4), and the use of this integral makes the rewards values pessimistic about the potential lack of knowledge described by the associated belief state.

However the number of possibilistic belief distributions, or *fuzzy epistemic states*, grows exponentially with the number of initial POMDP system states. The so called simplification of the problem does not transform the PSPACE POMDP problem into a polynomial one: as the new state space size is exponential in the previous one, the resulting problem is EXPTIME. The proposed translation tries to generate as few epistemic states as possible taking carefully into account potential factorized structures of the initial POMDP.

This chapter begins with the description of the first contribution of this work, which is the translation itself, presented in a formal way. As the resulting state space of the built MDP is too big to make this problem tractable without factorization tricks in practice, the next section details the proper way to preprocess its attributes. Finally, the last section illustrates the power of this approach, describing the translation in practice, and applying it on a simple factored POMDP problem.

V.2 A Hybrid POMDP

As claimed by Zadeh, "most information/intelligent systems will be of hybrid type" [154]: the idea developped here is to use a granulated representation of the agent knowledge using possibilistic belief states instead of probabilistic belief states in the POMDP framework. The first advantage of this granulation is that strategy computation is performed reasoning on a finite set of possibilistic belief states called then epistemic states: the set of all possibility distributions defined over $\mathcal{S}$, denoted by $\Pi_{\mathcal{L}}^{\mathcal{S}}$ is $\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \#\mathcal{L}^{\#\mathcal{S}} - (\#\mathcal{L} - 1)^{\#\mathcal{S}}$, due to the possibilistic normalization (see Equation I.60), while the set of probability distributions over $\mathcal{S}$ is infinite. The π-MDPs studied in the first chapters of this thesis are quite different from the model exposed in this chapter. For instance, Qualitative Possibilistic MDPs do not use quantitative data as probabilities or rewards. Dynamics is described in a purely qualitative possibilistic way. Frequentist information about the problem cannot be encoded: these frameworks are indeed dedicated to situations when the probabilistic dynamic of the studied system is lacking. Moreover, possible values of the reward function are chosen among the degrees of the qualitative possibilistic scale. A commensurability assumption between reward and possibility degrees, i.e. a meaning of why they share the same scale, is needed to use the criteria proposed in these frameworks. Our model bypasses these demands: a real number is assigned to each possibilistic belief (epistemic state), instead of a qualitative utility degree: it represents the reward got by the agent when reaching this belief (in a MDP fashion) as detailed in Section V.2.2. Moreover,

the dynamics of our process is described with probability distributions: approximate probabilistic transition functions between current and next beliefs, or epistemic states, are given section V.2.1. Finally, our model can be solved by any MDP solver in practice: it becomes eventually a classical probabilistic fully observable MDP whose state space is the finite set $\Pi_{\mathcal{L}}^{\mathcal{S}}$.

Here, the term hybrid is used because the beliefs only are defined as possibility distributions, and all variables keep a probabilistic dynamic: the agent reasons based on a possibilistic analysis of the system state (the possibilistic belief, or epistemic state), and transition probability distributions are defined for its epistemic states. Such beliefs are formally defined in Section I.2.5. As the set of the possibilistic beliefs is finite, they define the finite state space of an MDP, whose the probabilistic transitions are defined in the next section. At this step, a Markov process based on epistemic states is thus defined. Finally rewards are defined on epistemic states using the discrete Choquet integral and leading to the definition of the resulting MDP.

Consider that possibility distributions similar to those used to define the initial POMDP are available: a transition distribution, giving the possibility degree of reaching $s' \in \mathcal{S}$ from $s \in \mathcal{S}$ using action $a \in \mathcal{A}$, $\pi(s' \mid s, a) \in \mathcal{L}$; as well as an observation one, giving the possibility degree of observing $o \in \mathcal{O}$, in a system state $s \in \mathcal{S}$ after the use of action $a \in \mathcal{A}$, $\pi(o' \mid s', a) \in \mathcal{L}$. Indeed, this work is devoted to two kinds of practical problems. On the one hand real problems modeled as POMDPs are often intractable: our granulated approach is in this case a simplification of the initial POMDP, and possibility distributions are computed from the POMDP probability distributions, using a possibility-probability transformation [63]. On the other hand, some problems lead to POMDPs with partially defined probability distributions: some estimated probabilities have no strong guarantees. A more faithful representation is given with possibility distri- butions modeling the inherent imprecision, defining transition and observation possibility distributions.

V.2.1 Set transitions

First, we use here some notations: the transition probability distribution is denoted by $T(s, a, s') = \mathbf{p}(s' \mid s, a)$, and the observation probability distribution by $O(s', a, o') = \mathbf{p}(o' \mid s', a)$. If the agent selects action $a \in \mathcal{A}$ in the epistemic state $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, the next epistemic state depends only on the next observation, as highlighted by possibilistic belief update (see Theorem 14). The probability distribution over observations conditionned on the reached state is part of the POMDP definition via the observation function O. The probability distribution over observations conditionned on previous state is obtained using transition function T:

$$\mathbf{p}(o' \mid s, a) = \sum_{s' \in \mathcal{S}} O(s', a, o') \cdot T(s, a, s').$$

This distribution and the possibilistic belief β about the system state, can lead to an approximate probability distribution over the next observations. Indeed, a probability distribution over the system state, $\overline{\beta} \in \mathbb{P}_{\mathcal{S}}$, can be derived from β using an extension of the Laplace principle. Then approximate distribution over $o' \in \mathcal{O}$ is defined as

$$\mathbf{p}(o' \mid \beta, a) = \sum_{s \in \mathcal{S}} \mathbf{p}(o' \mid s, a) \cdot \overline{\beta}(s). \tag{V.1}$$

Finally, summing over concerned observations, the transition probability distribution over epistemic states is defined as

$$\mathbf{p}\left(\tilde{\beta} \,\middle|\, \beta, a\right) = \sum_{o' \mid u(\beta, a, o') = \tilde{\beta}} \mathbf{p}(o' \mid \beta, a). \tag{V.2}$$

A way to construct a probability distribution $\overline{\beta}$ from a possibility one β is the use of the pignistic transformation [52] minimizing the arbitrariness in the translation: numbering system states with the order induced by distribution β, $1 = \beta(s_1) \geqslant \beta(s_2) \geqslant \ldots \geqslant \beta(s_{\#\mathcal{S}+1}) = 0$, with $s_{\#\mathcal{S}+1}$ an artificial state such that $\pi(s_{\#\mathcal{S}+1}) = 0$ introduced to simplify the formula,

$$\overline{\beta}(s_i) = \sum_{j=i}^{\#\mathcal{S}} \frac{\beta(s_j) - \beta(s_{j+1})}{j} \tag{V.3}$$

Note that this probability distribution corresponds to the center of gravity of the probability distributions family induced by the possibility measure defined by distribution β [63], and respects the Laplace principle of Insufficient Reason (ignorance leads to uniform probability).

Although possibilistic belief states were so far defined in a qualitative way, degrees of $\mathcal{L}$ are considered as numerical in this section and the following: the section about factorization will make it clear that possibility distributions can be computed from T and O if the sole purpose is to simplify the POMDP. Numerical values are then used to compute the observation probability distribution here, and in order to aggregate rewards according to the current epistemic state in the next section.

V.2.2 Reward aggregation

After the transition function, it remains to assign a reward to each epistemic state: in the classical probabilistic translation, the reward assigned to a belief state b is the reward expectation according to the probability distribution b: $\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s)$. Here, the agent knowledge is represented with a possibility distribution β: it sums up the frequentist uncertainty of the problem, and imprecision due to the possibilistic discretization and/or due to partial ignorance about actual probability distributions defining the situation. A way to define a reward being pessimistic about these imprecisions is to aggregate the reward using the dual measure of the possibility distribution, and the *Choquet integral*.

The dual measure of a possibility measure $\Pi : 2^{\mathcal{S}} \to \mathcal{L}$ is called *necessity measure* and is denoted by N. This measure is defined by $\forall A \subseteq \mathcal{S}$, $N(A) = 1 - \Pi(\overline{A})$ where $\overline{A}$ is the complementary set of $A : \overline{A} = \mathcal{S} \setminus A$. We use now the notation $\mathcal{L} = \{ l_1 = 1, l_2, l_3, \ldots, 0 \}$. For a given action $a \in \mathcal{A}$, reward values, $\{ r(s,a) \mid s \in \mathcal{S} \}$ are denoted by $\{ r_1, r_2, \ldots, r_k \}$ with $r_1 > r_2 > \ldots > r_k$, with $k \leqslant \#\mathcal{S}$. An artificial value $r_{k+1} = 0$ is also introduced to simplify the formulae.

Discrete Choquet integral of the reward function against necessity measure N [2] is defined

as follows:

$$
\begin{aligned}
Ch(r,N) = & \sum_{i=1}^{k}(r_i - r_{i+1}) \cdot N(\{\, r(s,a) \geqslant r_i \,\}) && \text{(V.4)}\\
= & \sum_{i=1}^{k}(r_i - r_{i+1}) \cdot [1 - \Pi(\{\, r(s,a) < r_i \,\})]\\
= & \sum_{i=1}^{k}(r_i - r_{i+1}) \cdot \left(1 - \max_{s|r(s,a)<r_i} \pi(s)\right)\\
= & \; r_1 - r_{k+1} - \sum_{i=1}^{k}(r_i - r_{i+1}) \cdot \max_{s|r(s,a)<r_i} \pi(s)\\
= & \; r_1 - (r_1 - r_2) \cdot \max_{s|r(s,a)<r_1} \pi(s) - \ldots\\
& -(r_{k-1} - r_k) \cdot \max_{s|r(s,a)<r_{k-1}} \pi(s) - r_k \cdot \max_{s|r(s,a)<r_k} \pi(s)\\
= & \sum_{i=1}^{\#\mathcal{L}-1} (l_i - l_{i+1}) \cdot \min_{s|\pi(s)\geqslant l_i} r(s) && \text{(V.5)}
\end{aligned}
$$

More on possibilistic Choquet integrals can be found in [38, 60].

This reward aggregation using the necessity measure leads to a pessimistic estimation of the reward: as an example, the reward $\min_{s\in\mathcal{S}} r(s,a)$ is assigned to the total ignorance.

Note that, if the necessity measure N is replaced by a probability measure $\mathbb{P}$, *e.g* as the one induced by probability distribution β using V.3, Choquet integral coincides with the expected reward based on $\overline{\beta}$. This could be a good aggregation choice as well, but more optimistic than the one described above. The most optimistic way to aggregate the reward is to compute the Choquet integral with the possibilistic measure Π induced by distribution β, rather than with necessity one N, but this is not detailed here.

V.2.3 MDP with epistemic states

This section summarizes the complete translation using final equations of the previous sections. This translation takes for input a POMDP: $\langle \mathcal{S}, \mathcal{A}, T, \mathcal{O}, O, r\rangle$ and returns an epistemic states based MDP: $\langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{T}, \tilde{r}\rangle$ with

- $\tilde{\mathcal{S}} = \Pi_{\mathcal{L}}^{\mathcal{S}}$;
- $\tilde{T}$, such that $\forall(\beta,\tilde{\beta}) \in (\Pi_{\mathcal{L}}^{\mathcal{S}})^2$, $\forall a \in \mathcal{A}$
  $\tilde{T}(\beta, a, \tilde{\beta}) = \mathbf{p}\left(\tilde{\beta} \,\middle|\, \beta, a\right)$ using V.1 and V.2;
- $\tilde{r}(a,\beta) = Ch(r(a,.), N_\beta)$, using Equation V.5 and where N_β is the necessity measure computed from β.

Finally, as in the probabilistic framework (see Section I.1.4), the criterion of this MDP is the expected total reward:

$$\mathbb{E}_{(\beta_t)\sim\tilde{T}}\left[\sum_{t=0}^{+\infty} \gamma^t \tilde{r}(\beta_t, d_t)\right].$$

While the resulting state space is finite, only really small POMDP problems can be solved with this translation without computation tricks. Indeed, $\Pi_{\mathcal{L}}^{\mathcal{S}}$ grows exponentially with the number of system states (see Equation I.60), which makes the problem intractable even for state of the art MDP solvers.

V.3 Benefit from factorization

This section carefully derives a tractable MDP problem from a factored POMDP: the resulting MDP is equivalent to the former translation, but some factorization and computational tricks are described here to reduce its size and to fit to the factorized structure. First, the definition of a factored POMDP is quickly exposed, followed by some dependency notations helpful for describing how distributions are dealt with. Next, a classification of the state variables is made to strongly adapt computations according to the nature of the state. Then follows the definition of possibility distributions, and the description of the use of the possibilistic Bayes rule in practice ends this section.

V.3.1 Factored POMDP

Partially Observable Markov Decision Processes can be defined in a factorized way.

- state space $\mathcal{S} = s_1 \times \ldots \times s_m$ with $\forall j \in \{1, \ldots, m\}$, s_j boolean variable. The set of boolean state variables is denoted by $\mathbb{S} = \{s_1, \ldots, s_m\}$;
- observation space $\mathcal{O} = o_1 \times \ldots \times o_n$ with $\forall i \in \{1, \ldots, n\}$, o_i boolean variable. In the same way as to state variables, the set of observation variables is denoted by $\mathcal{O} = \{o_1, \ldots, o_n\}$;
- action space $\mathcal{A}$, a finite set of actions $a \in \mathcal{A}$.

Note that a problem with non boolean variables can be easily reduced to such a problem with the boolean variables assumption. For simplicity, and as state $s_j \in \mathcal{S}$ and observation $o_i \in \mathcal{O}$ notations are no longer reused in this chapter, only variables are denoted with these letters from now: $s \in \mathbb{S}$ and $o \in \mathbb{O}$.

Non-primed variables correspond to the current time step, and primed variables to the next time-step. This notation is also used for sets of variables: $\mathbb{S}'$ is the set of next state variables and $\mathbb{O}'$ the set of next observable ones. The factorized description continues with following probability distributions:

- $\forall j \in \{1, \ldots, m\}$, $\forall a \in \mathcal{A}$, a transition function is defined:
$$T_j^a(s_1, \ldots, s_m, s'_j) = \mathbf{p}\left(s'_j \,\middle|\, s_1, \ldots, s_m, a\right);$$
- One observation function is also given for each observation variable: $\forall i \in \{1, \ldots, n\}$, $\forall a \in \mathcal{A}$,
$$O_i^a(s'_1, \ldots, s'_m, o'_i) = \mathbf{p}\left(o'_i \,\middle|\, s'_1, \ldots, s'_m, a\right);$$
- and reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

These definitions lead to the following observations: $\left\{s'_j\right\}_{j \in \{1,\ldots,m\}}$ are post-action independent, and $\{o'_i\}_{j \in \{1,\ldots,n\}}$ post-transition independent.

V.3.2 Notations and Observation Functions

Transitions of the final MDP make it more handy if each variable depends on only few previous variables: the procedure to avoid blocking such simplifications brought by the structure of the initial POMDP during the translation, needs the following notations. In practice, for each $i \in \{1, \ldots, n\}$ not all state variables influence observation variable o'_i; similarly, for each $j \in \{1, \ldots, m\}$, not all current state variables influence next state variable s'_j:

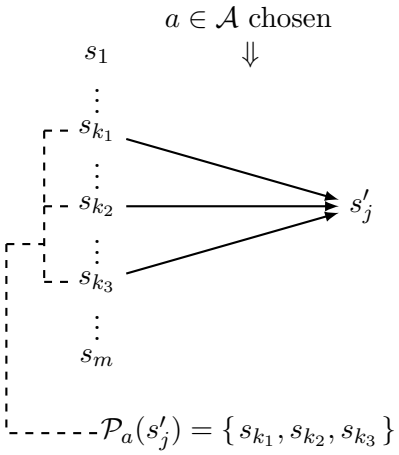


Figure V.1 – *For action $a \in \mathcal{A}$, only tree variables influence variable s'_j in this Bayesian network: s_{k_1}, s_{k_2} and s_{k_3}, which constitute $\mathcal{P}_a(s'_j)$.*

- for each action $a \in \mathcal{A}$, observation variable o'_i depends on some state variables which are denoted by

$$\mathcal{P}_a(o'_i) = \left\{ s'_j \in \mathbb{S}' \text{ s.t. } o'_i \text{ depends on } s'_j \text{ when } a \text{ applied} \right\}$$

They are called *parents* as they appears as "parents nodes" in a dynamic Bayesian network [42] illustrating dependencies of the process.

- as well, for each action $a \in \mathcal{A}$, probability distribution of next state variable s'_j depends on some current ones, denoted by

$$\mathcal{P}_a(s'_j) = \left\{ s_k \in \mathbb{S} \text{ s.t. } s'_j \text{ depends on } s_k \text{ when } a \text{ applied} \right\}$$

and illustrated in Figure V.1.

Now, let us define parents whatever the chosen action: $\forall\ i = 1, \ldots, n$,

$$\mathcal{P}(o'_i) = \cup_{a \in \mathcal{A}} \mathcal{P}_a(o'_i) \subseteq \mathbb{S}'$$

and $\forall j = 1, \ldots, m$,

$$\mathcal{P}(s'_j) = \cup_{a \in \mathcal{A}} \mathcal{P}_a(s'_j) \subseteq \mathbb{S}$$

It leads to the following rewriting of probability distributions:

$$T^a_j(\mathcal{P}(s'_j), s'_j) = \mathbf{p}\left(s'_j \,\middle|\, \mathcal{P}(s'_j), a \right)$$

and

$$O^a_i(\mathcal{P}(o'_j), o'_i) = \mathbf{p}\left(o'_i \,\middle|\, \mathcal{P}(o'_j), a \right).$$

Following subset of $\mathbb{S}$ is useful to specify observation dynamic:

$$\mathcal{Q}(o'_i) = \left\{ s_k \in \mathbb{S} \text{ s.t. } \exists s'_j \in \mathcal{P}(o'_i) \text{ s.t. } s_k \in \mathcal{P}(s'_j) \right\} = \cup_{s'_j \in \mathcal{P}(o'_i)} \mathcal{P}(s'_j) \subseteq \mathbb{S}$$

and is illustrated in Figure V.2.

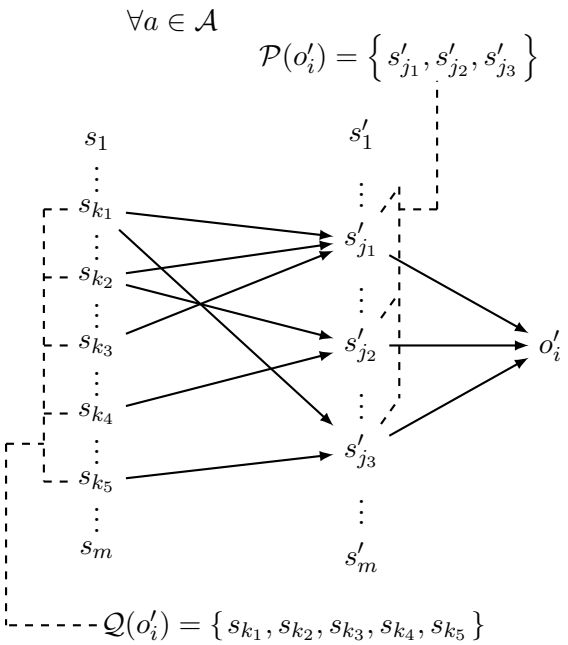


Figure V.2 – *Whatever the action $a \in \mathcal{A}$, only five state variables influence variable o'_i in this Bayesian network: s_{k_1}, s_{k_2}, s_{k_3}, s_{k_4}, s_{k_5} which constitute $\mathcal{Q}(o'_i)$.*

In order to simplify notations, and as it causes no confusion, $\mathbb{S}$, $\mathcal{P}(o'_i)$, $\mathcal{P}(s'_i)$ and $\mathcal{Q}(o'_i)$ designate as well a set of variables, or a vector comprised of these variables (with an arbitrary order). Distribution over $\mathcal{P}(o'_i)$ assignments benefits from previous rewritings:

$$\mathbf{p}\left(\mathcal{P}(o'_i) \mid \mathbb{S}, a\right) = \prod_{s'_j \in \mathcal{P}(o'_i)} T_j^a(\mathcal{P}(s'_j), s'_j) = \prod_{s'_j \in \mathcal{P}(o'_i)} \mathbf{p}\left(s'_j \,\middle|\, \mathcal{P}(s'_j), a\right) = \mathbf{p}\left(\mathcal{P}(o'_i) \mid \mathcal{Q}(o'_i), a\right) \quad \text{(V.6)}$$

Observation probability distributions, knowing previous state variables, are then defined $\forall i = 1, \ldots, n$

$$\mathbf{p}\left(o'_i \mid \mathcal{Q}(o'_i), a\right) = \sum_{v \in 2^{\mathcal{P}(o'_i)}} \mathbf{p}\left(o'_i \mid v, a\right) \cdot \mathbf{p}\left(v \mid \mathcal{Q}(o'_i), a\right) \quad \text{(V.7)}$$

Therefore a possibilistic belief defined on $2^{\mathcal{Q}(o'_i)}$ is enough to get the approximate probability distribution of an observation variable, Equation V.1: such an epistemic state leads via transformation V.3 to a probability distribution $\overline{\beta}$ over $2^{\mathcal{Q}(o'_i)}$. Finally, the approximate probability distribution of the observation variable i, factored counterpart of former equation V.1, is:

$$\mathbf{p}\left(o'_i \mid \beta, a\right) = \sum_{v \in 2^{\mathcal{Q}(o'_i)}} \mathbf{p}\left(o'_i \mid v, a\right) \cdot \overline{\beta}(v). \quad \text{(V.8)}$$

V.3.3 State variables classification

State variables $s \in \mathbb{S}$ do not play the same role in the process: as already studied in the literature [100], some variables can be visible for the agent, and namely this *mixed-observability* leads to important computational simplifications. Moreover, some variables do not affect observation variables, and factorization of the POMDP is then easily transmitted to the epistemic state based MDP. Finally, using rewrittings of previous sections, useless computations are highlighted.

- A state variable s_j is said to be **visible**, if $\exists o_i \in \mathbb{O}$, observation variable, such that $\mathcal{P}(o'_i) = \left\{ s'_j \right\}$ and $\forall a \in \mathcal{A}$, $\mathbf{p}\left(o'_i \mid s'_j, a\right) = \mathbb{1}_{\{o'_i = s'_j\}}$ *i.e.* if $o'_i = s'_j$ almost surely. The set of visible state variables is denoted by $\mathbb{S}_v = \{ s_{v,1}, s_{v,2}, \ldots, s_{v,m_v} \}$;

Observation variables corresponding to visible state variables can be removed from the set of observation variables: the number of observation variables becomes $\tilde{n}$, and remaining observation variables are denoted by $o_1, \ldots, o_{\tilde{n}}$.

- **Inferred hidden variables** are simply $\cup_{i=1}^{\tilde{n}} \mathcal{P}(o'_i)$, *i.e.* all hidden variables influencing (remaining) observation variables. The set of inferred hidden variables is $\mathbb{S}_h = \{ s_{h,1}, s_{h,2}, \ldots, s_{h,m_h} \}$ and contains possibly visible variables.
- **Non-inferred hidden variables** or **fully hidden variables**, denoted by $\mathbb{S}_f$, consists of hidden state variables which do not influence any observation, *i.e.* all remaining state variables. The fully hidden variables are denoted by $s_{f,1}, s_{f,2}, \ldots, s_{f,m_f}$, and the corresponding set is $\mathbb{S}_f$.

Of course, this classification leads to a partition of the initial set of state variables if potential visible variables are removed from inferred hidden variables: denoting purely inferred hidden variables by $\overline{\mathbb{S}}_h = \mathbb{S}_h \setminus \mathbb{S}_v$, and $\overline{m}_h = \#\overline{\mathbb{S}}_h$ the state variables partition is $\mathbb{S} = \mathbb{S}_v \sqcup \overline{\mathbb{S}}_h \sqcup \mathbb{S}_f$ and $m = m_v + \overline{m}_h + m_f$.

The classification defined here is used to avoid some computations for visible variables: if $s_v \in \mathbb{S}_v$ is visible, and $o_v \in \mathbb{O}$ is the associated observation ($s_v = o_v$ almost surely), computations of the distribution over $\mathcal{P}(o'_v)$, Equation V.6, and of the distribution over o'_v, Equation V.7, are unnecessary: the distribution over s'_v ($= o'_v$) needed is simply given by $\mathbf{p}\left(s'_v \mid \mathcal{P}(s'_v), a\right)$, data of the original problem. The counterpart of Equation V.8 is then

$$\mathbf{p}\left(s'_v \mid \beta, a\right) = \sum_{2^{\mathcal{P}(s'_v)}} \mathbf{p}\left(s'_v \mid \mathcal{P}(s'_v), a\right) \cdot \overline{\beta}(\mathcal{P}(s'_v)), \tag{V.9}$$

where $\overline{\beta}$ is the probability distribution over $2^{\mathcal{P}(s'_v)}$ extracted from the possibilistic belief over the same space, using transformation (V.3).

V.3.4 Belief updating process definition and handling

This section is meant to define marginal belief distributions instead of a global one, in order to benefit from the factorized structure of the initial POMDP. Indeed, possibilistic belief distributions have different definitions according to which class of state variables they concern:

- as visible state variables are directly observed, there is no uncertainty over these variables. Two epistemic states (possibilistic belief distribution) are possible for visible state variable $s'_{v,j}$: $b'_{v,T}(s'_{v,j}) = \mathbb{1}_{\{s'_{v,j} = \top\}}$ and $b'_{v,F}(s'_{v,j}) = \mathbb{1}_{\{s'_{v,j} = \bot\}}$. As a consequence, one boolean variable $\beta'_{v,j} \in \{\top, \bot\}$ per visible state variables is enough to represent this belief distribution in practice: if $s'_{v,j} = \top$, then next belief is $b' = b'_{v,T}$ represented by belief variable assignment $\beta'_{v,j} = \top$, otherwise, next belief is $b' = b'_{v,F}$, and $\beta'_{v,j} = \bot$. A belief variable of a visible state variable is denoted by β_v.
- for each $i \in 1, \ldots, \tilde{n}$, each inferred hidden variable constituting $\mathcal{P}(o'_i)$ is an input of the same possibilistic belief distribution: non-normalized belief is, $\forall i = 1, \ldots, \tilde{n}$

$$\tilde{b}'(\mathcal{P}(o'_i)) = \max_{v \in 2^{\mathcal{Q}(o'_i)}} \min \left\{ \pi\left(o'_i, \mathcal{P}(o'_i) \mid v, a\right), b(v) \right\}. \tag{V.10}$$

where joint possibility distributions over $o'_i \times \mathcal{P}(o'_i)$, needed for belief process definition (possibilistic belief update, Theorem 14), are computed in the following way:

$$\begin{aligned}\pi\left(o'_i, \mathcal{P}(o'_i) \mid \mathcal{Q}(o'_i), a\right) &= \min\left\{\pi\left(o'_i \mid \mathcal{P}(o'_i), a\right), \pi\left(\mathcal{P}(o'_i) \mid \mathcal{Q}(o'_i), a\right)\right\} \\ &= \min\left\{\pi\left(o'_i \mid \mathcal{P}(o'_i), a\right), \min_{s'_j \in \mathcal{P}(o'_i)} \pi\left(s'_j \,\middle|\, \mathcal{P}(s'_j), a\right)\right\}.\end{aligned}$$

A possibilistic normalization finalizes the belief update: for $w \in 2^{\mathcal{P}(o'_i)}$,

$$b'(w') = \begin{cases} 1 \text{ if } w' \in \operatorname{argmax}_{v' \in 2^{\mathcal{P}(o'_i)}} \tilde{b}'(v'); \\ \tilde{b}'(w') \text{ otherwise.} \end{cases} \tag{V.11}$$

In practice, if $l = \#\mathcal{L}$ is the size of the possibility scale, and $p_i = \#\mathcal{P}(o'_i)$, the number of belief states is $l^{2^{p_i}} - (l-1)^{2^{p_i}}$, and then the number of belief variables is $n_{h,i} = \lceil \log_2(l^{2^{p_i}} - (l-1)^{2^{p_i}}) \rceil$. A belief variable of an inferred hidden state variable is denoted by β_h.

- for each $j \in 1, \ldots, m_f$, non-normalized belief defined on fully hidden variable $s_{f,j}$ is

$$\tilde{b}'(s'_{f,j}) = \max_{v \in 2^{\mathcal{P}(s'_{f,j})}} \min\left\{\pi\left(s'_{f,j} \,\middle|\, v, a\right), b(v)\right\}, \tag{V.12}$$

which leads to the actual new belief state b' after normalization (V.11). In practice, as each fully hidden variable is considered independently from the others, following the previous reasoning for vector of inferred hidden s.v., the number of belief variables is $n_f = \lceil \log_2(l^2 - (l-1)^2) \rceil = \lceil \log_2(2l-1) \rceil$. A belief variable of a fully hidden state variable is denoted by β_f.

Finally the actual global epistemic state $b'(\mathbb{S}')$ is upper bounded by

$$b'(\mathbb{S}') = \min\left\{\min_{j=1}^{m_v} b'(s'_{v,j}), \min_{i=1}^{\tilde{n}} b'(\mathcal{P}(o'_i)), \min_{k=1}^{m_f} b'(s'_{f,k})\right\}, \tag{V.13}$$

where $\mathbb{S}$ has to be seen as a vector composed of all state variables.
The latter is considered as the agent belief to make the final MDP factorized.

V.3.5 Selection and use of belief variables

Starting with initial belief states defined for each visible state variable $s_{v,j}$, for each vector of inferred hidden variables $\mathcal{P}(o'_i)$ and for each fully hidden variable $s_{f,j}$, these belief states are updated at each time step according to the transformations described above.

However previous formulae V.10 and distribution over observation variable $o'_i \in \mathbb{O}$, Equation V.8, depend on belief distribution over $\mathcal{Q}(o'_i) \subseteq \mathbb{S}$. They can be computed from the available belief states as follow: $\forall i = 1, \ldots, \tilde{n}$,

$$b(\mathcal{Q}(o'_i)) = \max_{v \in 2^{\mathcal{K}_i}} \min\left\{\min_{s_v \in \mathcal{Q}(o'_i) \cap \mathbb{S}_v} b(s_v), \min_{j \in \mathcal{J}_i} b(\mathcal{P}(o_j)), \min_{s_f \in \mathcal{Q}(o'_i) \cap \mathbb{S}_f} b(s_f)\right\}, \tag{V.14}$$

where

- $\mathcal{J}_i = \{j \in \{1, \ldots, \tilde{n}\} \text{ s.t. } \mathcal{P}(o_j) \cap \mathcal{Q}(o'_i) \neq \emptyset\}$, *i.e.* $\mathcal{J}_i$ is the set of indices j for which $\mathcal{P}(o_j)$ shares (inferred hidden) state variables with $\mathcal{Q}(o'_i)$, and
- $\mathcal{K}_i = \{\cup_{j \in \mathcal{J}_i} \mathcal{P}(o_j)\} \setminus \mathcal{Q}(o'_i) \subseteq \mathbb{S}_h$, *i.e.* $\mathcal{K}_i$ is the set of (inferred hidden) state variables which are not present in $\mathcal{Q}(o'_i)$, but are present in a set $\mathcal{P}(o_j)$ sharing state variables with $\mathcal{Q}(o'_i)$.

As well, belief update for fully hidden state variables, Equation V.12, needs a belief distribution over variables $\mathcal{P}(s'_{f,j})$: $\forall j = 1, \ldots, m_f$,

$$b(\mathcal{P}(s'_{f,j})) = \max_{v \in 2^{\mathcal{N}_j}} \min \left\{ \min_{s_v \in \mathcal{P}(s'_{f,j}) \cap \mathbb{S}_v} b(s_v), \min_{k \in \mathcal{M}_j} b(\mathcal{P}(o_k)), \min_{s_f \in \mathcal{P}(s'_{f,j}) \cap \mathbb{S}_f} b(s_f) \right\},$$

where

- $\mathcal{M}_j = \left\{ k \in \{1, \ldots, m_f\} \text{ s.t. } \mathcal{P}(o_k) \cap \mathcal{P}(s'_{f,j}) \neq \emptyset \right\}$, *i.e.* $\mathcal{M}_j$ is the set of indices k for which $\mathcal{P}(o_k)$ shares (inferred hidden) state variables with $\mathcal{P}(s'_{f,j})$, and
- $\mathcal{N}_j = \left\{ \cup_{k \in \mathcal{M}_j} \mathcal{P}(o_k) \right\} \setminus \mathcal{P}(s'_{f,j}) \subseteq \mathbb{S}_h$, *i.e.* $\mathcal{N}_j$ is the set of (inferred hidden) state variables which are not present in $\mathcal{P}(s'_{f,j})$, but are present in a set $\mathcal{P}(o_k)$ sharing state variables with $\mathcal{P}(s'_{f,j})$.

Finally, a belief distribution over $\mathcal{P}(s'_{v,i})$ needed to define an approximate probability distribution over visible state variables (Equation V.9), can be defined in the same way, marginalizing (max) over unused variables.

V.4 Solving a POMDP with a discrete MDP solver

The previous section leads to a factored MDP, whose the version used in practice is defined here. A concrete POMDP problem and its resulting MDP are then described in order to highlight the power in state space size reduction of the possibilistic structured translation.

V.4.1 Resulting factored MDP:

Section V.3.3 classifies state variables in order to define epistemic states b over sets of state variables (respectively $\forall j = 1, \ldots, m_v$, $\left\{ s_j^v \right\}$, $\forall i = 1, \ldots, \tilde{n}$, $\mathcal{P}(o'_i)$, and $\forall k = 1, \ldots, m_f$, $\left\{ s_j^f \right\}$) and set of variables encoding them (respectively β_v, β_h and β_f) independently to each other. As belief updates are deterministic knowing the observation, a simple trick is used to keep this determinism in the final MDP: a *flipflop* boolean variable is introduced, changing its state at each step, denoted by f. It artificially divides a classical time step of the POMDP into two phases. During the first phase, called *the observation generation phase*, non-identity transition functions (*i.e.* which do not let the variable remain the same) are the probability distributions over observation variables V.8 and visible state variables V.9.

During the second phase, called *the belief update phase*, non-identity transition functions are the deterministic transitions of the belief variables:

- variable β_v is updated knowing value of the corresponding visible variable s_v;
- variables $\beta_h^1, \ldots, \beta_h^{n_{h,i}}$ are updated knowing value of observation variable o_i, and using update V.10, V.11;
- variables $\beta_f^1, \ldots, \beta_f^{n_f}$ using update V.12.

The state space is then defined as:
$\mathcal{S} = f \times s_v^1 \times \ldots \times s_v^{m_v} \times o^1 \times \ldots \times o^{\tilde{n}} \times \beta_v^1 \times \ldots \times \beta_v^{m_v} \times \beta_h^1 \times \ldots \times \beta_h^{\tilde{n}} \times \beta_f^1 \times \ldots \times \beta_f^{m_f}$,
where $\forall i = 1, \ldots, \tilde{n}$, β_h^i represents boolean variables $\beta_h^{1,i}, \ldots, \beta_h^{n_{h,i},i}$, and $\forall k = 1, \ldots, m_f$, β_f^j represents boolean variables $\beta_f^{1,j}, \ldots, \beta_f^{n_f,j}$.

The resulting MDP is illustrated in Figure V.3 where β_t represents all belief variables, and v_t the visible variables: flipflop variable f, observations o_i and visible state variables s_v.

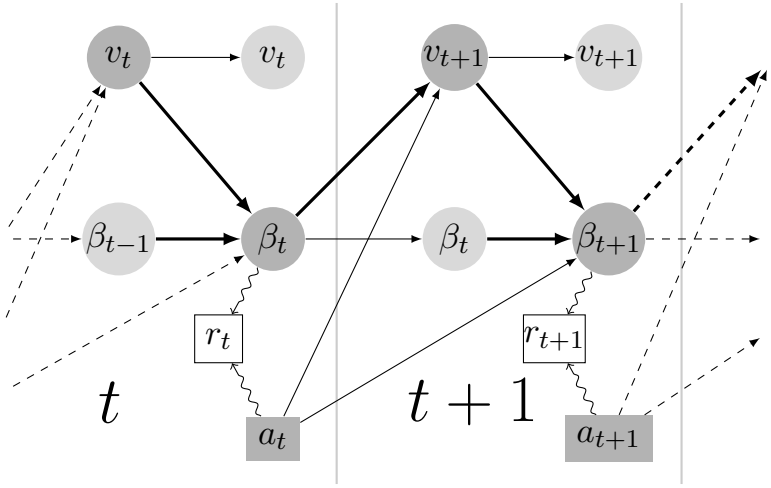


Figure V.3 – *Practical DBN of the resulting MDP: thickest arrows illustrate transitions which are not identity transitions.*

This trick makes the belief update phase deterministic. Each belief variable transition can be then deterministically defined, and independently from each other: as visible state and observation variables are already post-action independent, the resulting MDP is a factored MDP.

V.4.2 Results for a concrete POMDP problem

A problem inspired by the RockSample problem [136] is described in this section to illustrate the factorized possibilistic discretization of the agent belief, from a factored POMDP: a rover is navigating in a place described by a finite number of locations $l_1, \ldots, l_n$, and where stand m rocks. Some of these m rocks have an interest in the scientific mission of the rover, and it has to sample them. However, sampling a rock is an expensive operation. The rover is thus fitted with a long range sensor making him able to estimate if the rock has to be sampled. Finally operating time of the rover is limited, but its battery level is available.

Variables of this problem can now be set, and classified as in Section V.3.3: as the battery level is directly observable by the agent (the rover), the set of visible state variables consists of the boolean variables encoding it: $\mathbb{S}_v = \{ B_1, B_2, \ldots, B_k \}$. The agent knows the different locations of the rocks, however the nature of a rock is estimated. The set of inferred hidden state variables consists of m boolean variables R_i encoding the nature of the i^{th} rock, $\top$ for "scientifically good" and $\bot$ otherwise: $\mathbb{S}_h = \{ R_1, R_2, \ldots, R_m \}$. When the i^{th} rock is observed using the sensor, it returns a noisy observation of the rock in $\{ \top, \bot \}$, modeled by the boolean variable O_i: the set of observation variables is then $\mathbb{O} = \{ O_1, O_2, \ldots, O_m \}$. Finally, no localization equipment is provided: the agent estimates its location from its initial information, and its dynamics. Each location of the rover is formally described by a variable L_j, which equals $\top$ if the rover is at the j^{th} location, and $\bot$ otherwise. The set of fully hidden variables consists thus of these n variables: $\mathbb{S}_f = \{ L_1, L_2, \ldots, L_n \}$.

Initial location is known, described by variable L_1, and leading to a deterministic initial belief state: $\beta_0(\mathcal{S}_h) = 1$ if $L_1 = \top$ and $L_j = \bot \; \forall j \neq 1$, 0 otherwise. However the initial nature of each rock is not known. Instead of a uniform probability distribution over the rocks nature ("rock has to be sampled", or "rock is not interesting"), Possibility Theory allows to represent this initial ignorance with the marginal belief $\beta_0(\mathcal{S}_h) = 1$, for each assignment of the hidden inferred state variables modelling nature of the rocks.

Finally, the factorization trick leads to a reduction of the domain size: with a flat translation of this POMDP, the size of the resulting state space is described with $\lceil \log_2(\#\mathcal{L}^{2^{n+m+k}} - (\#\mathcal{L} -$

$1)^{2^{n+m+k}})\rceil$ boolean variables. Taking advantage of the POMDP structure, the resulting state space is encoded with $1+2\cdots k+m+(m+n).\lceil \log_2(2\#\mathcal{L}-1)\rceil$ Boolean variables: the flipflop variable, the visible variables and associated beliefs variables, the observation variables, and the belief variables associated to the fully hidden and inferred hidden variables.

Moreover, the dynamic of the resulting MDP is factorized: all variables are independent post-action, and lots of them are deterministic, thank to the flipflop variable trick. These structures are beneficial to the MDP solvers, leading to faster computations.

V.5 Conclusion

This chapter described a hydrid translation of a POMDP problem into a finite state space MDP one: Qualitative Possibility Theory is used here to maintain an epistemic state during the process. The MDP problem, result of this translation, is entirely built defining transition and reward functions over these epistemic states. Definitions of these functions use respectively the pignistic transformation, used to recover a probability distribution from an epistemic state, and the Choquet integral with respect to the necessity, making the agent pessimistic about the potential ignorance described by its epistemic state. A practical way to implement this translation is then described: with these computations, a factored POMDP leads to a factored and tractable MDP problem. The essential particularity of this translation is the granular modeling of the agent belief using a qualitative fuzzy knowledge representation Finally this promising approach will be tested on RDDL files of the IPPC competition [127] using a state of the art MDP planner like PROST [79]. Indeed these files describe factored POMDP problems as introduced in Section V.3.

Conclusion

Contributions of this thesis are mainly related to the preliminary works of Régis Sabbadin [121]. The latter proposes a possibilistic counterpart to the POMDPs modeling uncertainty with qualitative possibility distributions. Hence, in our work, qualitative possibilistic models for planning under uncertainty are further developed and studied. This study is meant to adress some issues in probabilistic POMDPs detailed in Introduction: the probabilistic framework is also formally introduced in the first chapter.

The major motivation of this study is computational complexity reduction: while the probabilistic belief space is infinite, the possibilistic one can be finite. The qualitative possibilistic framework thus offers an appropriate belief space discretization. Moreover, as the belief state is a possibility distribution over the system space in this framework, total ignorance can be defined by a possibility distribution equal to 1 on all states. If a given system state is perfectly known to be the actual one, the belief state that assign 1 to this state and to 0 to all other states is an appropriate representation.

Imprecision in the probability distributions are also naturally encoded with the possibilistic formalism, resulting in two criteria for the selection of the strategy: one is optimistic and the other pessimistic. A more practical advantage is that the qualitative possibilistic modeling needs less information about the system than the probabilistic one: the plausibilities of events are "only" classified in the possibilistic scale $\mathcal{L}$ but not quantified. Possibilistic models can be seen as a tradeoff between *non-deterministic* ones, whose uncertainties are not at all quantified yielding a very imprecise model, and probabilistic ones, where uncertainties are fully specified. Indeed, under the non-deterministic formalism, an elementary event is either "possible", or "impossible": no degree is available to differentiate a highly plausible event from an unlikely one.

In a nutshell, this thesis consists of theoretical and practical contributions: on the one hand, theoretical contributions are for instance the proposed updates of the qualitative possibilistic processes – mixed-observability and management of unbounded executions – or independence results on them, with associated proofs. On the other hand, practical contributions are the demonstration of the accuracy of qualitative possibilistic models to simplify computations or for modeling, via experimental results (e.g. IPPC 2014) and modeling examples (e.g. chapter on human-machine interaction). Particular emphasis is being placed on the motivations developed in Introduction besides complexity reduction and problem imprecisions: namely, robotic applications and belief management. For instance, target recognition missions are studied in the second and third chapters (e.g. *Rocksample* problem, or even the mission described in Figure II.3); IPPC 2014 contains also problems comparable to robotic systems (e.g. *Elevator* problem, or *Tamarisk* problem, also used in 2014 RL competition). Moreover, the fourth chapter shows good results in estimating human assessment with qualitative possibility distributions (*i.e.* belief states on the human assessment of the machine state). Finally, the hybrid probabilistic-possibilistic POMDP contribution can be seen as a concluding work, taking into account potential issues when using purely possibilistic models for planning under uncertainty. A more detailed review of the contributions follows.

New features for the π-POMDPs and first strategy executions

The very first contribution to the Qualitative possibilistic POMDPs [121] concerns the qualitative aggregations of the preferences over time: we first show that, much like in the probabilistic framework, preference aggregation can be derived from the properties of the framework – here the qualitative counterpart of linearity for Sugeno integrals for instance – as proved in Annex B.3 using the Theorem I.65 about belief-dependent value functions. The latter is also a contribution as the first formal construction of the π-POMDP model. Different approaches between the pessimistic and the optimistic criterion are also presented: note that the mixed optimistic-pessimistic criterion (see Definition II.1.3) is mainly used along our work since it is equivalent to a π-MDP with an optimistic criterion. As the optimistic criterion is compatible with proposed algorithms for time-unbounded processes, and produces often better strategies for the treated problems, this allows to take advantage of these points. The mixed optimistic-pessimistic criterion is pessimistic according to the belief state: we observe that it produces a good tradeoff with the optimistic global criterion.

We then adapted π-POMDPs to mixed-observability [100], which is part of the theoretical contributions of our work: this contribution, called π-MOMDP, dramatically reduces the size of the belief space and thus allows the first computations of strategies from π-POMDPs. Finally, another theoretical contribution is the qualitative counterpart to the value iteration algorithm, with the associated criterion for time-unbounded executions. As proved in Annex B.6 (and our publication [49]), there exists an optimal strategy, which is stationary *i.e.* which does not depend on the stage of the process t. This strategy can be computed by the proposed dynamic programming scheme: it is also shown that the number of iterations to make the value function converge is less than the size of the state space. The assumption of the existence of a "stay" action is not a constraint in practice as it is only selected in some goal states. Note also that the target recognition missions presented in this thesis have typically unbounded durations.

As already pointed out, the experiments in the second chapter use both previous theoretical contributions. Indeed, the mixed-observability property of the problem, makes computations feasible for our robotic example. The proposed criterion is also really useful: it is convenient to allow the computation of strategies for robotic missions with unbounded durations. For instance, in our example, it allows to define the mission properly: if the robot has not figured out which target is right one, we want the mission to be continued. The first practical contributions of this thesis is thus the computation of a strategy for this robotic mission and its execution. Indeed, we have shown that the qualitative possibilistic approach can outperform probabilistic POMDP ones for a target recognition problem where the agent's observations dynamics is not accurately defined. Note that the only information used to define system observation of the system, is that the more the robot is far from a target, the more the observation is noisy: it already shows that the possibilistic framework may be useful in case of restricted knowledge about the problem.

Finally, the second chapter highlights a very interesting behavior of qualitative possibilistic belief states: under some conditions, the possibilistic belief update, which is defined from the counterpart of Bayes rule [58], increases the knowledge associated to the belief state. Indeed, in our example, the belief state is responsible of the imprecision as it only takes into account more reliable observations, and may also change to the opposite belief if an observation contradicts the current one. On the contrary a probabilistic belief is modified in most cases (there is a finite number of normalized eigenvectors for a given matrix). Some conditions leading to this behavior are then presented, and associated proofs are given (see Annex B.7).

Graphical work on independence and factorization (PPUDD)

The next theoretical contribution is the introduction of the factored π-MOMDPs: indeed, we have considered processes whose state space is represented by n post-action independent variables, *i.e.* processes whose state variables of the same time step are independent (conditional on the past). The algorithm proposed to exactly solve these factorized processes, called PPUDD, is another contribution: it is the symbolic version of the dynamic programming scheme proposed in the previous chapter. Inspired by SPUDD [75], PPUDD means *Possibilistic Planning Using Decision Diagrams*. As SPUDD, it operates on ADDs encoding value, preference and transition functions. Finally, the last theoretical contributions of the third chapter are the proofs of independence results, Theorem 24 and Theorem 25: they relate to a proposed graphical model representing a particular factorization of the process *i.e.* particular independence assumptions on the variables. Thus, we have shown that these independence assumptions lead to a natural factorization of the space of the belief states. The belief state variables of a π-MOMDP satisfying these assumptions are post-action independent, and the associated problem can be more easily solved by PPUDD. The independence of sensors and of corresponding hidden state variables suffices to fulfill these conditions: for instance, the *RockSample* problem, well illustrates these conditions.

The motivation leading to the design of PPUDD, *i.e.* the guess that the use of operators min and max leads to smaller ADDs, and thus to faster computations, has been verified in practice: our experiments and the results of the International Probabilistic Planning Competition (IPPC 2014[1]) show that this possibilistic approach can involve less computation time and produce better policies than its probabilistic counterparts when computation time is limited, or for high dimensional problems. For instance, PPUDD performances have been compared to the probabilistic MOMDP solver APPL [84, 100] and the symbolic HSVI solver [133], in terms of computation time, and using the average of the total reward at execution. PPUDD computes a strategy maximizing exactly the possibilistic criterion while APPL and symb. HSVI compute strategies which approximately maximize the expected sum of rewards. The experimental results on the Rocksample problem show that using an exact algorithm (PPUDD) for an approximate model (π-MOMDPs) can run significantly faster than reasoning about exact models, while providing better policies than approximate algorithms (APPL) for exact models. Most of the cited contributions of this chapter have been published in [50], and an implementation of PPUDD to reproduce experiments can be found at the repository `www.github.com/drougui/ppudd`.

Finally, in order to focus on the behavior of the possibilistic qualitative approach in a wide panel of probabilistic problems, the next practical contribution is the participation in the fully observable track of IPPC 2014 with adapted versions of PPUDD. The implementation of our solver for this competition was performed with the *CU Decision Diagram Package* for the ADDs computations. Comparing only solvers using ADDs, PPUDD and a probabilistic solver called symbolic LRTDP [45] (as a variant of [14]), our solver produces better strategies than the probabilistic approach.

Discussion about the current results

Experiments concerning the Navigation problem is a good illustration of the optimistic and pessimistic criteria in the qualitative possibilistic framework: a robot has to reach a goal as soon as possible, avoiding unsafe locations (for instance, locations where there is a risk that it falls down and break). The strategy maximizing the optimistic criterion makes the robot unfrequently reach the goal, but more quickly than the one from the pessimistic criterion

[1] `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`

which makes it however often reach the goal. Proposed approaches involve then the choice of a criterion. This feature of the possibilistic approach explains also the difficulties experienced with the "Traffic" domain of the competition: the optimistic criterion can be too optimistic for the given problem, although strategies from it are generally more efficient than those from the pessimistic criterion (that is why the optimistic criterion has been used for the competition).

Presented models assume also that preference and uncertainty degrees share the same scale which makes it hard to set in practice. The fourth chapter about human-machine interaction is meant to show that, although qualitative possibilistic approach may raise questions concerning modeling, this approach can be very appropriate in some practical situations, as in our case, to produce diagnosis.

We got really good results with PPUDD in previous experiments among symbolic algorithms (SPUDD, symbolic LRTDP): for instance, SPUDD provides good strategies, but cannot solve big instances of the tested problem. However, state space search algorithms (PROST [79] and GOURMAND [83]) won IPPC 2014, and are yet far more efficient than ADD-based methods: MDP instances with a complex structure are encoded with ADDs big enough to disqualify these approaches. These observations and modeling considerations are the motivations of the last chapter, presenting the hybrid POMDP.

Estimating in a qualitative world (HMI)

Obviously, although qualitative possibilistic models can lead to appropriate approximate strategies for high dimensional problems, they cannot lead to the best results for simple problems in purely frequentist worlds. On the contrary, the fourth chapter focuses on the use of this framework in a purely qualitative world: indeed it proposes the study of human-machine interactions which only involve the deterministic behavior of the machine and a qualitative expert knowledge about the human behavior.

This joint work with Sergio Pizziol shows that the use of qualitative possibilistic processes is an appropriate choice when probabilities cannot be defined in practice. Indeed, this framework produces useful error detections and diagnosis in real experiments with pilots in a simulator.

Finally, this work also provides theoretical contributions. The first one is summed up in Theorem 27 and Theorem 28. These theorems lead to the definition of the possibilistic estimation of the human assessment within the framework of the qualitative possibilistic Hidden Markov Processes (π-HMPs): this estimation is used to detect human attentional errors. Another contribution is the diagnosis computation, using the *leximin* operator. Finally, a toy example is also provided to explain the use, the meaning and the behavior of these possibilistic tools in practice.

A possibilistic discretization of the belief space (hybrid POMDP

The work on the hybrid POMDP is motivated by the proper discretization provided by the use of a qualitative possibilistic belief state. The possibilistic nature of these belief states allows to define the reward attached to them in a pessimistic way using the Choquet integral. This model is proposed as a way to compute strategies more easily since the hybrid POMDP is finally solved as a classical MDP. This approach may also lead to cautious behaviors as the reward definition on belief states is pessimistic with respect to the lack of knowledge.

The computation of the transition function of the resulting MDP are given: it uses a *possibility-probability transformation* called *pignistic transformation* to recover a probability distribution from a possibilistic belief state, in a neutral way.

We described also how a factored POMDP leads to a factored and more tractable MDP: indeed, the final contribution shows how to take advantage from a large class of common structures (at least among IPPC domains) to make the resulting MDP as simple as possible.

Hence, efficient MDP solvers such as PROST or GOURMAND, are able to solve the factored MDP representing the hybrid POMDP. This model has been published in [48].

Note finally that all the partially observable processes presented in this work (purely qualitative and hybrid) can be translated into MDP or π-MDP and are thus computable in finite time, at most exponential in the description of the problem.

Perspectives

The first perspective comes from an observation: the memory limitation plausibly reached with the *Triangle tireworld* problem of IPPC 2014 may be bypassed by a stronger discretization: versions of PPUDD of IPPC 2014 used a precision of 10^{-3} on the initial probabilistic model, leading to a big scale $\mathcal{L}$ – defined in practice as all the transformed probability values and normalized reward values present in the given MDP. In any case, it would be instructive to have an idea of the impact of the precision of the discretization on the performances of PPUDD applied to probabilistic problems. More generally, more efforts on the translation from a probabilistic MDP into its possibilistic approximation may significantly improve the approach we proposed for IPPC 2014.

We focused on the symbolic resolution of the π-MDPs (PPUDD). However alternative resolution methods could be studied: for instance, an heuristic-based strategy computation, which is efficient for probabilistic problems [140], could be adapted to the possibilistic context. As regards reinforcement learning in the qualitative possibilistic context, we can cite the following work [124].

The second chapter does not propose any algorithm to compute strategies for missions with unbounded horizon according to the pessimistic criterion. The algorithm is straightforward, but the optimality of the resulting strategy for this criterion seems hard to prove. However, an optimal strategy in this sense, would be useful in order to manage long-term unsafe problems.

The independence results given in the third chapter are also valid for probabilistic MOMDPs: it would be interesting to determine if those results can have a positive impact on the optimal strategy computation.

The work [17] proposes an other framework for planning under uncertainty, also qualified as qualitative. However, this work uses quantitative operations as sum and product, and may be seen as a discretization of the probabilistic framework [149]. The authors remark however that Qualitative Possibility Theory leads to criteria which have not enough information for discriminating among optimal decisions.

Advances in Possibility Theory may lead to more refined possibilistic MDPs to improve modeling when needed. We can mention the *leximin* operator [54], used in the fourth chapter: it avoids the drowning effect of the operators min and max. It may be used for preference aggregation or even to compute a refined possibility degree of a trajectory. It may be a useful improvement, but note that it eventually makes the problem more complex. Another update which has still to be performed is the integration of more discriminative criteria [146, 71] to the π-MDP framework.

In the work on HMI, the human operator is supposed to be certain about the state of the machine even if her/his only guess may be wrong. A future work could define the human assessment in a more complex way: for instance as a possibility distribution: a possibilistic belief state of the human operator.

Finally, the promising approach presented under the name hybrid POMDP will be tested on the POMDPs of the IPPC competition [127] in a future work: indeed, the proposed problems are factored POMDPs as introduced in Section V.3.

Annex

A Proofs of Chapter I

A.1 Preliminaries

Firstly recall the more general definition of the *conditional expectation* with respect to a random variable: a random variable is a measurable function defined on the set Ω equipped with the σ-algebra $\mathcal{F}$ and the probability measure $\mathbb{P}$.

Definition A.1 (*Expectation of X Conditional on Y:* $\mathbb{E}[X \mid Y]$)

Let X et Y be two random variables defined on $(\Omega, \mathcal{F}, \mathbb{P})$: values of X are in $\mathbb{R}$ equipped with the Borel σ-algebra $\mathcal{B}(\mathbb{R})$ and X is integrable. Values of Y are in a set $\mathcal{Y}$ equipped with a σ-algebra $\mathcal{V}$. Let us denote by $\sigma(Y)$ the σ-algebra generated by Y i.e. $\sigma(Y) = \{Y^{-1}(V) \mid V \in \mathcal{V}\} \subset \mathcal{F}$.

The ***expectation of X conditional on Y****, denoted by $\mathbb{E}[X \mid Y]$, is* ***the unique random variable in $\mathbb{L}^1(\Omega, \mathcal{F}, \mathbb{P})$ which is***

- *$\sigma(Y)$-measurable,*
- *and such that $\forall A \in \sigma(Y)$, $\int_A \mathbb{E}[X|Y](\omega)\, d\mathbb{P}(\omega) = \int_A X(\omega) d\mathbb{P}(\omega)$.*

As the classical expectation, the conditional expectation is linear: if X_1 and X_2 are two random variables, $\forall c \in \mathbb{R}$, $\mathbb{E}[c \cdot X_1 + X_2 \mid Y] = c \cdot \mathbb{E}[X_1 \mid Y] + \mathbb{E}[X_2 \mid Y]$.

First, note that if X is $\sigma(Y)$-measurable, $X = \mathbb{E}[X \mid Y]$ $\mathbb{P}$-almost surely. Indeed, the function X meets both conditions to be $\mathbb{E}[X \mid Y]$. Note also that the second point of Definition A.1 implies that

$$\mathbb{E}\Big[\mathbb{E}[X \mid Y]\Big] = \int_\Omega \mathbb{E}[X \mid Y](\omega) d\mathbb{P}(\omega) = \int_\Omega X(\omega) d\mathbb{P}(\omega) = \mathbb{E}[X].$$

This second point can be replaced by an other characterization, given by the following property:

Property A.1

The random variable $\mathbb{E}[X \mid Y]$ can be defined as the unique function $\sigma(Y)$-measurable such that $\forall Z : \mathcal{Y} \to \mathbb{R}$ $\sigma(Y)$-measurable,

$$\mathbb{E}\Big[Z \cdot \mathbb{E}[X \mid Y]\Big] = \mathbb{E}[Z \cdot X].$$

Proof: Indeed if $Z = \mathbb{1}_A$ with $A \in \sigma(Y)$, we fall back into the second point of Definition A.1. Thanks to the linearity of the expectation, it remains true if Z is a linear combination of characteristic functions of elements of the σ-algebra. Finally, consider a non decreasing sequence $(Z_n)_{n\in\mathbb{N}}$, with $\forall n \in \mathbb{N}$, Z_n a combination of characteristic functions, and whose limit is a measurable function Z. Equality holds for each Z_n, and thanks to Beppo-Levi Theorem, the result is true for the measurable function Z. ■

This result makes the following property easier to show:

Property A.2

Let X be a real and integrable random variable, and Y_1, Y_2 two random variables:

$$\mathbb{E}\Big[\mathbb{E}[X \mid Y_1, Y_2]\Big|Y_2\Big] = \mathbb{E}[X \mid Y_2] \ \mathbb{P}\text{-almost surely.}$$

As well,

$$\mathbb{E}\Big[\mathbb{E}\left[X \mid Y_2\right]\Big|Y_1, Y_2\Big] = \mathbb{E}\left[X \mid Y_2\right] \ \mathbb{P}\text{-almost surely.}$$

Proof: The second equality is obvious because if X is $\sigma(Y)$-measurable, $\mathbb{E}[X \mid Y] = X$ $\mathbb{P}$-almost surely. Indeed, X meets the first condition to be $\mathbb{E}[X \mid Y]$ in Definition A.1, and of course, the second condition is met. Here, the random variable $\mathbb{E}[X \mid Y_2]$ is $\sigma(Y_2)$-measurable by definition, and thus it is $\sigma(Y_1, Y_2)$-measurable: thus the second equality holds.

For the first equality, since both conditional expectations are $\sigma(Y_2)$-measurable by definition, it is sufficient to show that $\forall Z$ $\sigma(Y_2)$-measurable, $\mathbb{E}\Big[Z \cdot \mathbb{E}\Big[\mathbb{E}\left[X \mid Y_1, Y_2\right]\Big|Y_2\Big]\Big] = \mathbb{E}[Z \cdot X]$, as $\mathbb{E}[X \mid Y_2]$ is the only random variable in $\mathbb{L}^1(\Omega, \mathcal{F}, \mathbb{P})$ such that for each $\sigma(Y_2)$-measurable random variable Z, $\mathbb{E}[Z \cdot \mathbb{E}[X \mid Y_2]] = \mathbb{E}[Z \cdot X]$. Let Z be $\sigma(Y_2)$-mesurable: as Z is $\sigma(Y_2)$-measurable, it is *a fortiori* $\sigma(Y_1, Y_2)$-measurable: $\sigma(Y_2) \subseteq \sigma(Y_1, Y_2)$. Thus

$$\mathbb{E}\Big[Z \cdot \mathbb{E}\Big[\mathbb{E}\left[X \mid Y_1, Y_2\right]\Big|Y_2\Big]\Big] = \mathbb{E}\Big[Z \cdot \mathbb{E}\left[X \mid Y_1, Y_2\right]\Big] = \mathbb{E}[Z \cdot X].$$

■

The conditional expectation has been defined as a random variable $\mathbb{E}[X \mid Y] : \Omega \to \mathbb{R}$. However, the following property implies that the $\sigma(Y)$-measurability condition of Definition A.1 may be replaced by "$\exists \varphi : (\mathcal{Y}, \mathcal{V}) \to \Big(\mathbb{R}, \mathcal{B}(\mathbb{R})\Big)$ measurable such that $\mathbb{E}[X \mid Y] = \varphi(Y)$". The function φ is called *Expectation of X Conditional on the Values of Y* and may be denoted by $\varphi(y) = \mathbb{E}[X \mid Y = y]$, $\forall y \in \mathcal{Y}$.

Property A.3

The function $Z : \Omega \to \mathbb{R}$ is $\sigma(Y)$-measurable, where $Y : (\Omega, \mathcal{F}) \to (\mathcal{Y}, \mathcal{V})$
$\Leftrightarrow \exists \varphi : (\mathcal{Y}, \mathcal{V}) \to \Big(\mathbb{R}, \mathcal{B}(\mathbb{R})\Big)$ measurable such that $Z = \varphi(Y)$.

Proof: The set $\{\varphi(Y) \mid \varphi : (\mathcal{Y}, \mathcal{V}) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ measurable $\}$ is denoted by Φ. If $Z \in \Phi$, then $Z = \varphi(Y)$ with φ measurable, and as Y is $\sigma(Y)$-measurable, Z is $\sigma(Y)$-measurable: in short, if $Z \in \Phi$, then Z is $\sigma(Y)$-measurable.

Now let us show that any $\sigma(Y)$-mesurable function can be written $\varphi(Y)$ with φ measurable. By definition $\forall A \in \sigma(Y)$, $\exists B \in \mathcal{V}$ such that $A = \{Y \in B\} = Y^{-1}(B)$. Thus $\mathbb{1}_A = \mathbb{1}_{\{Y \in B\}} = \mathbb{1}_B(Y)$ is in Φ as a function of Y. Now, as linear combinations of such characteristic functions are in Φ, and as non-decreasing limits of sequences of functions in Φ are in Φ, it can be concluded that Φ contains all $\sigma(Y)$-measurable functions. ■

Definition A.2 (*Expectation of X Conditional on the Values of Y*)

Let $X \in \mathbb{R}$ and $Y \in \mathcal{Y}$ two random variables: as $\mathbb{E}[X \mid Y]$ can be written $\varphi(Y)$ with $\varphi : \mathcal{Y} \to \mathbb{R}$ measurable,

$$\mathbb{E}\Big[\ X \ \Big|\ Y = y\ \Big] = \varphi(y)$$

is called the expectation of X conditional on the event $\{Y = y\}$. For each $y \in \mathcal{Y}$ such that $\mathbb{P}(Y = y) > 0$, it is the expectation of X if we know that $Y = y$.

If $\mathcal{Y}$ is countable, and $\mathbb{P}(Y = y) > 0$,

$$\mathbb{E}[X \mid Y = y] = \int_{\{Y = y\}} \frac{\mathbb{E}[X \mid Y](\omega)}{\mathbb{P}(Y = y)} d\mathbb{P}(\omega).$$

Consider that the values of variables X and Y are in $\mathcal{S}$, and let us introduce $f : \mathcal{S} \to \mathbb{R}$ measurable: $f(X)$ is a random variable whose values are in $\mathbb{R}$. The expectation of $f(X)$ conditional on the values of $Y \in \mathcal{S}$ is denoted by

$\mathbb{E}[f(X) \mid Y = y]$, $\forall y \in \mathcal{S}$. If $\mathcal{S}$ is a countable set equipped with the σ-algebra $\mathcal{P}(\mathcal{S})$ (the set of sets included in $\mathcal{S}$), $\varphi(y) = \mathbb{E}[f(X) \mid Y = y]$ can be computed explicitly.

Property A.4

Let X and Y two random variables whose values are in a countable set $\mathcal{S}$, and $f : \mathcal{S} \to \mathbb{R}$ a measurable function: the expectation of $f(X)$ conditional on the values of Y is

$$\forall y \in \mathcal{S} \text{ such that } \mathbb{P}(Y = y) > 0, \quad \mathbb{E}\Big[f(X)\Big|Y = y\Big] = \sum_{x \in \mathcal{S}} f(x) \cdot \mathbb{P}(X = x \mid Y = y).$$

where $\mathbb{P}(X = x \mid Y = y) = \frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(Y=y)}$.

Proof: For clarity, $\mathbb{E}[f(X) \mid Y = y]$ is denoted by $\varphi(y)$ in this proof. By only considering singletons $\{Y = y\} \subseteq \Omega$ in the second condition of Definition A.1, it leads to $\forall y \in \mathcal{S}$,

$$\int_{\{Y=y\}} \varphi(Y) d\mathbb{P} = \int_{\{Y=y\}} f(X) d\mathbb{P} \quad \Leftrightarrow \quad \varphi(y) \cdot \mathbb{P}(Y = y) = \int_{\{Y=y\}} f(X) d\mathbb{P}$$

and if $\mathbb{P}(Y = y) > 0$,

$$\varphi(y) = \int_\Omega f(X) \cdot \frac{\mathbb{1}_{\{Y=y\}} d\mathbb{P}}{\mathbb{P}(Y = y)} = \int_\Omega \sum_{x \in \mathcal{S}} f(x) \cdot \mathbb{1}_{\{X=x\}} \cdot \frac{\mathbb{1}_{\{Y=y\}} d\mathbb{P}}{\mathbb{P}(Y = y)} = \sum_{x \in \mathcal{S}} f(x) \cdot \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)}$$

thanks to the Fubini theorem. Note that if $f(X) = \mathbb{1}_{\{X=x\}}$, we get $\mathbb{E}\left[\mathbb{1}_{\{X=x\}} \,\middle|\, Y = y\right] = \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)}$, which is the discrete conditional probability $\mathbb{P}(X = x \mid Y = y)$. ■

A.2 Proof of Property I.1.1

Proof: Let $t \in \mathbb{N}$, $(s_0, s_1, \ldots, s_t) \in \mathcal{S}^t$ and $s' \in \mathcal{S}$: on the one hand,

$$\int_{\{S_0=s_0,\ldots,S_t=s_t\}} \mathbb{1}_{\{S_{t+1}=s'\}} d\mathbb{P} = \mathbb{P}(S_0 = s_0, \ldots, S_t = s_t, S_{t+1} = s'),$$

and on the other hand, if $\mathbb{E}\left[\mathbb{1}_{\{S_{t+1}=s'\}} \,\middle|\, S_t\right]$ is denoted by $\mathbb{P}(S_{t+1} = s' \mid S_t)$,

$$\int_{\{S_0=s_0,\ldots,S_t=s_t\}} \mathbb{P}(S_{t+1} = s' \mid S_t) \, d\mathbb{P} = \mathbb{P}(S_{t+1} = s' \mid S_t = s_t) \cdot \mathbb{P}(S_0 = s_0, S_1 = s_1, \ldots, S_t = s_t).$$

Both integrals are equal as $(S_t)_{t \in \mathbb{N}}$ is a Markov Chain. Since events of $\sigma(S_0, \ldots, S_t)$ are unions of events which can be written $\{S_0 = s_0, S_1 = s_1, \ldots, S_t = s_t\}$, for all $B \in \sigma(S_0, \ldots, S_t)$

$$\int_B \mathbb{1}_{\{S_{t+1}=s'\}} d\mathbb{P} = \int_B \mathbb{P}(S_{t+1} = s' \mid S_t) \, d\mathbb{P}$$

and then, using Definition A.1 and as $\mathbb{P}(S_{t+1} = s' \mid S_t)$ is $\sigma(S_0, \ldots, S_t)$-measurable

$$\mathbb{P}(S_{t+1} = s' \mid S_t) = \mathbb{E}\left[\mathbb{1}_{S_{t+1}=s'} \,\middle|\, S_0, \ldots, S_t\right] \ \mathbb{P}\text{-almost surely.}$$

Now,

$$\begin{aligned}
\mathbb{E}[f(S_{t+1}) \mid S_0, \ldots, S_t] &= \mathbb{E}\left[\sum_{s' \in \mathcal{S}} f(s') \cdot \mathbb{1}_{\{S_{t+1}=s'\}} \,\middle|\, S_0, \ldots, S_t\right] \\
&= \sum_{s' \in \mathcal{S}} f(s') \cdot \mathbb{E}\left[\mathbb{1}_{\{S_{t+1}=s'\}} \,\middle|\, S_0, \ldots, S_t\right] \\
&= \sum_{s' \in \mathcal{S}} f(s') \cdot \mathbb{P}(S_{t+1} = s' \mid S_t) \quad \mathbb{P}\text{-almost surely.}
\end{aligned}$$

The random variable $\sum_{s' \in \mathcal{S}} f(s') \cdot \mathbb{P}(S_{t+1} = s' \mid S_t)$ is $\sigma(S_t)$-measurable, thus $\mathbb{E}[f(S_{t+1}) \mid S_0, \ldots, S_t]$ is $\sigma(S_t)$-measurable too, and then,

$$\mathbb{E}[f(S_{t+1}) \mid S_0, \ldots, S_t] = \mathbb{E}\left[\mathbb{E}[f(S_{t+1}) \mid S_0, \ldots, S_t] \,\middle|\, S_t\right] = \mathbb{E}[f(S_{t+1}) \mid S_t],$$

because of Property A.2. Finally, the equalities of Property I.1.1 are achieved by integrating both parts of the equations over $\{ S_0 = s_0, \ldots, S_t = s_t \}$ and then by dividing them by $\mathbb{P}(S_0 = s_0, \ldots, S_t = s_t)$. ■

A.3 Proof of Theorem 1

Proof: First of all, consider that the process is at the stage $\tilde{t} \in \mathbb{N}$. All states from the beginning of the process are given as input to the agent: it has to choose the best action knowing these $\tilde{t}+1$ first system states $\{ s_0, s_1, \ldots, s_{\tilde{t}} \} \in \mathcal{S}^{\tilde{t}+1}$. Regardless the previously gathered rewards, its goal is to maximize the expectation of the sum of the next rewards. We show by induction on $\tilde{t}$ from $H-1$ to 0, that the highest expected total reward can be reached with a strategy $(d_t)_{t=0}^{H-1}$ *i.e.* a sequence of decision rules $d_t : \mathcal{S} \to \mathcal{A}$. A sequence of functions from all the states that the system has gone through, *i.e.* a sequence $(d_t)_{t=0}^{H-1}$ of functions $d_t : \mathcal{S}^t \to \mathcal{A}$, is not necessary.

Let $\tilde{t} = H-1$: the agent has to find the action a maximizing

$$\mathbb{E}\Big[\, r_{H-1}(S_{H-1}, a) + R(S_H) \;\Big|\; S_0 = s_0, \ldots, S_{H-1} = s_{H-1} \, \Big]$$

which is equal to $\mathbb{E}\Big[\, r_{H-1}(S_{H-1}, a) + \mathbb{E}\,[\, R(S_H) \,|\; S_{H-1} \,] \;\Big|\; S_0 = s_0, \ldots, S_{H-1} = s_{H-1} \, \Big]$

$$\begin{aligned} &= \mathbb{E}\,[\, f_a(S_{H-1}) \,|\; S_0 = s_0, \ldots, S_{H-1} = s_{H-1} \,] \text{ with } f_a : \mathcal{S} \to \mathbb{R} \text{ measurable,} \\ &= \mathbb{E}\,[\, f_a(S_{H-1}) \,|\; S_{H-1} = s_{H-1} \,] \end{aligned}$$

because of Property (I.1.1). Then, the value to be maximized depends on the state $s_{H-1} \in \mathcal{S}$ only: a decision rule $d_{H-1}^* : \mathcal{S} \to \mathcal{A}$ such that $\forall s \in \mathcal{S}$

$$d_{H-1}^*(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}\,[\, f_a(S_{H-1}) \,|\; S_{H-1} = s \,]$$

is sufficient. Now, assume that this result is true for the time step $\tilde{t}+1 \leqslant H-1$, and consider that a strategy $(d_t)_{t=\tilde{t}+1}^{H-1}$ has been computed. The agent has to find the action a maximizing

$$\mathbb{E}\left[r_{\tilde{t}}(S_{\tilde{t}}, a) + \sum_{t=\tilde{t}+1}^{H-1} r_t\Big(S_t, d_t(S_t)\Big) + R(S_H) \;\middle|\; S_0 = s_0, \ldots, S_{\tilde{t}} = s_{\tilde{t}} \right]$$

which is equal to $\mathbb{E}\left[r_{\tilde{t}}(S_{\tilde{t}}, a) + \mathbb{E}\left[\sum_{t=\tilde{t}+1}^{H-1} r_t\Big(S_t, d_t(S_t)\Big) + R(S_H) \;\middle|\; S_{\tilde{t}} \right] \;\middle|\; S_0 = s_0, \ldots, S_{\tilde{t}} = s_{\tilde{t}} \right]$

$$\begin{aligned} &= \mathbb{E}\,[\, f_a(S_{\tilde{t}}) \,|\; S_0 = s_0, \ldots, S_{\tilde{t}} = s_{\tilde{t}} \,] \text{ with } f_a : \mathcal{S} \to \mathbb{R} \text{ measurable,} \\ &= \mathbb{E}\,[\, f_a(S_{\tilde{t}}) \,|\; S_{\tilde{t}} = s_{\tilde{t}} \,] \end{aligned}$$

and then, the same conclusion holds: it is sufficient to compute a decision rule $d_{\tilde{t}}^* : \mathcal{S} \to \mathcal{A}$ such that $\forall s \in \mathcal{S}$

$$d_{\tilde{t}}^*(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}\,[\, f_a(S_{\tilde{t}}) \,|\; S_{\tilde{t}} = s \,].$$

We just proved by induction that it suffices to look for strategies of $\mathcal{D}_H$ as defined above, *i.e.* to look for a sequence of decision rules $(d_t)_{t=0}^{H-1}$, maximizing the criterion I.2.

In the following, we set up the Dynamic Programming equations used to compute the optimal value function and the optimal strategy. The size of the horizon i is the index used for the incremental computation of the optimal value function V^*. It is also the opposite modulo H of the stage of the process t, index used for the strategy. The initialization $V_0^*(s) = R(s)$ is obvious: when the horizon is zero, no action has to be selected by the agent, and it receives only the terminal reward. Next, let $s \in \mathcal{S}$, $i \in \{1, \ldots H\}$, and set $t_0 = H - i$:

$$V_i^*(s) \quad = \quad \sup_{(d_t)_{t=t_0}^{H-1} \in \mathcal{D}_{H-t_0}} \mathbb{E}\left[\sum_{t=t_0}^{H-1} r_t\Big(S_t, d_t(S_t)\Big) + R(S_H) \;\middle|\; S_{t_0} = s \right].$$

$$\begin{aligned}
\text{Yet } \mathbb{E}\left[\sum_{t=t_0}^{H-1} r_t\Big(S_i, d_i(S_i)\Big) + R(S_H) \;\middle|\; S_{t_0}\right] & \\
&= \mathbb{E}\left[r_{t_0}\Big(S_{t_0}, d_{t_0}(S_{t_0})\Big) + \mathbb{E}\left[\sum_{t=t_0+1}^{H-1} r_t\Big(S_i, d_i(S_i)\Big) + R(S_H) \;\middle|\; S_{t_0}, S_{t_0+1}\right] \;\middle|\; S_{t_0}\right] \\
&= \mathbb{E}\Big[r_{t_0}\Big(S_{t_0}, d_{t_0}(S_{t_0})\Big) + \mathbb{E}\left[V_{i-1}\Big(S_{t_0+1}, (d_t)_{t=t_0+1}^{H-1}\Big) \;\middle|\; S_{t_0}, S_{t_0+1}\right] \Big| S_{t_0}\Big] \\
&= \mathbb{E}\left[r_{t_0}\Big(S_{t_0}, d_{t_0}(S_{t_0})\Big) + \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}\Big(s' \Big| S_{t_0}, d_{t_0}(S_{t_0})\Big) \cdot V_{i-1}\Big(s', (d_t)_{t=t_0+1}^{H-1}\Big) \;\middle|\; S_{t_0}\right]
\end{aligned}$$

using properties (A.2) and (I.1.1). By integrating both part of this equality over $\{S_{t_0} = s\}$ and divising them by $\mathbb{P}(S_{t_0} = s) > 0$, it becomes

$$\begin{aligned}
V_i^*(s) &= \sup_{(d)_{t=t_0}^{H-1}\in\mathcal{D}_{H-t_0}} \left\{ r_t\Big(s, d_{t_0}(s)\Big) + \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}\Big(s' \Big| s, d_{t_0}(s)\Big) \cdot V_{i-1}\Big(s', (d_t)_{t=t_0+1}^{H-1}\Big) \right\} \\
&= \sup_{\left(a,(d')_{t=t_0+1}^{H-1}\right)\in\mathcal{D}_{H-t_0}} \left\{ r_t(s,a) + \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}\Big(s', (d'_t)\Big) \right\} \\
&= \max_{a\in\mathcal{A}} \left\{ r_t(s,a) + \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot \sup_{(d'_t)_{t=t_0+1}^{H-1}\in\mathcal{D}_{H-t_0-1}} V_{i-1}\Big(s', (d'_t)\Big) \right\} \qquad (15) \\
&= \max_{a\in\mathcal{A}} \left\{ r_t(s,a) + \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}^*(s') \right\}
\end{aligned}$$

where 15 is justified by:

- as $\displaystyle\sum_{s'\in\mathcal{S}_{a,s,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}\Big(s', (d_t)\Big) \leqslant \sum_{s'\in\mathcal{S}_{a,s,t}} \mathbf{p}_{t_0}(s' \mid s,a) \sup_{(d'_t)\in\mathcal{D}_{H-t_0-1}} V_{i-1}\Big(s', (d'_t)\Big)$ for each strategy $(d_t)_{t=t_0+1}^{H-1} \in \mathcal{D}_{H-t_0-1}$,

  $$\sup_{(d'_t)\in\mathcal{D}_{H-t_0-1}} \sum_{s'\in\mathcal{S}_{a,s,t}} \mathbf{p}_{t_0}(s' \mid s,a)\, V_{i-1}\Big(s', (d'_t)\Big) \leqslant \sum_{s'\in\mathcal{S}_{a,s,t}} \mathbf{p}_{t_0}(s' \mid s,a) \sup_{(d'_t)\in\mathcal{D}_{H-t_0-1}} V_{i-1}^*(s').$$

- let $(\varepsilon_n)_{n\in\mathbb{N}}$ be a sequence of positive real numbers, such that $\varepsilon_n \to 0$ when $n \to \infty$. For each $n \in \mathbb{N}$, let $(d_t^{\varepsilon_n})_{t=t_0+1}^{H-1} \in \mathcal{D}_{H-t_0-1}$ be a strategy such that

  $$V_{i-1}^*(s') - \varepsilon \leqslant V_{i-1}\Big(s', (d^{\varepsilon_n})\Big) \leqslant V_{i-1}^*(s'),\ \forall s' \in \mathcal{S}.$$

  Computing the mean with respect to $\mathbf{p}_{t_0}(.\mid s,a)$ whose support is finite, the inequality on the left becomes

  $$\sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}^*(s') - \varepsilon_n \leqslant \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}\Big(s', (d^{\varepsilon_n})\Big)$$

  where the right part is obviously lower than $\displaystyle\sup_{(d)\in\mathcal{D}_{H-t_0-1}} \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}\Big(s', (d)\Big)$.
  Now, as this couple of inequalities are true for each $n \in \mathbb{N}$, making $n \to \infty$, the result is

  $$\sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}^*(s') \leqslant \sup_{(d)\in\mathcal{D}_{H-t_0-1}} \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}\Big(s', (d)\Big).$$

Therefore, if at each iteration $i \in \{1, \ldots, H\}$, the decision rule d_{H-i}^* is defined as $\forall s \in \mathcal{S}$, $d_{H-i}^*(s) \in \operatorname{argmax}_{a\in\mathcal{A}} \Big\{ r_t(s,a) + \sum_{s'\in\mathcal{S}_{s,a,t}} \mathbf{p}_{t_0}(s' \mid s,a) \cdot V_{i-1}^*(s') \Big\}$, $V_i^*\Big(s, (d_t^*)_{t=H-i}^{H-1}\Big) = V_i^*(s)$: with $i = H$, it shows that $(d_t^*)_{t=0}^{H-1} \in \mathcal{D}_H$ is optimal. ■

A.4 Proof of the Bellman Equation (I.5)

Proof: The following calculus lines lead to the Bellman Equation. Let $(d)_{t\in\mathbb{N}} \in \mathcal{D}_\infty$.

$$\begin{aligned}
V^d(s) &:= \mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t\cdot r\Big(S_t,d_t(S_t)\Big)\;\middle|\; S_0=s\right]\\
&= \mathbb{E}\left[r\Big(S_0,d_0(S_0)\Big)+\sum_{t=1}^{+\infty}\gamma^t\cdot r\Big(S_t,d_t(S_t)\Big)\;\middle|\; S_0=s\right]\\
&= r\Big(s,d_0(s)\Big)+\mathbb{E}\left[\mathbb{E}\Big[\sum_{t=1}^{+\infty}\gamma^t\cdot r\Big(S_t,d_t(S_t)\Big)\Big|S_1\Big]\;\middle|\; S_0=s\right]\\
&= r\Big(s,d_0(s)\Big)+\gamma\cdot\sum_{s'\in\mathcal{S}_{s,d_0(s)}}\mathbf{p}\Big(s'\Big|s,d_0(s)\Big)\cdot\mathbb{E}\left[\sum_{t=1}^{+\infty}\gamma^{t-1}\cdot r\Big(S_t,d_t(S_t)\Big)\;\middle|\; S_1=s'\right]\\
&= r\Big(s,d_0(s)\Big)+\gamma\cdot\sum_{s'\in\mathcal{S}_{s,d_0(s)}}\mathbf{p}\Big(s'\Big|s,d_0(s)\Big)\cdot\mathbb{E}\left[\sum_{t'=0}^{+\infty}\gamma^{t'}\cdot r\Big(S^+_{t'},d^+_{t'}(S^+_{t'})\Big)\;\middle|\; S^+_0=s'\right]\\
&= r\Big(s,d_0(s)\Big)+\gamma\cdot\sum_{s'\in\mathcal{S}_{s,d_0(s)}}\mathbf{p}\Big(s'\Big|s,d_0(s)\Big)\cdot V^{d^+}(s').
\end{aligned}$$

The third and fourth lines come from the properties A.2 and I.1.1. In the fifth line, $(S^+_t)_{t\in\mathbb{N}}$ is defined as $S^+_t = S_{t+1}$. As well, $\forall t\in\mathbb{N}$, $\forall s\in\mathcal{S}$, $d^+_t(s) = d_{t+1}(s)$. ■

A.5 Proof of Theorem 2

Proof: Let $(V_n)_{n\in\mathbb{N}}$ be a Cauchy sequence in $(\mathcal{F}_B(\mathcal{S},\mathbb{R}),\|.\|_\infty)$. For each state $s\in\mathcal{S}$, $\Big(V_n(s)\Big)_{n\in\mathbb{N}}$ is a Cauchy sequence of $(\mathbb{R},|.|)$ because $|V_n(s)|\leqslant\|V_n\|_\infty$. Since $(\mathbb{R},|.|)$ is complete, $\forall s\in\mathcal{S}$, $\Big(V_n(s)\Big)_{n\in\mathbb{N}}$ has a limit in $\mathbb{R}$ that we denote by $V(s)$. A Cauchy sequence is bounded, therefore $\exists M>0$ such that $\forall n\in\mathbb{N}$, $\|V_n\|_\infty\leqslant M$. Thus, $\forall s\in\mathcal{S}$, $|V_n(s)|\to|V(s)|\leqslant M$, and then $V\in\mathcal{F}_B(\mathcal{S},\mathbb{R})$. Finally, as (V_n) is a Cauchy sequence,

$$\begin{aligned}
\forall\varepsilon>0,\exists N\geqslant 0,\text{ such that }\quad &\forall n\geqslant N,p\geqslant 0,\forall s\in\mathcal{S},\quad |V_n(s)-V_{n+p}(s)|<\varepsilon\\
\Rightarrow\quad &\forall n\geqslant N,\forall s\in\mathcal{S},\quad |V_n(s)-V(s)|<\varepsilon\\
\Rightarrow\quad &\forall n\geqslant N,\quad \|V_n-V\|_\infty<\varepsilon
\end{aligned}$$

Thus $V_n\longrightarrow V$ in $\mathcal{F}_B(\mathcal{S},\mathbb{R})$ when $n\to+\infty$, and then $(\mathcal{F}_B(\mathcal{S},\mathbb{R}),\|.\|_\infty)$ is a Banach space. ■

A.6 Proof of Property I.1.2

Proof: Let $(V,V')\in\mathcal{F}_B(\mathcal{S},\mathbb{R})^2$,

$$\begin{aligned}
\left\|\mathcal{B}^dV-\mathcal{B}^dV'\right\|_\infty &\leqslant \gamma\cdot\sup_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}_{s,d_0(s)}}\mathbf{p}\Big(s'\Big|s,d_0(s)\Big)\cdot|V(s')-V'(s')|\\
&\leqslant \gamma\cdot\sup_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}_{s,d_0(s)}}\mathbf{p}\Big(s'\Big|s,d_0(s)\Big)\cdot\|V-V'\|_\infty\\
&\leqslant \gamma\cdot\|V-V'\|_\infty
\end{aligned}$$

since $\forall s\in\mathcal{S}$, $\mathbf{p}\,(\,.\mid s,d_0(s)\,)$ is a probability distribution. ■

A.7 Proof of Property I.1.3

First, let us present the following result:

Property A.5

Let f and g be two functions defined on the finite set $\mathcal{A}$ and with values in $\mathbb{R}$:

$$\left|\max_{a\in\mathcal{A}} f(a) - \max_{a\in\mathcal{A}} g(a)\right| \leqslant \max_{a\in\mathcal{A}} |f(a) - g(a)| \tag{16}$$

Proof:

$$\begin{aligned}
& \forall a \in \mathcal{A}, \quad f(a) - g(a) \leqslant \max_{a'\in\mathcal{A}} |f(a') - g(a')| \\
\Rightarrow \quad & \forall a \in \mathcal{A}, \quad f(a) \leqslant \max_{a'\in\mathcal{A}} g(a') + \max_{a'\in\mathcal{A}} |f(a') - g(a')| \\
\Rightarrow \quad & \max_{a'\in\mathcal{A}} f(a') - \max_{a'\in\mathcal{A}} g(a') \leqslant \max_{a'\in\mathcal{A}} |f(a') - g(a')| \,.
\end{aligned}$$

Finally, the same inequalities hold starting with $g(a) - f(a)$, thus we get the result 16. ■

Here is the proof of Property I.1.3:

Proof: The contraction inequality of the operator $\mathcal{B}^d$ for $(V, V') \in \mathcal{F}_B(\mathcal{S}, \mathbb{R})^2$ is true for each strategy $(d) \in \mathcal{D}_\infty$, as stated by Property I.1.2:

$$\begin{aligned}
& \forall (d) \in \mathcal{D}_\infty, && \left\|\mathcal{B}^d V - \mathcal{B}^d V'\right\|_\infty \leqslant \gamma \cdot \|V - V'\|_\infty \\
\Rightarrow \quad & \forall (d) \in \mathcal{D}_\infty, \forall s \in \mathcal{S}, && \left|(\mathcal{B}^d V)(s) - (\mathcal{B}^d V')(s)\right| \leqslant \gamma \cdot \|V - V'\|_\infty \\
\Rightarrow \quad & \forall a \in \mathcal{A}, \forall s \in \mathcal{S}, && |(\mathcal{B}^a V)(s) - (\mathcal{B}^a V')(s)| \leqslant \gamma \cdot \|V - V'\|_\infty \\
\Rightarrow \quad & \forall s \in \mathcal{S}, && \max_{a\in\mathcal{A}} |(\mathcal{B}^a V)(s) - (\mathcal{B}^a V')(s)| \leqslant \gamma \cdot \|V - V'\|_\infty
\end{aligned}$$

and Property A.5 leads then to $\forall s \in \mathcal{S}$ $\left|\max_{a\in\mathcal{A}}(\mathcal{B}^a V)(s) - \max_{a\in\mathcal{A}}(\mathcal{B}^a V')(s)\right| \leqslant \gamma \cdot \|V - V'\|_\infty$

$$\Rightarrow \|\mathcal{B}^* V - \mathcal{B}^* V'\|_\infty \leqslant \gamma \cdot \|V - V'\|_\infty$$

■

A.8 Proof of Theorem 3

Proof: Here is how we show that the function V^* is the optimal value function, *i.e.*

$$V^*(s) = \sup_{d\in\mathcal{D}_\infty} V^d(s) = \sup_{d\in\mathcal{D}_\infty} \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t \cdot r\Big(S_t, d_t(S_t)\Big) \;\middle|\; S_0 = s\right].$$

Let $s \in \mathcal{S}$ and $d = (a, d^+)$ the strategy which consists in selecting action a at time step $t = 0$, and then actions $a_1 = d_0^+(s_1) := d_1(s_1), a_2 = d_1^+(s_2) := d_2(s_2)$, etc. Note that $d^+ \in \mathcal{D}_\infty$ as well,

and numbered from 0.

$$\begin{aligned}
\sup_{(d)\in\mathcal{D}_\infty} V^d(s) &:= \sup_{(d)\in\mathcal{D}_\infty} \mathbb{E}\left[\sum_{t=0}^{+\infty}\gamma^t\cdot r\Big(S_t,d_t(S_t)\Big)\,\middle|\, S_0=s\right]\\
&= \sup_{(a,d^+)\in\mathcal{D}_\infty}\left\{r(s,a)+\gamma\cdot\sum_{s'\in\mathcal{S}_{s,a}}\mathbf{p}\,(s'\mid s,a)\cdot V^{d^+}(s')\right\}\\
&= \max_{a\in\mathcal{A}}\left\{r(s,a)+\gamma\cdot\sum_{s'\in\mathcal{S}_{s,a}}\mathbf{p}\,(s'\mid s,a)\cdot\sup_{(d^+)\in\mathcal{D}_\infty}V^{d^+}(s')\right\} \qquad (17)\\
&= \max_{a\in\mathcal{A}}\left\{r(s,a)+\gamma\cdot\sum_{s'\in\mathcal{S}_{s,a}}\mathbf{p}\,(s'\mid s,a)\sup_{(d^+)\in\mathcal{D}_\infty}V^{d^+}(s')\right\}\\
&= (\mathcal{B}^*\sup_{(d^+)\in\mathcal{D}_\infty}V^{d^+})(s)=(\mathcal{B}^*\sup_{(d)\in\mathcal{D}_\infty}V^{d})(s).
\end{aligned}$$

The equality 17 is justified just like at the line 15 of the proof of Theorem 1.

Thus $V^*=\sup_{d\in\mathcal{D}_\infty}V^d$ as stated by the Fixed-Point Theorem. Indeed, as $\mathbb{B}^*$ is a contracting operator (see Property I.1.3), the solution V^* of the Dynamic Programming equation I.7 is unique.

First, as $d^*: s\mapsto a^*\in\operatorname{argmax}_{a\in A}(\mathcal{B}^aV^*)(s)$,

$$\mathcal{B}^{d^*}V^*=\max_{a\in\mathcal{A}}(\mathcal{B}^aV^*)(s)=\mathcal{B}^*V^*$$

The function V^* is then a fixed-point of the contracting operator $\mathcal{B}^{d^*}$ (see Property I.1.2). As noted earlier, V^{d^*} is a fixed-point of $\mathcal{B}^{d^*}$ too, and thus $V^{d^*}=V^*=\max_{(d)\in\mathcal{D}_\infty}V^d$. It means that d^* is an optimal strategy. It is thus shown that it is sufficient to look for stationary strategies because at least one of them, d^*, is optimal. ■

A.9 Proof of Theorem 4

Proof: First,

$$\left\|V^N-V^*\right\|_\infty=\left\|\mathcal{B}^*V^{N-1}-\mathcal{B}^*V^*\right\|_\infty\leqslant\gamma\cdot\left\|V^{N-1}-V^*\right\|_\infty\leqslant\gamma\cdot\left(\left\|V^{N-1}-V^N\right\|_\infty+\left\|V^N-V^*\right\|_\infty\right)$$

and then $\left\|V^N-V^*\right\|_\infty\leqslant\frac{\gamma}{1-\gamma}\cdot\left\|V^{N-1}-V^N\right\|_\infty$. Moreover,

$$\left\|V^{N-1}-V^N\right\|_\infty=\left\|(\mathcal{B}^*)^{N-1}V^0-(\mathcal{B}^*)^{N-1}V^1\right\|_\infty\leqslant\gamma^{N-1}\cdot\left\|V^0-V^1\right\|_\infty.$$

Finally,

$$\left\|V^N-V^*\right\|_\infty\leqslant\frac{\gamma^N}{1-\gamma}\cdot\left\|V^0-V^1\right\|_\infty.$$

■

A.10 Proof of Theorem 5

Proof: The Bellman equation for the strategy (d), is $V^d=\mathcal{B}^dV^d$, and the last iteration of the algorithm is $V^{N+1}=\mathcal{B}^*V^N=\mathcal{B}^dV^N$ (as (d^*) is greedy with respect to V_N, we consider V_{N+1} even if it is not actually computed). Thanks to these two equalities, it is possible to write

$$\begin{aligned}
\left\|V^d-V^{N+1}\right\|_\infty=\left\|\mathcal{B}^dV^d-\mathcal{B}^dV^N\right\|_\infty &\leqslant \gamma\cdot\left\|V^d-V^N\right\|_\infty\\
&\leqslant \gamma\cdot(\left\|V^d-V^{N+1}\right\|_\infty+\left\|V^{N+1}-V^N\right\|_\infty)
\end{aligned}$$

$$\Rightarrow\left\|V^d-V^{N+1}\right\|_\infty\leqslant\frac{\gamma}{1-\gamma}\cdot\left\|V^{N+1}-V^N\right\|_\infty\leqslant\frac{\gamma^{N+1}}{1-\gamma}\cdot\left\|V^1-V^0\right\|_\infty. \qquad (18)$$

Finally, thanks to results (I.8) and (18), we get the control of the strategy error:

$$\left\|V^d - V^*\right\|_\infty \leqslant \left\|V^d - V^N\right\|_\infty + \left\|V^N - V^*\right\|_\infty \leqslant \frac{2\cdot\gamma^N}{1-\gamma}\left\|V^1 - V^0\right\|_\infty .$$

■

A.11 Proof of theorem 6

Proof: If i_{t+1} is the current information, $\forall s' \in \mathcal{S}$,

$$\begin{aligned}
&b_{t+1}(s') \\
&:= \mathbb{P}\left(S_{t+1} = s' \mid I_{t+1} = i_{t+1}\right) \\
&= \frac{\mathbb{P}\left(S_{t+1} = s', O_{t+1} = o_{t+1} \mid I_t = i_t, a_t\right)}{\mathbb{P}\left(O_{t+1} = o_{t+1} \mid I_t = i_t, a_t\right)} \\
&= \frac{\sum_{s\in\mathcal{S}} \mathbb{P}\left(S_{t+1} = s', O_{t+1} = o_{t+1} \mid S_t = s, I_t = i_t, a_t\right)\cdot\mathbb{P}\left(S_t = s \mid I_t = i_t, a_t\right)}{\sum_{\tilde{s}\in\mathcal{S}} \mathbb{P}\left(O_{t+1} = o_{t+1} \mid S_t = \tilde{s}, I_t = i_t, a_t\right)\cdot\mathbb{P}\left(S_t = \tilde{s} \mid I_t = i_t, a_t\right)} \\
&= \frac{\sum_{s\in\mathcal{S}} \mathbb{P}\left(O_{t+1} = o_{t+1} \mid S_{t+1} = s', S_t = s, I_t = i_t, a_t\right)\cdot\mathbb{P}\left(S_{t+1} = s' \mid S_t = s, I_t = i_t, a_t\right)\cdot b_t(s)}{\sum_{\tilde{s}\in\mathcal{S}}\sum_{\tilde{s}'\in\mathcal{S}} \mathbb{P}\left(O_{t+1} = o_{t+1}, S_{t+1} = \tilde{s}' \mid S_t = \tilde{s}, I_t = i_t, a_t\right)\cdot b_t(\tilde{s})} \\
&= \frac{\sum_{s\in\mathcal{S}} \mathbf{p}\left(o_{t+1} \mid s', a_t\right)\cdot\mathbf{p}\left(s' \mid s, a_t\right)\cdot b_t(s)}{\sum_{\tilde{s}\in\mathcal{S}}\sum_{\tilde{s}'\in\mathcal{S}} \mathbf{p}\left(o_{t+1} \mid \tilde{s}', a_t\right)\cdot\mathbf{p}\left(\tilde{s}' \mid \tilde{s}, a_t\right)\cdot b_t(\tilde{s})} := u(b_t, a_t, o_{t+1})(s').
\end{aligned} \tag{19}$$

where line 19 is simply given by the fact that, for A, B, C subsets of Ω, $\mathbb{P}\left(A \mid B\cap C\right) = \frac{\mathbb{P}\left(A\cap B \mid C\right)}{\mathbb{P}\left(B \mid C\right)}$.

■

A.12 Proof of Theorem 7

Proof: We use here the following notations: $\widehat{A}_t = \{A_0, \ldots, A_t\}$, $\widehat{a}_t = \{a_0, \ldots, a_t\}$, $\widehat{O}_t = \{O_1, \ldots, O_t\}$, and $\widehat{o}_t = \{o_1, \ldots, o_t\}$. The belief at time step t can be written as a function of $\widehat{o}_t$, $\widehat{a}_{t-1}$ and b_0: the belief b_t is then denoted by $b_{b_0}^{i_t} = b_{b_0}^{\widehat{o}_t, \widehat{a}_{t-1}}$, and

$$b_{b_0}^{i_t}(s) = b_{b_0}^{\widehat{o}_t, \widehat{a}_{t-1}}(s) = \frac{\mathbb{P}\left(S_t = s, \widehat{O}_t = \widehat{o}_t \,\middle|\, \widehat{A}_{t-1} = \widehat{a}_{t-1}\right)}{\mathbb{P}\left(\widehat{O}_t = \widehat{o}_t \,\middle|\, \widehat{A}_{t-1} = \widehat{a}_{t-1}\right)} = u\Big(u\Big(\ldots u\left(b_0, a_0, o_1\right), \ldots\Big), a_{H-1}, o_H\Big)(s). \tag{20}$$

The following notation will be useful for the follow-up:

$$\mathbf{p}\left(\widehat{o}_t, s_t \mid \widehat{a}_{t-1}, s_0\right) = \mathbf{p}\left(o_t \mid s_t, a_{t-1}\right)\cdot \sum_{s_1,\ldots,s_{t-1}} \mathbf{p}\left(s_t \mid s_{t-1}, a_{t-1}\right)\cdot\prod_{i=1}^{t-1}\mathbf{p}\left(o_i \mid s_i, a_{i-1}\right)\cdot\mathbf{p}\left(s_i \mid s_{i-1}, a_{i-1}\right).$$

The numerator of the fraction (20) may be written

$$\mathbb{P}\left(S_t = s, \widehat{O}_t = \widehat{o}_t \,\middle|\, \widehat{A}_{t-1} = \widehat{a}_{t-1}\right) = \sum_{s_0\in\mathcal{S}} \mathbf{p}\left(\widehat{o}_t, s_t \mid \widehat{a}_{t-1}, s_0\right)\cdot b_0(s_0).$$

As well, the denominator of the fraction (20) is

$$\mathbb{P}\left(\widehat{O}_t = \widehat{o}_t \,\middle|\, \widehat{A}_{t-1} = \widehat{a}_{t-1}\right) = \sum_{s_t\in\mathcal{S}}\sum_{s_0\in\mathcal{S}} \mathbf{p}\left(\widehat{o}_t, s_t \mid \widehat{a}_{t-1}, s_0\right)\cdot b_0(s_0).$$

Finally, the probability of the system state at time step t conditioned on the random sequence of actions is

$$\begin{aligned}\mathbb{P}\left(S_t = s \mid \widehat{A}_{t-1} = \widehat{a}_{t-1}\right) &= \sum_{\substack{(s_0,\dots,s_{t-1})\\ \in \mathcal{S}^t}} \prod_{i=1}^{t} \mathbf{p}\left(s_i \mid s_{i-1}, a_{i-1}\right) \cdot b_0(s_0) \\ &= \sum_{\widehat{o}_t} \sum_{s_0 \in \mathcal{S}} \mathbf{p}\left(\widehat{o}_t, s_t \mid \widehat{a}_{t-1}, s_0\right) \cdot b_0(s_0).\end{aligned}$$

Then, the expectation of the reward conditioned on the random sequence of actions is

$$\begin{aligned}\mathbb{E}\left[r(S_t, A_t) \mid \widehat{A}_t = \widehat{a}_t\right] &= \sum_{s \in \mathcal{S}} r(s, a_t) \cdot \mathbb{P}\left(S_t = s \mid \widehat{A}_{t-1} = \widehat{a}_{t-1}\right) \\ &= \sum_{s \in \mathcal{S}} r(s, a_t) \cdot \sum_{s_0, s_1, \dots, s_{t-1}} \prod_{i=1}^{t} \mathbf{p}\left(s_i \mid s_{i-1}, a_{i-1}\right) \cdot b_0(s_0) \\ &= \sum_{s \in \mathcal{S}} r(s, a_t) \cdot \sum_{\widehat{o}_t} \sum_{s_0 \in \mathcal{S}} \mathbf{p}\left(\widehat{o}_t, s \mid \widehat{a}_{t-1}, s_0\right) \cdot b_0(s_0) \\ &= \sum_{\widehat{o}_t} \sum_{(s_0, s) \in \mathcal{S}^2} \mathbf{p}\left(\widehat{o}_t, s \mid \widehat{a}_{t-1}, s_0\right) \cdot b_0(s_0) \cdot r(s, a_t).\end{aligned}$$

Next, multiplying each term of the sum over observations $\widehat{o}_t$ by

$$1 = \frac{\sum_{(s^{(1)}, s'^{(1)}) \in \mathcal{S}^2} \mathbf{p}\left(\widehat{o}_t, s'^{(1)} \mid \widehat{a}_{t-1}, s^{(1)}\right) \cdot b_0(s^{(1)})}{\sum_{(s^{(2)}, s'^{(2)}) \in \mathcal{S}^2} \mathbf{p}\left(\widehat{o}_t, s'^{(2)} \mid \widehat{a}_{t-1}, s^{(2)}\right) \cdot b_0(s^{(2)})},$$

we get

$$\begin{aligned}&\sum_{\widehat{o}_t} \frac{\sum_{(s^{(1)}, s'^{(1)}) \in \mathcal{S}^2} \mathbf{p}\left(\widehat{o}_t, s'^{(1)} \mid \widehat{a}_{t-1}, s^{(1)}\right) \cdot b_0(s^{(1)})}{\sum_{(s^{(2)}, s'^{(2)}) \in \mathcal{S}^2} \mathbf{p}\left(\widehat{o}_t, s'^{(2)} \mid \widehat{a}_{t-1}, s^{(2)}\right) \cdot b_0(s^{(2)})} \cdot \sum_{(s_0, s) \in \mathcal{S}^2} \mathbf{p}\left(\widehat{o}_t, s \mid \widehat{a}_{t-1}, s_0\right) \cdot b_0(s_0) \cdot r(s, a_t) \\ &= \sum_{\widehat{o}_t} \sum_{(s^{(1)}, s'^{(1)})} \mathbf{p}\left(\widehat{o}_t, s'^{(1)} \,\middle|\, \widehat{a}_{t-1}, s^{(1)}\right) \cdot b_0(s^{(1)}) \cdot \sum_{s \in \mathcal{S}} \frac{\sum_{s_0 \in \mathcal{S}} \mathbf{p}\left(\widehat{o}_t, s \mid \widehat{a}_{t-1}, s_0\right) \cdot b_0(s_0)}{\sum_{(s^{(2)}, s'^{(2)})} \mathbf{p}\left(\widehat{o}_t, s'^{(2)} \,\middle|\, \widehat{a}_{t-1}, s^{(2)}\right) \cdot b_0(s^{(2)})} \cdot r(s, a_t) \\ &= \sum_{\widehat{o}_t} \sum_{(s^{(1)}, s'^{(1)})} \mathbf{p}\left(\widehat{o}_t, s'^{(1)} \,\middle|\, \widehat{a}_{t-1}, s^{(1)}\right) \cdot b_0(s^{(1)}) \cdot \sum_{s \in \mathcal{S}} \frac{\sum_{s_0 \in \mathcal{S}} \mathbf{p}\left(\widehat{o}_t, s \mid \widehat{a}_{t-1}, s_0\right) \cdot b_0(s_0)}{\sum_{(s^{(2)}, s'^{(2)})} \mathbf{p}\left(\widehat{o}_t, s'^{(2)} \,\middle|\, \widehat{a}_{t-1}, s^{(2)}\right) \cdot b_0(s^{(2)})} \cdot r(s, a_t) \\ &= \sum_{\widehat{o}_t} \sum_{(s^{(1)}, s'^{(1)})} \mathbf{p}\left(\widehat{o}_t, s'^{(1)} \,\middle|\, \widehat{a}_{t-1}, s^{(1)}\right) \cdot b_0(s^{(1)}) \cdot \sum_{s \in \mathcal{S}} b_{b_0}^{\widehat{o}_t, \widehat{a}_{t-1}}(s) \cdot r(s, a_t) \\ &= \sum_{\widehat{o}_t} \mathbb{P}\left(\widehat{O}_t = \widehat{o}_t \,\middle|\, \widehat{A}_{t-1} = \widehat{a}_{t-1}\right) \cdot \sum_{s \in \mathcal{S}} b_{b_0}^{\widehat{o}_t, \widehat{a}_{t-1}}(s) \cdot r(s, a_t) \\ &= \mathbb{E}\left[\sum_{s \in \mathcal{S}} B_t(s) \cdot r(s, A_t) \,\middle|\, \widehat{A}_t = \widehat{a}_t\right] \\ &= \mathbb{E}\left[r(B_t, A_t) \mid \widehat{A}_t = \widehat{a}_t\right]\end{aligned}$$

where $r(b, a) = \sum_{s \in \mathcal{S}} r(s, a) \cdot b(s)$. ■

A.13 Proof of Theorem 8

Proof: First, $\forall a_t \in \mathcal{A}, \forall o' \in \mathcal{O}$,

$$\begin{aligned}\mathbb{P}(O_{t+1}=o' \mid I_t=i_t,a_t) &= \sum_{s'\in\mathcal{S}} \mathbb{P}(O_{t+1}=o', S_{t+1}=s' \mid I_t=i_t,a_t)\\ &= \sum_{(s,s')\in\mathcal{S}^2} \mathbf{p}(o' \mid s',a_t)\cdot\mathbf{p}(s' \mid s,a_t)\cdot\mathbb{P}(S_t=s \mid I_t=i_t)\\ &= \sum_{(s,s')\in\mathcal{S}^2} \mathbf{p}(o' \mid s',a_t)\cdot\mathbf{p}(s' \mid s,a_t)\cdot b_t(s).\end{aligned}$$

where $b_t = b_{b_0}^{i_t}$ *i.e.* b_t is the belief obtained starting from b_0 and computed with information i_t. For the sake of readability, the result $\sum_{(s,s')\in\mathcal{S}^2} \mathbf{p}(o' \mid s',a_t)\cdot\mathbf{p}(s' \mid s,a_t)\cdot b_t(s)$ is denoted by $\mathbf{p}(o' \mid b_t,a_t)$. Then,

$$\begin{aligned}\mathbb{P}(B_{t+1}=b' \mid I_t=i_t,a_t) &= \mathbb{E}\left[\mathbb{1}_{\{B_{t+1}=b'\}} \,\middle|\, I_t=i_t,a_t\right]\\ &= \sum_{o'\in\mathcal{O}} \mathbb{P}(O_{t+1}=o' \mid I_t=i_t,a_t)\cdot\mathbb{1}_{\left\{u\left(b_{b_0}^{i_t},a_t,o'\right)=b'\right\}}\\ &= \sum_{o'\in\mathcal{O}}\sum_{(s,s')\in\mathcal{S}^2} \mathbf{p}(o' \mid s',a_t)\cdot\mathbf{p}(s' \mid s,a_t)\cdot b_t(s)\cdot\mathbb{1}_{\{u(b_t,a_t,o')=b'\}}\\ &= \sum_{o'\in\mathcal{O}} \mathbf{p}(o' \mid b_t,a_t)\cdot\mathbb{1}_{\{u(b_t,a_t,o')=b'\}}, \qquad (21)\end{aligned}$$

and thus $\mathbb{P}(B_{t+1}=b' \mid I_t,a_t)$ is $b_{b_0}^{I_t}$-measurable, *i.e.* B_t-measurable. Indeed, it is shown that $\mathbb{P}(B_{t+1}=b' \mid I_t=i_t,a_t)$ is a measurable function of $b_t = b_{b_0}^{i_t}$ when $a_t \in \mathcal{A}$ is fixed.

In order to show the equality (I.15) asserting that the belief process is a Markov process, it is sufficient to show the following equation (see Definition A.1): $\forall (b,b') \in \left(\mathbb{P}_{b_0}^{\mathcal{S}}\right)^2$:

$$\int_{\{B_t=b\}} \mathbb{E}\left[\mathbb{1}_{\{B_{t+1}=b'\}} \,\middle|\, I_t,a_t\right] d\mathbb{P} = \int_{\{B_t=b\}} \mathbb{1}_{\{B_{t+1}=b'\}} d\mathbb{P}. \qquad (22)$$

Indeed, as $\mathbb{P}(B_{t+1}=b' \mid I_t,a_t) = \mathbb{E}\left[\mathbb{1}_{\{B_{t+1}=b'\}} \,\middle|\, I_t,a_t\right]$ is B_t-measurable, it remains to show the equality (22) to prove that $\mathbb{E}\left[\mathbb{1}_{\{B_{t+1}=b'\}} \,\middle|\, I_t,a_t\right] = \mathbb{E}\left[\mathbb{1}_{\{B_{t+1}=b'\}} \,\middle|\, B_t,a_t\right]$ $\mathbb{P}$-amolst surely, *i.e.* to show equality (I.15).

On the one hand, the left part of the equation (22) is

$$\begin{aligned}&\int_{\{B_t=b\}} \sum_{o'\in\mathcal{O}} \mathbf{p}(o' \mid b_t,a_t)\cdot\mathbb{1}_{\{u(b_t,a_t,o')=b'\}} d\mathbb{P}\\ &\qquad = \int_{\Omega} \mathbb{1}_{\{B_t=b\}} d\mathbb{P} \cdot \sum_{o'\in\mathcal{O}} \mathbf{p}(o' \mid b,a_t)\cdot\mathbb{1}_{\{u(b,a_t,o')=b'\}}. \qquad (23)\end{aligned}$$

thanks to the equality (21).

On the other hand, the right part of equation 22 is

$$\begin{aligned}\int_{\{B_t=b\}} \mathbb{1}_{\{B_{t+1}=b'\}} d\mathbb{P} &= \int_{\Omega} \mathbb{1}_{\{B_t=b\}}\cdot\mathbb{1}_{\{u(b,a_t,O_{t+1})=b'\}} d\mathbb{P}\\ &= \mathbb{E}\left[\mathbb{1}_{\{B_t=b\}}\cdot\mathbb{1}_{\{u(b,a_t,O_{t+1})=b'\}}\right]\\ &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{B_t=b\}}\cdot\mathbb{1}_{\{u(b,a_t,O_{t+1})=b'\}} \,\middle|\, I_t,a_t\right]\right] \qquad (24)\\ &= \mathbb{E}\left[\mathbb{1}_{\{B_t=b\}}\cdot\mathbb{E}\left[\mathbb{1}_{\{u(b,a_t,O_{t+1})=b'\}} \,\middle|\, I_t,a_t\right]\right] \qquad (25)\\ &= \mathbb{E}\left[\mathbb{1}_{\{B_t=b\}}\cdot\sum_{o'\in\mathcal{O}} \mathbf{p}(o' \mid b,a_t)\cdot\mathbb{1}_{\{u(b,a_t,o')=b'\}}\right], \qquad (26)\end{aligned}$$

which is also equal to result (23). Line (24) comes from Definition A.1. Line (25) comes from Property A.1 and the fact that B_t (and thus $\mathbb{1}_{\{B_t=b\}}$) is $\sigma(I_t)$-measurable. The last line (26) is given by the result (21).

The belief process $(B_t)_{t\geqslant 0}$ is thus a Markov process. ■

A.14 Proof of Theorem 9

Proof: Let $V : \mathbb{P}^{\mathcal{S}}_{b_0} \to \mathbb{R}$ be a PWLC function. Then, $\exists \Gamma \subset \mathbb{R}^{\mathcal{S}}$, $\#\Gamma < +\infty$ such that

$$V(b) = \max_{\alpha\in\Gamma}\left\{\sum_{s\in\mathcal{S}} b(s)\cdot\alpha(s)\right\} = \max_{\alpha\in\Gamma}\langle\alpha, b\rangle_{\mathbb{R}^{\mathcal{S}}}.$$

For $b \in \mathbb{P}^{\mathcal{S}}_{b_0}$, the Dynamic Programming Equation is

$$\begin{aligned}
(\mathcal{B}^*V)(b) &= \max_{a\in\mathcal{A}}\left\{\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \gamma\cdot\sum_{o'\in\mathcal{O}}\mathbf{p}(o'\mid b,a)\cdot V\big(u(b,a,o')\big)\right\}\\
&= \max_{a\in\mathcal{A}}\left\{\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \gamma\cdot\sum_{o'\in\mathcal{O}}\mathbf{p}(o'\mid b,a)\cdot\max_{\alpha\in\Gamma}\left\{\sum_{s'\in\mathcal{S}} u(b,a,o')(s')\cdot\alpha(s')\right\}\right\}\\
&= \max_{a\in\mathcal{A}}\left\{\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \max_{(\alpha_o)_{o\in\mathcal{O}}\in\Gamma^{\mathcal{O}}}\left\{\gamma\cdot\sum_{o'\in\mathcal{O}}\mathbf{p}(o'\mid b,a)\cdot\sum_{s'\in\mathcal{S}} u(b,a,o')(s')\cdot\alpha_{o'}(s')\right\}\right\}\\
&= \max_{a\in\mathcal{A}}\max_{(\alpha_o)_{o\in\mathcal{O}}\in\Gamma^{\mathcal{O}}}\left\{\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \left\{\gamma\cdot\sum_{o'\in\mathcal{O}}\mathbf{p}(o'\mid b,a)\cdot\sum_{s'\in\mathcal{S}} u(b,a,o')(s')\cdot\alpha_{o'}(s')\right\}\right\}
\end{aligned}$$

where $(\alpha_o)_{o\in\mathcal{O}}$ is an element of $\Gamma^{\mathcal{O}}$ *i.e.* is a set of vectors from $\Gamma \subset \mathbb{R}^{\mathcal{S}}$: each vector $\alpha_o \in \Gamma$ is indexed by an observation $o \in \mathcal{O}$.

Thereafter, given a belief $b \in \tilde{\mathcal{S}}$ and an action $a \in \mathcal{A}$, we use the notation $\mathbf{p}(s'\mid b,a) = \sum_{s\in\mathcal{S}}\mathbf{p}(s'\mid s,a)\cdot b(s)$. Then, the belief update (I.10) becomes $\forall s' \in \mathcal{S}$,

$$u(b,a,o')(s') = \frac{\mathbf{p}(o'\mid s',a)\cdot\mathbf{p}(s'\mid b,a)}{\mathbf{p}(o'\mid b,a)}.$$

The result applying $\mathcal{B}^*$ to V is then

$$\begin{aligned}
(\mathcal{B}^*V)(b) &= \max_{a\in\mathcal{A}}\max_{(\alpha_{o'})\in\Gamma^{\mathcal{O}}}\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \gamma\cdot\sum_{o'\in\mathcal{O}}\mathbf{p}(o'\mid b,a)\cdot\sum_{s'\in\mathcal{S}}\frac{\mathbf{p}(o'\mid s',a)\cdot\mathbf{p}(s'\mid b,a)}{\mathbf{p}(o'\mid b,a)}\cdot\alpha_{o'}(s')\\
&= \max_{a\in\mathcal{A},(\alpha_{o'})\in\Gamma^{\mathcal{O}}}\sum_{s\in\mathcal{S}} r(s,a)\cdot b(s) + \gamma\cdot\sum_{o'\in\mathcal{O}}\sum_{s'\in\mathcal{S}}\mathbf{p}(o'\mid s',a)\cdot\sum_{s\in\mathcal{S}}\mathbf{p}(s'\mid s,a)\cdot b(s)\cdot\alpha_{o'}(s')\\
&= \max_{a\in\mathcal{A},(\alpha_{o'})\in\Gamma^{\mathcal{O}}}\sum_{s\in\mathcal{S}}\left(r(s,a) + \gamma\cdot\sum_{o'\in\mathcal{O}}\sum_{s'\in\mathcal{S}}\mathbf{p}(o'\mid s',a)\cdot\mathbf{p}(s'\mid s,a)\cdot\alpha_{o'}(s')\right)\cdot b(s)\\
&= \max_{\alpha'\in\Gamma'}\sum_{s\in\mathcal{S}}\alpha'(s)\cdot b(s) = \max_{\alpha'\in\Gamma'}\langle\alpha', b\rangle_{\mathbb{R}^{\mathcal{S}}}.
\end{aligned}$$

where

$$\Gamma' = \left\{\alpha'(s) = r(s,a) + \gamma\cdot\sum_{o'\in\mathcal{O}}\sum_{s'\in\mathcal{S}}\mathbf{p}(o'\mid s',a)\cdot\mathbf{p}(s'\mid s,a)\cdot\alpha_{o'}(s') \;\middle|\; a\in\mathcal{A},\ \text{and } \forall o'\in\mathcal{O}, \alpha_{o'}\in\Gamma\right\}$$

which is finite, and which size is $\#\mathcal{A}\cdot(\#\Gamma)^{\#\mathcal{O}}$.

The function $b \mapsto \mathcal{B}^*V(b)$ is thus PWLC and elements of Γ' are called α-vectors. ■

A.15 Proof of Theorem 10

Proof: First, note that

$$\begin{array}{ll} \forall (x,y) \in \mathcal{X} \times \mathcal{Y}, & \pi(x,y) < \min\{\pi(x), \pi(y)\} \\ & \Updownarrow \\ \forall (x,y) \in \mathcal{X} \times \mathcal{Y}, & \pi(x,y) < \max\limits_{x'} \pi(x',y) = \pi(y) \text{ and } \pi(x,y) < \max\limits_{y'} \pi(x,y') = \pi(x) \\ & \Updownarrow \\ \forall (x,y) \in \mathcal{X} \times \mathcal{Y}, & \pi(x \mid y) = \pi(x,y) < \pi(x) \text{ and } \pi(y \mid x) = \pi(x,y) < \pi(y) \end{array}$$

Thus, as $\pi(x,y) \leqslant \min\{\pi(x), \pi(y)\}$ is always true, $\forall (x,y) \in \mathcal{X} \times \mathcal{Y}$, $\pi(x,y) = \min\{\pi(x), \pi(y)\}$ $\Leftrightarrow \forall (x,y) \in \mathcal{X} \times \mathcal{Y}$, $\pi(x \mid y) \geqslant \pi(x)$ or $\pi(y \mid x) \geqslant \pi(y)$. ■

A.16 Proof of Theorem 11

Proof: Let us suppose that X and Y are MS-independent: in terms of possibility distributions, it can be written $\forall x \in \mathcal{X}$, $\forall y \in \mathcal{Y}$, $\pi(x) = \pi(x \mid y)$ and $\pi(y) = \pi(y \mid x)$. Suppose also that $\exists x_0 \in \mathcal{X}$ such that $\pi(x_0) < 1$ *i.e.* variable X is not fully unknown. Thus, using the qualitative possibilistic conditioning (Definition I.2.7), we know that, since $\pi(x_0 \mid y) = \pi(x_0) < 1$, $\forall y \in \mathcal{Y}$, $\pi(x_0 \mid y) = \pi(x_0, y)$. Thus, $\pi(x_0, y) < 1$. Using also the conditioning of Definition I.2.7, we get that $\forall y \in \mathcal{Y}$, $\pi(y \mid x_0) = 1$, since the equality $\pi(x_0) = \pi(x_0, y)$ is the condition which leads to the possibility degree 1. As $\forall x \in \mathcal{X}$, $\forall y \in \mathcal{Y}$, $\pi(y) = \pi(y \mid x)$, $\forall y \in \mathcal{Y}$, $\pi(y) = 1$. As a conclusion, if X is not fully unknown, Y is is fully unknown: thus, if Y is not fully unknown, X is fully unknown. ■

A.17 Proof of Property I.2.1

Proof: First, if $f^* = \min_{\omega \in \Omega} f(\omega)$, then $\forall \omega \in \Omega$, $f^* \leqslant f(\omega)$. Thus $\forall \omega \in \Omega$, $1 - f^* \geqslant 1 - f(\omega)$, *i.e.* $1 - f^* = 1 - \min_{\omega \in \Omega} f(\omega) = \max_{\omega \in \Omega} \{1 - f(\omega)\}$: the equation (I.26) is proved. The equation (I.27) can be shown using the previous one: setting $f = 1 - g$, we get $\forall g : \Omega \to \mathcal{L}$ $\max_{\omega \in \Omega} g(\omega) = 1 - \min_{\omega \in \Omega} \{1 - g(\omega)\}$ *i.e.* it equality is shown.

Now, the equation (I.30) is shown considering the case where $\lambda \geqslant \max_{\omega \in \Omega} f(\omega)$: on the one hand, in this case, both parts of the equality are equal to $\max_{\omega \in \Omega} f(\omega)$ since $\forall \omega \in \Omega$, $\lambda \geqslant f(\omega)$. On the other hand, when $\lambda \leqslant \max_{\omega \in \Omega} f(\omega)$, right part of equation (I.30) is equal to λ. As $\forall \omega \in \Omega$, $\min\{\lambda, f(\omega)\} \leqslant \lambda$, and for each ω such that $f(\omega) \geqslant \lambda$, $\min\{\lambda, f(\omega)\} = \lambda$, the maximum is equal to λ: the left part of the equation (I.30) is equal to λ too. The equation (I.29) can be be proved using the previous equation: setting f to $1 - g$, and λ to $1 - \mu$, we get $\max_{\omega \in \Omega} \min\{1 - g(\omega), 1 - \mu\} = \min\{\max_{\omega \in \Omega}\{1 - g(\omega)\}, 1 - \mu\}$. And then, using equations (I.26) and (I.27), $\forall \mu \in \mathcal{L}$, $\forall g : \Omega \to \mathcal{L}$, $1 - \min_{\omega \in \Omega} \max\{g(\omega), \mu\} = 1 - \max\{\min_{\omega \in \Omega} g(\omega), \mu\}$, *i.e.* the equation is shown.

The equations (I.28) and (I.32) are trivial: with the operator min (resp. max), no matter what is the order of elements and the repetitions, as they are associative[2].

The inclusions (I.31) and (I.33) are shown as follows: let $\omega^* \in \operatorname{argmax}_{\omega \in \Omega} f(\omega)$. If $f(\omega^*) \leqslant \lambda$, $\forall \omega \in \Omega$, $\min\{f(\omega), \lambda\} = f(\omega)$ and thus $\operatorname{argmax}_{\omega \in \Omega} \min\{f(\omega), \lambda\} = \operatorname{argmax}_{\omega \in \Omega} f(\omega)$. Now, if $f(\omega^*) > \lambda$, $\max_{\omega \in \Omega} \min\{f(\omega), \lambda\} = \lambda$, and then $\operatorname{argmax}_{\omega \in \Omega} \min\{f(\omega), \lambda\} = \{\omega \mid f(\omega) \geqslant \lambda\}$: thus $\operatorname{argmax}_{\omega \in \Omega} f(\omega) \subseteq \operatorname{argmax}_{\omega \in \Omega} \min\{f(\omega), \lambda\}$, *i.e.* the inclusion (I.31) is shown. For the inclusion (I.33), if $f(\omega^*) > \lambda$, then the maximizing elements are in $\omega\{\omega \mid f(\omega) > \lambda\}$, where $f(\omega) = \max\{f(\omega), \lambda\}$. Thus maximizing elements are the same $\operatorname{argmax}_{\omega \in \Omega} f(\omega) = \operatorname{argmax}_{\omega \in \Omega} \max\{f(\omega), \lambda\}$. Otherwise, if $f(\omega^*) \leqslant \lambda$, $\forall \omega \in \Omega$, $\max\{f(\omega), \lambda\} = \lambda$, then $\operatorname{argmax}_{\omega \in \Omega} \max\{f(\omega), \lambda\} = \Omega$, and thus obviously $\operatorname{argmax}_{\omega \in \Omega} f(\omega) \subseteq \operatorname{argmax}_{\omega \in \Omega} \max\{f(\omega), \lambda\}$.

[2] An operator $*$ over $\mathcal{L}$ is associative if $\forall (\lambda_1, \lambda_2, \lambda_3) \in \mathcal{L}^3$, $(\lambda_1 * \lambda_2) * \lambda_3 = \lambda_1 * (\lambda_2 * \lambda_3)$.

Finally, let us show the equality (I.34). Note first that functions $A : \omega \mapsto \max\{\min\{\lambda, f(\omega)\}, g(\omega)\}$ and $B : \omega \mapsto \min\{\lambda, \max\{f(\omega), g(\omega)\}\}$ are equal to $\omega \mapsto \max\{f(\omega), g(\omega)\}$ on the set $\Omega_1 = \{\omega \mid \lambda \geqslant \max\{f(\omega), g(\omega)\}\}$. It is trivial for B, and as $\lambda \geqslant f(\omega)$ on this set, the result comes for A. On the set $\Omega_2 = \overline{\Omega_1} = \{\omega \mid \lambda \leqslant \max\{f(\omega), g(\omega)\}\}$, B is of course equal to λ, and if $f(\omega) \geqslant \lambda$, $A(\omega) = \max\{\lambda, g(\omega)\} \geqslant \lambda$, otherwise if $f(\omega) \leqslant \lambda$, $A(\omega) = \max\{f(\omega), g(\omega)\} \geqslant \lambda$ (on the set Ω_2). A and B are thus smaller on Ω_1. Indeed, they are both equal to $\omega \mapsto \max\{f(\omega), g(\omega)\} \leqslant \lambda$ on Ω_1, whereas on Ω_2, $N = \lambda$ and $A \geqslant \lambda$. Thus, if $\Omega_1 \neq \emptyset$, then the result is shown: the minimum is on Ω_1, where functions are equal. Otherwise, if $\Omega_1 = \emptyset$, $\forall \omega \in \Omega$, $\lambda \leqslant \max\{f(\omega), g(\omega)\}$. As $g(\omega^*) = 0$, then $f(\omega^*) \geqslant \lambda$, and $A(\omega^*) = \max\{\lambda, g(\omega^*)\} = \lambda = B(\omega^*)$. For the other $\omega \in \Omega$, $A(\omega) = \max\{\lambda, g(\omega)\} \geqslant \lambda$ or $A(\omega) = \max\{f(\omega), g(\omega)\} \geqslant \lambda$, thus the minimum of both functions, λ, is reached with ω^*, where A and B are equal. The equality (I.35) can be proved using the equality (I.34): if we set $g = 1 - h$, g fulfills the condition $\exists \omega^* \in \Omega$ such that $g(\omega^*) = 0$. Setting $\lambda = 1 - \mu$ and $f' = 1 - f$, we get

$$\min_{\omega\in\Omega} \max\Big\{\min\{1-\mu, 1-f'(\omega)\}, 1-h(\omega)\Big\} = \min_{\omega\in\Omega} \min\Big\{1-\mu, \max\{1-f'(\omega), 1-h(\omega)\}\Big\}.$$

Using equations (I.26) and (I.27) of Property I.2.1, we get

$$1 - \max_{\omega\in\Omega} \min\Big\{\max\{\mu, f'(\omega)\}, h(\omega)\Big\} = 1 - \max_{\omega\in\Omega} \max\Big\{\mu, \min\{f'(\omega), h(\omega)\}\Big\},$$

i.e. one minus equation (I.35). ■

A.18 Proof of the equality of Definition I.2.11

Proof: Let us show that (I.36) is equal to (I.37). Note first that, by definition, $f(\omega_i)$ is non-decreasing with $i \in \{1, \ldots, \#\Omega\}$. Note as well that μ is monotone, and then $\mu(A_i) = \mu(\{\omega_i, \ldots, \omega_{\#\Omega}\}) \geqslant \mu(A_{i+1})$ since $A_{i+1} \subset A_i$. Let i^* be the highest $i \in \{1, \ldots, \#\Omega\}$ such that $\mu(A_{i^*}) \geqslant f(\omega_{i^*})$. As $\mu(A_1) = \mu(\Omega) = 1 \geqslant f(\omega_1)$, i^* exists. For each $i \leqslant i^*$, $\min\{f(\omega_i), \mu(A_i)\} = f(\omega_i)$, and for each $i > i^*$, $\min\{f(\omega_i), \mu(A_i)\} = \mu(A_i)$, thanks to the definition of i^*. As $f(\omega_i)$ is non-decreasing and $\mu(A_i)$ is non-increasing with i, highest values of $\Big(\min\big\{f(\omega_i), \mu(A_i)\big\}\Big)_{i=1}^{\#\Omega-1}$ are $f(\omega_{i^*})$ and $\mu(A_{i^*+1})$.

If $f(\omega_{i^*}) \leqslant \mu(A_{i^*+1})$, then (I.36) is equal to $\mu(A_{i^*+1})$. As well, $\max\{f(\omega_{i^*}), \mu(A_{i^*+1})\} = \mu(A_{i^*+1})$. Using the definition of i^*, and as μ is monotone, $f(\omega_{i^*+1}) > \mu(A_{i^*+1}) \geqslant \mu(A_{i^*+2})$. This implies that $\max\{f(\omega_{i^*+1}), \mu(A_{i^*+2})\} = f(\omega_{i^*+1})$. As $\mu(A_{i+1})$ is non-increasing with i, and $f(\omega_i)$ non-decreasing, $\mu(A_{i^*+1})$ and $f(\omega_{i^*+1})$ are the lowest values of $\Big(\max\big\{f(\omega_i), \mu(A_{i+1})\big\}\Big)_{i=1}^{\#\Omega-1}$. By definition of i^*, $f(\omega_{i^*+1}) > \mu(A_{i^*+1})$, and thus formula (I.37) is also equal to $\mu(A_{i^*+1})$): (I.36) and (I.37) are equal.

If $f(\omega_{i^*}) > \mu(A_{i^*+1})$, then formula (I.36) is equal to $f(\omega_{i^*})$, and $\max\{f(\omega_{i^*}), \mu(A_{i^*+1})\} = f(\omega_{i^*})$. As $f(\omega_i)$ is non-decreasing with i, and thanks to the definition of i^*, $f(\omega_{i^*-1}) \leqslant f(\omega_{i^*}) \leqslant \mu(A_{i^*})$ and thus $\max\{f(\omega_{i^*-1}), \mu(A_{i^*})\} = \mu(A_{i^*})$. As previously ($\mu(A_{i+1})$ non-increasing and $f(\omega_i)$ non-decreasing), $\mu(A_{i^*})$ and $f(\omega_{i^*})$ are the lowest values of $\Big(\max\big\{f(\omega_i), \mu(A_{i+1})\big\}\Big)_{i=1}^{\#\Omega-1}$. By definition of i^*, $f(\omega_{i^*}) \leqslant \mu(A_{i^*})$, and thus formula (I.37) is also equal to $f(\omega_{i^*})$.

Finaly, formula (I.36) and (I.37) are equal. ■

A.19 Proof of Theorem 12

Proof: First, let us rewrite the Sugeno integral of a function $f : \Omega \to \mathcal{L}$ with respect to a possibility measure Π, using formula (I.36):

$$\begin{aligned}\mathbb{S}_\Pi[f] &= \max_{i=1}^{\#\Omega} \min\{ f(\omega_i), \Pi(A_i) \} \\ &= \max_{i=1}^{\#\Omega} \min\left\{ f(\omega_i), \max_{j=i}^{\#\Omega} \pi(\omega_j) \right\} \\ &= \max_{\substack{(i,j)\in\{1,\dots,\#\Omega\}^2 \\ \text{s.t. } i\leqslant j}} \min\{ f(\omega_i), \pi(\omega_j) \}. \end{aligned} \tag{27}$$

Now, note that $\forall (i,j) \in \{1,\dots,\#\Omega\}^2$ such that $i < j$, $\min\{ f(\omega_i), \pi(\omega_j) \} \leqslant \min\{ f(\omega_j), \pi(\omega_j) \}$ since $f(\omega_i)$ is non-decreasing with i. As just shown, such pairs have a minimum lower than the minimum of an other pair: such pairs can thus be removed from the maximum operator in the equation (27): we get then the formula (I.38).

As well, using formula (I.37), the Sugeno integral of f with respect to a necessity measure $\mathcal{N}$ is

$$\begin{aligned}\mathbb{S}_\mathcal{N}[f] &= \min_{i=1}^{\#\Omega} \max\{ f(\omega_i), \mathcal{N}(A_{i+1}) \} \\ &= \min_{i=1}^{\#\Omega} \max\Big\{ f(\omega_i), 1 - \Pi\big(\{\omega_1,\dots,\omega_i\} \big) \Big\} \\ &= \min_{i=1}^{\#\Omega} \max\left\{ f(\omega_i), \min_{j=1}^{i} \{ 1 - \pi(\omega_j) \} \right\} \\ &= \min_{\substack{(i,j)\in\{1,\dots,\#\Omega\}^2 \\ \text{s.t. } i\geqslant j}} \min\{ f(\omega_i), 1 - \pi(\omega_j) \}. \end{aligned} \tag{28}$$

Note that the terms of $\min_{\substack{(i,j)\in\{1,\dots,\#\Omega\}^2 \\ \text{s.t. } i\geqslant j}}$, such that $i > j$, *i.e.* $\max\{ f(\omega_i), 1 - \pi(\omega_j) \}$, are greater or equal to $\max\{ f(\omega_j), 1 - \pi(\omega_j) \}$ (as $f(\omega_i)$ is non-dreasing with i) and can be removed from $\min_{(i,j)}$: the equation (I.39) is then deduced. ∎

A.20 Proof of Theorem 13

Proof: For the case $i = 0$, $\overline{U_0^*}(s) = \Psi(s)$, is obvious since no action has to be chosen. The case $i = 1$ consists in applying the formula (I.30) of Property I.2.1. Following sequence of equalities come from properties (I.2.1). Let i be in $\{2,\dots,H-1\}$ and $j = H - i$, and $\mathcal{T}_i$ the set of i-length system state trajectories $\mathcal{T} = (s_{j+1},\dots,s_H)$: $\forall s_j \in \mathcal{S}$,

$\overline{U_i^*}(s_j)$

$$
\begin{aligned}
&= \max_{(\delta)\in\Delta_i} \max_{\mathcal{T}\in\mathcal{T}_i} \min\left\{ \min_{t=j}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), \min_{t=j}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \\
&= \max_{(\delta)\in\Delta_i} \max_{s_{j+1}\in\mathcal{S}} \max_{\mathcal{T}\in\mathcal{T}_{i-1}} \min\left\{ \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big) \right\}, \min_{t=j}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \\
&= \max_{(\delta)\in\Delta_i} \max_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \max_{\mathcal{T}\in\mathcal{T}_{i-1}} \min\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), \min_{t=j}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \\
&= \max_{(\delta)\in\Delta_i} \max_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \min\left\{ \pi_j\Big(s_{j+1}\Big|s_j,\delta_j(s_j)\Big), \right.\right. \\
&\qquad\qquad \left.\left. \max_{\mathcal{T}\in\mathcal{T}_{i-1}} \min\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), \min_{t=j+1}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \right\} \\
&= \max_{\delta_j(s_j)\in\mathcal{A}} \max_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \min\left\{ \pi_j\Big(s_{j+1}\Big|s_j,\delta_j(s_j)\Big), \right.\right. \\
&\qquad\qquad \left.\left. \max_{(\delta)\in\Delta_{i-1}} \max_{\mathcal{T}\in\mathcal{T}_{i-1}} \min\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), \min_{t=j+1}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \right\} \\
&= \max_{a\in\mathcal{A}} \max_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,a\Big), \min\left\{ \pi_j\Big(s_{j+1}\Big|s_j,a\Big), \overline{U^*_{i-1}}(s_1) \right\} \right\} \\
&= \max_{a\in\mathcal{A}} \min\left\{ \rho_j\Big(s_j,a\Big), \max_{s_{j+1}\in\mathcal{S}} \min\left\{ \pi_j\Big(s_{j+1}\Big|s_j,a\Big), \overline{U^*_{i-1}}(s_1) \right\} \right\}.
\end{aligned}
$$

where the final preference function $\Psi(s)$ is denoted by $\rho_H(s,a)$ to simplify equations. This shows that the optimistic value function can be computed using the recursive formula (I.54). The strategy computed with formula (I.55) is indeed optimal, thanks to the inclusion (I.30) of Property I.2.1.

As well, for the pessimistic criterion, the case $i = 0$ is obvious too, and the case $i = 1$ consists in applying the formulae (I.26) and (I.29) of Property I.2.1. For $i \in \{2,\dots,H-1\}$, and $j = H - i$, using Property I.2.1,

$\underline{U^*_H}(s_j)$

$$
\begin{aligned}
&= \max_{(\delta)\in\Delta_i} \min_{\mathcal{T}\in\mathcal{T}_i} \max\left\{ \min_{t=j}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), 1-\min_{t=j}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \\
&= \max_{(\delta)\in\Delta_i} \min_{s_{j+1}\in\mathcal{S}} \min_{\mathcal{T}\in\mathcal{T}_{i-1}} \max\left\{ \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \min_{t=j+1}^{H} \rho_t(s_t,\delta_t(s_t)) \right\}, 1-\min_{t=j}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \\
&= \max_{(\delta)\in\Delta_i} \min_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \min_{\mathcal{T}\in\mathcal{T}_{i-1}} \max\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), 1-\min_{t=j}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \quad (29) \\
&= \max_{(\delta)\in\Delta_i} \min_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \min_{\mathcal{T}\in\mathcal{T}_{i-1}} \max\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), \max_{t=j}^{H-1} \left\{ 1-\pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \right\} \\
&= \max_{(\delta)\in\Delta_i} \min_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \max\left\{ 1-\pi_j\Big(s_{j+1}\Big|s_j,\delta_j(s_j)\Big), \right.\right. \\
&\qquad\qquad \left.\left. \min_{\mathcal{T}\in\mathcal{T}_{i-1}} \max\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), 1-\min_{t=j+1}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \right\} \\
&= \max_{\delta_j(s_j)\in\mathcal{A}} \min_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,\delta_j(s_j)\Big), \max\left\{ 1-\pi_j\Big(s_{j+1}\Big|s_j,\delta_j(s_j)\Big), \right.\right. \quad (30) \\
&\qquad\qquad \left.\left. \max_{(\delta)\in\Delta_{i-1}} \min_{\mathcal{T}\in\mathcal{T}_{i-1}} \max\left\{ \min_{t=j+1}^{H} \rho_t\Big(s_t,\delta_t(s_t)\Big), 1-\min_{t=j+1}^{H-1} \pi_t\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \right\} \right\} \right\} \\
&= \max_{a\in\mathcal{A}} \min_{s_{j+1}\in\mathcal{S}} \min\left\{ \rho_j\Big(s_j,a\Big), \max\left\{ 1-\pi_j\Big(s_{j+1}\Big|s_j,a\Big), \underline{U^*_{H-1}}(s_{j+1}) \right\} \right\} \\
&= \max_{a\in\mathcal{A}} \min\left\{ \rho_j\Big(s_j,a\Big), \min_{s_{j+1}\in\mathcal{S}} \max\left\{ 1-\pi_j\Big(s_{j+1}\Big|s_j,a\Big), \underline{U^*_{H-1}}(s_{j+1}) \right\} \right\}.
\end{aligned}
$$

where the equation (I.34) of Property I.2.1 is used for the row (29). The row (30) is explained by the following result: consider the function $F : \mathcal{S}\times\mathcal{A}\to\mathcal{L}$. Then,

$$\max_{\delta:\mathcal{S}\to\mathcal{L}} \min_{s\in\mathcal{S}} F(s,\delta(s)) = \min_{s\in\mathcal{S}} \max_{\delta:\mathcal{S}\to\mathcal{A}} F(s,\delta(s)).$$

Indeed, $\forall\delta : \mathcal{S}\to\mathcal{A}$, $\forall s\in\mathcal{S}$, $\min_{s'\in\mathcal{S}} F(s',\delta(s')) \leqslant \max_{\delta':\mathcal{S}\to\mathcal{A}} F(s,\delta'(s))$, and thus, $\max_{\delta:\mathcal{S}\to\mathcal{A}} \min_{s'\in\mathcal{S}} F(s',\delta(s')) \leqslant \min_{s\in\mathcal{S}} \max_{\delta':\mathcal{S}\to\mathcal{A}} F(s,\delta'(s))$. Now, consider $\delta^* : \mathcal{S}\to\mathcal{A}$ such that $\forall s\in\mathcal{S}$, $\delta^*(s)\in \mathrm{argmax}_{a\in\mathcal{A}} F(s,a)$. Then, $\forall\delta : \mathcal{S}\to\mathcal{A}$, $\min_{s\in\mathcal{S}} F(s,\delta^*(s)) \geqslant \min_{s\in\mathcal{S}} F(s,\delta(s))$, which implies that $\min_{s\in\mathcal{S}} F(s,\delta^*(s)) \geqslant \max_{\delta:\mathcal{S}\to\mathcal{A}} \min_{s\in\mathcal{S}} F(s,\delta(s))$, *i.e.* $\min_{s\in\mathcal{S}} \max_{\delta:\mathcal{S}\to\mathcal{A}} F(s,\delta(s)) \geqslant \max_{\delta:\mathcal{S}\to\mathcal{A}} \min_{s\in\mathcal{S}} F(s,\delta(s))$: the equality is shown.

Finally, the strategy computation (I.57) is explained by inclusions (I.31) and (I.33) of properties (I.2.1). ■

A.21 Proof of Theorem 14

Proof : Suppose that the belief state at time step $t\in\mathbb{N}$ is β_t. The joint distribution of the system state variable S_{t+1} and the observation variable O_{t+1} is

$$
\begin{aligned}
&\Pi\left(S_{t+1}=s', O_{t+1}=o' \mid\ I_t=i_t, a\right)\\
&\quad=\max_{s\in\mathcal{S}}\Pi\left(S_{t+1}=s', O_{t+1}=o', S_t=s \mid\ I_t=i_t, a_t\right) &(31)\\
&\quad=\max_{s\in\mathcal{S}}\min\left\{\Pi\left(S_{t+1}=s', O_{t+1}=o' \mid\ S_t=s, I_t=i_t, a_t\right), \Pi\left(S_t=s \mid\ I_t=i_t, a_t\right)\right\} &(32)\\
&\quad=\max_{s\in\mathcal{S}}\min\left\{\Pi\left(O_{t+1}=o' \mid\ S_{t+1}=s', a_t\right), \Pi\left(S_{t+1}=s' \mid\ S_t=s, a_t\right), \beta_t(s)\right\} &(33)\\
&\quad=\min\left\{\Pi\left(O_{t+1}=o' \mid\ S_{t+1}=s', a_t\right), \max_{s\in\mathcal{S}}\min\left\{\Pi\left(S_{t+1}=s' \mid\ S_t=s, a_t\right), \beta_t(s)\right\}\right\} &(34)\\
&\quad=\min\left\{\pi_t\left(o' \mid\ s', a_t\right), \max_{s\in\mathcal{S}}\min\left\{\pi_t\left(s' \mid\ s, a_t\right), \beta_t(s)\right\}\right\},
\end{aligned}
$$

denoted by $\pi_t\left(s', o' \mid\ \beta_t, a_t\right)$ to simplify notations: $\max_{s'\in\mathcal{S}}\pi_t\left(s', o' \mid\ \beta_t, a_t\right)$ is also denoted by $\pi\left(o' \mid\ \beta_t, a_t\right)$. Line (31) is the possibilistic marginalization over variable S_t. Line (32) is due to the definition of the conditioning, Definition I.2.7. Line (33) uses the Definition of the belief state, Definition I.2.16, and that S_t does not depend on the action a_t. Finally, line (34) is comes from equation (I.30) of Property I.2.1.

Suppose now that the observation received at time step $t+1$ is o_{t+1}: as, by definition, $\beta_{t+1}(s')=\Pi\left(S_{t+1}=s' \mid\ I_t=i_t, O_{t+1}=o_{t+1}, a_t\right)$, using the qualitative possibilistic conditioning (Definition I.2.7), we conclude that, $\forall s\in\mathcal{S}$,

$$
\beta_{t+1}(s')=\begin{cases}1 & \text{if } \pi_t\left(s', o_{t+1} \mid\ \beta_t, a_t\right)=\pi_t\left(o_{t+1} \mid\ \beta_t, a_t\right),\\ \pi_t\left(s', o_{t+1} \mid\ \beta_t, a_t\right) & \text{otherwise.}\end{cases}
$$

■

A.22 Proof of Theorem 15

Proof: Let us denote by $\pi\Big(s_H, \widehat{o}_H\Big|\beta_0, (\delta)\Big)$ the joint possibility degree of the last system state $s_H\in\mathcal{S}$ and the observation sequence $\widehat{o}_H$ when the strategy is $(\delta)=(\delta)_{t=0}^{H-1}$:
$\Pi\Big(S_H=s_H, \widehat{O}_H=\widehat{o}_H\ \Big|\ (\delta)\Big)$

$$
=\max_{(s_0,\ldots,s_{H-1})\in\mathcal{S}^H}\min_{t=0}^{H-1}\min\left\{\pi_t\Big(o_{t+1}\Big|s_{t+1}, \delta_t(i_t)\Big), \pi_t\Big(s_{t+1}\Big|s_t, \delta_t(i_t)\Big), \beta_0(s_0)\right\}.
$$

The possibility distribution over the observation sequence is also denoted by

$$
\pi\Big(\widehat{o}_H\Big|\beta_0, (\delta)\Big)=\max_{s_H\in\mathcal{S}}\pi\Big(s_H, \widehat{o}_H\Big|\beta_0, (\delta)\Big),
$$

and the one over the last state $s_H\in\mathcal{S}$ is denoted by

$$
\pi\Big(s_H\Big|\beta_0, (\delta)\Big)=\max_{\widehat{o}_H}\pi\Big(s_H, \widehat{o}_H\Big|\beta_0, (\delta)\Big).
$$

By definition, the qualitative belief state at the end of the execution, *i.e.* at time step $t=H$, is equal to $\beta_{\beta_0}^{\delta,\widehat{o}_H}(s)=\Pi\left(S_H=s \mid\ \widehat{O}_H=\widehat{o}_H, \Big(\delta_t(i_t)\Big)_{t=0}^{H-1}\right)$, see Definition I.2.16. It can be written as a function of $\widehat{o}_H$, (δ) and β_0, and this belief state is then denoted by $\beta_{\beta_0}^{\delta,\widehat{o}_H}$:

$$
\beta_{\beta_0}^{\delta,\widehat{o}_H}(s)=\begin{cases}1 & \text{if } \pi\Big(s, \widehat{o}_H\Big|\beta_0, (\delta)\Big)=\pi\Big(\widehat{o}_H\Big|\beta_0, (\delta)\Big),\\ \pi\Big(s, \widehat{o}_H\Big|\beta_0, (\delta)\Big) & \text{otherwise,}\end{cases}
$$

using the possibilistic conditioning, Definition I.2.7.

Following lines show the desired equality for the optimistic criterion: its rewriting, denoted by $\mathbb{S}_\Pi\Big[\max_{s\in\mathcal{S}}\min\{\Psi(s),B_H^\pi(s)\}\Big|\beta_0,(\delta)\Big]$, is equal to

$$\max_{\widehat{o}_H}\min\left\{\max_{s\in\mathcal{S}}\min\left\{\Psi(s),\beta_{b_0}^{\delta,\widehat{o}_H}(s)\right\},\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\}$$

$$=\max_{\widehat{o}_H,s\in\mathcal{S}}\min\left\{\Psi(s),\beta_{b_0}^{\delta,\widehat{o}_H}(s),\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\} \tag{35}$$

$$=\max_{s\in\mathcal{S}}\min\left\{\Psi(s),\max_{\widehat{o}_H}\min\left\{\Pi\left(S_H=s\mid\ \widehat{O}_H=\widehat{o}_H,\left(\delta_t(i_t)\right)_{t=0}^{H-1}\right),\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\}\right\} \tag{36}$$

$$=\max_{s\in\mathcal{S}}\min\left\{\Psi(s),\pi\Big(s\Big|\beta_0,(\delta)\Big)\right\} \tag{37}$$

$$=\mathbb{S}_\Pi\left[\Psi(S_H)\mid\ \beta_0,(\delta)\right]. \tag{38}$$

Line (35) comes using equality (I.30) of Property I.2.1. Line (36) uses the definition of the possibilistic belief state. Line (37) uses the definition of $\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)$, and the possibilistic conditioning. Finally, line (38) is a notation: it is used as the Sugeno integral is based on the distribution $\pi\Big(s_H\Big|\beta_0,(\delta)\Big)$.

As regards the pessimistic criterion (I.66), the rewriting is shown in the same way:
$\mathbb{S}_\mathcal{N}\Big[\min_{s\in\mathcal{S}}\max\{\Psi(s),1-B_H^\pi(s)\}\Big|\beta_0,(\delta)\Big]$

$$=\min_{\widehat{o}_H}\max\left\{\min_{s\in\mathcal{S}}\max\left\{\Psi(s),1-\beta_{b_0}^{\delta,\widehat{o}_H}(s)\right\},1-\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\}$$

$$=\min_{\widehat{o}_H,s\in\mathcal{S}}\max\left\{\Psi(s),1-\beta_{b_0}^{\delta,\widehat{o}_H}(s),1-\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\} \tag{39}$$

$$=\min_{s\in\mathcal{S}}\max\left\{\Psi(s),\min_{\widehat{o}_H}\max\left\{1-\Pi\left(S_H=s\mid\ \widehat{O}_H=\widehat{o}_H,\left(\delta_t(i_t)\right)_{t=0}^{H-1}\right),1-\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\}\right\} \tag{40}$$

$$=\min_{s\in\mathcal{S}}\max\left\{\Psi(s),1-\max_{\widehat{o}_H}\min\left\{\Pi\left(S_H=s\mid\ \widehat{O}_H=\widehat{o}_H,\left(\delta_t(i_t)\right)_{t=0}^{H-1}\right),\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)\right\}\right\} \tag{41}$$

$$=\min_{s\in\mathcal{S}}\max\left\{P(s),1-\pi\Big(s\Big|\beta_0,(\delta)\Big)\right\} \tag{42}$$

$$=\mathbb{S}_\mathcal{N}\left[P(S_H)\mid\ \beta_0,(\delta)\right]. \tag{43}$$

Line (39) comes using the equality (I.29) of Property I.2.1 and line (40) uses the definition of the possibilistic belief state. Line (41) uses the equatity (I.26) and the equality (I.27) of Property I.2.1. Line (42) uses the definition of $\pi\Big(\widehat{o}_H\Big|\beta_0,(\delta)\Big)$, and the possibilistic conditioning. Finally, as the Sugeno integral is based on the distribution $\pi\Big(s_H\Big|\beta_0,(\delta)\Big)$, the result can be denoted as line (43). ■

A.23 Proof of Theorem 16

Proof: First, $\forall a_t \in \mathcal{A}, \forall o' \in \mathcal{O}$,

$$\begin{aligned}\Pi\left(O_{t+1}=o' \mid I_t=i_t,a_t\right) &= \max_{s'\in\mathcal{S}} \Pi\left(O_{t+1}=o', S_{t+1}=s' \mid I_t=i_t,a_t\right)\\ &= \max_{(s,s')\in\mathcal{S}^2} \min\left\{\pi_t\left(o' \mid s',a_t\right), \pi_t\left(s' \mid s,a_t\right), \Pi\left(S_t=s \mid I_t=i_t\right)\right\}\\ &= \max_{(s,s')\in\mathcal{S}^2} \min\left\{\pi_t\left(o' \mid s',a_t\right), \pi_t\left(s' \mid s,a_t\right), \beta^{i_t}_{\beta_0}(s)\right\}.\end{aligned}$$

where $\beta^{i_t}_{\beta_0}$ is obtained starting from β_0 and with information i_t. To make the next equations clear, the next formula $\max_{(s,s')\in\mathcal{S}^2} \min\left\{\pi_t\left(o' \mid s',a_t\right), \pi_t\left(s' \mid s,a_t\right), \beta^{i_t}_{\beta_0}(s)\right\}$ is denoted by $\pi_t\left(o' \mid \beta^{i_t}_{\beta_0}, a_t\right)$. Then, as the belief state is a deterministic function of the current observation, the previous action, and the previous belief state,

$$\begin{aligned}\Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, I_t=i_t,a_t\right) &= \Pi\left(\cup_{\substack{o'\in\mathcal{O} \text{ s.t.}\\ \nu(\beta,a_t,o')=\beta'}} \left\{O_{t+1}=o'\right\} \,\middle|\, I_t=i_t,a_t\right)\\ &= \max_{\substack{o'\in\mathcal{O} \text{ s.t.}\\ \nu(\beta,a_t,o')=\beta'}} \Pi\left(O_{t+1}=o' \mid I_t=i_t,a_t\right)\\ &= \max_{\substack{o'\in\mathcal{O} \text{ s.t.}\\ \nu(\beta,a_t,o')=\beta'}} \pi_t\left(o' \mid \beta,a_t\right). \qquad (44)\end{aligned}$$

where $\beta^{i_t}_{\beta_0}(s)$ is denoted by β. The set of all the possible information sequences at a time step $t \geqslant 1$ is denoted by $\mathcal{I}_t = \mathcal{A}^t \times \mathcal{O}^t$. Thanks to the equation (44) for each belief $\beta \in \Pi^{\mathcal{S}}_{\mathcal{L}}$, $\Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, I_t=i_t,a_t\right)$ does not depend on the information $i_t \in \mathcal{I}_t$, provided i_t is in $\left\{i \in \mathcal{I}_t \mid \beta^i_{\beta_0}=\beta\right\}$. Thus, as $\Pi\left(I_t=i \mid B^\pi_t=\beta,a_t\right)=0$ if $\beta^i_{\beta_0} \neq \beta$, we can write

$$\begin{aligned}&\Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, B^\pi_t=\beta,a_t\right)\\ &\quad= \max_{i\in\mathcal{I}_t} \min\left\{\Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, I_t=i,a_t\right), \Pi\left(I_t=i \mid B^\pi_t=\beta,a_t\right)\right\}\\ &\quad= \max_{i\in\mathcal{I}_t} \min\left\{\Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, I_t=i^*,a_t\right), \Pi\left(I_t=i \mid B^\pi_t=\beta,a_t\right)\right\}\\ &\quad= \min\left\{\Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, I_t=i^*,a_t\right), \max_{i\in\mathcal{I}_t} \Pi\left(I_t=i \mid B^\pi_t=\beta,a_t\right)\right\}\\ &\quad= \Pi\left(B^\pi_{t+1}=\beta' \,\middle|\, I_t=i^*,a_t\right)\end{aligned}$$

where i^* is such that $\beta^{i^*}_{\beta_0}=\beta$, using the equation (I.30) of Property I.2.1, and the fact that $\max_{i\in\mathcal{I}_t} \Pi\left(I_t=i \mid B^\pi_t=\beta,a_t\right)=1$. ∎

B Proofs of Chapter II

B.1 Property linking $\mathbb{S}_\Pi$ and $\mathbb{S}_\mathcal{N}$

Property B.1

Let f be a function from Ω to $\mathcal{L}$:

$$\mathbb{S}_\Pi[1-f] = 1 - \mathbb{S}_\mathcal{N}[f]$$

Proof: By definition (see Definition I.2.11 and Theorem 12),

$$\begin{aligned}\mathbb{S}_\Pi[1-f] &= \max_{\omega\in\Omega} \min\{1-f(\omega), \pi(\omega)\} \\ &= \max_{\omega\in\Omega} \Big\{1 - \max\{f(\omega), 1-\pi(\omega)\}\Big\} \\ &= 1 - \min_{\omega\in\Omega} \max\{f(\omega), 1-\pi(\omega)\} \\ &= 1 - \mathbb{S}_\mathcal{N}[f],\end{aligned}$$

using equations (I.26) and (I.27) of Property I.2.1. ■

B.2 Proof of Property II.1.1

Proof: First, $\forall\omega\in\Omega$, $\max\{f(\omega), g(\omega)\} \geqslant f(\omega)$, and $\max\{f(\omega), g(\omega)\} \geqslant g(\omega)$. Thus, $\forall\omega\in\Omega$, $\max\{f(\omega), g(\omega)\} \geqslant \min\{f(\omega), \pi(\omega)\}$, and $\max\{f(\omega), g(\omega)\} \geqslant \min\{g(\omega), \pi(\omega)\}$. As well, note that $\forall\omega\in\Omega$, $\pi(\omega) \geqslant \min\{f(\omega), \pi(\omega)\}$, and $\forall\omega\in\Omega$, $\pi(\omega) \geqslant \min\{g(\omega), \pi(\omega)\}$. Thus, $\forall\omega\in\Omega$,

$$\min\Big\{\max\{f(\omega), g(\omega)\}, \pi(\omega)\Big\} \geqslant \min\{f(\omega), \pi(\omega)\}$$

and

$$\min\Big\{\max\{f(\omega), g(\omega)\}, \pi(\omega)\Big\} \geqslant \min\{g(\omega), \pi(\omega)\}.$$

Recall that the Sugeno integral $\mathbb{S}_\Pi[\max\{f, g\}]$ is equal to

$$\max_{\omega'\in\Omega} \min\Big\{\max\{f(\omega'), g(\omega')\}, \pi(\omega')\Big\}.$$

Using previous inequalities, $\forall\omega\in\Omega$,

$$\mathbb{S}_\Pi[\max\{f, g\}] \geqslant \min\{f(\omega), \pi(\omega)\}$$

and

$$\mathbb{S}_\Pi[\max\{f, g\}] \geqslant \min\{g(\omega), \pi(\omega)\},$$

and then

$$\mathbb{S}_\Pi[\max\{f, g\}] \geqslant \max_{\omega\in\Omega} \min\{f(\omega), \pi(\omega)\} = \mathbb{S}_\Pi[f],$$

and

$$\mathbb{S}_\Pi[\max\{f, g\}] \geqslant \max_{\omega\in\Omega} \min\{g(\omega), \pi(\omega)\} = \mathbb{S}_\Pi[g].$$

We can conclude that $\mathbb{S}_\Pi[\max\{f, g\}] \geqslant \max\{\mathbb{S}_\Pi[f], \mathbb{S}_\Pi[g]\}$.

Now, let us show that $\mathbb{S}_\Pi[\max\{f, g\}] \leqslant \max\{\mathbb{S}_\Pi[f], \mathbb{S}_\Pi[g]\}$: let us denote by ω^* an element from Ω such that

$$\omega^* \in \operatorname{argmax} \min\Big\{\max\{f(\omega), g(\omega)\}, \pi(\omega)\Big\}.$$

Thus, $\mathbb{S}_\Pi[\max\{f,g\}] = \min\Big\{\max\{f(\omega^*),g(\omega^*)\},\pi(\omega^*)\Big\}$. On the one hand, if $f(\omega^*) \geqslant g(\omega^*)$,

$$\begin{aligned}\mathbb{S}_\Pi[\max\{f,g\}] &= \min\{f(\omega^*),\pi(\omega^*)\}\\ &\leqslant \max_{\omega\in\Omega}\min\{f(\omega),\pi(\omega)\} = \mathbb{S}_\Pi[f] \leqslant \max\{\mathbb{S}_\Pi[f],\mathbb{S}_\Pi[g]\}.\end{aligned}$$

On the other hand, if $f(\omega^*) \leqslant g(\omega^*)$,

$$\begin{aligned}\mathbb{S}_\Pi[\max\{f,g\}] &= \min\{g(\omega^*),\pi(\omega^*)\}\\ &\leqslant \max_{\omega\in\Omega}\min\{g(\omega),\pi(\omega)\} = \mathbb{S}_\Pi[g] \leqslant \max\{\mathbb{S}_\Pi[f],\mathbb{S}_\Pi[g]\}.\end{aligned}$$

Finally, $\mathbb{S}_\Pi[\max\{f,g\}] = \max\{\mathbb{S}_\Pi[f],\mathbb{S}_\Pi[g]\}$.

Using Property B.1, equation (I.27) of Property I.2.1, and the previous result,

$$\begin{aligned}\mathbb{S}_\mathcal{N}[\min\{f,g\}] &= \mathbb{S}_\mathcal{N}[1-\max\{1-f,1-g\}]\\ &= 1-\mathbb{S}_\Pi[\max\{1-f,1-g\}]\\ &= 1-\max\{\mathbb{S}_\Pi[1-f],\mathbb{S}_\Pi[1-g]\}\\ &= \min\{1-\mathbb{S}_\Pi[1-f],1-\mathbb{S}_\Pi[1-g]\}\\ &= \min\{\mathbb{S}_\mathcal{N}[f],\mathbb{S}_\mathcal{N}[g]\}.\end{aligned}$$

■

B.3 Proof of Theorem 18

Proof: In order to make the following calculus lines easier to read, $\Psi(S_H)$ is denoted by $\rho_H(S_H,A_H)$, $\overline{\Psi}(B_t^\pi)$ is denoted by $\overline{\rho_H}(B_H^\pi,A_H)$, and $\underline{\Psi}(B_t^\pi)$ is denoted by $\underline{\rho_H}(B_H^\pi,A_H)$:

$$\mathbb{S}_\Pi\left[\overline{\mathcal{G}}\Big((S_t)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right] = \mathbb{S}_\Pi\left[\max_{t=0}^H \rho_t(S_t,A_t)\right] \tag{45}$$

$$= \max_{t=0}^H \mathbb{S}_\Pi[\rho_t(S_t,A_t)] \tag{46}$$

$$= \max_{t=0}^H \mathbb{S}_\Pi[\overline{\rho_t}(B_t^\pi,A_t)] \tag{47}$$

$$= \mathbb{S}_\Pi\left[\max_{t=0}^H \overline{\rho_t}(B_t^\pi,A_t)\right] \tag{48}$$

$$= \mathbb{S}_\Pi\left[\overline{\mathcal{G}}\Big((B_t^\pi)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right], \tag{49}$$

where $\overline{\rho_t}(B_t^\pi,A_t) = \max_{s\in\mathcal{S}}\min\{\rho_t(s,A_t),B_t^\pi(s)\}$. The definition of $\overline{\mathcal{G}}$ (see Definition II.1.1) explains the first line (45). Line (46) comes from the maxitivity of $\mathbb{S}_\Pi$, described in Property II.1.1. The Theorem 15 is applied to each terms of the maximum operator in line (47). Finally, line (48) uses Property II.1.1 and line (49) uses the definition of $\overline{\mathcal{G}}$ for belief states.

$$\mathbb{S}_\mathcal{N}\left[\overline{\mathcal{G}}\Big((S_t)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right] = \mathbb{S}_\mathcal{N}\left[\min_{t=0}^H \rho_t(S_t,A_t)\right] \tag{50}$$

$$= \min_{t=0}^H \mathbb{S}_\mathcal{N}[\rho_t(S_t,A_t)] \tag{51}$$

$$= \min_{t=0}^H \mathbb{S}_\mathcal{N}\left[\underline{\rho_t}(B_t^\pi,A_t)\right] \tag{52}$$

$$= \mathbb{S}_\mathcal{N}\left[\min_{t=0}^H \underline{\rho_t}(B_t^\pi,A_t)\right] \tag{53}$$

$$= \mathbb{S}_\mathcal{N}\left[\underline{\mathcal{G}}\Big((B_t^\pi)_{t=0}^H,(A_t)_{t=0}^{H-1}\Big)\right], \tag{54}$$

where $\underline{\rho_t}(B_t^\pi,A_t) = \min_{s\in\mathcal{S}}\max\{\rho_t(s,A_t),1-B_t^\pi(s)\}$. The definition of $\underline{\mathcal{G}}$ (see Definition II.1.1) explains line (50). The minitivity of $\mathbb{S}_\mathcal{N}$, described in Property II.1.1, allows to writte the line (51). For line (52), the Theorem 15 is applied to each terms of the minimum operator. Finally, Property II.1.1 is used for line (53), and line (54) comes from the definition of $\underline{\mathcal{G}}$ for belief states. ■

B.4 Proof of Theorem 19

Proof: We proceed by induction on $t \in \mathbb{N}$: as the initial visible state $s_{v,0}$ is known by the agent, only states $s = (s_v, s_h)$ for which $s_v = s_{v,0}$ are such that $\beta_0(s) > 0$. A belief over hidden states can be thus defined as $\beta_{h,0}(s_h) = \max_{s_v \in \mathcal{S}_v} \beta_0(s_v, s_h) = \beta_0(s_{v,0}, s_h)$.

At time t, if $\beta_t(s) = 0$ for each $s = (s_v, s_h) \in \mathcal{S}$ such that $s_v \neq s_{v,t}$, the same notation can be adopted: $\beta_{h,t}(s_h) = \beta_t(s_{v,t}, s_h)$. Thus, if the agent reaches state $s_{t+1} = (s_{v,t+1}, s_{h,t+1})$ using action $a_t \in \mathcal{A}$, and if $s' = (s'_v, s'_h)$ with $s'_v \neq s_{v,t+1}$, then $s'_v \neq o_{v,t+1}$ and:

$$\begin{aligned}\pi_t\left(o_{t+1}, s' \mid \beta_t, a_t\right) &= \min\left\{\pi_t\left(o_{t+1} \mid s', a_t\right), \max_{s \in \mathcal{S}} \min\left\{\pi_t\left(s' \mid s, a_t\right), \beta_t(s')\right\}\right\} \\ &= 0.\end{aligned}$$

thanks to Equation (II.15). Finally, belief update formula (I.62) ensures that $\beta_{t+1}(s') = 0$, since $0 = \pi_t\left(o_{t+1}, s' \mid \beta_t, a_t\right) < \pi_t\left(o_{t+1} \mid \beta_t, a_t\right)$ (as o_{t+1} is received, o_{t+1} is not impossible, otherwise the model is wrong). Then, β_{t+1} is entirely encoded by $(s_{v,t+1}, \beta_{h,t+1})$ with $s_{v,t+1} = o_{v,t+1}$ and $\beta_{h,t+1}(s_h) = \max_{s_v} \beta_{t+1}(s_v, s_h)$, $\forall s_h \in \mathcal{S}_h$. ■

B.5 Proof of Theorem 20

Proof: Starting from the standard belief update equation (I.62) of Theorem 14 with $o_{t+1} = (o_{v,t+1}, o_{h,t+1})$, and using equation (II.15) in the case of $o_{v,t+1} = s_{v,t+1}$ (*i.e.* $o'_v = s'_v$), we get that $\beta_{t+1}(s'_v, s'_h)$ is equal to the right part of the equation (II.16). As defined in Theorem 19, $\beta_{t+1,h}(s'_h) = \max_{\overline{s_v} \in \mathcal{S}_v} \beta_{t+1}(\overline{s_v}, s'_h)$; as shown in its proof, $\forall s'_v \in \mathcal{S}_v$ such that $s'_v \neq o_v$, $\beta_{t+1}(s_v, s_h) = 0$. Thus, $\beta_{h,t+1}(s'_h)$ is equal to $\beta_{t+1}(s'_v, s'_h)$ with $o'_v = s'_v$, *i.e.* to the right part of the equation. ■

B.6 Proof of Theorem 21

Proof: Using the classical dynamic programming equation, Theorem 19, and the facts $\mathcal{S}_v = \mathcal{O}_v$ and that $s'_v \neq o'_v$ is impossible,
$\widehat{U_i^*}(s_v, \beta_h) = \widehat{U_i^*}(\beta)$

$$\begin{aligned}
&= \max_{a \in \mathcal{A}} \widehat{\mathrm{M}} \left\{ \widehat{\rho_t}(\beta, a), \widehat{\mathbb{S}}\Big(\pi_t\left(o' \mid \beta, a\right), \widehat{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big)\Big)\right\} \\
&= \max_{a \in \mathcal{A}} \widehat{\mathrm{M}} \left\{ \widehat{\rho_t}(\beta, a), \widehat{\mathbb{S}}\Big(\pi_t\left(o'_v, o'_h \mid \beta, a\right), \widehat{U_{i-1}^*}\Big(\nu(\beta, a, o'_v, o'_h)\Big)\Big)\right\} \\
&= \max_{a \in \mathcal{A}} \widehat{\mathrm{M}} \left\{ \widehat{\rho_t}(s_v, \beta_h, a), \widehat{\mathbb{S}}\Big(\pi_t\left(s'_v, o'_h \mid \beta, a\right), \widehat{U_{i-1}^*}\Big(\nu(\beta, a, s'_v, o'_h)\Big)\Big)\right\} \\
&= \max_{a \in \mathcal{A}} \widehat{\mathrm{M}} \left\{ \widehat{\rho_t}(s_v, \beta_h, a), \widehat{\mathbb{S}}\Big(\pi_t\left(s'_v, o'_h \mid \beta, a\right), \widehat{U_{i-1}^*}\Big(s'_v, \nu_h(s_v, \beta_h, a, s'_v, o'_h)\Big)\Big)\right\}
\end{aligned}$$

where $\forall s_h \in \mathcal{S}_h$,

$$\begin{aligned}\beta'_h(s'_h) = \nu_h(s_v, \beta_h, a, s'_v, o'_h)(s'_h) = \max_{\overline{s}_v \in \mathcal{S}_v} \nu(\beta, a, s'_v, o'_h)(\overline{s}_v, s'_h) &= \nu(\beta, a, s'_v, o'_h)(s'_v, s'_h) \\ &= \beta_{t+1}(s'_v, s'_h).\end{aligned}$$

■

B.7 Proof of Theorem 23

Proof: Let β_t be a belief state in $\Pi_{\mathcal{L}}^{\mathcal{S}}$, a be an action in $\mathcal{A}$, and let us denote $\max_{s \in \mathcal{S}} \min\left\{\pi\left(s' \mid s, a\right), \beta_t(s)\right\}$ by $\beta_t^a(s')$, $\forall s' \in \mathcal{S}$.

For each $s' \in \mathcal{S}$, we denote by $\mathcal{P}_a(s')$ the set of system states which lead to s' selecting action $a \in \mathcal{A}$: $\{ s \in \mathcal{S} \mid \pi(s' \mid s, a) = 1 \}$. As the transition possibility distribution $\pi(s' \mid s, a)$ is deterministic, $\forall s \in \mathcal{S}$, $\exists! s' \in \mathcal{S}$ such that $\pi(s' \mid s, a) = 1$. It implies that $\forall (s', \tilde{s}) \in \mathcal{S}^2$, $\mathcal{P}_a(s') \cap \mathcal{P}_a(\tilde{s}) = \emptyset$ (otherwise, $\exists s \in \mathcal{S}$ leading to two different successors selecting action a, namely s' and $\tilde{s}$). Moreover, $\cup_{s' \in \mathcal{S}} \mathcal{P}_a(s') = \mathcal{S}$ (otherwise $\exists s \in \mathcal{S}$ without any successor $s' \in \mathcal{S}$). Thus, $\forall s' \in \mathcal{S}$, two cases are possible: either there is no system state $s \in \mathcal{S}$ leading to s' selecting action a, *i.e.* $\mathcal{P}_a(s') = \emptyset$, and then $\beta_t^a(s') = 0$ (since $\forall s \in \mathcal{S}$, $\pi(s' \mid s, a) = 0$). Otherwise, $\mathcal{P}_a(s') \neq \emptyset$ and $\beta_t^a(s') = \max_{s \in \mathcal{P}_a(s')} \beta_t(s)$. As, $\forall s' \in \mathcal{S}$, $\max_{s \in \mathcal{P}_a(s')} \beta_t(s) \leqslant \sum_{s \in \mathcal{P}_a(s')} \beta_t(s)$,

$$\sum_{s' \in \mathcal{S}} \beta_t^a(s') \leqslant \sum_{s' \in \mathcal{S}} \sum_{s \in \mathcal{P}_a(s')} \beta_t(s) = \sum_{s \in \mathcal{S}} \beta_t(s),$$

i.e. $\beta_t^a \preceq \beta_t$. Finally, since $\pi(o' \mid s', a) = 1$, $\min\{ \pi(o' \mid s', a), \beta_t^a(s') \} = \beta_t^a(s')$. Thus, using the belief update equation (I.62), $\beta_t^a = \beta_{t+1}$, and then $\beta_{t+1} \preceq \beta_t$. We can conclude that the first conditions lead to $\beta_{t+1} \preceq \beta_t$.

We prove now that the second conditions lead to the same conclusion. First note that if $\pi(s' \mid s, a) = \mathbb{1}_{\{ s' = s \}}$, then $\beta_t^a = \beta_t$. Two cases are now distinguished. First, suppose that it exists a system state $s' \in \mathcal{S}$ such that $\min\{ \pi(o' \mid s', a), \beta_t^a(s') \} = 1$: using the belief update (I.62), we get that $\forall s' \in \mathcal{S}$, $\beta_{t+1}(s') = \min\{ \pi(o' \mid s', a), \beta_t^a(s') \}$. Yet, $\min\{ \pi(o' \mid s', a), \beta_t^a(s') \} \leqslant \beta_t^a(s') = \beta_t(s')$, thus β_{t+1} is more specific than β_t (and then $\beta_{t+1} \preceq \beta_t$). Second, suppose that $\forall s' \in \mathcal{S}$, $\min\{ \pi(o' \mid s', a), \beta_t^a(s') \} < 1$. Then, only one $s' \in \mathcal{S}$ can maximize $\min\{ \pi(o' \mid s', a), \beta_t^a(s') \}$. Indeed, otherwise, it would be two system states s' and $\tilde{s}$ such that one of the following equalities hold: $\pi(o' \mid s', a) = \pi(o' \mid \tilde{s}, a)$; or $\pi(o' \mid s', a) = \beta_t^a(\tilde{s})$; or $\beta_t^a(s') = \beta_t^a(\tilde{s})$. We know that the belief update (I.62) defines $\beta_{t+1}(s')$ as the possibility degree $\beta_t^a(s')$, or $\pi(o' \mid s', a)$, or yet 1. Since $\forall s \in \mathcal{S}$, $\beta_0(s) = 1$ and since $\forall o' \in \mathcal{O}$, $\forall a \in \mathcal{A}$, $\forall (s', \tilde{s}) \in \mathcal{S}^2$, $\pi(o' \mid s', a) \neq \pi(o' \mid \tilde{s}, a)$, belief states obtained from β_0, using successive belief updates (I.62), never assign the same possibility degree to two different states, except if it is the possibility degree 1. It contradicts thus the fact that more than one system state $s' \in \mathcal{S}$ maximize $\min\{ \pi(o' \mid s', a), \beta_t^a(s') \}$. Let us note the maximizing system state s^*: using belief update (I.62), this state is the only system state such that $\beta_{t+1}(s^*) = 1$. System states $s' \in \mathcal{S}$ which do not maximize the joint possibility distribution ($s' \neq s^*$) are such that $\beta_{t+1}(s') = \min\{ \pi(o' \mid s', a), \beta_t^a(s') \} \leqslant \beta_t^a(s') = \beta_t(s')$. Thus if $\beta_t(s^*) = 1$, β_{t+1} is more specific than β_t (and then $\beta_{t+1} \preceq \beta_t$). In order to complete the proof, we now show that the inequality $\beta_{t+1} \preceq \beta_t$ remains true even if $\beta_t(s^*) < 1$. Let us denote by $\tilde{s} \in \mathcal{S}$ a system state such that $\beta_t(\tilde{s}) = 1$. We can affirm that $\beta_{t+1}(\tilde{s}) < \beta_t(s^*)$. Indeed, if this was not the case, then $\beta_{t+1}(\tilde{s}) \geqslant \beta_t(s^*)$, and

$$\beta_{t+1}(\tilde{s}) \geqslant \min\{ \pi(o' \mid s^*, a), \beta_t(s^*) \} = \min\{ \pi(o' \mid s^*, a), \beta_t^a(s^*) \},$$

where the last equality is due to the fact that $\beta_t^a = \beta_t$. Since $\beta_{t+1}(\tilde{s}) = \min\{ \pi(o' \mid \tilde{s}, a), \beta_t^a(\tilde{s}) \}$, s^* is not maximizing: it is a contradiction. Thus, $\beta_{t+1}(\tilde{s}) < \beta_t(s^*)$. Finally, $\sum_{s \in \mathcal{S} \setminus \{ s^*, \tilde{s} \}} \beta_{t+1}(s) \leqslant \sum_{s \in \mathcal{S} \setminus \{ s^*, \tilde{s} \}} \beta_t(s)$, $\beta_{t+1}(s^*) = \beta_t(\tilde{s}) = 1$ and $\beta_{t+1}(s^*) < \beta_t(\tilde{s})$, thus $\beta_{t+1} \preceq \beta_t$ (and even $\beta_{t+1} \prec \beta_t$). ■

C Proof of Theorem 22: optimality of the strategy computed by Algorithm 10

This appendix demonstrates that Algorithm 10 returns the maximum value of Equation (II.18) and an optimal strategy. Note that the computed optimal strategy is stationary, and is optimal regardless of the initial state. We recall that $\exists \widehat{a} \in \mathcal{A}$ such that $\forall s \in \mathcal{S}$, $\pi(s' \mid s, \widehat{a}) = 1$ if $s' = s$, and 0 otherwise. The existence of this action $\widehat{a}$ makes the maximum value of the criterion non-decreasing with respect to the horizon size. Let us denote by $(u_i^*)_{i \geqslant 0}$ the sequence of functions $\overline{U^*}$ computed by the algorithm: $\forall i \geqslant 1$, u_{i-1}^* is the function $\overline{U^*}$ at the beginning of the i^{th} iteration (line 5). That is, $u_0^* = \Psi$, and $\forall i \geqslant 1$, $\forall s \in \mathcal{S}$, $u_i^*(s) = \max_{a \in \mathcal{A}} \min\{\pi(s' \mid s, a), u_{i-1}^*(s)\}$. As well, $\forall s \in \mathcal{S}$, we denote by δ_i^* the strategy $\delta^* : \mathcal{S} \to \mathcal{A}$ computed after i iterations (according to the while loop). Finally, $\mathcal{T}_i$ is the set of i-length system state trajectories $(s_1, \ldots, s_i)$, and Δ_i the set of i-length strategies: $(\delta_t)_{t=0}^{i-1}$ such that $\delta : \mathcal{S} \to \mathcal{A}$.

Lemma C.1

$\forall s \in \mathcal{S}$, $\forall i \geqslant 0$, $u_i^*(s) \leqslant u_{i+1}^*(s)$.

Proof: Let $s_0 \in \mathcal{S}$. Looking at Algorithm 4 for optimistic finite-horizon π-MDP, and looking at Algorithm 10, we see that $\forall i \geqslant 0$, $u_i^* = \overline{U_i^*}$ (in the case of terminal preference only, *i.e.* $\forall t \geqslant 0$, $\forall a \in \mathcal{A}$, $\forall s \in \mathcal{S}$, $\rho_t(s, a) = 1$ since the global preference (II.3) is based on the miminum operator, see Section I.2.4), and thus, $\forall p \geqslant 1$

$$u_{p+1}^*(s_0) = \max_{(\delta) \in \Delta_{p+1}} \max_{\tau \in \mathcal{T}_{p+1}} \min\left\{ \min_{i=0}^{p} \pi\Big(s_{i+1} \Big| s_i, \delta_i(s_i)\Big), \Psi(s_{p+1}) \right\}.$$

Consider the particular trajectories $\tau' \in \mathcal{T}'_{p+1} \subset \mathcal{T}_{p+1}$ such that $\tau' = (s_1, \ldots, s_p, s_{p+1})$ and $s_p = s_{p+1}$. Consider also the particular policies $(\delta') \in \Delta'_{p+1} \subset \Delta_{p+1}$ such that $(\delta') = (\delta_0, \ldots, \delta_{p-1}, \widehat{\delta})$, where $\widehat{\delta}$ is the decision rule such that $\widehat{\delta}(s) = \widehat{a}$ (action "stay"). It is obvious that

$$u_{p+1}^*(s_0) \geqslant \max_{(\delta') \in \Delta'_{p+1}} \max_{\tau' \in \mathcal{T}'_{p+1}} \min\left\{ \min_{i=0}^{p} \pi\Big(s_{i+1} \Big| s_i, \delta'_i(s_i)\Big), \Psi(s_{p+1}) \right\}.$$

The right part of this inequality can be rewritten as

$$\max_{(\delta) \in \Delta_p} \max_{\tau \in \mathcal{T}_p} \min\left\{ \min_{i=0}^{p-1} \pi\Big(s_{i+1} \Big| s_i, \delta_i(s_i)\Big), \pi(s_p \mid s_p, \widehat{a}), \Psi(s_p) \right\} = u_p^*(s_0)$$

since $\pi(s_p \mid s_p, \widehat{a}) = 1$. Hence, $\forall p \geqslant 0$, $u_{p+1}^*(s_0) \geqslant u_p^*(s_0)$. ■

The meaning of this lemma is: it is always more possible to reach a state s from s_0 in at most $p+1$ steps than in at most p steps. As for each $s \in \mathcal{S}$, $\left(u_p^*(s)\right)_{p \in \mathbb{N}} \leqslant 1$, Lemma C.1 insures that the sequence $(u_p^*(s))_{p \in \mathbb{N}}$ converges. The next lemma shows that the convergence of this sequence occurs in finite time.

Lemma C.2

For all $\forall s \in \mathcal{S}$, the number of iterations of the sequence $\left(u_p^*(s)\right)_{p \in \mathbb{N}}$ up to convergence is bounded by $\#\mathcal{S} \times \#\mathcal{L}$.

Proof: Recall first that values of the possibility and preference distributions are in $\mathcal{L}$ which is finite and totally ordered: we can write $\mathcal{L}=\{0, l_1, l_2, \ldots, 1\}$ with $0 < l_1 < l_2 < \ldots < 1$. If two successive functions u_k^* and u_{k+1}^* are equal, then $\forall s \in \mathcal{S}$ sequences $\left(u_p^*(s)\right)_{p \geqslant k}$ are constant. Indeed this sequence can be defined by the recursive formula

$$u_p^*(s) = \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min\{\pi(s' \mid s, a), u_{p-1}^*(s')\}.$$

Thus if $\forall s \in \mathcal{S}$, $u_p^*(s) = u_{p-1}^*(s)$ then the next iteration $(p+1)$ faces the same situation $(u_{p+1}^*(s) = u_p^*(s), \forall s \in \mathcal{S})$. The slowest convergence can then be described as follows: for each $p \in \mathbb{N}$ only one $s \in \mathcal{S}$ is such that $u_{p+1}^*(s) > u_p^*(s)$. Moreover, for this s, if $u_p^*(s) = l_i$, then $u_{p+1}^*(s) = l_{i+1}$. We can conclude that for $p > \#\mathcal{L} \times \#\mathcal{S}$, the sequence is constant. ■

We conclude that the variable $\overline{U^*}$ of the algorithm converges to the maximal value of the criterion for an $(\#\mathcal{L} \times \#\mathcal{S})$-size horizon and can not be greater: the function $\overline{U^*}$ returned is thus optimal with respect to Equation (II.18) and is computed in a finite number of steps.

In the following, we prove the optimality of the strategy, based on the decision rule $\overline{\delta^*} : \mathcal{S} \to \mathcal{A}$, returned by Algorithm 10. For this purpose, we will construct a trajectory $\tau = (s_1, \ldots, s_p) \in \mathcal{T}_p$ of size p smaller than $\#\mathcal{S}$ which maximizes $\min\{\pi(\tau \mid s_0, (\delta)), \Psi(s_p)\}$ with strategy $(\overline{\delta^*}) = (\overline{\delta_t^*})_{t \geqslant 0}$ such that $\forall s \in \mathcal{S}$, $\forall t \geqslant 0$, $\overline{\delta_t^*}(s) = \overline{\delta^*}(s)$. The next two lemmas are needed for this construction and require some notations.

Let $s_0 \in \mathcal{S}$ and p be the smallest integer such that $\forall p' \geqslant p$, $u_{p'}^*(s_0) = \overline{U^*}(s_0)$, where $\overline{U^*}$ is here the optimal value of the infinite horizon criterion of Equation (II.18) returned by Algorithm 10 (variable $\overline{U^*}(s)$ of Algorithm 10 does not increase after p iterations). Algorithm 4 for optimistic finite-horizon π-MDP (in the case of terminal preference only) can be used to return an optimal strategy in the finite-horizon sence, see criterion (I.58), denoted by $(\delta^{(s_0)}) \in \Delta_p$ (this strategy is not stationary in general). With this notation, $\forall s \in \mathcal{S}$, $\overline{\delta^*}(s) = \delta_0^{(s)}(s)$, since $\delta_0^{(s)}(s)$ is the last selected action before convergence of $\left(\overline{U_i^*}(s)\right)_{i \geqslant 0}$ (see Algorithm 4).

Consider now a trajectory $\tau = (s_1, s_2, \ldots, s_p)$ which maximizes

$$\min\left\{ \min_{i=0}^{p-1} \pi\left(s_{i+1} \middle| s_i, \delta_i^{(s_0)}(s_i)\right), \Psi(s_p) \right\}.$$

This trajectory is called *optimal trajectory of minimal size from* s_0.

Lemma C.3

Let $\tau = (s_1, \ldots, s_p)$ be an optimal trajectory of minimal size from s_0. Then, $\forall k \in \{1, \ldots, p-1\}$,

$$\overline{U^*}(s_0) \leqslant u_{p-k}^*(s_k) \leqslant \overline{U^*}(s_k),$$

where $\overline{U^*}$ is the optimal value function returned by Algorithm 10, and $u_{p-k}^*(s_k)$ is equal to $\overline{U_{p-k}^*}(s_k)$ in Algorithm 4 (or defined above).

Proof: Let $k \in \{1, \ldots, p-1\}$.

$$\begin{aligned} \overline{U^*}(s_0) &= \min\left\{ \min_{i=0}^{p-1} \pi\left(s_{i+1} \middle| s_i, \delta_i^{(s_0)}(s_i)\right), \Psi(s_p) \right\} \\ &\leqslant \min\left\{ \min_{i=k}^{p-1} \pi\left(s_{i+1} \middle| s_i, \delta_i^{(s_0)}(s_i)\right), \Psi(s_p) \right\} \leqslant \overline{U_{p-k}^*}(s_k) = u_{p-k}^*(s_k) \leqslant \overline{U^*}(s_k) \end{aligned}$$

since $(u_p^*)_{p \in \mathbb{N}}$ is non-decreasing (see Lemma C.1), and $\forall i \geqslant 0$, $\overline{U_i^*} = u_i^*$ (see Algorithm 4) whose limit is $\overline{U^*}$. ■

Lemma C.4

Let $\tau = (s_1, \ldots, s_p)$ be an optimal trajectory of minimal size from s_0 and $k \in \{1, \ldots, p-1\}$. If $\overline{U^*}(s_0) = \overline{U^*}(s_k)$, then $\overline{\delta^*}(s_k) = \delta_k^{(s_0)}(s_k)$.

Proof: Suppose that $\overline{U^*}(s_0) = \overline{U^*}(s_k)$. Since $\overline{U^*}(s_0) \leqslant u_{p-k}^*(s_k) \leqslant \overline{U^*}(s_k)$ (Lemma C.3), we obtain that $u_{p-k}^*(s_k) = \overline{U^*}(s_k)$. The criterion in s_k is thus optimized within a $(p-k)$-horizon. Moreover a shorter horizon is not optimal: $\forall m \in \{1, \ldots, p-k\}$, $u_{p-k-m}^*(s_k) < \overline{U^*}(s_k)$ *i.e.* with

a $(p-k-m)$-size horizon the criterion in s_k is not maximized. Indeed if the contrary was true, the criterion in s_0 would be maximized within a $(p-m)$-size horizon: the strategy

$$\delta' = (\delta_0^{(s_0)}, \delta_1^{(s_0)}, \ldots, \delta_{k-1}^{(s_0)}, \delta_0^{(s_k)}, \ldots, \delta_{p-k-m-1}^{(s_k)}) \in \Delta_{p-m}$$

would be optimal. Indeed,

$$\begin{aligned} \overline{U^*}(s_0) &= \min\left\{ \min_{i=0}^{k-1} \pi\left(s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i)\right), u^*_{p-k}(s_k)\right\} \\ &= \min\left\{ \min_{i=0}^{k-1} \pi\left(s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i)\right), \overline{U^*}(s_k)\right\}. \end{aligned}$$

Then let $\overline{\tau} = (\overline{s}_1, \ldots, \overline{s}_{p-k-m}) \in \mathcal{T}_{p-k-m}$ be an optimal trajectory of minimal size from s_k. Setting $\overline{s_0} = s_k$, $\overline{\tau}$ thus maximizes $\min\left\{ \min_{i=0}^{p-k-m-1} \pi\left(\overline{s}_{i+1} \mid \overline{s}_i, \delta_i^{(s_k)}(\overline{s}_i)\right), \Psi(\overline{s}_{p-k-m})\right\}$, which is then equal to $\overline{U^*}(s_k)$. If $(s'_1, \ldots, s'_{p-m}) = (s_1, \ldots, s_{k-1}, \overline{s}_0, \ldots, \overline{s}_{p-k-m})$,

$$\overline{U^*}(s_0) = \min\left\{ \min_{i=0}^{p-m-1} \pi\Big(s'_{i+1}\Big|s'_i, \delta'_i(s_i)\Big), \Psi(s'_{p-m})\right\},$$

i.e. $\exists p' < p$ such that $\overline{U^*}(s_0) = u^*_{p'}(s_0)$: it contradicts the assumption that $(s_1, \ldots, s_p)$ is an optimal trajectory of minimum size. Thus $p-k$ is the smallest integer such that $u^*_{p-k}(s_k) = \overline{U}^*(s_k)$: we finally conclude that $\overline{\delta^*}(s_k)\ \left(:= \delta_0^{(s_k)}(s_k)\right) = \delta_k^{(s_0)}(s_k)$, as the action $\delta_k^{(s_0)}(s_k)$ is also selected during the iteration $p-k$, maximizing the same value function. ■

Theorem 29

Let $(\overline{\delta^*})$ be the strategy returned by Algorithm 10; $\forall s_0 \in \mathcal{S}$, there exists $p^* \leqslant \#\mathcal{S}$ and a trajectory $(s_1, \ldots, s_{p^*})$ such that

$$\overline{U^*}(s_0) = \min\left\{ \min_{i=0}^{p^*-1} \pi\Big(s_{i+1}\Big|s_i, \overline{\delta^*}(s_i)\Big), \Psi(s_{p^*})\right\},$$

i.e. $(\overline{\delta^*})$ is an optimal strategy.

Proof: Let s_0 be in $\mathcal{S}$ and $\tau \in \mathcal{T}_p$ be an optimal trajectory of minimal size p from s_0. If $\forall k \in \{1, \ldots, p-1\}$, $\overline{U^*}(s_k) = \overline{U^*}(s_0)$, then, using Lemma C.4, $\forall k \in \{1, \ldots, p-1\}$, $\overline{\delta^*}(s_k) := \delta_0^{(s_k)}(s_k) = \delta_k^{(s_0)}(s_k)$, and then the criterion in s_0 is maximized with $(\overline{\delta^*})$ since it is maximized with $(\delta_t^{(s_0)})_{t=0}^{p-1}$: the optimality of the strategy is shown.

Otherwise, let k be the smallest integer $\in \{1, \ldots, p-1\}$ such that $\overline{U^*}(s_k) > \overline{U^*}(s_0)$: the definition of k and Lemma C.3 implies that $\overline{U^*}(s_k) > \overline{U^*}(s_i) = \overline{U^*}(s_0)$, $\forall i \in \{0, \ldots, k-1\}$.

Reiterate beginning with $s_0^{(1)} = s_k$: let $p^{(1)}$ be the number of iterations until variable $\overline{U^*}(s^{(1)})$ of Algorithm 10 converges *i.e.* the smallest integer such that $\overline{U^*}(s_0^{(1)}) = u^*_{p^{(1)}}(s_0^{(1)})$. Let $\tau^{(1)} \in T_{p^{(1)}}$ be a trajectory which maximizes $\min\left\{ \min_{i=0}^{p^{(1)}-1} \pi\Big(s_{i+1}\Big|s_i, \delta_i^{(s_0^{(1)})}(s_i)\Big), \Psi(s^{(1)}_{p^{(1)}})\right\}$ ($\tau^{(1)}$ is an optimal trajectory of minimal size from $s_k = s_0^{(1)}$). We select $k^{(1)}$ in the same way as previously and reiterate beginning with $s_0^{(2)} = s^{(1)}_{k^{(1)}}$ which is such that $\overline{U^*}(s^{(1)}_{k^{(1)}}) > \overline{U^*}(s_0^{(1)})$, and $\overline{U^*}(s^{(1)}_{k^{(1)}}) > \overline{U^*}(s_i^{(1)})$ $\forall i \in \{0, \ldots, k^{(1)}-1\}$ etc. Lemma C.5 below shows that all selected states $(s_0, \ldots, s_{k-1}, s_0^{(1)}, \ldots, s^{(1)}_{k^{(1)}-1}, s_0^{(2)} \ldots, s^{(2)}_{k^{(2)}-1}, s_0^{(3)}, \ldots)$, are different. Thus this selection process ends since $\#\mathcal{S}$ is a finite set. The total number of selected states is denoted by $p^* = k + \sum_{i=1}^{q-1} k^{(i)} + p^{(q)}$ with $q \geqslant 0$ the number of new selected trajectories. Then the strategy

$$(\delta') = (\delta_0, \ldots, \delta_{k-1}, \delta_0^{(s_0^{(1)})}, \ldots, \delta^{(s_0^{(1)})}_{k^{(1)}-1}, \ldots, \delta^{(s_0^{(q)})}_{p^{(q)}})$$

corresponds to (δ^*) over $\tau' = (s'_1, \ldots, s'_{p^*}) = (s_0, s_1, \ldots, s_{k-1}, s_0^{(1)}, \ldots, s_{k^{(1)}-1}^{(1)}, \ldots, s_{p^{(q)}-1}^{(m)})$ and this strategy is optimal because $\overline{U^*}(s_0) = \overline{U}\Big(s_0, (\delta^*)\Big)$: indeed,

$$\begin{aligned}
\overline{U^*}(s_0) &= \min\left\{ \min_{i=0}^{k-1} \pi\left(s'_{i+1} \mid s'_i, \delta'(s'_i)\right), u^*_{p-k}(s_k) \right\} \\
&\leqslant \min\left\{ \min_{i=0}^{k-1} \pi\left(s'_{i+1} \mid s'_i, \delta'(s'_i)\right), \overline{U^*}(s_k) \right\} \\
&= \min\left\{ \min_{i=0}^{k^{(1)}-1} \pi\left(s'_{i+1} \mid s'_i, \delta'(s'_i)\right), u^*_{p^{(1)}-k^{(1)}}(s_{k^{(1)}}) \right\} \\
\ldots &\leqslant \min\left\{ \min_{i=0}^{p^*-1} \pi\left(s'_{i+1} \mid s'_i, \delta'(s'_i)\right), \Psi(s_{p^*}) \right\}.
\end{aligned}$$

The "$\leqslant$" signs are in fact "=" since otherwise we would find a strategy such that $\overline{U}(s_0, (\delta')) > \overline{U^*}(s_0)$, while $\overline{U^*}(s_0)$ is the maximal value. Thus (δ^*) is optimal: $\overline{U^*}(s_0) = \min\left\{ \min_{i=0}^{p^*-1} \pi\left(s'_{i+1} \mid s'_i, \delta^*(s'_i)\right), \Psi(s_{p^*}) \right\}$. ■

Lemma C.5

The process described in the previous proof in order to construct a trajectory maximizing the criterion with (δ^*) always selects different system states.

Proof: First, two equal states in the same selected trajectory $\tau^{(m)}$ would contradict the hypothesis that $p^{(m)}$ is the smallest integer such that $u^*_{p^{(m)}}(s_0^{(m)}) = \overline{U^*}(s_0^{(m)})$. Indeed let k and l be such that $0 \leqslant k < l \leqslant p^{(m)}$ and suppose that $s_k^{(m)} = s_l^{(m)}$. For clarity in the next calculations, we omit "(m)": $p = p^{(m)}$ and $\forall i \in \{0, \ldots, l\}$, $s_i = s_i^{(m)}$. We can write

$$u^*_{p-k}(s_k) = \min\left\{ \min_{i=k}^{l-1} \pi\left(s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i)\right), u^*_{p-l}(s_l) \right\} \leqslant u^*_{p-l}(s_l) = u^*_{p-l}(s_k),$$

as $s_l = s_k$. However $u^*_{p-k}(s_k) \geqslant u^*_{p-l}(s_k)$ (non-decreasing sequence and $p - k > p - l$). We finally get $u^*_{p-k}(s_k) = u^*_{p-l}(s_k)$, thus

$$\begin{aligned}
\overline{U^*}(s_0) &= \min\left\{ \min_{i=0}^{k-1} \pi\left(s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i)\right), u^*_{p-k}(s_k) \right\} \\
&= \min\left\{ \min_{i=0}^{k-1} \pi\left(s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i)\right), u^*_{p-l}(s_l) \right\} \\
&= \min\left\{ \min_{i=0,\ldots,k-1,l,\ldots,p-1} \pi\left(s_{i+1} \mid s_i, \delta_i^{(s_0)}(s_i)\right), \Psi(s_p) \right\}.
\end{aligned}$$

Consequently, a $(p^{(m)} - l + k)$-sized horizon is good enough to reach the optimal value: it is a contradiction. Finally, if we suppose that a state $\overline{s}$ appears two times in the sequence of selected states, then this state belongs to two different selected trajectories $\tau^{(m)}$ and $\tau^{(m')}$ (with $m < m'$). Lemma C.3, and the definition of $k^{(m)}$ which implies that $\overline{U^*}(s_0^{(m+1)})$ is strictly greater than the criterion's optimal values in each of the states $s_0^{(m)}, \ldots, s_{k^{(m)}-1}^{(m)}$, lead to the inequalities $\overline{U^*}(s_0^{(m')}) \leqslant \overline{U^*}(\overline{s}) < \overline{U^*}(s_0^{(m+1)})$, as $\overline{s}$ is in $\tau^{(m)}$ and in $\tau^{(m')}$. It is a contradiction. Indeed, as $m < m'$, the following inequality holds: $\overline{U^*}(s_0^{(m+1)}) \leqslant \overline{U^*}(s_0^{(m')})$. ■

D Proofs of Chapter III

D.1 Proof of Property III.2.1

Proof:

$$
\begin{aligned}
q^a &= \max_{(s'_v,\beta')\in\mathcal{S}_v\times\Pi_{\mathcal{L}}^{\mathcal{S}_h}} \min\left\{\pi(s'_v,\beta'|s_v,\beta,a),\overline{U^*}(s'_v,\beta')\right\}\\
&= \max_{X'\in\mathcal{S}_v\times\Pi_{\mathcal{L}}^{\mathcal{S}_h}} \min\left\{\min_{i=1}^{n}\pi\left(X'_i\mid parents(X'_i),a\right),\overline{U^*}(X')\right\}\\
&= \max_{X'_n\in\{\top,\bot\}} \min\Big\{\pi\left(X'_n\mid parents(X'_n),a\right),\cdots\\
&\qquad \max_{X'_2\in\{\top,\bot\}} \min\left\{\pi\left(X'_2\mid parents(X'_2),a\right),\right.\\
&\qquad\quad \left.\max_{X'_1\in\{\top,\bot\}} \min\{\pi\left(X'_1\mid parents(X'_1),a\right),\overline{U^*}(X')\}\right\}\cdots\Big\}
\end{aligned}
$$

where the last equation is due to the fact that, for any variables $x, y \in \mathcal{X},\mathcal{Y}$ finite spaces, and any functions $\varphi : \mathcal{X} \to \mathcal{L}$ and $\psi : \mathcal{Y} \to \mathcal{L}$, we have:

$$\max_{y\in\mathcal{Y}} \min\{\varphi(x),\psi(y)\} = \min\{\varphi(x),\max_{y\in\mathcal{Y}}\psi(y)\},$$

see the equation I.30 of Property I.2.1. ■

D.2 Proof of Theorem 24

Proof: First $S^1_{h,0},\ldots,S^l_{h,0}$ are initially NI-independent, see Definition I.2.6, *i.e.* $\forall(s^1,\ldots,s^l)\in\mathcal{S}^1_h\times\ldots\times\mathcal{S}^l_h$,

$$\Pi\left(S^1_{h,0}=s^1,\ldots,S^l_{h,0}=s^l\right)=\min\left\{\Pi\left(S^1_{h,0}=s^1\right),\ldots,\Pi\left(S^l_{h,0}=s^l\right)\right\}.$$

Then $\exists\left(\beta^j_0\right)_{j=1}^{l}\in\bigtimes_{j=1}^{l}\Pi_{\mathcal{L}}^{\mathcal{S}^j_h}$ such that $\forall s_h=(s^1,\ldots,s^l)\in\mathcal{S}^1_h\times\ldots\times\mathcal{S}^l_h$, $\beta_0(s_h)=\min_{j=1}^{l}\beta^j_0(s^j)$.

As discussed in Section III.3.2, the d-Separation criterion can be used to prove NI-independences on the DBN of Figure III.4: we show now that hidden variables of time step $t+1$ are NI-independent conditional on the information i_{t+1}. In fact, as shown in Figure III.4, given $1\leqslant i<j\leqslant l$, $S^i_{h,t+1}$ and $S^j_{h,t+1}$ are d-separated by the evidence $I_{t+1}=i_{t+1}$. Thus, $\forall s=(s^1,\ldots,s^l)\in\mathcal{S}_h$, $\Pi\left(S_{h,t+1}=s\mid I_{t+1}=i_{t+1}\right)=\min_{j=1}^{l}\Pi\left(S^j_{h,t+1}=s^j\;\middle|\;I_{t+1}=i_{t+1}\right)$ *i.e.* variables $(S^j_{h,t+1})_{j=1}^{l}$ are NI-independent conditional on information I_{t+1}. It shows that $\beta_{h,t+1}(s)=\min_{j=1}^{l}\beta^j_{t+1}(s^j)$.

Let us denote by $X\to Y$ the fact that there is an arrow from X to Y in the DBN. Note that the proved independence would not hold if the same observation variable O^k_{t+1}, $k\in\{1,\ldots,l\}$ concerned two different hidden state variables $S^i_{h,t+1}$ and $S^j_{h,t+1}$, *i.e.* if $S^i_{h,t+1}\to O^k_{t+1}$: and $S^j_{h,t+1}\to O^k_{t+1}$: indeed O^k_{t+1} is part of information I_{t+1}, thus there would be a convergent (towards O^k_{t+1}) relationship between $S^i_{h,t+1}$ and $S^j_{h,t+1}$ *i.e.* the hidden state variables would have been dependent (because d-connected) conditioned on information I_{t+1}. Moreover if the next hidden state variable $S^i_{h,t+1}$ depended on the current hidden state variable $S^j_{h,t}$, $(S^j_{h,t}\to S^i_{h,t+1})$ then $S^i_{h,t+1}$ and $S^j_{h,t+1}$ would have been dependent conditioned on information I_{t+1} because d-connected through $S^j_{h,t}$ ($S^j_{h,t}\to S^j_{h,t+1}$ is also true). ■

D.3 Proof of Lemma III.3.1

Proof: Let j be an integer in $\{1,\ldots,l\}$. Using Theorem 24, and M-independence assumptions of the DBN of Figure III.4 leading to the distributions (III.3), (III.4) and (III.5),

$$\begin{aligned}
&\Pi\left(O^j_{t+1}=o'_j, S^j_{h,t+1}=s'_j \;\middle|\; I_t=i_t,a_t\right)\\
&\quad=\max_{s^j_h\in\mathcal{S}^j_h}\min\left\{\Pi\left(O^j_{t+1}=o'_j, S^j_{h,t+1}=s'_j \;\middle|\; S^j_{h,t}=s^j_h, I_t=i_t,a_t\right), \Pi\left(S^j_{h,t}=s^j_h \;\middle|\; I_t=i_t,a_t\right)\right\}\\
&\quad=\max_{s^j_h\in\mathcal{S}^j_h}\min\left\{\Pi\left(O^j_{t+1}=o'_j, S^j_{h,t+1}=s'_j \;\middle|\; S^j_{h,t}=s^j_{h,t}, I_t=i_t,a_t\right), \beta^j_t(s^j_h)\right\}\\
&\quad=\max_{s^j_h\in\mathcal{S}^j_h}\min\left\{\Pi\left(O^j_{t+1}=o'_j \;\middle|\; S^j_{h,t+1}=s'_j, I_t=i_t,a_t\right), \Pi\left(S^j_{h,t+1}=s'_j \;\middle|\; I_t=i_t,a_t\right), \beta^j_t(s^j_h)\right\}\\
&\quad=\max_{s^j_h\in\mathcal{S}^j_h}\min\left\{\pi\left(o'_j \;\middle|\; s_{v,t}, s'_j, a_t\right), \Pi\left(s'_j \;\middle|\; s_{v,t+1}, s^j_h, a_t\right), \beta^j_t(s^j_h)\right\}
\end{aligned}$$

denoted then by $\pi\left(o'_j, s'_j \;\middle|\; s_{v,t}, \beta^j_t, a_t\right)$. The belief update can be then computed from this joint possibility distribution using the qualitative possibilistic conditioning (Definition I.2.7). ■

D.4 Proof of Theorem 25

Proof: First, as visible state variables $(S^i_{v,t+1})_{i=1}^m$ are d-separated by the evidence $I_t=i_t$, they are NI-independent: $\forall s_{v,t+1}=(s^1_{v,t+1},\ldots,s^m_{v,t+1})$, $\Pi(S_{v,t+1}=s_{v,t+1} \mid I_t=i_t,a_t) = \min_{i=1}^m \pi\left(s^i_{v,t+1} \;\middle|\; s_{v,t},a_t\right)$. The proof of Lemma III.3.1 shows that $\Pi\left(O^j_{t+1}=o^j_{t+1} \;\middle|\; I_t=i_t,a_t\right) = \Pi\left(O^j_{t+1}=o^j_{t+1} \;\middle|\; S_{v,t}=s_{v,t}, B^j_t=\beta^j_t,a_t\right)$, where β^j_t is the marginal belief state constructed from i_t. This distribution is then denoted by $\pi\left(o^j_{t+1} \;\middle|\; s_{v,t},\beta^j_t,a_t\right)$. Moreover, observation variables $(O^j_{t+1})_{j=1}^l$ are d-separated by the evidence $I_t=i_t$, and are then NI-independent: $\forall o_{t+1}=(o^1_{t+1},\ldots,o^l_{t+1})$, $\Pi(O_{t+1}=o_{t+1} \mid I_t=i_t,a_t) = \min_{j=1}^l \pi\left(o^j_{t+1} \;\middle|\; s_{v,t},\beta^j_t,a_t\right)$. Finally, $\forall i\in\{1,\ldots,m\}$ and $\forall j\in\{1,\ldots,l\}$, $S^i_{v,t+1}$ and O^j_{t+1} are d-separated by the evidence $I_t=i_t$, and then NI-independent. Thus, $\forall s_{v,t+1}=(s^1_{v,t+1},\ldots,s^m_{v,t+1})\in\mathcal{S}_v$, $\forall o_{t+1}=(o^1_{t+1},\ldots,o^l_{t+1})\in\mathcal{O}$,

$$\begin{aligned}
&\Pi(S_{v,t+1}=s_{v,t+1}, O_{t+1}=o_{t+1} \mid I_t=i_t,a_t)\\
&\qquad=\min\left\{\min_{i=1}^m \pi\left(s^i_{v,t+1} \;\middle|\; s_{v,t},a_t\right), \min_{j=1}^l \pi\left(o^j_{t+1} \;\middle|\; s_{v,t},\beta^j_t,a_t\right)\right\}.
\end{aligned}$$

As it only depends on the current visible state and belief state, we can denote it by $\Pi(S_{v,t+1}=s_{v,t+1}, O_{t+1}=o_{t+1} \mid B^\pi_t=\beta_t,a_t) = \pi(s_{v,t+1},o_{t+1} \mid \beta_t,a_t)$. Then, as B^π_{t+1} is equal to $\beta_{t+1}=(\beta^1_{t+1},\ldots,\beta^l_{t+1})$ with the possibility degree of the observations leading to it,

$$
\begin{aligned}
&\pi\left(s_{v,t+1}, \beta_{t+1} \mid \beta_t, a_t\right) \\
&\quad = \max_{\substack{(o^1,\dots,o^l)\in\mathcal{O}^j \text{ s.t. } \forall j,\\ \nu^j(s_{v,t},\beta_t^j,a_t,o^j)=\beta_{t+1}^j}} \pi\left(s_{v,t+1}, o_{t+1} \mid \beta_t, a_t\right) \\
&\quad = \max_{\substack{(o^1,\dots,o^l)\in\mathcal{O} \text{ s.t. } \forall j,\\ \nu^j(s_{v,t},\beta_t^j,a_t,o^j)=\beta_{t+1}^j}} \min\left\{ \min_{i=1}^{m} \pi\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right), \min_{j=1}^{l} \pi\left(o_{t+1}^j \,\middle|\, s_{v,t}, \beta_t^j, a_t\right) \right\} \\
&\quad = \min\left\{ \min_{i=1}^{m} \pi\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right), \max_{\substack{(o^1,\dots,o^l)\in\mathcal{O} \text{ s.t. } \forall j,\\ \nu^j(s_{v,t},\beta_t^j,a_t,o^j)=\beta_{t+1}^j}} \min_{j=1}^{l} \pi\left(o_{t+1}^j \,\middle|\, s_{v,t}, \beta_t^j, a_t\right) \right\} \\
&\quad = \min\left\{ \min_{i=1}^{m} \pi\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right), \min_{j=1}^{l} \pi\left(\beta_{t+1}^j \,\middle|\, s_{v,t}, \beta_t^j, a_t\right) \right\},
\end{aligned}
$$

using the equation (I.30) of Property I.2.1. ■

Bibliography

[1] Alexandre Albore, Héctor Palacios, and Hector Geffner. A translation-based approach to contingent planning. In Craig Boutilier, editor, *IJCAI*, pages 1623–1628, 2009. (Quoted on page 117.)

[2] Nahla Ben Amor, Hélène Fargier, and Wided Guezguez. Necessity-based choquet integrals for sequential decision making under uncertainty. In *Computational Intelligence for Knowledge-Based Systems Design, 13th International Conference on Information Processing and Management of Uncertainty, IPMU 2010, Dortmund, Germany, June 28 - July 2, 2010. Proceedings*, pages 521–531, 2010. (Quoted on page 148.)

[3] M. Araya-Lòpez, V. Thomas, O. Buffet, and F. Charpillet. A closer look at MOMDPs. In *Proceedings of the Twenty-Second IEEE International Conference on Tools with Artificial Intelligence (ICTAI-10)*, 2010. (Quoted on pages 23, 73, 81, 87, 91, 104, and 105.)

[4] Mauricio Araya-Lòpez, Olivier Buffet, Vincent Thomas, and François Charpillet. A pomdp extension with belief-dependent rewards. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *NIPS*, pages 64–72. Curran Associates, Inc., 2010. (Quoted on page 16.)

[5] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002. (Quoted on page 48.)

[6] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebric decision diagrams and their applications. *Form. Methods Syst. Des.*, 10(2-3):171–206, April 1997. (Quoted on pages 92 and 97.)

[7] Andrew G. Barto, Steven J. Bradtke, and Satinder P. Singh. Learning to act using real-time dynamic programming, 1993. (Quoted on page 47.)

[8] Richard Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515, 11 1954. (Quoted on pages 14 and 30.)

[9] Richard Bellman. A Markovian Decision Process. *Indiana Univ. Math. J.*, 6:679–684, 1957. (Quoted on page 7.)

[10] Nahla Ben Amor. *Qualitative possibilistic graphical models : from independence to propagation algorithms*. Thèse de doctorat, ISG - Université de Tunis, Tunis, juin 2002. (Quoted on pages 22 and 57.)

[11] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000. (Quoted on page 47.)

[12] B. Bonet. An ϵ-optimal grid-based algorithm for Partially Observable Markov Decision Processes. In C. Sammut and A. Hoffmann, editors, *Proc. 19th International Conf. on Machine Learning*, pages 51–58, Sydney, Australia, 2002. Morgan Kaufmann. (Quoted on page 47.)

[13] Blai Bonet. New grid-based algorithms for partially observable Markov decision processes: Theory and practice. (Quoted on pages 10 and 47.)

[14] Blai Bonet. Labeled RTDP: improving the convergence of real-time dynamic programming. In *In Proc. ICAPS-03*, pages 12–21. AAAI Press, 2003. (Quoted on pages 110 and 161.)

[15] Blai Bonet and Hector Geffner. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1641–1646, 2009. (Quoted on page 48.)

[16] Blai Bonet and Hector Geffner. Flexible and scalable partially observable planning with linear translations. In *AAAI*, 2014. (Quoted on page 117.)

[17] Blai Bonet and Judea Pearl. Qualitative mdps and pomdps: An order-of-magnitude approximation. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pages 61–68, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. (Quoted on page 163.)

[18] Byron Boots, Geoffrey J. Gordon, and Arthur Gretton. Hilbert space embeddings of predictive state representations. *CoRR*, abs/1309.6819, 2013. (Quoted on page 10.)

[19] Christian Borgelt, Jörg Gebhardt, and Rudolf Kruse. Graphical models. In *In Proceedings of International School for the Synthesis of Expert Knowledge (ISSEK'98*, pages 51–68. Wiley, 2002. (Quoted on page 22.)

[20] Léon Bottou. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91*, Nimes, France, 1991. EC2. (Quoted on page 12.)

[21] Léon Bottou. Stochastic gradient tricks. In Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller, editors, *Neural Networks, Tricks of the Trade, Reloaded*, Lecture Notes in Computer Science (LNCS 7700), pages 430–445. Springer, 2012. (Quoted on page 12.)

[22] Craig Boutilier. Correlated action effects in decision theoretic regression. In *UAI*, pages 30–37, 1997. (Quoted on page 94.)

[23] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121(1-2):49–107, 2000. (Quoted on page 92.)

[24] Xavier Boyen and Daphne Koller. Exploiting the architecture of dynamic systems. In *AAAI/IAAI*, pages 313–320, 1999. (Quoted on page 94.)

[25] Ronen I. Brafman. A heuristic variable grid solution method for POMDPs. In *In AAAI*, pages 727–733, 1997. (Quoted on pages 10 and 47.)

[26] Randal E Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24:293–318, 1992. (Quoted on page 92.)

[27] Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 54–61. Morgan Kaufmann Publishers, 1997. (Quoted on pages 10 and 43.)

[28] Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, AAAI'94, pages 1023–1028, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence. (Quoted on page 43.)

[29] Caroline Ponzoni Carvalho Chanel, Jean-Loup Farges, Florent Teichteil-Königsbuch, and Guillaume Infantes. POMDP solving: what rewards do you really expect at execution? In Thomas Agotnes, editor, *STAIRS*, volume 222 of *Frontiers in Artificial Intelligence and Applications*, pages 50–62. IOS Press, 2010. (Quoted on page 15.)

[30] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Patrick Fabiani. Décision séquentielle pour la perception active : p-pomdp versus pomdp. In *8èmes Journées Francophones Planification, Décision, et Apprentissage pour la conduite de systèmes (JFPDA)*, Lille, FR, 2013. (Quoted on page 16.)

[31] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Guillaume Infantes. Optimisation des processus décisionnels de Markov partiellement observables avec prise en compte explicite du gain d'information. In *17ème congrès francophone AFRIF-AFIA en Reconnaissance des Formes et Intelligence Artificielle (RFIA 2010)*, Caen, FR, 2010. (Quoted on page 15.)

[32] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. POMDP-based online target detection and recognition for autonomous UAVs. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, pages 955–960, 2012. (Quoted on page 9.)

[33] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. Multi-target detection and recognition by UAVs using online POMDPs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013. (Quoted on page 9.)

[34] Guillaume M. J-B. Chaslot, Mark H. M. Winands, H. Jaap van den Herik, Jos W. H. M. Uiterwijk, and Bruno Bouzy. Progressive strategies for Monte-Carlo tree search. *New Mathematics and Natural Computation*, 4:343–357, 2008. (Quoted on page 48.)

[35] Gustave Choquet. Theory of capacities. *Annales de l'institut Fourier*, 5:131–295, 1954. (Quoted on page 59.)

[36] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. (Quoted on pages 12 and 13.)

[37] S. Combéfis, D. Giannakopoulou, Ch. Pecheur, and M. Feary. A formal framework for design and analysis of human-machine interaction. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1801–1808, 2011. (Quoted on page 120.)

[38] Gert De Cooman. Integration and conditioning in numerical Possibility Theory. *Ann. Math. Artif. Intell.*, 32(1-4):87–123, 2001. (Quoted on page 149.)

[39] Ines Couso and Sébastien Destercke. Didier's groundhog day. 19(2):10–15, December 2012. (Quoted on page 48.)

[40] Luis M. de Campos and Juan F. Huete. Independence concepts in Possibility Theory: Part i. *Fuzzy Sets Syst.*, 103(1):127–152, April 1999. (Quoted on page 57.)

[41] B. De Finetti. *Theory of probability: a critical introductory treatment.* Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1974. (Quoted on pages 15 and 49.)

[42] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5(3):142–150, December 1989. (Quoted on pages 28, 94, and 151.)

[43] F. Dehais, M. Causse, and S. Tremblay. Mitigation of conflicts with automation. *Human Factors*, 53(3):448–460, 2011. (Quoted on page 120.)

[44] F. Dehais, M. Causse, F. Vachon, and S. Tremblay. Cognitive conflict in human-automation interactions: A psychophysiological study. *Applied Ergonomics*, 43(3):588–595, 2012. (Quoted on page 120.)

[45] KarinaValdivia Delgado, Cheng Fang, Scott Sanner, and Leliane Nunes de Barros. Symbolic bounded real-time dynamic programming. In Antônio Carlos da Rocha Costa, Rosa Maria Vicari, and Flavio Tonidandel, editors, *Advances in Artificial Intelligencev - SBIA 2010*, volume 6404 of *Lecture Notes in Computer Science*, pages 193–202. Springer Berlin Heidelberg, 2010. (Quoted on pages 110 and 161.)

[46] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Statist.*, 38(2):325–339, 04 1967. (Quoted on pages 15 and 55.)

[47] Finale Doshi-velez. The infinite partially observable Markov decision process. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 477–485. Curran Associates, Inc., 2009. (Quoted on page 14.)

[48] Nicolas Drougard, Didier Dubois, Jean-Loup Farges, and Florent Teichteil-Königsbuch. Planning in partially observable domains with fuzzy epistemic states and probabilistic dynamics. In *Scalable Uncertainty Management - 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings*, pages 220–233, 2015. (Quoted on pages 25, 163, and 211.)

[49] Nicolas Drougard, Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois. Qualitative possibilistic mixed-observable MDPs. In *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 192–201, Corvallis, Oregon, 2013. AUAI Press. (Quoted on pages 23, 92, 105, 131, 160, and 211.)

[50] Nicolas Drougard, Florent Teichteil-Königsbuch, Jean-Loup Farges, and Didier Dubois. Structured possibilistic planning using decision diagrams. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2257–2263, 2014. (Quoted on pages 24, 161, and 211.)

[51] D. Dubois and Ph. Fortemps. Selecting preferred solutions in the minimax approach to dynamic programming problems under flexible constraints. *European Journal of Operational Research*, 160(3):582–598, 2005. (Quoted on pages 132 and 133.)

[52] Didier Dubois. Possibility theory and statistical reasoning. *Computational Statistics and Data Analysis*, 51:47–69, 2006. (Quoted on pages 15, 20, and 148.)

[53] Didier Dubois and Hélène Fargier. Capacity refinements and their application to qualitative decision evaluation. In Claudio Sossai and Gaetano Chemello, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 5590 of *Lecture Notes in Computer Science*, pages 311–322. Springer Berlin Heidelberg, 2009. (Quoted on page 78.)

[54] Didier Dubois and HÃ©lÃ¨ne Fargier. Lexicographic refinements of sugeno integrals. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724 of *Lecture Notes in Computer Science*, pages 611–622. Springer Berlin Heidelberg, 2007. (Quoted on page 163.)

[55] Didier Dubois, Laurent Foulloy, Gilles Mauris, and Henri Prade. Probability-possibility transformations, triangular fuzzy sets and probabilistic inequalities. *Reliable Computing*, 10:2004, 2004. (Quoted on page 91.)

[56] Didier Dubois, Serafin Moral, and Henri Prade. A semantics for possibility theory based on likelihoods. *Journal of Mathematical Analysis and Applications*, 205(2):359 – 380, 1997. (Quoted on page 15.)

[57] Didier Dubois and Henri Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty (traduction revue et augmentée de "Théorie des Possibilités")*. Plenum Press, New York, 1988. (Quoted on page 48.)

[58] Didier Dubois and Henri Prade. The logical view of conditioning and its application to possibility and evidence theories. *International Journal of Approximate Reasoning*, 4(1):23 – 46, 1990. (Quoted on pages 55 and 160.)

[59] Didier Dubois and Henri Prade. Possibility Theory as a basis for qualitative decision theory. In *IJCAI*, pages 1924–1930. Morgan Kaufmann, 1995. (Quoted on page 60.)

[60] Didier Dubois and Henri Prade. Quantitative Possibility Theory and its probabilistic connections. In Przemyslaw Grzegorzewski, Olgierd Hryniewicz, and Maria Angeles-Gil, editors, *Soft Methods in Probability, Statistics and Data Analysis*, Advances in Soft Computing, pages 3–26. Physica Verlag, Heidelberg - Germany, 2002. bb. (Quoted on page 149.)

[61] Didier Dubois and Henri Prade. *Formal Representations of Uncertainty*, pages 85–156. ISTE, 2010. (Quoted on page 18.)

[62] Didier Dubois, Henri Prade, and Régis Sabbadin. Decision-theoretic foundations of qualitative possibility theory. *European Journal of Operational Research*, 128(3):459–478, 2001. (Quoted on pages 20, 22, and 60.)

[63] Didier Dubois, Henri Prade, and Sandra Sandri. On possibility/probability transformations. In *Proceedings of Fourth IFSA Conference*, pages 103–112. Kluwer Academic Publ, 1993. (Quoted on pages 87, 147, and 148.)

[64] Didier Dubois, Henri Prade, and Philippe Smets. Representing partial ignorance. *IEEE Trans. on Systems, Man and Cybernetics*, 26:361–377, 1996. (Quoted on page 15.)

[65] M. R. Endsley. Towards a theory of situation awareness in dynamic systems. *Human Factors*, 1(37):32–64, 1995. (Quoted on page 119.)

[66] Hélène Fargier, Nahla Ben Amor, and Wided Guezguez. On the complexity of decision making in possibilistic decision trees. *CoRR*, abs/1202.3718, 2012. (Quoted on page 21.)

[67] M. Feary. Formal identification of automation surprise vulnerabilities in design. 2005. (Quoted on pages 121 and 122.)

[68] Pascale Fonk. *Réseaux d'inférence pour le raisonnement possibiliste*. Ph.d. thesis, Université de Liège, Faculté des Sciences, 1994. (Quoted on page 100.)

[69] Laurent Garcia and Régis Sabbadin. Complexity results and algorithms for possibilistic influence diagrams. *Artificial Intelligence*, 172(8-9):1018 – 1044, 2008. (Quoted on page 21.)

[70] Hector Geffner and Blai Bonet. Solving large POMDPs using real time dynamic programming. In *In Proc. AAAI Fall Symp. on POMDPs*, 1998. (Quoted on pages 10 and 47.)

[71] Phan Hong Giang and Prakash P. Shenoy. A comparison of axiomatic approaches to qualitative decision making using possibility theory. In Jack S. Breese and Daphne Koller, editors, *UAI*, pages 162–170. Morgan Kaufmann, 2001. (Quoted on page 163.)

[72] Milos Hauskrecht. Incremental methods for computing bounds in partially observable Markov decision processes. In Benjamin Kuipers and Bonnie L. Webber, editors, *AAAI/IAAI*, pages 734–739. AAAI Press / The MIT Press, 1997. (Quoted on page 45.)

[73] Milos Hauskrecht. Value-function approximations for partially observable Markov decision processes. *CoRR*, abs/1106.0234, 2011. (Quoted on page 44.)

[74] Ellen Hisdal. Conditional possibilities independence and noninteraction. *Fuzzy Sets and Systems*, 1(4):283 – 297, 1978. (Quoted on page 53.)

[75] Jesse Hoey, Robert St-Aubin, Alan Hu, and Craig Boutilier. SPUDD: Stochastic planning using decision diagrams. In *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 279–288. Morgan Kaufmann, 1999. (Quoted on pages 23, 92, 93, 94, 98, 99, and 161.)

[76] Hideaki Itoh and Kiyohiko Nakamura. Partially observable Markov decision processes with imprecise parameters. *Artificial Intelligence*, 171(8-9):453 – 490, 2007. (Quoted on pages 13 and 17.)

[77] M.R. James, S. Singh, and M.L. Littman. Planning with predictive state representations. In *Machine Learning and Applications, 2004. Proceedings. 2004 International Conference on*, pages 304–311, Dec 2004. (Quoted on page 10.)

[78] A. Joshi, S. P Miller, and M.P. Heimdahl. Mode confusion analysis of a flight guidance system using formal methods. In *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*, volume 1, pages 2–D. IEEE, 2003. (Quoted on page 119.)

[79] Thomas Keller and Patrick Eyerich. PROST: probabilistic planning based on UCT. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012, Atibaia, São Paulo, Brazil, June 25-19, 2012*, 2012. (Quoted on pages 24, 105, 110, 157, and 162.)

[80] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. (Quoted on pages 48, 105, and 110.)

[81] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. (Quoted on page 22.)

[82] Andrey Kolobov, Peng Dai, Mausam Daniel, and S. Weld. Reverse iterative deepening for finite-horizon mdps with large branching factors. In *In ICAPS'12*, 2012. (Quoted on page 110.)

[83] Andrey Kolobov, Mausam, and Daniel S. Weld. LRTDP versus UCT for online probabilistic planning. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012. (Quoted on pages 24, 110, and 162.)

[84] Hanna Kurniawati, David Hsu, and Wee Sun Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008. (Quoted on pages 10, 46, 86, 93, and 161.)

[85] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006. (Quoted on page 85.)

[86] Yann Le Cun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag, 1998. (Quoted on page 12.)

[87] Y. Lecun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. volume 2, 2004. (Quoted on page 11.)

[88] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. (Quoted on pages 11 and 12.)

[89] Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *In Advances In Neural Information Processing Systems 14*, pages 1555–1561. MIT Press, 2001. (Quoted on page 10.)

[90] Michael Lederman Littman. Algorithms for sequential decision making, 1996. (Quoted on page 45.)

[91] William S. Lovejoy. Computationally Feasible Bounds for Partially Observed Markov Decision Processes. *Operations Research*, 39(1), 1991. (Quoted on page 47.)

[92] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 541–548, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence. (Quoted on pages 10, 21, 43, and 73.)

[93] Bhaskara Marthi. Robust navigation execution by planning in belief space. In *Robotics: Science and Systems VIII, University of Sydney, Sydney, NSW, Australia, July 9-13, 2012*, 2012. (Quoted on page 9.)

[94] David A. McAllester and Satinder P. Singh. Approximate planning for factored pomdps using belief state simplification. *CoRR*, abs/1301.6719, 2013. (Quoted on page 25.)

[95] Martin Mundhenk. The complexity of planning with partially-observable Markov decision processes. Technical report, Hanover, NH, USA, 2000. (Quoted on page 10.)

[96] Yaodong Ni and Zhi-Qiang Liu. Policy iteration for bounded-parameter POMDPs. *Soft Computing*, pages 1–12, 2012. (Quoted on pages 13 and 17.)

[97] Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5):780–798, September-October 2005. (Quoted on page 14.)

[98] M. Oaksford and N. Chater. *Bayesian Rationality.* Oxford University Press, Oxford, 2007. (Quoted on page 120.)

[99] D. P. O'Brien. Human reasoning includes a mental logic. *Behavioral and brain sciences*, 32(1):96–97, 1995. (Quoted on page 119.)

[100] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *Int. J. Rob. Res.*, 29(8):1053–1068, July 2010. (Quoted on pages 9, 23, 73, 81, 86, 91, 93, 104, 105, 109, 152, 160, and 161.)

[101] Takayuki Osogami. Robust partially observable Markov decision process. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 106–115, 2015. (Quoted on pages 14 and 17.)

[102] C. M. Ozveren and A. S. Willsky. Observability of discrete event dynamic systems. *IEEE transactions on automatic control*, 34(7):797–806, 1990. (Quoted on page 120.)

[103] Christos Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Math. Oper. Res.*, 12(3):441–450, August 1987. (Quoted on pages 10, 21, 43, 73, and 145.)

[104] Judea Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, pages 329–334, August 1985. (Quoted on page 27.)

[105] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. (Quoted on pages 27 and 100.)

[106] Patrice Perny, Olivier Spanjaard, and Paul Weng. Algebraic Markov Decision Processes. In *19th International Joint Conference on Artificial Intelligence*, pages 1372–1377, 2005. (Quoted on page 65.)

[107] Jean-Eric Pin. Tropical Semirings. In J. Gunawardena, editor, *Idempotency (Bristol, 1994)*, Publ. Newton Inst. 11, pages 50–69. Cambridge Univ. Press, Cambridge, 1998. (Quoted on page 21.)

[108] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025 – 1032, August 2003. (Quoted on pages 10 and 46.)

[109] Joelle Pineau and Geoffrey J. Gordon. POMDP planning for robust robot control. In Sebastian Thrun, Rodney A. Brooks, and Hugh F. Durrant-Whyte, editors, *ISRR*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 69–82. Springer, 2005. (Quoted on page 9.)

[110] Sergio Pizziol. *Conflict prediction in human-machine systems.* Thèse de doctorat, Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), Onera-DCSD, Systèmes-EdSys Doctoral School, Toulouse, France, November 2013. (Quoted on pages 24 and 119.)

[111] Sergio Pizziol, Catherine Tessier, and Frédéric Dehais. Petri net-based modelling of human–automation conflicts in aviation. *Ergonomics*, (ahead-of-print):1–13, 2014. (Quoted on pages 123, 137, and 141.)

[112] Pascal Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes.* PhD thesis, Department of Computer Science, University of Toronto, 2005. (Quoted on page 117.)

[113] Pascal Poupart, Kee-Eung Kim, and Dongho Kim. Closing the gap: Improved bounds on optimal POMDP solutions. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *ICAPS*. AAAI, 2011. (Quoted on page 44.)

[114] Cédric Pralet, Thomas Schiex, and Gérard Verfaillie. *Sequential Decision-Making Problems - Representation and Solution.* Wiley, 2009. (Quoted on page 90.)

[115] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. (Quoted on pages 28, 30, 35, 83, and 145.)

[116] Julia Radoszycki, Nathalie Peyrard, and Régis Sabbadin. Finding good stochastic factored policies for factored markov decision processes. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 1083–1084, 2014. (Quoted on page 92.)

[117] Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *J. Mach. Learn. Res.*, 12:1729–1770, July 2011. (Quoted on page 14.)

[118] J. Rushby. Using model checking to help discover mode confusions and other automation surprise. In *Reliability Engineering and System Safety*, volume 75, pages 167–177, 2002. (Quoted on pages 119 and 123.)

[119] J. Rushby, J. Crow, and E. Palmer. An automated method to detect potential mode confusions. 1999. Presentation slides. (Quoted on page 119.)

[120] Régis Sabbadin. *Une Approche Ordinale de la Decision dans l'Incertain: Axiomatisation, Representation Logique et Application à la Décision Séquentielle.* Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 1998. (Quoted on page 63.)

[121] Régis Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI'99, pages 567–574, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. (Quoted on pages 21, 22, 23, 62, 65, 66, 68, 73, 78, 83, 84, 87, 159, and 160.)

[122] Régis Sabbadin. Empirical comparison of probabilistic and possibilistic Markov decision processes algorithms. In Werner Horn, editor, *ECAI*, pages 586–590. IOS Press, 2000. (Quoted on pages 22, 62, 78, 80, 91, and 107.)

[123] Régis Sabbadin. Possibilistic Markov decision processes. *Engineering Applications of Artificial Intelligence*, 14(3):287 – 300, 2001. Soft Computing for Planning and Scheduling. (Quoted on pages 22, 62, 78, 83, and 84.)

[124] Régis Sabbadin. Towards possibilistic reinforcement learning algorithms. In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems, Melbourne, Australia, December 2-5, 2001*, pages 404–407, 2001. (Quoted on page 163.)

[125] Régis Sabbadin, Hélène Fargier, and Jérôme Lang. Towards qualitative approaches to multi-stage decision making. *Int. J. Approx. Reasoning*, 19(3-4):441–471, 1998. (Quoted on pages 21 and 60.)

[126] Scott Sanner. Relational dynamic influence diagram language (RDDL): Language description. (Quoted on pages 89 and 110.)

[127] Scott Sanner. Probabilistic track of the 2011 International Planning Competition. http://users.cecs.anu.edu.au/∼ssanner/IPPC_2011, 2011. (Quoted on pages 29, 107, 157, and 163.)

[128] Pierre Sermanet, Raia Hadsell, Marco Scoffier, Matthew Grimes, Jan Ben, Ayse Erkan, Chris Crudele, Urs Miller, and Yann LeCun. A multirange architecture for collision-free off-road robot navigation. *J. Field Robotics*, 26(1):52–87, 2009. (Quoted on page 12.)

[129] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976. (Quoted on page 15.)

[130] Guy Shani, Ronen I. Brafman, and Solomon E. Shimony. Forward Search Value Iteration for POMDPs. In *International Joint Conference on Artificial Intelligence*, 2007. (Quoted on page 46.)

[131] Guy Shani, Pascal Poupart, Ronen I. Brafman, and Solomon Eyal Shimony. Efficient ADD operations for point-based algorithms. In *ICAPS*, pages 330–337, 2008. (Quoted on pages 94 and 117.)

[132] David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2164–2172. Curran Associates, Inc., 2010. (Quoted on pages 10 and 48.)

[133] Hyeong Seop Sim, Kee-Eung Kim, Jin Hyung Kim, Du-Seong Chang, and Myoung-Wan Koo. Symbolic heuristic search value iteration for factored POMDPs. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 1088–1093. AAAI Press, 2008. (Quoted on pages 25, 93, 109, and 161.)

[134] Satinder Singh, Michael L. Littman, Nicholas K. Jong, David Pardoe, and Peter Stone. Learning predictive state representations. In *Proceedings of the Twentieth International Conference on Machine Learning*, August 2003. (Quoted on page 10.)

[135] Richard D. Smallwood and Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon*, volume 21. INFORMS, 1973. (Quoted on pages 9 and 41.)

[136] Trey Smith and Reid Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States, 2004. AUAI Press. (Quoted on pages 10, 46, 99, and 156.)

[137] Matthijs T. J. Spaan and Nikos Vlassis. Perseus: randomized point-based value iteration for POMDPs. Technical Report IAS-UVA-04-02, Informatics Institute, University of Amsterdam, November 2004. (Quoted on page 46.)

[138] Robert St-Aubin, Jesse Hoey, and Craig Boutilier. APRICODD: Approximate policy construction using decision diagrams. In *In Proceedings of Conference on Neural Information Processing Systems*, pages 1089–1095, 2000. (Quoted on pages 93, 99, and 107.)

[139] M. Sugeno. *Theory of fuzzy integrals and its applications.* PhD thesis, Tokyo Institute of Technology, 1974. (Quoted on page 59.)

[140] Florent Teichteil-Königsbuch, Vincent Vidal, and Guillaume Infantes. Extending classical planning heuristics to probabilistic planning with dead-ends. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011. (Quoted on page 163.)

[141] Judea P. Thomas Verma. Influence Diagrams and d-Separation. Technical report, July 1988. (Quoted on pages 24 and 100.)

[142] Felipe W Trevizan, Fabio Gagliardi Cozman, and Leliane Nunes de Barros. Planning under risk and knightian uncertainty. *IJCAI*, 2007:2023–2028, 2007. (Quoted on page 18.)

[143] Tiago S. Veiga, Matthijs T. J. Spaan, and Pedro U. Lima. Point-based POMDP solving with factored value function approximation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2512–2518, 2014. (Quoted on page 25.)

[144] Tom S. Verma and Judea Pearl. Causal networks: Semantics and expressiveness. *CoRR*, abs/1304.2379, 2013. (Quoted on page 100.)

[145] Peter Walley. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning*, 24(2-3):125 – 148, 2000. (Quoted on pages 14 and 15.)

[146] Paul Weng. Qualitative Decision-Making Under Possibilistic Uncertainty: Toward More Discriminating Criteria. In *21st International Conference on Uncertainty in Artificial Intelligence*, volume 21, pages 615–622, 2005. INT LIP6 DECISION. (Quoted on pages 78 and 163.)

[147] Paul Weng. Conditions générales pour l'admissibilité de la programmation dynamique dans la décision séquentielle possibiliste. *Revue d'Intelligence Artificielle*, 21(1):129–143, 2007. NAT LIP6 DECISION. (Quoted on pages 22 and 90.)

[148] Jason D. Williams. Factored partially observable markov decision processes for dialogue management. In *In 4th Workshop on Knowledge and Reasoning in Practical Dialog Systems*, pages 76–82, 2005. (Quoted on page 25.)

[149] Nic Wilson. An order of magnitude calculus. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 548–555, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. (Quoted on page 163.)

[150] Stefan Witwicki, Francisco S. Melo, Jesús Capitán, and Matthijs T. J. Spaan. A flexible approach to modeling unpredictable events in MDPs. In *Proc. of Int. Conf. on Automated Planning and Scheduling*, pages 260–268, 2013. (Quoted on page 100.)

[151] Hôakan L. S. Younes and Michael L. Littman. PPDDL 1.0: An extension to PDDL for expressing planning domains with probabilistic effects. Technical report. (Quoted on page 110.)

[152] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Journal of Information Science*, page 199, 1975. (Quoted on page 53.)

[153] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965. (Quoted on page 48.)

[154] Lotfi A. Zadeh. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. *Soft Comput.*, 2(1):23–25, 1998. (Quoted on page 146.)

Notations

$f : \mathcal{X} \to \mathcal{Y}$	f is a function from $\mathcal{X}$ to $\mathcal{Y}$
$f : x \mapsto y$	the function f maps the element $x \in \mathcal{X}$ to the element $y \in \mathcal{Y}$
$f(., x)$	function $f^x : y \mapsto f(x, y)$ if f is defined on $\mathcal{X} \times \mathcal{Y}$ and $x \in \mathcal{X}$
$\mathbb{1}_A$	indicator function of the set A
$\mathbb{1}_{F(s)}$	function equal to one if the formula $F(s)$ is true, and zero otherwise (Kronecker delta)
$\#A$	size of set A
$\overline{A}$	complementary set of A: if the working set is Ω, $A \cup \overline{A} = \Omega$ and $A \cap \overline{A} = \emptyset$
$\mathbb{N}, \mathbb{N}^*$	set of non negative integers, of positive integers
$\mathbb{R}, \mathbb{R}_+$	set of real numbers, non negative real numbers
$\mathbb{R}^d$	real vector space with d dimensions
$\mathcal{P}(\Omega)$	the set of all subsets of the set Ω
$parents(S)$	parents and descendants of a variable (node) S in a Bayesian Network
$descend(S)$	
$\mathbb{P}, \mathbb{E}$	probability and expectation
$\mathcal{F}_B(\mathcal{S}, \mathbb{R})$	the set of the bounded functions from the set $\mathcal{S}$ to the set $\mathbb{R}$
$X \perp\!\!\!\perp Y \vert Z$	X is independent from Y conditional on Z
$S \sim b$	b is the probability distribution of the random variable S: $\forall s$, $\mathbb{P}(S = s) = b(s)$
$\overline{b} \propto b$	$\overline{b}$ is the probabilistic normalization of b: $\overline{b}(s) = b(s) / \sum_{\tilde{s}} b(\tilde{s})$
$b^\pi \propto^\pi b$	b^π is the possibilistic nomalization of b: $b^\pi(s) = 1$ if $b(s) = \max_{s'} b(s')$, $b(s)$ otherwise
$\langle x, y \rangle_{\mathbb{R}^\mathcal{S}}$	the scalar product of two vectors x and y from $\mathbb{R}^\mathcal{S}$
$\bigcirc$	function composition operator
$\mathbb{P}^\mathcal{S}$	set of the probability distributions over $\mathcal{S}$
$\Pi^\mathcal{S}_\mathcal{L}$	set of the possibilistic distributions over $\mathcal{S}$ when the possibilistic scale is $\mathcal{L}$

Related Research Material

This PhD thesis has led to the development of following materials:

Conference papers

- *Qualitative Possibilistic Mixed-Observable MDPs*, Proceedings of the Twenty-Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI-13) with Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois, (2013) [49]. Presentation: poster and 2-minute speech in English.

- *Structured Possibilistic Planning Using Decision Diagrams*, Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, (AAAI-14) with Florent Teichteil-Königsbuch, Jean-Loup Farges and Didier Dubois, (2014) [50]. Presentation: poster and 20-minute speech in English.

- *Planning in Partially Observable Domains with Fuzzy Epistemic States and Probabilistic Dynamics*, Proceedings of the Ninth International Conference on Scalable Uncertainty Management (SUM-15) with Didier Dubois, Jean-Loup Farges and Florent Teichteil-Königsbuch, (2015) [48]. Presentation: 30-minute speech in English.

- *Processus Décisionnels de Markov Possibilistes à Observabilité Mixte*, Eighth Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA-13), with Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois, (2013). Presentation: poster and 30-minute speech in French.

- *Planification dans des domaines partiellement observables avec des états épistémiques flous et une dynamique probabiliste.*, Tenth Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA-15), with Didier Dubois, Jean-Loup Farges and Florent Teichteil-Königsbuch, (2015). Presentation: 30-minute speech in French.

Workshop paper

- *Structured Possibilistic Planning Using Decision Diagrams*, Workshop "Models and Paradigms for Planning under Uncertainty: a Broad Perspective" of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS-14). Presentation by Florent Teichteil-Königsbuch.

Journal papers

- *A Possibilistic Estimation of Human Attentional Errors*, submitted to the IEEE Transactions on Fuzzy Systems (TFS), with Sergio Pizziol, Catherine Tessier, Jean-Loup Farges, Didier Dubois and Frédéric Dehais.

- *Processus Décisionnels de Markov Possibilistes à Observabilité Mixte*, Special edition of the Revue d'Intelligence Artificielle (RIA. Volume 29 – n6/2015), with Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois.

Oral presentations

- "Possibilistic MDP and POMDP". Seminar in English of the SequeL team at INRIA Lille, France (2012).
- "Modèles possibilistes pour la décision séquentielle dans l'incertain". Seminar in French of the MAD team at GREYC Caen, France (2013).
- EDSYS days, doctoral school conference: 30-minute speech in French at ISAE-SUPAERO, Toulouse France (2014).
- PhD student days (JDD), 20-minute speech in French at Onera – The French Aerospace Lab, Toulouse France (2014 and 2015).
- Onera-DLR Aerospace Symposium, (ODAS 2015), 30-minute speech in English, at Onera – The French Aerospace Lab, Toulouse France (2015).

Code

- Participation in the International Probabilistic Planning Competition 2014 – MDP track, with the algorithm PPUDD.
- Github repository `http://www.github.com/drougui/ppudd`.

Title: Exploiting Imprecise Information Sources for Sequential Decision Making under Uncertainty

Abstract: Partially Observable Markov Decision Processes (POMDPs) define a useful formalism to express probabilistic sequential decision problems under uncertainty. When this model is used for a robotic mission, the *system* is defined as the features of the robot and its environment, needed to express the mission. The system state is not directly seen by the agent (the robot). Solving a POMDP consists thus in computing a strategy which, on average, achieves the mission best *i.e.* a function mapping the information known by the agent to an action. Some practical issues of the POMDP model are first highlighted in the robotic context: it concerns the modeling of the agent ignorance, the imprecision of the observation model and the complexity of solving real world problems. A counterpart of the POMDP model, called π-POMDP, simplifies uncertainty representation with a qualitative evaluation of event plausibilities. It comes from Qualitative Possibility Theory which provides the means to model imprecision and ignorance. After a formal presentation of the POMDP and π-POMDP models, an update of the possibilistic model is proposed. Next, the study of factored π-POMDPs allows to set up an algorithm named PPUDD which uses Algebraic Decision Diagrams to solve large structured planning problems. Strategies computed by PPUDD, which have been tested in the context of the competition IPPC 2014, can be more efficient than those produced by probabilistic solvers when the model is imprecise or for high dimensional problems. We show next that the π-Hidden Markov Processes (π-HMP), *i.e.* π-POMDPs without action, produces useful diagnosis in the context of Human-Machine interactions. Finally, a hybrid POMDP benefiting from the possibilistic and the probabilistic approach is built: the qualitative framework is only used to maintain the agent's knowledge. This leads to a strategy which is pessimistic facing the lack of knowledge, and computable with a solver of fully observable Markov Decision Processes (MDPs). This thesis proposes some ways of using Qualitative Possibility Theory to improve computation time and uncertainty modeling in practice.

Keywords: POMDP, Planning under Uncertainty, Possibility Theory, Autonomous Robotics, Imprecise Knowledge

Titre: Tirer Profit de Sources d'Information Imprécises pour la Décision Séquentielle dans l'Incertain

Résumé: Les Processus Décisionnels de Markov Partiellement Observables (PDMPOs) permettent de modéliser facilement les problèmes probabilistes de décision séquentielle dans l'incertain. Lorsqu'il s'agit d'une mission robotique, les caractéristiques du robot et de son environnement nécessaires à la définition de la mission constituent le *système*. Son état n'est pas directement visible par l'*agent* (le robot). Résoudre un PDMPO revient donc à calculer une stratégie qui remplit la mission au mieux en moyenne, *i.e.* une fonction prescrivant les actions à exécuter selon l'information reçue par l'agent. Ce travail débute par la mise en évidence, dans le contexte robotique, de limites pratiques du modèle PDMPO: elles concernent l'ignorance de l'agent, l'imprécision du modèle d'observation ainsi que la complexité de résolution. Un homologue du modèle PDMPO appelé π-PDMPO, simplifie la représentation de l'incertitude: il vient de la Théorie des Possibilités Qualitatives qui définit la plausibilité des événements de manière qualitative, permettant la modélisation de l'imprécision et de l'ignorance. Une fois les modèles PDMPO et π-PDMPO présentés, une mise à jour du modèle possibiliste est proposée. Ensuite, l'étude des π-PDMPOs factorisés permet de mettre en place un algorithme appelé PPUDD utilisant des Arbres de Décision Algébriques afin de résoudre plus facilement les problèmes structurés. Les stratégies calculées par PPUDD, testées par ailleurs lors de la compétition IPPC 2014, peuvent être plus efficaces que celles des algorithmes probabilistes dans un contexte d'imprécision ou pour certains problèmes à grande dimension. Nous montrons ensuite que les Processus de Markov Cachés possibilistes (π-PMCs), *i.e.* les π-PDMPOs sans les actions, produisent de bons diagnostics dans le contexte de l'intéraction Homme-Machine. Enfin, un PDMPO hybride tirant profit des avantages des modèles probabilistes et possibilistes est présenté: seule la connaissance de l'agent est maintenue sous forme qualitative. Ce modèle mène à une stratégie qui réagit de manière pessimiste au défaut de connaissance, et calculable avec des algorithmes de résolution des Processus Décisionnels de Markov entièrement observables (PDM). Cette thèse propose d'utiliser les possibilités qualitatives dans le but d'obtenir des améliorations en termes de temps de calcul et de modélisation de l'incertitude en pratique.

Mots-clés: PDMPO, Planification dans l'Incertain, Théorie des Possibilités, Robotique Autonome, Connaissance Imprecise

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace

Présentée et soutenue par :
Nicolas DROUGARD

le vendredi 18 décembre 2015

Titre :

Exploiting imprecise information sources in sequential decision making problems under uncertainty

Tirer profit des sources d'information imprécises pour la décision séquentielle d

---

École doctorale et discipline ou spécialité :

EDSYS : Systèmes embarqués

Unité de recherche :

Équipe d'accueil ISAE-ONERA CSDV

Directeur(s) de Thèse :

M. Didier DUBOIS (directeur de thèse)
M. Florent TEICHTEIL-KÖNIGSBUCH (co-directeur de thèse)

Jury :

M. Patrice PERNY - Président, Rapporteur
M. Didier DUBOIS - Directeur de thèse
M. Jean-Loup FARGES Ingénieur de recherche ONERA
M. Hector GEFFNER Professor - Rapporteur
M. Florent TEICHTEIL-KÖNIGSBUCH Research Engineer - Co-directeur de thèse
M. Bruno ZANUTTINI Assistant professor

Ce manuscrit est un résumé en français représentant un peu plus d'un quart du travail original, qui lui est en anglais.

Table des matières

Liste des figures

Introduction

Contexte

L'autonomie décisionnelle d'un robot provient, entre autre, du calcul d'une fonction appelée *stratégie* ou *politique* : celle-ci retourne le symbole de l'action à exécuter en fonction de l'historique des données des capteurs du robot. Les caractéristiques d'intérêt du robot et de son environnement forment un *système*. En général, pour une séquence donnée d'actions exécutées par le robot, l'évolution de ce système n'est pas déterministe. Cependant le comportement du système robotique peut être étudié en effectuant de nombreux tests sur lui *i.e.* des exécutions d'actions, ou en utilisant les connaissances d'un expert de ce système. De même, les données provenant des capteurs, brutes ou traitées, sont généralement des éléments incertains : le comportement de ces données, appelées aussi *observations* du système, dépend des actions du robot et des états successifs du système. Des relations peuvent aussi être obtenues entre les observations, les états du système et les actions à l'aide de tests des capteurs dans de nombreuses situations, de la description technique de ces même capteurs, du traitement des données utilisé, ou de n'importe quelle information experte. Par exemple, dans le cas d'un robot utilisant une vision artificielle, la sortie de l'algorithme de traitement d'image utilisé est considérée comme une observation du système puisque elle est à la fois le résultat d'un traitement des données des capteurs, et l'information sur laquelle se base le modèle de décision : ici, les données sont les images provenant de la camera. Pour une camera donnée, et un algorithme de vision donné, le comportement de l'observation est lié à l'action et à l'état du système lors du processus de prise d'image.

Ainsi, dans le but de rendre un robot autonome pour une *mission* donnée, nous cherchons une stratégie, *i.e.* une fonction précisant les actions à exécuter conditionnellement à la séquence d'observations du système, qui tient compte de l'incertitude à propos de l'évolution du système et de ses observations. Le domaine de recherche associé à ce type de problème, *i.e.* le problème du calcul de stratégies, n'est pas restreint à la robotique et est appelé *décision séquentielle dans l'incertain* : dans le cas général, l'entité qui doit effectuer l'action est appelée *l'agent*. Dans cette thèse, bien que les résultats fournis sont principalement théoriques et assez généraux pour concerner des applications plus variées, le problème du calcul de stratégie est étudié ici dans le contexte de la robotique autonome, et l'agent est la partie décisionnelle du robot. Calculer une stratégie pour une mission robotique donnée nécessite un cadre adapté : le modèle le plus connu décrit le comportement de l'état du système et des observations en utilisant la théorie de probabilités.

Un modèle probabiliste pour le calcul de stratégies

Les Processus Décisionnels de Markov (PDM) expriment aisément les problèmes de décision séquentielle sous incertitude probabiliste [5]. Ils sont adaptés au calcul de stratégies si l'état du système est connu par l'agent à chaque moment de la mission. Dans le contexte robotique, cette hypothèse signifie que dans la mission considérée, le robot a une connaissance parfaite des caractéristiques d'intérêt du problème via ses capteurs. Dans ce modèle, l'état du système

est noté s, et l'ensemble fini de tous les états possibles du système est noté $\mathcal{S}$. L'ensemble fini $\mathcal{A}$ est l'ensemble de toutes les actions disponibles pour l'agent qui sont notées $a \in \mathcal{A}$. Le temps est discrétisé en étapes de décision représentées par les entiers $t \in \mathbb{N}$.

Il est supposé que la dynamique de l'état du système est *markovienne* : à chaque étape de temps t, l'état suivant $s_{t+1} \in \mathcal{S}$, ne dépend que de l'état courant $s_t \in \mathcal{S}$ et de l'action choisie $a_t \in \mathcal{A}$. Cette relation est décrite par la fonction de transition $\mathbf{p}(s_{t+1} \mid s_t, a_t)$, définie comme la distribution de probabilité sur l'état suivant s_{t+1} conditionnellement à l'état courant $s_t \in \mathcal{S}$ lors de l'exécution de l'action $a_t \in \mathcal{A}$.

La mission de l'agent est décrite en termes de *récompenses*. Une fonction de récompense $r : (s, a) \mapsto r(s, a) \in \mathbb{R}$ est définie pour chaque action $a \in \mathcal{A}$ et chaque état du système $s \in \mathcal{S}$. Elle modélise le but de l'agent. Chaque valeur de récompense $r(s, a) \in \mathbb{R}$ est une motivation locale pour l'agent. En effet, plus l'agent récupère de récompenses pendant l'exécution du processus, mieux il a réalisé sa mission : une mission est considérée bien remplie si la séquence d'états du système $s_t \in \mathcal{S}$ rencontrés et d'actions $a_t \in \mathcal{A}$ sélectionnées mènent à des récompenses $r(s_t, a_t)$ grandes. La résolution d'un PDM correspond au calcul d'une stratégie optimale, *i.e.* d'une fonction prescrivant les actions $a \in \mathcal{A}$ qu'il faut exécuter au cours du temps afin de maximiser la moyenne de la somme des récompenses obtenues durant une exécution : cette moyenne est calculée en tenant compte du comportement probabiliste des états du système décrit par les fonctions de transition $\mathbf{p}(s_{t+1} \mid s_t, a_t)$. Par exemple, une bonne stratégie peut être une fonction d définie sur $\mathcal{S}$ et à valeurs dans $\mathcal{A}$, puisque l'on suppose ici que l'agent connaît l'état du système durant le processus.

Il a été montré que de telles stratégies markoviennes sont optimales pour certain critères tels que celui basé sur la somme des récompenses actualisées : en effet, un critère bien connu mesurant les performances d'une stratégie d est l'espérance de la somme actualisée des récompenses :

$$\mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t r(s_t, d_t)\right], \tag{1}$$

où $d_t = d(s_t) \in \mathcal{A}$ et $0 < \gamma < 1$ est un facteur d'actualisation assurant la convergence de la somme.

L'hypothèse que l'agent a une connaissance parfaite de l'état du système est assez forte : en particulier, dans le cas des robots réalisant des taches avec des capteurs conventionnels, ces derniers sont souvent incapable de fournir au robot toutes les caractéristiques d'intérêt pour la mission. Ainsi, un modèle plus flexible a été construit : il tient compte de *l'observation partielle* du système par l'agent.

Les PDM Partiellement Observables (PDMPO) [70] ont une puissance de modélisation plus importante, car ils peuvent représenter des situations dans lesquelles l'agent ne connaît pas directement l'état courant du système : ils modélisent de manière plus fine un agent exécutant des actions sous incertitude dans un environnement partiellement observable.

L'ensemble des états du système $\mathcal{S}$, l'ensemble des actions $\mathcal{A}$, la fonction de transition $\mathbf{p}(s_{t+1} \mid s_t, a_t)$ et la fonction de récompense $r(s, a)$ restent les mêmes que pour la définition des PDM. Dans ce modèle, puisque l'état courant du système $s \in \mathcal{S}$ ne peut pas être considéré comme une information accessible pour l'agent, la connaissance de l'agent à propos de l'état du système provient des observations $o \in \mathcal{O}$, où $\mathcal{O}$ est un ensemble fini. La fonction d'observation $\mathbf{p}(o_{t+1} \mid s_{t+1}, a_t)$ donne pour chaque action $a_t \in \mathcal{A}$ et état atteint $s_{t+1} \in \mathcal{S}$, la probabilité sur les observations possibles $o_{t+1} \in \mathcal{O}$. Enfin, *l'état de croyance initial* $b_0(s)$ définit la distribution de probabilité *a priori* sur l'état du système. Un exemple d'usage des PDMPO est illustré dans la figure 1.

Résoudre un PDMPO consiste à calculer une stratégie qui renvoie une action adéquate à chaque étape du processus, et dépendente des observations reçues et des actions sélectionnées

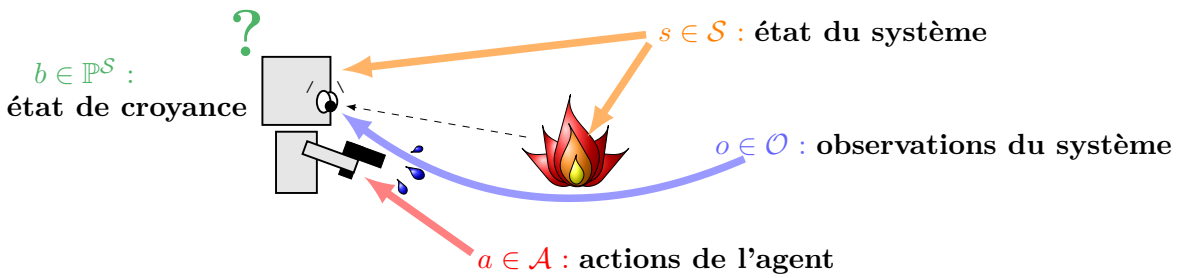


FIGURE 1 – *Utilisation d'un PDMPO pour la modélisation du robot pompier : dans cet exemple, la mission du robot est la prévention des incendies. L'état du système* $s \in \mathcal{S}$ *décrit par exemple la position du robot, l'orientation du jet d'eau, la quantité d'eau utilisée, la position du feu et son niveau sur une échelle de "petit feu" à "feu important", etc. En utilisant une vision artificielle et des capteurs de chaleur, le robot reçoit des* ***observations*** $o \in \mathcal{O}$ *qui sont les données brutes ou traitées provenant des capteurs : la sortie d'un classifieur dont l'entrée est une image de la scène (cf. Figure 3), et qui renvoie une estimation du niveau ou de la position du feu, peut être modélisée par une observation. Finallement, les* ***actions du robot*** $a \in \mathcal{A}$ *sont, par exemple, la mise en marche des moteurs impactant la rotation des roues du robot, le débit de pompage, l'orientation du jet d'eau ou des capteurs, etc. La* ***fonction de récompense*** $r(s,a)$ *décroît avec le niveau de l'incendie. Afin de ne pas gaspiller d'eau, un coût proportionnel à la quantité d'eau est soustrait à cette récompense : puisque une stratégie optimale maximise la moyenne de la somme des récompenses, le but du robot est donc d'éteindre les incendies sans gaspiller de l'eau. Cette moyenne peut-être calculée à l'aide des probabilités décrivant la dynamique stochastique du système. Les actions du robot* $a \in \mathcal{A}$ *ont un effet probabiliste sur le système, décrit par la* ***fonction de transition*** $\boldsymbol{p}(s' \mid s,a)$ *: par exemple, l'activation des roues du moteur modifie la position du robot, et la probabilité sur chacune des positions suivantes possibles, étant donnée la position courante, prend part à la définition du PDMPO. Un autre exemple est l'action modifiant l'orientation du jet d'eau, qui redéfinit la probabilité du nouveau niveau de feu étant donné l'état actuel du système. Les actions du robot* $a \in \mathcal{A}$ *et les états suivants* $s' \in \mathcal{S}$ *peuvent aussi impacter les observations des capteurs : cette influence est définie par la* ***fonction d'observation*** $\boldsymbol{p}(o' \mid s',a)$*, où* $o' \in \mathcal{O}$ *est l'observation : par exemple, l'orientation du capteur de vision peut modifier la probabilité de détection du feu, ou de l'évaluation de son intensité, qui font partie des observations* $o' \in \mathcal{O}$*. Finalement, l'état de croyance est la distribution de probabilité sur l'état courant du système conditionnellement à l'ensemble des observations et des actions successives depuis le début du processus : c'est la meilleure estimation possible puisque le robot n'a accès qu'aux actions et observations lors de l'exécution.*

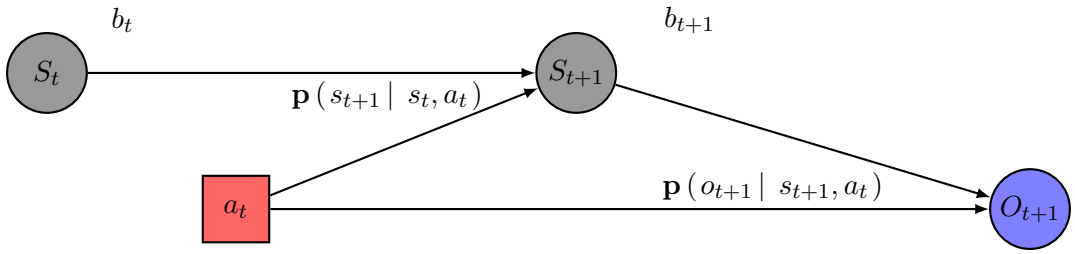


FIGURE 2 – *Réseau Bayésien illustrant la mise à jour de l'état de croyance : les états sont représentés par des ronds gris, l'action est représentée par le losange rouge, et l'observation par le rond bleu. La variable aléatoire S_{t+1} représentant l'état suivant s_{t+1} dépend de l'état courant s_t et de l'action courante a_t. La variable aléatoire O_{t+1} réprésentant l'observation suivante o_{t+1} dépend de l'état suivant s_{t+1} et de l'action courante a_t. L'état de croyance b_t (resp. b_{t+1}) est l'estimation probabiliste de l'état courant (resp. suivant) du système, s_t (resp. s_{t+1}).*

i.e. de toutes les données disponibles pour l'agent : un critère courant pour la stratégie est l'espérance de la somme actualisée des récompenses (1).

La plupart des algorithmes raisonnent sur *l'état de croyance*, défini comme la distribution de probabilité sur l'état du système conditionnellement à toutes les observations du système et les actions choisies par l'agent depuis le début du processus. Cet état de croyance est mis à jour à chaque étape de temps en utilisant la règle de Bayes, l'action courante, et la nouvelle observation. À une étape donnée $t \in \mathbb{N}$, $b_t(s)$ est défini comme la probabilité que le $t^{ième}$ état du système soit $s \in \mathcal{S}$, connaissant les observations et actions précédentes, ainsi que l'état de croyance initial b_0 : c'est une estimation de l'état du système qui utilise uniquement les données disponibles puisque l'état n'est pas directement observable.

Il peut être facilement calculé de manière récursive avec la règle de Bayes : à l'étape de temps t, si l'état de croyance est b_t, l'action choisie $a_t \in \mathcal{A}$ et la nouvelle observation $o_{t+1} \in \mathcal{O}$, l'état de croyance suivant est

$$b_{t+1}(s') \propto \mathbf{p}(o_{t+1} \mid s', a_t) \cdot \sum_{s \in \mathcal{S}} \mathbf{p}(s' \mid s, a_t) \cdot b_t(s). \tag{2}$$

comme illustré par le réseau bayésien de la figure 2.

Puisque les états de croyance successifs sont calculés avec les observations perçues par l'agent, ils sont considérés visible par l'agent. Notons $\mathbb{P}^{\mathcal{S}}$ l'ensemble continu des distributions de probabilité sur $\mathcal{S}$. Une stratégie optimale peut être cherchée parmi les fonctions d définies sur $\mathbb{P}^{\mathcal{S}}$ telles que les $d_t = d(b_t) \in \mathcal{A}$ successifs maximisent l'espérance des récompenses (1) : les décisions de l'agent sont alors basées sur l'état de croyance.

Les PDMPO fournissent un cadre flexible pour la robotique autonome, comme illustré par l'exemple du robot pompier, *cf.* figure 1 : ils permettent de décrire le système regroupant le robot et son environnement, ainsi que la mission du robot. Ils sont assez fréquemment utilisés en robotique [58, 52, 48, 15, 16]. En effet, ils prennent en compte le fait que le robot ne reçoit que les données des capteurs, et doit estimer l'état du système (qui lui est caché) afin de réaliser sa mission. Pour cela, il utilise ces données, appelées alors observations. Cependant, le modèle PDMPO soulève quelques problèmes, en particulier dans le contexte robotique.

PROBLÈMES PRATIQUES DES PDMPO

Complexité

Résoudre un PDMPO *i.e.* calculer une stratégie optimale, est PSPACE-hard en horizon fini [54] et même indécidable en horizon infini [47]. De plus, un espace exponentiel en la description

FIGURE 3 – *Exemple d'une méthode d'observation dans le contexte robotique : le robot, ici un drone, est équippé d'une caméra et utilise un classifieur (classifier) calculé à partir d'une base de données d'images (comme NORB, cf. figure 4). Le classifieur est généré avant la mission (hors-ligne, off-line) avec une base de données d'images (cf. partie droite de l'illustration, dataset), et la sortie du classifieur est utilisée lors de la mission (en ligne, online) comme une observation pour l'agent (cf. partie gauche de l'illustration). Ici, les observations sont donc générées par un algorithme de vision artificielle.*

du problème peut être requis pour la spécification explicite d'une telle stratégie. Le travail [49] est un bon résumé des l'analyses de complexité des PDMPO.

Cette forte complexité est très bien connue des utilisateurs des PDMPO : l'optimalité ne peut être atteinte que pour des petits problèmes, ou bien des problèmes très structurés. Les approches classiques essaient de résoudre ce problème en utilisant la programmation dynamique et des techniques de programmation linéaire [14]. Cependant, pour des problèmes non triviaux, seules des solutions approchées peuvent être calculées, et donc la stratégie n'a pas de garantie d'optimalité. Par exemple, les approches populaire telles que les méthodes basées sur les points, [57, 43, 71], celles basées sur des grilles [34, 13, 7] ou bien les approches de Monte Carlo [68], utilisent des calculs approchés.

Imprécision des Paramètres et Vision Artificielle

Considérons maintenant des robots utilisant la perception visuelle, et dont les observations proviennent d'algorithmes de vision basés sur de l'apprentissage statistique. (*cf.* figure 3). Dans cette situation, le robot utilise un *classifieur* pour reconnaître les objets dans les images : le classifieur est censé renvoyer le nom de l'objet qui se trouve dans l'image, et fait quelquefois des erreurs avec une faible probabilité. (*cf.* matrice de confusion de la figure 5).

Le classifieur est calculé en utilisant une *base de données d'entraînement* (comme *NORB*, *cf.* figure 4, mis en ligne par les auteurs à l'adresse `http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/`).

Les figures (4) et (5) illustrent l'exemple d'un classifieur calculé pour une mission dronique dans laquelle les caractéristiques d'intérêt (les état du système) sont liées à la présence (ou l'abscence) d'animaux (*animals*), voitures (*cars*), humains (*humans*), avions (*planes*) ou camions (*trucks*) : le problème statistique du calcul d'un classifieur permettant de reconnaitre de tels objets dans les images est appelé *classification multi-classes*.

Puisque le classifieur est appris à partir d'une base de données d'images, son comportement, et donc ses performances, (*i.e.* sa fréquence de bonne prédiction) est inévitablement dépendant de la base de données. Cela pose un pose un problème si la variabilité de la base de donnée est trop faible : dans ce cas, le comportement probabiliste du classifieur sera dépendant de ces images en particulier et le système robotique aura des mauvaises capacités d'observation lorsque la mission implique des images trop différentes de celles présentes dans la base de données.

Certaines bases de données à grande variabilité existent (par exemple *NORB*, figure 4, bien que la variabilité pourrait être idéalement supérieure) : notons cependant qu'avec ces bases de données, la performance de vision est réduite, ou bien, au moins, de bonne performances sont difficilement atteignables

base de données d'images *NORB* : $(\text{image}_i, \text{étiquette}_i)_{i=1}^{N}$

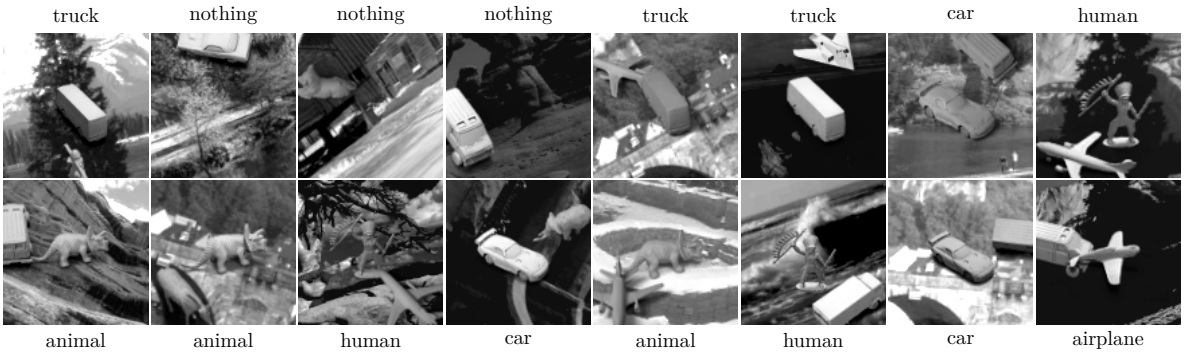


FIGURE 4 – *Exemple de base de données pour la vision artificielle : la base de données d'images étiquetées NORB [45], destinée à l'apprentissage statistique. La taille de NORB est supérieure à* 3.10^5*, et les images de cette base de données représentent des objets de ces* 5 *classes : "animal", "car", "human", "nothing", "plane" et "truck". Chaque élément d'une base de données d'images est composé d'une image (par exemple une image montrant une voiture) et une étiquette correspondant à la classe de l'objet représenté par l'image (dans l'exemple précédent, l'étiquette est "car"). Cette base de données peut être utilisée afin de calculer un classifieur par apprentissage supervisé. Dans le but de discerner les positions des cibles, une image est étiquetée avec le nom de l'objet qui est au centre ("nothing" si il n'y a rien au centre de l'image).*

Une matrice de confusion peut être calculée (*cf.* figure 5) en utilisant une telle base de données étiquetée. Elle doit être différente de celle utilisée pour l'entraînement et est appelée *base de données de test* : la fréquence des observations peut être déduit de cette matrice, en normalisant les lignes ce qui en fait des distributions de probabilité. Une ligne correspond à un objet de la scène et les probabilités de cette ligne sont les probabilités d'observation, *i.e.* chaque valeur de probabilité est la fréquence avec laquelle le classifieur renvoie le nom de l'objet de la colonne correspondante. Ici le classifieur a été calculé à partir d'un Réseau Convolutionnel [46].

Ces probabilités peuvent être utilisées pour définir la fonction d'observation $\mathbf{p}(o' \mid s', a)$ introduite au-dessus. Cette approche soulève encore le problème de la représentativité de la base de données pour la mission voulue. Si la base de donnée de test n'est pas représentative, ces probabilités d'observation risquent de ne pas être fiables, et le PDMPO mal défini : cependant, comme montré par l'équation (2) la mise à jour de l'état de croyance nécessite la connaissance parfaite de la fonction d'observation.

Finalement, si les bases de données considérées sont étiquetées plus précisément, (comme *NORB*, qui inclut des informations telles que la luminosité, ou l'échelle de l'objet), nous pouvons imaginer que les probabilités d'observation calculées (à partir de la matrice de confusion) seraient plus fiables, ou la performance de vision améliorée (puisque la séparation demandée au classifieur est plus simple avec cette précision). Cependant, comme cela implique l'utilisation de plus d'observations ou d'états, le PDMPO est plus dur à résoudre.

En conclusion, l'utilisation du modèle PDMPO fait l'hypothèse que les distributions de probabilité régissant le problème doivent être toutes connues parfaitement : malheureusement ces fréquences ne sont pas connue précisément en pratique. L'imprécision à propos de ces probabilités doit être prise en compte pour rendre le robot autonome en toutes circonstances. En général, le calcul des distributions de probabilité d'un PDMPO nécessite assez de tests pour chaque paire état-action, ce qui est dur à effectuer en pratique.

Quelques variations du cadre PDMPO a été constuit dans le but de prendre en compte l'imprécision sur les distributions de probabilité du modèle, aussi appelé *l'imprécision des paramètres.*

Travaux Tenant Compte de l'Imprécision des Paramètres

Ici, les fonctions de transition et d'observation, *i.e.* $\mathbf{p}(s' \mid s, a)$ et $\mathbf{p}(o' \mid s', a)$, $\forall (s, s', o', a) \in \mathcal{S}^2 \times \mathcal{O} \times \mathcal{A}$, sont appelées *paramètres* du PDMPO, ou encore *paramètres* du modèle. A notre connaissance, le premier modèle construit dans le but de gérer l'imprécision des paramètres est nommé PDMPOPI, pour *PDMPO à Paramètres Imprécis* [37]. Dans ce travail, chacun des paramètres du PDMPO est remplacé par un ensemble de paramètres possibles. Dans ce travail, une *croyance du second ordre* est introduite : elle est définie comme étant la distribution de probabilité sur les paramètres du modèles.

Un autre travail, appelé *PDMPO à Paramètres Bornés* (PDMPOPB) [50], traite aussi de l'imprécision des paramètres : dans ce travail, l'imprécision sur chaque paramètre est décrit à l'aide d'une borne supérieure et inférieure sur les distributions possibles. Aucune croyance du second ordre n'est introduite ici. Cependant, résoudre un PDMPOPB est similaire, dans l'esprit, à la résolution des PDMPOPI [37] : la flexibilité amenée par l'imprécision des paramètres est utilisée pour rendre les calculs les plus faciles possible, et le critère utilisé n'est pas explicite. Le problème majeur de ces approches (PDMPOPI et PDMPOPB) est que l'imprécision des paramètres n'est pas géré dans un but de robustesse, comme une approche pessimiste (pire cas), mais dans un but de simplification.

Un travail plus récent traite le problème de manière pessimiste et est donc appelé *PDMPO robuste* [53]. Inspiré par le travail correspondant dans le cas complètement observable (appelé PDM incertain [51]), ce travail utilise le critère dit *maximin*, ou du *pire cas*, qui vient de la théorie des jeux : dans ce cadre, une stratégie optimale maximise le plus petit critère parmi ceux induits par chaque paramètre possible. Si l'imprécision des paramètres n'est pas stationnaire, *i.e.* peut changer à chaque étape de temps, la stratégie optimale associée (au sens du maximin) peut être calculée en utilisant la *Programmation Dynamique* [4]. Cependant, lorsque l'imprécision des paramètres est stationnaire, les choses se compliquent : les calculs proposés mènent à une approximation de la stratégie optimale, puisque le critère utilisé est une borne inférieure du critère désiré. Pourtant, une hypothèse stationnaire pour l'imprécision des paramètres semble mieux adaptée lorsque le PDMPO est stationnaire.

Bien que l'utilisation d'ensembles de distributions de probabilité rend le modèle plus adapté à la réalité du problème (les paramètres sont imprécis en pratique), les prendre en compte augmente violemment la complexité du calcul d'une politique optimale (par exemple, lors de l'utilisation du critère maximin). Comme expliqué précédemment, résoudre un PDMPO est déjà une tâche très ardue, donc l'utilisation d'un cadre menant à des calculs plus complexes ne semble pas être une approche satisfaisante. En effet, modéliser le problème de manière

animal	human	plane	truck	car	nothing		
3688	575	256	48	144	149	animal	75.885%
97	4180	81	20	225	257	human	86.008%
292	136	3906	237	202	87	plane	80.370%
95	1	44	4073	514	133	truck	83.807%
129	3	130	1283	3283	32	car	67.551%
154	283	36	63	61	4263	nothing	87.716%

FIGURE 5 – *Exemple d'une matrice de confusion en classification multi-classes : cette matrice est calculée avec une base de données de test, différente de la base de données d'apprentissage. Chaque ligne ne considère que les images d'un certain objet, et les nombres représentent les réponses du classifieur : par exemple,* 3688 *images d'animaux sont bien reconnues, mais* 575 *sont confondues avec un humain. La proportion de réponses correctes pour cet objet est* 80.223%*. L'environnement Torch7 [18] a été utilisé pour obtenir un classifieur, et pour calculer cette matrice à partir du classifieur et de la base de donnée de test.*

trop fine mène à l'utilisation de nombreuses approximations dans les calculs en pratique, sans réel contrôle ou estimation de ces approximations comme dans le cas des PDMPOPI et des PDMPOPB. Il est peut-être plus judicieux de commencer avec un modèle plus simple qui peut être résolu plus facilement en pratique.

Un autre problème pratique du modèle PDMPO peut être mentionné : il concerne la définition de l'état de croyance durant les premières étapes du processus, et plus généralement, la manière avec laquelle la connaissance de l'agent est représentée.

Modéliser l'Ignorance de l'Agent

L'état de croyance initial b_0, ou distribution de probabilité *a priori* sur l'état du système, prend part dans la définition du PDMPO. Étant donné un état du système $s \in \mathcal{S}$, $b_0(s)$ est la fréquence de l'évènement "l'état initial est s". Cette quantité peut être dure à calculer rigoureusement, surtout lorsque le nombre d'expériences passées est limité : cette raison a déjà été invoquée au-dessus, menant à l'imprécision des fonctions de transition et d'observation.

Considérons l'exemple d'un robot qui est pour la première fois dans une salle dont la position de la sortie est inconnue (état de croyance initial) et qui doit trouver la sortie et l'atteindre. En pratique, aucune expérience ne peut être répétée dans le but d'extraire une fréquence de position pour cette sortie. Dans ce genre de situation, l'incertitude n'est pas due à un événement aléatoire, mais à un manque de connaissance : aucun état de croyance initial fréquentiste ne peut être utilisé pour définir le modèle.

L'agent peut aussi croire fortement que la sortie est positionnée dans un mur, comme dans la plupart des salles, mais il attribue quand-même une très petite probabilité p_ϵ au fait que la sortie peut être un escalier au plein milieu de la salle. Même s'il est très peu probable que ce soit le cas, cette seconde option doit être prise en compte dans l'état de croyance, sinon la règle de Bayes (*cf.* équation 2) ne peut pas le mettre à jour correctement si la sortie est vraiment au centre de la pièce. Quantifier p_ϵ sans expérience passée n'est pas une chose facile du tout, et ne se repose sur aucune justification rationnelle, mais peut impacter fortement la stratégie de l'agent.

L'état initial du système peut être délibérément défini comme étant inconnu, avec strictement aucune information probabiliste : considérons une mission robotique pour laquelle une partie de l'état du système, décrivant un fait que le robot est censé découvrir par lui-même, est initialement complètement inconnu. Dans un contexte d'exploration robotique, la position ou la nature d'une cible, ou encore la position initiale du robot, peut être défini comme étant absent de la connaissance de l'agent. Les approches classiques initialisent l'état de croyance par une distribution de probabilité uniforme. (*i.e.* sur toutes les positions possibles du robot/cible, ou sur toutes les natures possibles de cible), mais cette réponse provient de l'interprétation subjective des probabilités [20, 31]. En effet, les probabilités sont les mêmes puisqu'aucun événement n'est plus plausible qu'un autre : cela correspond à des paris égaux. Cependant, les mises à jour de l'état de croyance (*cf.* équation 2) mène fatalement à un mélange de probabilités fréquentistes, *i.e.* les fonctions de transition et observation, avec cet état de croyance initial qui est une distribution de probabilité subjective : cela n'a pas toujours de sens, et dans notre cas cette approche est douteuse. Ainsi, l'utilisation des PDMPO dans ces contextes fait face à la difficulté de représenter l'ignorance de l'agent.

Problème Général

Les sections précédentes ont présenté certain problèmes rencontrés en pratique lors de l'utilisation des PDMPO pour calculer des stratégies, en particulier dans le cadre robotique. La très grande complexité du calcul d'une stratégie optimale est un premier problème : les

missions robotiques sont souvent des problèmes à grandes dimensions, dont le calcul d'une stratégie suffisamment proche d'une stratégie optimale est impossible, du fait d'un trop grand temps de calcul ou d'un manque de mémoire vive. Ensuite, il a été mis en évidence que les distributions de probabilité définissant le PDMPO ne sont pas toujours connues précisément : par exemple, la fonction d'observation peut être difficile à définir lorsque les observations proviennent d'un algorithme de vision artificielle complexe. Enfin, le problème de la gestion de la connaissance et de l'ignorance de l'agent a été discuté : il n'y a pas de réponse formelle concernant la manière de représenter le manque de connaissance initial de l'agent à propos du monde dans lequel il évolue. La difficulté vient du fait que les PDMPO classiques (probabilistes) autorisent uniquement l'usage de distribution de probabilité fréquentiste, alors qu'un autre outil mathématique semble nécessaire.

Ces problèmes forment le point de départ de notre travail. En effet, ce dernier consiste à contribuer au problème du calcul d'une stratégie pour des domaines partiellement observables. Les stratégies calculées doivent permettre au robot de remplir sa mission aussi bien que possible, dès la première exécution *i.e.* le calcul de stratégie est opéré avant toute réelle exécution de la mission.

Le chalenge général guidant ce travail est de procéder aux calculs de stratégies en utilisant seulement les données et les connaissances vraiment disponibles en pratique, au lieu d'utiliser les PDMPO classiques, aux calculs très complexes et aux paramètres difficiles à définir. En d'autre mots, cela revient à prêter attention aux problèmes soulevés au-dessus : la complexité de calcul, l'imprécision du modèle, et la gestion de la méconnaissance de l'agent. Comme expliqué par la suite, la théorie des possibilités qualitatives [28] semble pouvoir répondre aux problèmes soulevés.

Une Théorie Qualitative

Cette théorie est généralement introduite en définissant une échelle qualitative $\mathcal{L}$, qui peut être définie par $\left\{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\right\}$, avec $k > 1$, ou tout autre ensemble fini totalement ordonné. Les valeurs de cette échelle ne sont pas importantes car elle servent seulement à matérialiser un ordre : nous utilisons donc le terme de *degré de possibilité* afin d'expliciter cette remarque. Le plus petit élément de l'échelle qualitative est noté 0, et le plus grand, 1. La section suivante clarifie pourquoi l'utilisation de ce cadre qualitatif est bénéfique en matière de complexité et de modélisation.

Notons tout d'abord les similarités entre la théorie des probabilités et des possibilités : une distribution de possibilité sur $\mathcal{S}$ est une fonction $\pi : \mathcal{S} \to \mathcal{L}$ telle que $\max_{s \in \mathcal{S}} \pi(s) = 1$. L'opérateur de marginalisation est l'opérateur maximum (max) : étant donné une distribution de possibilité jointe $\pi : \mathcal{S} \times \mathcal{O} \to \mathcal{L}$, la distribution de possibilité marginale sur $\mathcal{S}$ est $\pi(s) = \max_{o \in \mathcal{O}} \pi(s, o)$. De plus, l'opérateur minimum (min) est utilisé pour calculer une distribution de possibilité jointe $\pi(s, o)$, $\forall (s, o) \in \mathcal{S} \times \mathcal{O}$ à partir d'une distribution marginale $\pi(s)$, $\forall s \in \mathcal{S}$ et d'une distribution conditionnelle $\pi(o \mid s)$, $\forall (o, s) \in \mathcal{S} \times \mathcal{O} : \pi(s, o) = \min\{\pi(s), \pi(o \mid s)\}$. Ainsi, il est possible d'apprivoiser la théorie des possibilités en remplaçant l'opérateur + de la théorie des probabilités par l'opérateur max, et l'opérateur $\times$ par min.

PDMPO Qualitatifs Possibilistes

Une alternative possibiliste qualitative du modèle PDMPO a été proposée dans [62] : ce modèle s'appelle PDMPO qualitatif possibiliste, et est noté π-PDMPO. Un π-PDMPO est simplement un PDMPO avec des distributions possibilistes qualitatives comme paramètres, au lieu de distributions de probabilité. Comme les π-PDMPO sont qualitatifs, l'homogue de la fonction de récompense, appelée *fonction de préférence*, est aussi qualitative : en effet, la

fonction de préférence est à valeurs dans l'échelle possibiliste qualitative finie $\mathcal{L}$, et donc n'est pas additive.

L'une des propriétés les plus intéressantes des π-PDMPO est la simplification du calcul de la stratégie. En effet, les algorithmes proposés pour résoudre les PDMPO probabilistes sont souvent basés sur l'ensemble des états de croyance appelé *espace des croyances*. L'espace des croyances est infini dans le cas général : chaque étape de temps mène à un nombre fini d'états de croyance suivants, donc cet espace est dénombrable. Dans le but d'obtenir des propriétés utiles et des moyens de calculer une stratégie, l'ensemble de toutes les distributions de probabilité sur l'espace d'état $\mathcal{S}$ est souvent considéré, *i.e.* le simplex continu $\mathbb{P}^{\mathcal{S}} = \{\, \mathbf{p} : \mathcal{S} \to [0,1] \mid \sum_{s\in\mathcal{S}} \mathbf{p}(s) = 1, \text{ and } \mathbf{p}(s) \geqslant 0, \forall s \in \mathcal{S} \,\}$. La taille infinie de l'espace des croyances explique en partie pourquoi les PDMPO probabilistes sont vraiment difficiles à résoudre. Au contraire, les π-PDMPO ont un espace de croyances fini. En effet, le nombre de distributions de possibilité qualitatives sur les états du système $\mathcal{S}$ est inférieur à $\#\mathcal{L}^{\#\mathcal{S}}$, puisque l'échelle qualitative $\mathcal{L}$ est finie. La version complètement observable d'un π-PDMPO est appelé π-PDM : comme expliqué dans la section I.4 du chapitre I, tout π-PDMPO se réduit à un π-PDM dont l'espace d'état est l'ensemble des états de croyance possibilistes qualitatifs, et dont la taille est exponentielle en fonction du nombre d'états du système. Dans les travaux [32, 33, 62], il est montré que la complexité d'un π-PDM est plus faible que la complexité d'un PDM probabiliste, qui est polynomiale [54] : la complexité d'un π-PDMPO est au pire exponentiel en la description du problème, tandis qu'un PDMPO probabiliste peut être indécidable [47].

En plus de la simplification des calculs, le π-PDMPO peut être vraiment intéressant pour nos problèmes de modélisation. En effet, dans le cas d'un robot utilisant un algorithme de vision artificielle (*cf.* figure 3), nous avons précédemment mis en évidence la difficulté à définir rigoureusement les fonctions d'observation probabilistes : les probabilités des réponses des algorithmes de vision dans le contexte d'une mission robotique sont mal connues et difficiles à définir en pratique. Il est plus facile de trouver des estimations qualitatives des performances de reconnaissance de l'algorithme : le modèle π-PDMPO ne requiert que des données qualitatives, donc il permet de construire un modèle sans l'utilisation d'informations autres que de celles vraiment disponibles. Par exemple, la matrice de confusion de la figure 5 peut mener à une fonction d'observation qualitative possibiliste tenant compte uniquement de la manière dont les fréquences de réponses sont classées : en présence d'un humain (*i.e.* deuxième ligne), la réponse la plus fréquente est "*human*", la deuxième réponse est "*nothing*", la troisième est "*car*", etc. Donc, la distribution de possibilité correspondante est telle que, conditionnellement à la présence d'un humain, le degré de possibilité de la réponse "*human*" est plus grand que le degré de possibilité de la réponse "*nothing*", qui est plus grand que le degré de possibilité de la réponse "*car*", etc. Au lieu d'attribuer des fréquences qui ne sont pas vraiment fiable en pratique, le modèle possibiliste qualitatif exprime naturellement ces imprécisions relatives au problème.

Enfin, notons que la distribution de possibilité constante, dont les degrés sont tous égaux à 1 (élément maximal de $\mathcal{L}$), représente l'ignorance totale : cette distribution peut être utilisée pour définir l'état de croyance initial lorsqu'il doit représenter un agent qui ignore initialement une situation donnée. Donc, le modèle π-PDMPO permet une modélisation formelle du manque de connaissance de l'agent.

L'utilisation de la théorie des possibilités qualitatives [29] est donc étudié dans ce travail, puisqu'il semble être capable d'à la fois simplifier un PDMPO, et de modéliser l'imprécision des paramètres et l'ignorance associées aux missions robotiques. Ce cadre, en effet, simplifie les calculs, est capable de représenter le problème avec seulement les données disponibles, et modélise le manque de connaissance : ainsi cette théorie offre des solutions aux trois problèmes

mis en évidence précédemment. Cependant, notons qu'un cadre qualitatif ne permet pas de manipuler de l'information fréquentiste.

A notre connaissance, une étude plutôt limitée du modèle π-PDMPO existe dans la littérature jusqu'à aujourd'hui : en fait, le travail [62] semble être le seul, proposant à la fois une définition des π-PDMPO et un exemple jouet pour illustrer ce modèle. La version complètement observable (π-PDM) a généré plus de travaux [64, 63, 77].

Description de l'Étude

Cette thèse contribue à déterminer dans quelle mesure la thérie de possibilités qualitatives peut contribuer à la *planification dans l'incertain dans des domaines partiellement observables*, et plus généralement à la gestion séquentielle de l'incertitude, en matière de simplification des calculs et de modélisation. Elle présente de récentes contributions dans l'utilisation de cette théorie pour la planification dans l'incertain et la représentation des connaissances, avec une utilisation quasi systématique des modèles graphiques [40, 6, 10].

La fin de cette introduction décrit la structure de cette thèse. En effet, chacune des sections suivantes correspond à un chapitre de notre travail, et détaille son contenu.

État de l'Art

Les PDMPO qualitatifs et possibilistes constituent l'objet central de cette thèse : le *premier chapitre* est consacré à ce modèle. Une présentation rapide de la théorie des possibilités qualitatives est suivie de la présentation du modèle entièrement observable (π-PDM), puis du modèle partiellement observable (π-PDMPO). Comme noté précédemment, à notre connaissance, seul un article de dix pages a déjà traité des π-PDMPOs.

Mises à jour naturelles du modèle possibiliste qualitatif

Le *deuxième chapitre* propose quelques extensions au travail [62]. Tout d'abord, est construite une version possibiliste qualitative des PDM à observabilité mixte [52, 2] dans laquelle quelques variables d'état sont complètement observables. Elle est appelée π-PDMOM et généralise à la fois les π-PDM et les π-PDMPO. Cette contribution réduit considérablement la complexité de résolution des π-PDMPO, en manipulant de manière plus fine l'information des environnements dont certaines variables sont complètement observables. Par exemple, le niveau de batterie d'un robot peut être considéré comme une information directement observable pour la prise de décision, menant à des calculs plus simples. Plus généralement, il est très courant de manipuler des variables visibles en robotique [52].

Ensuite, un critère qualitatif est proposé pour pouvoir traiter des missions dont la durée n'est pas connue à l'avance : l'algorithme faisant le calcul de la stratégie optimale associée est alors présenté. Cet algorithme est utilisé pour calculer une stratégie pour une mission de reconnaissance de cible : les résultats expérimentaux comparent les exécutions utilisant cette stratégie à celles utilisant la stratégie d'un algorithme pour PDMPO probabiliste, dans des situations où la dynamique probabiliste des observations est en fait mal définie.

Notons que ces résultats expérimentaux constituent, à notre connaissance, la première utilisation du cadre π-PDMPO. Ils mettent aussi en évidence un comportement intéressant de l'état de croyance possibiliste qualitatif dans certaines situations détaillées ensuite. Les principales contributions de ce chapitre ont été publiées dans [24]. Il est à souligner que les résultats expérimentaux n'auraient pas pu être effectués sans les deux premières contributions, qui permettent de ne pas fixer un horizon arbitraire et d'alléger les calculs.

Cependant, ces contributions ne sont pas suffisantes pour atteindre un temps de calcul compétitif, ou pour pouvoir traiter des problèmes robotiques réels : l'orientation du second chapitre est dictée par cette remarque.

Modèles Factorisés et Algorithmes Symboliques

Les problèmes robotiques traités dans le chapitre précédent sont assez petits pour permettre à l'algorithme proposé de calculer une stratégie en un temps raisonnable. Les contributions du *troisième chapitre* permettent la résolution de problèmes de planification structurés dont l'espace des états est plus grand.

La première partie de ce chapitre propose d'introduire les *π-PDMOM* factorisés : il sont définis par des hypothèses d'indépendance supplémentaires. Les grands problèmes de planification satisfaisant ces hypothèses peuvent être résolus plus facilement : inspiré par l'algorithme pour PDM probabilistes, *SPUDD* [36], nous avons conçu un algorithme nommé *PPUDD* pour résoudre les π-PDMOM factorisés en utilisant des *arbres de décision algébriques* (*ADDs*). L'intuition motivant cette contribution, est que le calcul entre *ADDs* est moins coûteux en temps et en mémoire dans le cadre qualitatif possibiliste que dans le cadre probabiliste : les opérations qualitatives devraient mener à des *ADDs* plus petits puisque la somme et le produit sont remplacés par les opérateurs min et max. Ces derniers produisent des *ADDs* avec potentiellement moins de feuilles, car ils ne créent pas de nouvelles valeurs.

Les hypothèses d'indépendance définissant le modèle π-PDMOM factorisé concernent les variables représentant les états de croyance successifs. De plus, les variables définissant un π-PDMOM sont celles représentant les états du système et les observations. C'est pourquoi la section de ce chapitre qui suit propose des conditions suffisantes sur les variables d'état et d'observation menant aux indépendances désirées entre les variables de croyances. Un exemple robotique est utilisé en guise d'illustration de ces conditions. Puisque les preuves utilisent le concept graphique appelé *d-Séparation* [74], ces conditions mènent aussi à l'indépendance des variables de croyance dans le cadre des PDMOM probabilistes.

Les performances de notre algorithme *PPUDD* sont ensuite comparées à celles des homologues probabilistes, en termes de temps de calcul et avec des critères mesurant la réussite de la mission. Enfin, la dernière partie de ce chapitre décrit les résultats de *PPUDD* lors de la compétition internationale de planification probabiliste [1]. Nous avons participé à la compétition dans le but de tester les performances de *PPUDD* contre celles des algorithmes probabilistes, en termes d'espérances de la somme des récompenses, et sur de nombreux problèmes de planification.

Certaines contributions de ce chapitre ont été publiées dans [25]. Les nombreux problèmes de planification de la compétition mettent aussi en évidence certaines failles de notre algorithme, lorsqu'il est utilisé pour approximer le calcul d'une stratégie pour un problème probabiliste dans le but de bénéficier de calculs qualitatifs qui sont plus simples. De plus, bien que notre algorithme est meilleur que certains algorithmes utilisant des *ADDs* (notamment sont homologue direct, *SPUDD*), les algorithmes de la compétition utilisant des recherches dans l'espace d'état [38, 42] obtiennent de meilleurs résultats. L'approche proposée dans [23] prend en compte ces problèmes.

1. `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`

État de l'Art

I

Le principal sujet de cette thèse est le processus décisionnel markovien partiellement observable (PDMPO), ou plus précisément son homologue possibiliste qualitatif. L'utilisation pratique du modèle probabiliste a été critiquée en introduction, cependant ce modèle résume de manière appropriée les principales caractéristiques d'un système robotique : son homologue possibiliste étant très similaire, son étude semble prometteuse. Tout d'abord, la théorie des possibilités qualitative est présentée dans le but d'introduire les *processus décisionnels markoviens possibilistes qualitatifs* (π-PDM) et enfin les *processus décisionnels markoviens partiellement observables* (π-PDMPO) qui constituent le point de départ de notre travail.

I.1 Théorie des Possibilités Qualitatives

Les *ensembles flous* ont été introduits par Lotfi Zadeh [79], et étudiés par Didier Dubois [19] et Henri Prade : leur contributions ont mené à la fondation de la théorie des possibilités [27].

Comme la théorie des probabilités, cette théorie est basée sur une mesure d'incertitude, appelée *mesure de possibilité*. Contrairement à la mesure de probabilité $\mathbb{P}$ qui est une mesure classique, la mesure de possibilité, notée Π, est une *mesure floue*, ou *capacité*. Pour faire simple, une mesure floue n'est pas supposée additive, mais juste *monotone*, *i.e.* si $A \subset B$ alors la mesure de A est plus petite que la mesure de B. Dans cette thèse, les mesures de possibilité vont concerner uniquement des ensembles finis comme l'ensemble des états $\mathcal{S}$ et l'ensemble des observations $\mathcal{O}$.

Formellement, une mesure de possibilité est définie comme suit :

Définition I.1.1 (*Mesure de Possibilité*)

Une mesure de possibilité sur l'ensemble fini Ω est une fonction $2^\Omega \to [0,1]$ telle que
- $\Pi(\Omega) = 1$ *(normalisation) ;*
- $\forall A, B \subset \Omega$, $\Pi\{A \cup B\} = \max\{\Pi(A), \Pi(B)\}$ *(maxitivité).*

La théorie des probabilité modélise l'incertitude due à la variabilité des événements : en pratique, les probabilités sont les fréquences estimées des événements, définis comme le vrai modèle de variabilité des événements. Une autre interprétation de cette théorie est celle des probabilités subjectives définies par De Finetti [20] : la valeur de probabilité d'un événement est un pari échangeable, relatif aux connaissances d'une personne donnée.

La théorie des possibilités est dédiée à l'incertitude due à un manque de connaissance, ou une imprécision à propos d'un événement. La théorie des possibilités quantitatives est un cas particulier de probabilités imprécises *i.e.* une mesure de possibilité quantitative Π représente un ensemble particulier de mesures de probabilité définies sur Ω.

Contrairement à la théorie des possibilités quantitatives, la théorie des possibilités qualitatives utilise des mesures de possibilité dont les valeurs sont définies sur n'importe quelle

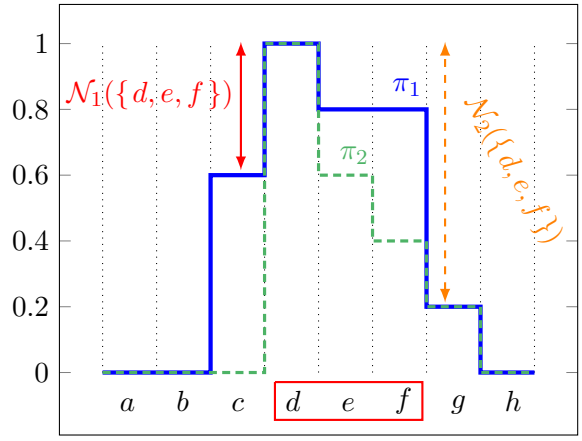


FIGURE I.1 – *Exemple de deux distributions de possibilité sur* $\Omega = \{a, b, c, d, e, f, g, h\}$ *:* π_1 *(ligne bleue continue) et* π_2 *(ligne verte pointillée), avec* π_2 *qui est plus spécifique que* π_1*. La mesure de nécessité* $\mathcal{N}_1$ *associée à* π_1 *est évaluée sur l'événement* $\{d, e, f\} \subset \Omega$ *: le degré de nécessité est égal à* $0.4 = 1 - 0.6$*, comme illustré par les flèches rouges continues. La mesure de nécessité* $\mathcal{N}_2$ *associée à* π_2 *est évaluée sur le même événement : le degré de nécessité est égal à* $0.8 = 1 - 0.2$*, comme illustré par les flèches oranges en pointillées.*

échelle ordonnée. Cette théorie nous permet de raisonner, même lors d'un manque d'informations quantitatives : la seule information donnée par une mesure de possibilité qualitative est l'ordre de plausibilité entre les événements *i.e.* pour $A, B \subseteq \Omega$, l'information "l'événement A est moins plausible que l'événement B", qui s'écrit $\Pi(A) \leqslant \Pi(B)$. Ainsi les mesures de possibilité qualitatives Π sont souvent définies comme des fonctions $2^\Omega \to \mathcal{L}$, où $\mathcal{L}$ est un ensemble fini, appelé *échelle possibiliste*, et possédant un ordre total. Dans ce travail, l'échelle possibiliste est définie par $\mathcal{L} = \left\{0, \frac{1}{k}, \ldots, 1\right\}$ pour simplifier les notations.

Nous pouvons définir la *distribution de possibilité* π comme suit : $\forall \omega \in \Omega$, $\pi(\omega) = \Pi(\{\omega\})$. D'après la définition I.1.1, une mesure de possibilité Π est entièrement définie par la distribution associée π. Pour chaque distribution (ou mesure) de possibilité, une mesure duale, appelée mesure de nécessité, peut être définie : le degré de nécessité d'un événement augmente si le degré de possibilité de l'événement contraire décroît.

Définition I.1.2 (*Mesure de nécessité associée à* Π)

La mesure de nécessité associée à la mesure de possibilité Π *est la mesure floue* $\mathcal{N} : 2^\Omega \to \mathcal{L}$ *telle que* $\forall A \subset \Omega$*,*

$$\mathcal{N}(A) = 1 - \Pi(\overline{A}),$$

où $\overline{A}$ *est l'événement complémentaire de* A *dans* Ω*.*

L'ignorance totale est modélisée par une distribution de possibilité π telle que $\forall \omega \in \Omega$, $\pi(\omega) = 1$ *i.e.* n'importe quel événement élémentaire est possible. Dans ce cas, $\forall A \subseteq \Omega$, $A \neq \Omega$, $\mathcal{N}(A) = 1 - \Pi(\overline{A}) = 1 - \max_{\omega \in \overline{A}} \Pi(\omega) = 0$: aucun événement n'est nécessaire, sauf l'univers entier Ω.

La connaissance parfaite que l'événement élémentaire $\omega_A \in \Omega$ est elle modélisée par une distribution de possibilité π telle que $\pi(\omega_A) = 1$ et $\pi(\omega) = 0$, $\forall \omega \neq \omega_A$. Le degré de nécessité du singleton $\{\omega_A\}$ est aussi égal à 1 : $\mathcal{N}(\{\omega_A\}) = 1 - \Pi(\overline{\{\omega_A\}}) = 1$. L'événement élémentaire ω_A est nécessaire, et tous les autres événements élémentaires ont un degré de nécessité nul : si $\omega_B \neq \omega_A$, $\mathcal{N}(\{\omega_B\}) = 1 - \Pi(\overline{\{\omega_B\}}) = 1 - \pi(\omega_A) = 0$.

Généralement, une distribution de possibilité est plus informative, ou plus *spécifique*, que l'ignorance totale, et moins *spécifique* que la connaissance parfaite :

Définition I.1.3 (*Spécificité*)

Une distribution de possibilité π_2 est plus spécifique que la distribution de possibilité π_1, si $\forall \omega \in \Omega$,

$$\pi_2(\omega) \leqslant \pi_1(\omega).$$

Les notions de nécessité et de spécificité sont illustrées figure I.1.

Les concepts de la théorie des possibilités qualitatives nécessaires à la compréhension de notre travail ont été présentés. Nous présentons maintenant l'homologue du critère basé sur l'espérance dans les modèles probabilistes : les critères qualitatifs utilisés dans les modèles possibilistes qualitatifs.

I.2 Critères Qualitatifs

L'espérance utilisée comme critère dans les PDMPO probabilistes, est simplement l'intégrale de la la fonction de récompense contre la mesure de probabilité. Le concept d'intégrale a été étendue aux mesure floues : quand la mesure est quantitative, l'extension est appelée *intégrale de Choquet* [17]. Dans le cas des mesures qualitatives, l'objet résultant est l'*intégrale de Sugeno* [72]. Nous définissons donc ici l'intégrale de Sugeno :

Définition I.2.1 (*Intégrale de Sugeno*)

L'intégrale de Sugeno d'une fonction $f : \Omega \to \mathcal{L}$ contre la capacité (mesure floue) $\mu : 2^\Omega \to \mathcal{L}$ est

$$\mathbb{S}_\mu[f] = \max_{i=1}^{\#\Omega} \min\{f(\omega_i), \mu(A_i)\} \tag{I.1}$$

$$= \min_{i=1}^{\#\Omega} \max\{f(\omega_i), \mu(A_{i+1})\} \tag{I.2}$$

où $f(\omega_1) \leqslant \ldots \leqslant f(\omega_{\#\Omega})$, $A_i = \{\omega_i, \omega_{i+1}, \ldots, \omega_{\#\Omega}\}$ et $A_{\#\Omega+1} = \emptyset$.

Comme illustré dans la figure I.2, l'intégrale de Sugeno de $f : \Omega \to \mathcal{L}$ contre la mesure floue μ est le plus grand degré $\lambda \in \mathcal{L}$ tel que la mesure μ de $\{\omega \mid f(\omega) \geqslant \lambda\}$ est plus grande ou égale à λ. Par exemple, le h-indice (ou indice de Hirsh) est l'intégrale de Sugeno de la fonction *papier* $\mapsto$ *#citations* contre la mesure de comptage.

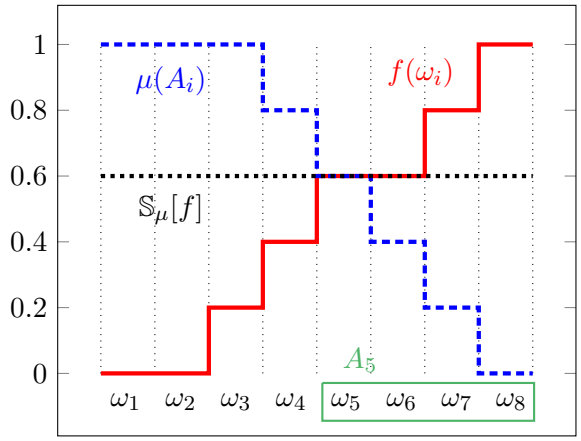


Figure I.2 – *Illustration du résultat de l'intégrale de Sugeno : l'axe des abscisses représente l'ensemble $\Omega = \{\omega_1, \ldots, \omega_{\#\Omega}\}$, où $\forall i \in \{1, \ldots, \#\Omega - 1\}$, $f(\omega_i) \leqslant f(\omega_{i+1})$. L'axe des ordonnées est $\mathcal{L}$. La courbe rouge représente les degrés $f(\omega_i)$, la bleue en pointillées représente les degrés $\mu(A_i)$ avec $A_i = \{\omega_i, \ldots, \omega_{\#\Omega}\}$, et la noire est le résultat de l'intégrale de Sugeno.*

L'intégrale de Sugeno contre une mesure de possibilité et celle contre la nécessité, mènent à deux critères pour la planification. Ces intégrales se réécrivent comme suit :

Théoreme 1 (*L'intégrale de Sugeno contre les mesures de possibilité et de nécessité*)

$$\mathbb{S}_{\Pi}[f] = \max_{i=1}^{\#\Omega} \min\{ f(\omega_i), \pi(\omega_i) \}, \tag{I.3}$$

$$\mathbb{S}_{\mathcal{N}}[f] = \min_{i=1}^{\#\Omega} \max\{ f(\omega_i), 1 - \pi(\omega_i) \}. \tag{I.4}$$

sont les réécritures des intégrales de Sugeno contre les mesures de possibilité et de nécessité.

Les critères possibilistes qualitatifs, *i.e.* les fonctions $\mathcal{A} \to \mathcal{L}$ mesurant la validité des actions étant donné un modèle possibiliste et une fonction de préférence, a été proposé dans [66, 29, 28], basé sur les intégrales de Sugeno (I.3) et (I.4). Rappelons que l'ensemble $\mathcal{S}$ (resp. $\mathcal{A}$) est comme dans l'introduction l'ensemble fini des états du système s (resp. des actions a). La variable représentant l'état du système est $S \in \mathcal{S}$. Soit $(\pi_a)_{a\in\mathcal{A}}$ une famille de distributions de possibilité sur $\mathcal{S}$, *i.e.* $\forall a \in \mathcal{A}$, $\pi_a(s) = \Pi_a(\{ S = s \})$ est le degré de possibilité de la situation $\{ S = s \} \subset \Omega$ lorsque l'action $a \in \mathcal{A}$ est sélectionnée. Soit $\rho : \mathcal{S} \to \mathcal{L}$ la fonction de préférence, définissant le degré de préférence de chaque état du système $s \in \mathcal{S}$.

Définition I.2.2 (*Critère de décision qualitatif*)

Soit π_a la distribution de possibilité décrivant l'incertitude à propos de l'état du système étant donné que l'action $a \in \mathcal{A}$ a été sélectionnée, et $\rho(s)$ la préférence de l'état du système $s \in \mathcal{S}$. En utilisant la formule (I.3) avec $f = \rho(S)$, l'intégrale de Sugeno de la préférence contre la mesure de possibilité Π_a mène à un critère optimiste pour $a \in \mathcal{A}$:

$$\mathbb{S}_{\Pi_a}[\rho(S)] = \max_{s\in\mathcal{S}} \min\{ \rho(s), \pi_a(s) \}. \tag{I.5}$$

De même, en utilisant la formule (I.4) avec $f = \rho(S)$, l'intégrale de Sugeno de la préférence contre la mesure de nécessité associée à Π_a, notée $\mathcal{N}_a$, mène au critère pessimiste pour $a \in \mathcal{A}$:

$$\mathbb{S}_{\mathcal{N}_a}[\rho(S)] = \min_{s\in\mathcal{S}} \max\{ \rho(s), 1 - \pi_a(s) \}. \tag{I.6}$$

Le critère pessimiste (I.6) cherche à éviter les états non désirés, tandis que le critère optimiste (I.5) souhaite rendre possible le fait d'atteindre les états préférés.

L'exemple suivant, illustré en figure I.3, montre bien la différence entre ces deux critères : soit $\mathcal{S} = \{ s_A, s_B, s_C \}$ l'ensemble des états et $\mathcal{A} = \{ a_1, a_2 \}$ l'ensemble des actions. Le modèle de préférence et le modèle d'incertitude sont décrits respectivement par ρ et $(\pi_a)_{a\in\mathcal{A}}$:

- $1 = \rho(s_A) > \rho(s_B) > \rho(s_C) = 0$;
- si l'action a_1 est sélectionnée, $\pi_{a_1}(s_A) = \pi_{a_1}(s_C) = 1$, et $\pi(s_B) = 0$;
- si l'action a_2 est sélectionnée, $\pi_{a_2}(s_A) = \pi(s_C) = 0$, et $\pi(s_B) = 1$, *i.e.* le système est dans l'état s_B de manière déterministe (avec nécessité 1).

Le critère optimiste est maximisé par l'action a_1, puisqu'avec cette action, le meilleur état du système, s_A, est entièrement possible. Cependant, cette action rend le pire état du système, s_C, entièrement possible, donc l'état s_A n'est pas du tout nécessaire : une action plus prudente est a_2, avec laquelle l'état devient s_B avec certitude, mais avec une plus petite préférence. On peut vérifier facilement que l'action a_2 maximise bien le critère pessimiste (I.6) :

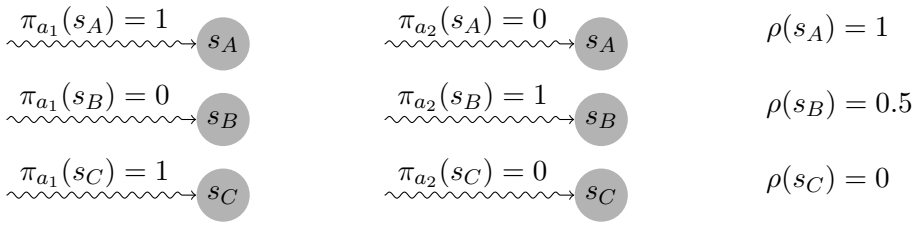


FIGURE I.3 – *Illustration de l'exemple de la section I.2 à propos des critères qualitatifs. L'action a_1 maximise le critère optimiste (I.5), qui peut mener au meilleur état (s_A), mais aussi au pire (s_C). Au contraire, l'action a_2 maximise le critère pessimiste (I.6) puisque le pire état n'est pas atteignable avec cette action.*

I.3 Processus Décisionnel Markovien Possibiliste Qualitatif (π-PDM)

Un *processus décisionnel markovien possibiliste qualitatif*, ou π-PDM, présenté dans [64, 63, 62], est la version possibiliste qualitative des PDM probabilistes décrits en introduction, basée sur les critères (optimistes et pessimistes) présentés en section I.2.

L'ensemble fini des états du système décrivant l'agent et son environnement, reste noté $\mathcal{S}$, comme vu en introduction avec les modèles probabilistes. L'ensemble fini des actions est toujours $\mathcal{A}$ et $\mathcal{L}$ est l'échelle possibiliste $\left\{0, \frac{1}{k}, \ldots, 1\right\}$, avec $k \geqslant 2$.

Comme dans le cas probabiliste, ce modèle considère que les états successifs du système, représentés par la séquence de variables $(S_t)_{t\in\mathbb{N}}$ avec $S_t \in \mathcal{S}\ \forall t \geqslant 0$, sont markovien. Dans ce cadre possibiliste qualitatif, cela signifie que la séquence $(S_t)_{t\in\mathbb{N}}$ est telle que $\forall t \geqslant 0, \forall (s_0, s_1, \ldots, s_{t+1}) \in \mathcal{S}^{t+2}$ et pour chaque séquence d'actions $(a_t)_{t\geqslant 0} \in \mathcal{A}^{\mathbb{N}}$, la variable S_{t+1} est indépendante (au sens causal) des variables $\{S_0, \ldots, S_{t-1}\}$, conditionnellement à $\{S_t = s\}$ et a_t :

$$\Pi\Big(S_{t+1} = s_{t+1} \;\Big|\; S_t = s_t, a_t \Big) = \Pi\Big(S_{t+1} = s_{t+1} \;\Big|\; S_t = s_t, S_{t-1} = s_{t-1}, \ldots, S_0 = s_0, (a_t)_{t\geqslant 0} \Big). \tag{I.7}$$

Cette indépendance possibiliste, ici causale, n'est pas la seule existante : une présentation générale des indépendances et des conditionnements, ainsi que de leurs conséquences dans les modèles graphiques, est disponible dans la thèse de N.Ben Amor [6].

En utilisant cette propriété markovienne, la dynamique du système est entièrement décrite avec les transitions possibilistes $\pi_t(s' \mid s, a) = \Pi\Big(S_{t+1} = s' \;\Big|\; S_t = s, a \Big) \in \mathcal{L} : \forall t \geqslant 0$, $(s, s') \in \mathcal{S}^2$ et $a \in \mathcal{A}$, $\pi_t(s' \mid s, a)$ est le degré de possibilité, qu'à l'étape de temps t, le système atteigne l'état s' lorsque l'agent choisit l'action a, conditionné au fait que l'état courant est s.

Enfin, un π-PDM est entièrement défini avec la séquence de fonctions de préférence $(\rho_t)_{t=0}^{H-1}$, où $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\rho_t(s, a)$ est le degré de préférence lorsque l'état du système est s et l'agent sélectionne l'action a au temps t. La fonction de préférence terminale, Ψ, donne pour chaque état du système $s \in \mathcal{S}$, le degré de préférence de $S_H = s : \Psi(s)$.

Afin de définir les critères des π-PDM à partir des critères possibilistes (I.5) et (I.6), nous introduisons, pour un horizon $H \geqslant 0$, les *trajectoires de longueur H*, $\mathcal{T} = (s_1, \ldots, s_H)$, et $\mathcal{T}_H = \mathcal{S}^H$ l'ensemble de telles trajectoires. Une règle de décision est notée $\delta : \mathcal{S} \to \mathcal{L}$, et une stratégie de longueur H est une séquence de règles de décision $\delta_t : (\delta_t)_{t=0}^{H-1}$. L'ensemble de toutes les stratégies de taille H est noté Δ_H. Dans [61], pour une stratégie donnée $(\delta) \in \Delta_H$, une séquence d'états du système $\mathcal{T} = (s_1, \ldots, s_H) \in \mathcal{T}_H$, et un état initial donné $s_0 \in \mathcal{S}$, la *préference d'une stratégie de taille H commençant par s_0* est définie comme le degré de

possibilité le plus petit de la trajectoire et de s_0 :

$$\rho\Big(\mathcal{T},(\delta)\Big) = \min\left\{\min_{t=0}^{H-1}\rho\Big(s_t,\delta_t(s_t)\Big),\Psi(s_H)\right\}.$$

En utilisant la propriété de Markov de ce processus d'états du système, pour un état initial donné $s_0 \in \mathcal{S}$, un horizon $H \in \mathbb{N}$, et une stratégie $(\delta_t)_{t=0}^{H-1}$, le degré de possibilité de la trajectoire $\mathcal{T} = (s_1,\ldots,s_H)$ est

$$\Pi\left(S_H = s_h, S_{H-1} = s_{h-1},\ldots,S_1 = s_1 \middle| S_0 = s_0,(\delta_t)_{t=0}^{H-1}\right) = \min_{t=0}^{H-1}\pi_{t+1}\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \quad \text{(I.8)}$$

noté $\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)$.

L'intégrale de Sugeno de la préférence de la trajectoire contre cette distribution est notée

$$\mathbb{S}_\Pi\left[\rho\Big(\mathcal{T},(\delta)\Big)\;\middle|\; S_0 = s_0,(\delta)\right] = \mathbb{S}_\Pi\left[\min\left\{\min_{t=0}^{H-1}\rho\Big(S_t,\delta_t(S_t)\Big),\Psi(S_H)\right\}\;\middle|\; S_0 = s_0,(\delta)\right]$$

et correspond du critère optimiste definissant la stratégie optimale, *i.e.* une fonction valeur optimiste :

$$\overline{U_H}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big) = \max_{\mathcal{T}\in\mathcal{T}_H}\min\left\{\rho\Big(\mathcal{T},(\delta)\Big),\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\right\}. \quad \text{(I.9)}$$

C'est équivalent au critère optimiste (I.5), cependant, l'intégrale est sur les trajectoires $\mathcal{T}_H$, et la préférence dépend de la stratégie. La *stratégie optimale optimiste* $\overline{\delta^*}$ est la stratégie maximisant la fonction valeur optimiste (I.9), et la *fonction valeur optimiste optimale* est la fonction valeur optimiste la plus grande en faisant varier $(\delta) \in \Delta_H$:

Définition I.3.1 (*Fonction valeur et stratégie optimiste optimale*)

$\forall s \in \mathcal{S}$,

$$\overline{U_H^*}(s) = \max_{(\delta)\in\Delta_H}\left\{\overline{U_H}\Big(s,(\delta)\Big)\right\} \qquad \textit{(fonction valeur optimale optimiste)}, \quad \text{(I.10)}$$

$$\overline{\delta^*}(s) \in \operatorname*{argmax}_{(\delta)\in\Delta_H}\left\{\overline{U_H}\Big(s,(\delta)\Big)\right\} \qquad \textit{(stratégie optimale optimiste)}. \quad \text{(I.11)}$$

De même, le critère possibiliste qualitatif optimal (I.6) mène à un critère prudent pour les stratégies : la fonction valeur pessimiste est l'intégrale de Sugeno de la préférence de la trajectoire contre la mesure de nécessité qui vient de la distribution de possibilité sur les trajectoires $\mathcal{T}_H$ (I.8) avec la stratégie $(\delta) \in \Delta_H$:

$$\underline{U_H}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big) = \min_{\mathcal{T}\in\mathcal{T}_H}\max\left\{\rho\Big(\mathcal{T},(\delta)\Big),1-\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\right\}. \quad \text{(I.12)}$$

notée $\mathbb{S}_\mathcal{N}\left[\rho\Big(\mathcal{T},(\delta)\Big)\;\middle|\; S_0 = s,(\delta)\right]$. Comme précédemment pour le cas optimiste, la *stratégie optimale pessimiste* $\underline{\delta^*}$ est la stratégie maximisant la fonction valeur pessimiste (I.12), et la *fonction valeur pessimiste optimale* est la fonction valeur maximale en faisant varier la stratégie $(\delta) \in \Delta_H$:

Définition I.3.2 (*Fonction valeur et stratégie pessimiste optimale*)

$\forall s \in \mathcal{S}$,

$$\underline{U_H^*}(s) = \max_{(\delta)\in\Delta_H} \left\{ \underline{U_H}\Big(s,(\delta)\Big) \right\} \qquad \textit{(fonction valeur optimale pessimiste)}, \tag{I.13}$$

$$\underline{\delta^*}(s) \in \operatorname*{argmax}_{(\delta)\in\Delta_H} \left\{ \underline{U_H}\Big(s,(\delta)\Big) \right\} \qquad \textit{(stratégie optimale pessimiste)}. \tag{I.14}$$

Les fonctions valeur optimales et les stratégies peuvent être calculées par programmation dynamique :

Théoreme 2 (*Programmation Dynamique pour π-PDM*)

Le critère optimiste optimal et la stratégie optimale associée peuvent être calculés comme suit : $\forall s \in \mathcal{S}$,

$$\begin{aligned} &\overline{U_0^*}(s) = \Psi(s), \qquad \textit{and, } \forall 1 \leqslant i \leqslant H, \\ &\overline{U_i^*}(s) = \max_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min \left\{ \pi_{H-i}\left(s' \mid s,a\right), \overline{U_{i-1}^*}(s') \right\} \right\}. \end{aligned} \tag{I.15}$$

$$\overline{\delta_{H-i}^*}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min \left\{ \pi_{H-i}\left(s' \mid s,a\right), \overline{U_{i-1}^*}(s') \right\} \right\}. \tag{I.16}$$

De même, le critère pessimiste optimal, et la stratégie associée peuvent être calculés comme suit : $\forall s \in \mathcal{S}$,

$$\begin{aligned} &\underline{U_0^*}(s) = \Psi(s), \qquad \textit{and, } \forall 1 \leqslant i \leqslant H, \\ &\underline{U_i^*}(s) = \max_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max \left\{ 1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U_{i-1}^*}(s') \right\} \right\}. \end{aligned} \tag{I.17}$$

$$\underline{\delta_{H-i}^*}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max \left\{ 1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U_{i-1}^*}(s') \right\} \right\}. \tag{I.18}$$

Dans ce théorème, l'horizon i est l'opposé de l'étape du processus t modulo H : durant l'éxécution, $\delta_t = \delta_{H-i}$ est utilisée à l'étape de temps t, *i.e.* lorsque il reste i étapes.

Notons qu'une classe plus large de modèles PDM, incluant les PDM probabilistes et possibilistes, est appelé *PDM algébrique* [56].

La section suivante est dédiée à la présentation de l'homologue possibiliste qualitatif du PDMPO noté π-PDMPO : le modèle π-PDMPO est la version partiellement observable du modèle π-PDM. Ce modèle a été présenté dans [62] dans le cadre pessimiste. L'algorithme pour le résoudre a été présenté dans le cas où aucune préférence intermédiaire n'est impliquée *i.e.* dans le cas où les fonctions de préférence ρ_t ne sont pas prises en compte : dans ce cas, seule la fonction de préférence terminale Ψ modélise le but de la mission. Enfin, on remarque qu'il suffit de considérer un π-PDM classique avec $\rho_t(s,a) = 1$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$ et $\forall t \in \{0,\ldots,H-1\}$.

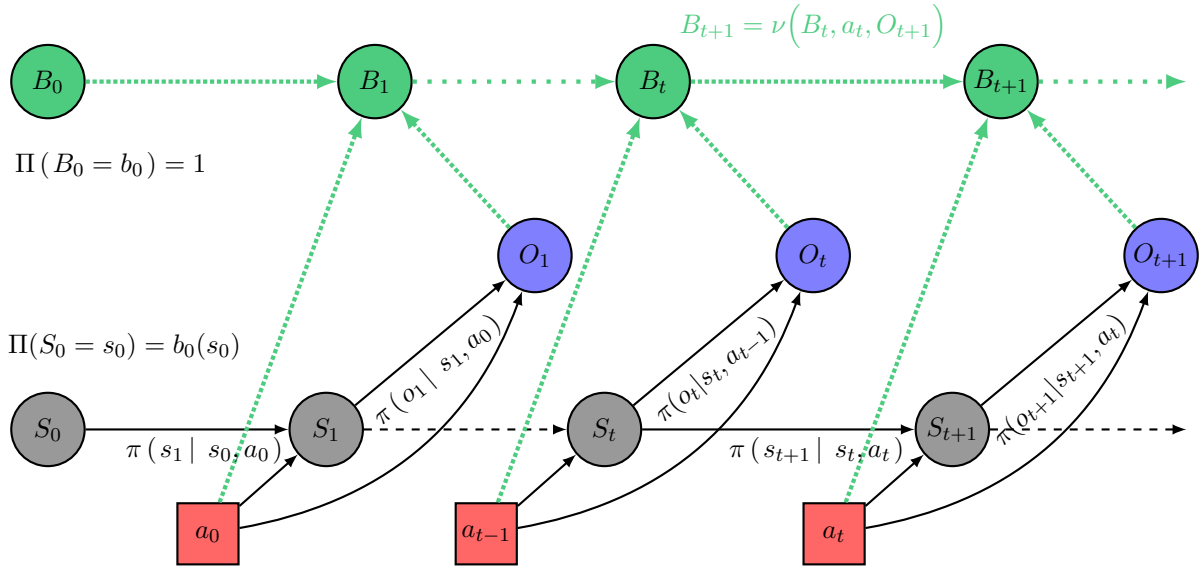


FIGURE I.4 – *Diagramme d'Influence d'un π-PDMPO et de son processus d'états de croyance : les ronds noirs représentent les états successifs du système S_t, les bleus représentent les observations successives O_t, et les carrés rouges sont les actions sélectionnées a_t. Les cercles verts en haut de la figure sont les états de croyance successifs B_t constituant le processus d'états de croyance, calculé avec la mise à jour $B_{t+1} = \nu(B_t, a_t, O_{t+1})$. Les lignes vertes en pointillé représentent des influences déterministes.*

I.4 PDM Partiellement Observable Possibiliste Qualitatif (π-PDMPO)

Le modèle PDMPO possibiliste qualitatif (π-PDMPO) a été présenté pour la première fois dans [62]. Comme pour le modèle probabiliste, dans le cadre partiellement observable, l'état du système n'est plus considéré comme une donnée d'entrée pour l'agent : l'agent doit l'estimer à partir des observations $o \in \mathcal{O}$ reçues à chaque étape de temps, représentées par le processus d'observation $(O_t)_{t\in\mathbb{N}}$. L'incertitude à propos des variables d'observation successives O_t ne dépend que de l'action et de l'état atteint : si l'agent choisit l'action $a \in \mathcal{A}$ au temps t, et le système a atteint l'état $s' \in \mathcal{S}$ à l'étape de temps $t+1$, l'observation $o' \in \mathcal{O}$ est reçue avec le degré de possibilité $\pi_t\,(o' \mid s', a\,) = \Pi\,(O_{t+1} = o' \mid S_{t+1} = s', a\,)$: conditionnellement à l'état suivant s' et à l'action courante a, la variable d'observation suivante est indépendante (au sens causal) de toutes les autres variables jusqu'à l'étape $t+1$. La figure I.4 illustre la dynamique et la structure d'un π-PDMPO. Un PDMPO probabiliste a la même structure, sauf que les distributions de possibilité de transition (resp. d'observation) $\pi\,(s' \mid s, a\,)$ (resp. $\pi\,(o' \mid s', a\,)$). doivent être remplacée par la distribution de probabilité $\mathbf{p}\,(s' \mid s, a\,)$ (resp. $\mathbf{p}\,(o' \mid s', a\,)$).

Comme avec le modèle probabiliste, le calcul des stratégies est effectué en traduisant le π-PDMPO en un π-PDM entièrement observable. L'espace d'état de ce dernier est l'ensemble des *états de croyance possibilistes qualitatifs* $\beta : \mathcal{S} \to \mathcal{L}$ décrivant la connaissance à propos de l'état du système, *i.e.* l'ensemble de distributions de possibilité sur $\mathcal{S}$. Cet ensemble est noté $\Pi_{\mathcal{L}}^{\mathcal{S}} = \{\,\pi : \mathcal{S} \to \mathcal{L} \mid \max_{s\in\mathcal{S}} \pi(s) = 1\,\}$. Notons tout d'abord que le nombre d'états de croyance à propos de l'état du système est

$$(\#\mathcal{L})^{\#\mathcal{S}} - (\#\mathcal{L} - 1)^{\#\mathcal{S}}\,. \tag{I.19}$$

En effet, il y a $\#\mathcal{L}^{\#\mathcal{S}}$ fonctions différentes de $\mathcal{S}$ vers $\mathcal{L}$, et $(\#\mathcal{L} - 1)^{\#\mathcal{S}}$ fonctions non normalisées *i.e.* fonctions $f : \mathcal{S} \to \mathcal{L}$ telles que $\max_{s\in\mathcal{S}} f(s) < 1$. Le nombre de distributions de

possibilité sur $\mathcal{S}$ est le nombre de fonctions normalisées de $\mathcal{S}$ vers $\mathcal{L}$, *i.e.* le nombre total de fonctions, moins le nombre de fonctions non normalisées.

Tout d'abord, définissons formellement un π-PDMPO comme le 7-uplet $< \mathcal{S}, \mathcal{A}, \mathcal{O}, T^\pi, O^\pi, \Psi, \beta_0 >$:

- $\mathcal{S}$, un ensemble fini d'états (cachés) du système ;
- $\mathcal{A}$ un ensemble fini d'actions ;
- $\mathcal{O}$ un ensemble fini d'observations ;
- T^π l'ensemble des distributions de possibilité de transition $\pi(s' \mid s, a)$;
- O^π, l'ensemble des distributions de possibilité d'observation $\pi(o' \mid s', a)$;
- Ψ la fonction de préférence, définissant pour chaque état $s \in \mathcal{S}$, la préférence associée à la situation où l'état du système est $s \in \mathcal{S}$;
- β_0, *l'état de croyance possibiliste initial*, est la distribution de possibilité définissant l'incertitude à propos de l'état initial : $\forall s \in \mathcal{S}$, $\beta_0(s) = \Pi(S_0 = s)$.

À chaque étape de temps, l'état de croyance possibiliste est calculé à partir de ces objets : 'état de croyance initial $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ fait partie de la définition du π-PDMPO.

À l'étape de temps $t \geqslant 1$, l'état de croyance est la distribution de possibilité sur l'état courant du système, conditionnellement à toutes les données visibles par l'agent.

Définition I.4.1 (*État de croyance possibiliste qualitatif*)

$$\beta_t(s) = \Pi(S_t = s \mid O_1 = o_1, \ldots, O_t = o_t, a_0, \ldots, a_{t-1}) = \Pi(S_t = s \mid I_t = i_t) \tag{I.20}$$

où $i_t = \{o_1, \ldots, o_t, a_0, \ldots, a_{t-1}\}$ est l'information visible par l'agent à l'étape de temps t ($i_0 = \{\} = \emptyset$), et I_t la variable correspondante.

La mise à jour possibiliste de l'état de croyance est basée sur l'homologue possibiliste de la règle de Bayes :

Théoreme 3 (*Mise à jour possibiliste qualitative de l'état de croyance*)

Si l'état de croyance à l'étape de temps t est β_t, l'action choisie est $a_t \in \mathcal{A}$, et l'état suivant est o_{t+1}, l'état suivant β_{t+1} est calculé comme suit :

$$\beta_{t+1}(s') = \begin{cases} 1 & \text{si } \pi_t(s', o_{t+1} \mid \beta_t, a_t) = \pi_t(o_{t+1} \mid \beta_t, a_t), \\ \pi_t(s', o_{t+1} \mid \beta_t, a_t) & \text{sinon.} \end{cases} \tag{I.21}$$

où la distribution de possibilité jointe sur la variable d'état S_{t+1} et la variable d'observation O_{t+1} conditionnellement à l'information courante, est notée $\pi_t(s', o' \mid \beta_t, a_t) = \min\left\{\pi_t(o' \mid s', a_t), \max_{s \in \mathcal{S}} \min\left\{\pi_t(s' \mid s, a_t), \beta_t(s)\right\}\right\}$. La notation $\pi(o' \mid \beta_t, a_t)$ est aussi utilisée pour $\max_{s' \in \mathcal{S}} \pi_t(s', o' \mid \beta_t, a_t)$.

Cette formule est appelée la ***mise à jour de la croyance possibiliste****, et puisque l'état de croyance β_{t+1} est une fonction de β_t, a_t and o_{t+1}, nous notons cette mise à jour*

$$\beta_{t+1} = \nu(\beta_t, a_t, o_{t+1}),$$

avec ν appelée fonction de mise à jour.

La mise à jour possibiliste de l'état de croyance (I.21) est notée

$$\beta_{t+1}(s') \propto^\pi \pi_t(s', o_{t+1} \mid \beta_t, a_t)$$

puisqu'elle consiste seulement à normaliser la fonction $s' \mapsto \pi(s', o_{t+1} \mid \beta_t, a_t)$ au sens possibiliste ($\max_s \pi(s) = 1$).

Nous notons B_t^π l'état de croyance lorsque considéré comme une variable : B_0^π est déterministe égal à β_0 (mais S_0 est incertain, ce qui est décrit par la distribution de possibilité β_0) et $B_{t+1}^\pi = \nu(B_t^\pi, a_t, O_{t+1})$ où O_{t+1} est la variable d'observation à l'étape de temps $t+1$.

Afin de rendre les choses plus claires, le modèle π-PDMPO est ici défini sans préférences intermédiaires ρ_t, mais avec une préférence terminale Ψ uniquement.

Nous pouvons maintenant exprimer la distribution de possibilité de transition du processus de croyance *i.e.* les éléments de $\tilde{T}$, comme suit : $\forall t \geqslant 0$,

$$\pi_t\left(\beta' \mid \beta, a\right) = \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \pi_t\left(o' \mid \beta, a\right), \tag{I.22}$$

où $\pi_t\left(o' \mid \beta, a\right) = \max_{(s,s') \in \mathcal{S}^2} \min\Big\{\pi_t\left(o' \mid s', a_t\right), \pi_t\left(s' \mid s, a_t\right), \beta(s)\Big\}$, est le degré de possibilité d'observer o' conditionnellement à toutes les informations précédentes.

Enfin, la fonction de préférence associée à l'état de croyance possibiliste β_H est défini de manière pessimiste : $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$,

$$\underline{\Psi}(\beta) = \min_{s \in \mathcal{S}} \max\left\{\Psi(s), 1 - \beta(s)\right\}. \tag{I.23}$$

Le cas optimiste donne une préférence maximale à l'ignorance totale, ce qui ne semble pas pouvoir mener l'agent à estimer l'état du système et à atteindre un état satisfaisant.

Il est possible de montrer qu'il est suffisant de chercher une stratégie basée sur les états de croyance *i.e.* parmi les suites de règles de décision $(\delta_t)_{t=0}^{H-1}$, telles que $\forall t \geqslant 0$, $\delta_t : \beta_t \mapsto \delta_t(\beta_t) \in \mathcal{A}$.

Le π-PDM $\langle \tilde{\mathcal{S}}^\pi, \mathcal{A}, \tilde{T^\pi}, \underline{\Psi} \rangle$ construit à partir d'un π-PDMPO $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T^\pi, O^\pi, \beta_0 \rangle$ est finalement :

- $\tilde{\mathcal{S}}^\pi = \Pi_{\mathcal{L}}^{\mathcal{S}}$, l'ensemble de tous les états de croyance (possibilistes qualitatifs) possibles ;
- $\tilde{T^\pi}$ contient toutes les distributions de possibilité de transition sur les états de croyance : $\forall a \in \mathcal{A}$, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, ces distributions de possibilité sont définies par l'équation (I.22), $\pi_t\left(. \mid \beta, a\right)$ is in $\tilde{T^\pi}$;
- la fonction de préférence est l'estimation pessimiste de la préférence : $\underline{\Psi}$, *cf.* équation (I.23).

Notons maintenant que, en utilisant la définition de la fonction de transition des états de croyance (I.22), pour chaque fonction de l'espace des états de croyance vers $\mathcal{L}$, $U : \Pi_{\mathcal{L}}^{\mathcal{S}} \to \mathcal{L}$,

$$\begin{aligned}
\max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min\left\{\pi_t\left(\beta' \mid \beta, a\right), U(\beta')\right\} &= \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min\left\{\max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \pi_t\left(o' \mid \beta, a\right), U(\beta')\right\} \\
&= \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \min\left\{\pi_t\left(o' \mid \beta, a\right), U(\beta')\right\} \\
&= \max_{o' \in \mathcal{O}} \min\Big\{\pi_t\left(o' \mid \beta, a\right), U\Big(\nu(\beta, a, o')\Big)\Big\},
\end{aligned}$$

Cette observation mène à l'algorithme 1 qui est l'application directe du schéma de programmation dynamique optimiste pour π-PDM (théorème 2) avec seulement une préférence terminale (qui est pessimiste) : ce schéma est utilisé sur le π-PDM $\langle \tilde{\mathcal{S}}^\pi, \mathcal{A}, \tilde{T^\pi}, \underline{\Psi} \rangle$.

L'algorithme 2, qui est la programmation dynamique pessimiste pour π-PDM (théorème 2), avec une préférence terminale seulement : elle est appliquée au π-MDP $\langle \tilde{\mathcal{S}}^\pi, \mathcal{A}, \tilde{T^\pi}, \underline{\Psi} \rangle$.

Dans ce premier chapitre la théorie des possibilités qualitatives ainsi que les modèle π-PDM et π-PDMPO ont été présentés.

Dans le cadre probabiliste des PDMPO, l'incertitude est décrite avec des probabilités $\mathbf{p}\left(s' \mid s, a\right) \in \mathbb{R}$ et $\mathbf{p}\left(o' \mid s', a\right) \in \mathbb{R}$ tandis qu'elle est définie par des distributions de possibilité

Algorithm 1: Algorithme de programmation dynamique pour π-PDMPO optimiste avec une préférence sur l'état final seulement

$\overline{U_0^*} \leftarrow \underline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\overline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ \pi_t\left(o' \mid \beta, a\right), \overline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\}$;
    $\overline{\delta_{H-i}}(\beta) \in \operatorname*{argmax}_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ \pi_t\left(o' \mid \beta, a\right), \overline{U_{i-1}^*}\left(\nu(\beta, a, o')\right) \right\}$;
return $\overline{U_H^*}$, $(\overline{\delta^*})$;

Algorithm 2: Algorithme de programmation dynamique pour π-PDMPO pessimiste avec une préférence terminale seulement

$\underline{U_0^*} \leftarrow \underline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\underline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t\left(o' \mid \beta, a\right), \underline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\}$;
    $\underline{\delta_{H-i}}(\beta) \in \operatorname*{argmax}_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t\left(o' \mid \beta, a\right), \underline{U_{i-1}^*}\left(\nu(\beta, a, o')\right) \right\}$;
return $\underline{U_H^*}$, $(\underline{\delta^*})$;

$\pi\left(s' \mid s, a\right) \in \mathcal{L} = \left\{0, \frac{1}{k}, \ldots, 1\right\}$ (avec $k \geqslant 1$) et $\pi\left(o' \mid s', a\right) \in \mathcal{L}$ dans le cadre π-PDMPO. De plus, le cadre probabiliste mesure l'intérêt de passer par un état $s \in \mathcal{S}$ et d'utiliser l'action $a \in \mathcal{A}$ avec la fonction de récompense $r(s, a) \in \mathbb{R}$ tandis que le cadre possibiliste qualitatif utilise des préférences $\rho(s, a) \in \mathcal{L}$ et $\Psi(s) \in \mathcal{L}$. Ainsi, le critère probabiliste pour une stratégie donnée, est l'espérance des récompenses, ce qui s'écrit $\mathbb{E}\left[rewards\Big((S_t)_{t \geqslant 0}\Big)\right] \in \mathbb{R}$: dans le cadre possibiliste, deux critères sont possibles, qui sont les intégrales de Sugeno de la préférence, $\mathbb{S}_{\Pi}\left[preferences\Big((S_t)_{t \geqslant 0}\Big)\right] \in \mathcal{L}$ pour l'optimiste, et $\mathbb{S}_{\mathcal{N}}\left[preferences\Big((S_t)_{t \geqslant 0}\Big)\right] \in \mathcal{L}$ pour la pessimiste.

Un PDMPO (resp. π-PDMPO), est redéfini en termes de PDM (resp. π-PDM) entièrement observable où l'état du système est l'état de croyance $b_t \in \mathbb{P}_{b_0}^{\mathcal{S}}$ (resp. $\beta_t \in \Pi_{\mathcal{L}}^{\mathcal{S}}$), *i.e.* les distribution de probabilité (resp. possibilité) sur les états du système du PDMPO : la récompense basée sur la croyance est définie comme étant $r(b, a) = \mathbb{E}_{S \sim b}\left[r(S, a)\right] = \sum_{s \in \mathcal{S}} r(s, a) \cdot b(s)$ dans le cas probabiliste. Dans le cas possibiliste qualitatif, la préférence basée sur l'état de croyance est $\underline{\rho}(b, a) = \mathbb{S}_{\mathcal{N}, S \sim \beta}\left[\rho(S, a)\right] = \min_{s \in \mathcal{S}} \max\left\{\rho(s, a), 1 - \beta(s)\right\}$ pour les pessimistes.

Le chapitre suivant propose certaines amélioration au modèle possibiliste qualitatif : La propriété d'observabilité mixte est définie : comme pour le modèle probabiliste, la complexité de résolution des problèmes ayant cette propriété est réduite. Enfin, l'homologue du problème à horizon infini est formellement défini, et il est prouvé que l'algorithme de résolution proposé pour le résoudre renvoie une stratégie optimale.

Mises à Jour et Étude Pratique des π-PDMPO

II

La fin du chapitre précédent a présenté le modèle π-PDMPO, l'homologue possibiliste qualitatif des PDMPO probabilistes. Rappelons que, dans le cadre possibiliste qualitatif, l'ensemble des états de croyance est fini, $\#\Pi_{\mathcal{L}}^{\mathcal{S}} < +\infty$ (*cf.* équation I.19) tandis que l'ensemble des états de croyance est infini dans le cadre probabiliste $\mathbb{P}_{b_0}^{\mathcal{S}}$: pour cette raison, les π-PDMPO peuvent être vus comme un modèle plus simple que les PDMPO probabilistes pour la décision séquentielle dans l'incertain avec observabilité partielle.

Ce modèle peut simplifié de plus belle, lorsque le problème satisfait la propriété d'*observabilité mixte*, comme montré dans la section qui suit.

II.1 Observabilité Mixte et π-PDM à Observabilité Mixte (π-PDMOM)

La résolution d'un π-PDMPO se confronte au fait que la taille de l'espace des états de croyance $\Pi_{\mathcal{L}}^{\mathcal{S}}$ est exponentielle en fonction de la taille de l'espace des états du système $\mathcal{S}$, *cf.* équation (I.19) de la section I.4. Cependant, en pratique, les états du système sont rarement totalement cachés. Utiliser la propriété d'observabilité mixte peut être une solution : inspiré par

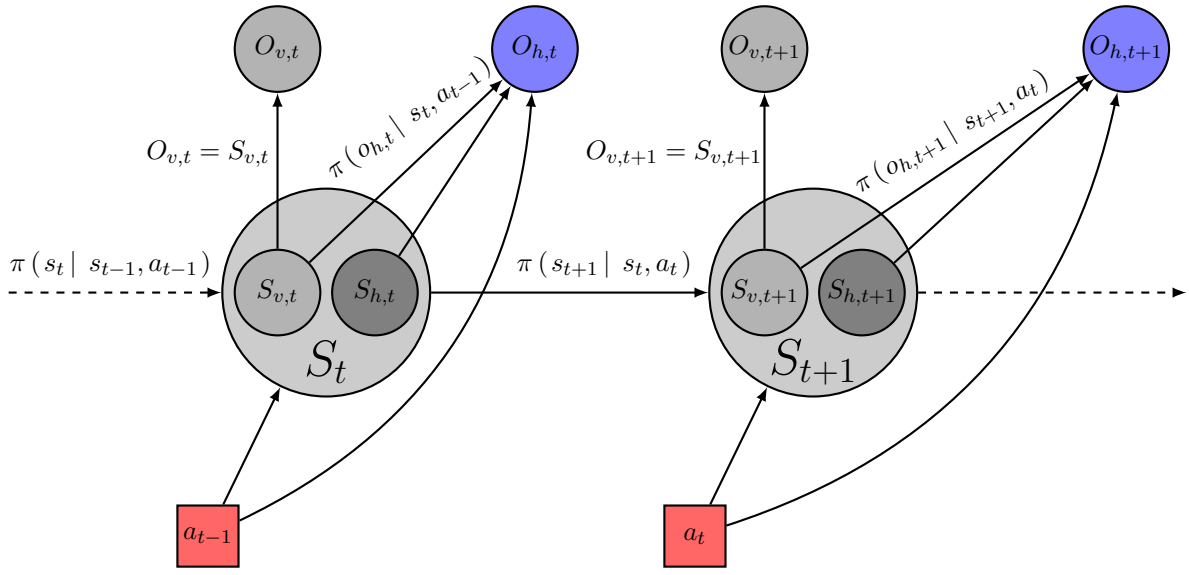


Figure II.1 – *Réseau bayésien dynamique d'un π-PDMOM : à l'étape de temps t, l'état du système est décrit par la variable $S_t = (S_{v,t}, S_{h,t})$. L'observation reçue est $O_t = (O_{v,t}, O_{h,t})$ avec $O_{v,t} = S_{v,t}$, et $O_{h,t}$ dépend de S_t et de l'action a_t.*

un travail récent dans le cadre des PDMPO probabilistes, [52, 2], nous présentons dans cette section un modèle structuré qui prend en compte les situations dans lesquelles l'agent observe directement une partie de l'état du système. Un π-PDMPO qui modélise une telle situation respecte la *propriété d'observabilité mixte*. Les états de croyance ne sont alors utilisés que pour les composantes partiellement observables et la taille de l'espace d'état est significativement résuite. Ainsi, ce modèle généralise les π-PDM et les π-PDMPO.

Comme dans [2], nous faisons l'hypothèse que l'espace d'état $\mathcal{S}$ d'un PDMOM possibiliste qualitatif (π-PDMOM) peut être écrit comme le produit cartésien d'un espace d'états visibles $\mathcal{S}_v$ et d'un espace d'états cachés $\mathcal{S}_h$: $\mathcal{S} = \mathcal{S}_v \times \mathcal{S}_h$. Soit $s = (s_v, s_h)$ un état du système. La composante s_v est directement observée par l'agent et s_h est seulement observé partiellement à travers les observations de l'ensemble $\mathcal{O}_h$: nous notons $\pi_t(o'_h \mid s', a)$, la distribution de possibilité sur l'observation suivante $o'_h \in \mathcal{O}_h$ à l'étape de temps t, connaissant l'état suivant $s' \in \mathcal{S}$ et l'action courante $a \in \mathcal{A}$. La figure II.1 illustre la structure de ce modèle à observabilité mixte.

L'espace d'état visible est identifié à l'espace des observations : $\mathcal{O}_v = \mathcal{S}_v$ et $\mathcal{O} = \mathcal{O}_v \times \mathcal{O}_h$. Ainsi, sachant que la composante visible de l'état est s_v, l'agent observe *nécessairement* $o_v = s_v$ ($\forall a \in \mathcal{A}$, si $o'_v \neq s_v$, $\pi_t(o'_v \mid s'_v, a) = 0$). Formellement, vu comme un π-PDMPO, sa distribution de possibilité sur les observations peut s'écrire :

$$\begin{aligned}
\pi_t(o' \mid s', a) &= \pi_t(o'_v, o'_h \mid s'_v, s'_h, a) \\
&= \min\{\pi_t(o'_h \mid s'_v, s'_h, a), \pi_t(o'_v \mid s'_v)\} \\
&= \begin{cases} \pi_t(o'_h \mid s', a) & \text{if } o'_v = s'_v \\ 0 & \text{sinon} \end{cases}, \qquad \text{(II.1)}
\end{aligned}$$

puisque $\forall a \in \mathcal{A}$, $\pi_t(o'_v \mid s'_v) = 1$ si $s'_v = o'_v$ et 0 sinon. Le théoreme suivant, basé sur cette égalité, permet de définir les états de croyance sur les états cachés du système.

Théoreme 4 (*Nature des états de croyance atteignables*)

Chaque état de croyance atteignable d'un π-PDMOM peut être écrit comme un élément de $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ où $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ est l'espace des distributions de possibilité sur $\mathcal{S}_h$: tout $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ atteignable peut s'écrire (s_v, β_h) avec $\beta_h(s_h) = \max_{\overline{s}_v \in \mathcal{S}_v} \beta(\overline{s}_v, s_h)$ et $s_v = \operatorname{argmax}_{\overline{s}_v \in \mathcal{S}_v} \beta(\overline{s}_v, s_h)$.

Comme tous les états de croyance sont dans $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ lorsque le π-PDMPO satisfait la propriété d'observabilité mixte, le théorème suivant réécrit la mise à jour du nouvel état de croyance $\beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ *i.e.* de l'état de croyance sur les états cachés du système $s_h \in \mathcal{S}_h$.

Théoreme 5 (*Mise à jour de la croyance pour un π-PDMOM*)

Si un problème peut se modéliser par un π-PDMOM

$$\Big\langle S_v \times S_h, \mathcal{A}, \mathcal{O}_h, T^\pi, O^\pi, \Psi, \beta_0 = (s_{v,0}, \beta_{h,0}) \Big\rangle,$$

une nouvelle fonction de mise à jour de l'état de croyance ν_h peut être définie : si, à l'étape de temps t, l'état visible courant est $s_{v,t} \in \mathcal{S}_v$, l'état de croyance courant sur l'état caché est $\beta_{h,t} \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, l'action choisie courante est $a_t \in \mathcal{A}$, l'état visible suivant est $s_{v,t+1} \in \mathcal{S}_v$ et l'observation suivante est $o_{h,t+1} \in \mathcal{O}_h$, alors l'état de croyance suivant à propos de l'état caché du système est

$$\beta_{h,t+1}(s'_h) = \begin{cases} 1 & \text{si } \begin{array}{l} \pi_t(s'_h, s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t) \\ = \pi_t(s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t) \end{array} \\ \pi_t(s'_h, s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t) & \text{sinon} \end{cases}, \qquad \text{(II.2)}$$

où

$$\pi_t\left(s'_v, s'_h, o'_h \mid\ s_v, \beta_h, a\right) = \min\left\{\pi_t\left(o'_h \mid\ s', a\right), \max_{s_h \in \mathcal{S}_h} \min\left\{\pi_t\left(s' \mid\ s_v, s_h, a\right), \beta_h(s_h)\right\}\right\}$$

est la distribution de possibilité jointe sur les états du système $s'_h \in \mathcal{S}_h$ et les objets visibles (état visible du système et observation) $s'_v \in \mathcal{S}_v$ et $o'_h \in \mathcal{O}_h$. La mise à jour de l'état de croyance est notée

$$\beta'_h = \nu_h\left(s_v, \beta_h, a, s'_v, o'_h\right).$$

L'espace d'état du π-PDM résultant d'un π-PDMOM peut alors être restreint à l'espace produit $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, *i.e.* un espace d'état plus petit grâce à l'observabilité mixte : le π-PDM résultant est $\langle \tilde{S^\pi}, \tilde{T^\pi}, \mathcal{A}, \tilde{\Psi}\rangle$, où

— l'espace des états du système est $\tilde{S^\pi} = \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$,

— la distribution de probabilité de transition dans $\tilde{T^\pi}$ est telle que $\forall\{0, \ldots, H-1\}$, $\forall a \in \mathcal{A}$, $\forall\Big[(s_v, \beta_h), (s'_v, \beta'_h)\Big] \in \left(\tilde{S^\pi}\right)^2$,

$$\pi_t\Big((s'_v, \beta'_h)\Big|(s_v, \beta_h), a\Big) = \max_{\substack{o'_h \in \mathcal{O}_h \text{ s.t.} \\ \nu_h(s_v, \beta_h, a, s'_v, o'_h) = \beta'_h}} \pi_t\left(s'_v, o'_h \mid\ s_v, \beta_h, a\right),$$

où $\pi_t\left(s'_v, o'_h \mid\ s_v, \beta_h, a\right)$ est défini dans le théorème au-dessus,

— la préférence pessimiste *i.e.* $\tilde{\Psi} = \underline{\Psi}$: elle peut être réécrite $\forall s_v \in \mathcal{S}_v, \forall \beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}, \forall a \in \mathcal{A}$,

$$\underline{\Psi}(s_v, \beta_h) = \min_{s_h \in \mathcal{S}_h} \max\left\{\Psi(s_v, s_h), 1 - \beta_h(s_h)\right\}.$$

Un algorithme standard aurait calculé la fonction valeur pour chaque $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ tandis que l'équation de programmation dynamique du π-PDM résultant mène à un algorithme qui la calcule seulement pour chaque $(s_v, \beta_h) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, puisque seuls ces états de croyance sont atteignables. La taille du nouvel espace des états de croyance est

$$\#(\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}) = \#\mathcal{S}_v \times \left(\#\mathcal{L}^{\#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_h}\right),$$

ce qui est exponentiellement plus petit que la taille de l'espace des états de croyance du π-PDMPO équivalent :

$$\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \#\mathcal{L}^{\#\mathcal{S}_v \times \#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_v \times \#\mathcal{S}_h}.$$

II.2 Horizon Indéterminé

Pour de nombreux problèmes en pratique, il est très difficile de déterminer un horizon H. Le but de cette section est de présenter un algorithme pour résoudre les π-PDMOM avec préférence terminale, et horizon indéterminé.

À notre connaissance, nous proposons ici le premier algorithme d'itération sur les valeurs (IV) pour π-PDM qui renvoie une stratégie optimale pour un critère spécifié. Comme mentionné dans [62], nous faisons l'hypothèse de l'existence d'une action "rester", notée $\widehat{a}$, qui retient le système dans son état courant avec nécessité 1. Cette action est l'homologue possibiliste du facteur d'actualisation γ dans le modèle probabiliste, puisqu'il garantit la convergence de l'algorithme d'itération sur les valeurs. Nous verrons cependant que l'action $\widehat{a}$ n'est finalement utilisée que sur certains états buts. Notons qu'une hypothèse similaire est utilisée pour calculer des stratégies optimales dans le cadre des processus déterministes (planification classique) dont l'horizon n'est pas spécifié [44].

Algorithm 3: Algorithme IV pour π-PDM – Préférence Terminale

for $s \in \mathcal{S}$ **do**
  $\overline{U^*}(s) \leftarrow 0$;
  $\overline{U^c}(s) \leftarrow \Psi(s)$;
  $\overline{\delta^*}(s) \leftarrow \widehat{a}$;
while $\overline{U^*} \neq \overline{U^c}$ **do**
  $\overline{U^*} = \overline{U^c}$;
  for $s \in \mathcal{S}$ **do**
    $\overline{U^c}(s) \leftarrow \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \left\{ \pi\left(s' \mid s, a\right), \overline{U^*}(s') \right\}$;
    if $\overline{U^c}(s) > \overline{U^*}(s)$ **then**
      $\overline{\delta^*}(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \left\{ \pi\left(s' \mid s, a\right), \overline{U^*}(s') \right\}$;
return $\overline{U^*}$, $\overline{\delta^*}$;

Nous notons $\widehat{\delta}$ la règle de décision telle que $\forall s \in \mathcal{S}$, $\widehat{\delta}(s) = \widehat{a}$. L'ensemble fini de toutes les statégies est $\Delta = \cup_{i \geqslant 1} \Delta_i$, et $\#\delta$ est la taille de la stratégie (δ) en termes d'étapes de décision. Nous pouvons maintenant définir le critère optimiste pour un horizon indéterminé : si $(\delta) \in \Delta$,

$$\overline{U}\Big(s_0, (\delta)\Big) = \max_{\mathcal{T} \in \mathcal{T}_{\#\delta}} \min \left\{ \pi\Big(\mathcal{T}\Big| s_0, (\delta)\Big), \Psi(s_{\#\delta}) \right\}, \tag{II.3}$$

où $\mathcal{T} = (s_1, \ldots, s_{\#\delta})$ est une trajectoire d'états du système, $\mathcal{T}_{\#\delta}$ l'ensemble de telles trajectoires, et

$$\pi\Big(\mathcal{T}\Big| s_0, (\delta)\Big) = \min_{i=0}^{\#\delta - 1} \pi\Big(s_{i+1}\Big| s_i, \delta_i(s_i)\Big).$$

Théoreme 6 (*Optimalité de l'algorithme d'IV pour les π-PDM optimistes*)

Si il existe une action $\widehat{a}$ telle que, pour chaque $s \in \mathcal{S}$, $\pi\left(s' \mid s, \widehat{a}\right) = 1$ si $s' = s$ et 0 sinon, alors l'algorithme 3 calcule le critère maximal et une stratégie optimale, i.e. qui maximise le critère (II.3), et qui est stationnaire (i.e. qui ne dépend pas de l'étape du processus t).

Soit s un état tel que $\overline{\delta^*}(s) = \widehat{a}$, où $\overline{\delta^*}$ est la stratégie renvoyée par l'algorithme. En regardant l'algorithme 3, nous pouvons remarquer que $\overline{U^*}(s)$ reste égal à $\Psi(s)$ durant les itérations de l'algorithme après le premier passage dans la boucle while. Donc, $\forall s' \in \mathcal{S}$, soit $\forall a \in \mathcal{A}$, $\Psi(s) \geqslant \pi\left(s' \mid s, a\right)$, soit $\Psi(s) \geqslant \overline{U^*}(s')$. Si le problème n'est pas trivial, cela signifie que s est un but $(\Psi(s) > 0)$ et que les degrés de possibilité de transition vers de meilleurs buts sont plus petit que le degré de préférence de s.

II.3 Résultats Expérimentaux

Considérons un robot sur une grille de taille $g \times g$, avec $g > 1$. Il connait parfaitement sa position sur la grille $(x, y) \in \{1, \ldots, g\}^2$ à chaque étape du processus, ce qui constitue l'espace des états visibles $\mathcal{S}_v$. Il se trouve initialement à la position $s_{v,0} = (1, 1)$. Deux cibles immobiles sont présentes sur la grille : la "cible 1" est en $(x_1, y_1) = (1, g)$, la "cible 2" se trouve en $(x_2, y_2) = (g, 1)$ sur la grille, et le robot connait parfaitement leurs positions. Une des deux cibles est A, l'autre est B, et la mission du robot est d'identifier et d'atteindre A aussi tôt que possible. Le robot ne sait pas quelle cible est A : les deux situations A_1 et A_2 correspondent respectivement à "la cible 1 est A" et "la cible 2 est A" et constituent l'espace des états cachés

$\mathcal{S}_h$. Les actions $\mathcal{A}$ sont les déplacements dans les quatre directions ainsi que l'action "rester" ; les déplacements du robot sont déterministes. A chaque étape du processus, le robot analyse une image de chaque cible et obtient alors une observation de la nature de la cible : les deux cibles peuvent sembler être A (oAA), ou bien seulement la cible 1 (oAB), ou seulement la cible 2 (oBA), ou alors aucune des deux (oBB).

Dans le cadre probabiliste, la probabilité de recevoir une bonne observation de la cible $i \in \{1,2\}$, n'est pas vraiment connue, mais est approchée par $Pr(good_i \mid x,y) = \frac{1}{2}\left[1+\exp\left(-\frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{D}\right)\right]$ où $(x,y) = s_v \in \{1,\ldots,g\}^2$ est la position du robot, (x_i,y_i) la position de la cible i, et $D > 0$ une constante de normalisation. Les processus d'observation de chaque cible sont considérés indépendants. Alors, par exemple, $Pr(oAB \mid (x,y), A_1)$ est égal à $Pr(good_1 \mid (x,y)) \cdot Pr(good_2 \mid (x,y))$, $Pr(oAA \mid (x,y), A_1)$ à $Pr(good_1 \mid (x,y)) \cdot [1 - Pr(good_2 \mid (x,y))]$, etc. Chaque étape du processus avant d'atteindre une cible coûte 1, atteindre la cible A et y rester est récompensé par 100, et par -100 pour B. La stratégie provenant du modèle probabiliste a été calculée en tenant compte de l'Observabilité Mixte du problème, avec *APPL* [52], en utilisant une précision de 0.046 (la limite en mémoire est atteinte pour une précision supérieure) et $\gamma = 0.99$. Ce problème ne peut pas être résolu par l'algorithme exact pour PDMOM [2] puisque cela entraîne l'utilisation de toute la mémoire vive disponible après 15 itérations.

Avec la théorie des possibilités qualitatives, il est toujours possible d'observer correctement la cible : $\pi(good \mid x,y) = 1$. Ensuite, plus le robot est loin de la cible i, plus il est susceptible de mal l'observer (par exemple observer A au lieu de B), ce qui est une hypothèse raisonnable compte tenu du fait que le modèle d'observation est mal connu : $\pi(bad_i \mid x,y) = \frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{\sqrt{2}(g-1)}$. Ainsi, par exemple, $\pi(oAB \mid (x,y), A_1) = 1$, $\pi(oAA \mid (x,y), A_1) = \pi(bad_2 \mid x,y)$, $\pi(oBA \mid (x,y), A_1) = \min\{\pi(bad_1 \mid x,y), \pi(bad_2 \mid x,y)\}$, etc. Puisque la situation est complètement connue lorsque le robot est sur la position d'une cible (observation déterministe), il n'y a pas de risque d'être bloqué dans un état non satisfaisant, et c'est pourquoi le modèle π-PDMOM *optimiste* fonctionne bien. L'échelle $\mathcal{L}$ est composée de 0, 1, et toutes les valeurs possibles de $\pi(bad_i \mid x,y) \in [0,1]$. Notons qu'une construction de ce modèle avec une transformation probabilité-possibilité [30] aurait été équivalente. La distribution de préférence Ψ est égale à 0 pour tous les états du système, et à 1 pour les états $[(x_1,y_1), A_1]$ et $[(x_2,y_2), A_2]$ où (x_i,y_i) est la position de la cible i. Comme mentionné dans [62], la stratégie calculée garantit un plus court chemin vers les états buts : la stratégie tend à réduire le temps de la mission.

Les algorithmes pour π-PDMPO standards, qui n'exploitent pas l'observabilité mixte contrairement à notre modèle π-PDMOM, ne peuvent pas résoudre le problème même pour de très petites grilles 3×3. En effet, pour ce problème, $\#\mathcal{L} = 5$, $\#\mathcal{S}_v = 9$, et $\#\mathcal{S}_h = 2$. Ainsi, $\#\mathcal{S} = \#\mathcal{S}_v \cdot \#\mathcal{S}_h = 18$ et le nombre d'états de croyance est alors $\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \mathcal{L}^{\#\mathcal{S}} - (\mathcal{L}-1)^{\#\mathcal{S}} = 5^{18} - 4^{18} \geqslant 3.7.10^{12}$ au lieu de 81 états avec un π-PDMOM. Par conséquent, les résultats expérimentaux qui suivent n'auraient pas pu être obtenus avec des π-PDMPO standards, ce qui justifie donc ce travail sur les π-PDMOM.

La comparaison des performances des modèles probabilistes et possibilistes peut se faire à l'aide des espérances de la somme des récompenses de leurs stratégies respectives : puisque la situation est complètement connue lorsque le robot est à la position d'une des cibles, il ne peut pas terminer en choisissant la cible B. Si k est le nombre d'étapes du processus pour identifier et atteindre la bonne cible, alors la somme des récompenses est $100 - k$.

Considérons maintenant qu'en réalité (donc ici pour les simulations), et contrairement à ce qui est décrit par le modèle, la situation en pratique fait que l'algorithme de vision artificielle utilisé par le robot est trompeur lorsque le robot est loin des cibles, *i.e.* si $\forall i \in \{1,2\}$, $\sqrt{(x-x_i)^2+(y-y_i)^2} > C$, avec C une constante positive, alors $Pr(good_i \mid x,y) = 1 - P_{bad} <$

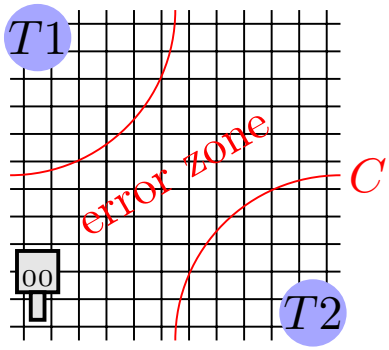


FIGURE II.2 – *Mission robotique de reconnaissance de cibles : deux cibles, la cible* 1 *(T1) et la cible* 2 *(T2) sont disposées sur une grille* $g \times g$*. Une des deux cibles est l'objet d'intérêt* A*, (et par élimination, l'autre cible est* B*) : soit* $T1 = A$*, soit* $T2 = A$*. Le robot reçoit des observations sur la nature de chaque cible : "*$Ti = A$*" ou "*$Ti = B$*" pour* $i \in \{1, 2\}$*, de manière bruitée. Plus le robot est proche d'une cible, moins l'observation à propos de la cible en question a de chances d'être fausse. Le robot doit reconnaître la cible* A *et l'atteindre. Cependant, sans avoir pu être pris en compte dans le modèle, lorsque le robot est dans une zone d'erreur (error zone en rouge) la probabilité qu'il observe mal* P_{bad} *est supérieure à* 0.5

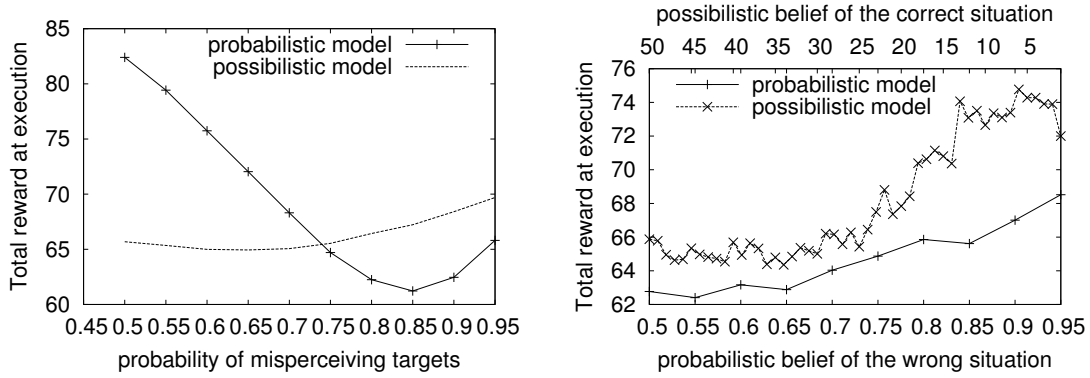
$\frac{1}{2}$. Dans tous les autres cas, le modèle probabiliste est bien décrit, *i.e.* en accord avec la réalité. La figure II.2 résume le problème, et indique la zone où le robot a une mauvaise perception par la dénomination "error zone". Pour les expérimentations numériques qui suivent, le nombre de simulations était de 10^4 pour calculer la moyenne de la somme des récompenses à l'exécution. La taille de la grille était de 10×10, $D = 10$ et $C = 4$.

La figure II.3.a montre que le modèle probabiliste est plus affecté par l'erreur introduite que le modèle possibiliste : elle représente la moyenne de la somme des récompenses à l'exécution obtenue par chaque modèle, comme une fonction de P_{bad}, la probabilité de mal observer une cible lorsque la position du robot est telle que $\sqrt{(x - x_i)^2 + (y - x_i)^2} > C$. Cela est dû au fait que la mise à jour possibiliste de l'état de croyance ne tient pas compte des nouvelles observations lorsque le robot en a déjà obtenu une plus fiable. Au contraire, le modèle probabiliste modifie l'état de croyance courant à chaque étape. En effet, puisqu'il n'y a que deux états cachés s_h^1 et s_h^2, si $\beta_h(s_h^1) < 1$, alors la normalisation possibiliste implique que $\beta_h(s_h^2) = 1$. La définition de la possibilité jointe de l'état et de l'observation (le minimum entre la distribution de possibilité sur l'état du système, *i.e.* l'état de croyance, et la possibilité de l'observation) assure que la possibilité jointe de s_h^1 et de l'observation obtenue, est plus petite que $\beta_h(s_h^1)$. L'équivalent possibiliste de l'équation de mise à jour de l'état de croyance (3) assure donc que l'état de croyance suivant se retrouve dans un des trois cas suivant :

- il est encore plus sceptique à propos de s_h^1 si l'observation est plus fiable, et confirme l'état de croyance précédent ($\pi\left(o_h \mid s_v, s_h^1, a\right)$ est plus petit que $\beta_h(s_h^1)$) ;
- il devient l'état de croyance opposé si l'observation est plus fiable et contredit l'état de croyance précédent ($\pi\left(o_h \mid s_v, s_h^2, a\right)$ est à la fois plus petit que $\beta_h(s_h^1)$ et que $\pi\left(o_h \mid s_v, s_h^1, a\right)$) ;
- il reste simplement le même si l'observation n'est pas plus informative que l'état de croyance courant.

Le théorème qui suit donne des conditions suffisantes menant à une mise à jour informative de l'état de croyance possibiliste. Classiquement un état de croyance $\beta_1 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ est dit plus spécifique qu'un état de croyance $\beta_2 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ si pour chaque $s \in \mathcal{S}$, $\beta_1(s) \leqslant \beta_2(s)$. La relation d'ordre $\preceq$ sur $\Pi_{\mathcal{L}}^{\mathcal{S}}$ peut alors être définie pour classer les états de croyance selon leur spécificité :

$$\beta_1 \preceq \beta_2 \Leftrightarrow \sum_{s \in \mathcal{S}} \beta_1(s) \leqslant \sum_{s \in \mathcal{S}} \beta_2(s)$$



(a) P_{bad} varie (b) β_0 varie, $P_{bad} = 0.8$

FIGURE II.3 – *Comparaison des moyennes de la somme des récompenses à l'exécution, pour les modèles probabilistes et possibilistes.*

Notons que si β_1 est plus spécifique que β_2, alors $\beta_1 \preceq \beta_2$.

Théoreme 7

Soit $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ l'état de croyance initial modélisant l'ignorance totale, i.e. pour tous les $s \in \mathcal{S}$, $\beta_0(s) = 1$. Si la fonction de transition est déterministe, et si les observations ne sont pas informatives, i.e. $\forall s' \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\forall o' \in \mathcal{O}$, $\pi(o' \mid s', a) = 1$, alors si l'état de croyance $\beta_{t+1} \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ est le résultat de la mise à jour de l'état de croyance $\beta_t \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, $\beta_{t+1} \preceq \beta_t$. Ce résultat reste valable si pour chaque action la fonction de transition est l'identité et $\forall (o', \tilde{o}) \in \mathcal{O}^2$, $\forall (a, \overline{a}) \in \mathcal{A}^2$ et $\forall s' \neq \tilde{s} \in \mathcal{S}$ t.q. $\pi(o' \mid s', a) < 1_{\mathcal{L}}$, $\pi(o' \mid s', a) \neq \pi(\tilde{o} \mid \tilde{s}, \overline{a})$.

La mise à jour probabiliste quant à elle ne permet pas à l'état de croyance de devenir directement l'état de croyance opposé, ou d'ignorer les observations moins fiables : le robot se dirige d'abord vers la mauvaise cible car il est initialement trop loin des deux cibles et les observe mal. Lorsqu'il est proche de cette cible, il reçoit de bonnes observations, et change petit à petit d'état de croyance : ce dernier devient assez informatif pour le convaincre de se diriger vers la cible A. Cependant, il passe alors inévitablement par la zone d'erreur : cela modifie peu à peu son état de croyance, qui devient faux avec grande probabilité, et le robot se retrouve dans la situation initiale. Il perd ainsi beaucoup de temps à sortir de cette boucle. On peut voir que la moyenne de la somme des récompenses croît lorsque la probabilité de mal observer, P_{bad}, est très grande : cela s'explique par le fait que cette grande erreur mène le robot à atteindre la mauvaise cible plus rapidement, et donc à être quasiment sûr que la cible A est l'autre cible.

Maintenant, fixons $P_{bad} = 0.8$ et évaluons la moyenne de la somme des récompenses à l'exécution pour différents faux états de croyance initiaux : la figure II.3.b illustre cette évaluation, avec les mêmes paramètres que pour la précédente expérimentation : nous comparons ici le modèle possibiliste, et la probabiliste lorsque l'état de croyance initial est fortement orientée vers la mauvaise cible (*i.e.* l'agent pense fortement que la cible 1 est B alors que c'est A en réalité). Notons que l'état de croyance possibiliste en la bonne cible décroît lorsque la nécessité en la mauvaise croît. Cette figure montre que le modèle possibiliste mène à de meilleures récompenses à l'exécution si l'état de croyance initial est mauvais et la fonction d'observation est imprécise : notons cependant que pour $\mathcal{P}_{bad} \leqslant 0.6$, la politique probabiliste est plus efficace [1].

1. L'implémentation de l'algorithme de résolution, ainsi que la description de ce problème de reconnaissance qui en est l'entrée, sont disponibles sur le dépôt accessible à l'adresse `https://github.com/drougui/ppudd` : le problème peut être simulé en utilisant la stratégie optimale possibiliste calculée par l'algorithme.

II.4 Conclusion

Nous avons proposé un algorithme d'Itération sur les Valeurs (IV) pour les PDM possibilistes. Celui-ci calcule une stratégie optimale stationnaire pour un horizon indéterminé contrairement aux méthodes précédentes. Une preuve complète de la convergence a été fournie : elle repose sur l'existence d'une action "rester" intermédiaire. Celle-ci est utilisée uniquement pour maintenir le processus dans les états buts. Enfin, le nouveau modèle des PDM possibilistes qualitatifs à observabilité mixte, a été présenté, et sa complexité est exponentiellement plus petite que celle des PDMPO possibilistes qualitatifs. De ce fait, nous avons pu comparer les π-PDMOM avec leurs équivalents probabilistes sur un problème robotique réaliste. Nos résultats expérimentaux montrent que ces stratégies possibilistes peuvent être plus performantes que les stratégies issues du modèle probabiliste lorsque la fonction d'observation n'est pas connue précisément.

La version de l'algorithme d'Itération sur les Valeurs pour π-PDM pessimiste peut aussi être construite, mais l'optimalité de la stratégie renvoyée semble dure à prouver. Les travaux [77] et [59] peuvent être utiles pour obtenir des résultats à propos de ces π-PDM pessimistes, afin de résoudre des problèmes contenant des situations dangereuses.

Finalement, comme cela a été mis en évidence par les expériences, bien que les π-PDMPO soient basés sur un modèle d'incertitude plus simple en termes de complexité que les PDMPO probabilistes, le processus de croyance peut avoir un comportement intéressant. Pour certaines conditions suffisantes données par le théorème 7, l'état de croyance n'est pas modifié par des informations moins fiables que celles accumulées avant elles, mais est capable de se transformer en un état de croyance quasi opposé si une information qui le suggère et qui est plus fiable est reçue. Des problèmes plus complexes doivent être étudiés pour avoir une meilleure idée de son comportement dans un panel de situations plus important. Cependant, les π-PDMPO avec un espace d'état trop grand (ou π-PDMOM avec un trop grand espace $\mathcal{S}_h$) ne peuvent pas être résolus en temps raisonnables par les algorithmes développés jusqu'à aujourd'hui. Le chapitre suivant présente et utilise d'autres structures du problème, décrites par l'homologue possibiliste qualitatif du modèle des *PDMPO factorisés* : ces structures mènent à des calculs de stratégies possibilistes plus simples, et permettent de résoudre de nombreux problèmes de planification.

III Développement d'Algorithmes Symboliques pour la Résolution des π-PDMPO

Dans ce chapitre, nous proposons l'étude des modèles π-PDMOM dans le but de résoudre de très grands problèmes de planification lorsqu'ils sont structurés. Inspirés par l'algorithme *Stochastic Planning Using Decision Diagrams* (*SPUDD*) construit pour résoudre les PDM probabilistes factorisés, nous avons mis en place un algorithme symbolique appelé PPUDD conçu pour résoudre les π-PDMOM. Tandis que le nombre de feuilles des arbres de décision utilisés par *SPUDD* peuvent devenir aussi grand que la taille de l'espace des états puisque leurs valeurs sont des nombres réels agrégés par des additions et des multiplications, le nombre de feuilles de *PPUDD* est borné par le nombre d'éléments dans $\mathcal{L}$ car leurs valeurs restent dans l'échelle finie $\mathcal{L}$ via les opérations min et max seulement. Enfin, nous présentons un π-PDMOM satisfaisant certaines hypothèses d'indépendance sur les variables visibles, cachées, et d'observation. Ce dernier résulte en un π-PDM factorisé, sur lequel *PPUDD* peut être lancé. Nos résultats expérimentaux montrent que le temps de calcul de *PPUDD* est beaucoup plus petit que *Symbolic-HSVI* et *APPL* pour les versions possibilistes et probabilistes du même benchmark, tout en fournissant des stratégies de bonne qualité. Les performances des stratégies calculées par *PPUDD* ont été testées pendant la compétition internationale de planification probabiliste (*IPPC* 2014) dont les résultats sont exposés ici.

III.1 Introduction

Les travaux sur les π-PDM(MO) présentés précédemment ne tirent pas totalement avantage de la structure du problème, *i.e.* les parties visibles ou cachées de l'état peuvent être elles-même factorisées en plusieurs variables d'états. Dans le cadre probabiliste, les PDM factorisés et les méthodes de calcul symboliques [11, 36] ont été étudiés intensivement dans le but de raisonner directement au niveau des variables plutôt que sur l'espace d'état. Un travail récent sur ces problématiques est par exemple [60]. Le célèbre algorithme *SPUDD* [36] résout des PDM factorisés en utilisant des représentations symboliques de la fonction valeur et des stratégies sous la forme d'arbres de décision algébriques (*ADDs*) [3], qui représentent de manière compacte les fonctions réelles de variables booléennes : les *ADDs* sont des arbres dont les noeuds représentent les variables d'état et les feuilles sont les valeurs de la fonction. Au lieu de mettre à jour les valeurs de la fonction pour chaque état individuellement à chaque itération de l'algorithme, ils sont aggrégés dans les *ADDs* et les opérations sont effectuées de manière symbolique directement entre les *ADDs* sur plusieurs variables en même temps. Cependant, *SPUDD* est limité par la manipulation potentielle d'énormes *ADDs* dans le pire des cas : par exemple, l'espérance implique des additions et des multiplications sur des valeurs

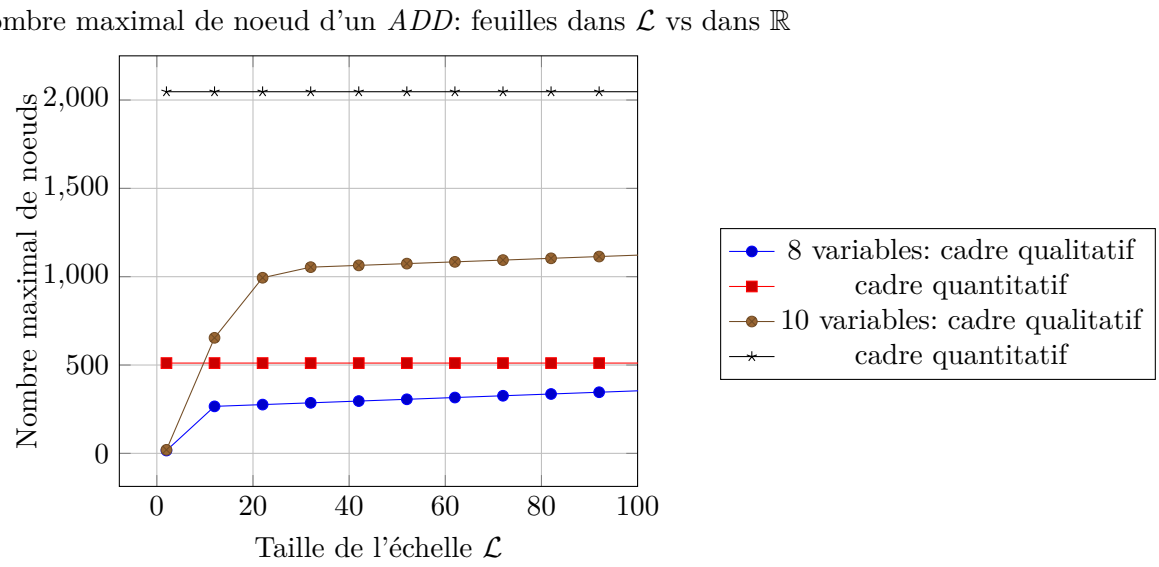


FIGURE III.1 – *La taille maximale (nombre total de noeuds) d'un ADD dont les valeurs sont dans $\mathcal{L}$ est limité : cette taille maximale est représentée par les courbes avec les ronds bleus et marrons, en fonction de la taille de $\mathcal{L}$. Lorsque les feuilles des ADDs sont dans $\mathbb{R}$, le nombre de ses noeuds est potentiellement exponentiel en le nombre de variables : la borne supérieure est représentée par les courbes avec les carrés rouges et noirs (fonctions constantes de la taille de $\mathcal{L}$).*

réelles (probabilités et récompenses), créant de nouvelles valeurs entres elles, de manière à ce que le nombre de feuille des *ADDs* puisse devenir égal à l'espace d'état, *i.e.* exponentiel en le nombre de variables d'état.

Ainsi, le travail présenté ici est motivé par la simple observation que les **opérations symboliques avec des PDM possibilistes devraient nécessairement limiter la taille des *ADDs*** : en effet, ce formalisme opère sur une échelle *finie* $\mathcal{L}$ avec seulement les opérations max et min, ce qui implique que les valeurs manipulées restent dans l'échelle finie $\mathcal{L}$, qui est généralement beaucoup plus petite que le nombre d'états.

La figure III.1 montre que les *ADDs* utilisés dans le cadre possibiliste a un nombre limité de noeuds puisque le nombre de feuilles est au plus égal à la taille de l'échelle possibiliste qualitative $\mathcal{L}$: la taille maximale (nombre maximal de noeuds) d'un *ADD* dont les feuilles sont dans $\mathcal{L}$, est représenté comme une fonction de $\#\mathcal{L}$, dans le cas de 8 et 10 variables.

Dans ce chapitre, nous présentons un algorithme basé sur la programmation dynamique symbolique pour résoudre les π-PDMOM factorisés appelé Possibilistic Planning Using Decision Diagram (*PPUDD*). Cette contribution seule n'est pas suffisante puisque les variables de croyance ont un nombre de valeurs exponentiel en la taille de l'espace des états cachés. Donc, notre seconde contribution est un théorème visant à factoriser l'état de croyance en de nombreuses variables de croyance marginales lorsque certaines hypothèses d'indépendance sur les variables d'état et d'observation d'un π-PDMOM sont vérifiées : cela permet de résoudre certains problèmes dont les calculs sont inabordable. Notons que notre idée de factorisation de l'état de croyance est assez général pour être valable pour les modèles probabilistes. Enfin, les performances de *PPUDD* sont comparées à celles de *symbolic HSVI* [69] (une version symbolique de l'algorithme pour PDMPO appelé *HSVI* [71]) et *APPL* [43, 52] (déjà utilisé dans le chapitre précédent, et basé sur *SARSOP*) sous observabilité mixte. Les résultats obtenus étant prometteurs, nous avons participé à la compétition internationale de planification probabiliste (*IPPC* 2014) et les résultats de *PPUDD* durant la session entièrement observable d'*IPPC* 2014 sont présentés et discutés. Un algorithme dédié à la résolution des π-PDMOM en utilisant des *ADDs* est disponible dans le dépôt `https://github.com/drougui/ppudd`).

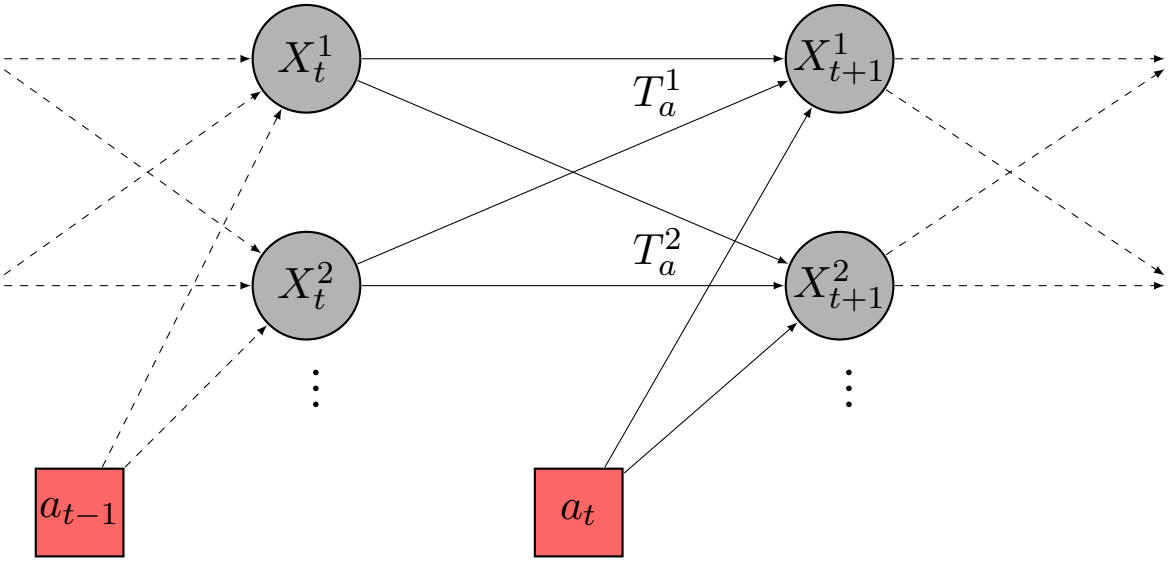


FIGURE III.2 – *Réseau dynamique bayésien pour des (π-)PDM : dans le cadre possibiliste (resp. probabiliste)* T_a^i *est la distribution de possibilité (resp. probabilité) de transition sur la variable d'état* X_{t+1}^i *conditionnellement à l'action sélectionnée* $a \in \mathcal{A}$ *et à ses parents* $parents(X_{t+1}^i) \subseteq \{X_t^1, \dots, X_t^n\}$ *(i.e.* $parents(X_{t+1}^i)$ *est un sous ensemble de l'ensemble des variables d'état courantes) où* $n \geqslant 1$ *est le nombre de variables décrivant l'espace d'état.*

III.2 Résoudre des π-PDMOM par la Programmation Dynamique Symbolique

Les PDM factorisés [36] ont été utilisés pour résoudre plus rapidement les problèmes structurés de décision séquentielle, sous incertitude probabiliste, en raisonnant symboliquement sur les fonctions des états du systèmes, à travers des arbres de décision algébriques. Inspiré par ce travail, cette section présente la résolution symbolique des π-PDMOM factorisés : dans ce modèle, l'espace des états visibles $\mathcal{S}_v$, l'espace des états cachés $\mathcal{S}_h$ et l'ensemble des observations $\mathcal{O}_h$ sont tels que l'espace d'état du π-PDM résultant (basé sur l'espace des états de croyances et l'espace des variables visibles) est sous la forme $\mathcal{S}_v^1 \times \cdots \times \mathcal{S}_v^m \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, où chacun de ces espaces sont finis. Nous verrons dans la section suivante comment $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ peut être factorisé de cette manière grâce à la factorisation de $\mathcal{S}_h$ et $\mathcal{O}_h$. La factorisation de la variable de la croyance probabiliste dans [12, 67] est approximative, tandis que celle présentée ici est exacte. Puisque les espaces finis de taille K peuvent être eux-même factorisés en $\lceil \log_2 K \rceil$ espaces binaires [36], nous pouvons faire l'hypothèse que nous raisonnons sur un π-PDM dont l'espace d'état est noté $\mathcal{X}$ et entièrement décrit par les variables $(X^1, \dots, X^n)$, avec $n \in \mathbb{N}^*$ et $\forall i$, $X^i \in \{\top, \bot\}$: $\mathcal{X} = \{\top, \bot\}^n$.

Rappelons que les réseaux bayésiens dynamiques (*DBNs*) [21] déjà utilisés en section I.4 (dans le diagramme d'influence figure I.4) et dans le chapitre précédent (figure II.1 illustrant la structure d'observabilité mixte) sont des représentations graphique très utiles pour les processus étudiés. Un *DBN* représentant la structure d'un π-PDM factorisé est dessiné dans la figure III.2 : les variables d'état à une étape de temps donnée $t \geqslant 0$ sont notées $X_t = (X_t^i)_{i=1}^n$ (variables courantes), et $(X_{t+1}^i)_{i=1}^n$ sont les variables d'état à l'étape de temps $t+1$ (variable suivante). Dans le cadre des *DBNs*, $parents(X_{t+1}^i)$ est l'ensemble des variables d'état dont la variable d'état suivante X_{t+1}^i dépend, *i.e.* une variable Y, représentée par un noeud dans le *DBN*, est dans $parents(X_{t+1}^i)$ si et seulement si il y a une flèche de Y à X_{t+1}^j. Nous supposons que $parents(X_{t+1}^i) \subseteq \{X_t^1, \dots, X_t^n\}$, *i.e.* les parents de la variable d'état suivante X_{t+1}^i font

partie des variables d'état courantes $\{ X_t^1, \ldots, X_t^n \}$: il ne peut y avoir de flèches entre les variables d'état de la même étape de temps.

Avec les notations du π-PDMOM, les hypothèses du réseau bayésien de la figure III.2 nous permettent de calculer la distribution de possibilité jointe : $\pi(s'_v, \beta'_h \mid s_v, \beta_h, a) = \pi(X' \mid X, a) = \min_{i=1}^{n} \pi(X'_i \mid parents(X'_i), a)$, où, étant donné l'étape de temps t, les variables primées sont les variables concernant l'étape de temps $t+1$ (variables suivantes), et les variables non-primées sont les variables courantes (à l'étape de temps t) : par exemple, X'_i est la notation pour X_{t+1}^i, et X_i celle pour X_t^i. Ainsi, un π-PDMOM factorisé peut être défini par des fonctions de transition $T_a^i = \pi(X'_i \mid parents(X'_i), a)$ pour chaque action a et chaque variable X'_i (si les transitions sont stationnaires).

Chaque fonction de transition peut être représentée par un arbre de décision algébrique (*ADD*) [3]. Un *ADD*, comme illustré dans la figure III.3a, est un arbre représentant de manière compacte une fonction réelle de variables binaires, dont les sous-graphes identiques sont confondus et les feuilles valant zéro ne sont pas mémorisées. Les notations suivantes sont utilisées pour rendre explicite le fait que nous travaillons avec des fonctions symboliques représentées par des *ADDs* :

— $\boxed{\min}\{f, g\}$ où f et g sont deux *ADDs* ;

— $\boxed{\max}_{X_i} f = \boxed{\max}\left\{ f^{X_i=0}, f^{X_i=1} \right\}$,

qui peut être facilement calculé car les *ADDs* sont construits sur la base de l'expension de Shanon : $f = \overline{X_i} \cdot f^{X_i=0} + X_i \cdot f^{X_i=1}$ où $f^{X_i=1}$ et $f^{X_i=0}$ sont les sous graphes représentant les cofacteurs de Shanon positifs et négatifs (*cf.* figure III.3a).

Le schéma de programmation dynamique, *i.e.* la ligne 8 de l'algorithme d'itération sur les valeurs 3 du chapitre précédent, peut être réécrite sous forme symbolique, de telle sorte que les états soient globalement mis à jour en un coup, plutôt qu'individuellement : en notant $X = (X_1, \ldots, X_n)$ la variable d'état courante et $X' = (X'_1, \ldots, X'_n)$ la suivante, la Q-valeur pour une action $a \in \mathcal{A}$ est $\overline{q^a} = \overline{q^a}(X) = \boxed{\max}_{X'} \boxed{\min}\left\{ \pi(X' \mid X, a), \overline{U^*}(X') \right\}$. Le calcul de cet *ADD* ($\overline{q^a}$) peut être décomposé en plusieurs calculs en utilisant la proposition suivante :

Propriété III.2.1 (*Régression possibiliste de la fonction valeur*)

Considérons la fonction valeur courante $\overline{U^*} : \{\top, \bot\}^n \to \mathcal{L}$. *Pour une action donnée* $a \in \mathcal{A}$, *définissons :*

— $\overline{q_0^a} = \overline{U^*}(X'_1, \cdots, X'_n)$,

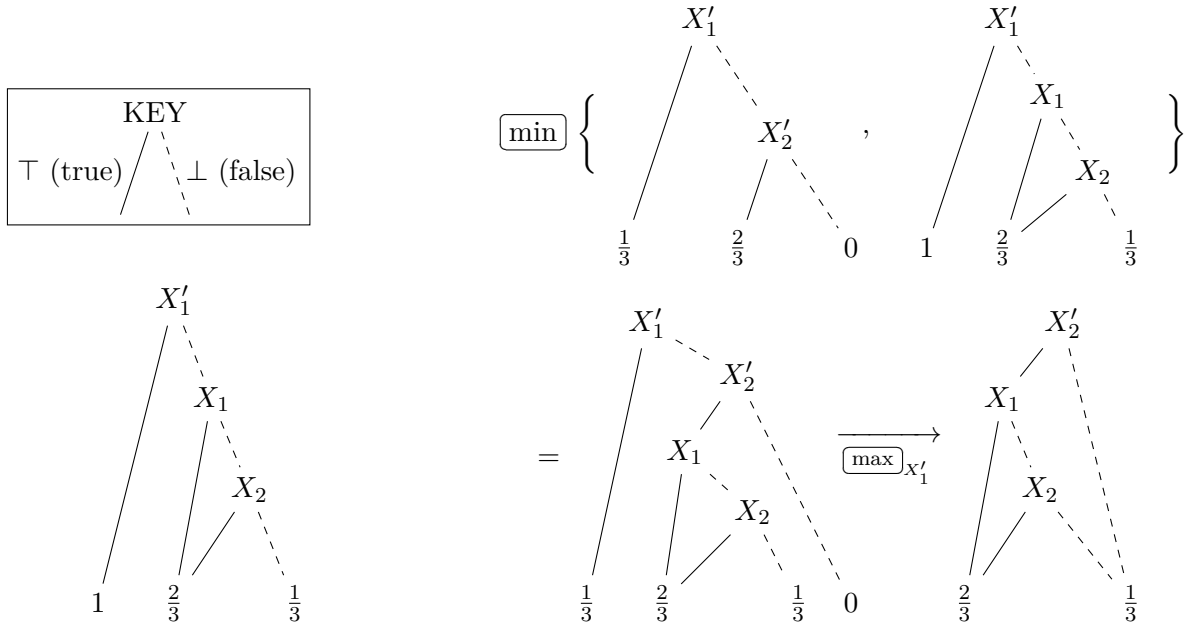
— $\overline{q_i^a} = \max_{X'_i \in \{\top, \bot\}} \min\left\{ \pi(X'_i \mid parents(X'_i), a), \overline{q_{i-1}^a} \right\}$.

Alors, la Q-valeur possibiliste d'une action a *est* $\overline{q^a} = \overline{q_n^a}$, *qui dépend des variables* $X_1, \ldots, X_n$, *et la fonction valeur suivante est* $\overline{U^*}(X_1, \ldots, X_n) = \max_{a \in \mathcal{A}} \overline{q_n^a}(X_1, \ldots, X_n)$.

La Q-valeur de l'action a, représentée par un *ADD*, peut être alors calculée en plusieurs étapes, une pour chaque variable d'état suivante $X'_i, 1 \leqslant i \leqslant n$. La figure III.3b illustre les calculs possibilistes effectués entre les arbres de décision algébriques (*ADDs*) pour calculer la Q-valeur d'une action.

L'algorithme 4 est une version symbolique de l'algorithme d'itération sur les valeurs pour π-PDM dans le chapitre précédent, algorithme 3), qui utilise le schéma de régression défini dans la proposition III.2.1. Inspiré de *SPUDD* [36], *PPUDD* signifie *Possibilistic Planning Using Decision Diagrams*. Comme pour *SPUDD*, l'action de transformer les variables non primées en variables primées est nécessaire à chaque itération (*cf.* ligne 5 de l'algorithme 4 et figure III.3b). Cette opération sert à différentier les états suivants des états courants lors des opérations entre les ADDs. Les lignes 4-9 appliquent la proposition III.2.1 et correspondent à la ligne 8 de l'algorithme 3.

Nous avons mentionné au début de la section que l'espace des états de croyance $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ pourrait être décrit par $\lceil \log_2 K \rceil$ variables binaires où $K = \#\mathcal{L}^{\#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_h}$. Cependant,



(a) *ADD représentant T_1^a de la figure III.2*

(b) *Calcul symbolique de la Q-valeur courante sous forme d'un ADD : elle est combinée à la fonction de transition représentée par un ADD dans la figure III.3a*

FIGURE III.3 – *Exemple d'arbre de décision algébrique comme utilisé dans l'algorithme PPUDD*

Algorithm 4: PPUDD (calcul pour un horizon indéterminé)

$\overline{U^*} \leftarrow 0$; $\overline{U^c} \leftarrow \Psi$; $\overline{\delta} \leftarrow \widehat{a}$;
while $\overline{U^*} \neq \overline{U^c}$ **do**
  $\overline{U^*} \leftarrow \overline{U^c}$;
  for $a \in \mathcal{A}$ **do**
    $\overline{q^a} \leftarrow$ swap each X_i variable in $\overline{U^*}$ with X_i' ;
    for $1 \leqslant i \leqslant n$ **do**
      $\overline{q^a} \leftarrow \boxed{\min}\left\{\overline{q^a}, \pi\left(X_i' \,\middle|\, parents(X_i'), a\right)\right\}$;
      $\overline{q^a} \leftarrow \boxed{\max}_{X_i'} \overline{q^a}$;
    $\overline{U^c} \leftarrow \boxed{\max}\left\{\overline{q^a}, \overline{U^c}\right\}$;
    update $\overline{\delta}$ to a where $\overline{q^a} = \overline{U^c}$ and $\overline{U^c} > \overline{U^*}$;
return $\overline{U^*}$, $\overline{\delta^*}$;

ce K peut être très grand, ainsi nous proposons dans la section qui suit, une méthode pour exploiter la factorisation de $\mathcal{S}_h$ et $\mathcal{O}_h$ dans le but de factoriser $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ en plusieurs variables, ce qui décomposera les *ADDs* de transition en plus petits *ADDs* facilement manipulables. Notons que *PPUDD* peut résoudre les π-PDMOM même si cette factorisation de la croyance n'est pas possible, mais les *ADDs* manipulés sont plus gros dans ce cas.

Le chapitre précédent met en évidence qu'un prétraitement est nécessaire pour traduire un π-PDMOM en un π-PDM dont l'espace d'état est $\mathcal{X}$. Nous pouvons alors raisonner sur l'espace d'état entièrement visible par l'agent $\mathcal{X} = S_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ et résoudre le π-PDMOM en tant que π-PDM. La section suivante fait le lien entre les propriétés structurelles d'un π-PDMOM, concernant les dépendences des variables originales (visibles, cachées et d'observation), à la factorisation du problème traité *i.e.* du π-PDM résultant, défini sur l'espace d'état $S_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$: la factorisation résultante concerne alors les dépendances des variables visibles et des variables de croyance.

III.3 Factorisation de la Variable de Croyance d'un π-PDMOM

La factorisation de la variable de croyance requiert trois hypothèse d'indépendance sur les variables du π-PDMOM, qui sont illustrées à travers le problème classique *Rocksample* [71].

III.3.1 Exemple de Motivation

Un robot se déplaçant sur une grille $g \times g$ doit collecter des échantillons scientifiques à partir de pierres dites intéressantes parmi R pierres, et enfin d'atteindre la sortie de la grille. Il connait les positions des R pierres $(x_i, y_i)_{i=1}^{R}$ mais pas lesquelles sont intéressantes. Les pierres intéressantes sont appelées les "bonnes" pierres. Cependant, échantillonner une pierre coûte cher : le robot est donc équipé d'un capteur qu'il peut utiliser pour déterminer si une pierre est "bonne" ou non ("mauvaise"). Lorsqu'une pierre est échantillonnée, elle devient (ou reste) "mauvaise" (plus intéressante scientifiquement). A la fin de la mission, le robot doit atteindre la position de sortie de la grille. Décrivons le PDMOM associé :

- $\mathcal{S}_v$ représente l'ensemble des positions possibles du robot en plus de la sortie ($\#\mathcal{S}_v = g^2 + 1$) ;
- $\mathcal{S}_h$ représente l'ensemble des natures possibles de chacune des pierres : $\mathcal{S}_h = \mathcal{S}_h^1 \times \ldots \times \mathcal{S}_h^R$, avec $\forall 1 \leqslant i \leqslant R$, $\mathcal{S}_h^i = \{\text{“}bonne''\text{,“}mauvaise''\}$;
- $\mathcal{A}$ contient les déplacements (déterministes) dans les 4 directions $(a_{north}, a_{east}, a_{south}, a_{west})$, tester la pierre numéro i, (a_{check_i}) $\forall 1 \leqslant i \leqslant R$, et échantillonner la pierre courante, (a_{sample}) ;
- $\mathcal{O} = \{o_{good}, o_{bad}\}$ sont les différentes réponses possibles des capteurs pour la pierre testée.

La dynamique des observations est la suivante : plus le robot est proche de la pierre testée, mieux il observe sa nature. Dans le problème probabiliste original, la probabilité d'une observation correcte est égale à $\frac{1}{2}\left(1 + e^{-c\sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}}\right)$ avec $c > 0$, une constante (plus c est petit, plus le capteur est efficace). Le robot reçoit la récompense $+10$ (resp. -10) pour chaque bonne (resp. mauvaise) pierre échantillonnée, et $+10$ lorsqu'il atteint la sortie de la grille.

Dans cadre possibiliste, la fonction d'observation est approximée en utilisant une distance critique au-delà de laquelle tester une pierre n'est pas informatif : $\pi(o_i' \mid s_i', a, s_v) = 1\ \forall o_i' \in \mathcal{O}_i$. Le degré de possibilité d'une observation erronée devient zéro lorsque le robot se trouve sur la pierre testée, et devient le plus petit degré de possibilité non nul sinon. Enfin, puisque la sémantique possibiliste ne permet pas de sommer les récompenses, une variable additionnelle

$s_v^2 \in \{1, \ldots, R\}$ est introduite : elle compte le nombre de pierres testées. La préférence qualitative de l'échantillonage est définie par $\Psi(s) = \frac{R+2-s_v^2}{R+2} \in \mathcal{L}$ si la position est la sortie, et zéro sinon. Enfin, la position du robot est notée $s_v^1 \in \mathcal{S}_v^1$ et donc la variable d'état visible est finalement $s_v = (s_v^1, s_v^2) \in \mathcal{S}_v^1 \times \mathcal{S}_v^2 = \mathcal{S}_v$.

Les observations $\{o_{good}, o_{bad}\}$ pour la pierre testée peuvent être modélisées de manière équivalente comme le produit cartésien des observations $\{o_{good_1}, o_{bad_1}\} \times \cdots \times \{o_{good_R}, o_{bad_R}\}$ pour chaque pierre. En utilisant cette modélisation, les espaces d'états et d'observations sont respectivement factorisés de la manière suivante, $\mathcal{S}_v^1 \times \ldots \times \mathcal{S}_v^m \times \mathcal{S}_h^1 \times \ldots \times \mathcal{S}_h^l$, et $\mathcal{O} = \mathcal{O}^1 \times \ldots \times \mathcal{O}^l$, et nous pouvons maintenant associer une variable d'observation $O^j \in \mathcal{O}^j$ à la variable d'état cachée correspondante $S_h^j \in \mathcal{S}_h^j$. Cela permet de raisonner sur le *DBN* de la figure III.4, qui exprime trois hypothèses importantes qui nous permettrons de factoriser l'état de croyance :

1. toutes les variables d'état $S_v^1, S_v^2, \ldots, S_v^m, S_h^1, S_h^2, \ldots, S_h^l$ sont indépendantes post-action, et une variable d'état suivante ne dépend par des variables d'état cachées courantes.
2. une variable d'état cachée ne dépend pas d'autres variables d'état cachées précédentes : par exemple la nature d'une pierre est indépendante de la nature des autres pierres.
3. une variable d'observation est disponible pour chaque variable d'état cachée, et dépend de cet état. Elle ne dépend pas d'autres variables d'état cachées, où des variables visibles courantes, mais des variables visibles précédentes et de l'action choisie :

Chaque variable d'observation est en effet seulement associée à la nature de la pierre correspondante. La qualité de l'observation dépend de la position du robot *i.e.* une variable d'état visible courante, ce qui n'est pas autorisé par le *DBN* : heureusement, puisque les déplacements sont déterministes, nous contournons le problème en considérant que cette qualité dépend de la position précédente et de l'action choisie, ce qui est équivalent ici.

III.3.2 Conséquences des Hypothèses de Factorisation

Dans cette section, nous présentons le résultat formel provenant des trois précédentes hypothèses : ce résultat est la factorisation de $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ en le produit cartésien $\bigtimes_{j=1}^{l} \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$. En effet, l'état de croyance β_h à propos de l'état caché du système $s_h \in \mathcal{S}_h$ peut être représenté des croyances marginales $\beta_h^j \in \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ sur les états cachés $s^j \in \mathcal{S}_h^j$, $\forall j \in \{1, \ldots, l\}$.

Le *critère de d-Séparation* [74] qui permet de montrer graphiquement des résultats d'indépendance à partir du *DBN*, se cache derrière les résultats fournis. Comme expliqué en section III.2, un *DBN* peut être construit à partir de relations d'indépendances. Notons $X \perp\!\!\!\perp Y \mid Z$ l'assertion "X est indépendant de Y conditionnellement à Z" : rappelons que pour une définition donnée de la relation d'indépendance, par exemple l'indépendance probabiliste, ou l'indépendance possibiliste causale, le *DBN* est construit de telle sorte que pour chaque variable X, $X \perp\!\!\!\perp \mathit{nondescend}(X) \mid \mathit{parents}(X)$, où $\mathit{nondescend}(X)$ est l'ensemble des variables qui ne sont pas des descendants de X. Si l'indépendance utilisée obéit aux axiomes des *semi-graphoides* [55, 75], le critère graphique appelé d-Séparation peut être utilisé pour identifier certaines indépendances sur le *DBN*. Ce critère est utilisé, par exemple, dans le cadre des probabilités dans le travail [78].

Tout d'abord, comme le *DBN* de la figure III.4 représente des hypothèses d'indépendance causales, la distribution de possibilité sur la variable d'état visible numéro i $s_{v,t+1}^i \in \mathcal{S}_v^i$ sachant les variables précédentes peut s'écrire :

$$\pi\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right) = \Pi\left(S_{v,t+1}^i = s_{v,t+1}^i \,\middle|\, S_{v,t} = s_{v,t}, a_t\right); \tag{III.1}$$

Définissons l'information i_t connue par l'agent à l'étape de temps $t \geqslant 1$ lorsque le modèle est un (π-)PDMOM : $i_0 = \{ s_{v,0} \}$, et pour chaque étape de temps $t \geqslant 1$, $i_t = \{ o_t, s_{v,t}, a_{t-1}, i_{t-1} \}$. La variable correspondante est notée I_t. Le théorème suivant assure que l'état de croyance courant peut être décomposé en états de croyance marginaux dépendant de l'information courante.

Théoreme 8 (*Indépendance des variables d'état cachées sachant i_t*)

Considérons un π-PDMOM décrit par le DBN de la figure III.4. Si les variables d'état cachées initiales $S^1_{h,0}, \ldots, S^l_{h,0}$ sont indépendantes, alors à chaque étape de temps $t > 0$ l'état de croyance sur les états cachés peut s'écrire

$$\beta_{h,t} = \min_{j=1}^{l} \beta^j_{h,t}$$

avec $\forall s \in \mathcal{S}^j_h$, $\beta^j_{h,t}(s) = \Pi\left(S^j_{h,t} = s \,\middle|\, I_t = i_t \right)$ l'état de croyance marginal concernant les états cachés du système de l'ensemble $\mathcal{S}^j_h$.

Grâce au théorème précédent, l'espace d'état visible par l'agent peut se réécrire $\mathcal{S}^1_v \times \ldots \times$

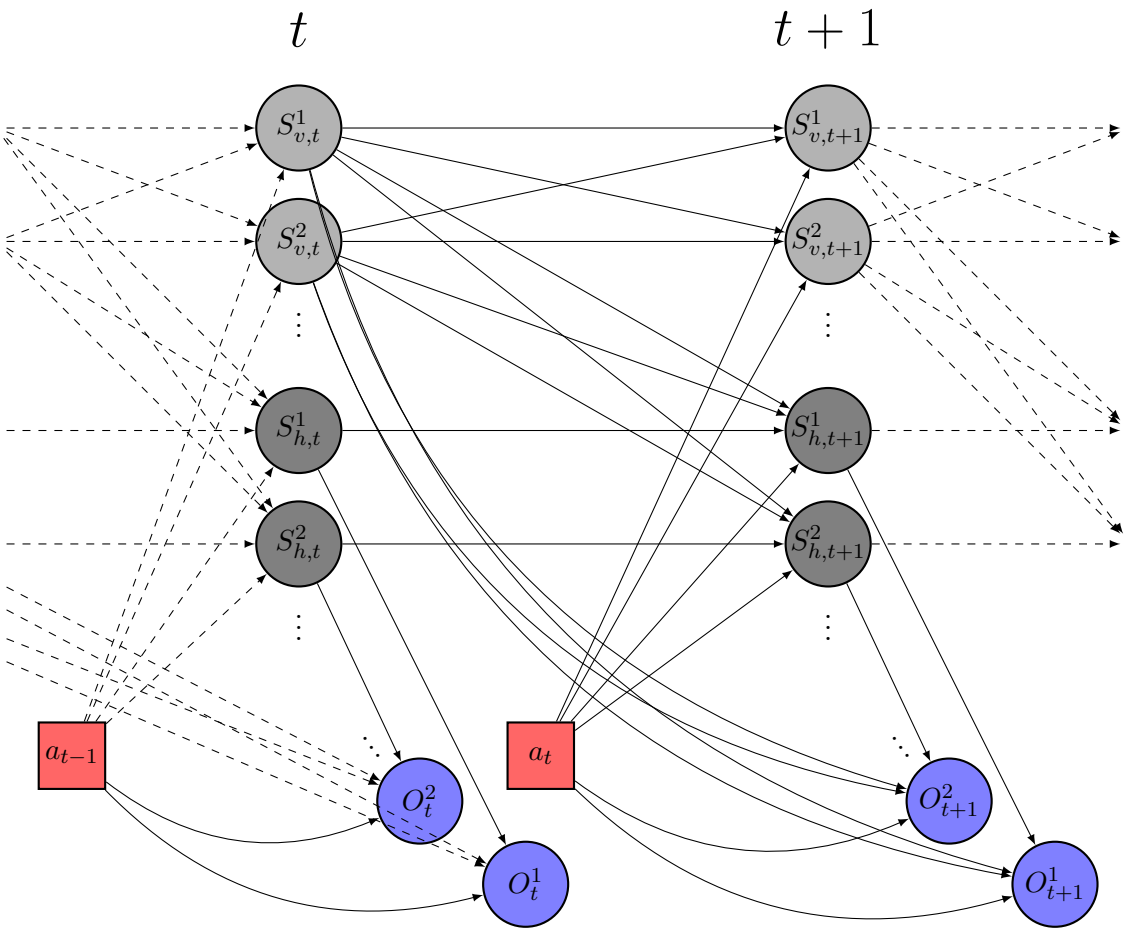


FIGURE III.4 – *DBN résumant les hypothèses d'indépendance d'un π-PDMOM menant à des variables de croyances marginales et un π-PDM avec une fonction de transition factorisée. Les parents d'une variable d'état visible sont les variables d'état visibles précédentes. Les parents d'une variable d'état cachée sont les variables d'état visibles précédentes et la variable d'état cachée précédente correspondante. Enfin, les parents d'une variable d'observation sont les variables d'état visibles précédentes, et la variables d'état cachée courante correspondante.*

$\mathcal{S}_v^m \times \Pi_{\mathcal{L}}^{\mathcal{S}_h^1} \times \cdots \times \Pi_{\mathcal{L}}^{\mathcal{S}_h^l}$ avec $\Pi_{\mathcal{L}}^{\mathcal{S}_h^j} \subsetneq \mathcal{L}^{\mathcal{S}_h^j}$. La taille de $\Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ est $\#\mathcal{L}^{\#\mathcal{S}_h^j} - (\#\mathcal{L}-1)^{\#\mathcal{S}_h^j}$ (*cf.* équation I.19). Si toutes les variables d'état sont binaires, $\#\Pi_{\mathcal{L}}^{\mathcal{S}_h^j} = 2\#\mathcal{L} - 1$ pour tout $1 \leqslant j \leqslant l$, ainsi $\#\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h} = 2^m(2\#\mathcal{L}-1)^l$: contrairement au cadre probabiliste, **les états cachés et les états visibles ont un impact similaire sur la complexité de résolution**, *i.e.* tous les deux simplement exponentiels en le nombre de variables d'état. Dans le cas général, en notant $\kappa = \max\{\max_{1\leqslant i\leqslant m} \#S_{v,i}, \max_{1\leqslant j\leqslant l} \#S_{h,j}\}$, il y a $\mathcal{O}(\kappa^m(\#\mathcal{L})^{(\kappa-1)l})$ états de croyances, ce qui est exponentiel en l'arité des variables d'état.

La **fonction de mise à jour de l'état de croyance marginal** est ν^j :

$$\beta_{h,t+1}^j = \nu^j(s_{v,t}, \beta_{h,t}^j, a_t, o_{t+1}^j),$$

Cette mise à jour peut se noter

$$\beta_{h,t+1}^j(s_{h,t+1}^j) \propto^\pi \pi\left(o_{t+1}^j, s_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)$$

puisqu'elle normalise au sens possibiliste la distribution de possibilité jointe sur la variable d'état cachée et la variable d'observation qui ont le numéro j.

Ainsi, le degré de possibilité que la variable de croyance marginale $B_{h,t+1}^{\pi,j}$ soit égale à $\beta_{h,t+1}^j \in \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ conditionnellement à $B_{h,t}^{\pi,j} = \beta_{h,t}^j$ et l'action $a_t \in \mathcal{A}$, peut se calculer :

$$\Pi\left(B_{h,t+1}^{\pi,j} = \beta_{h,t+1}^j \;\middle|\; S_{v,t} = s_{v,t}, B_{h,t}^{\pi,j} = \beta_{h,t}^j, a_t\right) = \max_{\substack{o^j \in \mathcal{O}^j \text{ s.t.} \\ \nu^j(s_{v,t},\beta_{h,t}^j,a_t,o^j)=\beta_{h,t+1}^j}} \pi\left(o^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right) \tag{III.2}$$

définissant la distribution de possibilité de transition des états de croyance marginaux $\pi\left(\beta_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)$.

Enfin, le théorème 9 nous assure que les variables du π-PDM résultant d'un tel π-PDMOM sont indépendantes post-action conditionnellement à l'état courant du système : cela permet alors d'écrire la fonction de transition de ce π-PDM sous forme factorisée :

Théoreme 9 (*Expression factorisée de la distribution de possibilité de transition*)

Si les hypothèses d'indépendance d'un π-PDMOM sont décrites par le DBN de la figure III.4, alors $\forall \beta_{h,t} = (\beta_{h,t}^1, \ldots, \beta_{h,t}^l) \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}, \beta_{h,t+1} = (\beta_{h,t+1}^1, \ldots, \beta_{h,t+1}^l) \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, $\forall (s_{v,t}, s_{v,t+1}) \in (\mathcal{S}_v)^2$, $\forall a_t \in \mathcal{A}$,

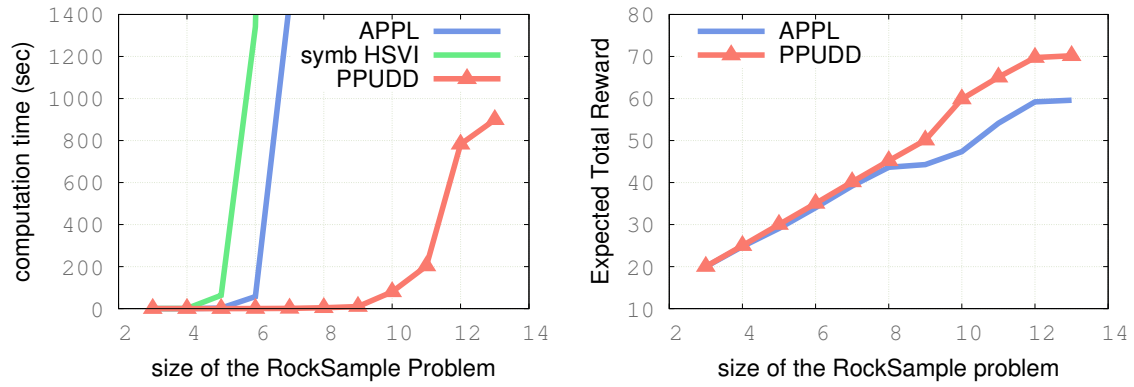
$$\pi\left(s_{v,t+1}, \beta_{h,t+1} \;\middle|\; s_{v,t}, \beta_{h,t}, a\right)$$

$$= \min\left\{\min_{i=1}^{m} \pi\left(s_{v,t+1}^i \;\middle|\; s_{v,t}, a_t\right), \min_{j=1}^{l} \pi\left(\beta_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)\right\},$$

où la distribution de possibilité de transition des variables d'état visibles est donnée par l'équation (III.1) et celle des variables de croyance marginales par l'équation (III.2).

En utilisant ce résultat, une telle expression factorisée de la distribution de possibilité de transition permet le calcul d'une fonction valeur avec $n = m + l$ étapes, comme décrit dans la section précédente : le π-PDMPO est en effet un π-PDM factorisé puisque les variables $(S_v^1, \ldots, S_v^m, B_h^{\pi,1}, \ldots, B_h^{\pi,l})$, peuvent jouer le rôle des variables $X_1, \ldots, X_n$ dans l'algorithme 4.

Notons que la relation d'indépendance probabiliste satisfait les axiomes des semi-graphoides : ainsi, les résultats d'indépendance dus à la d-Séparation sont aussi vrais pour le cadre probabiliste. Si les hypothèses d'indépendance d'un PDMOM probabiliste [52, 2] sont décrites par le *DBN* de la figure III.4, alors une factorisation similaire peut être déduite.



(a) *Temps de calcul (secondes)* (b) *Récompense totale espérée*

FIGURE III.5 – *Comparaison entre PPUDD et les planificateurs probabilistes APPL et symb-HSVI sur le problème RockSample : l'axe des abscisses représente l'indice de l'instance du problème qui est croissant avec la complexité de instance du problème.*

Le PDM construit à partir d'un tel PDMOM probabiliste est donc un PDM factorisé, dont l'espace d'état est infini.

Les théorèmes précédents permettent d'écrire la fonction de transition d'un (π-)PDM résultant d'un (π-)PDMOM avec des distributions qui concernent moins de variables. La mise à jour de la fonction valeur durant la programmation dynamique est alors divisée en $n = m + l$ étapes dans le cas possibiliste, comme décrit par la boucle *for* de l'algorithme 4. Cette boucle permet de manipuler des *ADDs* qui ont moins de noeuds ce qui rend les calculs plus rapides en général. Ces résultats sont utilisés dans la section suivante afin de calculer de manière plus efficace les stratégies optimales d'un π-PDMOM structuré.

III.4 Résultats Expérimentaux

Les calculs sont beaucoup plus simples dans le cadre possibiliste, mais il faut évidemment en payer le prix : les modèles possibilistes peuvent être considérés comme des approximations de modèles probabilistes. Comme montré dans cette section, une approximation possibiliste en utilisant *PPUDD* peut mener à de meilleures stratégies, au sens probabiliste, que celles calculées par certains algorithmes probabilistes lorsque les dimensions du problème considéré rendent les calculs insurmontables.

III.4.1 Missions Robotiques

Nous avons comparé *PPUDD* sur le problème *Rocksample* (RS), décrit en section III.3.1, contre un récent planificateur probabiliste pour PDMOM, *APPL* [52], et un planificateur pour PDMPO utilisant des *ADDs*, *symbolic HSVI* [69]. Ces deux algorithmes peuvent être arrêtés à n'importe quel moment, et plus les calculs durent, plus la stratégie calculée est performante : ainsi nous avons décidé d'arrêter les calculs lorsque l'erreur d'approximation passe en-dessous de 1. La figure III.5a, où l'instance du problème augmente avec la complexité du problème, montre que *APPL* déborde en mémoire à l'instance numéro 8, et *symbolic HSVI* à l'instance numéro 7, tandis que *PPUDD* peut calculer une stratégie pour des instances beaucoup plus compliquées. Nous pouvons aussi fixer une durée de calcul aux algorithmes probabilistes utilisés : le temps de calcul d'*APPL* est alors fixé au temps de calcul de *PPUDD*, et les performances de leurs stratégies respectives sont comparées en terme d'espérances de la somme des récompenses (et en utilisant les récompenses définies par le modèle probabiliste). Étonnam-

ment, la figure III.5b montre que les récompenses récupérées sont en moyenne plus grandes avec *PPUDD* qu'avec *APPL*. La raison est que *APPL* est en fait un planificateur probabiliste qui cherche à raffiner une approximation durant le temps de calcul, ce qui montre qui notre approche consistant à résoudre exactement un modèle approximé peut mieux fonctionner que de résoudre de manière approchée un modèle exact. Enfin, nous pouvons noter que les probabilités du modèle d'observation, qui représentent les incertitudes concernant les réponses des capteurs, peut être difficile à connaître précisément en pratique, auquel cas les modèles possibilistes peuvent plus physiquement rigoureux.

Ces résultats nous ont permis de juger pertinent la participation de *PPUDD* à la compétition internationale de planification probabiliste 2014, même si le calcul de stratégie pour les modèles probabilistes n'est pas la vocation initiale de ce planificateur. La section suivante discute des résultats des différents planificateurs.

III.4.2 Compétition Internationale de Planification Probabiliste 2014

La session entièrement observable de la compétition internationale de planification probabiliste permet de comparer des planificateurs de PDM en garantissant les même ressources en terme de puissance de calcul et de temps de calcul à chacun des algorithmes participants. Les planificateurs compétiteurs doivent calculer des stratégies pour certains problèmes qui ne sont pas connus à l'avance. Étant donné un de ces problèmes, les planificateurs ont un temps limité pour envoyer des actions à un serveur de la compétition qui simule l'évolution de l'état du système : les états successifs sont générés par le serveur de la compétition en utilisant la fonction de transition probabiliste du PDM définissant le problème, et envoyés à un des serveurs de compétiteurs. Pour chaque état du système reçu, le planificateur du compétiteur doit envoyer une action, qu'il a calculé, en retour. Ces échanges de données se produisent sur plusieurs exécutions d'horizon fini et le score du planificateur pour les problèmes considérés est la moyenne, sur les exécutions, de la somme finie des récompenses obtenue sur la trajectoire de l'état du système.

Plus d'informations à propos de cette compétition sont disponibles sur le site officiel de la compétition `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`. Les problèmes sont regroupés dans des *domaines*, qui sont des PDM dont un nombre fini de paramètres ne sont pas définis : le problème, ou PDM, utilisé en pratique durant la compétition est une *instance* d'un domaine, *i.e.* un domaine dont les paramètres ont été définis. Dans cette compétition, 8 domaines ont été proposés, appelé respectivement *Academic advising*, *Crossing traffic*, *Elevators*, *Skill teaching*, *Tamarisk*, *Traffic*, *Triangle tireworld* et *Wildfire*. La compétition consiste à évaluer les planificateurs sur 10 instances par domaine avec 30 exécutions de chaque instance et 18 minutes allouées par instance : elle dure donc 24 heures au total.

Quatre planificateurs ont été proposés pour cette compétition :

— *PROST* [38], basé sur l'algorithme *Upper Confidence bound applied to Trees* (UCT, [39]) ;
— *GOURMAND* [42, 41], basé sur l'algorithme *Labeled Real Time Dynamic Programming* (*LRTDP*, [8]) ;
— *symbolic LRTDP* [22] ;
— notre algorithme *PPUDD*.

Comme le score donné aux planificateurs ne dépend que des 40 premières étapes du processus, la version présentée de *PPUDD* est l'algorithme 4 avec la "condition du while" $\overline{U^*} \neq \overline{U^c}$ à la ligne 2 remplacée par la condition "numéro de l'itération $\leqslant 40$". Il augmente de manière incrémentale l'horizon de planification en maintenant un masque stocké sous forme d'un *BDD* (arbre de décision binaire, *i.e.* un *ADD* avec des feuilles dans $\{0, 1\}$) représentant les états atteignables à partir de l'état initial : le calcul de la fonction valeur courante est alors restreinte

aux états atteignables seulement. Bien que *PPUDD* soit un algorithme de calcul hors-ligne, nous avons aussi proposé *AnyTime PPUDD* (*ATPPUDD*), une version qui gère les temps de calcul comme décrit dans [42].

La bibliothèque utilisée pour faire les calculs entre *ADDs* s'appelle *CU Decision Diagram Package* (*CUDD*, `http://vlsi.colorado.edu/~fabio/CUDD/`), et les versions de *PPUDD* décrites sont disponibles à l'adresse `https://github.com/drougui/ppudd`.

Les figures qui suivent sont les résultats d'*IPPC* 2014 : les scores sont donnés en fonction de l'indice de l'instance, qui augmente généralement avec la complexité du problème associé.

La figure III.6 présente les scores obtenus par chacun des planificateurs pour chacune des 10 instances du domaine *Academic advising*, *i.e.* la moyenne sur les 30 exécutions, de la somme des récompenses rencontrées. Les performances de notre algorithme sont proches de celles des meilleurs algorithmes. Cependant, un bug non expliqué et indésirable s'est produit avec *ATPPUDD* lors de l'instance numéro 2, puisque seules trois exécutions ont été possibles avec ce planificateur. Pour les autres instances, *PPUDD* et *ATPPUDD* produisent des stratégies dont les performances sont semblables à celles de PROST et GOURMAND, et meilleures que *Symbolic LRTDP*. Ceci n'est plus vrai avec le problème *Crossing traffic*, dont les résultats sont décrits par la figure III.6. Ce problème modélise un robot qui doit atteindre un but qui est de l'autre côté d'une route à plusieurs voies que de nombreuses voitures empruntent. Ces voitures arrivent de manière aléatoire et vont vers la gauche. Comme le degré de possibilité attribué au fait qu'aucune voiture arrive a été fixé à 1 par notre traduction naïve de PDM en π-PDM, le critère optimiste mène à la décision de traverser la route même si une voiture (invisible au moment de la décision) peut arriver sur la droite (avec une probabilité < 0.5 mais assez grande pour devoir plutôt être prudent, ou pessimiste, et traverser la route dans une position avec une visibilité sur l'arrivée des voitures). Ceci explique la pauvre qualité des stratégies produites par *PPUDD* pour ce domaine. Notons cependant que, pour les 6 dernières intances (*i.e.* les problèmes les plus difficiles) notre approche mène à de meilleures stratégies que le planificateur probabiliste *Symbolic LRTDP*.

Le problème *Elevators* concerne des gens qui arrivent de manière aléatoire devant un ascenseur et qui doivent être transportés à l'étage qu'ils ont choisi d'un immeuble : comme l'information fréquentiste est perdue lors de l'utilisation de l'approche possibiliste, et semble très importante dans ce problème (les gens ne veulent pas attendre une fois arrivés devant l'ascenseur), cela explique pourquoi les scores de notre algorithme sont moins bons que ceux de PROST et GOURMAND. Cependant *PPUDD* et *ATPPUDD* sont meilleurs que Symbolic LRTDP, comme montré dans la figure III.7, et la stratégie qui consiste à ne rien faire ("noop") ou la stratégie qui choisit des actions de manière aléatoire ("random"). *PPUDD* et *ATPPUDD* ont des comportements plutôt bons avec le problème *Skill teaching* comme illustré par la même figure. De plus, *ATPPUDD* mène à de meilleurs résultats pour les trois dernières instances : puisque ces instances sont celles du domaine *Skill teaching* avec l'espace d'état le plus grand, la version anytime, qui gère les temps de calculs de manière plus recherchée, produit des stratégies avec de meilleures performances que *PPUDD*, qui résout classiquement le problème du π-PDM, associé, mais ne peut pas terminer les calculs et mène à des stratégies moins bonnes.

Par rapport aux autres planificateurs, les algorithmes possibilistes donnent de bon résultats avec le domaine *Tamarisk*, comme le montre la figure III.8. Cependant, certaines instances (par exemple les instances numéro 6, 8 et 10) ne sont même pas exécutées puisque l'instanciation des *ADDs* définissant le problème prennent trop de temps. *Symbolic LRTP* fait face au mêmes problèmes puisqu'il utilise aussi des *ADDs*. Le domaine *Traffic* est vraiment très dur à résoudre avec *PPUDD* et *ATPPUDD* (*cf.* figure III.8). En fait, les pires scores sont obtenus avec ce domaine, et même la stratégie aléatoire ou la stratégie "noop" sont meilleures. Il aurait été avantageux pour nous d'implémenter un garde-fou renvoyant des actions aléatoires lorsque la stratégie calculée est moins performante que la stratégie aléatoire. Comme mentionné au-dessus

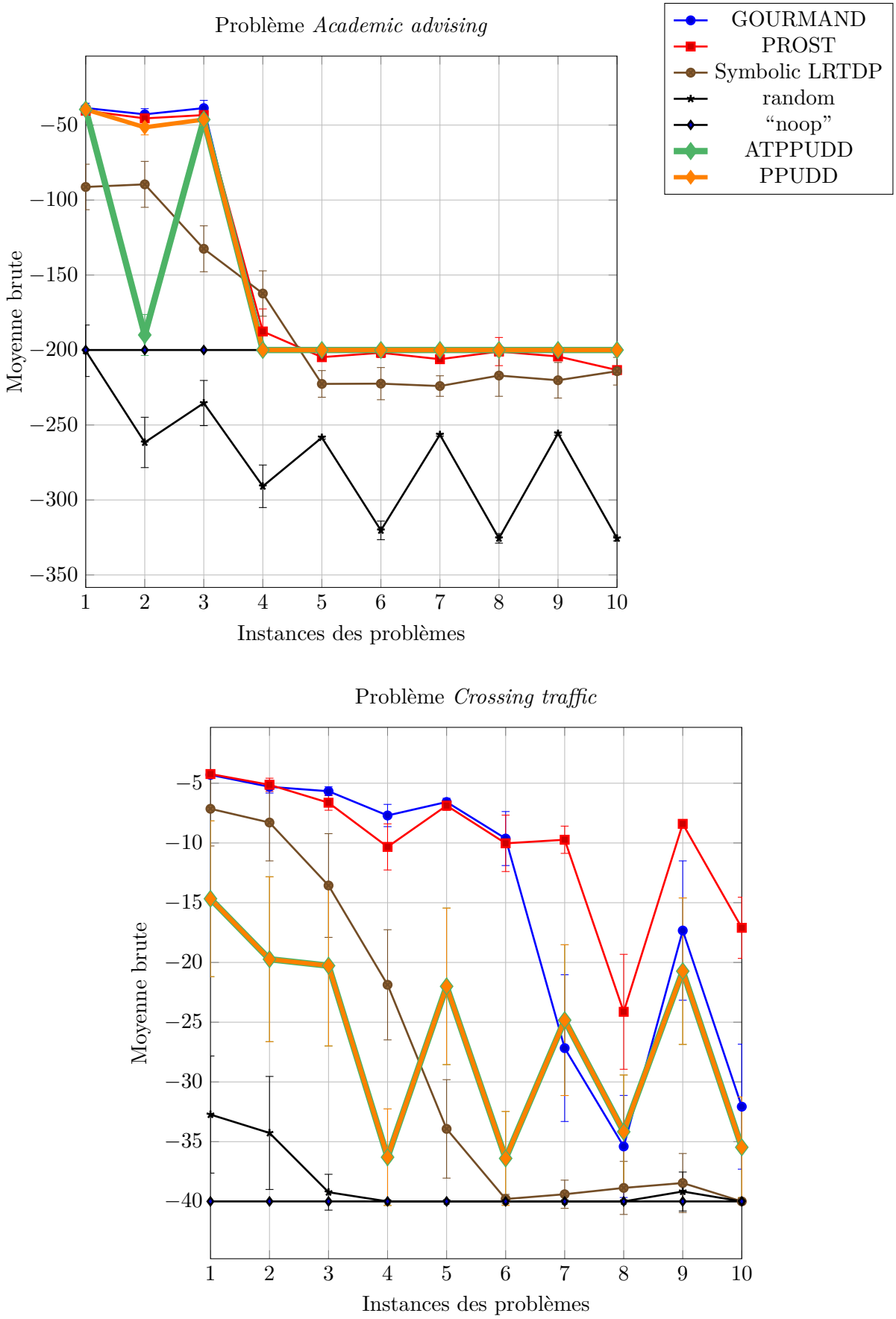


FIGURE III.6 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

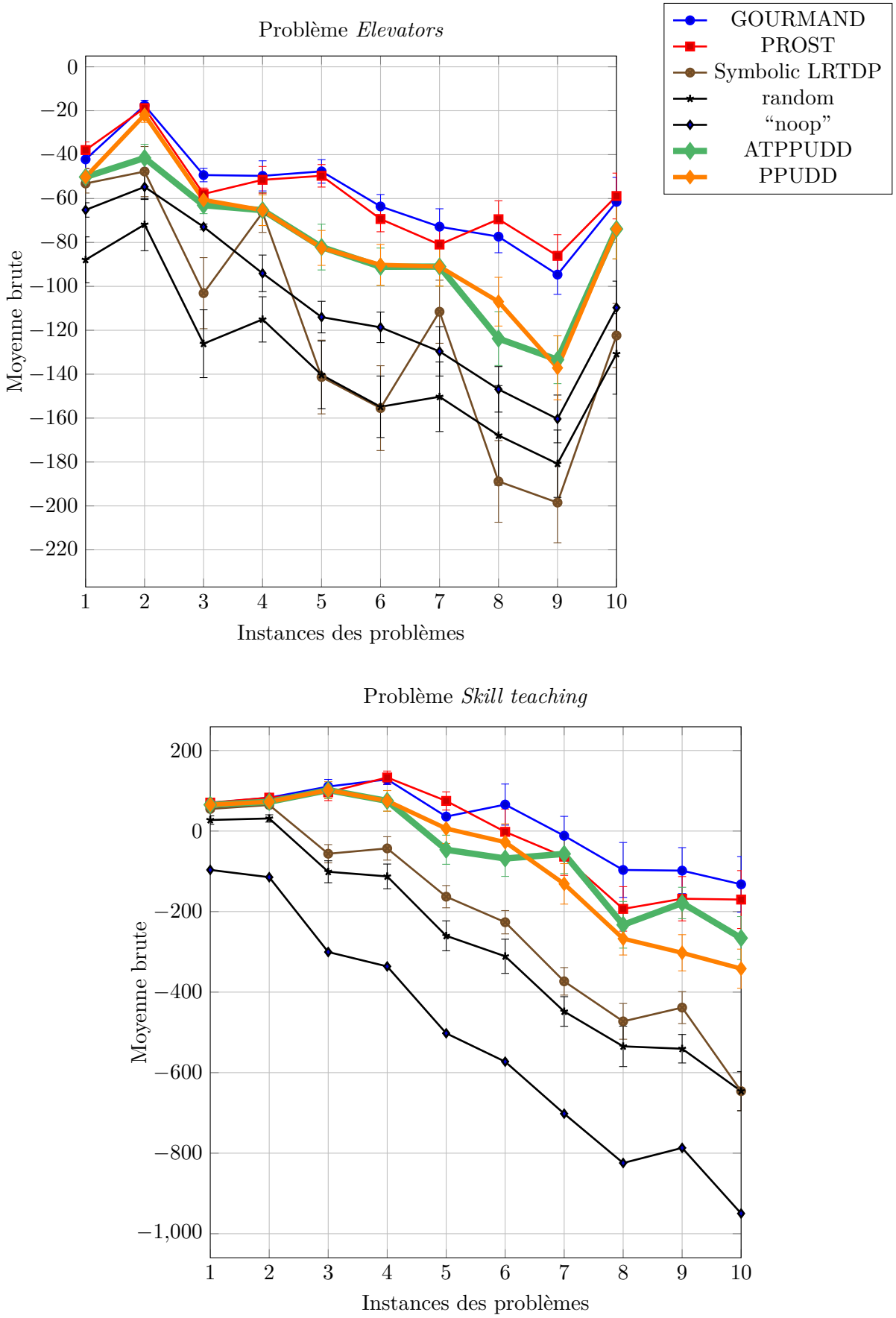


FIGURE III.7 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

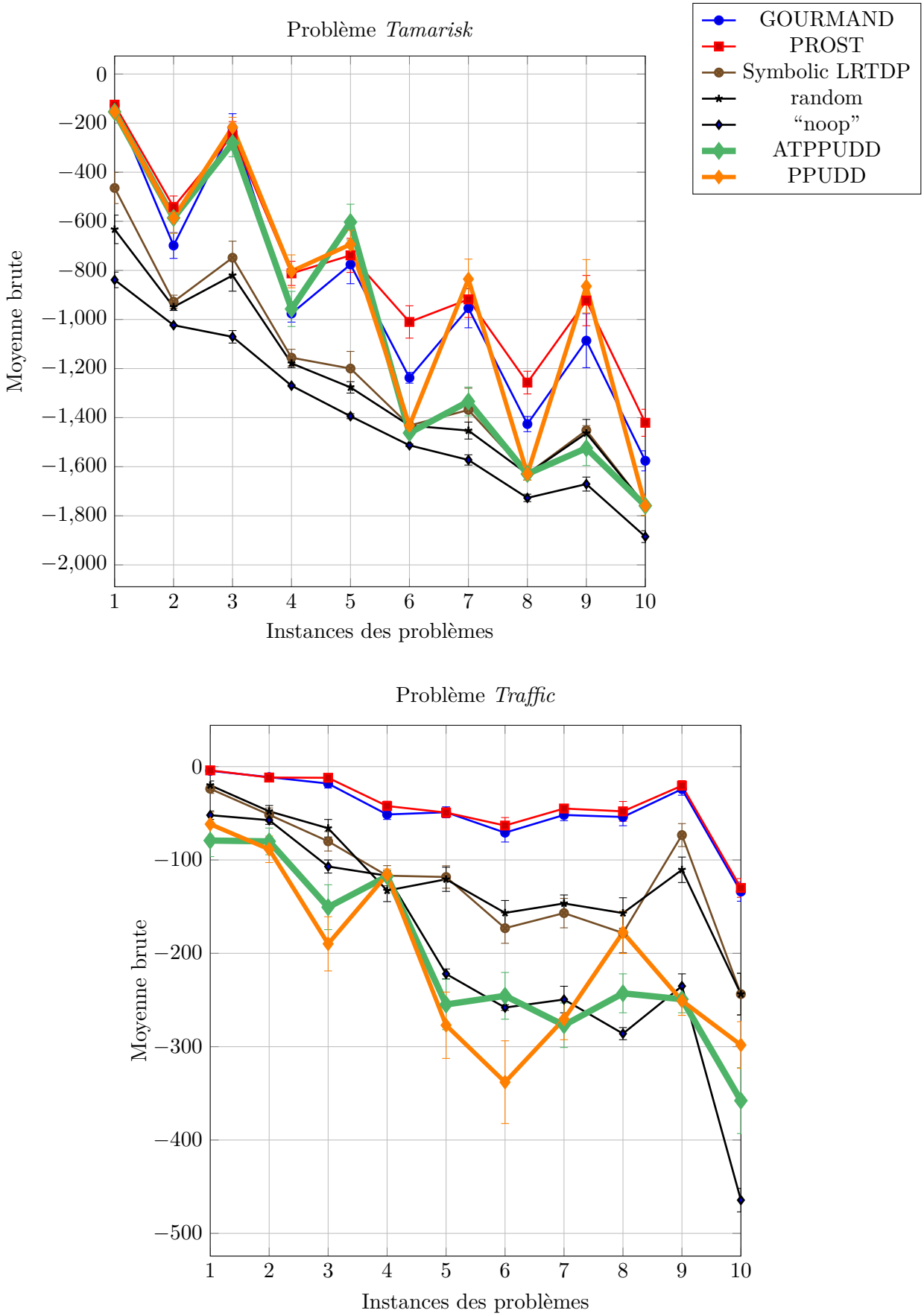


FIGURE III.8 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

avec le problème *Crossing traffic*, le critère optimiste peut mener à des actions dangereuses, comme cela se produit ici. De plus, puisque ce problème implique des informations fréquentistes (arrivées des voitures aléatoires), notre approche est sûrement inadaptée pour cet exemple. Enfin, le domaine *Traffic* est connu comme étant un des domaines les plus durs à résoudre, ainsi l'instanciation des *ADDs* en mémoire prennent trop de temps, tout comme les calculs, qui ne sont alors pas assez avancés pour produire des résultats satisfaisants.

Finalement, les deux derniers domaines, dont les résultats sont décrits dans la figure III.9, sont appelés *Triangle Tireworld* et *Wildfire*. Tout d'abord, *ATPPUDD* fait face à un bug inexpliqué pour chaque instance du domaine *Triangle Tireworld* : aucune exécution n'est réalisée à partir de l'instance numéro 5, et deux exécutions ont été réalisées pour les autres instances (ce qui explique le faible score pour chaque instance). Comme déjà mentionné pour le domaine *Tamarisk*, l'instanciation des *ADDs* prend trop de temps pour les dernières instances, et aucune exécution n'est réalisée pour les 4 dernières instances avec *PPUDD* : *Symbolic LRTDP* rencontre les même problèmes. Le dernier domaine, appelé *Wildfire*, constitue un problème très fréquentiste : il implique des départs de feux. C'est pourquoi *PPUDD* et *ATPPUDD* ne sont pas vraiment efficaces, mais pas trop distant non plus des résultats de *Symbolic LRTDP*.

III.5 Conclusion

Nous avons présenté *PPUDD*, le premier algorithme, à notre connaissance, qui résout les PDM(OM) possibilistes qualitatifs avec des calculs symboliques. Nous pensons que les modèles possibilistes constituent un bon compromis entre les modèles non-déterministes, où l'incertitude n'est pas quantifiée du tout, menant à un modèle très approximatif, et les modèles probabilistes, où l'incertitude est complètement spécifiée, et parfois de manière arbitraire en pratique. La résolution de problèmes de planification en utilisant le cadre du non-déterminisme est appelé *planification contingente/conformante*, étudiée par exemple dans [1, 9].

Nos résultats expérimentaux montrent que l'utilisation d'un algorithme exact (*PPUDD*) pour résoudre un modèle approximé (π-PDMOM) peut mener à des calculs plus rapides que de raisonner sur des modèles exacts et complexes, tout en générant des stratégies qui peuvent être de meilleure qualité que celles retournées par des algorithmes procédant à des approximations (*APPL*) sur des modèles exacts (PDMOM).

Enfin, ce chapitre présente les résultats de notre approche possibiliste durant *IPPC* 2014 : un des problèmes de cette approche est la traduction du modèle probabiliste en modèle possibiliste : la traduction automatique naïve utilisée est peut être à l'origine des mauvaises performances de l'approche pour certain domaines à la dynamique, ou la fonction de récompense complexe. Un autre problème semble être l'utilisation des *ADDs* : l'instanciation de ces objets avant même le début des calculs prend un temps trop important, où bien ne passe même pas en mémoire. Cette difficulté est partagée avec le planificateur *Symbolic LRTDP*. Des problèmes de modélisation ont aussi été mis en évidence : certains problèmes nécessitent une approche optimiste, et d'autres une approche pessimiste.

Cependant, bien que PROST et GOURMAND ont des performances deux fois supérieures à *PPUDD* et *ATPPUDD*, ces derniers proposent des stratégies dont les performances sont deux fois supérieures à son homologue probabiliste *Symbolic LRTDP*. Notons finalement que *PPUDD* est disponible dans le dépôt `https://github.com/drougui/ppudd`.

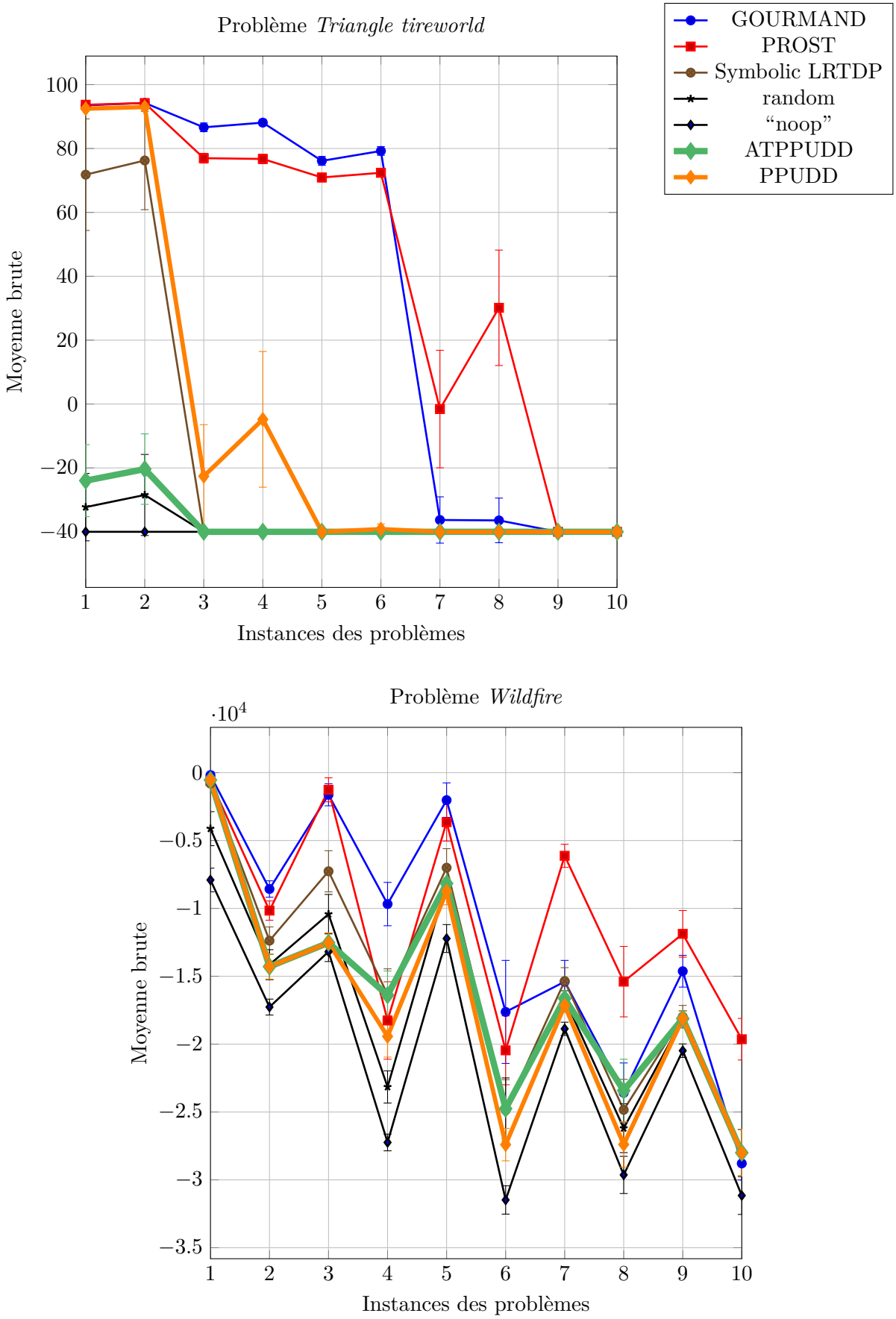


FIGURE III.9 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

Conclusion

Les contributions de cette thèse sont principalement centrées autour des travaux préliminaire de Régis Sabbadin [62]. Ce dernier propose un homologue possibiliste qualitatif aux PDMPO modélisant l'incertitude avec des distributions de possibilité qualitatives. Ainsi, dans notre travail, des modèles possibilistes qualitatifs pour la planification sous incertitude ont été développés et étudiés. Cette étude a été motivée par différents problèmes posés par les PDMPO détaillés en introduction.

La motivation majeure de cette étude est la réduction de la complexité des calculs offerte par les π-PDMPO : son espace d'états de croyance est fini, tandis que l'espace des états de croyance probabilistes est infini. De plus, comme ses états de croyance sont des distributions de possibilité sur l'espace des états, l'ignorance totale peut être définie par une distribution de possibilité égale à 1 sur tous les états. Si un état donné du système est parfaitement connu comme étant l'état courant, l'état de croyance associé attribue 1 à cet état du système et 0 à tous les autres puisqu'ils sont impossibles : c'est une représentation plus appropriée que la probabilité uniforme. L'imprécision des distributions de probabilités est aussi naturellement gérée par le formalisme possibiliste qualitatif.

Plus généralement, les modèles possibilistes qualitatifs ont besoin de moins d'information à propos du système que le modèle probabiliste : les plausibilités des événements sont "seulement" classifiées dans l'échelle possibiliste $\mathcal{L}$ plutôt que quantifiées. Cette thèse propose finalement des contributions théoriques et pratiques à propos de ces processus possibilistes qualitatifs : d'une part, les contributions théoriques sont par exemple l'observabilité mixte, la gestion des horizons indéfinis, ou les résultats d'indépendance. D'autre part, la principale contribution pratique est la démonstration que ces modèles possibilistes qualitatifs ont un intérêt dans la simplification des calculs ou la modélisation via des résultats expérimentaux (par exemple IPPC 2014) ou l'étude du comportement de la croyance possibiliste.

Ces contributions sont amenées à travers des applications robotiques afin de faire le lien avec les problématiques de départ : des missions de reconnaissance de cible sont étudiées (par exemple le problème *Rocksample*, ou même la mission décrite par la figure II.2) ; IPPC 2014 contient aussi des problèmes comparables à des systèmes robotiques (par exemple le problème *Elevator*, ou le problème *Tamarisk*, aussi utilisé lors de la compétition d'apprentissage par renforcement 2014).

Perspectives

La première perspective vient d'une observation : les problèmes de mémoire atteints avec le domaine *Triangle tireworld* d'IPPC 2014 pourrait être contourné à l'aide d'une discrétisation plus importante : les version de *PPUDD* pour IPPC 2014 utilisaient une précision de 10^{-3} sur le problème probabiliste initial, menant à une importante échelle possibiliste $\mathcal{L}$. Il serait instructif d'avoir une idée de l'impact de ce prétraitement sur les performances de *PPUDD* : plus généralement, un travail plus important sur la traduction d'un PDM probabiliste en approximation possibiliste pourrait améliorer de manière significative l'approche proposée pour IPPC 2014.

En terme de méthode de calculs, nous nous sommes concentrés sur des résolutions symboliques des π-PDM (*PPUDD*). Cependant, des méthodes de résolutions alternatives pourraient être étudiées : par exemple, des méthodes heuristiques, qui sont efficaces pour les problèmes probabilistes [73], pourraient être adaptées au contexte possibiliste. En ce qui concerne l'apprentissage par renforcement dans le cadre des possibilités qualitatives, nous pouvons citer le travail suivant [65].

Enfin, les avancées de la théorie des possibilités peuvent permettre de raffiner les PDM possibilistes qualitatifs afin d'améliorer la modélisation lorsque c'est nécessaire. Nous pouvons par exemple mentionner l'opérateur *leximin* [26] : il permet d'éviter l'effet de noyade des opérateurs min et max. Il peut être utilisé pour l'aggrégation de préférences ou pour calculer un degré de possibilité plus fin d'une trajectoire. Notons que cela peut être une amélioration utile, mais qu'elle complexifie inévitablement le problème. Des critères plus discriminants sont aussi étudiés dans la littérature [76, 35].

Bibliographie

[1] Alexandre Albore, Héctor Palacios, and Hector Geffner. A translation-based approach to contingent planning. In Craig Boutilier, editor, *IJCAI*, pages 1623–1628, 2009. (Cité page 50.)

[2] M. Araya-Lòpez, V. Thomas, O. Buffet, and F. Charpillet. A closer look at MOMDPs. In *Proceedings of the Twenty-Second IEEE International Conference on Tools with Artificial Intelligence (ICTAI-10)*, 2010. (Cité pages 13, 28, 31 et 43.)

[3] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebric decision diagrams and their applications. *Form. Methods Syst. Des.*, 10(2-3) :171–206, April 1997. (Cité pages 35 et 38.)

[4] Richard Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6) :503–515, 11 1954. (Cité page 9.)

[5] Richard Bellman. A Markovian Decision Process. *Indiana Univ. Math. J.*, 6 :679–684, 1957. (Cité page 3.)

[6] Nahla Ben Amor. *Qualitative possibilistic graphical models : from independence to propagation algorithms*. Thèse de doctorat, ISG - Université de Tunis, Tunis, juin 2002. (Cité pages 13 et 19.)

[7] Blai Bonet. New grid-based algorithms for partially observable Markov decision processes : Theory and practice. (Cité page 7.)

[8] Blai Bonet. Labeled RTDP : improving the convergence of real-time dynamic programming. In *In Proc. ICAPS-03*, pages 12–21. AAAI Press, 2003. (Cité page 45.)

[9] Blai Bonet and Hector Geffner. Flexible and scalable partially observable planning with linear translations. In *AAAI*, 2014. (Cité page 50.)

[10] Christian Borgelt, Jörg Gebhardt, and Rudolf Kruse. Graphical models. In *In Proceedings of International School for the Synthesis of Expert Knowledge (ISSEK'98*, pages 51–68. Wiley, 2002. (Cité page 13.)

[11] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121(1-2) :49–107, 2000. (Cité page 35.)

[12] Xavier Boyen and Daphne Koller. Exploiting the architecture of dynamic systems. In *AAAI/IAAI*, pages 313–320, 1999. (Cité page 37.)

[13] Ronen I. Brafman. A heuristic variable grid solution method for POMDPs. In *In AAAI*, pages 727–733, 1997. (Cité page 7.)

[14] Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning : A simple, fast, exact method for partially observable Markov decision processes. In *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 54–61. Morgan Kaufmann Publishers, 1997. (Cité page 7.)

[15] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. POMDP-based online target detection and recognition for autonomous UAVs. In *ECAI*

2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012, pages 955–960, 2012. (Cité page 6.)

[16] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. Multi-target detection and recognition by UAVs using online POMDPs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013. (Cité page 6.)

[17] Gustave Choquet. Theory of capacities. *Annales de l'institut Fourier*, 5 :131–295, 1954. (Cité page 17.)

[18] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7 : A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. (Cité page 9.)

[19] Ines Couso and Sébastien Destercke. Didier's groundhog day. 19(2) :10–15, December 2012. (Cité page 15.)

[20] B. De Finetti. *Theory of probability : a critical introductory treatment.* Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1974. (Cité pages 10 et 15.)

[21] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5(3) :142–150, December 1989. (Cité page 37.)

[22] KarinaValdivia Delgado, Cheng Fang, Scott Sanner, and Leliane Nunes de Barros. Symbolic bounded real-time dynamic programming. In Antônio Carlos da Rocha Costa, Rosa Maria Vicari, and Flavio Tonidandel, editors, *Advances in Artificial Intelligencev - SBIA 2010*, volume 6404 of *Lecture Notes in Computer Science*, pages 193–202. Springer Berlin Heidelberg, 2010. (Cité page 45.)

[23] Nicolas Drougard, Didier Dubois, Jean-Loup Farges, and Florent Teichteil-Königsbuch. Planning in partially observable domains with fuzzy epistemic states and probabilistic dynamics. In *Scalable Uncertainty Management - 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings*, pages 220–233, 2015. (Cité page 14.)

[24] Nicolas Drougard, Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois. Qualitative possibilistic mixed-observable MDPs. In *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 192–201, Corvallis, Oregon, 2013. AUAI Press. (Cité page 13.)

[25] Nicolas Drougard, Florent Teichteil-Königsbuch, Jean-Loup Farges, and Didier Dubois. Structured possibilistic planning using decision diagrams. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2257–2263, 2014. (Cité page 14.)

[26] Didier Dubois and HÃ©lÃ¨ne Fargier. Lexicographic refinements of sugeno integrals. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724 of *Lecture Notes in Computer Science*, pages 611–622. Springer Berlin Heidelberg, 2007. (Cité page 54.)

[27] Didier Dubois and Henri Prade. *Possibility Theory : An Approach to Computerized Processing of Uncertainty (traduction revue et augmentée de "Théorie des Possibilités")*. Plenum Press, New York, 1988. (Cité page 15.)

[28] Didier Dubois and Henri Prade. Possibility Theory as a basis for qualitative decision theory. In *IJCAI*, pages 1924–1930. Morgan Kaufmann, 1995. (Cité pages 11 et 18.)

[29] Didier Dubois, Henri Prade, and Régis Sabbadin. Decision-theoretic foundations of qualitative possibility theory. *European Journal of Operational Research*, 128(3) :459–478, 2001. (Cité pages 12 et 18.)

[30] Didier Dubois, Henri Prade, and Sandra Sandri. On possibility/probability transformations. In *Proceedings of Fourth IFSA Conference*, pages 103–112. Kluwer Academic Publ, 1993. (Cité page 31.)

[31] Didier Dubois, Henri Prade, and Philippe Smets. Representing partial ignorance. *IEEE Trans. on Systems, Man and Cybernetics*, 26 :361–377, 1996. (Cité page 10.)

[32] Hélène Fargier, Nahla Ben Amor, and Wided Guezguez. On the complexity of decision making in possibilistic decision trees. *CoRR*, abs/1202.3718, 2012. (Cité page 12.)

[33] Laurent Garcia and Régis Sabbadin. Complexity results and algorithms for possibilistic influence diagrams. *Artificial Intelligence*, 172(8-9) :1018 – 1044, 2008. (Cité page 12.)

[34] Hector Geffner and Blai Bonet. Solving large POMDPs using real time dynamic programming. In *In Proc. AAAI Fall Symp. on POMDPs*, 1998. (Cité page 7.)

[35] Phan Hong Giang and Prakash P. Shenoy. A comparison of axiomatic approaches to qualitative decision making using possibility theory. In Jack S. Breese and Daphne Koller, editors, *UAI*, pages 162–170. Morgan Kaufmann, 2001. (Cité page 54.)

[36] Jesse Hoey, Robert St-Aubin, Alan Hu, and Craig Boutilier. SPUDD : Stochastic planning using decision diagrams. In *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 279–288. Morgan Kaufmann, 1999. (Cité pages 14, 35, 37 et 38.)

[37] Hideaki Itoh and Kiyohiko Nakamura. Partially observable Markov decision processes with imprecise parameters. *Artificial Intelligence*, 171(8-9) :453 – 490, 2007. (Cité page 9.)

[38] Thomas Keller and Patrick Eyerich. PROST : probabilistic planning based on UCT. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012, Atibaia, São Paulo, Brazil, June 25-19, 2012*, 2012. (Cité pages 14 et 45.)

[39] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. (Cité page 45.)

[40] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models : Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. (Cité page 13.)

[41] Andrey Kolobov, Peng Dai, Mausam Daniel, and S. Weld. Reverse iterative deepening for finite-horizon mdps with large branching factors. In *In ICAPS'12*, 2012. (Cité page 45.)

[42] Andrey Kolobov, Mausam, and Daniel S. Weld. LRTDP versus UCT for online probabilistic planning. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012. (Cité pages 14, 45 et 46.)

[43] Hanna Kurniawati, David Hsu, and Wee Sun Lee. SARSOP : Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of Robotics : Science and Systems IV*, Zurich, Switzerland, June 2008. (Cité pages 7 et 36.)

[44] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006. (Cité page 29.)

[45] Y. Lecun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. volume 2, 2004. (Cité page 8.)

[46] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. (Cité page 8.)

[47] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 541–548, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence. (Cité pages 6 et 12.)

[48] Bhaskara Marthi. Robust navigation execution by planning in belief space. In *Robotics : Science and Systems VIII, University of Sydney, Sydney, NSW, Australia, July 9-13, 2012*, 2012. (Cité page 6.)

[49] Martin Mundhenk. The complexity of planning with partially-observable Markov decision processes. Technical report, Hanover, NH, USA, 2000. (Cité page 7.)

[50] Yaodong Ni and Zhi-Qiang Liu. Policy iteration for bounded-parameter POMDPs. *Soft Computing*, pages 1–12, 2012. (Cité page 9.)

[51] Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5) :780–798, September-October 2005. (Cité page 9.)

[52] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *Int. J. Rob. Res.*, 29(8) :1053–1068, July 2010. (Cité pages 6, 13, 28, 31, 36, 43 et 44.)

[53] Takayuki Osogami. Robust partially observable Markov decision process. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 106–115, 2015. (Cité page 9.)

[54] Christos Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Math. Oper. Res.*, 12(3) :441–450, August 1987. (Cité pages 6 et 12.)

[55] Judea Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. (Cité page 41.)

[56] Patrice Perny, Olivier Spanjaard, and Paul Weng. Algebraic Markov Decision Processes. In *19th International Joint Conference on Artificial Intelligence*, pages 1372–1377, 2005. (Cité page 21.)

[57] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration : An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025 – 1032, August 2003. (Cité page 7.)

[58] Joelle Pineau and Geoffrey J. Gordon. POMDP planning for robust robot control. In Sebastian Thrun, Rodney A. Brooks, and Hugh F. Durrant-Whyte, editors, *ISRR*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 69–82. Springer, 2005. (Cité page 6.)

[59] Cédric Pralet, Thomas Schiex, and Gérard Verfaillie. *Sequential Decision-Making Problems - Representation and Solution*. Wiley, 2009. (Cité page 34.)

[60] Julia Radoszycki, Nathalie Peyrard, and Régis Sabbadin. Finding good stochastic factored policies for factored markov decision processes. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 1083–1084, 2014. (Cité page 35.)

[61] Régis Sabbadin. *Une Approche Ordinale de la Decision dans l'Incertain : Axiomatisation, Representation Logique et Application à la Décision Séquentielle*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 1998. (Cité page 19.)

[62] Régis Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI'99, pages 567–574, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. (Cité pages 11, 12, 13, 19, 21, 22, 29, 31 et 53.)

[63] Régis Sabbadin. Empirical comparison of probabilistic and possibilistic Markov decision processes algorithms. In Werner Horn, editor, *ECAI*, pages 586–590. IOS Press, 2000. (Cité pages 13 et 19.)

[64] Régis Sabbadin. Possibilistic Markov decision processes. *Engineering Applications of Artificial Intelligence*, 14(3) :287 – 300, 2001. Soft Computing for Planning and Scheduling. (Cité pages 13 et 19.)

[65] Régis Sabbadin. Towards possibilistic reinforcement learning algorithms. In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems, Melbourne, Australia, December 2-5, 2001*, pages 404–407, 2001. (Cité page 54.)

[66] Régis Sabbadin, Hélène Fargier, and Jérôme Lang. Towards qualitative approaches to multi-stage decision making. *Int. J. Approx. Reasoning*, 19(3-4) :441–471, 1998. (Cité page 18.)

[67] Guy Shani, Pascal Poupart, Ronen I. Brafman, and Solomon Eyal Shimony. Efficient ADD operations for point-based algorithms. In *ICAPS*, pages 330–337, 2008. (Cité page 37.)

[68] David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2164–2172. Curran Associates, Inc., 2010. (Cité page 7.)

[69] Hyeong Seop Sim, Kee-Eung Kim, Jin Hyung Kim, Du-Seong Chang, and Myoung-Wan Koo. Symbolic heuristic search value iteration for factored POMDPs. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 1088–1093. AAAI Press, 2008. (Cité pages 36 et 44.)

[70] Richard D. Smallwood and Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon*, volume 21. INFORMS, 1973. (Cité page 4.)

[71] Trey Smith and Reid Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States, 2004. AUAI Press. (Cité pages 7, 36 et 40.)

[72] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974. (Cité page 17.)

[73] Florent Teichteil-Königsbuch, Vincent Vidal, and Guillaume Infantes. Extending classical planning heuristics to probabilistic planning with dead-ends. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011. (Cité page 54.)

[74] Judea P. Thomas Verma. Influence Diagrams and d-Separation. Technical report, July 1988. (Cité pages 14 et 41.)

[75] Tom S. Verma and Judea Pearl. Causal networks : Semantics and expressiveness. *CoRR*, abs/1304.2379, 2013. (Cité page 41.)

[76] Paul Weng. Qualitative Decision-Making Under Possibilistic Uncertainty : Toward More Discriminating Criteria. In *21st International Conference on Uncertainty in Artificial Intelligence*, volume 21, pages 615–622, 2005. INT LIP6 DECISION. (Cité page 54.)

[77] Paul Weng. Conditions générales pour l'admissibilité de la programmation dynamique dans la décision séquentielle possibiliste. *Revue d'Intelligence Artificielle*, 21(1) :129–143, 2007. NAT LIP6 DECISION. (Cité pages 13 et 34.)

[78] Stefan Witwicki, Francisco S. Melo, Jesús Capitán, and Matthijs T. J. Spaan. A flexible approach to modeling unpredictable events in MDPs. In *Proc. of Int. Conf. on Automated Planning and Scheduling*, pages 260–268, 2013. (Cité page 41.)

[79] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338 – 353, 1965. (Cité page 15.)

Title: Exploiting Imprecise Information Sources for Sequential Decision Making under Uncertainty

Abstract: Partially Observable Markov Decision Processes (POMDPs) define a useful formalism to express probabilistic sequential decision problems under uncertainty. When this model is used for a robotic mission, the *system* is defined as the features of the robot and its environment, needed to express the mission. The system state is not directly seen by the agent (the robot). Solving a POMDP consists thus in computing a strategy which, on average, achieves the mission best *i.e.* a function mapping the information known by the agent to an action. Some practical issues of the POMDP model are first highlighted in the robotic context: it concerns the modeling of the agent ignorance, the imprecision of the observation model and the complexity of solving real world problems. A counterpart of the POMDP model, called π-POMDP, simplifies uncertainty representation with a qualitative evaluation of event plausibilities. It comes from Qualitative Possibility Theory which provides the means to model imprecision and ignorance. After a formal presentation of the POMDP and π-POMDP models, an update of the possibilistic model is proposed. Next, the study of factored π-POMDPs allows to set up an algorithm named PPUDD which uses Algebraic Decision Diagrams to solve large structured planning problems. Strategies computed by PPUDD, which have been tested in the context of the competition IPPC 2014, can be more efficient than those produced by probabilistic solvers when the model is imprecise or for high dimensional problems. We show next that the π-Hidden Markov Processes (π-HMP), *i.e.* π-POMDPs without action, produces useful diagnosis in the context of Human-Machine interactions. Finally, a hybrid POMDP benefiting from the possibilistic and the probabilistic approach is built: the qualitative framework is only used to maintain the agent's knowledge. This leads to a strategy which is pessimistic facing the lack of knowledge, and computable with a solver of fully observable Markov Decision Processes (MDPs). This thesis proposes some ways of using Qualitative Possibility Theory to improve computation time and uncertainty modeling in practice.

Keywords: POMDP, Planning under Uncertainty, Possibility Theory, Autonomous Robotics, Imprecise Knowledge

Titre: Tirer Profit de Sources d'Information Imprécises pour la Décision Séquentielle dans l'Incertain

Résumé: Les Processus Décisionnels de Markov Partiellement Observables (PDMPOs) permettent de modéliser facilement les problèmes probabilistes de décision séquentielle dans l'incertain. Lorsqu'il s'agit d'une mission robotique, les caractéristiques du robot et de son environnement nécessaires à la définition de la mission constituent le *système*. Son état n'est pas directement visible par l'*agent* (le robot). Résoudre un PDMPO revient donc à calculer une stratégie qui remplit la mission au mieux en moyenne, *i.e.* une fonction prescrivant les actions à exécuter selon l'information reçue par l'agent. Ce travail débute par la mise en évidence, dans le contexte robotique, de limites pratiques du modèle PDMPO: elles concernent l'ignorance de l'agent, l'imprécision du modèle d'observation ainsi que la complexité de résolution. Un homologue du modèle PDMPO appelé π-PDMPO, simplifie la représentation de l'incertitude: il vient de la Théorie des Possibilités Qualitatives qui définit la plausibilité des événements de manière qualitative, permettant la modélisation de l'imprécision et de l'ignorance. Une fois les modèles PDMPO et π-PDMPO présentés, une mise à jour du modèle possibiliste est proposée. Ensuite, l'étude des π-PDMPOs factorisés permet de mettre en place un algorithme appelé PPUDD utilisant des Arbres de Décision Algébriques afin de résoudre plus facilement les problèmes structurés. Les stratégies calculées par PPUDD, testées par ailleurs lors de la compétition IPPC 2014, peuvent être plus efficaces que celles des algorithmes probabilistes dans un contexte d'imprécision ou pour certains problèmes à grande dimension. Nous montrons ensuite que les Processus de Markov Cachés possibilistes (π-PMCs), *i.e.* les π-PDMPOs sans les actions, produisent de bons diagnostics dans le contexte de l'intéraction Homme-Machine. Enfin, un PDMPO hybride tirant profit des avantages des modèles probabilistes et possibilistes est présenté: seule la connaissance de l'agent est maintenue sous forme qualitative. Ce modèle mène à une stratégie qui réagit de manière pessimiste au défaut de connaissance, et calculable avec des algorithmes de résolution des Processus Décisionnels de Markov entièrement observables (PDM). Cette thèse propose d'utiliser les possibilités qualitatives dans le but d'obtenir des améliorations en termes de temps de calcul et de modélisation de l'incertitude en pratique.

Mots-clés: PDMPO, Planification dans l'Incertain, Théorie des Possibilités, Robotique Autonome, Connaissance Imprecise