European Journal of Operational Research 231 (2013) 645–653

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

Algorithmic aspects of mean–variance optimization in Markov decision processes

Shie Mannor [a,*], John N. Tsitsiklis [b]

[a] Department of Electrical and Engineering, Technion, Haifa 32000, Israel
[b] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, United States

ARTICLE INFO

Article history:
Received 12 June 2011
Accepted 12 June 2013
Available online 24 June 2013

Keywords:
Markov processes
Dynamic programming
Control
Complexity theory

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.

© 2013 Elsevier B.V. All rights reserved.

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 (e.g., trying to optimize the mean subject to a variance constraint or vice versa), 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 (Chung & Sobel, 1987). (The exponential utility function is often viewed as a surrogate for trading off mean and variance, on the basis of a single tunable parameter. The difficulty of solving mean–variance optimization problems—which is the focus of this paper—does provide some support for using a surrogate criterion, more amenable to exact optimization.) Another interesting case where optimal policies can be found efficiently involves a "one-switch utility functions" (the sum of a linear and an exponential) Liu and Koenig (2005), or piecewise linear utility functions with a single break point (Liu & Koenig, 2006).

In another approach, the objective is to optimize a so-called coherent risk measure (Artzner, Delbaen, Eber, & Heath, 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 or risk-sensitive utility functions (e.g., exponential utility functions) 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. Counterintuitive and seemingly "irrational" behavior (i.e., incompatible with expected utility maximization) can even arise in static problems under a mean–variance formulation: for example, under a variance constraint, one may prefer to forgo a profit which is guaranteed to be positive but has a positive variance. Nevertheless, mean–variance optimization is a common 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

* Corresponding author. Tel.: +972 4 8293284; fax: +972 4 8295757.
E-mail address: shie@ee.technion.ac.il (S. Mannor).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.06.019

performance of the fund whose investment strategies will be judged according to 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, and it is important to know the least possible variance achievable under a set target for the mean return. While the applicability of the financial strategies arising from mean–variance optimization in multi-period fund management can be debated (due to the "irrational" aspects mentioned above), mean–variance optimization is definitely 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. In general, the applicability of the formulations studied in this paper will depend on the specifics of a particular application.

Mean–variance optimization problems resembling ours have been studied in the literature. For example, (Guo, Ye, & Yin, 2012) consider a mean–variance optimization problem, but subject to a constraint on the vector of expected rewards starting from each state, which results in a simpler problem, amenable to a policy iteration approach. Collins (1997) provides an apparently exponential-time algorithm for a variant of our problem, and Tamar, Di-Castro, and Mannor (2012) propose a policy gradient approach that aims at a locally optimal solution. 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, and do not lead to convex optimization problems. Similarly, much of the earlier literature (see Kawai (1987), Huang & Kallenberg (1994) for a unified approach) on the problem provides various mathematical programming formulations. In general, these formulations either deal with problems that differ qualitatively focusing on the variation of reward from its average (Filar, Kallenberg, & Lee, 1989; White, 1992) from ours or are nonconvex, and therefore do not address the issue of polynomial-time solvability which is our focus. Indeed, we are not aware on any complexity results on mean–variance optimization problems. We finally note some interesting variance bounds obtained by Arlotto, Gans, and Steel (2013).

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.

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, $p_t(s'|s,a)$ 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, $g_t(r|s,a)$ 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 $g_t(\cdot|S_t,A_t)$, and the next state S_{t+1} is chosen, according to $p_t(\cdot|S_t,A_t)$. 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. Intuitively, at each point in time, the policy can pick an action at random, with the probability of each action determined by the current information (which is H_t as well as the outcomes of earlier randomizations). Randomness can always be simulated by using an independent uniform random variable as the seed, which leads to the following formal definition. We assume that there is

[1] Negative complexity results are straightforward to extend to the more general case of infinite horizon problems. Also, some of the positive 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.

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 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 $J_\pi \geqslant \lambda$ and $V_\pi \leqslant v$?

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., $p_t(s|s,a) = 1$, for every t and a) and provides zero rewards (i.e., $g_t(0|s,a) = 1$, for every t and a).

3.1. Randomization Improves Performance

Our first observation is that randomization can strictly improve performance. This is not surprising given that we are dealing simultaneously with two criteria, and that randomization is helpful in constrained MDPs e.g., (Altman, 1999). (Clearly, it is not the case that there will always be improvement—consider a case where rewards are identically zero, so that all policy classes offer the same performance. The content of our result is that certain policies are not "equivalent," meaning that there exist instances for which the resulting performance is different).

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 $\Pi_{h,u}$.*

Proof. It is clear that performance cannot deteriorate when randomization is allowed. It therefore suffices to display an instance in which randomization improves performance.

Consider a one-stage MDP ($T = 1$). At time 0, we are at the initial state and there are two available actions, a and b. The mean and variance of the resulting reward are both zero under action a, and both equal to 1 under action b. After the decision is made, the rewards are obtained and the process terminates. Thus $W_T = R_0$, the reward obtained at time 0.

Consider the problem of maximizing $\mathbb{E}[R_0]$ subject to the constraint that $\mathrm{var}(R_0) \leqslant 1/2$. There is only one feasible deterministic policy (choose action a), and it has zero expected reward. On the other hand, a randomized policy that chooses action b with probability p has an expected reward of p and the corresponding variance satisfies

$$\mathrm{var}(R_0) \leqslant \mathbb{E}[R_0^2] = p\mathbb{E}[R_0^2|A_0 = b] = 2p.$$

When $0 < p \leqslant 1/4$, such a randomized policy is feasible and improves upon the deterministic one.

Note that for the above instance we have $\Pi_{t,s} = \Pi_{t,s,w} = \Pi_h$, and $\Pi_{t,s,u} = \Pi_{t,s,w,u} = \Pi_{h,u}$. Hence the above example establishes all three of the claimed statements. □

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.*

Proof.

(a) Consider the following MDP, with time horizon $T = 2$. The process starts at the initial state s_0, at which there are two actions. Under action a_1, the immediate reward is zero and the process moves to a terminal state. Under action a_2, the immediate reward R_0 is either 0 or 1, with equal probability, and the process moves to state s_1. At state s_1, there are two actions, a_3 and a_4: under action a_3, the immediate reward R_1 is equal to 0, and under action a_4, it is equal to 1. We are interested in the optimal value of the expected reward $\mathbb{E}[W_2] = \mathbb{E}[R_0 + R_1]$, subject to the constraint that the variance is less than or equal to zero (and therefore equal to zero). Let p be the probability that action a_2 is

chosen at state s_0. If $p > 0$, and under any policy in $\Pi_{t,s,u}$, the reward R_0 at state s_0 has positive variance, and the reward R_1 at the next stage is uncorrelated with R_0. Hence, the variance of $R_0 + R_1$ is positive, and such a policy is not feasible; in particular, the constraint on the variance requires that $p = 0$. We conclude that the largest possible expected reward under any policy in $\Pi_{t,s,u}$ (and, a fortiori, under any policy in $\Pi_{t,s}$) is equal to zero. Consider now the following policy, which belongs to $\Pi_{t,s,w}$ and, a fortiori, to $\Pi_{t,s,w,u}$: at state s_0, choose action a_2; then, at state s_1, choose a_3 if $W_1 = R_0 = 1$, and choose a_4 if $W_1 = R_0 = 0$. In either case, the total reward is $R_0 + R_1 = 1$, while the variance of $R_0 + R_1$ is zero, thus ensuring feasibility. This establishes the first part of the theorem.

(b) Consider the following MDP, with time horizon $T = 3$. At state s_0 there is only one available action; the next state S_1 is either s_1 or s_1', with probability p and $1 - p$, respectively, and the immediate reward R_0 is zero. At either state s_1 or s_1', there is again only one available action; the next state, S_2, is s_2, and the reward R_1 is zero. At state s_2, there are two actions, a and b. Under action a, the mean and variance of the resulting reward R_2 are both zero, and under action b, they are both equal to 1. Let us examine the largest possible value of $\mathbb{E}[W_3] = \mathbb{E}[R_2]$, subject to the constraint $\mathrm{var}(W_2) \leqslant 1/2$. The class $\Pi_{t,s,w}$ contains two policies, corresponding to the two deterministic choices of an action at state s_2; only one of them is feasible (the one that chooses action a), resulting in zero expected reward. However, the following policy in Π_h has positive expected reward: choose action b at state s_2 if and only if the state at time 1 was equal to s_1 (which happens with probability p). As long as p is sufficiently small, the constraint $\mathrm{var}(W) \leqslant 1/2$ is met, and this policy is feasible. It follows that $\Pi_{t,s,w}$ is inferior to Π_h. □

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* $\Pi_{h,u}$.

Proof. We will actually show NP-hardness for the special case of MV-MDP(Π), in which we wish to determine whether there exists a policy whose reward variance is equal to zero. (In terms of the problem definition, this corresponds to letting $\lambda = -KT$ and $v = 0$.) The proof uses a reduction from the PARTITION problem: Given n positive integers, does there exist a subset B of $\{1, \ldots, n\}$ such that $\sum_{i \in B} r_i = \sum_{i \notin B} r_i$?

Given an instance $(r_1, \ldots, r_n)$ of PARTITION, and for any of the policy classes of interest, we construct an instance of MV-MDP(Π), with time horizon $T = n + 1$, as follows. At the initial state s_0, there is only one available action, resulting in zero immediate reward ($R_0 = 0$). With probability 1/2, the process moves to a terminal state; with probability 1/2, the process moves (deterministically) along a sequence of states $s_1, \ldots, s_n$. At each state s_i ($i = 1, \ldots, n$), there are two actions: a_i, which results in an immediate reward of r_i, and b_i, which results in an immediate reward of $-r_i$.

Suppose that there exists a set $B \subset \{1, \ldots, n\}$ such that $\sum_{i \in B} r_i = \sum_{i \notin B} r_i$. Consider the policy that chooses action a_i at state s_i if and only if $i \in B$. This policy achieves zero total reward, with probability 1, and therefore meets the zero variance constraint. Conversely, if a policy results in zero variance, then the total reward must be equal to zero, with probability 1, which implies that such a set B exists. This completes the reduction.

Note that this argument applies no matter which particular class of policies is being considered. □

The above proof 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.*

Proof. As in the proof of Theorem 3, we will prove the result for the special case of MV-MDP, in which we wish to determine whether there exists a policy under which the variance of the reward is equal to zero. The proof involves a reduction from the 3-Satisfiability problem (3SAT). An instance of 3SAT consists of n Boolean variables $x_1, \ldots, x_n$, and m clauses $C_1, \ldots, C_m$, with three literals per clause. Each clause is the disjunction of three literals, where a literal is either a variable or its negation. (For example, $x_2 \vee \bar{x}_4 \vee x_5$ is such a clause, where a bar stands for negation.) The question is whether there exists an assignment of truth values ("true" or "false") to the variables such that all clauses are satisfied.

Suppose that we are given an instance of 3SAT, with n variables and m clauses, $C_1, \ldots, C_m$. We construct an instance of MV-MDP(Π) as follows. There is an initial state s_0, a state d_0, a state c_j associated with each clause C_j, and a state y_i associated with each literal x_i. The actions, dynamics, and rewards are as follows:

(a) Out of state s_0, there is equal probability, $1/(m + 1)$, of reaching any one of the states $d_0, c_1, \ldots, c_m$, independent of the action; the immediate reward is zero.
(b) State d_0 is a terminal state. At each state c_j, there are three actions available: each action selects one of the three literals in the clause, and the process moves to the state y_i associated with that literal; the immediate reward is 1 if the literal appears in the clause unnegated, and -1 if the literal appears in the clause negated. For an example, suppose that the clause is of the form $x_2 \vee \bar{x}_4 \vee x_5$. Under the first action, the next state is y_2, and the reward is 1; under the second action, the next state is y_4 and the reward is -1; under the third action, the next state is y_5, and the reward is 1.
(c) At each state y_i, there are two possible actions a_i and b_i, resulting in immediate rewards of 1 and -1, respectively. The process then moves to the terminal state d_0.

Suppose that we have a "yes" instance of 3SAT, and consider a truth assignment that satisfies all clauses. We can then construct a policy in $\Pi_{t,s}$ (and a fortiori in $\Pi_{t,s,u}$), whose total reward is zero (and therefore has zero variance) as follows. If x_i is set to be true (respectively, false), we choose action b_i (respectively, a_i) at state y_i. At state c_j we choose an action associated with a literal that makes the clause to be true. Suppose that state c_j is visited after the first transition, i.e., $S_1 = c_j$. If the literal associated with the selected action at c_j is unnegated, e.g., the literal x_i, then the immediate reward is 1. Since this literal makes the clause to be true, it follows that the action chosen at the subsequent state, y_i, is b_i, resulting in a reward of -1, and a total reward of zero. The argument for the case where the literal associated with the selected action at state c_j is negated is similar. It follows that the total reward is zero, with probability 1.

For the converse direction, suppose that there exists a policy in $\Pi_{t,s}$, or more generally, in $\Pi_{t,s,u}$ under which the variance of the total reward is zero. Since the total reward is equal to 0 whenever the first transition leads to state d_0 (which happens with probability $1/(m+1)$, it follows that the total reward must be always zero. Consider now the following truth assignment: x_i is set to be true if and only if the policy chooses action b_i at state y_i, with positive probability. Suppose that the state visited after the first transition is c_j. Suppose that the action chosen at state c_j leads next to state y_i and that the literal x_i appears unnegated in clause C_j. Then, the reward at state c_j is 1, which implies that the reward at state y_i is -1. It follows that the action chosen at y_i is b_i, and therefore x_i has been set to be true. It follows that clause C_j is satisfied. A similar argument shows that clause C_j is satisfied when the literal x_i associated with the chosen action at c_j appears negated. In either case, we conclude that clause C_j is satisfied. Since every state c_j is possible at time 1, it follows that every clause is satisfied, and we have a "yes" instance of 3SAT. □

Because the immediate rewards are bounded, it is easily seen that an instance with general rewards is equivalent to one with all positive (or all negative) rewards. It follows that our negative complexity results remain valid even if we restrict to instances in which all rewards are positive (respectively, negative).

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

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

and

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

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 e.g., (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

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

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

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 programming problem, 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 $V_\pi \leqslant v$, 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. □

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 MV-MDP($\Pi_{h,u}$). The reason is that the set

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

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

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

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) and (2). In contrast, P_{MV} is the image of P_{MQ} under a nonlinear mapping:

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

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 MV-MDP($\Pi_{h,u}$), or even the simpler problem of deciding whether there exists a policy $\pi \in \Pi_{h,u}$ that satisfies $V_\pi \leqslant v$, 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 $\mathrm{var}_\pi(W_T|U_{0:T})$, be the conditional variance of W_T, conditioned on the realization of the randomization variables $U_{0:T}$. We have $\mathrm{var}_\pi(W_T) \geqslant \mathbb{E}_\pi[\mathrm{var}_\pi(W_T|U_{0:T})]$, which implies that there exists some $u_{0:T}$ such that $\mathrm{var}_\pi(W_T|U_{0:T} = u_{0:T}) = 0$. 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 $|k| \leqslant KT$ 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:

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

and

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

If the constraint $J_\pi \geqslant \lambda$ (respectively, $V_\pi \leqslant v$) 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 $\lambda^*(v)$). 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.

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

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

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)-approximation 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. We finally sketch an alternative algorithm in Section 6.3 based on set-valued dynamic programming.

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. Without loss of generality, and only for the purposes of this proof, we can and will assume that the immediate rewards are nonnegative. Indeed, if the immediate rewards range in $[-K,K]$ we can redefine them, by adding K to the reward at each stage. Then, $\hat{v}(\lambda)$ for the original problem will be equal to $\hat{v}(\lambda+K)$ for the new problem. Since the rewards are bounded by K, we have $v^*(\lambda)=\infty$ for $\lambda > KT$ and $v^*(\lambda) = v^*(0)$ for $\lambda < 0$. For this reason, we only need to consider $\lambda \in [0,KT]$. 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 $\lambda_i = (i-1)\delta$, $i = 1,\ldots,n$, where n is chosen so that $\lambda_{n-1} \leqslant KT$, $\lambda_n > KT$. 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,

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

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

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

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

We have $\hat{q}(\lambda_i) \leqslant q^*(\lambda')$, for all $\lambda' \in [\lambda_i,\lambda_{i+1}]$. Also, $-\lambda_{i+1}^2 \leqslant -(\lambda')^2$, for all $\lambda' \in [\lambda_i,\lambda_{i+1}]$. By adding these two inequalities, we obtain $\hat{u}(\lambda_i) \leqslant u^*(\lambda')$, 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,

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

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 $i \geqslant k$ 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

$$\lambda_{i+1}^2 - \bar{\lambda}^2 \leqslant \lambda_{i+1}^2 - \lambda_i^2 = \delta(\lambda_i + \lambda_{i+1}) \leqslant 2\delta(KT+\delta) \leqslant 3\delta KT. \tag{6}$$

Then,

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

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 $\bar{\lambda} \geqslant \lambda - \delta$. (The latter fact is seen as follows: (i) if $i > k$, then $\lambda \leqslant \lambda_{k+1} \leqslant \lambda_i \leqslant \bar{\lambda}$; (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 $v^*(\cdot)$. □

Theorem 9 allows us to construct an approximate Pareto boundary. In addition, one may be interested in obtaining corresponding policies. As is common in Markov decision theory, the construction of suitable policies is implicit in value function calculations, and is immediate from the proof Theorem 9, as we now describe. Suppose that are given some λ that happens to lie in some $[\lambda_k,\lambda_{k+1}]$. As in the proof of the theorem, we find some i such that $\hat{v}(\lambda) = \hat{u}(\lambda_i) = \hat{q}(\lambda_i) - \lambda_{i+1}^2$. From the definition of $\hat{q}(\lambda_i)$, there exists some $(\bar{\lambda},q) \in P_{MQ}$ with $\bar{\lambda} \in [\lambda_i,\lambda_{i+1}]$ and $q = q^*(\bar{\lambda}) = \hat{q}(\lambda_i)$. The key observation is that we can easily find a policy for which $\mathbb{E}[W_T]$ and $\mathbb{E}[W_T^2]$ are equal to $\bar{\lambda}$ and q, respectively. This is done by finding a corresponding state-action frequency vector in the polyhedron $Z(\Pi_{h,u})$ (which is a linear programming feasibility problem), and expressing that vector as a convex combination of extreme points of $Z(\Pi_{h,u})$. As is well known, extreme points of $Z(\Pi_{h,u})$ are associated with deterministic policies. The desired policy is a randomized policy obtained by combining these deterministic policies according to the coefficients involved in the convex combination. The policy constructed in this manner has a variance equal to

$$q - (\bar{\lambda})^2 = \hat{q}(\lambda_i) - (\bar{\lambda})^2 \leqslant \hat{q}(\lambda_i) - \lambda_{i+1}^2 + 3\delta KT = \hat{v}(\lambda) + 3\delta KT \leqslant \hat{v}(\lambda) + \epsilon,$$

where the first inequality is obtained as in Eq. (6). We have thus found a policy whose performance is within ϵ of the computed approximately optimal performance $\hat{v}(\lambda)$.

Similar policy constructions are possible in the other cases considered in this paper (as, for example, in the next section). Given that these constructions do not involve any new ideas, we will not repeat them.

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

$$g'_t(\delta n|s,a) = \sum_{r:\ \delta n\leqslant r<\delta(n+1)} g_t(r|s,a).$$

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. A conceptually simple but somewhat tedious approach involves an argument along the lines of Whitt (1978), Whitt (1979), generalized to the case of constrained MDPs; we outline such an argument in Section 6.3. Here, we follow an alternative approach, 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*

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

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

Proof. We denote by $d(r)$ the discretized value of a reward r, that is, $d(r) = \max\{n\delta : n\delta \leqslant r,\ n \in \mathbb{Z}\}$. Let us consider a third MDP $\mathcal{M}''$ which is identical to $\mathcal{M}'$, except that its rewards R''_t are generated as follows. (We follow the convention of using a single or double prime to indicate variables associated with $\mathcal{M}'$ or $\mathcal{M}''$, respectively.) A random variable R_t is generated according to the distribution prescribed by $g_t(r|s_t, a_t)$, and its value is observed by the decision maker, who then incurs the reward $R''_t = d(R_t)$. Let P''_{MQ} be the polyhedron associated with $\mathcal{M}''$. We claim that $P''_{MQ} = P'_{MQ}$. The only difference between $\mathcal{M}'$ and $\mathcal{M}''$ is that the decision maker in $\mathcal{M}''$ has access to the additional information $R_t - d(R_t)$. However, this information is incosequential: it does not affect the future transition probabilities or reward distributions. Thus, $R_t - d(R_t)$ can only be useful as an additional randomization variable. Since P'_{MQ} is the set of achievable pairs using general (history-based randomized) policies, having available an additional randomization variable does not change the polyhedron, and $P''_{MQ} = P'_{MQ}$. Thus, to complete the proof it suffices to show that the polyhedra P_{MQ} and P''_{MQ} are close.

Let us compare the MDPs $\mathcal{M}$ and $\mathcal{M}''$. The information available to the decision maker is the same for these two MDPs (since all the history of reward truncations $\{R_\tau - d(R_\tau)\}_{\tau=1}^{t-1}$ is available in $\mathcal{M}''$ for the decision at time t). Therefore, for every policy in one MDP, there exists a policy for the other under which (if we define the two MDPs on a common probability space, involving common random generators) the exact same sequence of states $(S_t = S''_t)$, actions $(A_t = A''_t)$, and random variables R_t is realized. The only difference is that the rewards are R_t and $d(R_t)$, in $\mathcal{M}$ and $\mathcal{M}''$, respectively. Recall that $0 \leqslant R_t - d(R_t) \leqslant \delta$. We obtain that for every policy $\pi \in \Pi$, there exists a policy $\pi'' \in \Pi''$ for which $0 \leqslant W_T - W''_T = \sum_{\tau=0}^{T-1}(R_t - d(R_t)) \leqslant \delta T$, and therefore, $\left|W_T^2 - (W''_T)^2\right| \leqslant 2KT^2\delta$. Taking expectations, we obtain $|J_\pi - J''_\pi| \leqslant T\delta$, $\left|Q_\pi - Q''_\pi\right| \leqslant 2KT^2\delta$. This completes the proof of part (a). The proof of part (b) is identical. □

Theorem 10. *There exists an algorithm that, given ϵ, v, 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 $1/v$.*

Proof. Assume for simplicity that $v = \epsilon$. 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 $v^*(\lambda)$. □

6.3. An exact algorithm and its approximation

There are two general approaches for constructing approximation algorithms. (i) One can discretize the problem, to obtain an easier one, and then apply an algorithm specially tailored to the discretized problem; this was the approach in the preceding subsection. (ii) One can design an exact (but inefficient) algorithm for the original problem and then implement the algorithm approximately. This approach will work provided the approximations do not build up excessively in the course of the algorithm. In this subsection, we elaborate on the latter approach.

We defined earlier the polyhedron P_{MQ} as the set of achievable first and second moments of the cumulative reward starting at time zero at the initial state. We extend this definition by considering intermediate times and arbitrary (intermediate) augmented states. We let

$$C_t(s,w) = \{(\lambda, q) : \exists \pi \in \Pi_{h,u} \text{ s.t. } \mathbb{E}_\pi[W_T|S_t = s,\ W_t = w] = \lambda \text{ and } \mathbb{E}_\pi[W_T^2|S_t = s,\ W_t = w) = q\}. \tag{7}$$

Clearly, $C_0(s,0) = P_{MQ}$. Using a straightforward backwards induction, it can be shown that $C_t(\cdot,\cdot)$ satisfies the set-valued dynamic programming recursion[2]

$$C_t(s,w) = \text{conv}_{a\in\mathcal{A}}\left\{\sum_{s'\in\mathcal{S}} p_t(s'|s,a)\sum_{r\in\mathcal{R}} g_t(r|s,a)C_{t+1}(s', w+r)\right\}, \tag{8}$$

for every $s \in \mathcal{S}, w \in \mathbb{R}$, and for $t = 0,1,2,\ldots,T-1$, initialized with the boundary conditions

$$C_T(s,w) = \{(w, w^2)\}. \tag{9}$$

A simple inductive proof shows that the sets $C_t(s,w)$ are polyhedra; this is because $C_T(s,w)$ is either empty or a singleton and because the sum or convex hull of finitely many polyhedra is a polyhedron. Thus, the recursion involves a finite amount of computation, e.g., by representing each polyhedron in terms of its finitely many extreme points. In the worst case, this translates into an exponential time algorithm, because of the possibly large number of extreme points. However, such an algorithm can also be implemented approximately. If we allow for the introduction of an $O(\epsilon/T)$ error at each stage (where error is measured in terms of the Hausdorf distance), we can work with approximating polyhedra that involve only $O(1/\epsilon)$ extreme points, while ending up with a $O(\epsilon)$ total error; this is because we are approximating polyhedra in the plane, as opposed to higher dimensions where the dependence on ϵ would have been worse dependence. The details are straightforward but somewhat tedious and are omitted. On the other hand, in practice, this approach is likely to be faster than the algorithm of the preceding subsection.

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.

Two more problems of some interest deal with finding a policy that has the smallest possible, or the largest possible variance. There is not much we can say here, except for the following:

(a) The smallest possible variance is attained by a deterministic policy, that is,

$$\min_{\pi\in\Pi_{h,u}} V_\pi = \min_{\pi\in\Pi_h} V_\pi.$$

This is proved using the inequality $\text{var}_\pi(W_T) \geqslant \mathbb{E}_\pi[\text{var}_\pi(W_T|U_{0:T})]$.

(b) Variance will be maximized, in general, by a randomized policy. To see this, consider a single stage problem and two actions with deterministic rewards, equal to 0 and 1, respectively. Variance is maximized by assigning probability 1/2 to each of the actions. The variance maximization

[2] If X and Y are subsets of a vector space and α a scalar, we let $\alpha X = \{\alpha x | x \in X\}$ and $X + Y = \{x + y | x \in X,\ y \in Y\}$. Furthermore, if for every $a \in \mathcal{A}$, we have a set X_a, then $\text{conv}_{a\in\mathcal{A}}\{X_a\}$ is the convex hull of the union of these sets.

problem is equivalent to maximizing the concave function $q - \lambda^2$ subject to $(\lambda, q) \in P_{MQ}$. This is a quadratic programming problem over the polyhedron P_{MQ} and therefore admits a pseudopolynomial time algorithm, when the rewards are integer.

Our results suggest several interesting directions for future research, which we briefly outline below.

First, 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.

Second, a number of complexity questions have been left open. We list a few of them:

(a) Is there a pseudopolynomial time algorithm for computing $v^*(\lambda)$ or $\lambda^*(v)$ exactly?
(b) Is there a polynomial or pseudopolynomial time algorithm that computes $v^*(\lambda)$ or $\lambda^*(v)$ within a uniform error bound ϵ?
(c) Is the problem of computing $\hat{v}(\lambda)$ with the properties in Eq. (5) NP-hard?
(d) Is there a pseudopolynomial time algorithm the smallest possible variance in the absence of any constraints on the mean cumulative reward?

Third, bias-variance tradeoffs may pay an important role in speeding up certain control and learning heuristics, such as those involving control variates (Meyn, 2008). Perhaps mean–variance optimization can be used to address the exploration/exploitation tradeoff in model-based reinforcement learning, with variance reduction serving as a means to reduce the exploration time (see Sutton & Barto (1998) for a general discussion of exploration–exploitation in reinforcement learning). Of course, in light of the computational complexity of bias-variance tradeoffs, incorporating bias-variance tradeoffs in learning makes sense only if experimentation is nearly prohibitive and computation time is cheap. Such an approach could be particularly useful if a coarse, low-complexity, approximate solution of a bias-variance tradeoff problem can result in significant exploration speedup.

Fourth, 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 and 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.

Finally, it is reasonable to expect that our positive results (on approximation algorithms) can be extended to problems involving continuous states and actions and/or unbounded rewards, by first discretizing the problem, truncating the rewards, and then applying our algorithms to a discrete problem. Of course, one would have to deal with the generic issues that arise in discretizing MDPs Whitt (1978), Whitt (1979); we expect this line of work to be tedious without offering any substantial new insights, and have refrained from pursuing it in this paper.

Acknowledgments

The authors are grateful to the reviewers for their constructive comments. This research was partially supported by the Israel Science Foundation (contract 890015), a Horev Fellowship, and the National Science Foundation under grant CMMI-0856063. A preliminary version of this paper appeared at the 28th International Conference on Machine Learning.

References

Altman, E. (1999). *Constrained Markov decision processes*. Chapman and Hall.
Arlotto, A., Gans, N., & Steel, M. J. (2013). *Markov decision problems where means bound variances (Tech. Rep)*. <https://faculty.fuqua.duke.edu/~aa249/ArlottoGansSteele-MDPsWhereMeansBoundVariances.pdf>.
Artzner, P., Delbaen, F., Eber, J., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance, 9*(3), 203–228.
Bertsekas, D. (1995). *Dynamic programming and optimal control*. Athena Scientific.
Chung, K., & Sobel, M. (1987). Discounted MDP's: Distribution functions and exponential utility maximization. *SIAM Journal on Control and Optimization, 25*(1), 49–62.
Collins, E. J. (1997). Finite-horizon variance penalised Markov decision processes. *Operations-Research-Spektrum, 19*, 35–39.
Filar, J., Kallenberg, L. C. M., & Lee, H. M. (1989). Variance-penalised Markov decision processes. *Mathematics of Operations Research, 14*, 147–161.
Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: W.H. Freeman.
Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological), 41*(2), 148–177.
Guo, X., Ye, L., & Yin, G. (2012). A mean-variance optimization problem for discounted Markov decision processes. *European Journal of Operational Research, 220*, 423–429.
Huang, Y., & Kallenberg, L. C. M. (1994). On finding optimal policies for Markov decision chains: A unifying framework for mean-variance tradeoffs. *Mathematics of Operations Research, 19*, 434–448.
Iyengar, G. (2005). Robust dynamic programming. *Mathematics of Operations Research, 30*, 257–280.
Kawai, H. (1987). A variance minimisation problem for a Markov decision process. *European Journal of Operations Research, 31*, 140–145.
Le Tallec, Y. (2007). *Robust, risk-sensitive, and data-driven control of Markov decision processes*. Unpublished doctoral dissertation, Operations Research Center, MIT, Cambridge, MA.
Liu, Y., & Koenig, S. (2005). Risk-sensitive planning with one-switch utility functions: Value iteration. In *Proceedings of the twentieth AAAI conference on artificial intelligence* (p. 993–999).
Liu, Y., & Koenig, S. (2006). Functional value iteration for decision-theoretic planning with general utility functions. In *Proceedings of the twenty first AAAI conference on artificial intelligence* (p. 1186–1193).
Luenberger, D. (1997). *Investment science*. Oxford University Press.
Meyn, S. P. (2008). *Control techniques for complex networks*. New York NY: Cambridge University Press.
Nilim, A., & El Ghaoui, L. (2005). Robust Markov decision processes with uncertain transition matrices. *Operations Research, 53*(5), 780–798.
Papadimitriou, C.H., & Yannakakis, M. (2000). On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st symposium on foundations of computer science* (p. 86–92). Washington, DC, USA.
Riedel, F. (2004). Dynamic coherent risk measures. *Stochastic Processes and their Applications, 112*, 185–200.
Shapley, L. (1953). Stochastic games. *Proceedings of the National Academy of Sciences, Mathematics*, 1095–1100.
Sobel, M. (1982). The variance of discounted Markov decision processes. *Journal of Applied Probability, 19*, 794–802.
Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.
Tamar, A., Di-Castro, D., & Mannor, S. (2012). Policy gradients with variance related risk criteria. In *International conference on machine learning*.
White, D. J. (1992). Computational approaches to variance-penalised Markov decision processes. *Operations-Research-Spektrum, 14*, 79–83.
Whitt, W. (1978). Approximation of dynamic programs – I. *Mathematics of Operations Research, 3*, 231–243.
Whitt, W. (1979). Approximation of dynamic programs – II. *Mathematics of Operations Research, 4*, 179–185.
Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability, 25*, 287–298.