Approximating Optimal Policies for Agents with Limited Execution Resources

Dmitri A. Dolgov and **Edmund H. Durfee**
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109
{ddolgov, durfee}@umich.edu

Abstract

An agent with limited consumable execution resources needs policies that attempt to achieve good performance while respecting these limitations. Otherwise, an agent (such as a plane) might fail catastrophically (crash) when it runs out of resources (fuel) at the wrong time (in midair). We present a new approach to constructing policies for agents with limited execution resources that builds on principles of real-time AI, as well as research in constrained Markov decision processes. Specifically, we formulate, solve, and analyze the policy optimization problem where constraints are imposed on the probability of exceeding the resource limits. We describe and empirically evaluate our solution technique to show that it is computationally reasonable, and that it generates policies that sacrifice some potential reward in order to make the kinds of precise guarantees about the probability of resource overutilization that are crucial for mission-critical applications.

1 Introduction

Optimality is the gold standard in rational decision making (e.g., [Russell and Subramanian, 1995]), and, consequently, the problem of constructing optimal policies for autonomous agents has received considerable attention over the years. Traditionally, this problem has been viewed separately from the problem of actually carrying out the policies. However, real agents have limitations as to what they can execute, and, clearly, a policy is less useful if an agent might run out of resources while carrying out the policy.

In this paper, we present a new approach for constructing policies for agents that have limited consumable resources where running out of the resources can have negative consequences. Whereas AI research has mostly focused on (PO)MDP [Boutilier *et al.*, 1999; Dean *et al.*, 1993; Howard, 1960; Puterman, 1995] methods for formulating policies for agents without emphasizing constraints on their execution resources, the Operations Research literature has developed constrained MDP (CMDP) [Altman, 1999; Puterman, 1995] techniques that can account for resource constraints. CMDP methods are particularly useful for domains where the current resource amounts are unobservable and cannot be easily estimated by the agent, or where modeling resource amounts in the state description is computationally infeasible. In an aircraft scenario, some resources and situations where such methods are beneficial could include an airplane with a broken fuel gauge (fuel amount is unobservable), pilot fatigue (attention is a resource that cannot be easily estimated), or a combination of non-critical resources (ex. various refreshments) that should nevertheless not be exhausted, but explicit modeling of which unnecessarily complicates the optimization problem and increases policy size. In the rest of the paper, we will use the fuel example, simply because it is a very intuitive instance of a consumable resource.

However, as we will explain later, standard risk-neutral CMDP optimization techniques are not applicable to problems where violating the constraints can have negative or, in the limit, catastrophic[1] consequences. The main contribution of this work is that it extends the standard CMDP techniques to handle the types of hard constraints that naturally arise in problems involving critical resources. In particular, we formulate an optimization problem where constraints are imposed on the *probability* of resource overutilization, and show how the problem can be solved using standard linear programming algorithms. Our formulation yields sub-optimal solutions to the constrained problem because it sacrifices potential reward to make guarantees about the probability of violating the resource constraints. As we later show, when violating constraints incurs dire costs, these guarantees are worthwhile ([Musliner *et al.*, 1995] and references therein).

We introduce our model in section 2, where we review Markov models, introduce notation, and specify our assumptions about the problem domain. Section 3 describes and compares several candidate solutions for addressing aspects of our problem. We then present in section 4 our new method, and empirically evaluate it in section 5. We conclude with a discussion about the strengths and limitations of our results, and about our future directions.

2 The Model

The stochastic properties of the environments in the problems that we are addressing lead us to formulate our op-

[1] The term catastrophic is, of course, relative. We assume that the system designer is willing to accept certain risks to receive payoff.

timization problem as a stationary, discrete-time, Markov decision process. In this section, we review some well-known facts from the theory of standard [Puterman, 1995; Boutilier *et al.*, 1999] and constrained [Altman, 1999] fully-observable Markov decision processes and also discuss the assumptions that are particular to the class of problems that we address in this work. This section provides the necessary background for the subsequent sections, where we discuss resource-constrained optimization problems.

2.1 Markov Decision Processes

A standard Markov decision process can be defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R} \rangle$, where $\mathcal{S}$ is a finite set of states that the agent can be in, $\mathcal{A}$ is a finite set of actions that the agent can execute, $\mathbf{P} = [p^a_{ij}] : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ defines the transition function (p^a_{ij} is the probability that the agent will go to state j if it executes action a in state i), and $\mathbf{R} = [r_{ia}] : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function (agent receives a reward of r_{ia} for executing action a in state i).

Clearly, the total probability of transitioning out of a state, given a particular action, cannot be greater than 1, i.e. $\sum_j p^a_{ij} \leq 1$. As we discuss below, we are actually interested in domains where there exist states for which $\sum_j p^a_{ij} < 1$.

A policy is defined as a procedure for selecting an action in each state. A policy that makes its choices according to a probability distribution over the set of actions is called *randomized* and can be described as a mapping of states to probability distributions over actions: $\boldsymbol{\pi} = [\pi_{ia}] : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. A *deterministic* policy that always chooses the same action for a state is, of course, a special case of a randomized policy. It can be seen (similarly to the case of standard CMDPs, as described in [Kallenberg, 1983; Puterman, 1995]) that under our optimization criterion and constraints (section 4), deterministic policies can be suboptimal. Therefore, in this work, we focus on approximating optimal policies in the class of randomized ones.

If, at time 0, the agent has an initial probability distribution $\boldsymbol{\alpha} = [\alpha_i]$ over the state space, a Markov system follows the following trajectory:

$$\rho^{t+1}_j = \sum_i \Big(\sum_a p^a_{ij} \pi_{ia} \Big) \rho^t_i, \quad \rho^0_i = \alpha_i, \tag{1}$$

where $\boldsymbol{\rho}^t = [\rho^t_i]$ is the probability distribution at time t.

2.2 Assumptions

Typically, Markov decision processes are divided into two categories: *finite-horizon* problems, where the total number of steps that the agent spends in the system is finite and is known *a priori*, and *infinite-horizon* problems, where the agent is assumed to stay in the system forever (see [Puterman, 1995] for a detailed discussion of both types of models).

In this work we concentrate on dynamic real-time domains, where agents have tasks to accomplish. For example, consider an agent flying a plane, whose goal is to safely get to its destination and land there. This example does not naturally correspond to a finite-horizon problem, because the duration of executing various policies is not predetermined (unless we artificially impose such a finite duration, which is not easily justifiable). On the other hand, the problem does not naturally fit the definition of the infinite-horizon model, because the plane obviously cannot keep on flying forever.

This leads us to make a slightly different and less common (although, certainly, not novel) assumption about how much time the agent spends executing its policy. We assume that there is no predefined number of steps that the agent spends in the system, but that optimal policies always yield *transient* Markov processes (decision problems of this type were extensively studied by Kallenberg [1983]). A policy is said to yield a transient Markov process if the agent executing that policy will eventually leave the corresponding Markov chain, after spending a finite number of time steps in it. Given a finite state space, this assumption implies that there has to be some "leakage" of probability out of the system, i.e. there have to exist some state-action pairs (i, a) for which $\sum_j p^a_{ij} < 1$. One particular case where the above assumption holds is in a system in which all trajectories lead to *absorbing* states. Once an agent enters an absorbing state, it has finished (or failed to finish) some task and has nowhere else to go, i.e. the probability of transitioning out of an absorbing state i is zero: $\sum_j \sum_a p^a_{ij} = 0$.[2] In the plane-flying example, all trajectories lead to either a safe landing or a crash, and once the agent enters one of these states, the probability of transitioning to other states is zero.

The transient nature of our problems leads us to adopt the expected total reward as the policy evaluation criterion. Given that an agent receives a reward whenever it executes an action, the total expected utility of a policy can be expressed as:

$$V(\boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{t=0}^{T} \sum_i \boldsymbol{\rho}^t_i \sum_a \pi_{ia} r_{ia}, \tag{2}$$

where T is the number of steps during which the agent accumulates utility. For a transient system with bounded rewards, the above sum converges for any T.

3 Related Work

In this section, we briefly survey several approaches (based on well-known techniques) to solving the problem of finding optimal policies for agents with limited resources, and point out the assumptions, strengths, and limitations of these methods. This establishes a landscape of solution algorithms, to which we can compare our method (presented in the next section) in terms of complexity, efficiency, and solution quality.

3.1 Fully Observable MDPs

The most straightforward way of handling resource constraints in a MDP framework is to explicitly model the resources by making their current amounts a part of the state description. This yields a standard FOMDP, which can be solved by a wide variety of efficient methods. The benefit

[2] An alternative way of handling these states is to treat them as infinitely-recurrent, i.e. once the agent gets there, it infinitely transitions back to itself. We do not adopt this model, because it is less natural for our domains and also leads to unnecessary complications in the optimization problems.

of this approach is that it allows one to make use of all relevant information to construct the best possible policy, as the agent is able to base its action choice on the current state and resource amounts. The downside of this approach is that it requires an *a priori* discretization of resource amounts to be made when the world model is constructed. Also, there is an additional burden of specifying the rewards and state transitions as functions of current resource amounts. Furthermore, in this model, the size of the state space and, consequently, the policy size, explodes exponentially in the number of resources, as compared to the state space where the resource amounts are not folded into the state description.

The FOMDP approach relies on the fact that the agent can observe the exact amounts of all resources at runtime. However, this may not hold, especially in multiagent domains with shared resources, when an agent does not know what other agents have been doing and how much of the shared resources they have been consuming.

3.2 Constrained MDPs

An alternative to the FOMDP approach described above is to treat the problem as a constrained Markov decision process [Altman, 1999], where the resources are not explicitly modeled, but rather are treated as constraints that are imposed on the space of feasible solutions (policies).

A constrained MDP for a resource-limited agent differs from a standard MDP in that the agent, besides getting rewards for executing actions, also incurs resource costs. Consequently, a constrained MDP (CMDP) can be described as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R}, \mathbf{C}, \mathbf{Q} \rangle$, where $\mathbf{C} = [c_{iak}] : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^K$ defines a vector of actions' costs (c_{iak} units of resource $k \in [1, K]$ are used when action a is executed in state i), and $\mathbf{Q} = [q_k] \in \mathbb{R}^K$ is the vector of amounts of available resources (there is q_k units of resource k initially available).

The benefit of the CMDP is that it does not require one to explicitly model how the resources affect the world. Indeed, if all policies satisfy the resource constraints, one does not have to worry about what happens when the resources are overutilized. Consequently, the state space and the resulting policies are exponentially smaller than the ones in the FOMDP model. The standard CMDP formulation constrains the expected usage of all resources to be below a certain limit and can be formulated as the following linear program:

$$\max \sum_i \sum_a x_{ia} r_{ia} \left| \begin{array}{l} \sum_a x_{ja} - \sum_i \sum_a x_{ia} p^a_{ij} = \alpha_j \\ \sum_i \sum_a x_{ia} c_{iak} \le q_k, \; x_{ia} \ge 0, \end{array} \right. \qquad (3)$$

where the variables x_{ia} have the interpretation of the expected number of times action a is executed in state i.

A weakness of this approach is that it yields policies that can be suboptimal, as compared to the ones constructed by the FOMDP method, because the agent does not base its decision on the current resource amounts, but rather completely ignores that aspect of the state. However, as mentioned in the previous section, in domains where the resources are not observable, or if the policy size is of vital importance (for example if the agent's architecture imposes constraints on policies that it can store), this approach could prove fruitful.

However, the biggest problem with this approach (as pointed out, for example, by Ross and Chen in the telecommunication domain [1988]) is that a standard CMDP imposes constraints on the *expected* amounts. Clearly, this method does not work for critical resources, whose overutilization has negative consequences. Indeed, an agent that pilots aircraft would not be satisfied with a policy that *on average* uses an amount of fuel that does not exceed the tank capacity.

3.3 Sample Path MDPs

As just mentioned, Ross and Chen pointed out the weakness of the CMDP approach with constraints on the expected amounts. As a possible solution, Ross and Varadarajan [1989; 1991] propose an approach where constraints are placed on the actual *sample-path* costs. In their work, the space of feasible solutions is constrained to the set of policies whose probability of violating the constraints (overutilizing the resources) in the long run is zero. However, their work concentrates on the average per-step costs and rewards, whereas we are interested in the total amounts, whose distributions are not easily calculable. The approach of Ross and Varadarajan has the same benefits as the standard CMDP method, i.e. no explicit modeling of resources is required, and the state space and policies are small. In addition, unlike the standard CMDP, this method is suitable for problems with critical resources. However, a weakness of this approach is that for some problems it might be too restrictive, in that it allows no possibility of overutilizing the resources. Indeed, policies produced by the sample path method might have significantly lower payoff, as compared to policies that have some probability of resource overutilization. Furthermore, for some domains it might be desirable for the system designer to be able to control the probability of resource overutilization, as a means of balancing optimality and risk.

3.4 MDPs With Constraints on Variance

Another approach to handling deviations from the expectation in Markov models is to impose additional constraints on (or to assign additional cost to) the variance. Sobel [1985] proposed to constrain the expected cost and to maximize the mean-variance ratio of the reward. Huang and Kallenberg [1994] proposed a unified approach to handling variance via an algorithm based on parametric linear programming. These approaches have the same benefits as the standard CMDP and the sample-path methods (compared to the FOMDP formulation) in terms of state space and policy size, as well as the complexity of constructing the initial world model. Additionally, they allow one to somewhat balance payoff and the deviation from the expected. However, these methods do not allow one to make hard guarantees about the probability of overutilizing the resources.

4 Linear Approximation

As hinted at in the previous sections, we would like to be able to constrain the feasible solution space to the set of policies whose probability of overutilizing the resources is below a user-specified threshold ($\mathbf{p}_0 = [p_{0k}]$). In other words, we would like to be able to solve the following math program:

$$\max \sum_i \sum_a x_{ia} r_{ia} \left| \begin{array}{l} \sum_a x_{ja} - \sum_i \sum_a x_{ia} p^a_{ij} = \alpha_j \\ P(c_k > q_k) \le p_{0k}, \quad x_{ia} \ge 0, \end{array} \right. \qquad (4)$$

where c_k is the total amount of resource k that is used by the policy, and q_k is the upper bound on that resource (as before).

The trouble is that the optimization is in the space of x_{ia}, which can be interpreted as the *expected* number of times for executing the actions in the corresponding states. However, it is difficult to express $P(c_k > q_k)$ as a simple function of the optimization variables (x_{ia}), because the latter contain no information about the probability distribution of the random variable of the number of times the action is actually executed in the corresponding state – only the expected values.[3] In this section, we present a linear approximation to the above program, based on the Markov inequality:[4]

$$P(c_k \geq q_k) \leq E[c_k]/q_k \tag{5}$$

Using this inequality and the fact that the expected resource usage can be expressed as $E[c_k] = \sum_i \sum_a x_{ia} c_{iak}$, the optimization problem can be formulated as a linear program:

$$\max \sum_i \sum_a x_{ia} r_{ia} \left| \begin{array}{l} \sum_a x_{ja} - \sum_i \sum_a x_{ia} p^a_{ij} = \alpha_j \\ \frac{1}{q_k} \sum_i \sum_a x_{ia} c_{iak} \leq p_0 \\ x_{ia} \geq 0 \end{array} \right. \tag{6}$$

A potential weakness of this approach is that the Markov inequality gives a very rough upper bound, and the policies that correspond to solutions to this LP can be too conservative, in that their real probability of overutilizing the resources can be significantly lower than p_0. However, if making hard guarantees about the probability of overutilizing the resources is of vital importance, this method might prove valuable, as it yields policies that, in general, would have higher payoff than the ones obtained by the sample-path method (as the latter is often too restrictive). On the other hand, unlike the standard CMDPs that impose constraints on the expected resource usage, and the MDPs that constrain the variance, this method allows one to *explicitly* bound the allowable probability of resource overutilization, and to make precise guarantees about the behavior of the system in that respect.

It is also worth noting that, unlike the sample-path method or the methods that constrain on variance, this method relies on solving a standard linear program, whereas the former require solving quadratic or parametric linear programs. Therefore, the above formulation appears to be a reasonable approximation, because it should be no harder to solve than the standard CMDP (see section 5 for experimental results), while providing a means of balancing solution quality with the precisely quantifiable risk of resource overutilization.

5 Evaluation

To verify our hypotheses about the properties of the approximation described in the previous section, we have performed a set of numerical experiments that compare its behavior to a standard CMDP with constraints on the expected resource amounts (section 3), and to an unconstrained MDP.

To reduce the bias that might arise from using a small number of hand-crafted test cases, we have instead used a large number of randomly-generated constrained MDPs. All of the generated problems shared some common properties, among which the most interesting ones are the following (the values for our main experiments are given in parentheses):

$N_s,\ N_a,\ N_r$ The total number of states, actions, and resources, respectively. (20, 20, 2)

$M_R = \max(R)$ Maximum reward. Rewards are assigned from a uniform distribution on $[0, M_R]$. (10)

$M_C = \max(C)$ Maximum action cost. Resource costs are assigned to state-action pairs from $[0, M_C]$ based on a distribution of R and $\rho(C, R)$ (described below). (10)

$\rho = \rho(C, R)$ Correlation between rewards and resource costs; better actions are typically more costly. ([0.8,1])

$O_k = [minQ_k, maxQ_k]$ Upper bounds on resource amounts are assigned according to a uniform distribution from this range. ([200, 300])

γ Dissipation of probability - the probability that the agent exits the system at each time step. ([0.95,0.99])

The last parameter was used to ensure a transient chain. Instead of providing a small number of sink states, we have chosen to use a uniform dissipation of probability, in order to avoid additional randomization in our experiments, as the latter choice provides a more stable domain.

Our main concern was the behavior of our approximation, as a function of the probability threshold p_0. Therefore, we have run a number of experiments for various values of $p_0 \in [0, 1]$. To be more precise, we have gradually increased p_0 from 0 to 1 in increments of 0.05, and for each value, generated 50 random models and solved them using the three methods: 1) an unconstrained MDP without any regard for resource limitations, 2) a standard CMDP with constraints on the expected usage of resources, and 3) our CMDP with constraints on the probability of overutilizing the resources (eq. 6). We then evaluated (using a Monte-Carlo simulation) each of the three solutions (policies) in terms of expected reward and probability of overutilizing the resources.

Figure 1 shows a plot of the actual probability of overutilizing the resources for the policies obtained via the three methods as a function of the probability threshold p_0. The data points are averaged over the runs for a particular value of p_0. The curve that corresponds to the method that bounds the overutilization probability also shows the standard deviation for the runs. The other data have very similar variance, so we will use the plots of means (averaged over the runs for a given p_0) for our analyses.

Obviously, p_0 has no effect on the unconstrained and the standard CMDP (which maintain more or less a constant probability of overutilization), but it does affect to a large extent the solutions to the problem with constraints on overutilization probability. One can see that the overutilization probability for the solutions produced by our approach is always below p_0 (as it should be). Also, it is worth noting that when p_0 approaches 1, our approximation yields the same results as the other methods, which is good, since setting $p_0 = 1$ should not constrain the space of feasible solutions.

[3] Generally speaking, the total cost is a sum of a random number of differently-distributed random variables, and calculating its probability distribution is a nontrivial task.

[4] This inequality only holds for nonnegative random variables. However, for a transient system, we can make the assumption that all costs are nonnegative, without any loss of generality.

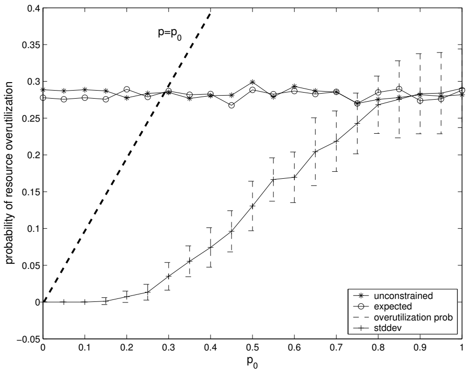

Figure 1: Probability of resource overutilization.

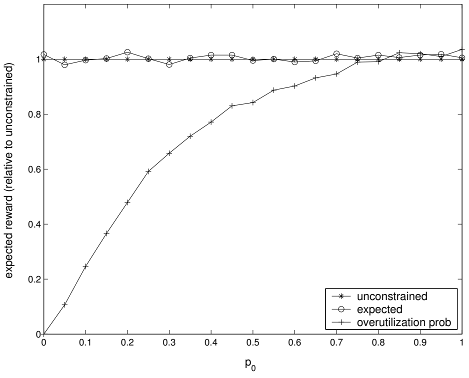

Figure 2: Average rewards for solutions to the three problems.

The rewards obtained by these policies are shown in Figure 2. These *actual* rewards do not necessarily equal the *expected* rewards (which are used during the optimization process). This is because only the runs that did not overutilize the resources were included in the average, and policies were not penalized for violating the resource constraints. This also explains why the total rewards received by solutions to the standard CMDPs were sometimes greater than the ones obtained by solutions to the unconstrained problems.

However, realistically, an agent always incurs a penalty for overutilizing a critical resource, where the penalty amount is based on the consequences of overutilization. For example, if the agent is flying a plane, and the resource in question is fuel, the consequences of trying to use too much of that resource are catastrophic, so the penalty is very high. If we take this into account by assigning a fixed penalty to policies that overutilize the resources, we can update the graph in Figure 2 to get a more realistic picture. Figure 3 shows the average rewards for solutions obtained via the three methods, recalculated to reflect the penalties: new rewards are $R' = (1 - p_{ou})R + p_{ou}W$, where p_{ou} is the overutilization probability, R is the average reward for successful runs of the given policy (as in Figure 2), and W is the penalty for overutilization. We see that, if we take the penalty into account, there is an interval of p_0 where the conservative policy produced by our linear approximation outperforms the other policies.

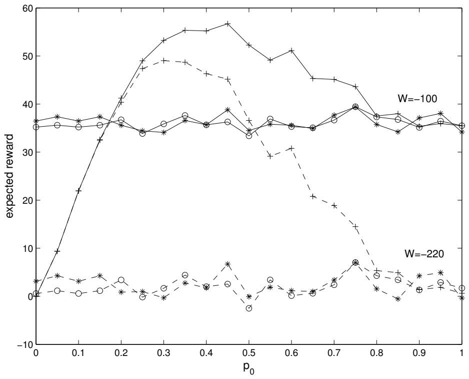

Figure 3: Average rewards with penalties for overutilization.

Moreover, for large penalties ($W = -220$), the conservative policy outperforms the others for all values of p_0. Note that here, the policy is re-evaluated *post-factum*. Section 6 briefly discusses an approach that explicitly models the penalties in the optimization program.

As we mentioned in section 4, the linear approximation should be no harder to solve than the standard constrained MDP, because both are formulated as linear programs with the same number of constraints. To experimentally verify this, we have timed the runs of all our experiments. Figure 4 contains a plot of the times that it took to solve the problems in all our experiments. One can see that the running times for all three methods are not appreciably different.[5] In particular, the average ratio of the running time for the standard CMDP to the running time of the unconstrained method is 1.25; the ratio of the running time of our linear approximation with constraints on overutilization probability to the running time of the unconstrained method is 1.06. The slight downward curvature of the plot of the running time of our approximation method appears to be a consequence of the specific implementation of the linear programming algorithm that we used in our experiments.

6 Discussion and Future Work

Our experiments substantiate the claims that our approximation provides an effective and efficient method that agents can use to formulate policies that not only consider limitations on execution resources, but that also explicitly bound the probability of resource overutilization. Our new approximation achieves the constraints on overutilization, and is essentially no more expensive to use than more standard CMDP and unconstrained MDP methods. Because our method constructs policies that are more careful about avoiding resource overutilization, the rewards associated with its policies when resources are not overutilized tend to be less than the rewards for the other methods' policies when resources are not overutilized. However, as we illustrated in Figure 3, when overutilization incurs penalties our new method can outper-

[5] Here we have to note that we used the linear programming approach to solving the unconstrained problem. A different algorithm such as value or policy iteration might be more efficient.

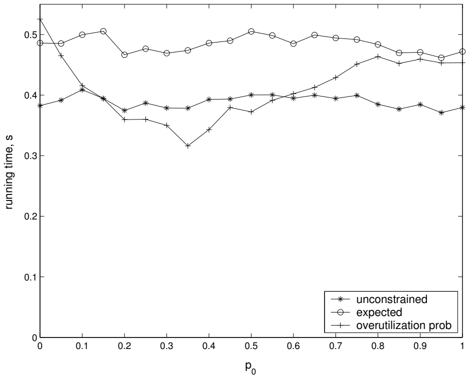

Figure 4: Running time for the three methods

form previous techniques. Thus, our new method is particularly suited to agents engaged in mission-critical domains.

Furthermore, if penalties for resource overutilization are known at design-time and can be expressed in the same units as the rewards, an interesting modification of our LP formulation is to include the penalties in the policy evaluation criterion, as opposed to modeling the constraints on the overutilization probability. This would yield an LP with the following objective function:

$$\max\Big(\sum_i\sum_a x_{ia}r_{ia} - \sum_k \frac{W_k}{q_k}\sum_i\sum_a x_{ia}c_{iak}\Big), \qquad (7)$$

where W_k is the penalty (in units of r_{ia}) incurred for overutilizing resource k. The maximization is subject to just the standard "conservation of probability" constraints as in (eq. 3). A benefit of this formulation is that, for certain initial probability distributions, deterministic policies are optimal. A downside is that the formulation does not allow one to explicitly control the acceptable overutilization probabilities.

As can be seen in Figure 1, our linear approximation is in general overly conservative. For example, given permission to overutilize the resource 20% of the time ($p_0 = 0.2$), the method generates a policy that overutilizes the resource only about 1% of the time. Since rewards and resource usage are typically correlated, we would expect that a policy that undershoots the permitted resource overutilization probability by a lower amount would also yield a higher expected reward. Toward this end, we have formulated a quadratic programming approximation, based on the Chebyshev inequality, which allows us to put a better upper bound on the probability of resource overutilization:

$$\sum_i\sum_a x_{ia}c_{iak} \leq q_k - \Delta_k, \quad \sigma_k^2/\Delta_k^2 \leq p_{0k}, \qquad (8)$$

where $\sigma_k^2 = \text{VAR}[c_k] = E[c_k^2] - E[c_k]^2$ is the variance in the amount of used resource c_k, and Δ_k specify the widths of the allowable regions for the amounts of used resources.

We are also investigating another formulation of the optimization program that should give a more accurate estimate of the resource overutilization probability. The method is based on a polynomial approximation of the pdf of the total resource-usage cost and uses the moments of the cost as the optimization variables. This approximation and the one based on the Chebyshev inequality are more costly to compute than the linear one. Our current efforts are to encode the approximations and evaluate their strengths and weaknesses.

7 Acknowledgments

This paper is based upon work supported by DARPA/ITO and the Air Force Research Laboratory under contract F30602-00-C-0017 as a subcontractor through Honeywell Laboratories. The authors thank Kang Shin, Haksun Li, and David Musliner for their valuable contributions, as well as one of the anonymous reviewers whose comments inspired (eq. 7).

References

[Altman, 1999] Eitan Altman. *Constrained Markov Decision Processes*. Chapman and HALL/CRC, 1999.

[Boutilier *et al.*, 1999] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[Dean *et al.*, 1993] Thomas Dean, Leslie Pack Kaelbling, Jak Kirman, and Ann Nicholson. Planning with deadlines in stochastic domains. In *Proc. of the Eleventh National Conf. on Artificial Intelligence*, pages 574–579, CA, 1993.

[Howard, 1960] R. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, 1960.

[Huang and Kallenberg, 1994] Y. Huang and L.C.M. Kallenberg. On finding optimal policies for Markov decision chains. *Math. of Operations Research*, 19:434–448, 1994.

[Kallenberg, 1983] L.C.M. Kallenberg. *Linear Programming and Finite Markovian Control Problems*. Mathematisch Centrum, Amsterdam, 1983.

[Musliner *et al.*, 1995] David John Musliner, James Hendler, Ashok K. Agrawala, Edmund H. Durfee, Jay K. Strosnider, and C. J. Paul. The challenges of real-time AI. *IEEE Computer*, 28(1), 1995.

[Puterman, 1995] M. L. Puterman. Markov decision processes. New York, 1995. John Wiley & Sons.

[Ross and Chen, 1988] K.W. Ross and B. Chen. Optimal scheduling of interactive and non-interactive traffic in telecommunication systems. *IEEE Transactions on Auto Control*, 33:261–267, 1988.

[Ross and Varadarajan, 1989] K.W. Ross and R. Varadarajan. Markov decision processes with sample path constraints: the communicating case. *OR*, 37:780–790, 1989.

[Ross and Varadarajan, 1991] K.W. Ross and R. Varadarajan. Multichain Markov decision processes with a sample path constraint: A decomposition approach. *Math. of Operations Research*, 16:195–207, 1991.

[Russell and Subramanian, 1995] S. J. Russell and D. Subramanian. Provably bounded optimal agents. *JAIR*, 2:575–609, 1995.

[Sobel, 1985] M.J. Sobel. Maximal mean/standard deviation ratio in undiscounted mdp. *OR Letters*, 4:157–188, 1985.