Artificial Intelligence in Medicine, vol 18 (2000), pp. 221–244 Submitted 5/99; Accepted 09/99

Planning Treatment of Ischemic Heart Disease with Partially Observable Markov Decision Processes

Milos Hauskrecht
Computer Science Department, Box 1910
Brown University
Providence, RI 02912
milos@cs.brown.edu

Hamish Fraser
Tufts-New England Medical Center
750 Washington Street
Boston, MA 02111
hamish@medg.lcs.mit.edu

Abstract

Diagnosis of a disease and its treatment are not separate, one-shot activities. Instead, they are very often dependent and interleaved over time. This is mostly due to uncertainty about the underlying disease, uncertainty associated with the response of a patient to the treatment and varying cost of different diagnostic (investigative) and treatment procedures. The framework of partially observable Markov decision processes (POMDPs) developed and used in the operations research, control theory and artificial intelligence communities is particularly suitable for modeling such a complex decision process. In this paper, we show how the POMDP framework can be used to model and solve the problem of the management of patients with ischemic heart disease (IHD), and demonstrate the modeling advantages of the framework over standard decision formalisms.

Keywords: dynamic decision making, partially observable Markov decision process, medical therapy planning, ischemic heart disease.

1. Introduction

The diagnosis of a disease and its treatment are not separate processes. Although the correct diagnosis helps to narrow the appropriate treatment choices, it is often the case that the treatment must be pursued without knowing the underlying patient state with certainty. The reason for this is that the diagnostic process is not a one-shot activity and it is usually necessary to collect additional information about the underlying disease, which in turn may delay the treatment and make the patients' outcome worse. This process may be even more complex when uncertainty associated with the reaction of a patient to different treatment choices and costs associated with various diagnostic (investigative) actions need to be considered. Thus, in a course of patient management one needs to carefully evaluate the benefit of possible diagnostic (investigative) and treatment steps and their ordering with regard to the overall global objective, the well being of a patient.

To model accurately the complex sequential decision process that combines diagnostic and treatment steps, we need a framework that is expressive enough to capture all relevant features of the problem. The tools typically used to model and analyze decision processes are (stochastic) decision trees [27]. Unfortunately stochastic decision trees are not always the best choice, especially when a problem domain is complex and long decision sequences need to be considered. The key drawback of stochastic trees is that they require a large number of parameters to be defined, and, thus, are hard to construct and modify. Also, their purpose is very narrow and it is hard to apply them to other tasks, e.g. predictions or explanations.

More compact decision models are typically used to alleviate the complexity problem. In general, these attempt to take advantage of the problem structure, most often, the time decomposability of the process and various forms of regularities. The model of choice in most cases is a Markov decision process (MDP) [3, 13, 26] and its refinements. The standard and widely-known MDP model – perfectly observable MDP — allows us to represent the dynamics and stochastic nature of the underlying process and captures nicely the uncertainty associated with the outcome of the treatment. However, it fails to capture processes in which a state describing the underlying disease is hidden (unknown) and is observed only indirectly via a collection of incomplete or imperfect observations. This feature is crucial for many therapy problems in which an underlying disease cannot be identified with certainty and thus more options need to be considered concurrently.

A framework more suitable for modeling the outlined therapy problem is partially observable Markov decision process (POMDP) [8, 2, 28]. A POMDP represents a controlled Markov process with two sources of uncertainty: stochasticity related to the dynamics of the control process (outcome of the treatment or diagnostic procedure is not deterministic), and uncertainty associated with the partial observability of the disease process by a decision-maker (the underlying disease state is observed indirectly via incomplete or imperfect observations). A POMDP model, like an ordinary MDP, is more compact and thus easier to build and modify, than a decision tree. Unfortunately, the modeling power of POMDPs comes with a price tag – the problem becomes computationally very costly and in practice, exact solutions can only be obtained for problems of small complexity. A challenging goal in this research area is to find and exploit additional structural properties of the domain and suitable approximations that perform well and can be used to obtain good solutions efficiently.

In this work, we apply POMDPs to medical therapy planning for patients with ischemic heart disease (IHD). To build and solve the problem, we use a factored version of a POMDP with a hierarchical dynamic belief network model of the disease dynamics. Dynamic belief networks [7, 5, 17] express additional regularities and independencies of the problem domain, which allow us to reduce even further the number of parameters needed to define the model. To reduce the computational complexity of problem-solving methods, we applied: (1) hybrid MDP-POMDP procedures taking advantage of perfectly and partially observable state components; (2) approximation (heuristic) methods, that trade off accuracy for speed. Using structure-based techniques and approximations, we were able to construct an IHD model of moderate complexity and successfully solve a number of cases. This is promising for further application of the POMDPs to this and other medical therapy problems.

2. Models of sequential decision making

2.1 Stochastic decision trees

The decision-making models used most often in practice are *stochastic decision trees* (see e.g. [27, 23, 21]). A stochastic decision tree represents possible choices of a decision-maker and their outcomes. A one step decision tree is illustrated in Figure 1. Here, rectangles represent decision nodes (or moves of the decision maker) and circles stand for chance nodes representing outcomes of actions (or moves of nature). As nature can behave stochastically an action can result in more than one outcome. The parameters of the tree are probability distributions of outcomes. [1] The leaves of a tree carry numerical values representing rewards or utilities for following the corresponding path.

The objective of a decision-maker is to choose an action that optimizes rewards (or utilities) associated with leaf nodes. The optimality criterion used is typically based on expectations; that is, the objective is to maximize expected reward of possible trajectories. [2] The optimal action at the

[1] Typically, outcomes correspond to observations and thus distributions can be estimated based on past experience.

[2] We note that other criteria might be used as well. For example, various risk-based criteria that take into account not only expectations but also higher moments can be used.

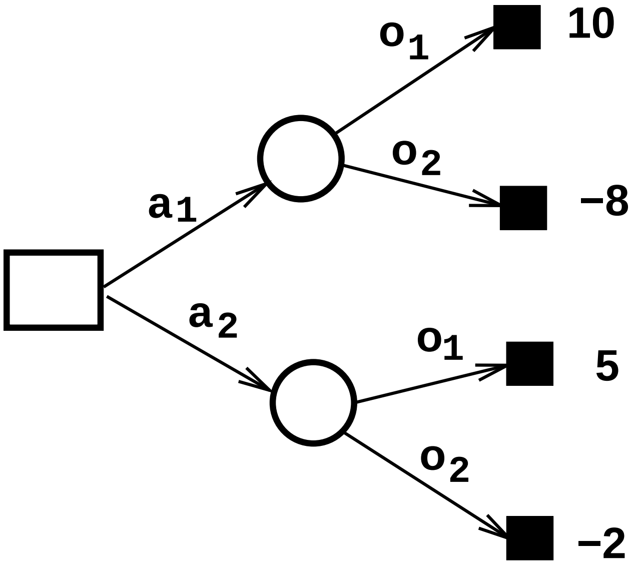


Figure 1: An example of a one-step decision tree. Rectangles correspond to decision nodes (moves of the decision-maker) and circles to chance nodes (moves of nature). Black rectangles represent leafs of the tree. Rewards are associated with every leaf (path) of the tree.

root of the tree (denoted s_0) then equals:

$$\mu^*(s_0) = \arg\max_{a \in A} E(r|s_0, a) = \arg\max_{a \in A} \sum_{o \in \Theta} P(o|s_0, a) r(\{s_0, a, o\}),$$

where A is a set of actions, r is a reward, o is the outcome (observation), Θ is a set of possible outcomes (observations) and $r(\{s_0, a, o\})$ is a reward associated with the path $\{s_0, a, o\}$.

The idea of a one-level stochastic decision tree can be refined into a multi-step model (Figure 2). In this model, the decision-maker can move more than once and outcomes of all the choices can be stochastic.

The optimal strategy for the multi-step tree is more complex and includes a sequence of choices which are conditioned on the outcomes of earlier steps. Let $V^*(x)$ denote the optimal (maximum) expected reward for a subtree starting at the decision node x. We refer to V^* as the *value function.* Given V^*, the optimal value and the optimal choice for any decision node y can be expressed in terms of its successor nodes as:

$$V^*(y) = \max_{a \in A} \sum_{x_i \in Next(y,a)} P(x_i|a, y) V^*(x_i)$$

$$\mu^*(y) = \arg\max_{a \in A} \sum_{x_i \in Next(y,a)} P(x_i|a, y) V^*(x_i),$$

where $Next(y, a)$ is a set of all successors of y and action a (see Figure 2). The value of a leaf node equals the reward associated with the corresponding path

$$V^*(x) = r(\text{path to } x).$$

The computation of the optimal strategy can then be performed backwards starting from the leaf nodes and proceeding towards the root of the tree.

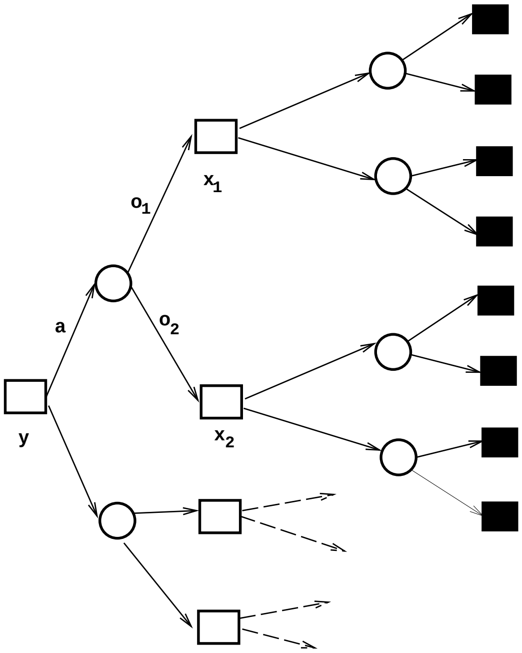


Figure 2: A two-step stochastic decision tree.

2.1.1 Modeling medical decisions using trees

In a stochastic decision tree, a reward score measures how good is a sequence of actions and their intermediate outcomes. Note that, under this interpretation, two different trajectories can be associated with two different rewards, even if the end-state of the system is the same. This is very important in medicine, where we must consider not only the end-outcome of a treatment, but also the means to achieve it. For example, different procedures carry different costs in terms of discomfort to the patient and potentially economic cost. Similarly, during the treatment a patient can end up in different intermediate states with various possible complications and with different level of pain, suffering and discomfort. Then, even if the end-state is the same, the paths are different and the history with more complications and more suffering should be penalized more.

2.1.2 Drawbacks of decision trees

A key problem one faces in applying the decision tree framework is to build the decision tree itself. As decision trees can become huge, the main issue that arises is how to assign probabilities and rewards to the tree. In the worst case, no simple solution to this problem exists and one has to deal with all the complexity. However, in reality, the problem often reflects more structure and regularities. An attempt to exploit this structure leads to various decision models that are simpler and often easier to handle. Examples of compact models are Markov decision processes [3] and influence diagrams [14](or their dynamic refinements [30]). In the following section, we focus on Markov decision processes, which represent a class of time-decomposable decision models.

2.2 Markov decision process

A *Markov decision process (MDP)* [3, 13, 26] describes a stochastic control process and formally corresponds to a 4-tuple (S, A, T, R), where S is a finite set of process states (e.g. patient states)

; A is a finite set of actions (e.g. diagnostic and treatment procedures); $T : S \times A \times S \rightarrow [0, 1]$ is a set of transition probabilities between states that describe the dynamics of the modeled system (e.g. disease); and $R : S \times A \times S \rightarrow \mathcal{R}$ denotes a reward (cost) model that assigns rewards to state transitions and models payoffs associated with such transitions. [3]

An MDP, once defined, can be expanded into a stochastic decision tree of an arbitrary depth. Simply, states at different time points correspond to decision nodes. The probability of reaching the next state under some action is then fully determined by the transition probability T in the MDP model. On the other hand, a reward for a specific action-outcome sequence (path) is constructed from partial rewards determined by R for that path. This allows us to incorporate costs and benefits of various procedures as well as intermediate outcomes. There are multiple ways to combine partial rewards into a value (global reward) for a path. The most typical are cumulative or average reward models. We focus on cumulative models. In this case, the reward for path of length T equals:

$$r_0^T = \sum_{t=0}^{T} \gamma^t r_t,$$

where r_t is a partial reward for time t and $0 < \gamma \leq 1$ is a discount factor. In the discounted version ($\gamma < 1$), the rewards for more distant future are considered to contribute less. [4]

Given an MDP and the cumulative reward criterion, the objective is to find actions maximizing $E(\sum_{t=0}^{T} \gamma^t r_t)$. This is a quite rich language to represent various objectives. For example, one can easily model goal-achievement tasks (a specific goal has to be reached) by giving large rewards for transitions to that state and zero or smaller rewards for other transitions.

Overall, MDPs allow us to express stochastic decision process in a more compact form compared to the decision tree which can grow its size exponentially in the horizon (time) of interest and would require to define as many parameters. In addition, an MDP model, once built, can be reused very easily for other tasks, for example, prediction of outcomes and explanation in terms of temporal dependencies.

2.2.1 Solving MDPs

The other crucial advantage of the MDP problem formulation comes when we need to compute the optimal decision choices. In particular, this process does not require to construct the full decision tree. Instead, the computation can be carried directly and more efficiently using *dynamic programming* techniques. The key here is that the decision tree for MDP consists of relatively small number of subtrees that can appear on different places, hence expanding and computing them separately over and over again leads to major inefficiencies. Dynamic programming [3] avoids these repetitions without the need to expand the tree.

Dynamic programming computes the value function (maximum expected reward) for all states $s \in S$ recursively using the Bellman's update [3]:

$$V_{i+1}^*(s) = \max_{a \in A} \left\{ R(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V_i^*(s') \right\}, \tag{1}$$

where V_{i+1}^* is a value function for $i+1$ steps to go, V_i^* a value function for i steps to go, and $R(s, a)$ is an expected one-step reward for state s and action a

$$R(s, a) = \sum_{s' \in S} P(s'|s, a) R(s, a, s').$$

[3] Costs are represented as negative rewards.

[4] The motivation for discounting comes from economics where the discount factor reflects the current interest rate. Simply, one dollar today is worth more than one dollar tommorow, the difference being the interest from holding the dollar in the bank.

Starting from V_0, which represents rewards for ending in different states, we can compute the value function for an arbitrary time horizon T, which then equals the number of steps. The optimal action choice for state s is

$$\mu^*_{i+1}(s) = \arg\max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*_i(s') \right\}. \tag{2}$$

As seen, this computation can be done efficiently in the size of the state and action space as well as time horizon T. In contrast, the computation in a fully expanded tree will be exponential in the time horizon T.

Finally, we note that the optimal value function and the optimal decision strategies can be computed efficiently also in the case when $\gamma < 1$ and the horizon $T \rightarrow \infty$ [26, 4]. Basically, the optimal value function for this case satisfies the fixed point equation

$$V^*(s) = \max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right\},$$

which is solvable using *value iteration* [3], *policy iteration* [13], or *linear programming* techniques [26, 4].

2.2.2 Modeling deficiencies of MDPs

Although MDPs offer a lot of very useful properties, they are often not sufficient for modeling more complex decision problems in medicine. The key problem here is that in using MDPs we have to assume that states defining the dynamics of the system are equal to observations, and that we know them (observe them) at any point in time. We also say that states of the system are *perfectly observable*. Unfortunately, the assumption of perfect observability is too strong for many practical planning problems. Essentially, it corresponds to the situation in which we know with certainty what is the disease (or complication) the patient suffers from at any point in time. [5] Also, if the disease is always known with certainty, there would be no need for investigative actions or procedures. This is in contrast to many medical problems in which investigative procedures are very common and play a major role.

2.3 Partially observable MDPs

A model that remedies the above disadvantages of perfectly observable MDPs and still preserves some of their good features, like time-decomposability and reduced model complexity, is *partially observable Markov decision process (POMDP)* [8, 2]. A POMDP model distinguishes between states defining the dynamics of the system (e.g. disease states) and observations, and thus, states can also be hidden and unobservable. In addition it allows us to handle investigative actions; that is, information gathering actions or actions enabling observations (e.g. a biopsy procedure allows us to see biopsy results).

Formally, a POMDP corresponds to a 6-tuple (S, A, Θ, T, O, R), where S is a finite set of process states (disease states); A is a finite set of actions (diagnostic and treatment procedures); Θ is a finite set of observations (findings, results of diagnostic tests); $T : S \times A \times S \rightarrow [0,1]$ is a set of transition probabilities between states that describe the dynamics of the modeled system; $O : S \times A \times \Theta \rightarrow [0,1]$ stand for a set of observation probabilities that describe the relationship among observations, states and actions; and $R : S \times A \times S \times \Theta \rightarrow \mathcal{R}$ denotes a reward (cost) model.

In POMDPs, process states are hidden, and decisions could only be based on observations seen and past actions performed. This makes a large difference when optimal action for all possible situations a decision-maker may face should be found. While in perfectly observable Markov processes

[5] Note that this is different from action-outcome uncertainty, where we are uncertain about the future outcomes of our actions.

one works with a finite number of states that are always known, in POMDPs underlying states are not known with certainty and one has to work with and base all decisions on belief states [2]. A belief state assigns a probability to every possible state $s \in S$, and there is an infinite number of possible belief states one may potentially encounter.

Similarly to MDP, a POMDP model can be used to generate a stochastic decision tree. In this case, decision nodes are associated with beliefs about the underlying (hidden) states and not states directly. A probability of seeing an observation o under action a for a belief state b is the parameter of the tree and can be calculated from the POMDP model

$$p(o|b,a) = \sum_{s' \in S} P(o|s',a)P(s'|s,a)b(s).$$

The next state belief, the one acquired after observation o and action a, is

$$b'(s) = \tau(b,a,o)(s) = \beta P(o|s,a) \sum_{s' \in S} P(s|s',a)b(s'), \tag{3}$$

with β being a normalizing constant. As before, the compact POMDP model can be used to generate a decision tree of any depth. However, we note that, by introducing hidden state variables, the probabilities associated with intermediate outcomes can be different even if the previous step observation and action are the same. Intuitively, the probability of an outcome (observation) at any point in time is given by our belief about the underlying state, which summarizes what we learned about the system in all previous steps.

2.3.1 Solving POMDPs

One of the advantages of the MDP model was that it allowed us to do the computation efficiently using dynamic programming techniques. Dynamic programming can be applied also in the POMDP case. The value function (the optimal expected reward) for $i+1$ steps to go satisfies the Bellman's update:

$$V^*_{i+1}(b) = \max_{a \in A} \left\{ R(b,a) + \gamma \sum_{o \in \Theta} P(o|b,a) V^*_i(\tau(b,a,o)) \right\}, \tag{4}$$

where b is a belief state, $R(b,a)$ denotes an expected one-step reward for a belief state b and action a and equals:

$$R(b,a) = \sum_{s' \in S} \sum_{s \in S} R(s,a,s')P(s'|s,a)b(s),$$

and τ is an update function that computes a new belief state b' using Equation 3. The optimal action for a belief state b is then expressed as:

$$\mu^*_{i+1}(b) = \arg\max_{a \in A} \left\{ R(b,a) + \gamma \sum_{o \in \Theta} P(o|b,a) V^*_i(\tau(b,a,o)) \right\}$$

The main complication in performing dynamic programming to solve a POMDP problem is that the belief state space is continuous. The key result in this respect is due to Sondik [29, 28], who proved that value functions for POMDPs are finite, piecewise linear and convex. More specifically, the value function for i steps to go has the form

$$V^*_i(b) = \max_{\alpha \in \Gamma^i} \sum_{s \in S} \alpha(s)b(s),$$

where α is a linear function and Γ^i is a finite collection of linear functions describing V^*_i. A piecewise linear and convex value function is illustrated in Figure 3.

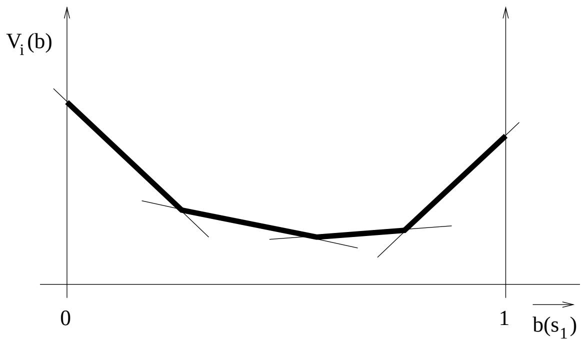


Figure 3: An example of a piecewise linear and convex value function for a POMDP with two process states $\{s_1, s_2\}$. Note that $b(s_1) = 1 - b(s_2)$ holds for any belief state.

The piecewise linear and convex property makes it possible to compute the update in a finite time for the complete belief space. Unfortunately, although computable, the computational cost for doing this is enormous [22, 18, 20] and, in general, only POMDPs of low complexity can be solved exactly in practice. The main problem here is that the complexity of value functions (in terms of the number of linear functions) can grow super-exponentially with the number of dynamic programming steps performed. To alleviate the problem, a substantial amount of research was directed to the exploration of various heuristic methods for calculating good approximations of the value function fast [25, 19, 10].

2.3.2 Combining dynamic programming and decision tree techniques

In general, we can apply two strategies to solve a POMDP: one constructs the decision tree first and then solves it; the other solves the problem in a backward fashion via dynamic programming. Unfortunately, both of these techniques are inefficient, one suffering from the exponential growth of the decision tree size, the other from the super-exponential growth of the value function complexity. The two techniques can be combined to at least partially eliminate their disadvantages. The idea is based on the fact that the two techniques work on the solution from two different sides (one forward and the other backwards) and the complexity for each of them worsens gradually. The solution is to compute the complete value function for k steps to go using the dynamic programming and cover the remaining steps by the forward decision tree expansion.

There are various modifications to the above idea. For example, one can often replace exact dynamic programming with two more efficient approximations providing upper and lower bounds of the value function. Then, the decision tree needs to be expanded only when bounds are not sufficient to determine the optimal action choice. However, for more complex problems, especially those with larger horizon T, the exact solution is hard to obtain and efficient approximations of the value function with some small forward lookahead are typically used.

3. Applying POMDPs to medical therapy planning

The expressiveness of the POMDP framework makes it suitable for medical therapy planning problems with hidden disease states, complex temporal cost-benefit trade-offs and both diagnostic and treatment procedures. In this work, we focus on the problem of management of patients with *ischemic heart disease (IHD)* [31] and its POMDP solution.

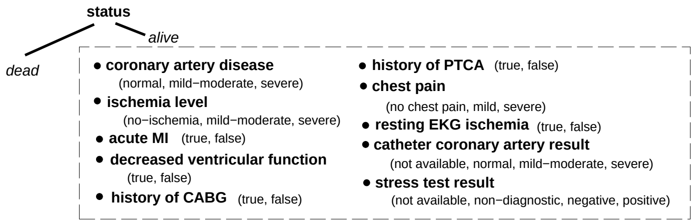

status
dead
alive
• coronary artery disease (normal, mild−moderate, severe)
• ischemia level (no−ischemia, mild−moderate, severe)
• acute MI (true, false)
• decreased ventricular function (true, false)
• history of CABG (true, false)
• history of PTCA (true, false)
• chest pain (no chest pain, mild, severe)
• resting EKG ischemia (true, false)
• catheter coronary artery result (not available, normal, mild−moderate, severe)
• stress test result (not available, non−diagnostic, negative, positive)

Figure 4: State variables for the IHD model and their values.

3.1 Management of ischemic heart disease

Ischemic heart disease is caused by an imbalance between the supply and demand of oxygen to the heart. 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 contingent on multiple factors.

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

The objective of the therapy planning is to develop a strategy that would minimize the expected cumulative cost of the treatment, where the cost is defined in terms of the dead-alive trade-off, quality of life, invasiveness of procedures and their economic cost. The optimal strategy depends not only on the immediate action choice, but also on future choices, thus reflecting complex temporal trade-offs.

3.2 POMDP model of IHD

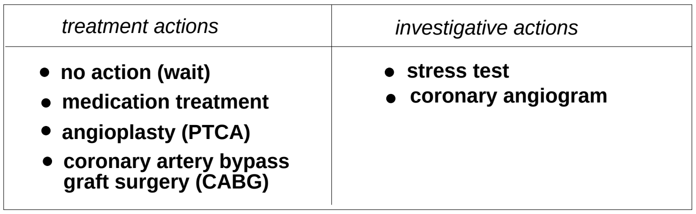
As previously pointed out, one of the main advantages of POMDPs is their compactness and the subsequent reduction in the number of parameters one has to define compared to the standard decision tree models. However, for more complex domains, like ischemic heart disease, even the definition of a POMDP model can become a very tedious task. Basically, if the state, observation and action spaces are large, the number of transition and observation probabilities we need to define can become extremely large and practically impossible. Thus, we seek further ways to reduce the complexity of the POMDP model by incorporating additional structure. In particular, we focus on factored POMDPs.

3.2.1 States, actions and observations

A state of the patient at any point in time is described using a set of state variables and their associated values. Figure 4 lists all state variables. An example of a state description is an *alive patient* with moderate *coronary artery disease*, severe *chest pain*, positive *rest EKG result*, etc.

A novel and interesting feature of our IHD model is that its state variable set is not flat, but hierarchically structured. More specifically, the state variable *patient status*, with two values (dead or alive), enables (activates) state variables providing more detailed description of the patient state (as e.g. *chest pain*) only when the patient is known to be alive.

State variables can represent both states as well as observations in the sense of a POMDP model, depending on whether they are hidden or observable. For example, variables representing status of



treatment actions	*investigative actions*
● no action (wait) **● medication treatment** **● angioplasty (PTCA)** **● coronary artery bypass graft surgery (CABG)**	**● stress test** **● coronary angiogram**

Figure 5: State variables for the IHD model and their values.

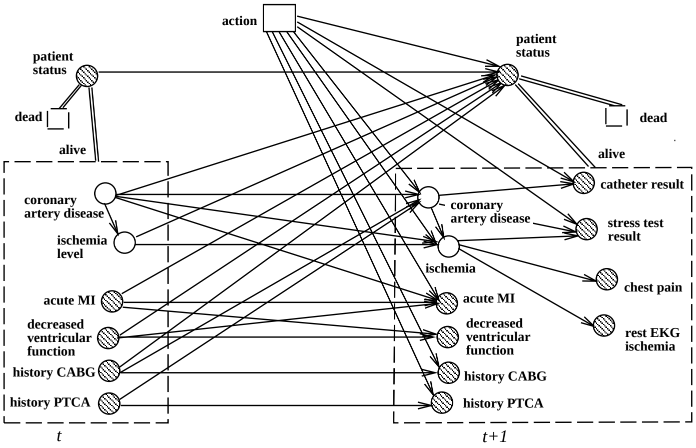


Figure 6: Model of the dynamics for the ischemic heart disease. State variables are represented by circles and an action choice as a rectangle. Patterned circles are used for observables. For time t, only state variables defining the transition are shown.

the *coronary artery disease* and *ischemia level* are hidden (not observable directly), while other state variables like *chest pain*, *rest EKG result* and *stress test result* are perfectly observable.

Actions correspond to treatment or investigative procedures (see Figure 5). *no-action* corresponds to the choice in which no treatment or investigative action has been selected. In general, treatment actions actively change the state of the patient to a more appropriate state. Investigative actions explore the state of the patient, especially the related hidden process state variables. However, investigative actions may not only reveal more about the underlying patient state, but can also lead to a change in the state (e.g. patient can die or get *acute MI* as a result of an *angiogram* procedure).

3.3 Model of dynamics

To represent the transition and observation models of the IHD problem more compactly, we use a hierarchical refinement of the dynamic belief network in Figure 6. The model represents independencies (marginal and conditional) that hold among state variables at two consecutive time steps. One of the advantages of the hierarchy is that it allows us to capture a new set of independencies within the dynamic belief network, which further reduce the complexity of the model and simplify its definition. For example, in our model, the transition probability between any previous state (labeled PRS) and a state in which the patient is alive and suffers from a severe coronary artery disease (CAD) is represented using two probability distributions:

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

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 independencies, i.e. the value of the *coronary artery disease* variable being independent of some previous state variables when the patient is alive. In general, our hierarchical belief network allows us to decompose the probability distribution and represent different structural independencies at different levels.

Note that some of the observations in Figure 6 are conditioned on actions. These correspond to the results of investigative procedures. For example, the value of *stress test result* is only available when a stress test procedure was chosen and performed in the previous step. In contrast to this, other observations are unconditional and assumed to be available at any point in time (e.g. *chest pain*).

3.4 Initial belief state

Once the model of the dynamics has been defined, we can expand it as many time steps as needed and use it to compute belief updates. However, to do this we first need to supply priors on all hidden variables at time $t = 0$. The prior probability distribution in the current version of the IHD model is fixed and corresponds to the target group of male patients, 65 years or older, non-smokers, and with no family history of coronary artery disease. We note that the current prior can be replaced by a more flexible model that targets different groups of patients and exploits other context information, for example, sex, age, smoking history, etc. There are logistic regression models developed for this purpose [1].

3.5 Reward (cost) model

The reward model is represented more compactly as well. The reward (cost) for a transition from state s to state s' under action a consists of two components:

$$R(s, a, s') = R(s') + R(a),$$

where $R(s')$ is a cost associated with a patient state only and $R(a)$ stands for a cost associated with an action (e.g. cost of performing coronary bypass surgery that includes the economic cost, patient's discomfort, and so forth). The cost associated with a patient state, $R(s')$, can be further broken down into individual state variable costs. That is, a cost for a patient state can be decomposed to costs associated with the amount of chest pain the patient suffers at a given time, occurrence of myocardial infarction, etc. In such a case, $R(s')$ can be expressed as:

$$R(s') = \sum_i R(s'_i),$$

where s'_i represents a value assigned to variable i. The above decomposition of the reward-cost model reduces the number of parameters we have to estimate, which greatly simplifies the model building process. The reward-cost values represent a combined measure of the dead-alive trade-off, quality of life and economic cost.

3.6 Objective criterion

To model the objectives of the IHD therapy planning, we use the cumulative reward model described earlier. In addition, we apply an infinite horizon ($T \to \infty$) and discounting. The objective function to be optimized is then $\max E(\sum_{t=0}^{\infty} \gamma^t r_t)$. This allows us to express longer term goals and not restrict the decision horizon to a finite number of steps. An interesting feature of our model is that we use transitions of different durations for different actions — transitions associated with surgical and investigative procedures occur within a day and transitions associated with non-invasive actions (*no-action* and *medication*) are assumed in 3 month periods. To account for this difference, we use discounting ($\gamma = 0.95$) for long-term actions (*no-action* and *medication*); all other short-term actions are undiscounted ($\gamma = 1$) and their costs are added fully within the model.

3.7 Acquisition of model parameters

One of the important problems associated with the ischemic heart disease model is to obtain a set of appropriate model parameters. The parameters define either probabilities or costs associated with state outcomes and actions. In general, these can be obtained by:

- acquiring them directly from the domain expert or from the literature;
- inferring them from the available data;
- or by using the combination of these two methods.

Obtaining such detailed probabilities from empirical data is difficult, requiring very large data sets. Therefore, we defined the parameters of the model by hand using published results supported by the experience of a cardiologist.

3.7.1 Transition and observation probabilities

To populate probabilistic transition and observation models, we primarily relied on Wong [31] who summarizes various studies in the area of chronic ischemic heart disease and compares outcomes for various interventions. For example, the probability of the patient staying alive or dying as a result of a surgical intervention can be estimated from mortality rates for a specific treatment and a specific patient condition. Similarly, one can obtain numbers for other parameters. For example, numbers reflecting the rate of change of the coronary disease under different interventions can be obtained from the published success rates of revascularization for PTCA and CABG. Unfortunately, in many cases the results of studies are presented independently for one or a few conditioning variables, leaving open the problem of how to deal with various combinations. In such cases, we either assumed independence, when it seemed reasonable, or adjusted probabilities by consulting a cardiologist. In general the process of defining probability parameters proved difficult and time consuming. The availability of large datasets should simplify the aquisition process and lead to more accurate parameter estimates.

3.7.2 Cost model

Unlike probabilities, costs are more subjective and reflect a combination of preferences of a physician, patient, etc. To deal with this issue, we have designed a new approach for acquiring parameters of the cost model for the ischemic heart disease.

The idea of the approach is to populate parameters of the cost-reward model by distributing a fixed amount of cost units among them. To simplify this process, we first construct a hierarchical structure defining the relations among model components, like state variables, their values and actions. Here we utilize also the hierarchical state variable structure already in place. The second step is to define a local weighting scheme that prescribes the proportion of the cost to be distributed

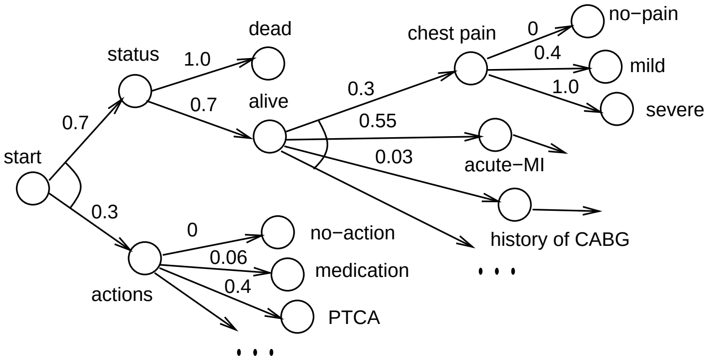


Figure 7: Model used for the acquisition of costs for the ischemic heart disease model. Links with arcs represent the *AND model*, links without arcs represent the *XOR model.* Numbers represent weights assigned to the lower level components.

to the lower level components of the structure from their predecessors. Using such a model, the cost for any component can be obtained in a top-down fashion, starting from the root of a hierarchy. Simply the cost for a component i is

$$C_i = w_i C,$$

where C represents the cost to be distributed and w_i is a weight associated with a lower level component i that satisfies $0 \leq w_i \leq 1$.

Part of the cost distribution model we built for the IHD problem is shown in Figure 7. We use two types of local weight models:

- *AND model*, represented by a parent-children combination with an arc on the outgoing links. It is used to distribute costs among complementary components, and its weights satisfy $\sum_i w_i = 1$, where i ranges over all lower level components. For example *status alive* is elaborated as lower level components *chest pain, acute-MI, history-CABG* and so forth that are complementary and each of them is assigned a portion of the cost accounted for by *status alive.*

- *XOR model*, represented by a parent-children subgraph without an arc. It is used to distribute costs to components that are exclusive. Subsequently, no restriction on weights is imposed. For example, the patient's *status* can be either *dead* or *alive* and the XOR model defines how much of the overall cost assigned to the patient state is accounted for by each alternative.

The key advantage of the model is that it simplifies significantly the whole cost-acquisition process. For example, we can ask the expert to quantify the cost associated with different severities of chest pain on scale $[0, 1]$, or ask the expert to quantify the importance of chest pain, acute MI, decreased ventricular function and so on, with regard to the cost. The numbers shown in graph 7 correspond to the weights we use for the IHD problem. They were defined by a cardiologist.

Once the weights of the cost distribution model are defined, we compute the cost (reward) for any component by simply multiplying weights for links on the path from the root of the distribution structure to a particular leaf node. For example, the reward for the severe chest pain is computed as:

$$R(\text{chest-pain-severe}) = -C_0 \cdot w_{\text{status}} \cdot w_{\text{alive}} \cdot w_{\text{chest-pain}} \cdot w_{\text{chest-pain-severe}},$$

where C_0 is the cost we expect to distribute and the w's stand for weights associated with different links. We used $C_0 = 100$ units for the IHD problem. This defines all parameters of the reward-cost model.

4. Solving the IHD problem

The POMDP, once defined, could be converted into the *belief-state MDP*, in which beliefs defined over all possible state variable value combinations are used as states. Then, the Bellman's equation 4 holds and either exact [29, 16] or approximate dynamic programming techniques [25, 19, 10] developed for the standard POMDPs can be applied.

4.1 Structural refinements

To solve the problem more efficiently, we can also take advantage of the additional structure present in the problem. We consider the following improvements to reduce the complexity of problem-solving procedures for IHD and similar medical problems.

4.1.1 Process state variables

The first improvement stems from the fact that not all variables representing the state of the patient at a given time are necessary to define the information belief state. For example, in the IHD problem, it is sufficient to use a belief state defined only over state variables that directly mediate transitions: *patient status*, *coronary artery disease*, *ischemia level*, *acute MI*, *decreased ventricular function*,
history of CABG and *history of PTCA*. We call these variables *process (or information) state variables.* In general, a state at any time is represented by a local belief network with many state variables. Process state variables represent their smaller subset which is consistent with the Bellman's equation and summarize all the information from previous steps. We denote a set of process state variables as d.

4.1.2 Hybrid information state and value function decomposition

The second improvement takes advantage of the fact that a process state variable can be both hidden and observable. In particular, when some of the process variables are observable, they should be treated that way and their exact values instead of beliefs over their values should be used. For example, in the IHD problem, four of the process state variables (*acute MI, decreased ventricular function, history of CABG* and *history of PTCA*) are assumed to be perfectly observable. In such a case the information belief state is better modeled as a hybrid state $\{o_d, b_d\}$ with two components: a vector of observable process variable values (o_d); and a belief over all possible combinations of values of hidden process variables (b_d).

As pointed out earlier, the value function for a POMDP is a piecewise linear and convex function defined over the complete belief state space. A nice property of a hybrid information state space is that the value function is decomposable into a collection of piecewise linear and convex value functions of smaller complexity, one function for each combination of perfectly observable process state variable values [12].

Let o be a vector of values of all observable variables, $o_d = proj_d(o)$ be a projection of the vector o to process state variables d, and $\{h_1, h_2, \cdots, h_j, \cdots, h_q\}$ be a set of all possible combinations of values of hidden process state variables. The value function for i steps to go can be rewritten as

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

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

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

where $\Gamma_i^{o_d}$ is a set of linear functions defining $V_i^{o_d}$.

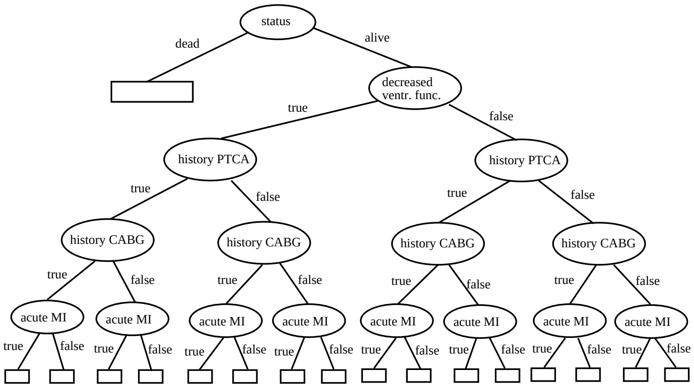


Figure 8: Decision tree representation of the information state space for the IHD problem. Internal nodes correspond to observables and leaf nodes to beliefs over all possible assignments to hidden state variables consistent with observables. The structure of the tree is sparse and reflects constraints imposed by the hierarchical variable set.

The fact that for each combination of observables the (partial) value function is piecewise linear and convex can be used to adapt both exact and approximation algorithms developed for the standard POMDP case also to hybrid circumstances. The main benefit is that the complexity of value functions is smaller, which in turn influences the complexity of problem-solving procedures. This step is best viewed as a combination of MDP-POMDP problem-solving methods and their advantages. In contrast to this, if we use a POMDP model with a flat state space, each (composite) state combines both observable and hidden components and we have to work with beliefs over all possible composite states. [6]

4.1.3 Exploiting additional state constraints

One drawback of the factored state representation is that it does not provide means for restricting combinations of state variable values that are not possible. [7] Thus, if we know that some combinations of values cannot occur, we need an additional model of constraints. In the IHD problem, we model these constraints by a hierarchically structured set of variables which explicitly restricts certain state variable combinations. In particular, when a patient is *dead*, values of other state variables are not relevant and a belief over possible combinations of their values is not considered. [8]

The hierarchy of state variables nicely extends our hybrid framework. In particular, the set of all possible combinations of observable process variables' values and the number of their associated hidden variables' value combinations may differ depending on the value assigned to variable *status*. Therefore, when *status* is dead, all other state variables are disabled and a hybrid information state corresponds to *status*=*dead* only (no other hidden or observable variables are used). On the other hand, when *status* is alive, other state variables become relevant and the hybrid state consists of a vector of all observable variables' values and a belief over all possible hidden variables' value combinations. In other words hierarchical constraints allow us to represent sparse information state

[6] We note that other structural refinements of a value function description are possible as well. For example, Boutilier and Poole [6] proposed and explored a method that uses compact representations of linear functions defining piecewise linear and convex value function.

[7] In contrast to this, if a POMDP with flat states is used, any redundant state can be directly excluded from the state space and not considered.

[8] The situation in which a state variable should be disabled can be always modeled using a distinct value *disabled*. However, in the case of a hierarchy, we can avoid modeling this value explicitly.

space with different observable and belief components. Such a state space can be represented more compactly, for example, using decision trees (see Figure 8).

4.2 Solution methods

To obtain the value function for the discounted, infinite horizon problem, we have implemented and tested the set of value function approximation methods proposed in [9, 10]. All methods were modified to handle additional structural refinements of the IHD model. To approximate the optimal action choices, we used one-step decision tree lookahead and the resulting value function approximations. Table 1 illustrates a sequence of recommendations for a single patient case with a follow-up obtained for two of the best-performing approximation methods from [10]. The first method is the incremental linear function method with 15 update cycles. Using this method, the value function for all possible belief states was computed off-line in about 25 minutes on a SPARC-10 using Lucid Common Lisp. The second method tested is the fast informed bound method and it took approximately 3 minutes. For every stage, the table shows a set of current observations and a list of possible actions, ordered with regard to the obtained cost score. The top (lowest cost) action is executed at each step. Note that the top choice for both methods is the same for all steps.

5. Evaluation

To test the IHD model we performed an initial evaluation on a set of 10 patient cases with follow-ups (similar to the one shown in Table 1). These cases where generated by a cardiologist with the objective to test the dynamic behavior of the model and identify its weaknesses. The program's analyses were critiqued by the cardiologist.

The IHD model we built is of moderate complexity and does not cover all the details of the problem domain. Interestingly, despite that and the need to estimate a large number of parameters, the model and obtained solutions demonstrated behavior that was in most instances clinically reasonable and justifiable. In particular, recommendations produced by the model were in almost all considered appropriate by the cardiologist.

The disagreements on patient cases helped us to uncover model deficiencies and lead us to further model improvement. These deficiencies were mainly attributable to model oversimplification and failure to represent all relevant details of the patient state. The main problem (accounting for most of the disagreements) we found was that the model required a variable representing the physical fitness of the patient, which influences the likelihood of reaching a non-diagnostic result for the stress test procedure. Omitting this variable lead in some instances to a repeated choice of the stress-test procedure even when the patient failed the procedure in the previous step. Simply, without representing the physical condition of the patient, the chance of a stress-test result failure was modeled using a random variable stress test result, with higher probability being assigned (based on population study) to one of the diagnostic outcomes. As a non-diagnostic result is not significant from the point of diagnosis, when it occurred, it did not affect our belief about the underlying disease. Therefore, the stress test was recommended again, because of its random outcome model and its expected benefit.

An interesting observation we made during the tests is that in a few instances the second best action choice in terms of scores (expected rewards) came very close to the leading choice. However, these situations, when analyzed, were typically situations where both choices were about equally good from a clinical point of view and “close” to each other, as suggested by the score. Overall these results suggest that the POMDP paradigm has real potential for solving this type of problem. Future evaluation with larger numbers of cases will be required after refinements of the model.

step	current patient status	actions	score (method 1)	score (method 2)
0	chest pain: mild-moderate;	**stress-test**	**285.22**	**248.53**
	acute MI: false;	no action	285.62	249.82
	rest EKG ischemia: negative;	medication	286.75	250.98
	decreased ventricular function: false;	PTCA	288.75	252.36
	catheter result: not available;	angiogram	292.92	256.68
	stress test result: not available;	CABG	491.94	427.77
	history CABG: false; history PTCA: false			
1	chest pain: mild-moderate;	**PTCA**	**298.47**	**262.54**
	acute MI: false	stress test	316.39	280.33
	rest EKG ischemia: negative;	no action	321.92	288.24
	decreased ventricular function: false;	medication	322.72	289.12
	catheter result: not-available;	angiogram	323.79	287.91
	stress test result: positive;	CABG	503.73	440.77
	history CABG: false; history PTCA: false			
2	chest pain: no chest pain;	**no action**	**259.07**	**226.23**
	acute MI: false;	medication	260.62	227.78
	rest EKG ischemia: negative;	stress test	264.35	229.87
	decreased ventricular function: false;	angiogram	273.34	239.16
	catheter result: normal;	PTCA	276.98	243.24
	stress test result: not available;	CABG	481.36	417.28
	history CABG: false; history PTCA: true			
3	chest pain: mild-moderate;	**medication**	**451.50**	**418.07**
	acute MI: true;	no action	452.81	419.47
	rest EKG ischemia: negative;	PTCA	464.58	429.87
	decreased ventricular function: false;	angiogram	470.62	435.62
	catheter result: not available;	stress-test	479.68	445.22
	stress test result: not available;	CABG	657.77	608.11
	history CABG: false; history PTCA: true			
4	chest pain: mild-moderate;	**PTCA**	**471.16**	**433.98**
	acute MI: false;	medication	483.11	447.85
	rest EKG ischemia: negative;	no action	485.15	450.04
	decreased ventricular function: true;	stress-test	486.32	448.75
	catheter result: not available;	angiogram	496.38	458.87
	stress test results: not available;	CABG	661.98	610.81
	history CABG: false; history PTCA: true			

Table 1: Patient case with a follow-up. Values of observables are shown for every step. Recommendations for each step are ordered according to the best cost score (the top choice is in bold). Two scores listed are computed by the incremental linear function method (method 1) and the fast informed bound method (method 2). Note that both methods suggest the same action choices.

6. Conclusion

The partially observable Markov decision process provides an elegant framework for modeling medical therapy planning problems with both action-outcome uncertainty and partial observability. POMDPs overcome some of the modeling deficiencies of alternative decision-making models, like standard stochastic decision trees or (perfectly observable) Markov decision processes. We successfully applied the framework to the problem of management of patients with ischemic heart disease (IHD), characterized by hidden disease states, investigative and treatment procedures, and temporal cost-benefit trade-off of these procedures and their outcomes. Beside our work [10, 11], the application of a POMDP framework to medical therapy planning has also been explored recently in [15, 24].

The POMDP model we constructed for the IHD problem uses a hierarchical Bayesian belief network to represent the disease dynamics, and a factored cost model to represent payoffs associated with treatment choices and intermediate outcomes. Such a model allowed us to take advantage of the regularities and specificities of the problem domain which greatly simplified the model construction process, and also lead to improved problem-solving performance compared to standard POMDPs.

The current IHD model is of moderate size and does not cover all aspects of the ischemic heart disease problem. For example, the model at this stage does not distinguish between left main stem and multiple vessel coronary artery disease and combines them into a single category — severe coronary artery disease. [9] Despite simplifications, the solutions obtained for the IHD therapy planning domain are promising and showed that POMDPs could provide a useful framework for modeling and analyzing the complex decision process. This justifies further refinement and extension of the current IHD model as well as, the application of the framework to other complex decision problems.

The key issues we need to deal with in order to further refine the IHD problem are the complexity of the model and the computational complexity of associated problem-solving procedures. An interesting research direction to address these problems is a combination of two decision models. Firstly a one-step (myopic) decision model, with a lower granularity and higher detail, capable of evaluating the immediate consequences of different action choices (for example, distinguishing among different locations of a coronary artery disease). Secondly a more coarse POMDP model focusing on the temporal trade-offs of the planning process. Values (expected rewards) computed for the POMDP model are then supplied as approximations of rewards to end-states in the myopic model.

7. Acknowledgments

We would like to thank Peter Szolovits and William Long for their valuable feedback and encouragement. This research was supported by the grant 1T15LM07092 from the National Library of Medicine and by DOD Advanced Research Project Agency (ARPA) under contract number N66001-95-M-1089.

References

[1] K.M. Anderson, P.M. Odell, P.W.F. Wilson, and W.B. Kannel. Cardiovascular disease risk profiles. *American Heart Journal*, 121:293–298, 1990.

[2] K. J. Astrom. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

[3] Richard E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[9] A drawback of this abstraction is that it leads to similar success rates for the CABG and PTCA procedures. When two types of diseases are considered individually, the success rates for the two procedures become different.

[4] Dimitri P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, Belmont, MA, 1995.

[5] Carlo Berzuini, Riccardo Bellazzi, and David Spiegelhalter. Bayesian networks applied to therapy monitoring. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 35–43, 1991.

[6] Craig Boutilier and David Poole. Exploiting structure in policy construction. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1168–1175, Portland, OR, 1996.

[7] Thomas Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150, 1989.

[8] Alvin Drake. *Observation of a Markov Process through a Noisy Channel.* PhD thesis, Massachusetts Institute of Technology, 1962.

[9] Milos Hauskrecht. Incremental methods for computing bounds in partially observable Markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 734–739, Providence, RI, 1997.

[10] Milos Hauskrecht. *Planning and control in stochastic domains with imperfect information.* PhD thesis, Massachusetts Institute of Technology, 1997.

[11] Milos Hauskrecht and Hamish Fraser. Modeling treatment of ischemic heart disease with partially observable Markov decision processes. In *Proceedings of American Medical Informatics Association annual symposium on Computer Applications in Health Care*, pages 538–542, Orlando, FL, 1998.

[12] Milos Hauskrecht and Hamish Fraser. Planning medical therapy using partially observable Markov decision processes. In *Proceedings of the Ninth International Workshop on Principles of Diagnosis (DX-98)*, pages 182–189, Cape Cod, MA, 1998.

[13] Ronald A. Howard. *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, 1960.

[14] Ronald A. Howard and J.E. Matheson. Influence diagrams. *Principles and applications of decision analysis*, 2, 1984.

[15] Chuanpu Hu, William S. Lovejoy, and Steven L. Shafer. Comparison of some suboptimal control policies in medical drug therapy. *Operations Research*, 44:696–709, 1996.

[16] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101, 1999.

[17] Uffe Kjaerulff. A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pages 121–129, 1992.

[18] Michael L. Littman. *Algorithms for sequential decision making.* PhD thesis, Brown University, 1996.

[19] William S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991.

[20] Christopher Lusena Martin Mundhenk, Judy Goldsmith and Eric Allender. Encyclopaedia of complexity results for finite-horizon Markov decision process problems. Technical report, UK CS Dept TR 273-97, University of Kentucky, 1997.

[21] Douglas K. Owens and Harold C. Sox. Medical decision making: Probabilistic medical reasoning. In E.H. Shortliffe and L.E. Perreault, editors, *Medical Informatics: Computer Applications in Health Care*. Addison Wesley, 1990.

[22] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12:441–450, 1987.

[23] Judea Pearl. *Probabilistic reasoning in Intelligent systems*. Morgan Kaufman, 1988.

[24] Niels Peek. Predictive probabilistic models for treatment planning in paediatric cardiology. In *Computational Engineering in Systems Applications*, Nabeul-Hammamet, Tunesia, 1998.

[25] Loren K. Platzman. *Finite memory estimation and control of finite probabilistic systems*. PhD thesis, Massachusetts Institute of Technology, 1977.

[26] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

[27] Harold Raiffa. *Decision analysis. Introductory lectures on choices under uncertainty*. Addison-Wesley, 1970.

[28] Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

[29] Edward J. Sondik. *The optimal control of partially observable Markov decision processes*. PhD thesis, Stanford University, 1971.

[30] J.A. Tatman and Ross D. Schachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 20:365–379, 1990.

[31] J.B. Wong, F.A. Sonnenberg, D.N. Salem, and S.G. Pauker. Myocardial revascularization for chronic stable angina. *Annals of Internal Medicine*, 113:852–871, 1990.