Mean-Variance Optimization in Markov Decision Processes

Shie Mannor SHIE@EE.TECHNION.AC.IL
Technion, Israel Institute of Technology

John N. Tsitsiklis JNT@MIT.EDU
Massachusetts Institute of Technology, Cambridge, MA

Abstract

We consider finite horizon Markov decision processes under performance measures that involve both the mean and the variance of the cumulative reward. We show that either randomized or history-based policies can improve performance. We prove that the complexity of computing a policy that maximizes the mean reward under a variance constraint is NP-hard for some cases, and strongly NP-hard for others. We finally offer pseudopolynomial exact and approximation algorithms.

1. Introduction

The classical theory of Markov decision processes (MDPs) deals with the maximization of the cumulative (possibly discounted) expected reward, to be denoted by W. However, a risk-averse decision maker may be interested in additional distributional properties of W. In this paper, we focus on the case where the decision maker is interested in both the mean and the variance of the cumulative reward, and we explore the associated computational issues.

Risk aversion in MDPs is of course an old subject. In one approach, the focus is on the maximization of $\mathbb{E}[U(W)]$, where U is a concave utility function. Problems of this type can be handled by state augmentation, e.g., Bertsekas (1995), namely, by introducing an auxiliary state variable that keeps track of the cumulative past reward. In a few special cases, e.g., with an exponential utility function, state augmentation is unnecessary, and optimal policies can be found by solving a modified Bellman equation such as Chung & Sobel (1987). Another interesting case where optimal policies can be found efficiently involves piecewise linear utility functions with a single break point; see Liu & Koenig (2005).

In another approach, the objective is to optimize a so-called coherent risk measure (Artzner et al., 1999), which turns out to be equivalent to a robust optimization problem: one assumes a family of probabilistic models and optimizes the worst-case performance over this family. In the multistage case (Riedel, 2004), problems of this type can be difficult (Le Tallec, 2007), except for some special cases (Iyengar, 2005; Nilim & El Ghaoui, 2005) that can be reduced to Markov games (Shapley, 1953).

Mean-variance optimization lacks some of the desirable properties of approaches involving coherent risk measures and sometimes leads to counterintuitive policies. Bellman's principle of optimality does not hold, and as a consequence, a decision maker who has received unexpectedly large rewards in the first stages, may actively seek to incur losses in subsequent stages in order to keep the variance small. Nevertheless, mean-variance optimization is an important approach in financial decision making e.g., Luenberger (1997), especially for static (one-stage) problems. Consider, for example, a fund manager who is interested in the 1-year performance of the fund, as measured by the mean and variance of the return. Assuming that the manager is allowed to undertake periodic re-balancing actions in the course of the year, one obtains a Markov decision process with mean-variance criteria. Mean-variance optimization can also be a meaningful objective in various engineering contexts. Consider, for example, an engineering process whereby a certain material is deposited on a surface. Suppose that the primary objective is to maximize the amount deposited, but that there is also an interest in having all manufactured components be similar to each other; this secondary objective can be addressed by keeping the variance of the amount deposited small.

We note that expressions for the variance of the discounted reward for stationary policies were developed in Sobel (1982). However, these expressions are quadratic in the underlying transition probabilities,

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

and do not lead to convex optimization problems.

Motivated by considerations such as the above, this paper deals with the computational complexity aspects of mean-variance optimization. The problem is not straightforward for various reasons. One is the absence of a principle of optimality that could lead to simple recursive algorithms. Another reason is that, as is evident from the formula $\mathrm{Var}(W) = \mathbb{E}[W^2] - (\mathbb{E}[W])^2$, the variance is not a linear function of the probability measure of the underlying process. Nevertheless, $\mathbb{E}[W^2]$ and $\mathbb{E}[W]$ are linear functions, and as such can be addressed simultaneously using methods from multicriteria or constrained Markov decision processes (Altman, 1999). Indeed, we will use such an approach in order to develop pseudopolynomial exact or approximation algorithms. On the other hand, we will also obtain various NP-hardness results, which show that there is little hope for significant improvement of our algorithms.

The rest of the paper is organized as follows. In Section 2, we describe the model and our notation. We also define various classes of policies and performance objectives of interest. In Section 3, we compare different policy classes and show that performance typically improves strictly as more general policies are allowed. In Section 4, we establish NP-hardness results for the policy classes we have introduced. Then, in Sections 5 and 6, we develop exact and approximate pseudopolynomial time algorithms. Unfortunately, such algorithms do not seem possible for some of the more restricted classes of policies, due to strong NP-completeness results established in Section 4. Finally, Section 7 contains some brief concluding remarks. Some of the proofs are deferred to Mannor & Tsitsiklis (2011).

2. The Model

In this section, we define the model, notation, and performance objectives that we will be studying. Throughout, we focus on finite horizon problems. [1]

2.1. Markov Decision Processes

We consider a Markov decision process (MDP) with finite state, action, and reward spaces. An MDP is formally defined by a sextuple $\mathcal{M} = (T, \mathcal{S}, \mathcal{A}, \mathcal{R}, p, g)$ where:

(a) T, a positive integer, is the time horizon;

(b) $\mathcal{S}$ is a finite collection of states, one of which is designated as the initial state;

(c) $\mathcal{A}$ is a collection of finite sets of possible actions, one set for each state;

(d) $\mathcal{R}$ is a finite subset of $\mathbb{Q}$ (the set of rational numbers), and is the set of possible values of the immediate rewards. We let $K = \max_{r \in \mathcal{R}} |r|$.

(e) $p : \{0, \ldots, T-1\} \times \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{Q}$ describes the transition probabilities. In particular, $p_t(s' \mid s, a)$ is the probability that the state at time $t+1$ is s', given that the state at time t is s, and that action a is chosen at time t.

(d) $g : \{0, \ldots, T-1\} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to \mathbb{Q}$ is a set of reward distributions. In particular, $g_t(r \mid s, a)$ is the probability that the immediate reward at time t is r, given that the state and action at time t is s and a, respectively.

With few exceptions (e.g., for the time horizon T), we use capital letters to denote random variables, and lower case letters to denote ordinary variables. The process starts at the designated initial state. At every stage $t = 0, 1, \ldots, T-1$, the decision maker observes the current state S_t and chooses an action A_t. Then, an immediate reward R_t is obtained, distributed according to $g_t(\cdot \mid S_t, A_t)$, and the next state S_{t+1} is chosen, according to $p_t(\cdot \mid S_t, A_t)$. Note that we have assumed that the possible values of the immediate reward and the various probabilities are all rational numbers. This is in order to address the computational complexity of various problems within the standard framework of digital computation. Finally, we will use the notation $x_{0:t}$ to indicate the tuple $(x_0, \ldots, x_t)$.

2.2. Policies

We will use the symbol π to denote policies. Under a *deterministic policy* $\pi = (\mu_0, \ldots, \mu_{T-1})$, the action at each time t is determined according to a mapping μ_t whose argument is the history $H_t = (S_{0:t}, A_{0:t-1}, R_{0:t-1})$ of the process, by letting $A_t = \mu_t(H_t)$. We let Π_h be the set of all such history-based policies. (The subscripts are used as a mnemonic for the variables on which the action is allowed to depend.) We will also consider *randomized* policies. For this purpose, we assume that there is available a sequence of i.i.d. uniform random variables $U_0, U_1, \ldots, U_{T-1}$, which are independent from everything else. In a randomized policy, the action at time t is determined by letting $A_t = \mu_t(H_t, U_{0:t})$. Let $\Pi_{h,u}$ be the set of all randomized policies.

In classical MDPs, it is well known that restricting to Markovian policies (policies that take into account

[1] Some of the results such as the approximation algorithms of Section 6 can be extended to the infinite horizon discounted case; this is beyond the scope of this paper.

only the current state S_t) results in no loss of performance. In our setting, there are two different possible "states" of interest: the original state S_t, or the augmented state (S_t, W_t), where

$$W_t = \sum_{k=0}^{t-1} R_k,$$

(with the convention that $W_0 = 0$). Accordingly, we define the following classes of policies: $\Pi_{t,s}$ (under which $A_t = \mu_t(S_t)$), and $\Pi_{t,s,w}$ (under which $A_t = \mu_t(S_t, W_t)$), and their randomized counterparts $\Pi_{t,s,u}$ (under which $A_t = \mu_t(S_t, U_t)$), and $\Pi_{t,s,w,u}$ (under which $A_t = \mu_t(S_t, W_t, U_t)$. Notice that

$$\Pi_{t,s} \subset \Pi_{t,s,w} \subset \Pi_h,$$

and similarly for their randomized counterparts.

2.3. Performance Criteria

Once a policy π and an initial state s is fixed, the cumulative reward W_T becomes a well-defined random variable. The performance measures of interest are its mean and variance, defined by $J_\pi = \mathbb{E}_\pi[W_T]$ and $V_\pi = \mathrm{Var}_\pi(W_T)$, respectively. Under our assumptions (finite horizon, and bounded rewards), it follows that there are finite upper bounds of KT and K^2T^2, for $|J_\pi|$ and V_π, respectively, independent of the policy.

Given our interest in complexity results, we will focus on "decision" problems that admit a yes/no answer, except for Section 6. We define the following problem.

Problem MV-MDP**(Π):** Given an MDP $\mathcal{M}$ and rational numbers λ, v, does there exist a policy in the set Π such that $J_\pi \geq \lambda$ and $V_\pi \leq v$?

Clearly, an algorithm for the problem MV-MDP(Π) can be combined with binary search to solve (up to any desired precision) the problem of maximizing the expected value of W_T subject to an upper bound on its variance, or the problem of minimizing the variance of W_T subject to a lower bound on its mean.

3. Comparison of Policy Classes

Our first step is to compare the performance obtained from different policy classes. We introduce some terminology. Let Π and Π' be two policy classes. We say that Π is *inferior* to Π' if, loosely speaking, the policy class Π' can always match or exceed the "performance" of policy class Π, and for some instances it can exceed it strictly. Formally, Π is inferior to Π' if the following hold: (i) if $(\mathcal{M}, c, d)$ is a "yes" instance of MV-MDP(Π), then it is also a "yes" instance of MV-MDP(Π'); (ii) there exists some $(\mathcal{M}, c, d)$ which is a "no" instance of MV-MDP(Π) but a "yes" instance of MV-MDP(Π'). Similarly, we say that two policy classes Π and Π' are *equivalent* if every "yes" (respectively, "no") instance of MV-MDP(Π) is a "yes" (respectively, "no") instance of MV-MDP(Π').

We define one more convenient term. A state s is said to be *terminal* if it is absorbing (i.e., $p_t(s \mid s, a) = 1$, for every t and a) and provides zero rewards (i.e., $g_t(0 \mid s, a) = 1$, for every t and a).

3.1. Randomization Improves Performance

Our first observation is that randomization can improve performance. This is not surprising given that we are dealing simultaneously with two criteria, and that randomization is helpful in constrained MDPs (Altman, 1999).

Theorem 1. *(a) $\Pi_{t,s}$ is inferior to $\Pi_{t,s,u}$;*

(b) $\Pi_{t,s,w}$ is inferior to $\Pi_{t,s,w,u}$;

(c) Π_h is inferior to $\Pi_{h,u}$.

The proof of this theorem is provided in Mannor & Tsitsiklis (2011); it is based on a simple counterexample.

3.2. Information Improves Performance

We now show that in most cases, performance can improve strictly when we allow a policy to have access to more information. The only exception arises for the pair of classes $\Pi_{t,s,w,u}$ and $\Pi_{h,u}$, which we show in Section 5 to be equivalent (cf. Theorem 6).

Theorem 2. *(a) $\Pi_{t,s}$ is inferior to $\Pi_{t,s,w}$, and $\Pi_{t,s,u}$ is inferior to $\Pi_{t,s,w,u}$.*

(b) $\Pi_{t,s,w}$ is inferior to Π_h.

The proof of this theorem is provided in (Mannor & Tsitsiklis, 2011); it is constructive by providing an example.

4. Complexity Results

In this section, we establish that mean-variance optimization in finite horizon MDPs is unlikely to admit polynomial time algorithms, in contrast to classical MDPs.

Theorem 3. *The problem* MV-MDP*(Π) is NP-hard, when Π is $\Pi_{t,s,w}$, $\Pi_{t,s,w,u}$, Π_h, or $\Pi_{h,u}$.*

The proof of this theorem is provided in Mannor & Tsitsiklis (2011); it is based on a reduction from the subset sum problem.

The proof of Theorem 3 also applies to the policy classes $\Pi_{t,s}$ and $\Pi_{t,s,u}$. However, for these two classes, a stronger result is possible. Recall that a problem is *strongly NP-hard*, if it remains NP-hard when restricted to instances in which the numerical part of the instance description involves "small" numbers; see Garey & Johnson (1979) for a precise definition.

Theorem 4. *If Π is either $\Pi_{t,s}$ or $\Pi_{t,s,u}$, the problem* MV-MDP*(Π) is strongly NP-hard.*

The proof of this theorem is provided in Mannor & Tsitsiklis (2011); it involves a reduction from the 3-Satisfiability problem.

5. Exact Algorithms

The comparison and complexity results of the preceding two sections indicate that the policy classes $\Pi_{t,s}$, $\Pi_{t,s,w}$, $\Pi_{t,s,u}$, and Π_h are inferior to the class $\Pi_{h,u}$, and furthermore some of them ($\Pi_{t,s}$, $\Pi_{t,s,w}$) appear to have higher complexity. Thus, there is no reason to consider them further. While the problem MV-MDP($\Pi_{h,u}$) is NP-hard, there is still a possibility for approximate or pseudopolynomial time algorithms. In this section, we focus on exact pseudopolynomial time algorithms.

Our approach involves an augmented state, defined by $X_t = (S_t, W_t)$. Let $\mathcal{X}$ be the set of all possible values of the augmented state. Let $|\mathcal{S}|$ be the cardinality of the set $\mathcal{S}$. Let $|\mathcal{R}|$ be the cardinality of the set $\mathcal{R}$. Recall also that $K = \max_{r\in\mathcal{R}} |r|$. If we assume that the immediate rewards are integers, then W_t is an integer between $-KT$ and KT. In this case, the cardinality $|\mathcal{X}|$ of the augmented state space $\mathcal{X}$ is bounded by $|\mathcal{S}| \cdot (2KT + 1)$, which is polynomial. Without the integrality assumption, the cardinality of the set $\mathcal{X}$ remains finite, but it can increase exponentially with T. For this reason, we study the integer case separately in Section 5.2.

5.1. State-Action Frequencies

In this section, we provide some results on the representation of MDPs in terms of a state-action frequency polytope, thus setting the stage for our subsequent algorithms.

For any policy $\pi \in \Pi_{h,u}$, and any $x \in \mathcal{X}$, $a \in \mathcal{A}$, we define the state-action frequencies at time t by

$$z_t^\pi(x,a) = \mathbb{P}_\pi(X_t = x, A_t = a), \qquad t = 0, 1, \ldots, T-1,$$

and

$$z_t^\pi(x) = \mathbb{P}_\pi(X_t = x), \qquad t = 0, 1, \ldots, T.$$

Let z^π be a vector that lists all of the above defined state-action frequencies.

For any family Π of policies, let $Z(\Pi) = \{z^\pi \mid \pi \in \Pi\}$. The following result is well known (Altman, 1999). It asserts that any feasible state-action frequency vector can be attained by policies that depend only on time, the (augmented) state, and a randomization variable. Furthermore, the set of feasible state-action frequency vectors is a polyhedron, hence amenable to linear programming methods.

Theorem 5. *(a) We have $Z(\Pi_{h,u}) = Z(\Pi_{t,s,w,u})$.*

(b) The set $Z(\Pi_{h,u})$ is a polyhedron, specified by $O(T\cdot|\mathcal{X}|\cdot|\mathcal{A}|)$ linear constraints.

Note that a certain mean-variance pair (λ, v) is attainable by a policy in $\Pi_{h,u}$ if and only if there exists some $z \in Z(\Pi_{h,u})$ that satisfies

$$\sum_{(s,w)\in\mathcal{X}} w z_T(s,w) \quad = \quad \lambda, \tag{1}$$

$$\sum_{(s,w)\in\mathcal{X}} w^2 z_T(s,w) \quad = \quad v + \lambda^2. \tag{2}$$

Furthermore, since $Z(\Pi_{h,u}) = Z(\Pi_{t,s,w,u})$, it follows that if a pair (λ, v) is attainable by a policy in $\Pi_{h,u}$, it is also attainable by a policy in $\Pi_{t,s,w,u}$. This establishes the following result.

Theorem 6. *The policy classes $\Pi_{h,u}$ and $\Pi_{t,s,w,u}$ are equivalent.*

Note that checking the feasibility of the conditions $z \in Z(\Pi_{h,u})$, (1), and (2) amounts to solving a linear program, with a number of constraints proportional to the cardinality of the augmented state space $\mathcal{X}$ and, therefore, in general, exponential in T.

5.2. Integer Rewards

In this section, we assume that the immediate rewards are integers, with absolute value bounded by K, and we show that pseudopolynomial time algorithms are possible. Recall that an algorithm is a pseudopolynomial time algorithm if its running time is polynomial in K and the instance size. (This is in contrast to polynomial time algorithms in which the running time can only grow as a polynomial of $\log K$.)

Theorem 7. *Suppose that the immediate rewards are integers, with absolute value bounded by K. Consider the following two problems:*

(i) determine whether there exists a policy in $\Pi_{h,u}$ for which $(J_\pi, V_\pi) = (\lambda, v)$, where λ and v are given rational numbers; and,

(ii) determine whether there exists a policy in $\Pi_{h,u}$ for which $J_\pi = \lambda$ and $V_\pi \leq v$, where λ and v are given rational numbers.

Then,

(a) these two problems admit a pseudopolynomial time algorithm; and,

(b) unless P=NP, these problems cannot be solved in polynomial time.

Proof.

(a) As already discussed, these problems amount to solving a linear program. In the integer case, the number of variables and constraints is bounded by a polynomial in K and the instance size. The result follows because linear programming can be solved in polynomial time.

(b) This is proved by considering the special case where $\lambda = v = 0$ and the exact same argument as in the proof of Theorem 3; see Mannor & Tsitsiklis (2011). □

Similar to constrained MDPs, mean-variance optimization involves two different performance criteria. Unfortunately, however, the linear programming approach to constrained MDPs does not translate into an algorithm for the problem $\textsc{mv-mdp}(\Pi_{h,u})$. The reason is that the set

$$P_{MV} = \{(J_\pi, V_\pi) \mid \pi \in \Pi_{h,u}\}$$

of achievable mean-variance pairs need not be convex. To bring the constrained MDP methodology to bear on our problem, instead of focusing on the pair (J_π, V_π), we define $Q_\pi = \mathbb{E}_\pi[W_T^2]$, and focus on the pair (J_π, Q_π). This is now a pair of objectives that depend *linearly* on the state frequencies associated with the final augmented state X_T. Accordingly, we define

$$P_{MQ} = \{(J_\pi, Q_\pi) \mid \pi \in \Pi_{h,u}\}.$$

Note that P_{MQ} is a polyhedron, because it is the image of the polyhedron $Z(\Pi_{h,u})$ under the linear mapping specified by the left-hand sides of Eqs. (1)-(2). In contrast, P_{MV} is the image of P_{MQ} under a nonlinear mapping:

$$P_{MV} = \{(\lambda, q - \lambda^2) \mid (\lambda, q) \in P_{MQ}\},$$

and is not, in general, a polyhedron.

As a corollary of the above discussion, and for the case of integer rewards, we can exploit convexity to devise pseudopolynomial algorithms for problems that can be formulated in terms of the convex set P_{MQ}. On the other hand, because of the non-convexity of P_{MV}, we have not been able to devise pseudopolynomial time algorithms for the problem $\textsc{mv-mdp}(\Pi_{h,u})$, or even the simpler problem of deciding whether there exists a policy $\pi \in \Pi_{h,u}$ that satisfies $V_\pi \le v$, for some given number v, except for the very special case where $v = 0$, which is the subject of our next result. For a general v, an approximation algorithm will be presented in the next section.

Theorem 8. *(a) If there exists some $\pi \in \Pi_{h,u}$ for which $V_\pi = 0$, then there exists some $\pi' \in \Pi_{t,s,w}$ for which $V_{\pi'} = 0$.*

(b) Suppose that the immediate rewards are integers, with absolute value bounded by K. Then the problem of determining whether there exists a policy $\pi \in \Pi_{h,u}$ for which $V_\pi = 0$ admits a pseudopolynomial time algorithm.

Proof.

(a) Suppose that there exists some $\pi \in \Pi_{h,u}$ for which $V_\pi = 0$. By Theorem 6, π can be assumed, without loss of generality, to lie in $\Pi_{t,s,w,u}$. Let $\mathrm{Var}_\pi(W_T \mid U_{0:T})$, be the conditional variance of W_T, conditioned on the realization of the randomization variables $U_{0:T}$. We have $\mathrm{Var}_\pi(W_T) \ge \mathbb{E}_\pi[\mathrm{Var}_\pi(W_T \mid U_{0:T})]$, which implies that there exists some $u_{0:T}$ such that $\mathrm{Var}_\pi(W_T \mid U_{0:T} = u_{0:T}) = 0$. By fixing the randomization variables to this particular $u_{0:T}$, we obtain a deterministic policy, in $\Pi_{t,s,w}$ under which the reward variance is zero.

(b) If there exists a policy under which $V_\pi = 0$, then there exists an integer k, with $|k| \le KT$ such that, under this policy, W_T is guaranteed to be equal to k. Thus, we only need to check, for each k in the relevant range, whether there exists a policy such that $(J_\pi, V_\pi) = (k, 0)$. By Theorem 7, this can be done in pseudopolynomial time. □

The approach in the proof of part (b) above leads to a short argument, but yields a rather inefficient (albeit pseudopolynomial) algorithm. A much more efficient and simple algorithm is obtained by realizing that the question of whether W_T can be forced to be k, with probability 1, is just a reachability game: the decision maker picks the actions and an adversary picks the ensuing transitions and rewards (among those that have positive probability of occurring). The decision maker wins the game if it can guarantee that $W_T = k$. Such sequential games are easy to solve in time polynomial in the number of (augmented) states, decisions, and the time horizon, by a straightforward backward recursion. On the other hand a genuinely polynomial time algorithm does not appear to be possible; indeed, the proof of Theorem 3 shows that the problem is NP-complete.

6. Approximation Algorithms

In this section, we deal with the optimization counterparts of the problem MV-MDP($\Pi_{h,u}$). We are interested in computing approximately the following two functions:

$$v^*(\lambda) = \inf_{\{\pi\in\Pi_{h,u}:J_\pi\geq\lambda\}} V_\pi, \qquad (3)$$

and

$$\lambda^*(v) = \sup_{\{\pi\in\Pi_{h,u}:V_\pi\leq v\}} J_\pi. \qquad (4)$$

If the constraint $J_\pi \geq \lambda$ (respectively, $V_\pi \leq v$) is infeasible, we use the standard convention $v^*(\lambda) = \infty$ (respectively, $\lambda^*(v) = -\infty$). Note that the infimum and supremum in the above definitions are both attained, because the set P_{MV} of achievable mean-variance pairs is the image of the polyhedron P_{MQ} under a continuous map, and is therefore compact.

We do not know how to efficiently compute or even generate a uniform approximation of either $v^*(\lambda)$ or $\lambda^*(v)$ (i.e., find a value v' between $v^*(\lambda) - \epsilon$ and $v^*(\lambda)+\epsilon$, and similarly for $\lambda^*(v)$). One key obstacle for obtaining a uniform approximation arises due to the difficulty of handling the discontinuity of $v^*(\lambda)$ near the edges where the constraint $J_\pi \geq \lambda$ becomes infeasible. In the following two results we consider a weaker notion of approximation that is computable in pseudopolynomial time. We discuss $v^*(\lambda)$ as the issues for $\lambda^*(v)$ are similar. Another approximation algorithm that is based on set-valued dynamic programming is presented in Mannor & Tsitsiklis (2011).

For any positive ϵ and ν, we will say that $\hat{v}(\cdot)$ is an (ϵ, ν)-aproximation of $v^*(\cdot)$ if, for every λ,

$$v^*(\lambda - \nu) - \epsilon \leq \hat{v}(\lambda) \leq v^*(\lambda + \nu) + \epsilon. \qquad (5)$$

This is an approximation of the same kind as those considered in Papadimitriou & Yannakakis (2000): it returns a value $\hat{v}$ such that $(\lambda, \hat{v})$ is an element of the "$(\epsilon+\nu)$-approximate Pareto boundary" of the set P_{MV}. For a different view, the graph of the function $\hat{v}(\cdot)$ is within Hausdorf distance $\epsilon + \nu$ from the graph of the function $v^*(\cdot)$.

We will show how to compute an (ϵ, ν)-aproximation in time which is pseudopolynomial, and polynomial in the parameters $1/\epsilon$, and $1/\nu$.

We start in Section 6.1 with the case of integer rewards, and build on the pseudopolynomial time algorithms of the preceding section. We then consider the case of general rewards in Section 6.2.

6.1. Integer Rewards

In this section, we prove the following result.

Theorem 9. *Suppose that the immediate rewards are integers. There exists an algorithm that, given ϵ, ν, and λ, outputs a value $\hat{v}(\lambda)$ that satisfies (5), and which runs in time polynomial in $|\mathcal{S}|$, $|\mathcal{A}|$, T, K, $1/\epsilon$, and $1/\nu$.*

Proof. Since the rewards are bounded in absolute value by K, we have $v^*(\lambda) = \infty$ for $\lambda > KT$ and $v^*(\lambda) = v^*(-KT)$ for $\lambda < -KT$. For this reason, we only need to consider $\lambda \in [-KT, KT]$. To simplify the presentation, we assume that $\epsilon = \nu$. We let δ be such that $\epsilon = 3\delta KT$.

The algorithm is as follows. We consider grid points λ_i defined by $\lambda_i = -KT + (i-1)\delta$, $i = 1, \ldots, n$, where n is chosen so that $\lambda_{n-1} \leq KT$, $\lambda_n > KT$. Note that $n = O(KT/\delta)$. For $i = 1, \ldots, n-1$, we calculate $\hat{q}(\lambda_i)$, the smallest possible value of $\mathbb{E}[W_T^2]$, when $\mathbb{E}[W_T]$ is restricted to lie in $[\lambda_i, \lambda_{i+1}]$. Formally,

$$\hat{q}(\lambda_i) = \min\Big\{q \;\Big|\; \exists\ \lambda' \in [\lambda_i, \lambda_{i+1}] \text{ s.t. } (\lambda', q) \in P_{MQ}\Big\}.$$

We let $\hat{u}(\lambda_i) = \hat{q}(\lambda_i) - \lambda_{i+1}^2$, which can be interpreted as an estimate of the least possible variance when $\mathbb{E}[W_T]$ is restricted to the interval $[\lambda_i, \lambda_{i+1}]$. Finally, we set

$$\hat{v}(\lambda) = \min_{i\geq k} \hat{u}(\lambda_i), \qquad \text{if } \lambda \in [\lambda_k, \lambda_{k+1}].$$

The main computational effort is in computing $\hat{q}(\lambda_i)$ for every i. Since P_{MQ} is a polyhedron, this amounts to solving $O(KT/\delta)$ linear programming problems. Thus, the running time of the algorithm has the claimed properties.

We now prove correctness. Let $q^*(\lambda) = \min\{q \mid (\lambda, q) \in P_{MQ}\}$, and $u^*(\lambda) = q^*(\lambda) - \lambda^2$, which is the least possible variance for a given value of λ. Note that $v^*(\lambda) = \min\{u^*(\lambda') \mid \lambda' \geq \lambda\}$.

We have $\hat{q}(\lambda_i) \leq q^*(\lambda')$, for all $\lambda' \in [\lambda_i, \lambda_{i+1}]$. Also, $-\lambda_{i+1}^2 \leq -(\lambda')^2$, for all $\lambda' \in [\lambda_i, \lambda_{i+1}]$. By adding these two inequalities, we obtain $\hat{u}(\lambda_i) \leq u^*(\lambda')$, for all $\lambda' \in [\lambda_i, \lambda_{i+1}]$. Given some λ, let k be such that $\lambda \in [\lambda_k, \lambda_{k+1}]$. Then,

$$\hat{v}(\lambda) = \min_{i\geq k} \hat{u}(\lambda_i) \leq \min_{\lambda'\geq\lambda_k} u^*(\lambda') \leq \min_{\lambda'\geq\lambda} u^*(\lambda') = v^*(\lambda'),$$

so that $\hat{v}(\lambda)$ is always an underestimate of $v^*(\lambda)$.

We now prove a reverse inequality. Fix some λ and let k be such that $\lambda \in [\lambda_k, \lambda_{k+1}]$. Let $i \geq k$ be such that $\hat{v}(\lambda) = \hat{u}(\lambda_i)$. Let also $\bar{\lambda} \in [\lambda_i, \lambda_{i+1}]$ be such that $q^*(\bar{\lambda}) = \hat{q}(\lambda_i)$. Note that

$$\begin{aligned} \lambda_{i+1}^2 - \bar{\lambda}^2 &\leq \lambda_{i+1}^2 - \lambda_i^2 = \delta(\lambda_i + \lambda_{i+1}) \\ &\leq 2\delta(KT + \delta) \leq 3\delta KT. \end{aligned} \tag{6}$$

Then,

$$\begin{aligned} \hat{v}(\lambda) \overset{(a)}{=} \hat{u}(\lambda_i) &\overset{(b)}{=} \hat{q}(\lambda_i) - \lambda_{i+1}^2 \\ &\overset{(c)}{=} q^*(\bar{\lambda}) - \lambda_{i+1}^2 \\ &\overset{(d)}{\geq} q^*(\bar{\lambda}) - \bar{\lambda}^2 - 3\delta KT \\ &\overset{(e)}{=} u^*(\bar{\lambda}) - 3\delta KT \\ &\overset{(f)}{\geq} v^*(\bar{\lambda}) - 3\delta KT \\ &\overset{(g)}{\geq} v^*(\lambda - \delta) - 3\delta KT \\ &\overset{(h)}{\geq} v^*(\lambda - \epsilon) - \epsilon. \end{aligned}$$

In the above, (a) holds by the definition of i; (b) by the definition of $\hat{u}(\lambda_i)$; (c) by the definition of $\bar{\lambda}$; and (d) follows from Eq. (6). Equality (e) follows from the definition of $u^*(\cdot)$. Inequality (f) follows from the definition of $v^*(\cdot)$; and (g) is obtained because $v^*(\cdot)$ is nondecreasing and because $\bar{\lambda} \geq \lambda - \delta$. (The latter fact is seen as follows: (i) if $i > k$, then $\lambda \leq \lambda_{k+1} \leq \lambda_i \leq \bar{\lambda}$; (ii) if $i = k$, then both λ and $\bar{\lambda}$ belong to $[\lambda_k, \lambda_{k+1}]$, and their difference is at most δ.) Inequality (h) is obtained because of the definition $\epsilon = 3\delta KT$, the observation $\delta < \epsilon$, and the monotonicity of $v^*(\cdot)$. □

6.2. General Rewards

When rewards are arbitrary, we can discretize the rewards and obtain a new MDP. The new MDP is equivalent to one with integer rewards to which the algorithm of the preceding subsection can be applied. This is a legitimate approximation algorithm for the original problem because, as we will show shortly, the function $v^*(\cdot)$ changes very little when we discretize using a fine enough discretization.

We are given an original MDP $\mathcal{M} = (T, \mathcal{S}, \mathcal{A}, \mathcal{R}, p, g)$ in which the rewards are rational numbers in the interval $[-K, K]$, and an approximation parameter ϵ. We fix a positive number δ, a discretization parameter whose value will be specified later. We then construct a new MDP $\mathcal{M}' = (T, \mathcal{S}, \mathcal{A}, \mathcal{R}', p, g')$, in which the rewards are rounded down to an integer multiple of δ. More precisely, all elements of the reward range $\mathcal{R}'$ are integer multiples of δ, and for every $t, s, a \in \{0, 1, \ldots, T-1\} \times \mathcal{S} \times \mathcal{A}$, and any integer n, we have

$$g_t(\delta n \mid s, a) = \sum_{r:\ \delta n \leq r < \delta(n+1)} g_t(r \mid s, a).$$

We denote by J, Q and by J', Q' the first and second moments of the total reward in the original and new MDPs, respectively. Let $\Pi_{h,u}$ and $\Pi'_{h,u}$ be the sets of (randomized, history-based) policies in $\mathcal{M}$ and $\mathcal{M}'$, respectively. Let P_{MQ} and P'_{MQ} be the associated polyhedra.

We want to to argue that the mean-variance tradeoff curves for the two MDPs are close to each other. This is not entirely straightforward because the augmented state spaces (which include the possible values of the cumulative rewards W_t) are different for the two problems and, therefore, the sets of policies are also different. We follow an approach that is based on a coupling argument.

Proposition 1. *There exists a polynomial function $c(K,T)$ such that the Hausdorf distance between P_{MQ} and P'_{MQ} is bounded above by $2KT^2\delta$. More precisely,*

(a) For every policy $\pi \in \Pi_{h,u}$, there exists a policy $\pi' \in \Pi'_{h,u}$ such that

$$\max\Big\{|J'_{\pi'} - J_\pi|,\ |Q'_{\pi'} - Q_\pi|\Big\} \leq 2KT^2\delta.$$

(b) Conversely, for every policy $\Pi'_{h,u}$, there exists a policy $\Pi_{h,u}$ such that the above inequality again holds.

The proof of this proposition is provided in Mannor & Tsitsiklis (2011).

Theorem 10. *There exists an algorithm that, given ϵ, ν, and λ, outputs a value $\hat{v}(\lambda)$ that satisfies (5), and which runs in time polynomial in $|\mathcal{S}|$, $|\mathcal{A}|$, T, K, $1/\epsilon$, and $1/\nu$.*

Proof. Assume for simplicity that $\nu = \epsilon$. Given the value of ϵ, let δ be such that $\epsilon/2 = 2KT^2\delta$, and construct the discretized MDP $\mathcal{M}'$. Run the algorithm from Theorem 9 to find an $(\epsilon/2, \epsilon/2)$-approximation $\hat{v}$ for $\mathcal{M}'$. Using Proposition 1, it is not hard to verify that this yields an (ϵ, ϵ)-approximation of $v^*(\lambda)$. □

7. Conclusions

We have shown that mean-variance optimization problems for MDPs are typically NP-hard, but sometimes admit pseudopolynomial approximation algorithms. We only considered finite horizon problems, but it is clear that the negative results carry over to their infinite horizon counterparts. Furthermore, given that the contribution of the tail of the time horizon in infinite horizon discounted problems (or in "proper" stochastic shortest path problems as in Bertsekas, 1995) can be made arbitrarily small, our approximation algorithms can also yield approximation algorithms for infinite horizon problems.

Our negative results apply to general MDPs. It would be interesting to determine whether the hardness results remain valid for specially structured MDPs. One possibly interesting special case involves multi-armed bandit problems: there are n separate MDPs ("arms"); at each time step, the decision maker has to decide which MDP to activate, while the other MDPs remain inactive. Of particular interest here are index policies that compute a value ("index") for each MDP and select an MDP with maximal index; such policies are often optimal for the classical formulations (see Gittins, 1979 and Whittle, 1988). Obtaining a policy that uses some sort of an index for the mean-variance problem or alternatively proving that such a policy cannot exist would be interesting.

We only considered mean-variance tradeoffs in this paper. However, there are other interesting and potentially useful criteria that can be used to incorporate risk into multi-stage decision making. For example, Liu & Koenig (2005) consider a utility function with a single switch. Many other risk aware criteria have been considered in the single stage case. It would be interesting to develop a comprehensive theory for the complexity of solving multi-stage decision problems under general (monotone convex or concave) utility function and under risk constraints. This is especially interesting for the approximation algorithms presented in Section 6.

Acknowledgements: This research was partially supported by the Israel Science Foundation (contract 890015), a Horev Fellowship, and the National Science Foundation under grant CMMI-0856063.

References

Altman, E. *Constrained Markov Decision Processes.* Chapman and Hall, 1999.

Artzner, P., Delbaen, F., Eber, J., and Heath, D. Coherent measures of risk. *Mathematical Finance*, 9 (3):203–228, 1999.

Bertsekas, D.P. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995.

Chung, K. and Sobel, M. Discounted MDP's: distribution functions and exponential utility maximization. *SIAM Journal on Control and Optimization*, 25(1): 49 – 62, 1987.

Garey, M. R. and Johnson, D. S. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* W.H. Freeman, New York, 1979.

Gittins, J. C. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):148–177, 1979.

Iyengar, G. Robust dynamic programming. *Mathematics of Operations Research*, 30:257–280, 2005.

Le Tallec, Y. *Robust, Risk-Sensitive, and Data-driven Control of Markov Decision Processes.* PhD thesis, Operations Research Center, MIT, Cambridge, MA, 2007.

Liu, Y. and Koenig, S. Risk-sensitive planning with one-switch utility functions: Value iteration. In *Proceedings of the Twentieth AAAI Conference on Artificial Intelligence*, pp. 993–999, 2005.

Luenberger, D. *Investment Science.* Oxford University Press, 1997.

Mannor, S. and Tsitsiklis, J. Mean-variance optimization in Markov decision processes. *CoRR*, abs/1104.5601, 2011. URL `http://arxiv.org/abs/1104.5601`.

Nilim, A. and El Ghaoui, L. Robust Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Papadimitriou, C. H. and Yannakakis, M. On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st Symposium on Foundations of Computer Science*, pp. 86–92, Washington, DC, USA, 2000.

Riedel, F. Dynamic coherent risk measures. *Stoch. Proc. Appl.*, 112:185–200, 2004.

Shapley, L. Stochastic games. *Proc. of National Academy of Science, Math.*, pp. 1095–1100, 1953.

Sobel, M.J. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19: 794–802, 1982.

Whittle, P. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25: 287–298, 1988.