# Optimizing Dec-POMDP Agent-State Policies via Risk-Seeking Utility

**Amit Sinha**[1], **Matthieu Geist**[2], **Aditya Mahajan**[1]
amit.sinha@mail.mcgill.ca, matthieu.geist@gmail.com, aditya.mahajan@mcgill.ca

[1]**Department of ECE, McGill University, Montreal, Quebec**
[2]**Earth Species Project**

## Abstract

Solving decentralized decision-making problems modeled as Dec-POMDPs is notoriously NEXP-complete, as optimal solutions require policies conditioned on an agent's entire action-observation history. To maintain tractability, it is common to restrict agents to finite-memory models, known as agent-state policies. Although this constrained policy class may not contain the globally optimal solution, finding the highest-performing agent-state policy remains a critical objective for practical applications. Addressing the challenge of planning under bounded memory, we introduce an iterated best-response algorithm that converges monotonically to a local optimum in polynomial runtime relative to the Dec-POMDP model size. To discover superior policies within this restricted memory space, we employ a novel objective that pairs a risk-seeking incentive with conservative policy updates. Our experiments on standard Dec-POMDP benchmarks demonstrate that this approach is competitive with state-of-the-art methods, delivering near-optimal results despite the limited memory.

## 1 Introduction

Decentralized Partially Observable Markov Decision Processes (Dec-POMDP) model multi-agent sequential decision-making problems where agents with asymmetric partial observations must coordinate their actions to optimize a shared cumulative reward. Optimally solving Dec-POMDPs presents three fundamental challenges:

***(i) Information decentralization:*** Let $I_t^i$ denote the information available to agent $i$ at time $t$. Typically, in Dec-POMDPs agents lack perfect recall ($I_t^i \not\subset I_{t+1}^i$) and possess asymmetric information ($I_t^i \neq I_t^j$ for $i \neq j$). Thus Dec-POMDPs have a *non-classical information structure* (Witsenhausen, 1971), which precludes the use of standard dynamic programming (DP) due to the absence of a single decision-maker with perfect recall.

***(ii) Partial observations with agent states:*** Agents receive only partial observations of the environment state and maintain a compressed summary of past observations and actions as a local internal state (called agent state). Due to fixed memory of the agent state, agents must perform lossy state updates, selectively discarding non-salient details to prioritize information critical for decision-making.

***(iii) Coordination among agents:*** Agents must coordinate their policies not only to maximize immediate rewards but also to strategically signal information, thereby facilitating the optimization of future joint actions. Because an update in one agent's policy alters the best response of others, the learning task becomes a joint optimization problem. Consequently, agents must coordinate their learning processes within the joint policy space rather than optimizing each policy independently.

The sheer complexity of these challenges is underscored by their theoretical hardness: solving a Dec-POMDP optimally is NEXP-complete (Bernstein et al., 2002), a computational barrier that severely restricts even approximate solution methods (Rabinovich et al., 2003).

In many practical applications, agents are constrained to a finite memory capacity, formally represented as "agent states" (Dong et al., 2022). Under such bounded constraints, the objective naturally shifts from discovering an optimal history-based policy to isolating the highest-performing solution within the restricted class of *agent-state policies*. ***Our main contribution is a scalable algorithm equipped with rigorous theoretical convergence guarantees for optimizing these bounded-memory agent-state policies.***

**Main Idea:** We are interested in obtaining a locally optimal solution in polynomial runtime. Given a cost function $J$ and a time horizon of $T$, a locally optimal joint non-stationary agent-state policy $\pi^\star = (\pi^{1,\star}, \pi^{2,\star}, \cdots, \pi^{N,\star})$ for $N$ agents (where each non-stationary policy $\pi^{i,\star} = (\pi_1^{i,\star}, \cdots, \pi_T^{i,\star})$) is defined as:

$$\pi_t^{i,\star} \in \arg\max_{\pi_t^i \in \Pi_t^i} J(\pi_{1:t-1}^{i,\star}, \pi_t^i, \pi_{t+1:T}^{i,\star}, \pi^{-i,\star}), \quad \forall i \in \{1, \cdots, N\}, \quad \forall t \in \{1, \cdots, T\}. \tag{1}$$

Note that this represents the simultaneous maximization for the $N$ different agents at all possible time indices $t$. We consider an approach that uses iterated best response to obtain a local optimum. Given a starting joint policy of $N$ agents $\pi = (\pi^1, \pi^2, \cdots, \pi^N)$, this involves repeatedly applying the following update sequentially for each agent $i \in \{1, \cdots, N\}$:

$$\bar{\pi}^i \in \arg\max_{\pi^i \in \Pi^i} J(\pi^i, \pi^{-i}), \tag{2}$$

where $\pi^{-i}$ denotes the policy of all agents except $i$. Repeatedly applying such an update (with ties broken with a fixed rule) provably leads to monotonic improvements and convergence to a local optimum. This fundamental idea is similar to the line of work on Joint Equilibrium-based Search for Policies (JESP) (Nair et al., 2003).

In particular, we combine the following ideas that empirically allow convergence to better local optima (higher $J$):

1. **Sequential Equilibrium**: Keeping the policies of all agents fixed except the one we are optimizing for, allows us to write a novel dynamic program in terms of the agent state which enables guaranteed monotonic improvements and convergence.
2. **Risk-seeking objective** (Shen et al., 2013): Using such an objective allows states with higher rewards to be perceived with a larger weight than those with lower rewards. We provide a concrete example illustrating how this can lead to better local optima in Appendix A.
3. **Conservative-policy iteration (CPI)** (Kakade & Langford, 2002): Without conservatism the policies generated using iterated best response are deterministic after the first iteration. Consequently, some parts of the state space have zero or negligible marginal distribution, preventing further improvement regardless of how actions are chosen in those states.

We refer to our overall approach as ***Risk-Seeking Conservative Policy Iteration (RS-CPI)***.

Our main contributions are summarized as follows:

1. We provide a scalable algorithm RS-CPI with guaranteed convergence and monotonic performance improvement.
2. We demonstrate comparable performance of RS-CPI with the state-of-the-art algorithms from the literature which are (approximately) optimal on several standard Dec-POMDPs.
3. We show that the performance of RS-CPI improves as the allowed agent-state size is increased.
4. We also provide ablations that show the adding risk-seeking and conservatism is critical to performance.

**Notation:** Consider a discrete time system that operates from step 1 to step $T$. We use $\overleftarrow{x}_t$ to denote all quantities up to (and excluding) time $t$, i.e., $x_{1:t-1}$. We use $\vec{x}_t$ to denote all quantities after (and

excluding) time $t$, i.e., $x_{t+1:T}$. We use capital letters $X$ to denote random variables and small letters $x$ to denote their realizations. We use sans-serif font $\mathsf{X}$ to denote the corresponding spaces. We use subscripts for the time index $t$ and superscripts for the agent index $i$ as $x_t^i$. We use the notation $x_t$ without any superscript to denote the quantity for all agents, i.e., $x_t \coloneqq (x_t^1, \cdots, x_t^N)$, where $N$ is the number of agents. We also use $x^i$ to denote the quantity over time for agent $i$, i.e., $x^i \coloneqq (x_1^i, \cdots, x_T^i)$. Similarly, we also use $x \coloneqq (x_1, \cdots, x_T)$.

## 2 Related Work

The designer's approach (Mahajan, 2008; Witsenhausen, 1973) can be used to used to obtain a dynamic program to obtain the optimal solution for a general finite-state controller. However, the size of such a dynamic program is exponential in the size of the Dec-POMDP spaces and so, it does not scale well to larger settings. In some Dec-POMDPs, there may be some structure in the form of common information between agents that can be exploited by ideas like the common information approach (Nayyar et al., 2013; Tang et al., 2023), but structural assumption of common information does not always hold. As a result these ideas only work for small scale environments.

Feature-based heuristic search value iteration (FB-HSVI) (Dibangoye et al., 2016) involves the construction of a dynamic program similar to the designer's approach (referred to as the "occupancy MDP") and the exponential complexity can be dealt with by clustering and a sliding window of the history. Koops et al. (2024) also uses follows a similar outline in their Policy-finding multi-agent A$^*$ (PF-MAA$^*$) algorithm. These approaches involve finding near-optimal history-based solutions. One important limitation of these works is that they are only able to use a small window size, because otherwise the exponential growth of trajectories becomes intractable to deal with. A more detailed literature review is given in Appendix D due to space limitations.

## 3 Background

### 3.1 Dec-POMDPs with agent-state based policies

Decentralized partially observable Markov decision processes (Dec-POMDP) model sequential decision-making problems involving multiple cooperative agents acting under partial observability and asymmetric information.

**Dec-POMDP Model.** A finite-horizon Dec-POMDP is defined by the tuple $\mathcal{D} = \langle \mathsf{N}, \mathsf{S}, \mathsf{A}, \mathsf{Y}, P, r, \zeta_1, T \rangle$, where $\mathsf{N} \coloneqq \{1, \cdots, N\}$ denotes the set of agents and $\mathsf{S}$ is a finite set of environment states. The joint action and observation spaces are $\mathsf{A} = \mathsf{A}^1 \times \cdots \times \mathsf{A}^N$ and $\mathsf{Y} = \mathsf{Y}^1 \times \cdots \times \mathsf{Y}^N$, respectively, where $\mathsf{A}^i$ and $\mathsf{Y}^i$ are the local action and observation sets for agent $i \in \mathsf{N}$. At each time $t$, the environment state is $s_t \in \mathsf{S}$, and the agents take a joint action $a_t = (a_t^1, \ldots, a_t^N)$ and receive a joint observation $y_t = (y_t^1, \ldots, y_t^N)$. The system evolves according to the joint transition–observation distribution $P(s_{t+1}, y_{t+1} \mid s_t, a_t) \coloneqq \mathbb{P}(s_{t+1}, y_{t+1} \mid s_{1:t}, y_{1:t}, a_{1:t})$, and the team receives a shared per-step reward $r(s_t, a_t)$ governed by $r : \mathsf{S} \times \mathsf{A} \to \mathbb{R}$. Finally, $\zeta_1 \in \Delta(\mathsf{S} \times \mathsf{Y})$ specifies the joint distribution of the initial state and observations, and $T \in \mathbb{N}$ is the finite decision horizon.

**Agent-state based policies.** We assume that instead of using their entire history of observations and actions, agents act based on an internal state called an *agent state* (Dong et al., 2022).

Let $\mathsf{Z} = \mathsf{Z}^1 \times \cdots \times \mathsf{Z}^N$ be the space of agent states of all agents, where $\mathsf{Z}^i$ is the agent-state space of agent $i \in \mathsf{N}$. An **agent-state based policy** for agent $i$ is $\pi^i = (\pi_1^i, \ldots, \pi_T^i)$, where $\pi_t^i \colon \mathsf{Y}^i \times \mathsf{Z}^i \to \Delta(\mathsf{A}^i \times \mathsf{Z}^i)$. When following such a policy, at each time each agent samples its action and next agent state as a function of its current observation and agent state, i.e., $(a_t^i, z_t^i) \sim \pi_t^i(y_t^i, z_{t-1}^i)$. We denote the joint policy of all agents as $\pi = (\pi^1, \cdots, \pi^N)$ or more compactly as $\pi = (\pi^i, \pi^{-i})$, when emphasizing the perspective of agent $i$.

Given a joint policy $\pi \in \Pi$, the system evolves as follows. Each agent $i$ initializes its agent state $z_0^i$ independently at random according to some arbitrary distribution $\phi^i$. The initial state and observation

$(s_1, y_1) \sim \zeta_1$. Then, for each time, agent $i$ chooses $(a_t^i, z_t^i) \sim \pi_t^i(y_t^i, z_{t-1}^i)$. The team of all agents receives the reward $r(s_t, a_t)$ and the environment state and agent's observation evolve as $(s_{t+1}, y_{t+1}) \sim P(\cdot|s_t, a_t)$.

We denote the set of all agent-state based policies for agent $i$ by $\Pi^i$, with $\Pi = \prod_{i \in \mathsf{N}} \Pi^i$ and $\Pi^{-i} = \prod_{j \neq i} \Pi^j$. The performance of a joint policy $\pi \in \Pi$ is given by the expected total cumulative reward over horizon $T$:

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} r(S_t, A_t)\right].$$

There are two notions of optimality in Dec-POMDPs. A joint policy $\pi^\star \in \Pi$ is called **globally optimal** if

$$J(\pi^\star) \geq J(\pi), \quad \forall \pi \in \Pi.$$

A joint policy $\pi^\star \in \Pi$ is called **person-by-person optimal (PBPO)** if

$$J(\pi^{i,\star}, \pi^{-i,\star}) \geq J(\pi^i, \pi^{-i,\star}), \quad \forall \pi^i \in \Pi^i, \forall i \in \mathsf{N}.$$

Thus, PBPO corresponds to a Nash equilibrium of Dec-POMDP where all agents share a common utility function (Radner, 1962; Marschak & Radner, 1972). Note that the locally optimal solution is a weaker notion of equilibrium than PBPO, which means that the above inequality is applied to the time index $t$ in addition to the agent index $i$.

### 3.2 Risk-aware MDPs with entropic risk utility

In this section, we review the basic theory of risk-aware MDPs with entropic risk utility (Howard & Matheson, 1972). Consider an MDP $\mathcal{M} = \langle \mathsf{S}, \mathsf{A}, P, r, \zeta_1, T \rangle$, where the components are similar to those defined for Dec-POMDPs. To distinguish with the notation for Dec-POMDPs, we use $\psi = (\psi_1, \ldots, \psi_T)$ to denote the policy, where $\psi_t \colon \mathsf{S} \to \Delta(\mathsf{A})$. The performance of a policy $\psi$ from an initial time $t$ to final time $T$ is given by the entropic risk operator $\mathcal{R}$ with temperature $\lambda$:

$$\begin{aligned} J^{\mathcal{R}}_{1:T}(\psi) &\coloneqq \mathcal{R}^{\psi_1}_{\zeta_1}(r(S_1, A_1) + \mathcal{R}^{\psi_2}(r(S_2, A_2) + \cdots + \mathcal{R}^{\psi_T}(r(S_T, A_T)))) \\ \mathcal{R}^{\psi_t}[r(S_t, A_t)] &= \tfrac{1}{\lambda} \log \mathbb{E}^{\psi}[e^{\lambda r(S_t, A_t)}]. \end{aligned}$$

For a given policy, the risk-aware expected performance can be computed using dynamic programming. Define $V^{\psi}_{T+1}(s) \coloneqq 0$, for any state $s$. Then for any $t \in \{T, \cdots, 1\}$ recursively define:

$$\begin{aligned} Q^{\psi}_t(s, a) &\coloneqq \frac{1}{\lambda} \log\left(\sum_{s_+} P(s_+|s, a) e^{\lambda[r(s,a) + V^{\psi}_{t+1}(s_+)]}\right), \\ V^{\psi}_t(s) &\coloneqq \frac{1}{\lambda} \log\left(\sum_{a} \psi_t(a \mid s) e^{\lambda Q^{\psi}_t(s,a)}\right) \\ \implies J^{\mathcal{R}}_{t:T}(\psi) &= \frac{1}{\lambda} \log\left(\sum_{s} \mathbb{P}^{\overleftarrow{\psi}_t}(S_t = s) e^{\lambda V^{\psi}_t(s)}\right). \end{aligned}$$

The optimal value functions can be computed as follows. Define $V^{\star}_{T+1}(s) \coloneqq 0$, for any state $s$. Then for any $t \in \{T, \cdots, 1\}$ recursively define:

$$\begin{aligned} Q^{\star}_t(s, a) &\coloneqq \frac{1}{\lambda} \log\left(\sum_{s_+} P(s_+|s, a) e^{\lambda[r(s,a) + V^{\star}_{t+1}(s_+)]}\right), \\ V^{\star}_t(s) &\coloneqq \max_{\psi_t} \frac{1}{\lambda} \log\left(\sum_{a} \psi_t(a \mid s) e^{\lambda Q^{\star}_t(s,a)}\right) = \max_a Q^{\star}_t(s, a), \end{aligned}$$

where the last equality follows from the monotonicity of $e^{\lambda x}$. Thus, the optimal policy $\psi^\star = (\psi^\star_1, \ldots, \psi^\star_T)$ is deterministic and satisfies $\psi^\star_t(s) \in \arg\max_a Q^{\star}_t(s, a)$.

# 4 Main approach

A risk-seeking MDP involves a risk-seeking objective. We are interested in optimizing the original risk-neutral objective (expected utility) and we want to use the ideas from the risk-aware literature to converge to better local optima through using a risk-seeking objective in the start, but then gradually annealing down the risk-seeking temperature $\lambda$. Thus, a risk-seeking objective is used to provide a better initialization for the risk-neutral approach.

## 4.1 Applying the risk measure to the Dec-POMDP

For simplicity of notation, we fix $N = 2$. However, the results can easily be extended to any general number of agents $N$. Recall that $y_t \coloneqq (y_t^1, y_t^2)$, $z_t \coloneqq (z_t^1, z_t^2)$ and $a_t \coloneqq (a_t^1, a_t^2)$. We will use the notation $\pi_t(a_t, z_t \mid y_t, z_{t-1}) \coloneqq \pi_t^1(a_t^1, z_t^1 \mid y_t^1, z_{t-1}^1)\pi_t^2(a_t^2, z_t^2 \mid y_t^2, z_{t-1}^2)$.

**Marginal distribution induced by a policy:** The joint environment and agent state process (for all agents) $\{(s_t, y_t, z_{t-1})\}_{t\geq 1}$ is a controlled Markov chain for a given policy $\pi$. Let $\zeta^\pi = (\zeta_1, \zeta_2^{\overleftarrow{\pi}_2}, \zeta_3^{\overleftarrow{\pi}_3}, \cdots, \zeta_T^{\overleftarrow{\pi}_T})$ denote the marginal distribution on $(s_t, y_t, z_{t-1})$ induced by the policy $\pi$, i.e., $\zeta_t^{\overleftarrow{\pi}_t}(s, y, z_-) = \mathbb{P}^{\overleftarrow{\pi}_t}(s_t, y_t, z_{t-1} = s, y, z_- \mid \zeta_1, \overleftarrow{\pi}_t)$.

For ease of notation, we continue to use $\zeta_t^{\overleftarrow{\pi}_t}$ to denote the marginal and conditional distributions w.r.t. $\zeta_t^{\overleftarrow{\pi}_t}$. In particular, for marginals we use $\zeta_t^{\overleftarrow{\pi}_t}(y, z_-)$ to denote $\sum_{s\in\mathsf{S}} \zeta_t^{\overleftarrow{\pi}_t}(s, y, z_-)$ and for conditionals we use $\zeta_t^{\overleftarrow{\pi}_t}(s|y, z_-)$ to denote $\zeta_t^{\overleftarrow{\pi}_t}(s, y, z_-)/\zeta_t^{\overleftarrow{\pi}_t}(y, z_-)$.

The marginal distribution $\zeta_t^{\overleftarrow{\pi}_t}$ can be updated recursively from $\zeta_{t-1}^{\overleftarrow{\pi}_{t-1}}$ and $\pi_{t-1}$ as follows:

$$\zeta_t^{\overleftarrow{\pi}_t}(s_t, y_t, z_{t-1}) = \sum_{\binom{s_{t-1}, y_{t-1},}{a_{t-1}, z_{t-2}}} \zeta_{t-1}^{\overleftarrow{\pi}_{t-1}}(s_{t-1}, y_{t-1}, z_{t-2}) P(s_t, y_t \mid s_{t-1}, a_{t-1}) \pi_{t-1}(a_{t-1}, z_{t-1} \mid y_{t-1}, z_{t-2}).$$

**Applying the risk map function to the original Dec-POMDP objective:** We define the risk-seeking Dec-POMDP objective by applying the risk map at each time step in a dynamic programming fashion, where the probabilities are obtained from the marginal distribution $\{\zeta_t^{\overleftarrow{\pi}_t}\}_{t=1}^T$ for a given policy $\pi$.

The risk-seeking objective for a time-varying policies $\pi^1 = (\pi_1^1, \cdots, \pi_T^1)$ and $\pi^2 = (\pi_1^2, \cdots, \pi_T^2)$ with marginal distribution $\{\zeta_t^{\overleftarrow{\pi}_t}\}_{t=1}^T$ can be written as follows:

$$J_{1:T}^{\mathcal{R}}(\pi) \coloneqq \mathcal{R}_{\zeta_1}^{\pi_1}(r(S_1, A_1) + \mathcal{R}^{\pi_2}(r(S_2, A_2) + \cdots + \mathcal{R}^{\pi_T}(r(S_T, A_T)))). \tag{3}$$

We explain how to compute this risk-seeking objective with the help of a centralized Q-function in the next section.

## 4.2 Risk-seeking conservative policy iteration (RS-CPI)

We now present the CPI approach modified with the influence of risk-seeking rewards. In particular, we want to optimize the original risk-neutral objective, but in order to improve the local optimum that we converge to, we propose to use a risk-seeking objective during the CPI iterations. We start with a value of $\lambda > 0$ and reduce the value to $\lambda \to 0$ as the CPI iterations proceed to completion.

**Risk-seeking centralized Q-function.** Consider the centralized Q-function:

$$Q_T^{\vec{\pi}_T}(s, y, z_-, a, z) \coloneqq r(s, a)$$

and for $t \in \{T-1, \ldots, 1\}$, and with $\lambda > 0$, we have

$$Q_t^{\vec{\pi}_t}(s, y, z_-, a, z) \coloneqq r(s, a) + \frac{1}{\lambda} \log \Big\{ \sum_{\binom{s_+, y_+,}{a_+, z_+}} P(s_+, y_+|s, a) \pi_{t+1}(a_+, z_+ \mid y_+, z) \exp(\lambda Q_{t+1}^{\vec{\pi}_{t+1}}(s_+, y_+, z, a_+, z_+)) \Big\}.$$

**A policy update procedure.** We now present a procedure for updating a policy. The procedure consists of the following steps:

1. Given a policy $\pi$, compute the risk-seeking $Q$-functions $\{Q_t^{\vec{\pi}_t}\}_{t=1}^T$ and compute the marginal distribution $\{\zeta_t^{\overleftarrow{\pi}_t}\}_{t=1}^T$.
2. Compute the sequence of *risk-seeking averaged local Q-functions* defined as follows for a given agent $i$ at time $t$:

$$\bar{Q}_t^{\overleftarrow{\pi}_t, \pi_t^{-i}, \vec{\pi}_t}(y^i, z_-^i, a^i, z^i) = \frac{1}{\lambda} \log \Big\{ \sum \zeta_t^{\overleftarrow{\pi}_t, \pi_t^{-i}}(\cdot \mid y^i, z_-^i) \exp(\lambda Q_t^{\vec{\pi}_t}(s, y, z_-, a, z)) \Big\}.$$

   where the summation is over $s, y^{-i}, z_-^{-i}, a^{-i}, z^{-i}$.
3. Update $\pi_t^i$ with $\bar{\pi}_t^i(y^i, z_-^i) \in \arg\max_{(a^i, z^i)} \bar{Q}_t^{\overleftarrow{\pi}_t, \pi_t^{-i}, \vec{\pi}_t}(y^i, z_-^i, a^i, z^i)$.

We will use the short-hand notation $\bar{\pi}_t^i \in \mathcal{G}_t^i(\pi)$ to denote the update procedure for agent $i$ at time $t$. A key feature of the policy update procedure is the following.

**Proposition 4.1** (Performance Improvement Guarantee)**.** *For any $\pi \in \Pi$, any $t$ and any agent $i$, let $\bar{\pi}_t^i \in \mathcal{G}_t^i(\pi)$. Then, we have $J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \bar{\pi}_t^i, \pi_t^{-i}, \vec{\pi}_t) \geq J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \pi_t, \vec{\pi}_t)$ and a monotonic performance improvement guarantee on $J_{1:T}^{\mathcal{R}}(\pi)$.*

*Proof.* The proof is given in Appendix H. □

**Conservative Policy Update.** Following Kakade & Langford (2002), we present a conservative policy update procedure. Given a forgetting factor $\alpha \in [0, 1]$, define the conservative policy update operator $\mathcal{G}_t^{i,\alpha}$ as follows:

$$\pi_t^{i,\alpha} \in \mathcal{G}_t^{i,\alpha}(\pi) \coloneqq (1 - \alpha)\pi_t^i + \alpha \mathcal{G}_t^i(\pi).$$

Thus, for $\alpha = 1$, the conservative policy update is same as $\mathcal{G}_t^i(\pi)$, but for $\alpha < 1$, the conservative policy update mixes between the old policy $\pi_t$ and a policy in $\mathcal{G}_t^i(\pi)$. The RS-CPI approach is summarized in Algorithm 1.

**Corollary 4.2.** *The result of Proposition 4.1 holds for $\bar{\pi}_t^i = \mathcal{G}_t^{i,\alpha}(\pi)$ for any $\alpha \in [0, 1]$.*

*Proof.* The proof is given in Appendix I. □

**Algorithm 1** RS-CPI for Dec-POMDPs

**Input**: $\pi^{(0)} = (\pi_1^{(0)}, \cdots, \pi_T^{(0)})$, $K$ steps
**Input**: Forgetting rates $(\alpha^{(1)}, \alpha^{(2)}, \cdots, \alpha^{(K)})$.
**Input**: Temperature rates $(\lambda^{(1)}, \lambda^{(2)}, \cdots, \lambda^{(K)})$.

```
k ← 0
while k < K do
    k ← k + 1
    λ ← λ^(k)
    for t ← T : 1 do
        π_t^(k) ← G_t^{α^(k)}(←π_t^(k−1), π_t^(k−1), →π_t^(k))
    end for
end while
return π^(k)
```

**Proposition 4.3** (Convergence Guarantee)**.** *For any choice of forgetting rate $\alpha$ and fixed $\lambda$, the performance $\{J_{1:T}^{\mathcal{R}}(\pi^{(k)})\}_{k \geq 0}$ of the policies generated by RS-CPI converges to a limit $\pi^\circ = (\pi_1^\circ, \ldots, \pi_T^\circ)$ that has the following local optimality property:*

$$\pi_t^{i,\circ} \in \mathcal{G}_t^i(\pi^\circ), \quad \forall t \in \{1, \ldots, T\}, \forall i. \tag{4}$$

*When conservatism is absent (i.e., $\alpha^{(k)} = 1$ for all $k$), the convergence takes place in a finite number of steps. This implies that the converged solution is a locally optimal solution.*

*Proof.* The proof is given in Appendix J. □

*Remark* 4.4. The space complexity of computing the centralized and individual Q-functions ($Q$ and $\bar{Q}$) is $\mathcal{O}(T\mathsf{S}|\mathsf{Y}|^N|\mathsf{Z}|^{2N}|\mathsf{A}|^N)$ and the runtime complexity is $\mathcal{O}(T|\mathsf{S}|^2|\mathsf{Y}|^{2N}|\mathsf{Z}|^{3N}|\mathsf{A}|^{2N})$.

# 5 Numerical Experiments

## 5.1 Details of the experiments

**Environments.** We use the following Dec-POMDPs from the MASPlan benchmark Amato et al. (2014), which is a well known standard for evaluating Dec-POMDPs.

1. DEC TIGER Nair et al. (2003), in which two agents must coordinate their actions (opening a door) based on individual noisy observations.
2. RECYCLING ROBOTS Amato et al. (2012), in which two robots must independently decide whether to recycle small or large cans or return to a charging station.
3. COOPERATIVE BOX PUSHING Seuken & Zilberstein (2012), in which two agents may either push small boxes or coordinate their actions to push a bigger box.
4. MARS ROVERS Amato & Zilberstein (2009), in which multiple rovers must coordinate their trajectories to visit various sites of interest and collect data, under constraints of limited communication.

**Baselines.** We consider the following baselines which have been shown to be near-optimal:[1]

1. Feature-based heuristic search value iteration (FB-HSVI) Dibangoye et al. (2016), which is a state-of-the-art algorithm to find $\epsilon$-optimal *history-based* policies for Dec-POMDPs.
2. Policy-finding multi-agent A$^*$ (PF-MAA$^*$) Koops et al. (2024), which is a computationally efficient heuristic algorithm to find good *history-based* policies in Dec-POMDPs.

**Details for RS-CPI.** For RS-CPI, we keep a constant $\alpha$ and start from an initial $\lambda > 0$ (thus, a risk-seeking objective) which is then linearly decayed to 0, reverting to a risk-neutral objective. We then run additional iterations at $\lambda = 0$ until convergence. The values of the constant $\alpha$ and the initial $\lambda$ are independently chosen for each environment by a hyperparameter search (where performance is estimated over 5 random seeds). Exact performance can be calculated as shown in Appendix B. The initial policy is randomized. For each constant $\alpha$ and initial $\lambda$, we re-run RS-CPI across 5 initializations and report the best performance obtained after convergence.

## 5.2 Experimental Results

**Results for $|\mathsf{Z}^i| = 2$.** For our initial set of experiments, we fix $|\mathsf{Z}^i| = 2$ and report the results in Table 1. We directly use the performance metrics reported by the respective authors for the baselines. We can see from the comparisons across all environments and all horizons that the performance of our RS-CPI algorithm is comparable despite using only 2 possible memory states. Since most of the performance metrics reported from using FB-HSVI and PF-MAA$^*$ are already near-optimal (or optimal), it is not possible to show significant improvements over those baselines.

**Ablation studies.** We performed two sets of ablations. The first was to consider the individual effects of risk-seeking and CPI. We provide the results in Appendix E due to space limitations. The main conclusions here were that using risk-seeking and CPI jointly led to improved performance. Secondly, we considered the effect of increasing the agent state size on the performance. Here we observed an increasing trend in performance with agent state size.

[1] We do not consider the use of the designer's approach Mahajan (2008) or the common information approach Nayyar et al. (2013); Tang et al. (2023) as baselines as they do not provide implementations of their methods and the results in their papers are provided only for small Dec-POMDPs with small horizons, while we require comparisons for larger settings.

Table 1: Performance comparison across benchmarks and planning horizons $T$. RS-CPI uses $|\mathsf{Z}^i| = 2$. The "–" entries were not reported in the corresponding paper.

| $T$ | FB-HSVI | PF-MAA* | RS-CPI |
|---|---|---|---|
| DEC TIGER | | | |
| 6 | **10.38** | **10.38** | **10.38** |
| 7 | **9.99** | **9.99** | 8.75 |
| 8 | **12.22** | 8.36 | 8.38 |
| 9 | **15.57** | **15.57** | **15.57** |
| 10 | **15.18** | **15.18** | 13.76 |
| 20 | 28.75 | **29.12** | 27.63 |
| 50 | 80.66 | **81.03** | 54.11 |
| 100 | 170.90 | **170.91** | 120.30 |
| MARS ROVERS | | | |
| 6 | **18.62** | **18.62** | **18.62** |
| 7 | **20.90** | **20.90** | **20.90** |
| 8 | 22.47 | **22.48** | **22.48** |
| 9 | 24.31 | **24.32** | **24.32** |
| 10 | 26.31 | 26.31 | **26.32** |
| 20 | **52.13** | 49.37 | 50.47 |
| 50 | **128.95** | 122.56 | 126.43 |
| 100 | 249.92 | 234.06 | **252.63** |

| $T$ | FB-HSVI | PF-MAA* | RS-CPI |
|---|---|---|---|
| COOPERATIVE BOX PUSHING | | | |
| 4 | **98.59** | **98.59** | 98.17 |
| 5 | 107.72 | **107.73** | 107.64 |
| 6 | 120.67 | **120.68** | 120.27 |
| 7 | **156.42** | 155.99 | 155.01 |
| 8 | 191.22 | **191.23** | 186.04 |
| 9 | **210.27** | 210.26 | 209.61 |
| 10 | 223.74 | **224.28** | 223.78 |
| 20 | 458.10 | **474.97** | 469.39 |
| 50 | 1134.70 | **1209.80** | 1192.46 |
| 100 | – | **2433.51** | 2352.18 |
| RECYCLING ROBOTS | | | |
| 100 | 308.78 | **308.79** | **308.79** |
| 500 | – | **1539.56** | 1539.17 |
| 1000 | – | **3078.02** | 3077.63 |
| 2000 | – | **6154.94** | 6154.56 |

**Polynomial space and runtime algorithm.** The RS-CPI algorithm has polynomial runtime and memory in the size of the state, observation, action, agent-state spaces of the individual agents. While this is still large due to the product of several spaces being considered, it is still far better in terms of computational complexity than dealing with exponential runtime/memory algorithms. This makes our algorithm much more scalable over larger horizons. We show the runtime and memory scaling over agent state size $Z$ and the horizon $T$ in Appendix F.

**Interpreting reactive policies.** We qualitatively analyze some of the reactive policies for COOPERATIVE BOX PUSHING and MARS ROVERS in Appendix G and we highlight how it is possible to obtain reactive policies that are near-optimal for these Dec-POMDPs. We see that the policies obtained tend to have some interesting emergent properties such as periodicity.

## 6 Conclusions

In this work, we introduced RS-CPI, a novel algorithm that balances risk-seeking exploration with conservative policy updates to optimize agent-state policies under finite memory constraints. Empirical evaluations on standard Dec-POMDP benchmarks demonstrate that our approach matches state-of-the-art performance, despite operating with strictly limited memory. Furthermore, our ablation studies confirm a predictable relationship between memory capacity and performance: expanding the agent state systematically improves the resulting policy.

Beyond these algorithmic contributions, our results offer a broader insight into the nature of Dec-POMDP complexity. We observed that reactive policies perform surprisingly well in certain domains traditionally considered challenging due to their large state spaces. This suggests that the inherent difficulty of a Dec-POMDP is not governed solely by the size of the environment, but fundamentally by the amount of memory agents require to act optimally. When agents require less memory to coordinate effectively, the effective search space of policies shrinks dramatically, rendering seemingly complex problems substantially easier to solve. Moving forward, we plan to explore using risk-seeking updates into a model-free learning framework in order to scale to larger, unknown environments where explicit models are unavailable.

## References

Chris Amato, Frans Oliehoek, and Stefan Witwicki. MASPlan: Decentralized multi-agent planning. https://masplan.org, 2014.

Christopher Amato and Shlomo Zilberstein. Achieving goals in decentralized POMDPs. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 593–600, 2009.

Christopher Amato, Daniel S Bernstein, and Shlomo Zilberstein. Optimizing memory-bounded controllers for decentralized POMDPs. *arXiv preprint arXiv:1206.5258*, 2012.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Jilles Steeve Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. Optimally solving Dec-POMDPs as continuous-state MDPs. *Journal of Artificial Intelligence Research*, 55: 443–497, 2016.

Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. Simple agent, complex environment: Efficient reinforcement learning with agent states. *Journal of Machine Learning Research*, 23(255):1–54, 2022.

Bruno Floriano, Geovany A Borges, and Henrique Ferreira. Planning for decentralized formation flight of uav fleets in uncertain environments with dec-pomdp. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 563–568. IEEE, 2019.

Ronald A Howard and James E Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pp. 267–274, 2002.

Wietze Koops, Sebastian Junges, and Nils Jansen. Approximate Dec-POMDP solving using multi-agent A$^*$. *International Joint Conference on Artificial Intelligence*, 2024.

Aditya Mahajan. *Sequential decomposition of sequential dynamic teams: applications to real-time communication and networked control systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 2008.

J. Marschak. Elements for a theory of teams. *Management Science*, 1(2):127–137, January 1955.

J. Marschak and R. Radner. *Economic Theory of Teams*. Yale University Press, New Haven, 1972.

Ranjit Nair, Milind Tambe, Makoto Yokoo, David Pynadath, and Stacy Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multi-agent settings. In *IJCAI*, volume 3, pp. 705–711, 2003.

Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Frans A Oliehoek, Shimon Whiteson, Matthijs TJ Spaan, et al. Lossless clustering of histories in decentralized pomdps. In *AAMAS (1)*, pp. 577–584, 2009.

Frans A Oliehoek, Matthijs TJ Spaan, Christopher Amato, and Shimon Whiteson. Incremental clustering and expansion for faster optimal planning in Dec-POMDPs. *Journal of Artificial Intelligence Research*, 46:449–509, 2013.

Shayegan Omidshafiei, Ali-Akbar Agha-Mohammadi, Christopher Amato, Shih-Yuan Liu, Jonathan P How, and John Vian. Decentralized control of multi-robot partially observable markov decision processes using belief space macro-actions. *The International Journal of Robotics Research*, 36 (2):231–258, 2017.

Zinovi Rabinovich, Claudia V Goldman, and Jeffrey S Rosenschein. The complexity of multiagent systems: The price of silence. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 1102–1103, 2003.

Roy Radner. Team decision problems. *Annals of Mathematical Statistics*, 33:857–881, 1962.

N.R. Sandell, P. Varaiya, M. Athans, and M.G. Safonov. Survey of decentralized control methods for large scale systems. 23(2):108–128, 1978.

Sven Seuken and Shlomo Zilberstein. Improved memory-bounded dynamic programming for decentralized POMDPs. *arXiv preprint arXiv:1206.5295*, 2012.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on stochastic programming: Modeling and Theory*. SIAM, 2021.

Yun Shen, Wilhelm Stannat, and Klaus Obermayer. Risk-sensitive Markov control processes. *SIAM Journal on Control and Optimization*, 51(5):3652–3672, 2013.

Daniel Szer, François Charpillet, and Shlomo Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. *arXiv preprint arXiv:1207.1359*, 2012.

Dengwang Tang, Ashutosh Nayyar, and Rahul Jain. A novel point-based algorithm for multi-agent control using the common information approach. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 1432–1437. IEEE, 2023.

Roy van Zuijlen and Duarte Antunes. Memoryless policy iteration for episodic POMDPs. *arXiv preprint arXiv:2512.11082*, 2025.

Weikai Wang and Erick Delage. Planning and learning in average risk-aware MDPs. *arXiv preprint arXiv:2503.17629*, 2025.

Hans S Witsenhausen. Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, 59(11):1557–1566, 1971.

Hans S Witsenhausen. A standard form for sequential stochastic control. *Mathematical systems theory*, 7(1):5–11, 1973.

Zhiyuan Yao, Zihan Ding, and Thomas Clausen. Multi-agent reinforcement learning for network load balancing in data center. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3594–3603, 2022.

# Supplementary Materials

*The following content was not necessarily subject to peer review.*

## A Motivating risk-seeking incentives with an example

Standard Markov decision theory typically focuses on maximizing the expected cumulative return, representing a *risk-neutral* objective. In contrast, *risk-aware* formulations evaluate the expected utility of a stochastic return $X$, denoted by $\mathbb{E}[U(X)]$. When the utility function $U$ is concave, Jensen's inequality implies $\mathbb{E}[U(X)] \leq U(\mathbb{E}[X])$; in this case, the decision-maker prefers a deterministic return over a random return with the same expectation, a behavior termed *risk-averse*. Conversely, if $U$ is convex, the inequality is reversed and the agent is *risk-seeking*. The special case where $U$ is linear corresponds to the standard risk-neutral paradigm.

A commonly used risk-aware measure is the *entropic risk* utility, defined as $U(x) = e^{\lambda x}$, where $\lambda \in \mathbb{R}$ parameterizes the degree of risk sensitivity. While the exponential function is always convex, the agent's objective depends on the sign of $\lambda$: a risk-seeking agent ($\lambda > 0$) maximizes $\mathbb{E}[e^{\lambda X}]$, whereas a risk-averse agent ($\lambda < 0$) minimizes $\mathbb{E}[e^{\lambda X}]$. To ensure the risk-aware metric remains on the same scale as the original rewards, we apply the inverse transformation $U^{-1}(\cdot)$ after taking the expectation, yielding the *certainty equivalent* return:

$$\mathcal{R}(X) = \tfrac{1}{\lambda} \log \mathbb{E}[e^{\lambda X}],$$

where $\mathcal{R}(X)$ represents the deterministic return that provides the same utility as the stochastic return $X$.

Risk awareness changes the behavior of the agent. Consider an agent choosing among $m$ mutually exclusive outcomes with with different success probabilities. A risk-seeking agent (convex $U$) will typically bet on a single outcome that offers the highest potential utility, effectively gambling on the best-case scenario. Conversely, a risk-averse agent (concave $U$) will spread their bet across multiple outcomes, sacrificing maximum potential returns to ensure a more consistent and robust baseline performance. For this reason, there is a rich literature on risk *sensitive* Markov decision processes Shen et al. (2013).

Our core insight is that risk-*seeking* behavior naturally encodes a form of *optimism* in the face of uncertainty. By prioritizing high-potential outcomes, this optimistic bias enables iterated best-response methods to escape poor local optima and converge toward superior local solutions.

We consider the iterated best response update (2), similar to JESP Nair et al. (2003). Iterated best response is guaranteed to converge to a local optimum (1). In order to improve the local optimum (higher $J$), we use a risk-seeking modified objective. We provide the following example to illustrate how a risk-seeking objective can converge to a better local optimum.

**Example**: Consider a single stage two-player game with the same reward function (mentioned in table below). Agent 1 has actions: $\{a^1, b^1\}$ and Agent 2 has actions $\{a^2, b^2\}$.

| | | Agent 2 | |
|---|---|---|---|
| | | $a^2$ | $b^2$ |
| Agent 1 | $a^1$ | 2 | $-10$ |
| | $b^1$ | $-10$ | 6 |

Suppose we denote the policy of Agent 1 by $\pi^1(a^1)$ and that for Agent 2 as $\pi^2(a^2)$. Clearly the global optimum (best local optimum) is at $\begin{bmatrix} \pi^1(a^1) \\ \pi^2(a^2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ with a reward of $+6$. There is also an inferior local optimum at $\begin{bmatrix} \pi^1(a^1) \\ \pi^2(a^2) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ with a reward of $+2$.

A JESP style approach typically involves randomly initializing the policy for all agents and then applying iterative best response for each agent one by one while keeping the other agents' policies

fixed at the most recent policy update. For example, if we start with the initialization $\begin{bmatrix} \pi^1(a^1) \\ \pi^2(a^2) \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix}$, then we fix Agent 2's policy and update Agent 1's policy and then vice-versa. This process is repeated until convergence, which follows the trajectory:

$$\begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix} \to \begin{bmatrix} 1 \\ 0.9 \end{bmatrix} \to \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to \cdots,$$

which gives us the inferior local optimum. Similarly, if we start with $\begin{bmatrix} \pi^1(a^1) \\ \pi^2(a^2) \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$:

$$\begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0.9 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \cdots,$$

it gives us the global optimum. This shows that the initialization determines the equilibrium we converge to when using iterated best response. In the first case, the best action combination has a low probability of $0.01$ which is the reason for convergence to the local optimum.

We can remedy this situation by considering the application of risk-seeking rewards because it assigns a higher weight to higher rewards (optimistic), whereas the usual expected reward is risk-neutral. For example, an entropic risk map uses the logsumexp function with a positive temperature $\lambda$. If we then replace the expectation with this logsumexp function in the iterated response for the first case, we get:

$$\begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0.9 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \cdots,$$

and for the second case:

$$\begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \cdots,$$

which both converge to the global optimum! Thus, it is possible to use risk-seeking rewards to reach better local optima.

## B Exact policy evaluation for agent-state based polices in polynomial time

In this section, we show how policy evaluation can be done in a Dec-POMDP for a set of joint agent-state based policies. The key intuition is Lemma B.1.

**Lemma B.1.** *For any joint agent-state based policy $\pi$, the process $\{(S_t, Z_t)\}_{t\geq 1}$ is Markov. The processes $\{(S_t, Z_t, A_t)\}_{t\geq 1}$, $\{(S_t, Y_t, Z_t)\}_{t\geq 1}$ and $\{(S_t, Y_t, Z_t, A_t)\}_{t\geq 1}$ are also Markov.*

Thus, there exists a centralized MDP which has the following time-varying reward-function and dynamics:

$$\bar{r}_t^{\pi_t}(s, y, z) = \sum_{a\in\mathsf{A}} \prod_{i=1}^{N} \pi_t^i(a^i|y^i, z^i) r(s, a), \tag{5}$$

$$\bar{P}_t^{\pi_t}(s', y', z'|s, y, z) = \sum_{a\in\mathsf{A}} \prod_{i=1}^{N} \pi_t^i(a^i, z^{i'}|y^i, z^i) P(s', y'|s, a) \tag{6}$$

We can then evaluate the performance of this time-varying Markov chain via the performance evaluation formulas for MDPs. In particular, define

$$J(s_1, y_1, z_0; \pi) = \bar{r}_1^{\pi_1} + \bar{P}_1^{\pi_1}\bar{r}_2^{\pi_2} + \cdots + \bar{P}_1^{\pi_1}\bar{P}_2^{\pi_2}\cdots\bar{P}_{T-1}^{\pi_{T-1}}\bar{r}_T^{\pi_T}.$$

to be the performance when $z_0$ is the arbitrary initial agent state and $(s_1, y_1)$ are the staring state and observation respectively. Note that $J$ as written above is a tensor of dimensions $|\mathsf{S}| \times |\mathsf{Y}| \times |\mathsf{Z}|$. Thus the final performance by using the initial state observation distribution $\zeta_1$ and can be calculated as:

$$J(\pi) = \sum_{s,y} \zeta_1(s, y) J(s, y, z_0; \pi).$$

Also note that we are only reporting final performance for the risk-neutral problem, however, if one is interested in the risk-seeking problem's performance, then the expectations should be modified by the appropriate function, e.g.: logsumexp function in the case of entropic risk.

# C Risk-aware MDPs with rewards instead of costs based on Shen et al. (2013); Wang & Delage (2025)

## C.1 Risk measure

First consider the definition for the basic risk map from Shapiro et al. (2021), which consists of a mapping from a random variable (reward) to a real number and essentially quantifies the risk of that particular random variable. Consider a finite probability space $(\Omega, \mathcal{F}, \mathcal{P})$: where $\Omega$ is a finite set of outcomes, $\mathcal{F}$ is the sigma-algebra of the outcomes, and $\mathcal{P}$ is the associated probability mass function. Let $\mathcal{L}(\Omega)$ denote the set of finite real-valued functions (random variables) on $\Omega$.

**Definition C.1.** Given a finite probability space $(\Omega, \mathcal{F}, \mathcal{P})$, a risk measure $\rho\colon \mathcal{L}(\Omega) \to \mathbb{R}$ maps a random variable (reward) to a real value quantifying its risk. The risk measure $\rho$ is called a monetary risk measure if it satisfies the following three properties:

1. **Monotonicity**: $\rho(v) \leq \rho(w)$, for all $v, w \in (\Omega)$ such that $v \leq w$ a.s.
2. **Translation Invariance**: $\rho(v + \lambda) = \rho(v) + \lambda$ for any $v \in \mathcal{L}(\Omega), \lambda \in \mathbb{R}$.
3. **Normalization**: $\rho(0) = 0$.

Furthermore, such a monetary risk measure $\rho$ may also satisfy the following properties:

1. **Convexity**: For all $v, w \in \mathcal{L}(\Omega), \alpha \in [0, 1]$, $\rho(\alpha v + (1 - \alpha)w) \leq \alpha\rho(v) + (1 - \alpha)\rho(w)$.
2. **Concavity**: For all $v, w \in \mathcal{L}(\Omega), \alpha \in [0, 1]$, $\rho(\alpha v + (1 - \alpha)w) \geq \alpha\rho(v) + (1 - \alpha)\rho(w)$.
3. **Positive Homogeneity**: For all $v \in \mathcal{L}(\Omega), \lambda \geq 0$, $\rho(\lambda v) = \lambda\rho(v)$.

If the monetary risk measure $\rho$ satisfies concavity and positive homogeneity properties, then it is called a coherent risk measure.

*Remark* C.2. Since we are using the reward formulation instead of the standard cost formulation, a higher value given by $\rho$ indicates a more desirable outcome. So ideally we would want to maximize the objective constructed using the risk measure $\rho$.

Coherent risk measures are common in the finance literature. Concavity is associated with risk-averse behavior and convexity is associated with risk-seeking behavior. See Section 4.2 of Shen et al. (2013) for an intuitive explanation of how concave/convex corresponds to risk-averse/seeking in terms of economics and uncertainty.

*Remark* C.3. Note that outside being a monetary risk measure, none of the more advanced properties like concavity and coherency are necessary to consider risk-aware behavior. Although it is convenient to consider concave/convex risk maps to guarantee the appropriate risk-aware behavior everywhere.

## C.2 Markov decision processes

A Markov decision process (MDP) models sequential decision-making problems in which a single agent interacts with a stochastic environment over time. A finite-horizon MDP is defined by the tuple

$$\mathcal{M} = \langle \mathsf{S}, \mathsf{A}, P, r, \zeta_1, T \rangle,$$

where:

- $\mathsf{S}$ is a finite set of environment states.
- $\mathsf{A}$ is a finite set of actions available to the agent.
- $P$ is the state transition kernel, where

$$P(s' \mid s, a) = \mathbb{P}(s_{t+1} = s' \mid s_t = s,\, a_t = a),$$

for all $s, s' \in \mathsf{S}$ and $a \in \mathsf{A}$.

- $r : \mathsf{S} \times \mathsf{A} \to \mathbb{R}$ is the reward function.
- $\zeta_1 \in \Delta(\mathsf{S})$ is the initial state distribution.
- $T \in \mathbb{N}$ is the finite decision horizon.

At each time step $t = 1, \cdots, T$, the system is in state $s_t \in \mathsf{S}$. The agent selects an action $a_t \in \mathsf{A}$, receives a reward $r(s_t, a_t)$, and the environment transitions to a new state $s_{t+1}$ according to the transition kernel $P$. A decision making rule (or policy) at each time step can be used to select the action $a_t$ based on the state $s_t$, i.e., $\psi_t \colon \mathsf{S} \to \Delta(\mathsf{A})$. We refer to the combined decision making rules over all time jointly as the overall policy $\psi = (\psi_1, \cdots, \psi_T)$.

The objective function $J$ for an MDP is the cumulative reward over the finite horizon $T$ and is a function of the policy:

$$J(\psi) = \mathbb{E}^{\psi}\left[\sum_{t=1}^{T} r(S_t, A_t)\right].$$

The objective is to find a policy $\psi$ that maximizes the cumulated reward over time $J(\psi)$:

$$\psi^{\star} \coloneqq \sup_{\psi \in \Psi} J(\psi).$$

### C.3 Risk maps

Given the definition of a risk measure, we can now look at the definition of a risk map $\mathcal{R}$ applied to an MDP $\mathcal{M}$ as follows.

**Definition C.4.** A risk map $\mathcal{R}$ is a function that maps each state action pair $(s, a) \in \mathsf{S} \times \mathsf{A}$ to a monetary risk measure on the space $(\mathsf{S}, P(\cdot \mid s, a))$. For a given value function random variable $V$ (over all states), the risk map for $(s, a)$ is given by $\mathcal{R}(V \mid s, a)$. Suppose we are using a policy $\psi$ at state $s$, then we write this as $\mathcal{R}^{\psi}(V|s)$. This can be further simplified to $\mathcal{R}_s^{\psi}(V)$.

The risk aware objective for a time-varying policy $\pi = (\psi_1, \cdots, \psi_T)$ can then be written as follows:

$$J^{\mathcal{R}}(\psi) \coloneqq r^{\psi_1}(S_1) + \mathcal{R}_{S_1}^{\psi_1}(r^{\psi_2}(S_2) + \mathcal{R}_{S_2}^{\psi^2}(r^{\psi_3}(S_3) + \cdots + \mathcal{R}_{S_{T-1}}^{\psi^{T-1}}(r^{\psi_T}(S_T)))). \tag{7}$$

The goal is to solve for the optimal time-varying policy for the risk-aware MDP:

$$\psi^{\star} \coloneqq \sup_{\psi \in \Psi} J^{\mathcal{R}}(\psi).$$

### C.4 Example choices for risk map functions

The following are some examples from Shen et al. (2013) for risk map functions $\mathcal{R}$.

1. **Classical MDP risk neutral**: This is the standard risk-neutral reward function used in MDPs. Since it is linear in the value function $V$, it satisfies concavity and convexity. Since it also satisfies positive homogeneity, it is coherent as well.

$$\mathcal{R}(V \mid s, a) \coloneqq \mathbb{E}^{P}\left[V \mid s, a\right] = \sum_{s'} P(s' \mid s, a) V(s').$$

2. **Entropic map**: This is based on the logsumexp function. The parameter $\lambda \in \mathbb{R}$ controls the risk-preference of $\mathcal{R}$. If $\lambda > 0$, then $\mathcal{R}$ is convex everywhere and therefore risk-seeking everywhere. If $\lambda < 0$, $\mathcal{R}$ is concave everywhere and therefore risk-averse everywhere. It can also be shown

that the limit as $\lambda \to 0$ is the same as the classical MDP risk neutral reward.

$$\begin{aligned}\mathcal{R}(V \mid s, a) &\coloneqq \frac{1}{\lambda} \log \mathbb{E}^P \left[\exp(\lambda V) \mid s, a\right] \\ &= \frac{1}{\lambda} \log \left\{ \sum_{s'} P(s' \mid s, a) \exp(\lambda V(s')) \right\}.\end{aligned}$$

3. **Mean-semideviation trade-off**: This involves modifying risk based on deviation from the expected values of rewards. $\lambda \in \mathbb{R}$ works similar to the entropic map. The notation $x_+$ denotes $\max(x, 0)$ and $k \geq 1$.

$$\mathcal{R}(V \mid s, a) \coloneqq \mathbb{E}^P \left[V \mid s, a\right] + \lambda \mathbb{E}^P \left[(V - \mathbb{E}^P[V \mid s, a])_+^k \mid s, a\right]^{\frac{1}{k}}.$$

# D Related work

Decentralized Partially Observable Markov Decision Processes (Dec-POMDP) model multi-agent sequential decision-making problems where agents with asymmetric partial observations must coordinate their actions to optimize a shared cumulative reward. Such models arise in various applications including autonomous drone fleets (Floriano et al., 2019), network load balancing (Yao et al., 2022), and multi-robot co-ordination (Omidshafiei et al., 2017). Consequently, Dec-POMDPs have been studied extensively across multiple research disciplines including Economics and Organizational Behavior as team theory (Marschak, 1955; Radner, 1962; Marschak & Radner, 1972) and Systems and Control as decentralized stochastic control (Witsenhausen, 1971; 1973; Sandell et al., 1978; Nayyar et al., 2013).

## D.1 Monotonic improvement dynamic programs / policy iteration schemes for POMDPs

A similar approach to the monotonic improvement DP explored in this paper was developed by van Zuijlen & Antunes (2025), which applies the same idea but for a single agent POMDP to learn a time-varying reactive (considers only the most recent observation as input) policy for the finite horizon case.

## D.2 Risk-aware methods

Instead of using risk-neutral MDPs, we can consider risk-aware MDPs as Wang & Delage (2025). This involves modifying the original expected reward to a reward which is mapped through some risk measure. Such risk measures can lead to risk-averse/risk-seeking behaviors depending on the hyper-parameters used.

Shen et al. (2013) give a very useful overview of general risk measures and how to appropriately apply them for MDPs (since most papers in the space use a cost-based risk-averse approach). They define a risk measure and explain the specific of the properties that a risk measure needs in order to be considered risk-averse or risk-seeking for the MDP case. They look at obtaining time-homogeneous policies for the infinite horizon discounted case and the infinite horizon average cost setting. They show the existence of optimal policies for both these settings and show that it is well defined for the MDP case. While Wang & Delage (2025) also consider time-homogeneous policies in the average cost setting for risk-aware MDPs, the distinction in their work is that they present planning and learning algorithms to obtain provably optimal policies for general risk-aware measures.

## D.3 A$^*$ variants

There are several variants of the A$^*$ algorithm which involves using a heuristic function to optimistically estimate the remaining utility from any given part of the exponential search tree. This idea was first combined with Dec-POMDPs in Multi-agent A$^*$ (Szer et al., 2012). They proposed

trying several heuristic such as the MDP heuristic and centralized tighter but harder to compute POMDP heuristic and also a recursive heuristic using MAA$^*$. They present empirical results on small Dec-POMDPs. This idea is generalized by GMAA$^*$ (Oliehoek et al., 2008) which breaks down the overall MAA$^*$ algorithm into fundamental steps denoted as "Next": which controls how partial policies are expanded and also returns their heuristic values; and "Select": which chooses the partial policy from the current policy pool to expand. They connect the MAA$^*$ problem to a tree search over collaborative Bayesian games. The GMAA$^*$-Incremental Clustering and Expansion (Oliehoek et al., 2013) algorithm uses lossless incremental clustering to group histories together that satisfy the notion of probabilistic equivalence which is based on having the same joint distribution over the states and histories of the other agents. Once the clustering is done at a certain stage, the clusters themselves can be incrementally updated, rather than updating all the histories from scratch while retaining the lossless property (still an exact search). To further reduce the size of exponential tree search, the propose using incremental expansion which selectively expands the tree one node at a time based on the heuristic function.

Koops et al. (2024) introduced the Policy-Finding Multi-agent A$^*$ (PF-MAA) algorithm, which uses a sliding window of the past observations along with a novel heuristic that periodically uses the state which is tighter than the MDP heuristic. They cluster the sliding window observations (Oliehoek et al., 2009; 2013). In addition, they propose the use of a heuristic for pruning nodes which are less promising in order to reduce the tree size of the exponential search. While an (approximately) optimal solution is not guaranteed by this method, they provide empirical results using large horizons for several Dec-POMDP problems which are very close to the optimal value.

### D.4 Exact and approximate exponential complexity algorithms

Dibangoye et al. (2016) introduce a method called Feature-based Heuristic Search Value Iteration (FB-HSVI) and Mahajan (2008) introduces a method called the designer's approach. They construct an MDP with the state space as the joint state, agent-state space and action space as the space of single step decision rules rather than individual joint actions of each agent. They propose using a dynamic programming formulation to obtain optimal policies. Mahajan (2008) refers to this idea as the designer's approach (originally introduced by Witsenhausen (1973)), whereas Dibangoye et al. (2016) refers to it as the occupancy MDP. A distinction between the two approaches is that the dynamic program provided by Mahajan (2008) gives the optimal solution within the given class of agent-state based policies, whereas Dibangoye et al. (2016) gives the history-based optimal solution. These methods have exponential time complexity. However, Dibangoye et al. (2016) groups similar histories/decision-rules together and uses finite history windows rather than the full history to make the occupancy MDP more compact. This makes it more efficient in practice in Dec-POMDPs with a lot of compact structure, however, it still remains of exponential time complexity and this would give sub-optimal solutions when there is important information beyond the finite history window. Another notable contribution is the use of feature based heuristics to further reduce the exponential search and provide upper and lower bounds, which lead to an anytime algorithm which is very useful in practice. This allows them to scale the results on several standard Dec-POMDPs.

Another idea is that of the common information approach (Nayyar et al., 2013) which relies on common information structure between agents to simplify the size of the centralized MDP. This helps in reducing the overall computation when their is common information, however, this still has exponential runtime complexity. A further variation on the common information approach has been considered in Tang et al. (2023).

### D.5 Sequential iterated best response algorithms

Joint Equilibrium-based Search for Policies (JESP) (Nair et al., 2003) finds locally optimal joint policies by iteratively optimizing one agent's policy while keeping others fixed using a multi-agent-belief system. This approach, particularly the dynamic programming version (DP-JESP), leverages

Table 2: Ablations with risk-seeking and conservatism for the different planning horizons $T$. All experiments use $|\mathsf{Z}^i| = 2$.

| $T$ | CPI only | RS only | No CPI + No RS | RS-CPI |
|---|---|---|---|---|
| | | DEC TIGER | | |
| 6 | **10.38** | −12.00 | −12.00 | **10.38** |
| 7 | **8.75** | −14.00 | −14.00 | **8.75** |
| 8 | 6.50 | −16.00 | −16.00 | **8.38** |
| 9 | 10.10 | −18.00 | −18.00 | **15.57** |
| 10 | 8.42 | −20.00 | −20.00 | **13.76** |
| 20 | 23.17 | −40.00 | −40.00 | **27.63** |
| 50 | 47.95 | −100.00 | −100.00 | **54.11** |
| 100 | 91.93 | −200.00 | −200.00 | **120.30** |
| | | RECYCLING ROBOTS | | |
| 100 | 308.40 | 308.40 | 308.40 | **308.79** |
| 500 | 1536.21 | 1536.21 | 1536.21 | **1539.17** |
| 1000 | 3074.56 | 3071.01 | 3071.01 | **3077.63** |
| 2000 | 6148.05 | 6143.33 | 6143.33 | **6154.56** |
| | | COOPERATIVE BOX PUSHING | | |
| 4 | 19.43 | 18.70 | 19.43 | **98.17** |
| 5 | 35.28 | 26.39 | 35.28 | **107.64** |
| 6 | 78.56 | 35.94 | 78.37 | **120.27** |
| 7 | 82.73 | 36.96 | 79.32 | **155.01** |
| 8 | 53.00 | 44.17 | 53.00 | **186.04** |
| 9 | 96.55 | 53.60 | 95.63 | **209.61** |
| 10 | 55.95 | 55.20 | 103.25 | **223.78** |
| 20 | 124.03 | 115.44 | 157.79 | **469.39** |
| 50 | 435.48 | 294.58 | 404.04 | **1192.46** |
| 100 | 771.35 | 597.68 | 753.44 | **2352.18** |
| | | MARS ROVERS | | |
| 6 | 10.89 | **18.62** | 10.75 | **18.62** |
| 7 | 12.63 | 18.42 | 12.60 | **20.90** |
| 8 | 14.34 | 18.22 | 12.97 | **22.48** |
| 9 | 16.16 | 18.64 | 15.83 | **24.32** |
| 10 | 18.02 | 24.68 | 17.74 | **26.32** |
| 20 | 35.80 | 49.37 | 35.72 | **50.47** |
| 50 | 89.37 | 124.31 | 88.21 | **126.43** |
| 100 | 178.60 | 249.29 | 178.40 | **252.63** |

the piece-wise linear and convexity (PWLC) properties of the value function to provide exponential speedups over exhaustive search methods.

# E Ablation studies

We perform two sets of ablation studies. The first considers the effects of using risk-seeking utility and conservatism. The second considers the effect of varying size of the agent-state memory $|\mathsf{Z}|$.

**Ablation study with risk-seeking behavior and conservatism.** The effect of adding risk-seeking and conservatism in the overall performance can be seen in Table 2. We perform the experiments under the same hyperparameter settings (i) with CPI only, (ii) with risk-seeking only, (iii) without CPI and risk-seeking and compare it with our resulting algorithm RS-CPI that combines all of them. Clearly the ablations show that it is critical to jointly use CPI and risk-seeking together as we can see that performance deteriorates when we remove either/both of them.

Table 3: Table for comparison of number of agent states $|\mathsf{Z}^i|$ with performance. We did not perform experiments for MARS ROVERS for $|\mathsf{Z}^i| = 4$.

| $T$ | $\|\mathsf{Z}^i\| = 1$ | $\|\mathsf{Z}^i\| = 2$ | $\|\mathsf{Z}^i\| = 3$ | $\|\mathsf{Z}^i\| = 4$ |
|---|---|---|---|---|
| DEC TIGER | | | | |
| 20 | −31.77 | 27.63 | 31.58 | 30.69 |
| 50 | −80.90 | 54.11 | 71.64 | 69.71 |
| 100 | −123.01 | 120.30 | 145.14 | 152.04 |
| COOPERATIVE BOX PUSHING | | | | |
| 20 | 448.09 | 469.39 | 472.97 | 474.25 |
| 50 | 1112.67 | 1192.46 | 1200.06 | 1204.52 |
| 100 | 2228.57 | 2352.18 | 2410.70 | 2421.33 |
| MARS ROVERS | | | | |
| 20 | 50.47 | 52.36 | 50.59 | – |
| 50 | 126.43 | 126.34 | 127.24 | – |
| 100 | 252.63 | 251.95 | 253.93 | – |

**Ablation study with increasing agent state size $|\mathsf{Z}^i|$.** It is natural to expect greater performance if we allow larger agent state sizes. Thus, we perform this experiment for $|\mathsf{Z}^i| = 1, 2, 3, 4$ and show the results in Table 3. Note that $|\mathsf{Z}^i| = 1$ corresponds to the open-loop policy (reactive policy). We can see from the results that using more memory helps most in the case of Dec-Tiger, where the reactive policy performs poorly. However, surprisingly, the reactive policies on the Cooperative Box Pushing and Mars Rovers Dec-POMDPs perform well and adding memory to these agents does not lead to a significant performance increase. We give intuitions on the obtained reactive policies for COOPERATIVE BOX PUSHING and MARS ROVERS in Appendix G and we highlight how it is possible to obtain reactive policies that are near-optimal for these Dec-POMDPs.

This suggests that while these Dec-POMDPs are significantly larger, they are still simpler than the smaller Dec-Tiger Dec-POMDP in the sense that they can consider simplified reactive policies without having to worry about maintaining an explicit agent state representation. As a result, this greatly simplifies the policy search process in Cooperative Box Pushing and Mars Rovers Dec-POMDPs which means that apart from having large state, observation and action spaces — these problems are not as hard as one might think. This highlights the requirement for more challenging benchmarks for Dec-POMDPs, not only in the sense of larger spaces, but also in terms of agents requiring memory to perform well.

## F Memory and runtime analysis

The update function involved in Algorithm 1 involves storing a Q-table that depends linearly on the horizon $T$, which implies that the memory required by our algorithm is linear in the horizon $T$ of the problem rather than exponential. Furthermore, the runtime of computing the centralized Q-function $Q_t^{\vec{\pi}_t}$ is polynomial in the state, observation, action spaces and agent-state spaces of the individual agents in the Dec-POMDP, which is a significant reduction from exponential.

We consider Dec-POMDPs which have two agents and we use the same agent-state size across them. This means that the memory and runtime complexity ends up being polynomial in the size of the agent state that we choose commonly for both agents. We show the memory metrics in the first two rows of Figure 1 and the runtime metrics over the next two rows. For each setting we took the mean and standard deviation of the 5 random seeds considered.

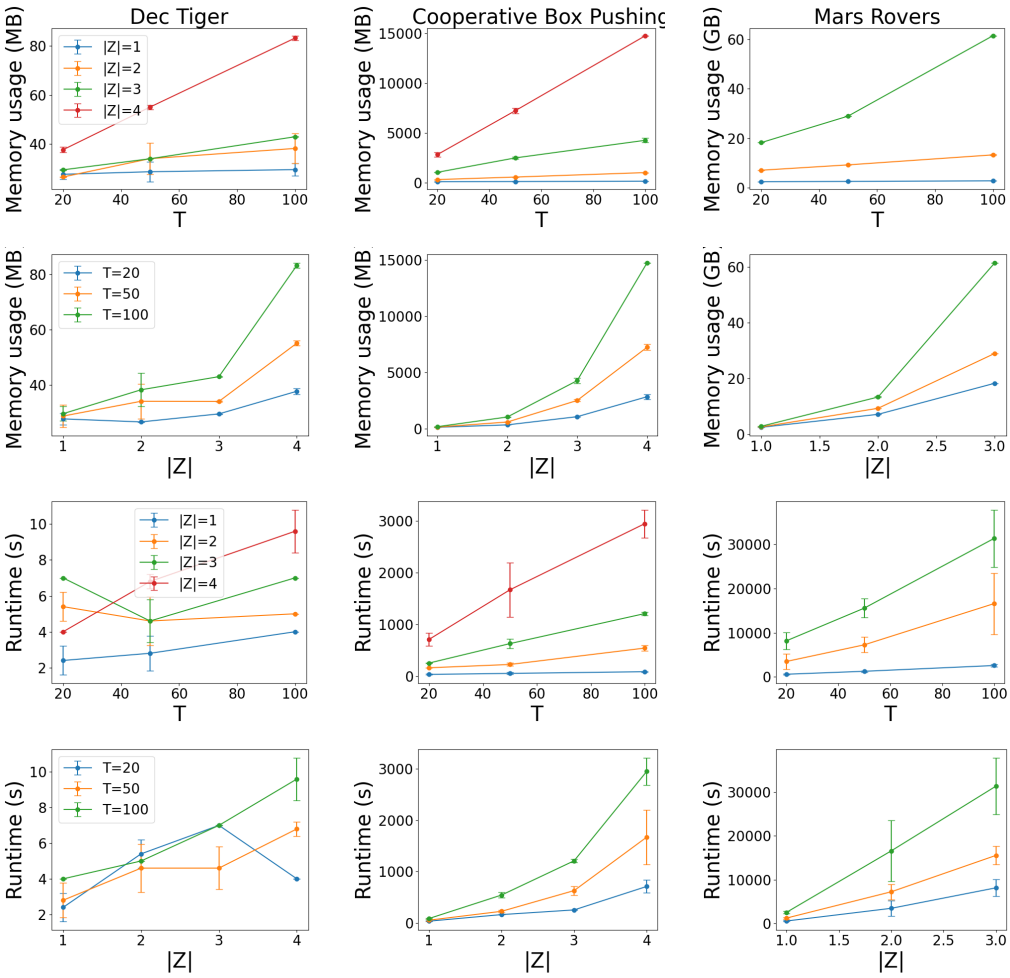


Figure 1: Scaling of memory and runtime with horizon $T$ and agent-state size $Z$ across three benchmarks: Dec-Tiger, Cooperative Box Pushing, and Mars Rovers. Rows correspond to different metrics, columns correspond to Dec-POMDPs.

# G Interpreting Reactive Policies

## G.1 Cooperative Box Pushing Reactive Policy analysis

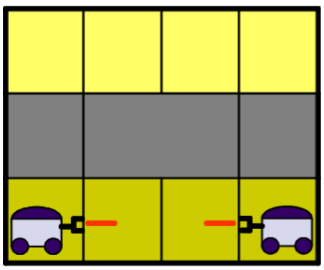

Figure 2: Cooperative Box Pushing (Image take from Seuken & Zilberstein (2012)

In this Dec-POMDP, there are two agents, one in the left corner and the other in the right corner. They are initially facing the center of the arena (indicated by the red lines in Fig. 2). There is a small box denoted as SmallB immediately adjacent to each of their starting position.

A larger box denoted as LargeB lies in the middle. To reach this box the agents would have to move to the center. Agents can push a box by moving into it. The SmallB can be moved by a single agent, however, the LargeB requires both agents to simultaneously move into it. There is a larger reward for pushing the LargeB and a minor reward for pushing the SmallB , while agents are penalized for bumping into other objects.

The can turn its orientation using TurnL or TurnR and can move one step in front of it by moving ForwardİT can also choose to Stay in the same spot (which is useful sometimes for cooperating with the other agent to push the LargeB ). There is also some probability that agents fail to Forward or TurnL or TurnR which makes the environment more challenging.

We present our learned RS-CPI reactive policy which is near-optimal in Table 4. In order to understand how to read this policy and understand what is happening properly, we provide the following steps where we illustrate what happens at each time step. We trace out the trajectory of what happens when the movement actions of both agents always succeed. We mark this trajectory in bold in the Table 4. We abbreviate Agent 1 as A1 (left) and Agent 2 as A2 (right).

1. A1 sees **Empty** and chooses **Forward**.
   A2 sees **Empty** and chooses **Forward**.
2. A1 sees **OtherA** and chooses **TurnL**.
   A2 sees **OtherA** and chooses **TurnR**.
   (Both agents are now facing the **LargeB**)
3. A1 sees **LargeB** and chooses **Forward**.
   A2 sees **LargeB** and chooses **Forward**.
   (Both agents move forward and get large reward for pushing the **LargeB**)
4. The environment then gets reset back to the same initial state in this step, the actions do not matter in a reset step.
5. It is interesting to note that same cycle from steps 1-3 continue in a periodic fashion.

It might also be instructive to observe what happens in case an action fails. For example let's assume that when both agents do Forward , then A1 fails. We mark the actions in blue. Then:

1. A1 sees **Empty** and chooses Forward (but fails).
   A2 sees **Empty** and chooses Forward (success).
2. A1 sees **Empty** and chooses TurnL.
   A2 sees **Empty** and chooses Forward.
3. A1 sees **SmallB** and chooses Forward.
   A2 sees **OtherA** and chooses TurnR.
   (A small reward is received for pushing the small box successfully)
4. This step is the rest step again.
5. And then we get back into the cycle!

So even when the best outcome doesn't work out, there is some contingency for the agents to finish the episode and restart it to try again. In this manner many different contingencies can be worked out, where the resulting actions are just the outcome of a process that overall has a tendency to maximize reward.

### G.2 Cooperative Box Pushing Reactive Policy analysis

This domain consists of two agents in a $2 \times 2$ grid. The agents can move UP , DOWN , LEFT , RIGHT , take a SAMPLE or DRILL at any of the $4$ locations. Two of the sites require only one agent

| Time | Empty | Wall | OtherA | SmallB | LargeB |
|---|---|---|---|---|---|
| 1 | **Forward** | TurnL | TurnL | TurnL | TurnL |
| | **Forward** | TurnL | TurnL | TurnL | TurnL |
| 2 | TurnL | TurnL | **TurnL** | Stay | TurnL |
| | Forward | TurnR | **TurnR** | TurnL | TurnL |
| 3 | TurnL | TurnL | TurnL | Forward | **Forward** |
| | Forward | TurnR | TurnR | TurnL | **Forward** |
| 4 | TurnL | TurnL | TurnL | Forward | Forward |
| | Forward | TurnR | TurnR | TurnL | Forward |
| 5 | **Forward** | TurnL | TurnL | Forward | Forward |
| | **Forward** | TurnR | TurnR | Forward | Forward |
| 6 | Forward | TurnL | **TurnL** | Stay | Stay |
| | Forward | TurnR | **TurnR** | Stay | Stay |
| 7 | Forward | Stay | TurnL | Forward | **Forward** |
| | Stay | TurnR | TurnR | Forward | **Forward** |
| 8 | TurnL | TurnL | TurnL | Forward | Forward |
| | Forward | TurnR | TurnR | Forward | Forward |
| 9 | **Forward** | TurnL | TurnL | Stay | Stay |
| | **Forward** | TurnR | TurnR | Stay | Stay |
| 10 | Forward | TurnL | **TurnL** | Forward | Forward |
| | Forward | TurnR | **TurnR** | Forward | Forward |
| 11 | TurnL | TurnL | TurnL | Forward | **Forward** |
| | Forward | TurnR | TurnR | Forward | **Forward** |
| 12 | Forward | TurnL | TurnL | Forward | Forward |
| | Forward | TurnR | TurnR | Forward | Forward |
| 13 | **Forward** | TurnL | TurnL | Stay | Stay |
| | **Forward** | TurnR | TurnR | Stay | Stay |
| 14 | Forward | TurnL | **TurnL** | Forward | Forward |
| | Forward | TurnR | **TurnR** | Forward | Forward |
| 15 | TurnL | TurnL | TurnL | Forward | **Forward** |
| | Forward | TurnR | TurnR | Forward | **Forward** |
| 16 | Forward | TurnL | TurnL | Forward | Forward |
| | Forward | TurnR | TurnR | Forward | Forward |
| 17 | **Forward** | TurnL | TurnL | Forward | Forward |
| | **Forward** | TurnR | TurnR | Forward | Forward |
| 18 | Forward | TurnL | **TurnL** | Forward | Forward |
| | Forward | TurnL | **TurnR** | Forward | Forward |
| 19 | TurnL | TurnR | TurnL | Forward | **Forward** |
| | TurnR | TurnL | TurnR | Forward | **Forward** |
| 20 | TurnL | TurnL | TurnL | Forward | Forward |
| | TurnL | TurnL | TurnL | Forward | Forward |

Table 4: **Cooperative Box Pushing**: Joint policies over time. Top action in each cell: Agent 1, bottom action: Agent 2.

| $t$ | Pos1-NS | Pos2-NS | Pos3-NS | Pos4-NS | Pos1-S | Pos2-S | Pos3-S | Pos-S |
|---|---|---|---|---|---|---|---|---|
| 1 | SAMPLE | UP | UP | UP | UP | UP | UP | UP |
| | RIGHT | UP | UP | UP | UP | UP | UP | UP |
| 2 | SAMPLE | SAMPLE | LEFT | UP | DOWN | UP | UP | UP |
| | SAMPLE | UP | SAMPLE | UP | RIGHT | UP | UP | UP |
| 3 | SAMPLE | SAMPLE | LEFT | LEFT | DOWN | UP | LEFT | DOWN |
| | RIGHT | SAMPLE | SAMPLE | UP | RIGHT | DOWN | DOWN | UP |
| 4 | SAMPLE | SAMPLE | LEFT | LEFT | DOWN | RIGHT | LEFT | DOWN |
| | SAMPLE | RIGHT | SAMPLE | SAMPLE | RIGHT | RIGHT | DOWN | UP |
| 5 | DRILL | DOWN | UP | DOWN | DOWN | RIGHT | LEFT | UP |
| | DRILL | UP | DOWN | DOWN | RIGHT | RIGHT | DOWN | DOWN |
| 6 | DOWN | SAMPLE | SAMPLE | DRILL | DOWN | SAMPLE | SAMPLE | UP |
| | SAMPLE | RIGHT | SAMPLE | DRILL | RIGHT | RIGHT | DOWN | LEFT |
| 7 | UP | SAMPLE | LEFT | DOWN | DOWN | RIGHT | LEFT | DOWN |
| | UP | SAMPLE | SAMPLE | SAMPLE | RIGHT | UP | DOWN | UP |
| 8 | DRILL | DOWN | SAMPLE | DOWN | DOWN | RIGHT | LEFT | UP |
| | DRILL | DOWN | UP | DOWN | RIGHT | RIGHT | DOWN | UP |
| 9 | DRILL | SAMPLE | SAMPLE | DRILL | DOWN | SAMPLE | LEFT | DOWN |
| | DRILL | SAMPLE | SAMPLE | DRILL | RIGHT | RIGHT | DOWN | UP |
| 10 | DRILL | SAMPLE | LEFT | DOWN | DOWN | RIGHT | LEFT | DOWN |
| | DRILL | RIGHT | SAMPLE | DOWN | RIGHT | RIGHT | DOWN | DOWN |
| 11 | SAMPLE | DOWN | LEFT | DOWN | DOWN | RIGHT | DOWN | DOWN |
| | RIGHT | RIGHT | DOWN | DOWN | RIGHT | RIGHT | DOWN | UP |
| 12 | SAMPLE | SAMPLE | DOWN | DRILL | DOWN | SAMPLE | SAMPLE | UP |
| | RIGHT | UP | SAMPLE | DRILL | RIGHT | UP | DOWN | UP |
| 13 | DRILL | DOWN | UP | LEFT | DOWN | RIGHT | DOWN | UP |
| | DRILL | RIGHT | UP | DOWN | RIGHT | RIGHT | DOWN | UP |
| 14 | DRILL | SAMPLE | SAMPLE | DRILL | RIGHT | SAMPLE | SAMPLE | UP |
| | DRILL | SAMPLE | SAMPLE | DRILL | DOWN | RIGHT | DOWN | UP |
| 15 | DRILL | UP | SAMPLE | DOWN | RIGHT | RIGHT | DOWN | UP |
| | DRILL | SAMPLE | SAMPLE | DOWN | DOWN | RIGHT | DOWN | DOWN |
| 16 | SAMPLE | UP | UP | DOWN | RIGHT | RIGHT | DOWN | DOWN |
| | DOWN | RIGHT | DOWN | DOWN | DOWN | RIGHT | DOWN | LEFT |
| 17 | UP | UP | SAMPLE | DRILL | RIGHT | SAMPLE | SAMPLE | UP |
| | UP | SAMPLE | DOWN | DRILL | DOWN | RIGHT | DOWN | LEFT |
| 18 | DRILL | SAMPLE | UP | DOWN | RIGHT | RIGHT | DOWN | LEFT |
| | DRILL | DOWN | SAMPLE | DOWN | DOWN | RIGHT | DOWN | LEFT |
| 19 | UP | SAMPLE | SAMPLE | DRILL | DOWN | UP | SAMPLE | LEFT |
| | UP | SAMPLE | SAMPLE | DRILL | RIGHT | RIGHT | DOWN | LEFT |
| 20 | DRILL | SAMPLE | SAMPLE | SAMPLE | UP | UP | UP | UP |
| | DRILL | SAMPLE | SAMPLE | SAMPLE | UP | UP | UP | UP |

Table 5: **Mars Rovers**: Joint policies over time. Top action in each cell: Agent 1, bottom action: Agent 2.

to SAMPLE them while two of the sites require both agents to coordinate and DRILL at the same time in order to get the maximum reward.

At any time the agents can observe their current position in addition to whether an experiment ( SAMPLE or DRILL ) had previously been carried out at that particular location. The NS-suffix in the observation denotes that an experiment has not been done on the site and S-suffix denotes that an experiment has been done. If a site needs to be SAMPLE , but DRILL is done instead, then it incurs a large penalty for ruining the sample, whereas doing the opposite gives a small positive reward. Performing an experiment at any location more than once incurs negative rewards.

We present our learned RS-CPI reactive policy which is near-optimal in Table 5. A similar approach can be followed to the Cooperative Box Pushing example to analyze the policy given in Table 5.

## H Proof of Proposition 4.1

**Risk-seeking centralized value and action-value functions.** We start by defining centralized value and action-value functions for risk-seeking evaluation of a policy $\pi$ with risk parameter $\lambda > 0$.

$$Q_{\lambda,T}^{\vec{\pi}_{\lambda,T}}\left(s, \left[\begin{smallmatrix} y \\ z_- \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ z \end{smallmatrix}\right]\right) = r(s,a)$$

and for $t \in \{T-1, \dots, 1\}$, we have

$$Q_{\lambda,t}^{\vec{\pi}_t}\left(s, \left[\begin{smallmatrix} y \\ z_- \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ z \end{smallmatrix}\right]\right) = r(s,a) + \tfrac{1}{\lambda}\log\Bigg[\sum_{s_+, y_+} P(s_+, y_+ \mid s, a)\exp\Big(\lambda V_{\lambda,t+1}^{\pi_{t+1}\vec{\pi}_{t+1}}\left(s_+, \left[\begin{smallmatrix} y_+ \\ z \end{smallmatrix}\right]\right)\Big)\Bigg],$$

where

$$V_{\lambda,t}^{\pi_t,\vec{\pi}_t}\left(s, \left[\begin{smallmatrix} y \\ z_- \end{smallmatrix}\right]\right) = \tfrac{1}{\lambda}\log\Bigg[\sum_{a,z} \pi_t(a, z \mid y, z_-)\exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\left(s, \left[\begin{smallmatrix} y \\ z_- \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ z \end{smallmatrix}\right]\right)\Big)\Bigg].$$

Based on this risk-seeking policy evaluation, define a risk-seeking performance as

$$\begin{aligned} J_{1:T}^{\mathcal{R}}(\pi) &\coloneqq \mathcal{R}_{\zeta_1}^{\pi_1}(r(S_1, A_1) + \mathcal{R}^{\pi_2}(r(S_2, A_2) + \cdots + \mathcal{R}^{\pi_T}(r(S_T, A_T)))) \\ &= \tfrac{1}{\lambda}\log\Bigg[\sum_{s_1, y_1, z_0} \zeta_1(s_1, y_1)\phi(z_0)\exp\Big(\lambda V_{\lambda,1}^{\pi}\left(s_1, \left[\begin{smallmatrix} y_1 \\ z_0 \end{smallmatrix}\right]\right)\Big)\Bigg] \end{aligned}$$

where $\phi(z_0) = \phi^1(z_0^1)\phi^2(z_0^2)$ (recall that $\phi^i$ is the distribution of the initial agent state of agent $i$). Recall that the local average risk-seeking action-value functions is defined as

$$\begin{aligned} \bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^{-i}, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^i \\ z_-^i \end{smallmatrix}\right], \left[\begin{smallmatrix} a^i \\ z^i \end{smallmatrix}\right]\right) = \tfrac{1}{\lambda}\log\Bigg[&\sum_{s, y^{-i}, z_-^{-i}, a^{-i}, z^{-i}} \zeta_t^{\bar{\pi}_t, \pi_t^{-i}}\left(s, \left[\begin{smallmatrix} y^{-i} \\ z_-^{-i} \end{smallmatrix}\right], \left[\begin{smallmatrix} a^{-i} \\ z^{-i} \end{smallmatrix}\right] \,\middle|\, \left[\begin{smallmatrix} y^i \\ z_-^i \end{smallmatrix}\right]\right) \\ &\exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\left(s, \left[\begin{smallmatrix} y \\ z_- \end{smallmatrix}\right], \left[\begin{smallmatrix} a \\ z \end{smallmatrix}\right]\right)\Big)\Bigg]. \end{aligned}$$

By definition of $\bar{\pi}_t^2$ (which is deterministic), for all $(y^2, z_-^2) \in \mathsf{Y} \times \mathsf{Z}$,

$$\bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^1, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right], \bar{\pi}_t^2\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right]\right)\right) \geq \bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^1, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right], \left[\begin{smallmatrix} a^2 \\ z^2 \end{smallmatrix}\right]\right), \quad \forall \left[\begin{smallmatrix} a^2 \\ z^2 \end{smallmatrix}\right] \in \mathsf{A}^2 \times \mathsf{Z}^2.$$

Moreover, since $\lambda > 0$, we have for all $\left[\begin{smallmatrix} a^2 \\ z^2 \end{smallmatrix}\right] \in \mathsf{A}^2 \times \mathsf{Z}^2$

$$\exp\Big(\lambda \bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^1, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right], \bar{\pi}_t^2\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right]\right)\right)\Big) \geq \exp\Big(\lambda \bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^1, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right], \left[\begin{smallmatrix} a^2 \\ z^2 \end{smallmatrix}\right]\right)\Big).$$

Hence,

$$\begin{aligned} \exp\Big(\lambda \bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^1, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right], \bar{\pi}_t^2\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right]\right)\right)\Big) \geq \sum_{a^2, z^2} &\pi_t^2\left(\left[\begin{smallmatrix} a^2 \\ z^2 \end{smallmatrix}\right] \,\middle|\, \left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right]\right) \\ &\exp\Big(\lambda \bar{Q}_{\lambda,t}^{(\bar{\pi}_t, \pi_t^1, \vec{\pi}_t)}\left(\left[\begin{smallmatrix} y^2 \\ z_-^2 \end{smallmatrix}\right], \left[\begin{smallmatrix} a^2 \\ z^2 \end{smallmatrix}\right]\right)\Big) \end{aligned}$$

Substituting the definition of the averaged action-value functions, we get

$$\begin{aligned}
&\sum_{s,y^1,z^1_-,a^1,z^1} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^1}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix} \Big| \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big) \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \bar{\pi}_t^2\Big(\begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big)\Big)\Big) \\
&\geq \sum_{a^2,z^2} \pi_t^2\Big(\begin{bmatrix} a^2 \\ z^2 \end{bmatrix} \Big| \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big) \sum_{s,y^1,z^1_-,a^1,z^1} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^1}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix} \Big| \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big) \\
&\qquad \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big)\Big) \\
&= \sum_{s,y^1,z^1_-,a^1,z^1,a^2,z^2} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^1,\pi_t^2}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix} \Big| \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big) \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big)\Big).
\end{aligned} \tag{8}$$

Observe that

$$\zeta_t^{\overleftarrow{\pi}_t,\pi_t^1,\pi_t^2}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix} \Big| \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big) = \frac{\zeta_t^{\overleftarrow{\pi}_t,\pi_t^1,\pi_t^2}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big)}{\zeta_t^{\overleftarrow{\pi}_t}\Big(\begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}\Big)}.$$

Substituting this in (8) and canceling the denominator $\zeta_t^{\vec{\pi}_t}(y^2, z^2_-)$ from both sides, we get

$$\begin{aligned}
&\sum_{s,y^1,z^1_-,a^1,z^1,a^2,z^2} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^1,\bar{\pi}_t^2}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big) \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big)\Big) \\
&\geq \sum_{s,y^1,z^1_-,a^1,z^1,a^2,z^2} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^1,\pi_t^2}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big) \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big)\Big),
\end{aligned} \tag{9}$$

where in the LHS we are using the fact that $\bar{\pi}_t^2$ is a deterministic policy and therefore assigns positive weight to only $\bar{\pi}_t^2(y^2, z^2_-)$.

Recall that the expected risk-seeking performance from time $t$ onward is given by

$$\begin{aligned}
J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \pi_t^1, \pi_t^2, \vec{\pi}_t) = \tfrac{1}{\lambda} \log\Bigg[ &\sum_{s,y^1,z^1_-,a^1,z^1,a^2,z^2} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^1,\pi_t^2}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big) \\
&\exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y^1 \\ z^1_- \end{bmatrix}, \begin{bmatrix} y^2 \\ z^2_- \end{bmatrix}, \begin{bmatrix} a^1 \\ z^1 \end{bmatrix}, \begin{bmatrix} a^2 \\ z^2 \end{bmatrix}\Big)\Big)\Bigg].
\end{aligned}$$

Then, since $\lambda > 0$, (9) implies that

$$J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \pi_t^1, \bar{\pi}_t^2, \vec{\pi}_t) \geq J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \pi_t^1, \pi_t^2, \vec{\pi}_t).$$

The result then follows from the observation that for any policy $\pi$:

$$J_{1:T}^{\mathcal{R}}(\pi) = \mathcal{R}_{\zeta_1}^{\pi_1}(r(S_1, A_1) + \mathcal{R}^{\pi_2}(r(S_2, A_2) + \cdots + J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \pi_t, \vec{\pi}_t))).$$

# I Proof of Corollary 4.2

$$\pi_t^{i,\alpha} \in \mathcal{G}_t^{i,\alpha}(\pi) \coloneqq (1-\alpha)\pi_t^i + \alpha \mathcal{G}_t^i(\pi).$$

$$\begin{aligned}
J_{t:T}^{\mathcal{R}}(\overleftarrow{\pi}_t, \pi_t^{i,\alpha}, \pi_t^{-i}, \vec{\pi}_t) = (1-\alpha)\tfrac{1}{\lambda} \log&\Bigg[\sum_{s,y,z_-,a,z} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^{-i},\bar{\pi}_t^i}\Big(s, \begin{bmatrix} a \\ z \end{bmatrix} \Big| \begin{bmatrix} y \\ z_- \end{bmatrix}\Big) \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y \\ z_- \end{bmatrix}, \begin{bmatrix} a \\ z \end{bmatrix}\Big)\Big)\Bigg] \\
+ \alpha \tfrac{1}{\lambda} \log&\Bigg[\sum_{s,y,z_-,a,z} \zeta_t^{\overleftarrow{\pi}_t,\pi_t^{-i},\pi_t^i}\Big(s, \begin{bmatrix} a \\ z \end{bmatrix} \Big| \begin{bmatrix} y \\ z_- \end{bmatrix}\Big) \exp\Big(\lambda Q_{\lambda,t}^{\vec{\pi}_t}\Big(s, \begin{bmatrix} y \\ z_- \end{bmatrix}, \begin{bmatrix} a \\ z \end{bmatrix}\Big)\Big)\Bigg].
\end{aligned}$$

Since we have that

$$J^{\mathcal{R}}_{t:T}(\overleftarrow{\pi}_t, \bar{\pi}^i_t, \pi^{-i}_t, \vec{\pi}_t) \geq J^{\mathcal{R}}_{t:T}(\overleftarrow{\pi}_t, \pi_t, \vec{\pi}_t),$$

we have by linearity that

$$J^{\mathcal{R}}_{t:T}(\overleftarrow{\pi}_t, \pi^{i,\alpha}_t, \pi^{-i}_t, \vec{\pi}_t) \geq J^{\mathcal{R}}_{t:T}(\overleftarrow{\pi}_t, \pi_t, \vec{\pi}_t).$$

## J Proof of Proposition 4.3

Corollary 4.2 implies that $\{J^{\mathcal{R}}_{1:T}(\pi^{(k)})\}_{k\geq 0}$ is a non-decreasing sequence which is uniformly bounded (since the per-step rewards are bounded). Therefore, the sequence $J^{\mathcal{R}}_{1:T}(\pi^{(k)})$ converges. When conservatism is absent (i.e., $\alpha^{(k)} = 1$ for all $k$), the policies $\{\pi^{(k)}\}_{k\geq 1}$ are deterministic (assume that ties in $\arg\max$ are broken by an arbitrary but fixed rule). Since there are a finite number of deterministic policies, the convergence must happen in a finite number of steps.

We now show that the limit policy will satisfy (4). Using Corollary 4.2, we can show that for any $t$, $J^{\mathcal{R}}_{t:T}(\pi^{(k)})$ also converges to a limit. Therefore, Eq. (4) must hold (otherwise, we can show by contradiction that $J^{\mathcal{R}}_{t:T}(\pi^{(k)})$ has not converged).