

# Solving POMDPs Using Quadratically Constrained Linear Programs

Christopher Amato  
camato@cs.umass.edu

Daniel S. Bernstein  
bern@cs.umass.edu

Shlomo Zilberstein  
shlomo@cs.umass.edu

Department of Computer Science  
University of Massachusetts  
Amherst, MA 01003

## Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search; G.1.6 [Numerical Analysis]: Optimization—*Constrained Optimization*

## General Terms

Algorithms

## Keywords

POMDPs, Planning under uncertainty, Optimization

## 1. INTRODUCTION

Since the early 1990's, Markov decision processes (MDPs) and their partially observable counterparts (POMDPs) have been widely used by the AI community for planning under uncertainty. POMDPs offer a rich language to describe situations involving uncertainty about the domain, stochastic actions, noisy observations, and a variety of possible objective functions. Even though an optimal solution may be concise, current exact algorithms that use dynamic programming often require an intractable amount of space. POMDP approximation algorithms can operate with a limited amount of memory, but as a consequence they provide very weak theoretical guarantees. In contrast, we describe a new approach that addresses the space requirement of POMDP algorithms while maintaining well-defined optimality guarantees.

Current methods to find optimal fixed-size controllers rely solely on local information [5, 6], thus can give solutions that are unboundedly below the global optimum. We present a formulation that defines the optimal fixed-size controller, and we employ standard nonlinearly constrained optimization techniques in order to find POMDP policies using this formulation. The quadratic constraints and linear objective function of our formulation belong to a special class of optimization problems for which many robust and efficient algorithms have been developed. While these techniques only guarantee locally optimal solutions, the new formulation facilitates a more efficient search of the solution space and pro-

duces high-quality results. Moreover, the algorithms employ more advanced techniques than those previously used [5, 6] to avoid convergence at certain suboptimal points. Using these locally optimal techniques, we demonstrate improved controller quality for a set of large POMDPs while maintaining similar running times when compared with a current state-of-the-art approach. This suggests that by using our QCLP, small, high-valued controllers can be efficiently found for a large assortment of POMDPs.

## 2. BACKGROUND

A POMDP can be defined as a tuple  $\langle S, A, P, R, \Omega, O \rangle$ , with  $S$ , a finite set of states with designated initial state distribution  $b_0$ ,  $A$ , a finite set of actions,  $P(s'|s, a)$ , the state transition model,  $R(s, a)$ , the reward model,  $\Omega$ , a finite set of observations, and  $O(o|s', a)$ , the observation model. In this paper, we consider the case in which the process unfolds over an infinite sequence of stages. At each stage the agent selects an action, which yields an immediate reward, and receives an observation. The agent must choose an action based on the history of observations seen. Because the state is not directly observed, it may be beneficial for the agent to remember the observation history. The agent's goal is to maximize the expected discounted sum of rewards received. To maintain finite sums, a discount rate,  $0 \leq \gamma < 1$  is used.

Finite-state controllers are an elegant way of representing POMDP policies using a finite amount of memory. The controller can formally be defined by the tuple  $\langle Q, \psi, \eta \rangle$ , where  $Q$  is the finite set of controller nodes,  $\psi : Q \rightarrow \Delta A$  is the action selection model for each node, and  $\eta : Q \times A \times O \rightarrow \Delta Q$  represents the node transition model for each node, given an action was taken and an observation seen. We allow for stochastic transitions and action selection, as this can help to make up for limited memory [8]. The value of a node  $q$  at state  $s$ , given action selection and node transition probabilities, is provided by:  $V(q, s) = \sum_a P(a|q) [R(s, a) + \gamma \sum_{s'} P(s'|s, a) \sum_o O(o|s', a) \sum_{q'} P(q'|q, a, o) V(q', s')]$  This equation is referred to as the Bellman equation.

Two previous POMDP solution techniques have sought to optimize fixed-size controllers. Poupart and Boutilier [6] have developed a method called bounded policy iteration (BPI). This approach alternates between policy improvement and evaluation, iterating through the nodes in the controller and using a linear program to examine the value of probabilistically taking an action and then transitioning into the old controller. If an improvement can be found, the action selection and node transition probabilities are updated

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.  
Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

For variables:  $x(q', a, q, o)$  and  $y(q, s)$

Maximize  $\sum_s b_0(s)y(q_0, s)$

Given the Bellman constraints:

$$\forall q, s \ y(q, s) = \sum_a \left[ \left( \sum_{q'} x(q', a, q, o) \right) R(s, a) + \gamma \sum_{s'} P(s'|s, a) \sum_o O(o|s', a) \sum_{q'} x(q', a, q, o) y(q', s') \right]$$

And probability constraints:

$$\forall q, o \ \sum_{q', a} x(q', a, q, o) = 1, \forall q, o, a \ \sum_{q'} x(q', a, q, o) = \sum_{q'} x(q', a, q, o_k) \text{ and } \forall q', a, q, o \ x(q', a, q, o) \geq 0$$

**Table 1: The QCLP for finding the optimal fixed-size controller. Variable  $x(q', a, q, o)$  represents  $P(q', a|q, o)$ ,  $y(q, s)$  represents  $V(q, s)$ ,  $q_0$  is the initial controller node and  $o_k$  is an arbitrary fixed observation.**

accordingly. This approach guarantees to at least maintain the value of a provided controller, but it is not likely to find a concise optimal controller. Also, while heuristics can be used to incorporate start state knowledge [7], they may be difficult to adjust and are not applicable in all domains. Meuleau et al. [5] use gradient ascent based on a cross-product MDP created from the controller and the POMDP. Due to the complex and incomplete gradient calculation, this method can be time consuming and error prone.

### 3. OPTIMAL FIXED-SIZE CONTROLLERS

The linear program used by BPI may allow for controller improvement, but can easily get stuck in local maxima. While Poupart and Boutilier suggest heuristics for becoming unstuck, a nonlinear approach can allow these local solutions to be avoided. Also, since BPI requires that value is increased for all states, possible improvements are not always made. We believe that many POMDPs have small optimal controllers or can be approximated concisely. Optimization techniques permit these controllers to be found while maintaining moderate resource usage.

Unlike BPI and Meuleau et al.'s gradient ascent, our formulation defines the optimal controller for a given size. This is done by considering the values of each node and state pair variables. To ensure that these values are correct given the action selection and node transition probabilities, quadratic constraints must be added. This results in a quadratically constrained linear program (QCLP). QCLPs may contain quadratic terms in the constraints, but have a linear objective function. They are a subclass of general nonlinear programs that has structure which algorithms can exploit. Although it is often difficult to solve a QCLP exactly, many robust and efficient algorithms can be applied. Our QCLP has a simple gradient calculation and an intuitive representation that matches well with common optimization models.

Contrary to BPI, which alternates between policy improvement and evaluation, our QCLP improves and evaluates the controller in one phase. As seen in Table 1, the value of an initial node is maximized at an initial state distribution using parameters for the action selection and node transition probabilities at each node,  $P(q', a|q, o)$ , and the values of each node in each state,  $V(q, s)$ . To ensure that the values are correct given the action and node transition probabilities, nonlinear constraints representing the corresponding Bellman equations must be added to the optimization. The second and last constraints in the table ensure that the variables represent proper probabilities, and the third constraint guarantees that action selection does not depend on the observation that has not yet been seen.

**THEOREM 1.** *An optimal solution of the QCLP results in an optimal stochastic controller for the given size and initial state distribution.*

**PROOF.** The optimality of the controller follows from the Bellman equation constraints and maximization of a given node at the initial state distribution. The Bellman equation constraints restrict the value variables to valid amounts based on the chosen probabilities, while the maximum value is found for the initial node and state. Hence, this produces an optimal controller.  $\square$

## 4. EXPERIMENTS

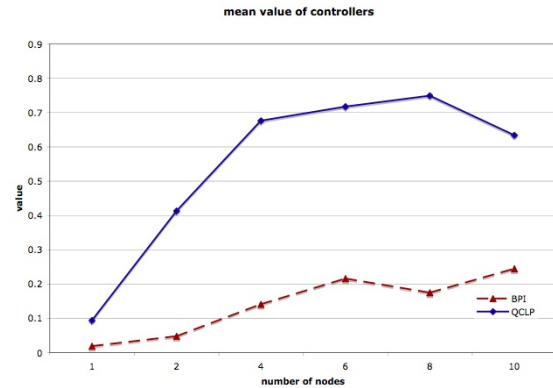
For this paper, we used a freely available nonlinearly constrained optimization solver *snopt* [3] on the NEOS server. BPI was implemented in a manner very similar to the original techniques in [6]. Meuleau et al.'s gradient ascent was also implemented, but produced significantly worse results in substantially more time than the other techniques.

In our experiments, we used BPI and *snopt* to optimize a range of fixed-size controllers. Each method was initialized with the same ten random deterministic controllers and we report mean values and times. While the experiments were conducted on different computers, we expect solution times to only differ by a small constant factor in our experiments.

### 4.1 Hallway benchmark

The hallway domain introduced by Littman, Cassandra and Kaelbling [4], is a frequently used benchmark for POMDP algorithms. A robot must navigate a grid world with 57 states, 21 observations and 5 actions.

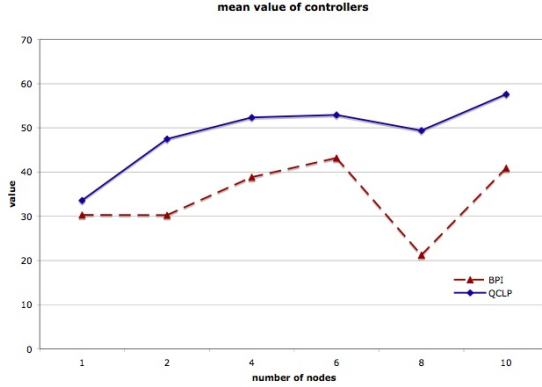
The results for this domain are shown in Table 2 and Figure 1. We see that for all controller sizes the mean value produced by the QCLP is more than twice that of BPI. Also, the time taken to produce these controllers remains similar, but the difference increases as controller size grows. It can be noted that much higher values can be produced by the



**Figure 1: Hallway domain: the mean values using BPI and the QCLP for increasing controller size**

| # nodes | $QCLP_{hall}$ | $BPI_{hall}$ | $QCLP_{mach}$ | $BPI_{mach}$ |
|---------|---------------|--------------|---------------|--------------|
| 1       | < 1 min       | < 1 min      | < 1 min       | 1.3 mins     |
| 2       | < 1 min       | < 1 min      | < 1 min       | 4.6 mins     |
| 4       | < 1 min       | < 1 min      | 7.9 mins      | 14.1 mins    |
| 6       | 1.4 mins      | 1.6 mins     | 42.4 mins     | 25.5 mins    |
| 8       | 6.9 mins      | 2.9 mins     | 57.5 mins     | 42.9 mins    |
| 10      | 9.1 mins      | 4 mins       | 130.8 mins    | 62.8 mins    |

**Table 2: Mean running times for the QCLP and BPI on the hallway and machine problems**



**Figure 2: Machine domain: the mean values using BPI and the QCLP for increasing controller size**

QCLP with markedly smaller controllers, thus reducing the time required.

Poupart and Boutilier also report results for BPI on a 1500 node controller. In their implementation, an escape technique that attempts to avoid local maxima was also used. After 69 hours, BPI produced a controller with a value of 0.51. Our QCLP formulation generates a controller with 65% higher value in 0.22% of the time with 0.67% of the representation size.

## 4.2 Machine maintenance

In order to test performance on a larger problem, we consider a machine maintenance domain with 256 states, 4 actions and 16 observations. For more details, see [2].

For this problem, the optimal policy for the underlying MDP is known to be 66.236. We see in Table 2 and Figure 2 that very small controllers produce solutions which are close to this upper bound. The techniques using the QCLP show higher mean values for all controller sizes than those produced by BPI. While the differences are less dramatic than the previous example, they are no less consistent. And although we again notice a disparity in running time as the number of nodes increases, we also notice that the mean value of the 10 node controller produced by BPI has lesser value and longer running time than the 2 node controller for the QCLP. This suggests that smaller controllers could replace those produced by BPI.

## 5. CONCLUSIONS

We introduced a new approach for solving POMDPs using a nonlinear programming formulation of the problem, which defines an optimal fixed-size stochastic controller. Near-optimal controllers can be found using standard nonlinear optimization techniques. This provides a promising new way

for solving POMDPs and approximating optimal solutions with concise controllers. We showed that by using nonlinear optimization algorithms, we can produce higher valued controllers than those found by BPI, the existing state-of-the-art approach. We demonstrated empirically a consistent improvement over a range of controller sizes. We also showed that higher quality solutions can be found with significantly smaller controllers and similar running times. These results suggest that optimal or near-optimal controllers could be found for large POMDPs, making our approach very useful in a range of practical application domains.

In the future, we plan to examine the relationship between the nonlinear formulation and the performance of various optimization algorithms. Different representations may better match current nonlinearly constrained optimization methods and may thus produce better solutions. We also plan to study the applicability of the QCLP approach to problems involving multiple agents modeled as decentralized POMDPs. The BPI technique has already been generalized successfully to the multi-agent case [1]. In future work, we expect to be able to demonstrate similar performance gains in optimizing fixed-size controllers for a set of agents in large multi-agent domains.

## 6. ACKNOWLEDGMENTS

Support for this work was provided in part by the National Science Foundation under Grant No. IIS-0535061 and by the Air Force Office of Scientific Research under Agreement No. FA9550-05-1-0254.

## 7. REFERENCES

- [1] D. S. Bernstein, E. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005.
- [2] A. R. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Brown University, Providence, RI, 1998.
- [3] P. E. Gill, W. Murray, and M. Saunders. Snopt: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47:99–131, 2005.
- [4] M. L. Littman and A. R. C. L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. Technical Report CS-95-11, Brown University, Department of Computer Science, Providence, RI, 1995.
- [5] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cassandra. Solving POMDPs by searching the space of finite policies. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 417–426, Stockholm, Sweden, 1999.
- [6] P. Poupart and C. Boutilier. Bounded finite state controllers. In *Advances in Neural Information Processing Systems*, 16, Vancouver, BC, 2003.
- [7] P. Poupart and C. Boutilier. VDCBPI: an approximate scalable algorithm for large scale POMDPs. In *Advances in Neural Information Processing Systems*, 17, pages 1081–1088, Vancouver, BC, 2004.
- [8] S. Singh, T. Jaakkola, and M. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 284–292, 1994.