ISBN 978-80-7378-250-4 © MATFYZPRESS

Probabilistic Modeling of Dynamic Systems

R. Brunetto

Charles University in Prague, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. This article briefly introduces selected probabilistic approaches to search optimal behaviour of an agent in dynamic systems. It describes two generalizations of Markov decision processes (MDP) which are the standard for probabilistic planning. Our uncertainty about transition probabilities is described by Markov decision processes with imprecise probabilities (MDP-IP). Missing information about actual state can be handled by Partially observable Markov decision processes (POMDP). This article compares both approaches and discusses their possible combination.

Introduction

Planning has a long tradition in artificial intelligence [*Pineau et al.*, 2006]. Classical planning assumes fully observable discrete deterministic environment. In some situations such as making plans for real physical robot these assumptions are not valid. When the robot decides to take an action it uses the actuators which are supposed to change the environment or the robot's position. The exact result of such actions, e.g., robot's position is often nondeterministic or it depends on many unmodelled factors that can be treated as randomness. Problems where the results of actions can be described as probability distributions over robot positions are modelled using Markov decision processes (MDPs). This framework assumes that the agent fully observes new state of the world after performing the action (Figure 1a).

In real world this is not sufficient enough for modelling robots. It is common that exact probability distributions of actions' results are not given and can change over time. Markov decision processes with imprecise probabilities (MDP-IPs) framework was created to model this kind of scenarios.

The problem becomes even more complicated when the agent resp. robot does not know the exact state of the world resp. its own location. This can be modelled by so-called partially observable Markov decision processes (POMDPs). Instead of the exact starting state of agent's world resp. robot's starting location only probability distribution over the possibilities is known. This probability distribution is called belief state or only belief. Knowing this information agent chooses an action and performs it. Afterwards it does not know the exact action's result but it can sense observation which may contain some information about the new state of the agent's world. Then the agent is facing the next decision. One way to make it is to use the observation in order to calculate new belief state (denoted b) from which the same action selection policy (denoted π) can be reused (Figure 1b). The component which computes the new belief state is labeled SE for state estimator. It uses previous belief state, previous action and the observation as it's input.

Throughout this paper we use traffic example inspired by example in *Delgado et al.* [2011]. It models a busy intersection directed by the lights. Each car stops if there is a red light or another car in front. In the opposite case it continues through the intersection in a random direction.

Structure of following text

The following sections compare all three classes of the models mentioned above. First section after this introduction defines MDP, MDP-IP and POMDP formally and shows their differences on the traffic example. The next section defines policy for both fully and partially observable environments which allows us to define optimal policy for all three classes of models. Computing optimal policy is general goal. The way to achieve it is similar for all three classes of models. This article explains it and compares the calculations needed for solving all three classes of models also. The contribution of this article is the discussion of possible combination of all three model types. It can be found in the last section.

Definitions

This section formally defines MDPs, POMDPs and MDP-IPs and shows their possible usage on examples.

Markov Decision processes (MDPs)

MDPs are defined as tuples of the form (S, A, R, T) where

- S is a set of states
- A is a set of actions
- R is a reward function $S \times A \rightarrow R$ telling how much the agent is rewarded for taking a given action in a given state.
- T is a transition function $S \times A \times S \rightarrow [0, 1]$ giving the probability $T(s_t, a_t, s_{t+1})$ of moving from state s_t to state s_{t+1} by executing action a_t.

In our example the set of states encodes positions of all the cars. The set of actions contains possible light combinations. Transition function would describe car movements and their probabilities. The reward function would be set up the way to minimize number of cars waiting at the intersection.

MDPs are standard framework for planning under uncertainty in fully obserable environments with known transition probabilities.

Partially observable Markov decision processes (POMDPs)

One of the reasons why modelling intersection with MDP is unrealistic is that it is not common that the sensors at the intersection could know the precise location of all cars. More typically only partial information about positions of some cars at the intersection is observed. The probabilities of receiving possible observations are defined by observation function.

POMDPs are defined as tuples of the form (S, A, R, T, Ω, O) where

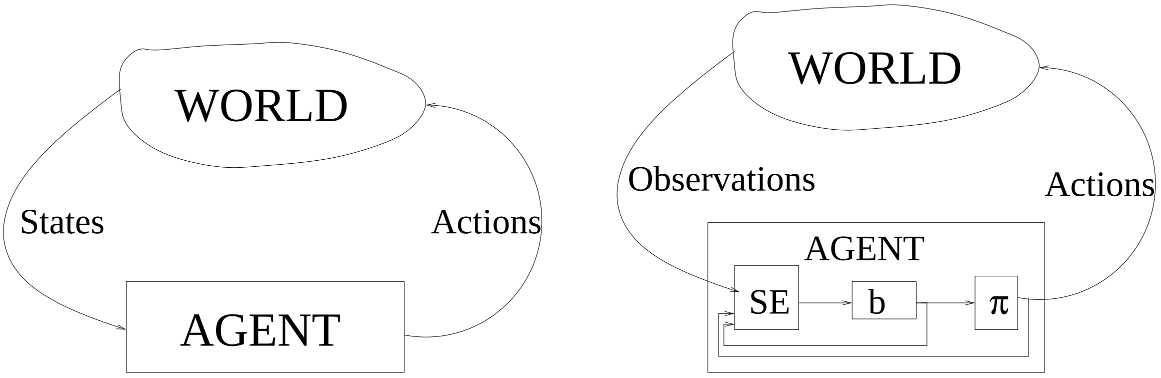
- S, A, R, and T are the same as in the MDP.
- Ω is a set of observations
- $O : S \times A \times \Omega \rightarrow [0, 1]$ is the observation function giving the probability of observations for given states and actions.

The difference of POMDP agent from MDP agent is the fact that the POMDP agent can not observe the state. Instead of that it only receives observation o which is randomly drawn from Ω according to probablitities given by observation function.

Markov decision processes with imprecise transition probabilities (MDP-IPs)

Another reason why modelling intersection using MDP is unrealistic is that we usually do not know the exact probabilities of cars' behaviour. It depends on the time, on the day of week, on holidays, on regional events and on other factors. Instead of searching optimal intersection behaviour for one setup of probabilities one would rather find policy always behaving reasonably. MDP-IPs can be utilized to find it. MDP-IPs are defined as tuples of the form (S, A, R, K) where

- S, A, and R are defined the same way as in MDPs.



(a) Agent in fully observable environment.

(b) Agent in partially observable environment.

Figure 1. Full and partial observability comparison.

- K is a set of transition functions, $K \subseteq \{T | T : S \times A \times S \to [0, 1]\}$, i.e., set of functions which prescribes the probabilities of resulting states after performing given action from given previous state.

This definition allows us to model non-stationary processes. The state transits itself to new states according to unknown transition probabilities given by a function, which is non-deterministically drawn from the set K. Note that this setup is different from unknown but stationary transition probabilities, which can be learned by popular reinforcement techniques [*Moore et al.*, 1996]. Note that we also do not assume any prior distribution over transition function. In the contrary we expect that the enemy can change the transition probabilities any time the way which is the worst for the agent. This prevents us from using bayesian methods and it prevents the agent from taking advantages of learning. The pros and cons of these assumptions can be seen in the intersection example. The agent can't learn to act according to typical traffic, but is robust because of assuming the worst case scenarios.

Delgado et al. [2011] shown how to solve MDP-IPs effectively under some conditions. They assumed that K is either finite or can be expressed by finite set of linear constrains.

MDPs are special cases of MDP-IPs where $K = \{T\}$.

Common background

In the beginning the agent is in the state s_0. The MDP and MDP-IP frameworks assume that this state is known by the agent. The POMDP framework assume that the agent knows the probability distribution b_0 of initial states.

At each time period t the agent chooses action $a_t \in A$ accordingly to state s_t resp. belief state b_t. Perfoming this action causes the state random change to s_{t+1}. The probabilities of transitions are given by transition function T, $P(s_{t+1}|s_t, a_t) = T(s_t, a_t, s_{t+1})$.

The agent receives a reward r_t according to reward function, $r_t = R(s_{t+1}, a_t)$. Note that this reward is not observed by the agent in partially observable case. The only information agent has in partially observable case is observation o_t which is randomly drawn from the set of observations O with respect to probabilities given by observation function O, $P(o_t|s_{t+1}, a_t) = O(s_{t+1,a_t,o_t})$.

Knowing previous belief state, action performed and observation received agent can compute the next belief state using transition function T and observation function O. We denote $\tau_{O,T}(b, a, o)$ the next belief comuted this way.

$$\tau_{O,T}(b_{t-1}, a_{t-1}, o_t) = b_t(s') = P(s'|b_{t-1}, a_{t-1}, o_t) = \frac{O(s', a_{t-1}, o_t) \sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)}{P(o_t|b_{t-1}, a_{t-1})}.$$

The belief that agent's next state will be s' is calculated from the probability of observing o_t as $O(s', a_{t-1}, o_t)$ multiplied by probability of reaching the state s' given by $\sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)$. Denominator causes all believes sum to one. It is equal to $\sum_{s \in S} b_{t-1}(s) O(s, a_{t-1}, o_t)$.

Solutions

Agent's goal is receiving as much reward as possible. In this text we will assume, that the future rewards are less important. This can be modelled in MDP and POMDP cases as maximizing expected discounted reward over infinite horizon, i.e., maximizing $E[\sum_{i=0}^{\infty} \gamma^i r_i]$, where γ is a so-called discount factor. It is a real number $0 < \gamma < 1$ which is used to the weight future rewards. The higher discount factor the more important future rewards are. Modeling rewards this way is called discounted-reward model. Although we assume this reward model, our suggestions and results can be easily generalized to other reward models such as finite-horizon reward model.

The MDP-IP problem is slightly different because it contains another kind of uncertainty. The transition function is not known and the its choice is not necessarily random. Hence it cannot be described by the probability theory. It doesn't make sence to consider agent's expected discounted reward. However the expected discounted reward can be considered when using one additional assumption: The assumtion that the transition function which minimizes agent's reward is chosen at every time step. Although this assumption does not necessarily needs to be true, it can be usefull. When we would teach the agent to act well under this pessimistic assumtion then it would also act well when any other transition function would be used. That is the reason why we will use this assumtion from now on. We define the MDP-IP agent's goal to be also maximizing the expected discounted reward.

Policy π for MDP resp. MDP-IP is mapping $\pi : S \to A$ from states S to actions A.

POMDP agent's policy π is mapping from belief states to actions A.

MDP resp. MDP-IP agent can observe the state of the world for the whole time and it uses its policy π to choose the action at every time step.

In the POMDP case the agent does not know the exact state but it knows the belief state b which can be updated after each action a and observation o to $\tau_{O,T}(b, a, o)$. Again the POMDP-IP agent can follow its policy π at every time step.

Given policy π one can express expected discounted reward over infinite horizon as a function of initial state s resp. belief state b. We will call this function the value function for π and denote it V_π. It can be expressed as a unique solution of the following sets of equations.

MDP case: $V_\pi(s) := R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s')V_\pi(s')$

MDP-IP case: $V_\pi(s) := R(s, \pi(s)) + \gamma \min_{T \in K} \sum_{s'} T(s, \pi(s), s')V_\pi(s')$

POMDP case: $V_\pi(b) := \sum_s b(s)[R(s, \pi(b)) + \gamma \sum_o V_\pi(\tau_{O,T}(b, \pi(b), o)) \sum_{s'} T(s, \pi(b), s')O(s', \pi(b), o)]$

Explanation: Agent's immediate reward $R(s, \pi(s))$ resp. expected reward $\sum_s b(s)R(s, \pi(b))$ is added to discounted (multiplication by γ) expected future reward given by $\sum_{s'} T(s, a, s')V_\pi(s')$ in the MDP case. If the probabilities are imprecise the minimization over the transition probabilities returns the worst possible case reward. The expression is longer in the POMDP case because all possible current states and the states following are taken into account and the rewards are weighted by their probabilities which are given by observation and transition function.

These complicated expressions actually say how good each state resp. belief state is when the agent is following policy π.

Note that the recursive definition for V_π does not allow us to compute its values according to this definition directly. The values of V_π can be approximated by easy modification[1] of algorithm from section . But the algorithm for computing values of V_π is not our concern. We utilize the definition of V_π only for the definition of optimal policy.

Optimal policy

Optimal policy denoted π^* is defined as policy such that value function for π^* returns greater or equal value than any other value function does in the same state. $(\forall \pi \forall s \in S)\ V_{\pi^*}(s) > V_\pi(s)$ resp. $(\forall \pi \forall b \in \Pi(s))\ V_{\pi^*}(b) > V_\pi(b)$

Greedy policy π_V with respect to the value function V for MDP, MDP-IP is defined as $\pi_V(s) = argmax_{a \in A}[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s')V(s')]$.

Greedy policy π_V with respect to the value function V for POMDP is defined as $\pi_V(b) = argmax_{a \in A} \sum_s b(s)[R(s, \pi(b)) + \gamma \sum_o V(\tau_{O,T}(b, a, o)) \sum_{s'} T(s, a, s')O(s', a, o)]$.

Value functions

Optimal value function V^* is a mapping from states resp. belief states to real numbers defined as the following conditions called Bellman equations.

MDP: $V^*(s) = \max_{a \in A}[R(s, a) + \gamma \sum_{s \in S} T(s, a, s')V^*(s')]$

MDP-IP: $V^*(s) = \max_{a \in A} \min_{T \in K}[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s')V^*(s')]$

POMDP: $V^*(b) = max_{a \in A}[\sum_{s \in S} R(s, a)b(s) + \gamma \sum_{o \in \Omega} P(o|a, b)V^*(\tau_{O,T}(b, a, o))]$

In all three cases the set of equations has a unique solution [*Mausam et al.*, 2012]. These functions V^* can be intuitively interpreted as a functions of initial states s resp. belief states b returning expected discounted reward when the best possible action would be taken at every time step. The only difference between the definition of V^* and V_π is that the actions according to policy π were replaced by best possible actions. The functions V^* and V_π in the MDP-IP case return expected discounted reward under the assumption that the state changes according to the worst possible transition function.

Nevertheless much more important than intuitive explanation of V^* is its usefulness in the search for the optimal policy summarized by the following theorem:
Agent's goal is to find the optimal policy π^*. It can use the following theorem to do so: $\forall s \in S\ V^*(s) = V_{\pi^*}(s)$ resp. $\forall b\ V^*(b) = V_{\pi^*}(b)$.

As a collary agent can search for optimal value function V^* and use it to find the optimal policy π^* which is actually the greedy policy $\pi^* = \pi_{V^*}$ with respect to the optimal value function V^*.

Value iteration

Optimal value function can be approximated by using value iteration algorithm (1). In order to show only one algorithm for both fully observable and partially observable cases we denote both states and belief states by letter x. The x means state in algorithm for MDP and MDP-IP and it means belief state in the algorithm for POMDP.

[1] Replace $max_{a \in A}$ by $a := \pi(s)$ resp. by $a := \pi(b)$.

```
Data: $\epsilon > 0$
Result: count of iterations $t$
$V_t$ such that $[\max_x V^*(x) - V_t(x)] < \epsilon$
$t = 0$;
$\forall x V_0(x) = 0$;
repeat
    t = t + 1;
    compute $V_t$ from $V_{t-1}$
until $\max_x[V_t(x) - V_{t-1}(x)] < \epsilon\frac{1-\gamma}{2\gamma}$;
```

Algorithm 1: Value iteration algorithm

The update of the value function is computed as follows:
MDP: $V_t(b) = max_{a\in A}[R(s,a) + \gamma \sum_{s\in S} P(s'|s,a)V_{t-1}(s')]$
MDP-IP: $V_t(s) = max_{a\in A}min_{T\in K}[R(s,a) + \gamma \sum_{s'\in S} T(s,a,s')V_{t-1}(s')]$
POMDP: $V_t(b) = max_{a\in A}\sum_{s\in S}[R(s,a)b(s) + \gamma \sum_{o\in\Omega} P(o|a,b)V_{t-1}(b')]$ where $b' = \tau_{O,T}(b,a,o)$

$V_1(s)$ equals the best reward agent can get after performing one action from state s. $V_1(b)$ in the POMDP case is the best expected reward agent can receive after performing one action.

$V_2(x)$ is the best expected reward agent can get after performing two actions. As t goes to infinity, $t \to \infty$, V_t approximates V^* more accurately, $V_t \to V^*$.

The value iteration algorithm stops when enough[2] precision is reached.

Update of value function for POMDPs

Although we have a formula determining the way how each belief should be updated there is still an infinite number of believes. To store and update so many values of V_t more spare representation of this value function is needed. The value function V_t has some properties which can be utilized to do so, namely the value function is piecewise linear and convex. It is sufficient to keep in memory the coefficients for each hyperplane and update only these coefficients at each step.

The belief space has $|S|$ dimensions. So each hyperplane can be represented as vector with $|S|$ elements. The set Γ of hyperplanes α represents the value function as $V_t(b) = max_{\alpha\in\Gamma}\alpha(b)$. *Pineau et al.* [2006] shown how the value function represented as a set of hyperplanes can be updated to function from the next step which is again represented as a set of hyperplanes. It appeared that one of the key computational bottlenecks was a growing number of hyperplanes in Γ during time. Significant speed up of the value iteration algorithm was achieved by excluding hyperplanes which did not contribute to the maximization of above formula in any belief point.

Finding such hyperplane involves the linear optimization. Despite these speed-ups, looking for the optimal solution for POMDP is computationally complex. Specially it is complex when POMDP has many states hence when the belief space is many dimensional. That motivates searching the approximate techniques.

Roy et al. [2005] described a scalable approach which uses the low-dimensional representations of belief space. They used a variant of Principal Components Analysis (PCA) called Exponential family PCA in order to compress certain kinds of large real-world POMDPs and in order to find good policies for these problems faster. Another way to do the update of value function approximately was introduced by *Pineau et al.* [2006]. They speeded up the update of the value function by doing update not over the whole belief space but only in carefully selected significant points.

Handling imprecise probabilities in POMDPs

There was already an attempt to combine MDP-IPs and POMDPs. *Itoh et al.* [2007] tried to handle uncertainty about transition and observation models by defining partially observable Markov decision processes with imprecise probabilities (POMDPIPs).

They searched for the solution of problems in two following cases: 1. when the sets of possible models form polytopes, 2. when the sets of possible models are finite. They shown how to solve this kind of problems effectively but they used quite different definition of optimal solution than we did. They used so called second order believes to define probabilities over spaces of transition and observation models.

[2] If the maximal difference in values from two successive iterations is bounded by $\epsilon\frac{1-\gamma}{2\gamma}$ then the the result V_t differs from optimal value function V^* at most by ϵ. See for example *Kaelbling et al.* [2013].

Itoh et al. [2007] defined POMDPIP's policy as optimal whenever the optimal policy existed for any second order belief.

This definition of optimal policy is unintuitive. Recall our transportation example. The transition models depend on holidays regional events and other unmodelled factors. This uncertainty is encoded as a set of possible transition models. Similarly the observability can be different during the day and night.

According to definition of *Itoh et al.* [2007] we can think of any second order belief, e.g., belief such that the probability of the night observation model is 95% and the probability of the day observation model is 5% and then if we the found optimal policy for this second order belief we would call it the optimal policy for the whole POMDPIP.

In many cases as in the traffic example one would like to find the policy which performs well for all (or at least for most) possible transition and observation models rather than the strategy performing well only for some specific combination of these models.

Good choice of the action to take would be the action maximizing expected value for the worst possible choice of transition and observation model, i.e., action according to policy π_{V^*} where

$V^*(b) = max_{a \in A} min_T min_O \sum_{s \in S} [R(s,a)b(s) + \gamma \sum_{o \in \Omega} P(o|a,b) V^*(\tau_{O,T}(b,a,o))]$

Another possibility of overcoming imprecision of probabilities is taking expectation over all possible observation and transition models.[3] V^* could be defined as follows:

$V^*(b) = max_{a \in A} E_T E_O \sum_{s \in S} [R(s,a)b(s) + \gamma \sum_{o \in \Omega} P(o|a,b) V^*(\tau_{O,T}(b,a,o))]$

In both cases optimal value function could be approximated using the value iteration algorithm. Relation between V_t and V_{t-1} would be:

$V_t(b) = max_{a \in A} min_T min_O \sum_{s \in S} [R(s,a)b(s) + \gamma \sum_{o \in \Omega} P(o|a,b) V_{t-1}(\tau_{O,T}(b,a,o))]$

resp. $V_t(b) = max_{a \in A} E_T E_O \sum_{s \in S} [R(s,a)b(s) + \gamma \sum_{o \in \Omega} P(o|a,b) V_{t-1}(\tau_{O,T}(b,a,o))]$

We are again facing similar problem to the one we previously talked about. Again the continuum of values need to be updated effectively. The approach of *Pineau et al.* [2006] assumed convexity and piecewice linearity of value function.

The following paragraphs show that the minimization over finite transition and observation model spaces does not keep convexity of the value function.

Value function from tth time step can be seen as the expected reward the agent would get during first t steps when the world would act against it. By acting agains the agent we mean choosing the transition and observation models which are the worst for the agent. But the agent does not know its state precisely. It knows only probabalities of states given be its belief. The world chooses transitions and observations models according to this belief.

Let's illustrate everything on a simple example. Let the set of states $S = \{s_0, s_1\}$. Because we are in this example interested in updates of value function we will for simplicity consider only one-element set of states and set of action, $\Omega = \{o_1\}$, $A = \{a_1\}$. Because we have only one possible observation, the agent will observe this one observation every time. So we will consider only one observation model which assigns to this observation the probability 1. However we will consider uncertainty in transition model. There will be two possible transition models T_0 and T_1. To even more simplify the consideration we let both transitions models behave deterministically. The model T_0 never changes the state, i.e., the probabiliry of changing the state is 0 and the probability of staying in the same state is 1. The model T_1 always changes the state, i.e., the probability of staying in the same state is 0. Futhermore let $\gamma = 0.8$ and let the reward function be defined as follows: $R(s_0) = 0$ $R(s_1) = 1$

Because the set of states we consider has only two elements the belief space is only one dimensional. $b(s_1)$ is real number from interval $[0, 1]$ and $b(s_0)$ always equals to $1 - b(s_1)$. This allows us to show the plots of value functions.

The Figure 2a shows V_1 created by the first iteration of value iteration algorithm. The axis X shows $b(s_1)$. The axis Y shows the values of V_1 in b. It corresponds to the reward agent expects to receive after performing one action. It doesn't depend on chosen transition function.

However V_2 depends on transition function. The Figure 2b shows how V_2 looks in POMDP with transition model T_0. The Figure 2c shows how V_2 looks in POMDP with transition model T_1. If we constructed the value function the way that the world chooses the worst possible model then the V_2 looks like in Figure 2d. Hence it is not convex.

More generally the approximations V_t of value function can have more non-convex parts which are alternated by convex parts.

The cause of this phenomena is that in lower entropy belief states it can be easier to act against

[3] Since any prior probabilities of observation and transitions models are given, we can regard all models as the same probable.

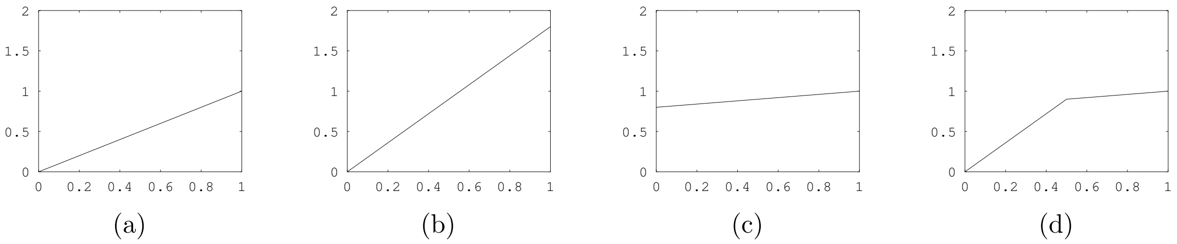


Figure 2. (a) Function V_1. (b) Function V_2 created with transition model T_0. (c) Function V_2 created with observation model T_1. (d) Influence of imprecise transition model.

the agent than in high entropy beliefs in which the agent has higher chance that the choice of transition and observations models will not affect it so much. This is a difference from POMDPs, where nothing similar could happen and where the value functions were always convex.

Another aspect which is needed to be taken into consideration when representing value function approximations in computer is the question whether they are piecewise linear, because it would be harder to represent function which is not piecewice linear. This apparently relates to question which sets of transitions and observations to allow, whether to allow models which contradict each other, wheather to allow infinite sets of possible models and if so how these sets should be restricted.[4]

The same questions hold also for the second proposed approach which replaced minimization by expectation.

Allowing only the sets of transition and observation models which keep some properties could lead to value functions which can be represented more easily. Otherwise novel techniques for representation and effective updates of more complicated value functions will be needed to be researched. Other branches of research could consist of approximative techniques.

Conclusion

POMDPs combined with MDP-IPs have greater representational power than the other frameworks because they combine the most essential features for planning under uncertainty. Naturally, the main drawback of optimizing a universal plan is complexity of doing so and effective techniques are needed to be discovered. This article showed that value functions of POMDPs extended by imprecise probabilities are not convex. This difference from conventional POMDPs prevents the use of existing algorithms. Effective techniques for solving POMDPs with imprecise probabilities are needed to be discovered.

Acknowledgments. The article [*Brunetto*, 2013] which was presented at ITAT2013 workshop was based on this article. It successfully passed referring process and is about to be printed. The contents of the article largely overlaps with this article. This research was supported by SVV project number 267 314.

References

Brunetto R., Probabilistic modeling of dynamic systems, *Informacne Technologie - Aplikacie a Teoria*, 2013, in print.

Delgado K. V., S. Sanner, and L. Nunes de Barros, Efficient solutions to factored MDPs with imprecise transition probabilities, *Artificial Intelligence*, 175, 1498-1527, 2011

Itoh H., and K. Nakamura, Partially observable Markov decision processes with imprecise parameters, *Artficial Inteligence*, 171, 453-490, 2007

Kaelbling L. P., M. L. Littman, and A. R. Cassandra, Planning and acting in partially observable stochastic domains, *Artificial Intelligence*, 101, 99-134, 1998

Kolobov A., Mausam, and D. S. Weld, Discovering hidden structure in factored MDPs, *Artificial Intelligence*, 189, 19-47, 2012

Mausam, Kolobov A., Planning with Markov Decision Processes: An AI Perspective

Moore A. W., L. P. Kaelbling, M. L. Littman, Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research 4*, 1996

Pineau J., G. Gordon and S. Thrun, Anytime Point-Based Approximations for Large POMDPs, *Journal of Artificial Intelligence Research 27*, 2006

Roy N., G. Gordon and S. Thrun, Finding Approximate POMDP Solutions Through Belief Compression, *Journal of Artificial Intelligence Research 23*, 2005

[4] Interesting idea would be considering the sets of models which are convex envelopes of finite sets of models.