Combining perfectly and partially observable MDPs

Milos Hauskrecht
Computer Science Department, Box 1910
Brown University
Providence, RI 02912

Working paper
January, 1998

Abstract

Standard Partially observable Markov decision process (POMDP) framework assumes a flat state space in which states are always considered to be hidden. However, this does not always fit many real-world problems that exhibit more structure and have states with both perfectly and partially observable components. Modelling explicitly such a structure then provides an excellent opportunity to reduce the complexity of associated problem-solving procedures. In the paper we describe a hybrid MDP framework that allows one to model processes with both perfectly and partially observable components. The framework relies on factored models in which process states are represented using state variables and the dynamics of the system using dynamic belief networks. We derive the Bellman's update formula for the hybrid framework and compare it to the POMDP case. Finally, we describe an application of the framework to the medical therapy planning and a hierarchical version of the hybrid MDP model used to solve it.

Introduction

Markov decision processes (MDPs) (Bellman 57) (Puterman 94)and Partially observable Markov decision process (POMDPs) (Astrom 65) provide useful theoretical frameworks for planning and control under uncertainty. MDPs allows one to represent uncertainty associated with the dynamics of the system under different interventions. POMDPs are more expressive in that they allow one two represent two sources of uncertainty: uncertainty related to the dynamics of the control process (same as MDPs), and uncertainty associated with the partial observability of the process by a control agent. The partial observability of the process is caused by an existence of noisy or imperfect observations very common in many real-world domains. Robot navigation, medical therapy planning and machine maintainance and replacement are typical examples.

The observability of the process state makes a large difference in solving optimization problems that try to find the sequence of control actions for all possible situations a control agent may encounter. While MDPs can be solved efficiently using dynamic programming techniques (see (Bellman 57), (Puterman 94)), POMDPs must be first transformed into a belief-state MDPs, which is a special type of a continuous-state MDP. Unfortunately, belief-state MDPs are hard to solve and only very small problems can be solved exactly in practice. This leads to various approximation methods that try to calculate good solution fast (see (Lovejoy 91), (Hauskrecht 97)). Thus the difference between perfectly observable and partially observable MDPs reflects the tradeoff between expressivity of the framework and computational complexity of the associated problem-solving procedures.

Classic POMDP models assume that the underlying process is hidden (partially observable). However, this does not always fit many real-world problems. The reason is that in many cases the problems exhibit more structure and one often faces processes with states that consist of both perfectly and partially observable components. Modelling explicitly such a structure then provides an excellent opportunity to improve and speed-up problem-solving procedures.

In our paper we describe a hybrid MDP framework that allows us to model processes with both perfectly and partially observable components, and we show how one can go about solving the optimization problems within it. The framework relies on factored models in which states are respresented using state variables and the dynamics of the system using the dynamic belief networks (see e.g. (Kjaerulff 92)). The factorization of the state space makes it possible to separate and represent both perfectly observable and hidden components of the process. In addition to this the dynamic belief network allows us to model explicitly additional structure (independencies that hold among state components), which often simplifies significantly the phase of model construction.

In the following we will first describe the basics of the hybrid MDP framework. Then we will derive the basic update (Bellman's) formulas for the new framework and point out similarities and differences compared to the POMDP case. Finally at the end we will describe an application of the framework to the medical therapy planning we used to test such ideas. The model built for the problem extends further the ideas of the hybrid MDP framework and allows one to handle MDPs with hierarchically structured sets of state variables.

Hybrid MDP

Hybrid Markov decision process model is a 5-tuple (S, O, A, D, R) where:

- $S = \{S_1, S_2, \cdots, S_n\}$ is a set of state variables describing the state of the system at any time step. Each state variable has two or more values, for example variable S_1 may have values: $\{s_1^1, s_1^2, \cdots, s_1^{m_1}\}$.
- $O = \{O_1, O_2, \cdots, O_k\}$ is a set of observable state variables $O \subseteq S$. Values of observable state are assumed to be known at any time step.
- A is a set of actions.
- D is a model of dynamic and describes the relation between state variables and states in two consecutive time steps, contingent on action choice. D is represented using a dynamic belief network (Kjaerulff 92) which effectively combines the decription of transition and observation models for standard POMDPs.
- R is a reward model and describes payoffs associated with occurence of various state variables or with transitions between the two states.

The main difference between the new model and standard MDP and POMDP models is that states are described in the factored form, and transition and observation models are represented together within the dynamic belief network D. The belief network captures the probabilistic relation between states and explicitly represents independencies and regularies that hold among state components. An example of dynamic belief network is in figure 1. The network represent the probabilistic relation between state variables for subsequent time steps. The network can be expanded to an arbitrary length by replicating its structure. Relations between state variables for a single time step are described using local belief networks.

The reward model can be described separately or in the factored form together with the model of the dynamics, using dynamic decision networks or dynamic influence diagrams (Tatman and Schachter 90). In such a model rewards are associated with state variables and combined using additive schemes. Using factored and structured models to represent both the model of the dynamics and the model of rewards usually simplifies and speed-ups the process of building the MDP for the application domain.

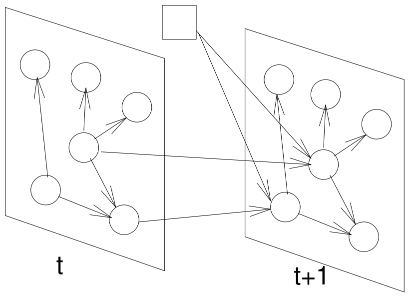

Figure 1: An example of a dynamic belief network. Circles denote state variables and a rectangle denotes an action. State variables for each state

Control objective

The goal of the planning problem is to find a policy (actions choices for all possible situations one may encounter) that optimizes a temporal objective function. An *objective (or value) function* quantifies control objectives by combining rewards obtained in multiple time steps, using various kinds of models. Typically, the value function is additive and based on expectations. For example, the function often uses:

- a finite horizon model that maximizes expected rewards for the next n steps: $\max E(\sum_{t=0}^{n} r_t)$, where r_t is the reward obtained at time t;
- an infinite discounted horizon that maximizes expected discounted reward: $\max E(\sum_{t=0}^{\infty} \gamma^t r_t)$, with $0 \leq \gamma < 1$ being a discount factor.

Solving the problem

The new model can be solved using a standard conversion in to the information state MDP. Let I_C represent a complete information state (Hauskrecht 97) that consists of a sequence of past observable state variable values, a sequence of past actions and a prior belief for initial states. Then the optimal value function for i steps-to-go and a complete information state is given by the Bellman's equation:

$$V_i(I_C) = \max_{a \in A} \left\{ \rho(I_C, a) + \gamma \sum_{I'_C} p(I'_C | I_n, a) V_{i-1}(I'_C) \right\}$$
$$= \max_{a \in A} \left\{ \rho(I_C, a) + \gamma \sum_{o \in \Theta_{next}} p(o | I_C, a) V_{i-1}(\tau(I_C, a, o)) \right\} \quad (1)$$

where $\rho(I_C, a)$ denotes one step expected reward resulting from performing an action a in I_C, Θ_{next} denotes a set of all possible combinations of values for observable state variables, I'_C is next complete information state for a and observation o (computed using an update function τ). For the complete information state, the update function τ simply adds a new action $a \in A$ and a new observation $o \in \Theta_{next}$ to I_C. The

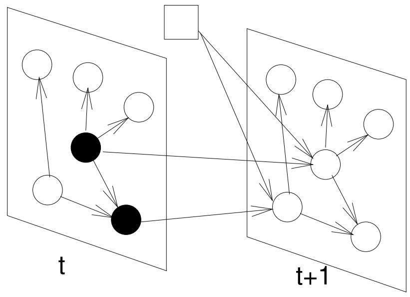

Figure 2: Example: periphery variables. Black circles denote periphery state variables.

optimal control (defines the policy) for the complete information state and i steps-to-go is:

$$\mu_i(I_C) = \arg\max_{a\in A}\left\{\rho(I_C,a)+\gamma\sum_{o\in\Theta_{next}} p(o|I_C,a)V_{i-1}(\tau(I_C,a,o))\right\}$$

The main problem with the above formula is that the description of the complete information state grows with every step. Luckily, the complete information state can be often replaced with sufficient statistics I that is of smaller complexity and defines an information process with a Markov property. The replacement is warrantied whenever it holds:

$$\begin{aligned} I' &= \tau'(I,a,o) \\ p(o|I,a) &= p(o|I_c,a) \\ \rho(I,a) &= \rho(I_C,a) \end{aligned}$$

The notion of sufficient statistics allows us to convert standard POMDPs to the belief state MDP (see e.g. (Hauskrecht 97)). However, in the standard POMDP case we face a monolithic state space, while in the factored case we have to deal with a set of state variables. One simple solution to the problem would be to take a belief over all possible combinations of state variable instances. However this is not very good and efficient and we can find often a far better sufficient statistics.

Process state variables

Let d be a set of all variables in S that directly influence the next state. In other words d consists of all current state variables that are also parents of one or more next state variables. We will refer to a set of such variables as to *exit periphery or process state* variables. An example of periphery state variables is illustarted in figure 2.

One of the properties of a periphery set d is that it d-separates (Pearl 88) past states from future states. Then knowing (observing) the values of such state variables the future will become independent of the past. Although in our case we deal with variables that may not be observable and thus not know, a set d can be still used to construct a sufficient statistic with the Markov property for the factored POMDP model.

A sufficient statistic for the factored POMDP model can be constructed by taking the belief over all possible combinations of values of state variables on the periphery[1]. Let d be a set of periphery variables, $\{h_1, h_2, \cdots, h_j, \cdots h_r\}$ be a set of all possible instances of peripheral variable values, and b_d be a belief over all such instances. Because components of a new belief state can be computed simply as:

$$b'_d(h_i) = \tau(b_d,a,o) = P(h_i|b_d,a,o) = \sum_{j=1}^{r} P(h_i|h_j,a)b_d(h_j),$$

and because:

$$p(o|b_d,a) = p(o|I_C,a),$$

$$\rho(b_d,a) = \rho(I_C,a),$$

also hold, the equation 1 (update formula) can be rewritten by means of beliefs over all possible instances of process state variable values:

$$V_i(b_d) = \max_{a\in A}\left\{\rho(b_d,a)+\gamma\sum_{o\in\Theta_{next}} P(o|b_d,a)V_{i-1}(\tau(b_d,a,o))\right\}.$$

The resulting formula is identical to the Bellman's update formula one would obtain for the standard POMDP. The only difference is that in the standard POMDP one works with a belief over all possible state instances while now we use beliefs defined only over periphery variable instances. In a case of large state variable space this approach leads to a significant reduction in the complexity of the information state process one has to use.

For the standard POMDP, the value function obtained after a finite number of dynamic programming or value iteration updates is piecewise linear and convex (Smallwood and Sondik 73). The same property holds for the hybrid MDP model with belief information states created for state variables on the periphery. Thus exact and approximation methods developed for the standard POMDPs can be directly applied.

Note that it is possible to construct a sufficient statistics using other state variables that d-separate future and past. However, in such cases the information state may need to include also observations not in the d-separating set and one has to make sure that

[1]Note that for the model with observation lags the sufficient statistics can change. See, for example (Hauskrecht 97) for the discussion of POMDPs with delays.

probabilities consistent with such observations are always used. In the following we will assume that sufficient statistic is always created using variables on the periphery.

Two-component information state

Using the belief over periphery states reduces the complexity of the information state compared to the naive case when belief over all possible state variable values was used. However the complexity of the information state can be cut further by treating separately observable and hidden process state variables. In a case of both observable and hidden process variables the information state corresponding originally to the belief state can be converted into a (hybrid) information state $I_d = \{o_d, b_d\}$ with two components:

- o_d — a vector of observable process variable values in d;
- b_d — a belief over all possible combinations of values of hidden variables in periphery d.

Treating observable and hidden variables separately allows us to reduce the complexity of the information state space and subsequently also the complexity of the planning procedures. The fact that we work and combine state components that are both perfectly observable and hidden, in principle combines perfectly and partially observable MDP frameworks and their advanatges. Note that standard POMDP assumes and uses a monolithic state space and thus states are always treated as hidden.

Value function for a two-component information state

Let o be a vector of all observations, $proj_d(o)$ be a projection of the vector o to the vector of process state observable values, and $\{h_1, h_2, \cdots, h_j, \cdots, h_q\}$ be a set of all possible combinations of values of hidden variables on the periphery d. Then the Bellman's update formula for the hybrid case can be written as:

$$V_i(\{o_d, b_d\}) = \max_{a \in A} \{\rho(\{o_d, b_d\}, a) + \gamma \sum_{o \in \Theta_{next}} p(o|\{o_d, b_d\}, a) V_{i-1}(\tau(\{o_d, b_d\}, a, o))\},$$

where the components of the next hybrid state $\{o'_d, b'_d\} = \tau(\{o_d, b_d\}, a, o)$ are computed as:

$$\begin{aligned} o'_d &= proj_d(o) \\ b'_d(h_j) &= \frac{P(h_j, o|\{o_d, b_d\}, a)}{\sum_k P(h_k, o|\{o_d, b_d\}, a)}. \end{aligned}$$

A two-component information state makes it possible to decompose the value function description. Now the value function can be represented as a set of partial piecewise linear and convex functions, one function for each possible value of o_d. That is:

$$V_i(\{o_d, b_d\}) = V_i^{o_d}(b_d),$$

where $V_i^{o_d}(b_d)$ is a partial value function for a vector o_d and equals:

$$V_i^{o_d}(b_d) = \max_{\alpha \in \Gamma_i^{o_d}} \sum_{j=1}^{q} \alpha(h_j) b_d(h_j),$$

such that $\Gamma_i^{o_d}$ denotes a set of linear vectors defining the partial value function $V_i^{o_d}$.

The Bellman's update formula for an observation vector o_d and a belief b_d can be rewritten also in terms of linear vectors defining V_{i-1}:

$$V_i^{o_d}(b_d) = \max_{a \in A} \left\{ \sum_{j=1}^{q} b_d(h_j)[\rho(o_d, h_j, a) + \gamma \sum_{o' \in \Theta_{i-1}} \sum_{k=1}^{q} P(h_k, o'|h_j, a) \alpha_{i-1}^{o_d, b_d, o', a}] \right\},$$

where $\alpha_{i-1}^{o_d, b_d, o', a}$ is a linear vector from $\Gamma_{i-1}^{proj_d(o')}$ maximizing:

$$\alpha_{i-1}^{o_d, b_d, o', a} = \arg\max_{\alpha_{i-1} \in \Gamma_{i-1}^{proj_d(o')}} \sum_{j=1}^{q} \sum_{k=1}^{q} P(h_k, o'|h_j, a) b_d(h_j) \alpha_{i-1}(h_k).$$

The expression in brackets corresponds to a new linear vector α_i optimizing the value function for $\{o_d, b_d\}$. This is reminiscent of formulas one would obtain for the standard POMDP and one can easily adapt various exact and approximation algorithms developed for the standard POMDPs to handle it (see e.g. (Lovejoy 91),(Cassandra 97), (Hauskrecht 97)). The only difference is that here we were able to decompose the information state and subsequently value function along process state observables and use separate belief spaces and value functions for each observable. This leads to the simplification of the computation process compared to the POMDP with a single belief space and value function.

Application to the medical therapy planning problem

We applied the new hybrid MDP framework to model a medical planning problem – the management of patients with chronic ischemic heart disease (Wong 90) (Hauskrecht 97). Here we focus more on modelling and complexity reduction issues. More detailed description of the application can be found in (?).

Management of the ischemic heart disease

Ischemic heart disease (IHD) is caused by an imbalance between the supply and demand of oxygen by a heart muscle. The condition is most often caused by the narrowing of coronary arteries (coronary artery disease) and an associated reduction in the oxygenated blood flow. The coronary artery disease tends to progress over time. The pace of the disease progress is stochastic and contigent on multiple factors.

At any point in time the physician has different options to intervene: do nothing, treat the patient with the medication, perform the surgical procedures (angioplasty, coronary artery bypass surgery), or one of the investigative procedures (angiogram, stress test) that tend to reveal more about the underlying status of the coronary disease. Some of the interventions has a low cost, but some carry a significant cost associated with the increased risk of death, or invasiveness of the procedure.

The objective of the therapy planning is to come up with a strategy that would minimize the expected integral cost of the treatment, where the cost is defined in terms of the death of the patient, quality of life (chest pain) over a period of time, invasiveness of surgical and investigative procedures and their economical cost.

Hybrid MDP for IHD

To model the dynamics and the decision process for the IHD domain we used a hierarchical refinement of the hybrid MDP presented in previous sections. The dynamic belief network representing built for the IHD is shown in figure 3.

In the IHD model, a state of the patient is described using a set of variables reflecting the status of the coronary artery disease, ischemia level, the occurence of acute myocardial infarction, and a set of observations (chest pain, rest EKG results, history findings, angiogram and stress test results). Some of the observations are unconditional and assumed to be available at any point in time (e.g. chest pain), while others are triggered (enabled) only with an appropriate action (e.g. stress test result is available only when stress test procedure is chosen and performed).

An interesting feature of the IHD model is that its state variable set is not flat, but hierarchically structured. More specifically, state variable *patient status*, with two values (dead or alive) enables (activates) state variables providing the more detailed description of the patient state only when patient is known to be alive. The reason for this is that evaluating values for such variables either does not make much sense or it does not influence the decision process.

The hierarchical variable structure can be used for two purposes. First, the fact that some of the state variables are enabled or disabled based on values of other variables effectively restricts the information state space one has to consider. Second, the variable structure can be used to represent new sets of indendencies within the dynamic belief network, which in turn may simplify its definition (by means of the number of paramaters one has to provide). For example, the transition probability between previous state (prs) and a state where the patient is alive and suffers from a severe coronary artery disease (CAD) is represented using two probability distributions as:

$$\begin{aligned} &P(status = alive, CAD = severe|action, prs) = \\ &\qquad P(status = alive|action, prs) \\ &\qquad P(CAD = severe|action, prs, status = alive). \end{aligned}$$

The first probability distribution concerns the variable *status* and represents the distribution of a patient being alive as a result of some procedure performed in the previous state. This distribution depends on values of state variables in the previous time step. The second distribution represents a conditional distribution of *coronary artery disease* given a previous state, an action and patient being alive. Such a distribution can exploit a different set of independences, i.e. the value of the *coronary artery disease* variable being independent of some previous state variables when the patient is alive. In general, the hierarchical belief network may represent different independences for different levels of details.

Extension of a hybrid framework to handle hierarchies

The hierarchies of state variables are best viewed as constraints on possible combinations of state variable values. This is because one can always assume a flat variable state space in which state variables have a new value ("inactive" or "don't care"), representing situations in which respective state variables are disabled. However, using the hierarchical structure we can always check which variables are disabled and need to be considered and no additional state value is needed. This can be implemented easily by ordering state variables with regard to their position in the hierarchy and allowing a state variable to be considered only when it is consistent with its predecessors.

Hierarchical state variable structures restricts a set of possible state variable combinations. This in turn can be exploited to reduce the size of the information state space one has to work with. For example, in the IHD problem, the set of all possible combinations of observable variable values on the periphery and the number of their associated hidden variable value combinations differ depending on the value assigned to variable *status*. A tree structure representing the information state space for the IHD problem is shown in figure 4.

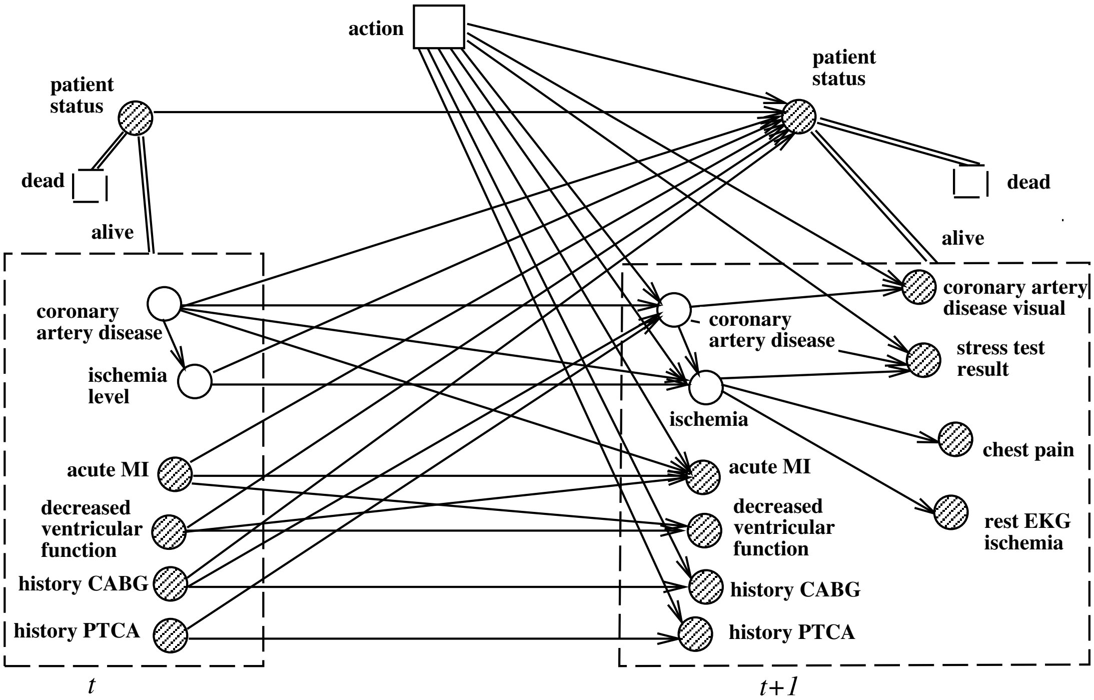

Figure 3: Model of the dynamics for the ischemic heart disease. State variables are represented by cicles, an action choice as a rectangle. Patterned circles are used for observables. For time t, only peripheral state variables are shown.

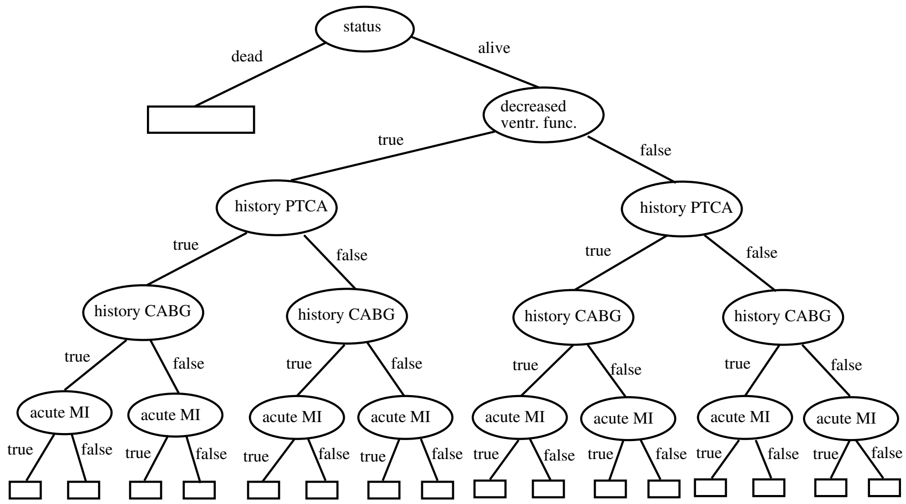

Figure 4: Information state space for the IHD problem represented by a tree structure. Internal nodes correspond to process state observables and leaf nodes to beliefs over all possible assignments to hidden process state variables.

In general, hierarchical constraints lead to heterogeneous information states, in which the size and complexity of both observable and belief components may differ. In the previous section we described the Bellman's update formula for the hybrid POMDP and a flat variable state space. These formulas can be adapted relatively easily to handle hierarchies of state variables and information states with components of different complexities. The key here is to respect hierarchical constraints and assume only state variables combinations consistent with them. In the implementation phase (we implemented few exact and approximation methods) we relied mostly on tree structures similar to the one in figure 4.

Benefits of the hybrid POMDP framework

The benefits of modelling the additional structure for the IHD problem are:

- definition of the POMDP model is simpler (one has to define smaller number of parameters);
- sufficient information state space is of smaller comlexity.

The information state space for the structured IHD model uses 17 possible combinations of observable process variable values: one for *status dead* and 16 for the alive state (four binary variables). All *status alive* combinations require an additional 9-dimensional belief space (all possible combinations of values for hidden-variables *coronary artery disease* and *ischemia*, each with three possible values). To illustrate the reduction in complexity, let us assume a flat process state space. Such a space does not allow to combine observable and hidden components and consists of 145 states (this figure counts only state variable combinations that are possible). As states are now assumed to be hidden, the information state space corresponds to a 145-dimensional belief space. The incorporation of observable variables thus reduces the complexity of the high dimensional belief state space to a set of belief spaces of small dimension. The overall number of observations is 1153 (all possible combinations of observable variable values) and it is same for both cases.

The above analysis illustrates savings from the model factorization and combination of observable and hidden process state variables. However, there are also savings that can be attributed to a hierarchy of state variables. Assume we use the flat state variable space, that is values for every state variable can be combined without restrictions. This would lead to the information state space with 32 combinations of observable variable values, compared to the hierarchical state variable space that allows only 17 different combinations.

Conclusion

In our work, we proposed a hybrid MDP framework with both perfectly and partially observable components. The model uses a factored state space and a dynamic belief network to define the dynamics of the underlying system. The hybrid model allows us to treat perfectly observable and hidden components of state separately, and effectively combines strengths of MDP and POMDP frameworks — expressiveness and model of partial observability from POMDPs and efficiency of problem solving procedures for the MDP. The other advantage of a factored model is that it usually speeds up the model construction process, as one has to usually define a smaller number of model parameters.

Sufficent information states for a hybrid MDP can be constructed using state variables on the exit periphery of a local belief network. This leads to the information states of smaller complexity as a number of exit peripheries can be significantly smaller compared to the overall number of state variables. Further reduction in the complexity is achieved by separating perfectly observable and hidden periphery variables. An information state for such a case consists of an observable component (a set of observations for observable periphery variables) and a hidden component (a belief over all possible combinations of hidden variable values on the periphery). This in turn simplifies the complexity of value functions and subsequently their computation.

The basic hybrid MDP model can be further improved by adding constraints that explicitly restrict some of the state variable value combinations. Note that in the standard POMDP case the state set is flat and there is no need for constraints (state is either included in the set or not). One possible approach to handle additional constraints is based on a hierarchical version of hybrid-MDP. The hierarchical version of the model was described and analyzed in connection with the application medical therapy planning.

The exploitation of the structure present in the problem is often the key to more efficient solutions. Hybrid MDPs try take advantage of the fact that problems often modeled using standard POMDPs rarely use states that are completely hidden. Other models and methods, exploiting the structure and subsequently improving problem-solving performance of POMDP problems are possible (see for example (Boutilier and Poole 95)). However, further exploration of different structured approaches, as well as their combination is needed.

References

K.J. Astrom. Optimal control of Markov decision processes with incomplete state estimation. Journal of

Mathematical Analysis and Applications, 10, pp. 174-205, 1965

E.R. Bellman. Dynamic Programming. Princeton University Press, 1957.

C. Boutlier, D. Poole. Computing Optimal Policies for Partially Observable Decision Processes. In the Proceedings of IJCAI-95, 1995.

A. R. Cassandra. Exact and Approximate Algorithms for Partially Observable Markov Decision Processes. Ph. D. Thesis. Brown University, Providence, RI, 1997.

M. Hauskrecht. Planning and control in stochastic domains wit h imperfect information. MIT-LCS-TR-738, 1997.

U. Kjaerulff. A computational scheme for reasoning in dynamic probabilistic networks. In the Proceedings of UAI-92, pp. 121-129, 1992.

W.S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. Annals of Operations Research, 28, pp. 47-66, 1991 .

J. Pearl. Probabilistic reasoning in Intelligent systems. Morgan Kaufmann, 1988.

M. Puterman. Markov Decision Processes. John Wiley and Sons, 1994.

R.D. Smallwood, E.J. Sondik. The optimal control of partially observable processes over a finite horizon. Operations Research, 21, pp. 1071-1088. cave)

J.A. Tatman, R.D. Schachter. Dynamic programming and influence diagrams. IEEE Transactions on Systems, Man and Cybernetics, 20(2), pp. 365-379, 1990

J.B. Wong et.al. Myocardial revascularization for chronic stable angina. Annals of Internal Medicine, 113 (1), pp. 852-871, 1990.