Planning in Cost-Paired Markov Decision Process Games

H. Brendan McMahan, Geoffrey J. Gordon

October 24, 2003

Abstract

We describe applications and theoretical results for a new class of two-player planning games. In these games, each player plans in a separate Markov Decision Process (MDP), but the costs associated with a policy in one of the MDPs depend on the policy selected by the other player. These cost-paired MDPs represent an interesting and computationally tractable subset of adversarial planning problems. To solve them, we extend the Double Oracle Algorithm of [3].

1 Introduction

Consider a mobile sensor platform that must decide on an observation strategy (represented as a policy in an MDP). Its rewards depend on its own policy as well as on the motion of the entity which it is trying to observe. Suppose that the output from the sensor cannot be processed in real time due to latency, insufficient on-board computation, or the need for human expert analysis; suppose also that the entity being observed is aware that it may be observed, but cannot detect when observations happen.

One natural instance of this problem is scientific data collection from a satellite or planetary rover. We want to maximize the amount of time which the sensor spends observing a particular natural phenomenon. Communication delays prevent the sensor from altering its actions based on the collected data. Nature is oblivious to the sensor's actions, but we treat her as an adversary in order to compute a robust plan.

Our model captures an interesting subset of adversarial planning problems, but it is nonetheless computationally tractable: we can solve problems in this model in polynomial time using linear programming. The linear programming solution, however, can still be impractical for large models. So, we focus on an alternative solution method, the Double Oracle Algorithm, which can take advantage of existing fast planners for MDPs by using them as best-response oracles.

The Double Oracle Algorithm, first introduced in [3], is extended here in several important ways. First, we show that the algorithm applies to the more general problem outlined above. Second, we show that by using approximate oracles in place of exact oracles we can construct an Approximate Double Oracle Algorithm that converges to an approximate minimax equilibrium for the game. This expands the range of planning techniques that can be applied. Finally, we justify the use of such algorithms by showing that the game at hand has an exact minimax solution with small support. A small-support solution is a mixture of a small number of deterministic plans, and therefore we can hope to find such a solution with only a small number of calls to our oracles. In particular, even though the MDP has exponentially many possible deterministic policies, we can get an exact minimax solution by randomizing among only a linear-size subset; we will need even fewer to get an approximate minimax solution.

1.1 Problem Model

Let $X = (\mathcal{S}^{\mathrm{X}}, \mathcal{A}^{\mathrm{X}}, \mathcal{P}^{\mathrm{X}}, \mu^{\mathrm{X}})$ and $Y = (\mathcal{S}^{\mathrm{Y}}, \mathcal{A}^{\mathrm{Y}}, \mathcal{P}^{\mathrm{Y}}, \mu^{\mathrm{Y}})$ be MDPs. For each MDP, $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P} : (\mathcal{S} \times \mathcal{A}) \to \Delta(\mathcal{S})$ is a transition function, and μ is a distribution over start states. ($\Delta(\mathcal{S})$ is the set of probability distributions over $\mathcal{S}$.) Each MDP would normally have a vector of state-action costs, but we leave the costs unspecified until later in this subsection. (Costs in X will depend on the policy in Y, and vice versa.) Let $m = |\mathcal{S}^{\mathrm{X}}| \cdot |\mathcal{A}^{\mathrm{X}}|$ and $n = |\mathcal{S}^{\mathrm{Y}}| \cdot |\mathcal{A}^{\mathrm{Y}}|$. Let Π^{X}_D (Π^{X}_{ND}) be the set of deterministic (stochastic) policies for X, and define Π^{Y}_D and Π^{Y}_{ND} analogously for Y. We rule out policies with infinite visitation frequencies; we can do so either by introducing a discount factor (in

which case all discounted frequencies will be finite); or by assuming positive edge costs for X, negative costs for Y, and no "orphan" states (in which case the agents will never choose nonterminating policies).

A policy (stochastic or deterministic) can be represented as a function from states to $\Delta(\mathcal{A})$, or equivalently as a vector of state-action visitation frequencies. To simplify notation we will write x or y for a policy, and view it as either a function or a vector of frequencies when needed. For example, given a cost vector c (length m) on state-action pairs in X, we can write $x \cdot c$ for the dot product of the costs with the visitation frequencies, giving the value of policy x under cost vector c in MDP X.

Every stochastic policy, when represented as visitation frequencies, can be decomposed as a convex combination of deterministic policies; and, every convex combination of deterministic policies corresponds to some stochastic policy. In particular, for any $\bar{x} \in \Pi^{\mathrm{X}}_{ND}$ we can write $\bar{x} = \sum_{i=1}^{k} p_i x_i$ for some set $\{x_1, \ldots, x_k\} \subseteq \Pi^{\mathrm{X}}_{D}$ and probability distribution p. A detailed proof can be found in [4, Sec. 6.9].

The cost vector for X will be a linear function of Y's policy, and vice versa. Because we are interested in zero-sum games, Y's cost will be the negative of X's. In particular, write

$$V(x, y) = x \cdot c^{\mathrm{X}} + x \cdot Gy + y \cdot c^{\mathrm{Y}}.$$

for the cost to X. Here c^{X} and c^{Y} are fixed cost vectors for X and Y, while the matrix G governs the interaction between the two players. Player one tries to minimize V, and player two tries to maximize it. So, our goal is to find

$$\min_{x \in \Pi^{\mathrm{X}}_{ND}} \max_{y \in \Pi^{\mathrm{Y}}_{ND}} V(x, y) \tag{1}$$

1.2 Equivalent problems

Equation (1) can be represented either as a linear program or as a matrix game in which each deterministic policy for X and Y becomes a pure strategy. The LP representation demonstrates that we can solve (1) in polynomial time, while the matrix representation will be useful in describing our Double Oracle Algorithm.

To show that Equation (1) is equivalent to a matrix game, we define a payoff matrix M which has one row corresponding to each deterministic policy for X and one column for each deterministic policy for Y. The entry of M corresponding to policies x and y is $M_{x,y} = V(x, y)$.

Theorem 1 *There is a one-to-one mapping between minimax equilibria in M and solutions to (1).*

The above theorem (proved in the appendix) implies that solving the game M is equivalent to solving the optimization problem (1).

Because a cost-paired MDP is a multiagent planning problem, and because it is equivalent to an exponentially large matrix game, one might suspect that it is computationally intractable. However, this is not the case; we conclude this section by noting that cost-paired MDPs can be solved in polynomial time via linear programming. The technique is an extension of an idea described in [2]. Our experience in [3] suggests, however, that translating a cost-paired MDP to an LP will not be a practical solution algorithm. So, in the next section we will describe our Double Oracle Algorithm.

2 Algorithms and Analysis

We now show how to solve the game M using the Double Oracle algorithm first introduced in [3]. Pseudocode for the algorithm is given in Listing 1. The algorithm relies on a best response oracle $\mathcal{R}$ which provides a best response pure strategy for the row player against a mixed strategy $\bar{y}$ of the column player, and an analogous oracle $\mathcal{C}$ for the column player. The subroutine solve_game finds minimax solutions to a matrix game M.

Convergence and correctness of the double oracle algorithm were first proved in [3]. We extend that result to the case where $\mathcal{R}$ and $\mathcal{C}$ are only approximate best response oracles, and show that the corresponding Approximate Double Oracle Algorithm will converge to a good approximation of the game.

Theorem 2 *If $\mathcal{R}$ returns a pure policy x such that $V(x, \bar{y}) \leq V(x^*, \bar{y}) + \epsilon$ for all $x^* \in \Pi^X_d$ and similarly $\mathcal{C}$ returns a pure policy y such that $V(\bar{x}, y) \geq V(\bar{x}, y^*) - \epsilon$ for all $y^* \in \Pi^Y_d$, then the Double Oracle Algorithm converges to a pair of mixed strategies that form a 2ϵ-approximate minimax equilibrium.*

Theorem 3 *The matrix game M has an exact minimax solution (p, q), where p has positive probability on at most $m + 1$ pure strategies and q has positive probability on at most $n + 1$ pure strategies.*

$\bar{R}^0 \leftarrow \{x_0\}$ // any pure policy for X
$\bar{C}^0 \leftarrow \{y_0\}$ // any pure policy for Y
$\bar{M}_{0,0} \leftarrow V(x_0, y_0)$ // $\bar{M}$ is a 1x1 matrix
$t \leftarrow 0$
while $((t = 0)$ OR $(\bar{R}^t \neq \bar{R}^{t-1})$ OR $(\bar{C}^t \neq \bar{C}^{t-1}))$
  $t \leftarrow t + 1$
  $(p, q) \leftarrow$ solve_game$(\bar{M})$ // p, q are minimax strategies for $\bar{M}$
  $\bar{x} \leftarrow \sum_{i=1}^{|\bar{R}|} p_i x_i$ $\bar{y} \leftarrow \sum_{j=1}^{|\bar{C}|} q_j y_j$
  $x \leftarrow \mathcal{R}(\bar{y})$ $y \leftarrow \mathcal{C}(\bar{x})$
  $\bar{R}^{t+1} \leftarrow \bar{R}^t \cup \{x\}$ $\bar{C}^{t+1} \leftarrow \bar{C}^t \cup \{y\}$
  $\bar{M}_{i,j} \leftarrow V(x_i, y_j)$ for all new pairs i, j // $\bar{M}$ is a $|\bar{R}^{t+1}| \times |\bar{C}^{t+1}|$ matrix
end
return $(\bar{x}, \bar{y})$

Figure 1: Double-Oracle Algorithm

3 Discussion

We have introduced cost-paired MDPs as a tractable model for solving a useful class of multiagent planning problems. In the future we hope to conduct experiments which demonstrate the practical application of cost-paired MDPs and which confirm the efficiency of our Double Oracle Algorithm. We also believe that we can strengthen our theoretical claims; for example, it is possible to terminate the algorithm before convergence and still provide performance guarantees.

The experiments in [3] correspond to a special case of cost-paired MDPs, in which the Y player has a single state and n actions. So, these experiments provide initial evidence that our Double Oracle Algorithm is faster than the direct linear programming approach. And, they demonstrate the practical application of cost-paired MDPs to a surveillance problem.

Acknowledgements

The authors wish to thank Jeff Schneider and Chuck Rosenberg for useful discussions. This work was supported in part by AFRL contract F30602–01–C–0219, DARPA's MICA program. The opinions and conclusions are the authors' and do not reflect those of the US government or its agencies.

References

[1] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows.* John Wiley & sons, 1990.

[2] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pages 750–759, 1994.

[3] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

[4] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Interscience, 1994.

A Proofs

A.1 Proof of Theorem 1

A mixed strategy for X is a distribution over rows of M. Suppose p defines such a distribution, with $\sum_{i=1}^{\ell} p_i = 1$. Then $\bar{x} = \sum_{i=1}^{\ell} p_i x_i$ are the visitation frequencies of the corresponding mixed policy in X's MDP, where each x_i is a pure policy. The cost of x_i against a pure strategy y_j in our cost-paired MDP is $V(x_i, y_j)$, which is the same as the cost in M, namely $x_i \cdot M y_j$.

The cost in M of p against y_j is $\sum_{i=1}^{\ell} p_i M_{ij} = \sum_{i=1}^{\ell} p_i V(x_i, y_j)$. But, since $V(\cdot, y)$ is linear, that cost is just $V(\bar{x}, y_j)$. So, the cost of a mixed strategy for X in M is the same as the cost of the corresponding mixed policy in X's MDP; a similar argument holds for Y's costs. That means that deviations from a mixed strategy in M are penalized exactly the same way as deviations in our cost-paired MDP.

A.2 Proof of Theorem 2

Suppose that after completing some iteration t the Approximate Double Oracle Algorithm returns. Let $\bar{x}$ and $\bar{y}$ be stochastic policies corresponding to the minimax equilibrium for the $|\bar{R}^t| \times |\bar{C}^t|$ matrix game $\bar{M}$ solved on iteration t. Since the algorithm returned, it must be that $x = \mathcal{R}(\bar{y})$ was already in $\bar{R}^t$, and $y = \mathcal{C}(\bar{x})$ was already in $\bar{C}^t$.

Let $V^t = V(\bar{x}, \bar{y})$. Suppose x^* and y^* are the actual best responses, $x^* = \text{argmin}_{x \in \Pi^{\text{X}}_D} V(x, \bar{y})$ and $y^* = \text{argmax}_{y \in \Pi^{\text{Y}}_D} V(\bar{x}, y)$. Because we are optimizing over a larger set, $V(x^*, \bar{y}) \leq V_t$ and $V(\bar{x}, y^*) \geq V_t$. And, by our assumption about the accuracy of our oracles, $V(x, \bar{y}) \leq V(x^*, \bar{y}) + \epsilon$ and $V(\bar{x}, y) \geq V(\bar{x}, y^*) - \epsilon$.

Since x is a row in $\bar{R}^t$, we have $V(x, \bar{y}) \geq V^t$, and similarly $V(\bar{x}, y) \leq V^t$. Combining all the above inequalities yields

$$V^t - \epsilon \leq V(x, \bar{y}) - \epsilon \leq V(x^*, \bar{y}) \leq V^t \leq V(\bar{x}, y^*) \leq V(\bar{x}, y) + \epsilon \leq V^t + \epsilon$$

Letting V^* be the value of the game M, we know that $V(x^*, \bar{y}) \leq V^* \leq V(\bar{x}, y^*)$, and so we conclude $|V^t - V^*| \leq \epsilon$.

Now, we wish to show that player one can do almost as well as playing the minimax optimal strategy to M by playing $\bar{x}$. For all $y' \in \Pi^{\text{Y}}_D$, we have $V(\bar{x}, y') \leq V(\bar{x}, y^*) \leq V^t + \epsilon$. Since $V^t \leq V^* + \epsilon$, we have $V(\bar{x}, y') \leq V^* + 2\epsilon$, and so by playing $\bar{x}$, player one does almost as well as playing an optimal strategy. A symmetric argument shows the corresponding bound for the second player, and so $(\bar{x}, \bar{y})$ is a $2\epsilon-$approximate minimax equilibrium.

A.3 Proof of Theorem 3

Π^{X}_{ND} is a polytope in $\mathbb{R}^m$, with extreme points and directions corresponding to pure policies in Π^{X}_D. (See, for example, equation (2) in [3]). A mixed strategy for the row player in M corresponds to a stochastic policy $\bar{x} \in \Pi^{\text{X}}_{ND}$. As a consequence of the representation theorem (one version can be found in [1, Corollary 2.1]), $\bar{x}$ can be represented as a convex combination of at most $m+1$ extreme points and directions of Π^{X}_{ND}. In fact, since we have ruled out policies with infinite visitation frequencies (which correspond to extreme rays), $\bar{x}$ is a convex combination of at most $m+1$ extreme points. Since these extreme points are deterministic policies, any stochastic policy can be written as a convex combination of at most $m+1$ deterministic policies. In particular, this holds for the minimax optimal policy $\bar{x}^*$.