A Rigorous Risk-aware Linear Approach to Extended Markov Ratio Decision Processes with Embedded Learning

Alexander Zadorojniy¹, Takayuki Osogami², Orit Davidovich¹
¹IBM Research - Israel
²IBM Research - Tokyo
zalex@il.ibm.com, osogami@jp.ibm.com, orit.davidovich@ibm.com

Abstract
We consider the problem of risk-aware Markov Decision Processes (MDPs) for Safe AI. We introduce a theoretical framework, Extended Markov Ratio Decision Processes (EMRDP), that incorporates risk into MDPs and embeds environment learning into this framework. We propose an algorithm to find the optimal policy for EMRDP with theoretical guarantees. Under a certain monotonicity assumption, this algorithm runs in strongly-polynomial time both in the discounted and expected average reward models. We validate our algorithm empirically on a Grid World benchmark, evaluating its solution quality, required number of steps, and numerical stability. We find its solution quality to be stable under data noise, while its required number of steps grows with added noise. We observe its numerical stability compared to global methods.

1 Introduction
A Markov Decision Process (MDP) is a fundamental model for sequential decision making. MDPs have been studied extensively [Puterman, 1994], including efficient algorithms for finding optimal policies and MDP extensions to allow for more flexible objectives and constraints. One can incorporate risk into the MDP framework to address issues of Safe AI.

Meggendorfer, 2022 minimizes Conditional Value-at-Risk (CVaR), where the algorithm overall runtime is exponential. [Chow et al., 2015] minimize CVaR with approximation, and their finite-time convergence error bound cannot be made zero with that approximation. [Borkar and Jain, 2014] maximizes the expected return such that CVaR is below a threshold. Their algorithm is inapplicable for infinite horizon problems, which we consider in this paper. Moreover, even for finite horizon problems, they require the separability of the cost function as well as some additional function approximations. We observe that these works lack efficient algorithms, in particular, strongly-polynomial ones.¹

Optimal policy algorithms were found in strongly-polynomial time in the size of the MDP in the discounted cost model and, in some cases, in the expected average cost model. Since MDP extensions are of particular interest in real-world applications, a key question is whether strongly-polynomial algorithms exist for those as well.

An important MDP extension is the Markov Ratio Decision Process (MRDP) [Derman, 1962], where one optimizes for the (non-linear) ratio of two linear cost functions, e.g. reward over risk. MRDP can be reduced to a linear programming (LP) problem in the undiscounted [Derman, 1962] and discounted [Aggarwal et al., 1977] cost cases. Since LP can be solved in polynomial time [Khachiyan, 1979], so does MRDP. A strongly-polynomial algorithm for MRDP, however, is an open problem for all cost models; a strongly-polynomial algorithm is not known for general LPs [Schrijver, 1998], and the LP formulation of MRDP differs from that of MDP [de Gellinck, 1960; d’Epenoux, 1963].

1.1 Contribution
We generalize MRDPs to Extended Markov Ratio Decision Processes (EMRDPs) whose objective \(\tau(\pi)/\sigma^2(\pi)\) for policy \(\pi\) consists of a linear cost function \(\tau(\pi)\) in the numerator and an \(\omega\)-exponentiated linear cost function \(\sigma(\pi)\) in the denominator, where \(\omega \in [0,1]\). EMRDPs reduce to MDPs when \(\omega = 0\) and to MRDPs when \(\omega = 1\). We establish a strongly-polynomial algorithm for optimal policy in these general settings, under a certain monotonicity assumption (that applies, for instance, to applications in financial markets). We provide a data-driven variant of our algorithm that incorporates off-policy evaluation, utilizing available historical data rather than knowledge of an exact environment model. Its solution quality is found to be stable under data noising, while its required number of steps grows with added noise. We analyze its numerical stability compared to global methods.

1.2 Related Work
Designing strongly-polynomial time MDP algorithms with respect to its environment parameters (the number \(n\) of states, the number \(k\) of actions, and the discount factor \(\beta\)) has been a topic of interest for decades. For discounted cost MDPs with a fixed discount factor (a critical assumption), [Ye, 2005] showed that the interior point method is strongly polynomial, and [Ye, 2011] showed that the Simplex algorithm with Dantzig’s pivoting rule is strongly polynomial. [Hansen et al., 2013] showed that the strategy iteration algorithm

¹A strongly-polynomial algorithm runs in time polynomial with the problem size, independent of numerical input size.
A Markov Decision Process (MDP) is a tuple \((S, A, P, r)\) that consists of a finite set of states \(S\), a finite set of actions \(A\), a transition probability matrix \(P\) of size \(n \times n\), where \(n = |S|\) and \(k = |A|\), and an immediate reward vector \(r \in \mathbb{R}^n\).

Let \(S_t\) be the random variable representing the state of the process at time \(t\) and \(A_t\) be the random variable representing the action at \(t\). The time horizon is infinite. We define \(P = \{P(a_1), \ldots, P(a_k)\}\) to be a concatenation of \(k\) square matrices, \(P(a)\) for \(a \in A\), of size \(n \times n\), where the entries of \(P(a)\) represent \(P(s'|s, a)\), \(\forall s, s' \in S\) for an arbitrary \(t\). Let \(r(s, a)\) be the immediate reward for taking action \(a\) at state \(s\).

Throughout this paper, we consider stationary policies \(\pi : S \rightarrow \Delta_A\); for each state \(s\), \(\pi(s, a) := \pi(s)(a)\) denotes the probability of taking action \(a\) at state \(s\).

We consider two reward models: discounted and expected average. We use the occupation measure to represent these reward models in a unified manner. Specifically, for a policy \(\pi\), let \(\rho^{\pi}(s, a) = (1 - \beta) \sum_{t=0}^{\infty} \beta^t P^\pi(S_t = s, A_t = a)\) be the occupation measure for the discounted reward model with a discount factor \(\beta \in (0, 1)\), and let \(\rho^{\pi}(s, a) = \lim_{T \to \infty} \frac{1}{T} \sum_{t \leq T} \mathbb{E}^\pi(S_t = s, A_t = a)\) for the expected average reward model. Then, the expected reward is given as

\[\tau(\pi) := \mathbb{E}^\pi[r(S, A)] = r^T \rho^{\pi},\]

where \(\mathbb{E}^\pi\) denotes the expectation with respect to the corresponding occupation measure induced by \(\pi\).

Constrained MDPs (CMDPs) [Altman, 1999] incorporate constraints. A CMDP is a tuple \((S, A, P, r, d)\), where \(d \in \mathbb{R}^{n \times k}\) is an immediate risk vector; \(d(s, a)\) represents the immediate risk for taking action \(a\) at state \(s\). Analogously to reward, we consider two risk models (discounted and expected average risk models), with expected risk given by

\[\sigma(\pi) := \mathbb{E}^\pi[d(S, A)] = d^T \rho^{\pi}.\]

The objective of the CMDP is to maximize the expected reward under the constraint that the expected risk is at a target value \(\alpha\), which gives rise to the parameterized LP

\[\text{LP}(\alpha) : \max \{\tau(\pi) | Q \rho^{\pi} = \mu, \sigma(\pi) = \alpha, \rho^{\pi} \geq 0\}.\]

The matrix \(Q\) and vector \(\mu, \sigma\) are defined according to the model of choice. For the discounted reward/risk model, \(Q = Q(\beta) = [I - \beta P(a_1), \ldots, I - \beta P(a_k)]\), where \(I\) is the \(n \times n\) identity matrix, and \(\mu \in \mathbb{R}^n\) is the initial state distribution multiplied by \((1 - \beta)^2\). Notice, that the first equality constraint in (1) ensures that \(\rho^{\pi}\) (which induces policy \(\pi\)) is a valid occupation measure. For the average reward/risk model, \(Q\) is the \((n + 1) \times nk\) matrix defined by appending an additional row of ones to \(Q(1)\), and \(\mu = e_{n+1} \in \mathbb{R}^{n+1}\) is the standard basis vector with a single 1 at the last entry. We refer to the \(\alpha\)-parameterized CMDP in (1) as CMDP(\(\alpha\)). The variable of LP(\(\alpha\)) is \(\rho^{\pi}\), which yields its corresponding policy \(\pi\) via a standard procedure. We thus write \(\max \pi\) when the maximization is over \(\rho^{\pi}\).

### 3 Extended Markov Ratio Decision Process

We define the Extended Markov Ratio Decision Process (EMRDP) using the same five-tuple \((S, A, P, r, d)\) as the CMDP. Unlike the CMDP, however, the EMRDP does not include constraints. Its objective is to maximize the ratio of the expected reward to the expected risk to the power of some \(\omega \in [0, 1]\). The problem of finding the optimal policy for the EMRDP can thus be formulated as:

\[
\max_{\pi} \left\{ \frac{\tau(\pi)}{\sigma^\omega(\pi)} \mid Q \rho^{\pi} = \mu, \rho^{\pi} \geq 0 \right\}
\]

(2)

An EMRDP reduces to a MDP [Puterman, 1994] when \(\omega = 0\) and to a MRDP [Derman, 1962; Aggarwal et al., 1977] when \(\omega = 1\). It thus interpolates between MDP and MRDP. We will focus on \(\omega \in (0, 1]\) for the rest of this paper, since it is there that the question of strong polynomiality is still unresolved.

### 4 A General Algorithm for EMRDP

In this section we present a general algorithm for EMRDP. Though we cannot prove that the running time of this algorithm will be polynomial in general, we provide sufficient conditions to make it run in strongly-polynomial time.

#### 4.1 Algorithm Description

We solve (1) by finding the set of optimal deterministic policies for all feasible \(\alpha\) and then prove it is sufficient to consider just this set of policies to solve (2).
Consider a deterministic policy $\pi$. Let $\pi^{s,a}$ be the deterministic policy that follows $\pi$, except at state $s$ where action $a$ is taken instead of $\pi(s)$. Denote the change in expected reward over the change in expected risk by

$$\nabla_{s,a}(\pi) \equiv \frac{\tau(\pi^{s,a}) - \tau(\pi)}{\delta(\pi^{s,a}) - \delta(\pi)} = \frac{r^\top \rho^{\pi^{s,a}} - r^\top \rho^\pi}{d^\top \rho^{\pi^{s,a}} - d^\top \rho^\pi}. \quad (3)$$

In Line 1 of Algorithm 1, we initialize $\pi$ with a feasible policy that minimizes $\delta(\pi)$ by solving the MDP

$$\pi = \arg\min_{\pi} \{ \delta(\pi) \mid Q \rho^\pi = \mu, \rho^\pi \geq 0 \}. \quad (4)$$

The constraints in (4) ensure the feasibility of $\pi$. In Line 2, this minimal policy, together with its corresponding reward and risk, is added to a set $T$. At each while-loop iteration, the algorithm considers policies $\pi^{s,a}$ with $\delta(\pi^{s,a}) > \delta(\pi)$. In Line 4, $\pi^{s,a}$ is chosen such that $\nabla_{s,a}(\pi)$ is maximal. In Line 5, $\pi^{s,a}$ (with its corresponding reward and risk) is added to $T$. In Line 6, the policy $\pi$ is updated with $\pi^{s,a}$ for the next iteration of the while loop. In Line 8, the policy that corresponds to the maximal return is returned.

4.2 Correctness

We prove that Algorithm 1 finds the optimal policy for problem (2) under the standard assumptions in the literature. Let $G(V, E)$ be the policy graph, where $V$ is the set of deterministic policies and $E$ is the set of the pairs of neighbouring deterministic policies (i.e., policies that disagree in exactly one state). A policy $\pi$ is strictly 1-randomized if it is deterministic in all but one state $s$ and the set $\{a : \pi(s, a) > 0\}$ contains exactly two actions. Let $\Gamma_0$ be the set of deterministic and strictly 1-randomized policies for CMDP($\alpha$). Let $\Gamma_0 \subseteq \Gamma_0$ be the set of optimal policies in $\Gamma_0$, and let $\Gamma^* = \bigcup \Gamma_0$.

As in [Zadorojniy et al., 2009], we assume uniqueness, which in practice can be achieved by perturbation [Megiddo and Chandrasekaran, 1989; Zadorojniy et al., 2009].

**Assumption 1 (Uniqueness).** For every $\alpha$, CMDP($\alpha$) satisfies the following property: if $\pi^*$ is deterministic and optimal for CMDP($\alpha$), any other deterministic or strictly 1-randomized policy is not optimal for LP($\alpha$), meaning that $\tau(\pi) < \tau(\pi^*)$, $\forall \pi \in \{ \pi' \mid \pi' \in \Gamma_0, \pi' \neq \pi^* \}$.

Another standard assumption is irreducibility (also following [Zadorojniy et al., 2009]). We say that a CMDP is irreducible if every deterministic policy of the CMDP induces an irreducible Markov chain. In practice, this can be achieved by connecting an arbitrary state, normally the initial one, with all others under all actions with “small” transition probabilities by bi-directional arcs.

**Assumption 2 (Irreducibility).** CMDP($\alpha$) is irreducible.

The next Proposition which requires just Assumption 2 shows that considering only deterministic and 1-randomized policies is sufficient for finding the optimal policy.

**Proposition 1** (Theorem 5.1 [Zadorojniy et al., 2009]). If LP($\alpha$) is feasible, then there exists an optimal policy $\pi^*$ of LP($\alpha$) that is deterministic or strictly 1-randomized.

The intuition behind following two Propositions is a s following, we start from the vertex corresponding to a deterministic minimal-risk policy. Then, we gradually increase risk level, switching to the strictly 1-randomized policy. By continuing to increase the risk value further, we reach the next reward-optimal deterministic policy. This procedure is repeated until the stopping condition is satisfied.

The following propositions hold under Assumptions 1-2.

**Proposition 2** (Lemma 5.4 [Zadorojniy et al., 2009]). The set $\Gamma^*$ forms a path in the policy graph $G$.

**Proposition 3** (Theorem 6.1 [Zadorojniy et al., 2009]). Let $\pi^i$ be the $(i + 1)$-st policy added to $T$ by Algorithm 1 for $i = 0, 1, \ldots$. Then, for any index $i$, there is an edge in $G$ between $\pi^i$ and $\pi^{i+1}$, and the set of all the policies on those edges, including their endpoints, equals $\Gamma^*$.

Without loss of generality, let $\pi^0$ and $\pi^1$ be any two neighbouring deterministic policies on $\Gamma^*$, and $\pi^0$ be a strictly 1-randomized policy on the edge between $\pi^0$ and $\pi^1$ (i.e., $\pi^0 = (1 - q) \pi^0 + q \pi^1$, where $q \in (0, 1)$). Then the expected reward and the expected risk have a linear relation:

**Proposition 4** (Proposition 5.5 [Zadorojniy et al., 2009]). If $\delta(\pi^0) \neq \delta(\pi^1)$, then $\tau(\pi^q)$ is linear in $\delta(\pi^q)$ over the range $q \in [0, 1]$.

Figure 1 illustrates the relation between $\tau$ and $\delta$. Policies $\pi^0$, $\pi^1$, $\pi^2$ and $\pi^3$ are deterministic and optimal, and policy $\pi^q$ is strictly 1-randomized and optimal. Policies $\pi^1$ and $\pi^2$ are suboptimal and satisfy $\delta(\pi^1) = \delta(\pi^2) = \delta(\pi^3) = \alpha$ for
an $\alpha \in \mathbb{R}^+$. In general, there are multiple policies $\pi$ with
$\bar{d}(\pi) = \alpha$, and these policies can have different values of $v(\pi)$. However, if we consider just the $v(\pi^*)$ corresponding
to the optimal policy $\pi^*$ for each $\alpha$, we get a piecewise linear function from $\bar{d}(\pi^*)$ to $v(\pi^*)$ due to Proposition 4. Formally,

**Corollary 1.** Let $\Gamma^{\text{det}}$ be the set of deterministic policies
returned by Algorithm 1 and $\Gamma^{\text{rand}}$ be the set of strictly $I$-
randomized policies on the edges between the neighboring policies in $\Gamma^{\text{det}}$. Then $\Gamma^{\text{det}} \cup \Gamma^{\text{rand}} = \Gamma^*$, and the relation
$\bar{d}(\pi^*) \rightarrow v(\pi^*)$ for $\pi \in \Gamma^*$ forms a piecewise linear function.

**Proof.** Proposition 3 implies $\Gamma^{\text{det}} \cup \Gamma^{\text{rand}} = \Gamma^*$. Proposition 2 and Proposition 4 imply that the relation in the corollary is a
piecewise linear function.

We further assume that any policy yields positive expected reward and positive expected risk. More formally,

**Assumption 3.** (Positivity) For any policy $\pi$, we have
$\bar{d}(\pi), v(\pi) \in \mathbb{R}^+$. The following Lemma holds under Assumption 3:

**Lemma 1.** The relation $\bar{d}(\pi) \rightarrow \frac{v(\pi)}{\bar{d}(\pi)}$ for $\pi \in \Gamma^*$ is a con-
tinuous function, and $\frac{v(\pi^*)}{\bar{d}(\pi^*)} \geq \frac{v(\pi)}{\bar{d}(\pi)}$ for any $\pi^* \in \Gamma^*$ and
any $\pi$ such that $\bar{d}(\pi) = \bar{d}(\pi^*)$.

**Proof.** Corollary 1 implies that the relation in the lemma is a continuous function under Assumption 3. Also, for any $\pi^* \in \Gamma^*$ and any $\pi$ such that $\bar{d}(\pi) = \bar{d}(\pi^*)$, we have $v(\pi^*) \geq v(\pi)$ and hence $\frac{v(\pi^*)}{\bar{d}(\pi^*)} \geq \frac{v(\pi)}{\bar{d}(\pi)}$ for $\omega \in (0, 1)$.

Algorithm 1 considers only deterministic optimal policies, ignoring strictly $I$-randomized policies that are optimal for
some $\alpha$. We will now show that such strictly $I$-randomized
policies can indeed be ignored.

**Lemma 2.** Let $0 < \alpha_{\min} < \alpha_{\max} < +\infty$ and $f(\alpha) = \frac{c\alpha + b}{a}$
for $\alpha \in [\alpha_{\min}, \alpha_{\max}]$, where $\omega \in (0, 1)$ and $c, b$ are constants
such that $c\alpha + b > 0$ for any $\alpha \in [\alpha_{\min}, \alpha_{\max}]$. Then $f(\alpha)$ is
quasiconvex (i.e., its maximum is attained either at $\alpha = \alpha_{\min}$
or $\alpha = \alpha_{\max}$).

The following theorem shows that we can solve the EM-
RDP problem with Algorithm 1:

**Theorem 1.** Under Assumptions (1-3), Algorithm 1 returns
the $\pi$ that maximizes $v(\pi)/\bar{d}(\pi)$ over all policies $\pi$.

**Proof.** Let $f^* : \bar{d}(\pi) \rightarrow \frac{v(\pi)}{\bar{d}(\pi)}$ for $\pi \in \Gamma^*$ be the function
as defined in Lemma 1. Then, following Lemmas 1-2, $f^*$
is a piecewise quasiconvex function as depicted in Figure 2
($\alpha = \bar{d}(\pi)$). Hence, $f^*$ has no local maximas, corresponding
to strictly $I$-randomized policies, meaning that Algorithm 1 finds a set of deterministic policies such that one of them must
be globally optimal. Thus, by choosing the $\pi$ with maximal
ratio $v(\pi)/\bar{d}(\pi)$, Algorithm 1 returns the globally optimal policy for the EM-
RDP problem.

**5 A Strongly Polynomial Algorithm**

We now show that Algorithm 1 can run in strongly polynomial
time with an additional assumption that often holds in
applications to financial markets. In financial applications,
“larger” actions, such as investing a larger amount, are fundamen-
tally riskier than “smaller” ones. We may thus assume that
the risk increases monotonically with the amount of invest-
ment, when actions determine the amount of investment.

**Assumption 4** (Monotonicity). There is a linear order
$(a_1, \ldots, a_k)$ on $A$ such that
$$\bar{d}(\pi^{a_1}) < \ldots < \bar{d}(\pi^{a_k}), \forall \pi, a.$$  (5)

**5.1 Algorithm Description**

Algorithm 2 shows the version of our algorithm that runs in
strongly-polynomial time when monotonicity holds. Algorithm 2 is identical to Algorithm 1 except in Line 1, where
we initialize $\pi$ with the policy that takes the smallest action
$a_1$, defined in Assumption 4, at every state.

**5.2 Correctness and Running Time**

When Assumption 4 holds, this initial policy minimizes the
risk. Algorithm 2 thus finds the deterministic policy $\pi$ that

**Algorithm 2 A Strongly Polynomial Algorithm for EMRDP**

Problem (2) when Monotonicity Assumption Holds.

1: Initialize: $\pi \leftarrow (a_1, \ldots, a_1)$
2: $T \leftarrow \{(\pi, v(\pi), \bar{d}(\pi))\}$
3: while $\exists (s, a)$ s.t. $\bar{d}(\pi^{s,a}) > \bar{d}(\pi)$ do
4: $\pi^{s,a} \leftarrow \arg\max \{v(\pi^{s,a}) | \bar{d}(\pi^{s,a}) > \bar{d}(\pi)\}$
5: Add $(\pi^{s,a}, v(\pi^{s,a}), \bar{d}(\pi^{s,a}))$ into $T$
6: $\pi \leftarrow \pi^{s,a}$
7: end while
8: return $\arg\max \{v(\pi)/\bar{d}(\pi) | (\pi, v(\pi), \bar{d}(\pi)) \in T\}$
minimizes \( \tau(\pi) \) without solving the MDP in (4), and this solves the issue of strong polynomiality in the case of the expected reward model. In this sense, Algorithm 2 is identical to Algorithm 1 and finds the optimal solution for EMRD. What remains to show is the running time of Algorithm 2.

**Lemma 3.** Algorithm 2 runs in \( O(n^4 k^2) \) time when Assumption 4 is satisfied.

**Proof.** By Assumption 4, the while loop of Algorithm 2 is repeated only \( O(nk) \) times, since the action must increase in Line 4. In Line 4, there are at most \( nk \) candidates of the next policy, and the evaluation of each candidate is dominated by the inversion (i.e., inverse of the basis matrix of size \( n \times n \)) of the matrix which differs from the current vertex matrix by one column. By the Sherman-Morrison formula [Meyer, 2000], such matrix inversion can be done in \( O(n^3) \) time, which completes the proof of the lemma.

The results in [Even and Zadorojniy, 2012] suggest that Line 1-7 of Algorithm 2 is equivalent to the Simplex algorithm with the Gass-Saaty shadow-vertex pivoting rule, where Line 4 of Algorithm 2 is the pivoting rule of the next vertex. This suggests that the running time of Algorithm 2 is actually \( O(n^3 k^2) \) by the following proposition:

**Proposition 5** (Proposition 4 from [Even and Zadorojniy, 2012]). The running time of the Simplex algorithm with the Gass-Saaty shadow-vertex pivoting rule is \( O(n^3 k^2) \) for finding an optimal policy of a controlled random walk.

Notice that there are minor adjustments with respect to the Simplex algorithm related to the ratio maximization in Line 8 of Algorithm 2.

### 6 Embedded Learning

We can validate the practical relevance of our algorithmic approach empirically, introducing a variant of Algorithm 1 with embedded learning.

Consider a dataset \( D \) derived from some ground-truth environment, whose columns include state and action features as well as reward and risk outcomes; its rows constitute a sequential log of historical instances of state, action, reward, and risk values. When state or action features are continuous, they can be binned (i.e., discretized), while the immediate reward and risk associated with each bin can be averaged. Binning can thus give rise to an empirical CMDP formulation of \( D \) with finite state and action spaces and corresponding immediate reward and risk vectors. Transition probability matrices can be estimated from \( D \) by counting the number of occurrences of neighboring (binned) rows.

Compared to the discretized ground-truth environment, not all discretized states or actions will necessarily be present in \( D \). Thus, we derive empirical \( \mathcal{S} \) and \( \mathcal{A} \) from \( D \). In addition, the empirical state-action space \( \mathcal{S} \times \mathcal{A} \) may not be present in \( D \) in its entirety. We denote its subset \((\mathcal{S} \times \mathcal{A})_D\) to include all pairs \((s, a)\in \mathcal{S} \times \mathcal{A} \) present in the binned \( D \).

**Empirical algorithm.** Algorithm 3 embeds learning within our EMRD framework. It relies on two essential components to supplement Algorithm 1: (i) a state-dependant ordering of action-associated risk (Line 1) and (ii) a data-driven estimate of expected reward and risk (e.g., Line 4). Addressing (i), we define \((s, a') < (s, a'')\) for a fixed \(s\in \mathcal{S}\) and a pair \((s, a'), (s, a'')\in (\mathcal{S} \times \mathcal{A})_D\) iff \(d(s, a') < d(s, a'')\). In Line 1, we initialize with the policy that takes the minimal action at \(s_i\), over all \(a_i\) s.t. \((s_i, a_i)\in (\mathcal{S} \times \mathcal{A})_D\). This choice is not guaranteed to be the global minimal-risk policy. However, we will see that it is sufficient in practise. Addressing (ii), we use an offline policy evaluation (OPE) algorithm [Voloshin et al., 2021] that relies on \( D \) to output estimates of expected reward \( \mathcal{E}^\pi(\pi) \) and risk \( \mathcal{D}^\pi(\pi) \) for a policy \(\pi\) supported in \((\mathcal{S} \times \mathcal{A})_D\). In Lines 3 and 4, we utilize OPE to produce estimates \( \mathcal{E}^\pi(\pi) \) and \( \mathcal{D}^\pi(\pi) \) and derive \( \nabla \mathcal{D}^\pi(\pi) \) as in (3). In our experiments, we use Q-Evaluation as our OPE [Kostrikov and Nachum, 2020], which is the standard fitted Q-Evaluation without functional approximation.

#### 7 Experiments

We compared Algorithm 3 empirically to the theoretically optimal policy on the benchmark Grid World problem.\(^3\) We examined its performance in terms of solution quality, number of steps, and numerical stability. We found Algorithm 3 solution quality to be stable under data noising, while its required number of steps grows with added noise. We also observed its numerical stability compared to a global optimization method.

#### 7.1 Environment

Solving the benchmark Grid World problem requires finding the optimal policy for traversing an \( h \times w \) grid from an initial to a terminal cell in a minimal number of moves in the presence of obstacles and action-associated risk. We fixed the initial cell to be the top left cell.

The action space is \( \mathcal{A} = \{ \text{Up}, \text{Down}, \text{Left}, \text{Right}, \text{None} \} \) for this problem, where None refers to a ‘do nothing’ move. Adding the None move allows us to incorporate the notion of infinite horizon. We define transition probability matrices \( P(a), a \in \mathcal{A} \) so that there is a \( \delta = 0.1 \) probability to end up at a random state under each \(a \in \mathcal{A} \setminus \{ \text{None} \} \) move and a

---

\(^3\)Since we are dealing with data-derived risk, we may assume this ordering is strict.

\(^2\)The code for our empirical evaluation is available on https://github.com/IBM/IBM-Extended-Markov-Ratio-Decision-Process.
Figure 3: A theoretically-optimal policy in a $h \times w = 5 \times 5$ grid. The initial cell is marked blue, while the terminal cell is marked red. The obstacle cells are crossed out.

$\delta_N = 0.5$ probability under the $a = \text{None}$ move.\(^5\) The non-zero probability for random moves ensures irreducibility.

We fix the immediate cost associated with each cell to be $\text{cost} = 5.0$ and the immediate negative reward to be $-\text{cost}$, except at the terminal cell, where the cost is 0.0, and at obstacle cells, where the cost is $M = 2/(1 - \beta)$,\(^6\) following [Chow et al., 2015]. Thus, maximizing reward aims at finding the policy that establishes the shortest (stochastic) path to the terminal cell, while avoiding obstacles.

To the standard benchmark Grid World problem setup, introduced thus far, we add the novel notion of risk. We define risk as high for walking along the top or bottom boundaries of the grid and medium for walking left into the left boundary or right into the right boundary. Risk is otherwise set to a baseline low value.\(^7\) Our goal is to minimize the ratio $\tau(\pi)/\beta(\pi)$ (where $\omega = 1$), thereby increasing (negative) reward while decreasing (negative) risk.

We add Gaussian randomization to the immediate reward and risk vectors with $\epsilon = 0.01$ standard deviation.\(^8\) We also sample a pre-fixed number of obstacles. Thus, we effectively generate a Grid World environment at random, with varying immediate reward and risk vectors. Moreover, changing the environment hyper-parameters $-\beta, \delta, \delta_N, \epsilon$, the cost per cell, the low, mid, and high risk values, and the number of obstacles — means that we are, in fact, considering a family of stochastic Grid World environments, rather than a single one (in contrast to [Chow et al., 2015]).

### 7.2 Theoretically-Optimal Policy

We can compare Algorithm 3 empirically to the theoretical optimum with its full Pareto frontier, calculated using the theory of parametric LPs applied to (1) [Walkup and Wets, 1969]. Figure 3 shows a theoretically optimal policy for a randomly generated Grid World environment. Policy optima are sensitive to Grid World hyper-parameters. A small gap between high and low risk parameter values, for example, will affect risk expression non-trivially: when the expected reward of a policy is high, its occupation measure is high at or near the terminal cell and low on boundary cells further away. Then, their effect on the expected risk is negligible. Minute reward fluctuations may then tilt the scale towards walking along the boundary, effectively ignoring risk. This subtle effect is seen in Figure 3 at the bottom left side of the boundary.

#### 7.3 Solution Quality and Number of Steps

We tested Algorithm 3 on the Grid World problem as follows. First, we calculated the theoretically-optimal policy for a randomly generated $h \times w$ Grid World environment. Then, we introduced random noise to the theoretically-optimal policy, at the 10%, 30%, or 100% level. We then generated data from the noised optimal policy, following it for $10 \cdot |S|^2 \cdot 4$ moves. In our Grid World dataset, each row was associated with a single state (grid cell), action (move), immediate reward, and immediate risk. We learned the empirical Grid World environment from the dataset (Section 6) and ran Algorithm 3.

We ran 150 tests overall on a $5 \times 5$ grid with 3 obstacles, each time, randomly sampling the obstacles and the randomized immediate reward and risk vectors. A $5 \times 5$ grid allowed us to witness the non-trivial effects of risk at a relatively low computational cost. The bigger the grid, the lower the likelihood of hitting the boundary, hence, the lower the effect of boundary risk. Thus, with a bigger grid, the problem is likely to degenerate to an MDP. To witness the effect of risk in a bigger grid would have necessitated a more elaborate risk model.

Figures 4a–4b show the distribution of Algorithm 3 optimal ratio over the theoretical optimal ratio across our three noise levels. We provide the mean confidence interval for $p = 0.05$. As can be seen, Algorithm 3 solution quality is stable w.r.t. data noise. Algorithm 3 optimal ratio, on average, is $2.64 \pm 0.30$ (10% noise), $2.64 \pm 0.36$ (30% noise), and $2.63 \pm 0.42$ (100% noise) times the theoretical optimal ratio.

Figures 5a–5b show the distribution across the same three noise levels of the number of steps required by Algorithm 3 over the number of steps to the optimum along the theoretical Pareto frontier. The number of steps required by Algorithm 3, on average, is $1.38 \pm 0.26$ (10% noise), $1.77 \pm 0.19$ (30% noise), and $2.37 \pm 0.25$ (100% noise) times the number of step to the theoretical optimum. The average number of steps to the optimum along the theoretical Pareto frontier was $14.96 \pm 0.56$ ($p = 0.05$) over our 150 Grid World environments.

We can thus confirm empirically that Algorithm 3 requires more optimization steps as noise grows, though solution quality remains stable. We observe that noise does not weaken learning as a whole. However, the initial, empirically-minimal policy, from which Algorithm 3 starts, gets further away from the ground-truth risk-minimizing policy as noise grows. As a result, Algorithm 3 requires more steps compared to the number of steps to the theoretical optimum along the theoretical Pareto frontier, which starts at the ground-truth risk-minimizing policy.
Figure 4: Empirical distribution of Algorithm 3 optimal ratio over the theoretical optimal ratio across three noise levels.

(a) 10% noise, mean 2.64 ± 0.30.
(b) 30% noise, mean 2.64 ± 0.36.
(c) 100% noise, mean 2.63 ± 0.42.

Figure 5: Empirical distributions of the number of steps required by Algorithm 3 over the number of step to the optimum along the theoretical Pareto frontier across three noise levels.

(a) 10% noise, mean 1.38 ± 0.26.
(b) 30% noise, mean 1.77 ± 0.19.
(c) 100% noise, mean 2.37 ± 0.25.

7.4 Numerical Stability

Strongly-polynomial algorithms are often considered more numerically stable [Gill and Murray, 1973; Olver and Végh, 2020; Borgwardt, 2020]. The issue of numerical stability comes up in our Grid World example. Prior to randomizing the immediate risk and reward vectors, its optimization landscape has multiple global optima, since, risk-wise and in the absence of obstacles, moving Right then Down from a (non-boundary) cell is equivalent to moving Down then Right.

When we add randomization to immediate reward and risk, we introduce a unique global optimum with probability 1. Numerically, however, we may still have multiple global optima due to machine precision. Several numerical issues come up in calculating the theoretically-optimal policy. Since we are solving multiple LPs, solutions may not exist or may not give rise to bona fide occupation measures (up to sufficient precision). Our implementation of the theoretically-optimal policy algorithm had a fail rate of 4.7% ± 0.4% ($p = 0.05$), calculated for $n = 25$ tests, sampling 100 non-degenerate Grid World environments for each.

The issue of numerical stability is inherent to any global search for the optimal policy. A gradient ascent-type algorithm, such as Algorithm 3, that “follows its nose” to some critical point does not encounter such numerical issues.

Parallel computing. From a computational standpoint, estimations of expected reward and risk at the next-best-policy step (Line 4) are mutually independent. We can thus parallelize Algorithm 3 easily. We used the cloud distribution framework Ray to scale up Algorithm 3 [Moritz et al., 2018]. We ran our experiments on an OpenShift cluster with 16 CPUs x 64 RAM x 3 workers.

8 Summary

We introduced a novel parameterized control process, EM-RDP, that incorporates both MDP and RMDP. We introduced Algorithm 1 that finds its optimal policy under Assumptions (1)–(3). Its variant, Algorithm 2, does so in strongly-polynomial time under the additional Assumption (4), both for the discounted and expected average reward models. We demonstrated policy optimization with embedded learning in Algorithm 3, using the Grid World problem as a benchmark.
References


