Planning and control in stochastic domains with imperfect information

Milos Hauskrecht
PhD thesis proposal - short version

August 27, 1996

1 Introduction

The construction of control agents functioning in the real world has become a focus of interests of many researchers in the AI community in recent years. This line of research was triggered by an attempt to benefit from advances and results in fields of data interpretation, diagnosis, planning, control and learning, and combine them into more sophisticated systems, capable of solving more complex problems.

What do we expect from a control agent?
The agent is expected to live in the world. It accomplishes goals and fulfills its intentions by observing and actively changing the world. In order to do so it can exploit the combination of perceptual, acting and reasoning capabilities. Examples of control agents can include: elevator movement control; robot arm controller; autopilot; medical life support device that monitors patient status and executes appropriate actions when needed.

The control agent interacts with the environment via actions and observations. Actions allow the agent to change the environment. On the other hand observations allow it to receive and collect the information about it. The control agent is designed to achieve some goal. In order to achieve the goal it coordinates its perceptual and acting capabilities: actions to change the environment in the required direction and observations to check the results of action interventions.

2 Two basic control agent designs

In the ideal case the control agent would perform the best possible sequence of actions leading to the goal satisfaction. In order to achieve the optimal or close to optimal control sequence the agent can be designed to either:

- follow hard coded and preprogrammed control sequences;

- use the agent's model of the world's behavior and the agent's goals and try to figure out (compute) the appropriate control autonomously.

The first design alternative is based on a simple idea of knowing directly what to do or how to respond in every situation. The idea, although simple and "unintelligent," can be the basis of a high quality control agent. The major advantage of this approach is that it can usually provide rapid control responses and thus may be suitable for various time critical applications. Its disadvantage stems from the fact that it relies on the external control plan source, and responsibility for the quality of the control is entirely on the shoulders of the control plan provider. This means that the external provider (usually human) must do the hardest part and "solve" the problem of how to achieve the goal considering every situation and encode this into the control plan. The other disadvantage of the approach can be that a control agent has no means to justify selected control responses with regard to goals, a feature that may be very important in some application areas.

The second alternative assumes that control is inferred by the control agent autonomously from the description of the environment behavior under different control strategies and goals to be pursued. In this case the responsibility for the quality of control is more on the side of the control agent itself and is mostly dependent on the design of its inference procedures, although providing wrong models can cause suboptimal control with regard to goals as well. The advantage of this approach is that the task of finding and selecting optimal control is performed by the controller autonomously and the external provider is required to supply only the appropriate models, a task that is usually far simpler than to provide complete control plans. Its obvious disadvantage is that the control response must be computed, which usually leads to longer reaction times.

In my work I will focus mostly on the model based alternative and explore the problem of computing control responses from models of the environment and goals. This will include both on-line control response computations as well as off-line computations of complete control plans.

3 Control in stochastic and partially observable domains

Models of environments and goals can be of different types and complexity. The model of the environment can be deterministic or stochastic, described using discrete or continuous states, discrete or continuous time, described by simple transition relations or by differential equations. The goal can be a simple state or it can be defined to vary over time.

The task of inferring the optimal control from models that are provided is largely dependent on the selected modeling framework and its complexity. The relation between the two is proportional: the more complex the modeling framework, the more complex is the associated computation of optimal control. Therefore with regard to control one must often trade off the benefits and costs of applying different models. That is, selecting a simpler model usually leads to simpler computation procedures for finding optimal control. On other hand by selecting a simpler model one usually uses a cruder approximation to reality, loses required precision and produces suboptimal control. Therefore one must carefully consider what features of reality to abstract away and which to consider.

3.0.1 Partially observable Markov decision processes

There are many reasonable modeling frameworks one can study with regard to various control problems. In my work I explore and study the framework of partially observable Markov decision processes (POMDP) (see [Astrom 65] [Smallwood, Sondik 73] or [Lovejoy 91a]) that allow one to model stochastic environments as well as their partial observability by the control agent. This framework has been studied by researchers from different areas, mostly in control theory and operations research and recently also by researchers in Artificial Intelligence. Within AI the POMDP framework is gradually becoming a basic formalism for planning under uncertainty with imperfect information.

The main features of the framework are:

- the world (environment) is described using a finite number of *discrete states*, and the control agent can actively change them using a finite number of *discrete actions*;
- the dynamics of the world is described using *stochastic transitions* between world states that occur in *discrete time* steps;
- information about the actual world state is not available to the control agent directly but via a *discrete set of observations*;
- the quality of control is modeled by means of numerical values representing *cost (or reward)* associated with state transitions;
- the *goal* is to optimize the expected costs collected over some time horizon.

More formally, a partially observable Markov decision process is defined as (S, A, Θ, T, O, C) where:

- S corresponds to the set of world states;
- A is a set of actions;
- Θ is a set of observations;
- $T : S \times A \times S \rightarrow [0, 1]$ defines the transition probability distribution $P(s|s', a)$ that describes the effect of actions on the state of the world;
- $O : \Theta \times S \times A \rightarrow [0, 1]$ defines the observation probability distribution $P(o|s, a)$ that models the effect of actions and states on observations;
- C correspond to the cost model $S \times A \times S \rightarrow R^+$ that models payoffs incurred by state transitions under specific actions.

The *decision (planning) problem* in the context of POMDP requires one to find an action or a sequence of actions for one or more perceived (information) states that minimize the expected cost incurred over some time horizon. A *perceived or information state* represents all information necessary for finding optimal control and may consist of the history of actions and observations or corresponding sufficient statistics. The overall *expected cost* to be minimized combines contributions of one step transition costs from the POMDP model. The most common decision (goal) criteria used are:

- finite horizon criterion: minimize expected cost for next n steps:

$$\min E(\sum_{t=0}^{n} c_t);$$

- infinite horizon criteria:

  1. minimize expected discounted cost:

$$\min E(\sum_{t=0}^{\infty} \gamma^t c_t),$$

     where $0 \leq \gamma < 1$ is a discount factor

  2. minimize average expected cost per transition:

$$\min \lim_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N} c_t;$$

- target state horizon criterion: minimize expected cost to some target state G.

The focus of my work is the exploration of optimization problems described using two of the above additive criteria: n-step-to-go finite horizon and infinite discounted horizon criteria. Typical applications of the POMDP problem would be robot navigation in noisy environments and management of an ill patient in the face of imperfect information about the patient's actual state.

4 Markov decision process

The POMDP framework is closely related to the more common and simpler formalism of Markov decision processes (MDP) ([Bellman 57], [Howard 60] or [Puterman 94], [Bertsekas 95]). The distinction between the two is that POMDP assumes and models partial observability of the controlled process while in the MDP framework the state of the system is assumed to be known at any point in time. Therefore MDP can be viewed as a special case of POMDP, in which the information state corresponds to the world state.

The perfect observability of MDPs, despite the action outcome uncertainty, makes the problem of finding optimal control significantly simpler compared to the partially observable case. The optimal control in the MDP case can be found using the following recursive formula:

$$V_n^*(s) = \min_{a \in A} \rho(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_{n-1}^*(s')$$

where $V^*(.) : S \rightarrow \mathcal{R}^+$ is a *value function* that stands for the minimum expected cost, s is a state at n steps to go, $\rho(s,a)$ is the expected transition cost of a from state s, γ is the discount factor and s' is at $n-1$ steps to go. The optimal control for state s and the n-step-to-go problem then corresponds to the action that minimizes the value function, i.e.:

$$\mu_n^*(s) = \text{argmin}_{a \in A} \rho(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_{n-1}^*(s')$$

where $\mu : S \rightarrow A$ stands for the *control function* that maps states to actions.

Similar recursive formulas can be derived for the infinite discounted horizon problem with a stationary control policy. The optimal value and control functions are:

$$V^*(s) = \min_{a \in A} \rho(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s')$$

$$\mu^*(I) = \text{argmin}_{a \in A} \rho(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s')$$

To find the optimal control for n steps to go one can adopt two general strategies:

- forward method that can be best viewed as an expansion of the decision tree and may be suitable for finding a one step control action for a single state;
- backward method that corresponds to dynamic programming [Bellman 57], [Howard 60] and is better when one needs to find the optimal control for all possible states and steps.

The optimal control decision for the infinite discounted horizon problem with stationary policy and single state can be solved, similarly to the finite horizon case, using forward decision tree methods. On the other hand optimal control for all possible states can be found via:

- value iteration [Bellman 57]
- policy iteration [Howard 60]
- or linear programming task (see e.g. [Puterman 94], [Bertsekas 95]).

Variations of basic methods can be found in [Puterman 94] and [Bertsekas 95]. The important thing, from the computational point of view, is that the problem can be formulated as a linear program, which implies that it can be solved efficiently in time polynomial in the size of the state and action spaces.

5 Computing optimal control for POMDPs

The best action (or sequence of actions) in the POMDP context is based on the computation of the minimum expected cost for an information vector (state) that reflects all relevant information about the controlled process at a specific time. The information state can be either complete history of observations and actions or corresponding sufficient statistics. This is different from the fully observable MDP where information state corresponds directly to the world state.

For the n-step-to-go problem the minimum expected cost for information state I_n can be computed using the following recursive formulas:

$$V_n^*(I_n) = \min_{a \in A} \sum_{s \in S} \rho(s,a) P(s|I_n) + \gamma \sum_{o \in \Theta_{next}} P(o|I_n, a) V_{n-1}^*(\tau(I_n, o, a))$$

where I_n is an information vector (state) at n steps to go and $\tau(b_n, o, a)$ is an update function yielding the information state at $n-1$ steps to go. The optimal control for state I_n and the n-step-to-go problem then corresponds to the action that minimizes the value function, i.e.:

$$\mu_n^*(I_n) = \text{argmin}_{a \in A} \sum_{s \in S} \rho(s,a) P(s|I_n) + \gamma \sum_{o \in \Theta_{next}} P(o|I_n, a) V_{n-1}^*(\tau(I_n, o, a))$$

where $\mu : \mathcal{I} \rightarrow A$ stands now for the control function that maps information vectors to actions.

Similar recursive formulas hold for the infinite discounted horizon problem with a stationary control policy. The optimal value and control functions are then:

$$V^*(I) = \min_{a \in A} \sum_{s \in S} \rho(s,a) P(s|I) + \gamma \sum_{o \in \Theta_{next}} P(o|I, a) V^*(\tau(I, o, a))$$

$$\mu^*(I) = \text{argmin}_{a \in A} \sum_{s \in S} \rho(s,a) P(s|I) + \gamma \sum_{o \in \Theta_{next}} P(o|I, a) V^*(\tau(I, o, a))$$

5.1 Exact optimization methods

The optimal control can be computed using the recursive optimization formulas. This may be done in a relatively straightforward way for some finite horizon problems with a single initial information state and a small number of steps using a forward decision method. However even in this case the number of information states one needs to visit grows exponentially with the number of steps to be explored and such problems were shown to be PSPACE complete [Papadimitriou, Tsitsiklis 87].

A far worse situation emerges when one is required to find the optimal or near optimal control solution for all possible information states. The main reason for this is that the information vector space is of infinite size and one must compute value and control functions defined over such space. Although this is theoretically possible thanks to the important result proved by Sondik [Smallwood, Sondik 73], that the value function for a finite number of steps is piecewise linear and concave, the computational complexity of available methods make them suitable only for small problems.

Methods that compute optimal value functions for a finite horizon problem are based on the dynamic programming approach. A step of such an approach computes a set of linear functions defining the optimal value function. This can be accomplished either by:

- generating all possible linear functions first and eliminating redundant ones afterwards, as for example in Monahan's method [Monahan 82]; or
- generating useful linear functions by evaluating and checking a finite number of points of the information state space, as in Sondik's method [Smallwood, Sondik 73], Cheng's linear support algorithm ([Cheng 88], see also [Cassandra 94]) and the Witness algorithm [Cassandra et.al. 94] [Cassandra 94].

The major source of inefficiency of all methods is that the number of piecewise linear regions defining the value function can grow exponentially with the number of steps. Moreover all algorithms implementing one step of dynamic programming can be inefficient in computing a useful set of linear vectors. In general this problem can be solved efficiently only when $RP = NP$ [Littman et.al. 95c].

Methods developed for the computation of the optimal value function for the finite step problem can be reused to compute approximations of the optimal value function for the infinite discounted horizon problem. This is because one can implement value iteration method, that is guaranteed to converge to optimal value function solution, simply by using updates corresponding to one step of the dynamic programming. This is best viewed as the standard value iteration algorithm with parallel updates. A slight modification of the update procedure can implement more efficient Gauss-Seidel analogue of the value iteration procedure.

The fact that methods producing complete optimal policies are suitable only for small problems naturally leads to the exploration of alternative control optimization solutions. One possible and straightforward approach is to implement the controller using a simpler decision making module that tries to select an optimal control action for a state. Such a module is then reinvoked at every decision step. The control problem that the module solves repeatedly then corresponds to the decision problem. Although the problem is in PSPACE, it is often easier and sufficient to solve it repeatedly rather than to solve the complete all state problem. Not much attention has thus far been devoted to the exploration of methods addressing this problem.

5.2 Approximation methods

The problem of computational efficiency of available optimization methods leads naturally to the exploration of various approximation methods and shortcuts that allow one to acquire good solutions with less computation. There are many different approaches one can use for this task and they can be applied to solve both decision and policy problems. Most of the approximation methods were developed by researchers in control theory or operations research over many years of work, and one can hardly expect a completely new approach to emerge quickly. Although the basic ideas tend to repeat in recent work, there is still a lot of room for various modifications and extensions.

In general, approximation methods can be divided into two groups (see attached text):

- approximation of value or control functions;
- approximation of sufficient information states.

The main idea behind the first approach is to approximate the value (action-value) or control functions that are defined over the sufficient information vector space with functions $\widehat{V} : \mathcal{I} \rightarrow \mathcal{R}^+$ or $\widehat{\mu} : \mathcal{I} \rightarrow A$. These functions are usually computed by exploring only a finite number of information vectors (points) of the complete information vector space; they have a finite description and very often also a simple form.

The second approach focuses on the approximation of sufficient information states with feature vectors $\widehat{I}$ and assumes that value and/or control functions defined over feature vectors $\widehat{V} : \widehat{\mathcal{I}} \rightarrow \mathcal{R}^+$ and $\widehat{\mu} : \widehat{\mathcal{I}} \rightarrow A$ are far easier to compute. The approximate value or control function for a specific information state is then computed from the associated feature vector. The feature space is usually of smaller size and summarizes the important characteristics of the world state with regard to the control problem.

The two approximation approaches are not exclusive and can be combined. This leads to both approximation on the level of information state space and on the level of functions defined upon

such an approximate space.

5.2.1 Approximations of value functions

The most common approximations target value functions. In this case once the approximate value function $\widehat{V}$ is computed, the corresponding control can be acquired simply via solving a recursive formula; e.g. for the infinite discounted horizon case:

$$\widehat{\mu}(I) = \text{argmin}_{a \in A} \sum_{s \in S} \rho(s,a) P(s|I) + \gamma \sum_{o \in \Theta_{next}} P(o|I,a) \widehat{V}(\tau(I,o,a))$$

There are numerous methods that can be tried to compute value function approximations. The basic methods include:

- MDP approximation;
- blind policies approximation;
- point-based approximations:
  1. curve fitting (least square error);
  2. memory based (instance-based) approximations (with point interpolation/extrapolation rules);
- restricted Sondik's method;

Approximations based on the optimal MDP solution or blind policies are used to compute general value function bounds. They are computed with relatively simple procedures that completely ignore uncertainty related to imperfect observability and work only with the perfect world states.

All other value function approximation methods try to account for the imperfect observability by using a finite number of points of the infinite information vector space corresponding to sufficient statistics and by restricting the associated value function in various ways. Point-based methods are based on standard function approximation approaches, that are usually combined with dynamic programming or value iteration methods. Methods sample the infinite information state space and use acquired samples to either learn parameters of the value function model (curve fitting) [Bertsekas 95] or store and use them directly to derive function values at points not seen before, using various interpolation-extrapolation rules (memory-based) [Lovejoy 91a] [Lovejoy 91b] [Lovejoy 93]. The restricted Sondik method (see, e.g., [Lovejoy 91b] [Lovejoy 93]) is similar to exact methods and works with linear functions rather than values. However the method does not search for the points of the information state space that seed all useful linear functions and uses a set of non-optimized points.

5.2.2 Approximations of information states

The complementary class of approximation methods aims to reduce the complexity of the original POMDP problem by approximating the sufficient information state space. The reduction is achieved

by substituting sufficient information vectors by a *feature state space* [Bertsekas 95] that is of smaller size, summarizes the important characteristics of the information state with regard to control and is easier to manipulate and work with.

The relation between the information and feature vectors is captured by a *feature extraction mapping* $\mathcal{F}$, that maps information states to feature states, i.e.:

$$\mathcal{F} : \mathcal{I} \rightarrow \widehat{\mathcal{I}}.$$

Then assuming the feature-based value or control function are known:

$$\widehat{V}_{\mathcal{F}} : \widehat{\mathcal{I}} \rightarrow \mathcal{R}^{+}$$

$$\widehat{\mu}_{\mathcal{F}} : \widehat{\mathcal{I}} \rightarrow A$$

one can express approximate value or control functions for information state I as:

$$\widehat{V}(I) = \widehat{V}_{\mathcal{F}}(\mathcal{F}(I))$$

$$\widehat{\mu}(I) = \widehat{\mu}_{\mathcal{F}}(\mathcal{F}(I)).$$

The important property of feature based methods is that the definition of the smaller feature space introduces a bias. That allows one to incorporate prior domain knowledge by telling what is relevant and needs to be considered and what can be abstracted out. This is unlike previous approximation methods that were based on either random sampling of the information vector space or at most utilized some very general heuristics. Because of the prior knowledge, the feature based methods are suitable especially for problems with large state spaces. The problem of finding a relevant feature space automatically from the POMDP problem description has not been addressed by the researchers so far and remains an open problem.

5.2.3 Evaluation of approximation methods

There is a relatively large number of methods that can be used to compute approximate control. Unfortunately there has not been, to our knowledge, a good summary and comparison of properties and performance of various approaches made so far. The studies that have been done were, most of the time, oriented toward specific methods and evaluation of their performance and did not include comparison with other approaches. Although there have been some recent cross evaluation studies and comparisons of different methods, as in, e.g., [Littman et. al. 95a] [Parr, Russell 95], these were based mostly on exact methods, simple MDP based approximations and/or gradient descent approaches.

This suggests that more evaluation studies, comparing different approaches, need to be conducted. The alternative to this will be to prove theoretically some properties that will allow one to differentiate between alternate approaches. Although it is unlikely that one single approximation method will turn out to be the best all the time for any problem, it can be the case that some approaches are better than others on problems with certain characteristics. The finding of these is an open problem.

6 Applying the POMDP framework

POMDP algorithms are usually tested on small problems, many of them toy like and far from real world problems. The lack of larger scale real world applications that use the POMDP framework and their comparison to other alternative approaches cast doubts on practical usability of the approach. This prompts more real world applications that can prove and justify the place of the formalism in solving practical problems.

The application of the framework to real world problems is also closely related to the problem of framework expressiveness. This is because any real world problem needs to be expressed in the language underlying the formalism and it may lead to the following difficulties:

- the underlying standard POMDP formalism is too restrictive;
- the acquisition of appropriate model parameters can be hard.

The standard POMDP formalism assumes that observations associated with some state are available immediately after the exploratory action is performed. This is not always the case; e.g., in medicine where exploratory action and the results of such observation may not be available immediately but only after some delay. For example, the results of a blood test may become available only after several hours of work in the clinical laboratory. To handle such features of the problem properly, it is often necessary to go beyond the standard formalism and consider various modifications and extensions.

The other problem associated with the exploitation of the POMDP framework in real world domains is related to the acquisition of the model, especially of model parameters. Model parameters correspond to probabilities and costs or rewards associated with transitions or observations. Such parameters can often be hard to acquire from a human directly, especially those representing costs (or utilities) associated with transitions and reflecting preferences of the designer. Therefore any tool facilitating the acquisition of the model or its parameters can be of great help.

7 Thesis objective

The objective of my thesis research work is to explore various aspects of the POMDP framework and address some of the problems that were outlined above. These include topics such as:

- extensions and modifications of the standard POMDP framework;
- design of exact and approximate methods for solving different POMDP planning problems, exploration of their properties and their comparison;
- application of the framework in control and decision making systems in medicine;
- learning of model parameters or policies from data.

8 Current status

In the following I will describe the current status of my research work on the outlined thesis objectives and results accomplished so far. Later I will focus on the problems I plan to address in the future.

8.1 Extending basic POMDP framework

The standard POMDP model assumes that observations are always conditioned on actions performed in the previous step and pertain to the current state of the world. Such a model enjoys some nice properties: e.g., the current state is sufficiently represented using the belief space and the optimal value function is piecewise linear and concave for a finite n-step horizon. However the conditioning model defining dependencies between observations, states and actions need not be sufficient and appropriate for all domains. Therefore one part of my work has been devoted to the exploration of alternative models defining the relation between model components. Using the notion of sufficient information vector we showed that one can fully represent the perceived state POMDP with backward triggered observations as well as a richer model combining forward (standard) and backward triggered observations using belief vectors. Moreover it was shown that the value function is piecewise linear and concave whenever the sufficient information vector corresponds to a belief state, thus extending the result of Smallwood and Sondik to all such models. This effectively makes all algorithms developed for the standard model applicable for the whole class of models after appropriate modifications in the belief update procedure.

There are other extension of the POMDP modeling framework one can consider and incorporate into the standard framework. These include various delays in observation and action channels. In general the solution formulas for such extensions can be written using sufficient statistics. In my work I have shown how one can go about describing the POMDP model where observations are delayed at most k time units. Unfortunately the nature of sufficient statistics for general delayed models cause the solution value function not to have the properties of piecewise linearity and concaveness that are typical for models with sufficient belief states. This makes it impossible to solve them by computing the optimal value function using dynamic programming or value iteration techniques and one needs to use either forward decision tree or various value function approximation methods to solve them.

8.2 POMDP decision tree algorithms

Most of the attention of researchers in the area has been devoted to the problem of finding the optimal policy. However in many cases the far simpler decision problem that tries to select a control response from the single initial state can be sufficient for implementing the control agent. Such problems can be solved in the forward fashion by a process corresponding to the expansion of the decision tree. In our work we have proposed, designed and implemented various incremental algorithms: breadth first, randomized heuristic, and linear space for solving such problems. These methods try to reduce the growth of the decision tree via pruning based on computed value function bounds.

The breadth-first algorithm always expands all nodes on the fringe of the partially constructed tree, employing pruning to cut off the suboptimal branches. The heuristic expansion algorithm, unlike blind breadth-first expansion, tries to expand those regions of the decision tree that have the

largest potential to induce pruning or achieve required solution precision. The heuristic potential is measured by a span between the upper and lower bounds of the value function for a specific node. We have implemented a variant of the heuristic method, called the randomized heuristical method, that tries to expand more than one branch of the decision tree in one expansion step. The branches chosen for expansion are selected randomly in proportion their heuristic value. Theoretically interesting is the linear-space algorithm that exploits active span heuristics and computes bounds in the iterative deepening fashion. This algorithm uses space that is linear in the number of observations, actions and minimum depth of the decision tree needed to select the optimal action.

The quality of incremental forward algorithms is strongly dependent on the quality of supplied value function bounds. In general, the tighter the bounds the better is the chance to prune suboptimal branches. The typical property of incremental algorithms is that they rely on the ability to improve bounds via expansion of the partially expanded decision tree. This is crucial as not all value function bounds supplied need to induce such improvements for all possible information states. The value function approximations that always induce improvement via forward expansion are said to satisfy *recursive improvement property*. In my work I have explored, described and designed various methods for computing value function bounds. These in general correspond to various approximation methods and range from simple MDP-based bounds suitable for generic POMDP models to more complex methods based on, e.g., interpolation-extrapolation rules.

Simple incremental decision methods assume that bounds used at leaves of the decision tree are given at the beginning and that they do not change. However when the decision tree becomes large the forward improvement step need not always produce the best choice. This is when the alternative strategy aimed to improve initial bounds is present and can lead to better improvement. I have proposed a new class of decision methods that combines advantages of both forward and backward (bound) improvement steps via a metalevel adaptive decision procedure.

8.3 POMDP approximation methods

In my work I have summarized, described and analyzed various approximation methods that can be used to solve POMDP problems. In many cases important properties of these methods or their solutions such as convergence of methods and the recursive improvement property were described and proved. Some of these are based on previously published proofs; some are new or extended to cover generic POMDP models with delays, as for example the proof of convergence of approximate value iteration with linear point interpolation rules.

An important part of my work has concerned the design of various new approximation methods or modifications of the old ones. These are mostly based on value function approximations and include:

- blind policy methods for computing lower bound value functions for generic POMDP problems;
- approximate dynamic programming and value iteration methods with a simple interpolation rule that uses randomized grids and that produces lower bound value functions for the standard POMDP model;
- approximation methods with a simple linear extrapolation rule that computes an upper bound of the value function;
- some least square error methods with forward simulation of the state.

I have performed some preliminary experiments with value functions and policies computed by the approximate dynamic programming methods with various interpolation-extrapolation rules. The test problem chosen and used for testing: "maze20", is a toy problem that belongs to the class of robot navigation problems. Results of experiments suggest that the method with the simple interpolation rule performs well despite concern about the shape of the value function, and on average always outperformed a controller based on the MDP approximation. In the experiment I have also tried other efficient local estimation rules (closest linear combination and closest neighbor). However testing for smaller sample sizes, none of these rules achieved the performance of the linear interpolation rule and the results achieved by them were even surpassed by the simple MDP-based controller. This shows that interpolation-extrapolation rules that are suitable for function approximation may not perform well when combined with approximate dynamic programming methods.

9 Future work

The objectives of my research work as described above concern exploration of various aspects associated with solving control problems and exploitation of the POMDP framework in practical real life problems. The main portion of the work has been so far directed towards design and summary of various exact and approximation methods and the description of their properties. However these methods (especially approximation methods) have not been thoroughly tested and compared with regard to the quality of control. This is one topic that I plan to focus on and explore more in my future work. The other topics of my future interest include the application of the framework in decision making systems in medicine and issues related to learning of POMDP models or policies from control scenarios.

9.1 Comparison of different methods

I have performed only limited experiments with approximation methods based on different interpolation-extrapolation rules so far. Therefore in the future I plan to do more tests and experiments with more approximation methods, for both policy and decision problems, and compare achieved results.

The need for experiments and comparison of different methods is also prompted by results acquired from my preliminary tests. The preliminary experiments showed that the combination of function approximation methods with approximate dynamic programming or value iteration methods may not behave very well and can lead to poor control performance. This was shown for example for the simple closest neighbor and minimum distance linear interpolation-extrapolation rule. I think these results are the consequence of a more general feature of error-accumulation that is caused by using functions that can concurrently overestimate and underestimate the real value function and thus significantly increase the span between value function values at two points. This in turn can cause the selection of bad control responses. This reasoning is also supported by the fact that value functions acquired via a simple interpolation rule or MDP approximations that are guaranteed to lower bound the real value function gained better results. When this is true, the application of the least square error rule for function approximation can suffer from the same deficiency.

9.2 Application of the framework

I have applied the POMDP framework only to the simple maze navigation problem, mostly for the purpose of testing. In the near future I plan to look on more serious applications of the framework in real problem domains. Currently I am considering the problem of management of the patient with ischemic heart disease (see e.g. [Wong et.al. 90] [Leong 94]) and/or management of the acute patient cases in time critical environments like the Emergency Room or Intensive Care Unit. Both of these problems require consideration of benefits and costs associated with performing observations that in turn can help to treat better the underlying disease process that is not known with certainty. In the following, the basic idea about modeling the problem of management of ischemic heart disease patients using the POMDP framework will be outlined.

9.2.1 Management of ischemic heart disease

Three basic components of the POMDP model for the ischemic heart disease (IHD) example, corresponding to states, actions and observations are shown below. The *state* of the patient (internal state) in the example is modeled using state variables that represent relevant history information (past MI, or a history of corrective procedures), the current status of coronary arteries, especially from the point of narrowing, and current level of ischemia. A special state is associated with death.

State variables

death
history of myocardial infarction (MI): T/F
history of coronary bypass surgery (CABG): T/F
history of angioplasty (PTCA): T/F
coronary artery disease:
- normal
- mild (no significant stenosis)
- moderate (1 or 2 vessels stenosis > 70%, no left main coronary artery - LMCA disease)
- severe (3 vessel stenosis, no LMCA)
- LMCA (stenosis > 50%)

ischemia (O_2 supply/demand):
- normal
- mild
- severe
- acute MI

Actions

no action
angiogram investigation
stress test
medication treatment
angioplasty (PTCA)
coronary bypass surgery (CABG)

Observations

angiogram investigation results
stress test results: positive/negative
resting EKG: positive/negative
chest pain history:
    no chest pain
    typical pain
    atypical pain
detected-MI: T/F

The set of *actions* in the model can have exploratory, transitional or cost effects. The exploratory effect of actions is based on their ability to induce observations that in turn can be suggestive of some internal states. An example is an angiogram investigation or stress test. The transition effect of the action is represented by its capability to change the internal state of the patient: e.g. PTCA can lead to the reopening of the blood supply in the main vessels. Note also that actions with intended exploratory effects can lead to changes in the patient state: e.g. increased incidence of MI due to angiogram investigation. The third effect of actions is their cost: the cost measured by the patient's suffering, discomfort and/or economic cost associated with the specific action. Actions that have only an exploratory effect and are neither associated with a cost nor affect the state transition are not explicitly represented in the action set.

Observations in the example are modeled through a set of observation variables. These are either triggered through actions, e.g., angiogram result, or corresponds to unconditional observations. Unconditional observations are assumed to be "costless" and available at any time. In our example rest EKG results or chest pain history are assumed to be unconditional due to their relatively low cost. This is unlike the angiogram result that is obtained through risky and costly investigation. In order to focus on the IHD aspect of the disease, we also assume that the acute MI is observed through the auxiliary variable MI-detected.

The *transition* and *observation models* are defined by conditional probability distributions and represent the stochastic nature of the patient's state changes on one side and uncertainty about the actual state on the other. For example the patient with coronary disease can either die, suffer from MI, or get the coronary artery repair as a result of PTCA or CABG, with different probabilities associated with every outcome. Similarly, typical chest pain can, to various degrees, point to different types of ischemia and by inference to coronary artery status.

The *cost model* describes payoffs associated with possible transitions: for example cost associated with the transition to the dead state, or cost associated with the occurrence of MI. The decision criteria that try to reduce the expected cost then in fact try to avoid these negative states.

The POMDP, like many other frameworks, models continuous time through discretization. In the IHD example it is assumed that every action is associated with a fixed duration and that any change in state occurs between the discretized time points. The way duration of the transitions is set up for the model may in many cases influence the transition probabilities. In the IHD case it is assumed that transitions associated with invasive actions occur within minutes or hours, and transitions associated with non-invasive actions within months (approx. 3 months).

9.3 Learning in partially observable stochastic control domains

In many instances the problem of providing either a control plan or an underlying model of the environment and goals can turn out to be a hard task itself. For example the assignment of costs or utilities or other parameters of the model is often hard to do consistently by human expert. Therefore the important question in this respect is if we can build a good control or decision making agent in less painful ways, e.g., directly from observed control scenarios. The problem of learning in partially observable and stochastic control domains is both hard and extremely challenging and I plan to spend more time investigating it in the future.

The basic learning scenario in the control domain is that the learner observes sequences of control actions, observations and reinforcements. Reinforcements represent either costs or rewards and quantify the goodness of the transitions that occurred with regard to the control goal. The learner can be either combined with the controller with the capability to perform actions or it may be only a passive observer. Using active learning can often lead to shorter learning times due to the fact that the controller can explore those control sequences it considers more relevant. On the other hand passive learning assumes that the learner is given information about a control case without any active intervention, which can be crucial in some domains like medicine.

In general, depending on what we want to learn, we can speak about two main learning approaches in control domains :

- learning of POMDP models
- learning of control policies

9.3.1 Learning of the model

The first approach is trying to learn the model form observed data and reinforcements. Such a model is then used to compute the optimal or approximate control in the obvious way. Learning of the model can consist of learning of the complete model (both structure and parameters) or learning of model parameters. The problem of learning of model parameters is far easier and methods for learning parameters of probabilistic networks with hidden variables, like EM [Rabiner, Juang 86] [Spiegelhalter et.al. 93][Lauritzen 94], Gibbs sampling or gradient descent methods [Russell et.al. 95], can be applied.

The agent with a built-in parameter learning mechanism can be the basis of an adaptive control agent that can adapt its behavior with regard to specificities of the control cases that have been solved. The adaptation of the model parameters can be important, e.g., when there is a natural variation in cases the control agent repeatedly solves and when one can incorporate in the model initially only population estimates at best.

The learning of a complete POMDP model is a far harder task, as one is supposed to go beyond learning of parameter values and also derive the underlying hidden structure. The learning of POMDP models has not been explored to a sufficient depth so far. Two approaches published are the predictive distinction approach [Chrisman 92] and the utile distinction approach [McCallum 93]. Both of these operate under various simplifying assumptions (restricted value function form) and gradually increase the number of states needed to fit the observed control data.

9.3.2 Learning of control policies

The second approach is based on the assumption that one can build a good controller without the detailed underlying model by building control policies directly based on action-observation sequences. This is in many respects related to the approach of feature based approximation with truncated histories [Platzman 77] [White, Scherer 94]. The control policies using truncated histories can be learned, e.g., using reinforcement learning techniques (see, e.g., [Watkins 89], [Barto et.al. 91], [Hauskrecht 94], [Kaelbling et.al. 96]). The problems with this are that the number of items in the history is not known in advance and also that not all observations and actions are equally relevant to control. An approach that attempts to dynamically identify the relevant history items to be used in the control policy definition was presented in [McCallum 95].

10 Schedule of work

The following summarizes the schedule of work to be done along the outlined research objectives.

Summer - Fall 1996:

- testing of various approximation algorithms on a moderate sized (around 20-30 states) POMDP problem/s, most probably related to robot navigation;
- application of the framework to the medical area, more specifically in the management of ischemic heart disease or in a time critical application that requires one to trade off and combine diagnostic and treatment steps (like in the ER, ICU or OR environments);
- start to work on the problem of learning in dynamical and stochastic control domains, look at ideas related to learning of POMDP models or control policies from control scenarios (data)

Spring 1997

- proceed with the topic of learning in dynamical and stochastic control domains
- evaluation of results of various approximation algorithms
- writing the thesis

Expected date of submission: May 1997 (August 1997)

References

[Albus 71] J. Albus. A theory of cerebellar functions. Mathematical biology. 10, pp. 26-61, 1971

[Astrom 65] K.J. Astrom. Optimal control of Markov decision processes with incomplete state estimation. Journal of Mathematical Analysis and Applications, 10,, pp. 174-205, 1965

[Barto et.al. 83] A.G. Barto, R.S. Sutton, C.W. Anderson. Neuronlike adaptive elements that can solve dificult learning control problems. IEEE Transactions on Systems, Man, and Cybernetics, 13, pp.835 -846, 1983

[Barto et.al. 91] A.G. Barto, S.J. Bradtke, S.P. Singh. Real-time learning and control using asynchronous dynamic programming. Umass, TR-91-57, 1991.

[Bellman 57] R. Bellman. Adaptive Control Processes. Princeton University Press. Princeton, New Jersey, 1961

[Bellman, Dreyfus 62] R. Bellman, S. Dreyfus. Applied Dynamic Programming. Princeton University Press, New Jersey, 1962

[Bertsekas 95] D.P. Bertsekas. Dynamic programming and optimal control. Prentice-Hall, 1995.

[Boutillier, Dearden 94] C. Boutilier, R. Dearden. Using abstractions for decision-theoretic planning with time constraints. AAAI-94, pp.1016-1022.

[Boutillier et.al. 95] C. Boutilier, R. Dearden, M. Goldszmidt. Exploiting structure in policy construction. IJCAI-95, 1104-1111 .

[Boutillier, Puterman 95] C. Boutilier, M. Puterman. Process oriented planning and Avarage-Reward optimality. IJCAI-95, 1096-1103.

[Buntine 94] W.L. Buntine. Operations for learning with graphical models. Journal of Artificial Intelligence Research, 2, pp.159-225, 1994.

[Cassandra et.al. 94] A.R. Cassandra, L.P. Kaebling, M.L. Littman. Acting optimally in partially observable stochastic domains. AAAI-94, pp. 1023-1028, 1994.

[Cassandra 94] A.R. Cassandra. Optimal policies for partially observable Markov decision processes. Brown University, Technical report CS-94-14, 1994.

[Cheng 88] H.-T. Cheng. Algorithms for partially observable Markov decision processes. PhD thesis, University of British Columbia, 1988.

[Chrisman 92] L. Chrisman. Reinforcement learning with Perceptual Aliasing: The perceptual distinction approach. In the Proceeding of the 10-th AAAI conference, pp.183-188, 1992.

[Dean 91] T. Dean. Decision-theoretic control of inference for time-critical applications. International journal of Intelligent Systems, vol. 6., 1991, pp. 417-441.

[Dean, Wellman 91] T.L.Dean, M.P. Wellman. Planning and control. Morgan Kaufmann, San Mateo, 1991

[Dean et.al. 93] T. Dean, L.P. Kaelbling, J. Kirman, A. Nicholson. Planning with deadlines in stochastic domains. AAAI-93, pp.574-579

[Dean 95a] T. Dean. Decision Theoretic planning and Markov decision processes. on www-document, 1995.

[Dean, Lin 95] T. Dean, S.-H. Lin. Decomposition techniques for planning in stochastic domains. In IJCAI-95, pp. 1121-1127, 1995.

[Dean, Lin 95] T. Dean, S.-H. Lin. Decomposition techniques for planning in stochastic domains. Brown univesrtity, TR CS-95-10, 1995.

[Dean 95a] T. Dean. Decision Theoretic planning and Markov decision processes. on www-document, 1995.

[Deng, Moore 95] K. Deng, A.W. Moore. Multiresolution Instance-based learning. Proceedings of the IJCAI-95,1995.

[Doyle 92] J. Doyle. Rationality and its roles in reasoning. Computational Intelligence, vol. 8., 3, 1992, pp. 376-409.

[Eagle 84] J.N. Eagle. The optimal search for a moving target when search path is constrained. Operations Research, 32:5, pp. 1107-1115, 1984.

[Hansen 94] E.A. Hansen. Cost-effective sensing during plan execution. AAAI-94, pp. 1029-1035.

[Heckerman et.al. 89] D.E. Heckerman, J.S. Breese, E.J. Horvitz. The compilation of decisions models. Knowledge Systems Laboratory, KSL-89-58, 1989.

[Hauskrecht 94] M. Hauskrecht. Reinforcement learning of control policies. Term paper, Machine learning, pp.22, 1994.

[Hauskrecht 95] M. Hauskrecht. Learning Bayesian belief networks from data. MIT EECS Area Exam paper, 36 pages, February 1995.

[Hauskrecht 96] M. Hauskrecht. Dynamic decision making in stochastic partially observable domains. AAAI Spring symposium, 1996.

[Howard 60] R.A. Howard. Dynamic Programming and Markov Processes. MIT press, Cambridge, 1960

[Howard, Matheson 84] R.A. Howard, J.E. Matheson. Influence Diagrams. In the Principles and Applications of Decision Analysis, Vol.2, 1984.

[Jaakkola el.al. 93] T. Jaakkola, M.I. Jordan, S.P. Singh. On the convergence of Stochastic Iterative Dynamic Algorithms. AI memo 1441, Massachusetts Institute of Technology, 1993

[Kanazawa et.al. 95] K. Kanazawa, D. Koller, S. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In: UAI-95, pp.346-351, 1995.

[Kaelbling et.al.95] L.P. Kaelbling, M.L. Littman and A.R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains," Unpublished.,1995

[Kaelbling et.al. 96] L.P. Kaelbling, M.P. Littman, A.W. Moore. Reinforcement learning: a survey. Journal of Artificial Intelligence Research, 4:237-285, 1996.

[Kushmerick et.al. 94] N. Kushmerick, S. Hanks, D. Weld. An algorithm for probabilistic least commitment planning. In Proceedings of AAAI-94, pp. 1073-1078, 1994.

[Lauritzen 94] S.L. Lauritzen. The EM algorithm for graphical association models with missing data. Computational Statistics and Data Analysis, ? (I have TR), 1994.

[Leong 94] T.-Y. Leong. An integrated approach to dynamic decision making under uncertainty. MIT/LCS/TR-631, 1994.

[Littman 94] M.L. Littman. The Wittness algorithn: solving partially observable Markov decision processes. Brown University, Technical report CS-94-40, 1994.

[Littman et. al. 95a] M.L. Littman, A.R. Cassandra, L.P. Kaelbling. Learning policies for partially observable environmnets: scaling up. In Proceedings of the 12-th international conference on Machine Learning, 1995

[Littman et.al. 95b] M.L. Littman, T.L. Dean, L.P. Kaelbling. On the complexity of solving Markov decision problems. In the Proceedings of UAI-95, 1995

[Littman et.al. 95c] M.L. Littman, A.R. Cassandra, L.P. Kealbling. Efficient dynamic programming updates in partially observable Markov decision processes. submitted to Operations Research, 1995.

[Lovejoy 91a] W.S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. Annals of Operations Research, 28, pp. 47-66, 1991.

[Lovejoy 91b] W.S. Lovejoy. Computationally feasible bounds for partially observed Markov decision processes. Operations Research, 39:1, pp. 192-175, 1991.

[Lovejoy 93] W.S. Lovejoy. Suboptimal policies with bounds for parameter adaptive decision processes. Operations Research, 41:3, pp. 583-599, 1993.

[McCallum 93] R.A. McCallum. Overcoming Incomplete Perceptions with Utile Distinction Memory. In the Proceedings of the 10-th Machine Learning conference, pp. 190-196, 1993.

[McCallum 95] R.A. McCallum. Instance-based utile distinctions for reinforcement learning with hidden state. In the Proceedings of the 12-th International conference on Machine Learning, 1995.

[Monahan 82] G.E. Monahan. A survey of partially observable Markov decision processes: theory, models, and algorithms. Management Science, 28:1, pp. 1-16, 1982.

[Moore, Atkenson 93] A.W. Moore, C.G. Atkenson. Prioritized Sweeping: Reinforcement Learning With Less Data and Less Time. Machine Learning,10, pp. 103-130, 1993

[Papadimitriou, Tsitsiklis 87] C.H. Papadimitriou, J.N. Tsitsiklis. The complexity of Markov decision processes. Mathematics of Operations Research, 12:3, pp. 441-450, 1987.

[Parr, Russell 95] R. Parr, S. Russell. Approximating optimal policies for partially observable Stochastic domains. IJCAI-95, xxx, 1995.

[Pearl 89] J. Pearl. Probabilistic reasoning in intelligent systems. Morgan Kaufmann, 1989.

[Platzman 77] L.K. Platzman. Finite memory estimation and control of finite probabilistic systems. MIT EECS Department, PhD. thesis, 1977.

[Puterman 94] M.L. Puterman. Markov Decision Processes: discrete stochastic dynamic programming. John Wiley and Sons, 1994.

[Rabiner, Juang 86] L.R. Rabiner, B.H. Juang. An introduction to Hidden Markov Models. In IEEE ASSP magazine 3(1), pp. 4-16, 1986.

[Rumelhart et.al 86] D.E. Rumelhart, G.E.Hinton, R.J. Williams. Learning internal representations by error propagation. In Parallel Distributed Processing, chapter 8,pp. 318-362, 1986.

[Russell, Wefald 91] S. Russell, E. Wefald. Principles of metareasoning. Artificial Intelligence, 49, 1991, pp. 361-395.

[Russell, Subramanian 95] S. Russell, D. Subramanian. Provably bounded-optimal agents. Journal of Artificial Intelligence Research 2, 1995, 575-609.

[Russell et.al. 95] S. Russell, J. Binder, D. Koller, K. Kanazawa. Local learning in probabilistic networks with hidden variables. In the Proceedings of IJCAI-95, pp. 1146-1152, 1995.

[Schachter 86] R.D. Schachter. Evaluating influence diagrams. Operations Research, 34:6, pp. 871-882, 1986.

[Singh et.al. 94] S.P. Singh, T. Jaakkola, M.I. Jordan. Learning Without State-Estimation in Partially observable Markovian Decision Processes. In the Proceedings of the 11-th conference on Machine Learning, pp. 284-292, 1994.

[Schachter, Peot 89] R.D. Schachter, M.A. Peot. Simulation approaches to general probabilistic inference on belief networks. In: Proceedings of UAI-89, 1989.

[Schachter, Peot 92] R.D. Schachter, M.A. Peot. Decision making using probabilistic inference methods. In the proceedings of the Eight conference on the uncertainty in AI 92, pp. 276-283, 1992

[Smallwood, Sondik 73] R.D. Smallwood, E.J. Sondik. The optimal control of Partially observable processes over a finite horizon. Operations Research, 21, pp. 1071-1088.

[Spiegelhalter et.al. 93] D.J. Spiegelhalter, A.P. Dawid, S.L. Lauritzen, R.G. Cowell. Bayesian Analysis in Expert Systems. Statistical Science, 8:3, pp. 219-247, 1993.

[Sondik 78] E.J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: discounted cost. Operations Research, 26, pp. xxxx, 1978.

[Sutton 88] R.S. Sutton. Learning to Predict by the Methods of Temporal Differences. Machine Learning, 3, pp. 9-44, 1988

[Sutton 90] R.S. Sutton. Integrated architecture for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. Proceedings of the Seventh International Conference on Machine Learning, Morgan Kaufmann, pp.216-224, 1990

[Tash, Russell 94] J. Tash, S. Russell. Control strategies for a stochastic planner. In AAAI-94, pp. 1079-1085, 1994.

[Watkins 89] C.J.C.H. Watkins. Learning from Delayed Rewards. PhD thesis, Cambridge University, 1989

[Webber et.al. 92] B.L. Webber, R. Rymon, J.R. Clarke. Flexible support for Trauma management through goal directed reasoning and planning. Artificial Intelligence in Medicine, 4:2, 1992.

[White, Scherer 89] C.C. White III, W.T. Scherer. Solution procedures for partially observed Markov decision processes. Operations Research, 37:5, pp. 791-797, 1989.

[White, Scherer 94] C.C. White III, W.T. Scherer. Finite memory suboptimal design for partially observed Markov decision processes. Operations Research, 42:3, pp. 439-455, 1994.

[Whitehead, Ballard 90] S.D. Whitehead, D.H, Ballard. Active Perception and Reinfocement learning. In the Proceedings of the 7-th conference on Machine Learning, pp.179-188, 1990.

[Wong et.al. 90] J.B. Wong et.al. Myocardial revascularization for chronic stable angina. Annals of Internal Medicine, 113 (1), pp. 852-871, 1990.

[Zilberstein, Russell 95] S. Zilberstein, S. Russell. Optimal composition of real-time systems. To appear in Artificial Intelligence, 1995

[Zilberstein 95a] S. Zilberstein. Models of bounded rationality, concept paper through WWW, 1995