TEAMWORK REINFORCEMENT LEARNING WITH CONCAVE UTILITIES

Zheng Yu
Princeton University
zhengy@princeton.edu

Junyu Zhang
National University of Singapore
junyuz@nus.edu.sg

Zheng Wen
DeepMind
zhengwen@google.com

Andrea Tacchetti
DeepMind
atacchet@google.com

Mengdi Wang
Princeton University
mengdiw@princeton.edu

Ian Gemp
DeepMind
imgemp@google.com

ABSTRACT

Complex reinforcement learning (RL) tasks often require a divide-and-conquer approach, where a large task is divided into pieces and solved by individual agents. In this paper, we study a teamwork RL setting where individual agents make decisions on disjoint subsets (blocks) of the state space and have private interests (reward functions), while the entire team aims to maximize a general long-term team utility function and may be subject to constraints. This team utility, which is not necessarily a cumulative sum of rewards, is modeled as a nonlinear function of the team's joint state-action occupancy distribution. By leveraging the inherent duality of policy optimization, we propose a min-max multi-block policy optimization framework to decompose the overall problem into individual local tasks. This enables a federated teamwork mechanism where a team lead coordinates individual agents via reward shaping, and each agent solves her local task defined only on their local state subset. We analyze the convergence of this teamwork policy optimization mechanism and establish an $O(1/T)$ convergence rate to the team's joint optimum. This mechanism allows team members to jointly find the global socially optimal policy while keeping their local privacy.

1 INTRODUCTION

Recent years have witnessed advances in reinforcement learning with multiple agents in many areas Zhang *et al.* (2019) such as sensor/communication networks Choi *et al.* (2009); Xu *et al.* (2020), autonomous driving Shalev-Shwartz *et al.* (2016), and cyber-physical systems Yong and Brintrup (2019). Most of these works discuss a multi-agent reinforcement learning (MARL) setup, which addresses a sequential decision-making problem of multiple autonomous agents that operate in a **common** environment, each of which aims to optimize its own **long-term return** by interacting with the environment and other agents.

In this work, we propose a new reinforcement learning model with multiple agents involved, namely the *teamwork reinforcement learning* (teamwork RL) model. In contrast to typical MARL models, we focus on the teamwork setting where each agent governs **a distinct subset of the state space** and makes decisions independently from others. Specifically, we consider a decomposition of the state space of a Markov decision process (MDP), where the resulting disjoint state blocks are assigned to individual agents. Each agent can only obtain access to the transition dynamics and choose actions inside its own block while transitions out of the block are hidden. Each agent makes sole decisions in her block, and may have private interests that are not revealed to other members of the team. To coordinate individual agents together, we let there be a team lead who only has access to the transition information between different blocks. The team lead helps individual agents collaborate together to maximize a general team utility, which can include not only agents' private rewards but also **nonlinear team goals**. Such a model captures a broad class of cooperative RL tasks where each agent is only permitted access to a part of the environment.

We are particularly interested in two common scenarios under the teamwork RL model: 1) teamwork RL with a joint team utility, where the team goal can be represented as a nonlinear function of individual blocks' dynamics, such as minimizing KL-divergence in imitation learning tasks or maximizing state-wise entropy in exploration tasks; and 2) teamwork RL with shared costs constraints, where we consider linear cost constraints across agents, for example, to prevent a robotic arm from relying too heavily on a specific joint motor throughout a sequence of assembly tasks. Due to space complexity, we focus on the first scenario and refer the discussion of shared costs constraints to Appendix D.

To solve the teamwork RL model, we propose a min-max multi-block policy optimization framework using the dual domain to decompose the problem into individual local tasks. This enables a semi-decentralized teamwork mechanism where the team lead coordinates individual agents through "reshaping" their local rewards, and each agent solves the resulting regularized variant of the MDP defined on its own state block in parallel, which can be handled by any viable methods such as the variational policy gradient (VPG) method Zhang *et al.* (2020) or a Frank-Wolfe type algorithm Hazan *et al.* (2019). The proposed collaboration framework coordinates through the connecting states between different blocks so that little internal information within each state block will be exposed. We analyze the convergence of this teamwork mechanism and establish an $O(1/T)$ sublinear convergence rate. Our main contributions are summarized as follows:

- We study a state-decomposed teamwork model, that allows us to handle both agents' private interests and coupled utility across agents. This model differs from a typical multi-agent RL model in two aspects: first, the agents operate on disjoint subsets of the state space; second, the utility is not simply the cumulative return, but a general concave function. To our knowledge, teamwork RL is the first framework to handle this challenging combination.
- We discuss duality-inspired multi-agent coordination algorithms, making use of a team lead and regularization, for finding a socially optimal solution. Under this framework, each agent faces local MDP problems defined on its own block without disclosing much private information.
- We discuss two interesting cases for teamwork RL models: 1) teamwork RL with joint team utility; and 2) teamwork RL with a shared cost constraint. We propose two augmented Lagrangian methods (ALMs) and analyse their sublinear convergence guarantees. To our knowledge, our proposed method is the first existing ALM-type algorithm for multi-block affinely constrained min-max problems. We make use of the bilinear structure of the problem by incorporating a novel consensus term into the augmented Lagrangian, which guarantees sublinear convergence without introducing significant communication overhead.

2 A Teamwork RL Model

2.1 Problem Formulation

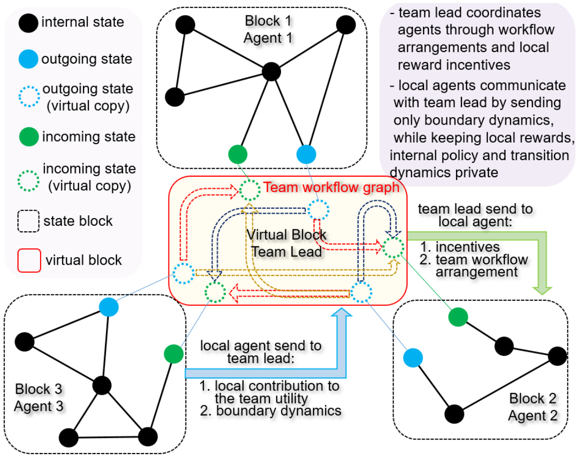
Consider a discounted Markov decision process (MDP), denoted as the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, \gamma, \Xi\}$, over a finite state set $\mathcal{S}$ and a finite action set $\mathcal{A}$. For each state $s \in \mathcal{S}$, the probability of transiting to state $s' \in \mathcal{S}$ when selecting action $a \in \mathcal{A}$ is given by the transition kernel $P(s'|s,a)$. Let $\gamma \in (0,1)$ be the discount factor and $\Xi : \mathcal{S} \to [0,1]$ be the given initial state distribution of the MDP.

Team and state blocks Let there be a team of N agents choosing actions in this MDP. Let the state space $\mathcal{S}$ be divided into N disjoint blocks, $\mathcal{S}_1, \ldots, \mathcal{S}_N$, handled by the N agents respectively. For each state block $\mathcal{S}_i$, the local policy $\pi_i(\cdot|s_i)$ provides a distribution over $\mathcal{A}$ for all $s_i \in \mathcal{S}_i$. Each member agent has direct access to the environment model within its state block, but cannot access what happens outside.

Let there be a team lead who can oversee block-to-block state transitions and communicate with member agents, but cannot access information on the model regarding the interior (excluding the boundary) of each block.

Incoming and outgoing state sets To characterize the connection between disjoint state blocks, we define the incoming state set from block j to block i as: $I_{j\to i} = \{s_i \in \mathcal{S}_i | \exists\ s_j \in \mathcal{S}_j, \exists\ a \in \mathcal{A},\ \text{s.t. } P(s_i|s_j,a) > 0\}$, and the outgoing state set from block i to block j as: $O_{i\to j} = \{s_i \in \mathcal{S}_i | \exists\ s_j \in \mathcal{S}_j, \exists\ a \in \mathcal{A},\ \text{s.t.} P(s_j|s_i,a) > 0\}$.

We further denote $O_i = \cup_{j\neq i} O_{i\to j}$, $O_{-i} = \cup_{j\neq i} O_{j\to i}$ and $I_i = \cup_{j\neq i} I_{j\to i}$. For ease of exposition, we assume the state sets O_{-i}'s are non-overlapping, i.e., $O_i = \sqcup_{j\neq i} O_{i\to j}$. Our results directly extend to the general cases.



● internal state
● outgoing state
○ outgoing state (virtual copy)
● incoming state
○ incoming state (virtual copy)
□ state block
□ virtual block
Block 1 Agent 1
- team lead coordinates agents through workflow arrangements and local reward incentives
- local agents communicate with team lead by sending only boundary dynamics, while keeping local rewards, internal policy and transition dynamics private
Team workflow graph
Virtual Block Team Lead
team lead send to local agent:
1. incentives
2. team workflow arrangement
local agent send to team lead:
1. local contribution to the team utility
2. boundary dynamics
Block 3 Agent 3
Block 2 Agent 2

Figure 1: Illustration of teamwork RL model.

Team workflow graph The transitions from one agent's block to another agent's block is governed by a team workflow graph, which can be viewed as a stochastic graph with directed edges from outgoing and incoming states. For example, the workflow graph can model how different parts of an assembly line connect with each other. The team lead may directly have control over this workflow graph, e.g., directly assigning workload to an agent.

More generally, the team lead can choose a control $u \in \mathcal{A}'$ at an outgoing state s so that the workload from s will be assigned to other agents' incoming states with probability distribution $T(\cdot|s, u)$. For example, the workflow graph may be pre-fixed, so that the graph's transition matrix $T(s|s')$ stays unchanged. For another example, the team lead may have power to directly assign traffic from one to another agent, in this case the team lead's optimization problem is equivalent to directly optimizing over static graphs. We visualize the whole picture of our model in Figure 1.

Team objective We consider maximizing a team objective function that is not limited to the sum of individual agents' cumulative rewards, but can also be a general joint team utility function $R(\cdot)$. In particular, we consider the problem

$$\max_{\pi=\{\pi_i\}_{i=1}^N} R(\pi) = \rho(g_1(\pi_1; \pi_{-1}), \cdots, g_N(\pi_N; \pi_{-N})). \tag{1}$$

Here the mapping $g_i(\pi_i; \pi_{-i})$ denotes the contribution of the i-th agent to the team utility, which is defined to be a *linear* functional of the state-action occupancy measure restricted to block $\mathcal{S}_i$, and ρ is some general smooth concave utility coupling across agents. We remark such scenario indeed captures any smooth concave utility when we set mapping g_i to be identity.

Consider the dual formulation of (1). In many interesting cases, the team utility $R(\pi)$ can be written as a general concave function of the *unnormalized cumulative discounted state-action occupancy measure*, or *flux* under policy π, denoted as μ^π, given by $\mu^\pi(s, a) := \sum_{t=0}^\infty \gamma^t P(s_t = s, a_t = a \,|\, \pi, s_0 \sim \Xi)$. We emphasize that, even in the single-agent case, problem (1) is no longer a typical MDP, and the overall objective is no longer a sum of rewards. The loss of additivity invalidates the foundation of dynamic programming and RL: *the value function is no longer defined, the Bellman equation no longer holds, and dynamic programming fails.*

3 A Multi-Block Optimization Framework for Teamwork RL

In this section, we propose a distributed optimization framework for solving the teamwork RL problem. We inspect the dual space of the problem and propose an iterative multi-agent primal-dual optimization framework, which coordinates each agent through workflow arrangement and reward reshaping. This transformation allows the resulting local problem for each agent to be modeled as a regularized local MDP defined on its own state block, as further discussed in Section 4.1.

3.1 Min-Max Policy Optimization via Occupancy Measures

We consider the dual problem of teamwork RL (1). It can be formulated in terms of the unnormalized cumulative discounted state-action occupancy measure. Specifically, denote the occupancy measure restricted to each state block $\mathcal{S}_i$ as μ_i. Problem (1) is equivalent to an optimization problem with flux and nonnegativity constraints, where the decision variable is the state-action occupancy measure μ:

$$\max_{\mu\geq 0} \rho(C_1\mu_1, \cdots, C_N\mu_N) \quad \text{s.t.} \quad \sum_a \mu(s,a) = \gamma \sum_{s',a'} \mu(s',a')P(s|s',a') + \Xi(s) \;\; \forall s \tag{2}$$

where C_i's are the user-given coefficient matrices with respect to each block, which defines the local linear function $g_i(\pi_i, \pi_{-i}) = C_i\mu_i$. Here we overload the notation of ρ to be a concave joint team utility that couples individual agents.

To handle the non-linear utility, we use Fenchel duality $\rho(C_1\mu_1, \cdots, C_N\mu_N) = \inf_z \sum_i \langle C_i\mu_i, z_i\rangle - \rho^*(z)$, where ρ^* is the concave dual function of ρ and z_i's can be viewed as shadow rewards. In this way we can rewrite problem (2) as:

$$\max_{\mu\geq 0} \min_z \sum_i \langle C_i\mu_i, z_i\rangle - \rho^*(z) \quad \text{s.t.} \quad \sum_a \mu(s,a) = \gamma \sum_{s',a'} \mu(s',a')P(s|s',a') + \Xi(s) \;\; \forall s \tag{3}$$

We remark that the flux constraints and the non-negativity constraints ensure that any feasible occupancy measure μ corresponds to a unique policy π, given by $\pi(a \mid s) = \mu(s,a)/(\sum_{a\in\mathcal{A}} \mu(s,a))$.

Note the occupancy measure μ_i's of different agents are coupled together in the flux constraints. In what follows, we will leverage the linear program nature of the overall MDP and propose a separable reformulation, in order to motivate a semi-decentralized teamwork algorithm.

Adding dummy variables We introduce the following dummy variables on connecting states:

- For each outgoing state $s \in O_i$ and workflow arrangement u, we create a dummy copy $\nu_{\text{out}}(s,u)$ for its workflow graph occupancy, satisfying $\nu_{\text{out}}(s,u) = \mu(s,u)$.
- For each incoming state $s \in I_i$, we create a dummy variable $\nu_{\text{in}}(s)$ for its state-wise occupancy, satisfying $\nu_{\text{in}}(s) = \sum_{s'\in O_{-i},u'} \nu_{\text{out}}(s',u')T(s|s',u')$.

A multi-Block reformulation Using the dummy variables above, the flux constraints in Eq. (3) can be rewritten as:

$$\sum_{a\in\mathcal{A}} \mu(s,a) = \gamma \sum_{s'\in\mathcal{S}_i, a'\in\mathcal{A}} \mu(s',a')P(s|s',a') + \gamma \mathbb{1}_{\{s\in I_i\}}\nu_{\text{in}}(s) + \Xi(s) \quad \forall s \in \mathcal{S}_i, \forall i \in [N]$$

(local flux / MDP's dynamics constraints)

$$\nu_{\text{in}}(s) = \sum_{s'\in O_{-i},u'} \nu_{\text{out}}(s',u')T(s|s',u') \;\; \forall s \in I_i, \forall i \qquad \text{(block-to-block dynamics constraint)}$$

$$\nu_{\text{out}}(s,u) = \mu(s,u) \quad \forall s \in O_i, i \in [N], \forall u. \qquad \text{(boundary condition)}$$

Based on the above reformulation, problem (2) can be abstracted as a multi-agent min-max optimization problem with a bilinear coupling term for some given constant A_i, B_i,b:

$$\min_{x\in\mathcal{X}} \max_z \; f(x,z) = \rho^*(z) - \sum_{i=1}^{N+1} z_i^\top B_i x_i \quad \text{s.t.} \quad Ax = \sum_{i=1}^{N+1} A_i x_i = b \tag{4}$$

where the variable x_i for $i \in [N]$ denotes the local variable $\{\mu_i, \nu_{\text{in},i}\}$ for the i-th agent, characterizing its local dynamics on state block $\mathcal{S}_i$, and the variable x_{N+1} denotes the dummy variables ν_{out} updated by the team leader. We denote the constraint set $\mathcal{X} = \otimes_{i=1}^{N+1}\mathcal{X}_i$, where $\mathcal{X}_i$'s are closed convex sets defined by the intersection of the local flux / MDP's dynamics constraints and the non-negativity constraint to help us formulate the local subproblem as a MDP. We refer to Appendix E for their detailed definitions and expressions.

3.2 Primal-Dual Proximal Block Coordinate Method

To handle the multi-agent min-max optimization problem (4), we propose an augmented Lagrangian method (ALM) approach, where we dualize the coupled linear constraints $Ax = b$ to have the following augmented Lagrangian with λ being the dual variable:

$$\mathcal{L}_\eta(x, \lambda, z) = -\lambda^\top(Ax - b) + \frac{\eta_1}{2}\|Ax - b\|_2^2 + \rho^*(z) - \sum_{i=1}^{N+1} z_i^\top B_i x_i + \frac{\eta_2}{2}\sum_{i=1}^{N+1}\|B_i x_i - \nabla_{z_i}\rho^*(z)\|_2^2$$

Here we introduce an extra regularization term $\|B_i x_i - \nabla_{z_i}\rho^*(z)\|_2^2$ corresponding to the optimality condition for dual variable z, which shares the same spirit of consensus optimization in min-max problems Mescheder *et al.* (2017); Abernethy *et al.* (2019). Now we propose the following min-max ALM update scheme:

$$\begin{cases} \text{primal: } x^{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \mathcal{L}(x, \lambda^k, z^k; \eta_1, \eta_2) + \frac{1}{2}\|x - x^k\|_Q^2 \\ \text{dual: } \lambda^{k+1} = \lambda^k - \eta_1(Ax^{k+1} - b), \\ \qquad z_i^{k+1} = z_i^k + \eta_2(\nabla_{z_i}\rho^*(z^k) - B_i x_i^{k+1}) \end{cases}$$

We pick the predetermined proximal term $Q = \alpha I - \eta_1 A^\top A \succeq 0$ to make the primal update separable for each agent, so that a parallel implementation can be achieved.

A sub-linear convergence guarantee

Theorem 1 (Min-max sublinear convergence)**.** *Assume ρ^* is concave with an L^*-Lipschitz continuous gradient. Assume $D_x := \sup_{x \in \mathcal{X}, Ax=b}\|x\|_Q < +\infty$ and $D_z := \sup_{x \in \mathcal{X}, Ax=b}\{\|z\| : z \in \operatorname{argmax}_{z'} f(x, z')\} < +\infty$. Then by choosing parameters $\eta_1 > 0, \eta_2 \in (0, 1/L^*]$ and $\alpha \geq \eta_1\|A\|_2^2$, we have for any $\gamma > 0$:*

$$\max_z f(\widetilde{x}^t, z) - \min_{x \in \mathcal{X}, Ax=b} f(x, \widetilde{z}^t) + \gamma\|A\widetilde{x}^t - b\|_2 \leq \frac{1}{2(1+t)}\Big(\frac{1}{\eta_1}(\gamma + \|\lambda_0\|)^2 + (D_x + \|x_0\|)^2 + \frac{1}{\eta_2}(D_z + \|z_0\|)^2\Big)$$

where $\widetilde{x}^t = \frac{1}{1+t}\sum_{k=0}^t x^{k+1}, \widetilde{z}^t = \frac{1}{1+t}\sum_{k=0}^t z^{k+1}, \widetilde{\lambda}^t = \frac{1}{1+t}\sum_{k=0}^t \lambda^{k+1}$.

We reiterate that despite the wealth of results on ALM for convex minimization, to our best knowledge, this is the first existing ALM-type algorithm for multi-block affinely constrained min-max problems. We make use of the bilinear structure of problem (4) by incorporating the novel consensus term $\|B_i x_i - \nabla_{z_i}\rho^*(z)\|_2^2$ into the augmented Lagrangian, which guarantees sublinear convergence without introducing significant communication overhead. As a direct corollary, Theorem 1 implies both the duality gap and the constraint violation decrease at a sublinear rate of $O(1/t)$.

4 Concrete Algorithmic Realization for Solving Teamwork RL

Inspired by the multi-block optimization analysis in Section 3, we now develop the concrete Algorithm 1 for maximizing a team's joint utility via coordinating agents. See Figure 2 for a diagram that illustrates how the algorithm works. The full algorithm involves the following information sharing:

Communication: All communications take place between a local agent and the team lead. Only information about connecting states and global utilities are to be communicated. Specifically, in each iteration the local agent $i \in [N]$ passes the following to the team lead:
- $\{\mu(s, u) | s \in O_i, u\}, \nu_{\text{in},i}$: boundary dynamics of state block $\mathcal{S}_i$ that should match team lead's workflow
- $g_i(\mu_i) = C_i\mu_i$: local contribution to team utility

And the team lead propagates the following information back to each agent i:
- $\nu_{\text{out},i}, \xi_i$: workflow arrangement on the connecting states of block $\mathcal{S}_i$ that should match local dynamics
- $z_i, \nabla_{z_i}\rho^*(z), \lambda_i, \beta_i$: dual variables that reshape and regularize the agent i's' local reward

Local privacy: Each local agent $i \in [N]$ never needs to reveal its private reward or local policy. We emphasize that the following private information is never shared:

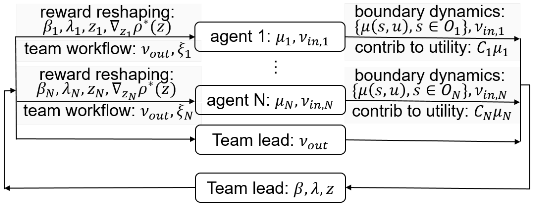

reward reshaping:
$\beta_1, \lambda_1, z_1, \nabla_{z_1}\rho^*(z)$
team workflow: ν_{out}, ξ_1
reward reshaping:
$\beta_N, \lambda_N, z_N, \nabla_{z_N}\rho^*(z)$
team workflow: ν_{out}, ξ_N
agent 1: $\mu_1, \nu_{in,1}$
agent N: $\mu_N, \nu_{in,N}$
Team lead: ν_{out}
boundary dynamics:
$\{\mu(s,u), s \in O_1\}, \nu_{in,1}$
contrib to utility: $C_1\mu_1$
boundary dynamics:
$\{\mu(s,u), s \in O_N\}, \nu_{in,N}$
contrib to utility: $C_N\mu_N$
Team lead: β, λ, z

Figure 2: Diagram of Algorithm 1. At each iteration, agents update their local policy and send out the boundary dynamics and local contribution to the team utility; team lead receives the local information and uses it to coordinates the agents by arranging an appropriate team workflow graph and reshaping local utilities.

Algorithm 1 Teamwork RL with joint team utility

```
Initialize: Individual agents' and team leader's variables,
            parameter $\eta_1, \eta_2, \alpha$.
for $k = 1, 2, \cdots, t$ do
  for $i = 1, 2, \cdots, N+1$ perform following in parallel:
    if $i \in [N]$:
      The i-th agent updates its local dynamics
      $\{\mu_i^{(k)},\ \nu_{\text{in},i}^{(k)}\}$ by solving subproblem (5) and passes
      the following information to the team leader:
      • dynamics on its connecting states:
      $\{\mu^{(k)}(s,u) | s \in O_i, u\}$ and $\nu_{\text{in},i}^{(k)}$;
      • individual contribution to team utility $C_i \mu_i^{(k)}$.
    if $i == N+1$:
      The team leader updates copied connection
      dynamics $\{\nu_{\text{out}}^{(k)}, \xi^{(k)}\}$ by (6).
  The team leader updates the dual variables and passes the following to agent $i \in [N]$:
  • shadow reward $z$ (also pass $\nabla_{z_i}\rho^*(z^{(k)})$):
    $z_i^{(k)} = z_i^{(k-1)} + \eta_2(\nabla_{z_i}\rho^*(z^{(k-1)}) - C_i\mu_i^{(k)})$
  • connection dynamics $\lambda, \beta$:
    $\beta_i^{(k)} = \beta_i^{(k-1)} - \eta(\xi_i^{(k)} - \nu_{\text{in},i}^{(k)})$
    $\lambda^{(k)}(s,u) = \lambda^{(k-1)}(s,u) - \eta(\nu_{\text{out}}^{(k)}(s,u) - \mu^{(k)}(s,u))$
  • copied connection dynamics $\{\nu_{\text{out}}^{(k)}(s,u) | s \in O_i\}$, $\xi_i^{(k)}$;
end for
for $i = 1, 2, \cdots, N$ do
  The $i$-th agent calculates local policies by
    $\pi_i^{(t)}(a|s) = \frac{\mu^{(t)}(s,a)}{\sum_{a \in \mathcal{A}} \mu^{(t)}(s,a)}, \forall s \in \mathcal{S}_i, a \in \mathcal{A}$.
end for
Output: Individual policies $\{\pi_i^{(t)}\}_{i=1}^N$.
```

• $\{P(s'|s,a), s, s' \in \mathcal{S}_i, a\}$: information related to internal transition probability within state block $\mathcal{S}_i$

• $\{\mu(s,a), s \in \mathcal{S}_i \backslash O_i, a\}$: information related to transition dynamics within the internal states of block $\mathcal{S}_i$

Local agent's subproblem after reward reshaping The primal update $\{\mu_i^{(k+1)}, \nu_{\text{in},i}^{(k+1)}\}$ for agent $i \in [N]$ can be formulated as the solution of the following quadratic subproblem defined on block $\mathcal{S}_i$:

$$\begin{aligned} \underset{\mu_i \geq 0,\ \nu_{\text{in},i}}{\text{argmax}}\ & r_i^{(k)\top}\mu_i - \beta_i^{(k)\top}\nu_{\text{in},i} + h_i^{(k)}(\mu_i, \nu_{\text{in},i}) \\ \text{s.t.}\ & \sum_{a \in A}\mu(s,a) = \gamma \sum_{s' \in S_i, a' \in A} \mu(s',a')P(s|s',a') + \gamma \mathbb{1}_{\{s \in I_i\}}\nu_{\text{in}}(s) + \Xi(s)\ \forall s \in \mathcal{S}_i \end{aligned} \quad (5)$$

where $r_i^{(k)}(s,a) = C_i(:,s,a)^\top z_i^{(k)} - \mathbb{1}_{\{s\in O_i\}}\lambda^{(k)}(s,a)$ for any $s \in \mathcal{S}_i, a \in \mathcal{A}$ (for outgoing states, a is replaced by u),

$$\begin{aligned} h_i^{(k)}(\mu_i, \nu_{\text{in},i}) = {} & \eta_1 \sum_{s\in O_i,u} (\nu_{\text{out}}^{(k)}(s,u) - \mu^{(k)}(s,u))\mu(s,u) + \eta_1 \sum_{s\in I_i} (\xi_i^{(k)}(s) - \nu_{\text{in}}^{(k)}(s))\nu_{\text{in}}(s) \\ & - \frac{\eta_2}{2}\|C_i\mu_i - \nabla_{z_i}\rho^*(z^{(k)})\|_2^2 - \frac{\alpha}{2}\sum_{s\in I_i}(\nu_{\text{in}}(s) - \nu_{\text{in}}^{(k)}(s))^2 - \frac{\alpha}{2}\sum_{\mathcal{S}_i,\mathcal{A}}(\mu(s,a) - \mu^{(k)}(s,a))^2 \end{aligned}$$

Here $\xi_i^{(k)}(s) = \sum_{s'\in O_{-i},u'} \nu_{\text{out}}^{(k)}(s',u')T(s|s',u')$ is the incoming measure for block i given by the team lead. We emphasize that despite the local problem being presented in a quadratic form, it is equivalent to a local regularized MDP as shown in Section 4.1, indicating we can view it as a planning or a learning problem depending on the situation and solve the local problem via any viable methods.

Team Leader's Problem for Coordination The primal update $\nu_{\text{out}}^{(k+1)}$ for the team leader is the solution of an unconstrained quadratic program:

$$\begin{aligned} & \underset{\nu_{\text{out}}}{\operatorname{argmax}} -\frac{\alpha}{2}\|\nu_{\text{out}} - \nu_{\text{out}}^{(k)}\|_2^2 + \sum_{i\in[N]}\sum_{s\in O_i,u} \lambda^{(k)}(s,u)\nu_{\text{out}}(s,u) + \sum_{i\in[N]}\sum_{s\in I_i}\beta^{(k)}(s)\xi_i(s) \\ & - \eta_1 \sum_{i\in[N]}\sum_{s\in I_i}(\xi_i^{(k)}(s) - \nu_{\text{in}}^{(k)}(s))\xi_i(s) - \eta_1 \sum_{i\in[N]}\sum_{s\in O_i,u}(\nu_{\text{out}}^{(k)}(s,u) - \mu^{(k)}(s,u))\nu_{\text{out}}(s,u) \qquad (6) \end{aligned}$$

where $\xi_i(s) = \sum_{s'\in O_{-i},u'} \nu_{\text{out}}(s',u')T(s|s',u')$ denotes the incoming measure for block i.

Following the result of Theorem 1, we obtain the following sublinear convergence for Algorithm 1.

Theorem 2. *Assume the occupancy measure under optimal policy π^* has full support on all states and the utility function ρ is Lipschitz. By choosing appropriate parameters according to Theorem 1, the output policy $\pi^{(t)}$ of Algorithm 1 satisfies $R(\pi^*) - R(\pi^{(t)}) = O(1/t)$.*

4.1 Local Subproblem Reformulation – Local Regularized MDP

Now we discuss how to interpret the subproblem (5) of an individual agent as a private regularized MDP defined on its own state blocks. Such reformulation allows us to go beyond the planning setting to the learning setting where the transition probability within each block are unknown. Consider the quadratic subproblems (5) of individual agents $i \in [N]$ in the primal updates of the joint utility. We can rewrite them into the following abstract form defined on state block $\mathcal{S}_i$:

$$\begin{aligned} \{\mu_i^{(k+1)},\, \nu_{\text{in},i}^{(k+1)}\} = {} & \underset{\mu_i\geq 0,\nu_{\text{in},i}}{\operatorname{argmax}} \;\; F_i(\mu_i, \nu_{\text{in},i}) \qquad (7) \\ \text{s.t.} \sum_{a\in\mathcal{A}} \mu(s,a) = {} & \gamma \sum_{s'\in\mathcal{S}_i,a'\in\mathcal{A}} \mu(s',a')P(s|s',a') + \gamma\mathbb{1}\{s\in I_i\}\nu_{\text{in}}(s) + \Xi(s)\ \forall s\in\mathcal{S}_i \end{aligned}$$

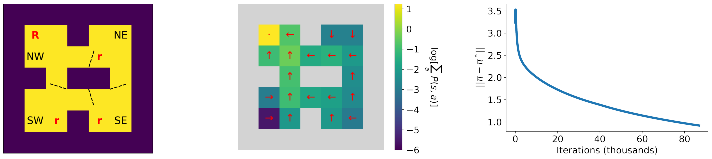
where function F_i is a concave quadratic objective. We emphasize that the above problem is purely defined on state block $\mathcal{S}_i$, with constraints similar to the MDP constraints. Hence, by incorporating a virtual starting state s_{start} and ending state s_{end}, we can construct an equivalent private MDP with quadratic regularization (see Appendix F). Each agent can solve their respective subproblem in parallel with the other agents by solving a **private** MDP with **quadratic regularization** defined on its own state block, without disclosing much private information. Such local MDPs can be handled by any viable methods such as the variational policy gradient (VPG) method or a Frank-Wolfe type algorithm in a learning scenario. For the completeness, we present solving via VPG in Appendix G.

5 Experiments

Here, we empirically support our theoretical findings by demonstrating our algorithm on a state-partitioned MDP. All experiments were run with a single CPU each for training and evaluation. The agent and leader updates require solving the quadratic programs defined in equations (5) and (6) respectively; both were solved using `cvxopt` Andersen *et al.* (2021).

The four-room MDP we consider is illustrated in Figure 3a. Each agent controls one of the four rooms (delineated by the doors), each containing a reward. The 20 yellow states are implicitly numbered sequentially left to right, top to bottom. This information is useful for interpreting the time series heatmaps. For example, row 1 in Figure 4d corresponds to the top left cell with the big "R" and row 8 corresponds to the small "r" in the top right room; columns correspond to actions and are listed below the figure; each heatmap is titled with the training iteration at which it was recorded. In all the heatmaps, the colorbar label indicates the quantity measured. Likewise, the heatmaps overlayed on the gridworld show the log of the state visitation measure with arrows indicating the most likely action taken by the final policy (a *dot* indicates the "Stay" action). The stimulus plots reveal how the coordinator incentivizes collaboration in the constrained and nonlinear utility settings. We assume a uniform initial state distribution.

If each agent optimizes for their own reward, they will learn policies that seek the reward in their respective room. In contrast, a globally optimal policy seeks the largest reward in the top left room, and this is the policy found by our algorithm (Figures 3b and 3c).



(a) (b) (c)

Figure 3: (3a) Each of the four distinct rooms, partitioned by the dotted lines (doors), is controlled by a single agent. One room contains a big reward (R=10), while the other rooms contain a small reward (R=1). (3b) Algorithm 1 recovers the optimal policy. (3c) Convergence to the optimal policy.

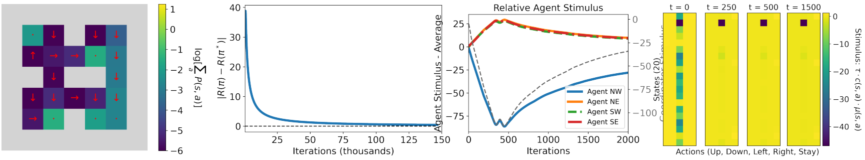
To demonstrate the effect of a global constraint, we limit the amount of mass that the occupancy measure can place on "left" actions in Figure 4. A linear constraint is added to the MDP to enforce that the total occupancy measure associated with the "left" action is one thousandth the total measure given by all other actions combined. This makes it unlikely for the left action to be taken by the policy. Also notice in Figure 4a that the brightest states are those with reward, while the next brightest are those in the rightmost column. The occupancy measure in these states is moderately high because there is no way to reach a reward state without moving left; therefore, the learned policy remains somewhat arbitrary in these states (despite the uniformly downward arrows in the column).

To demonstrate the generalizability of our algorithm beyond linear reward functions (average/discounted reward), we substitute a convex KL penalty for deviating from a uniform occupancy measure in Figure 5. This reward encourages joint team exploration of the MDP and is added as a regularizer to the linear reward function. Contrasting Figure 5a with Figure 3b reveals the effect of the KL penalty on encouraging a more uniform occupancy measure.

In Figures 4 and 5 (c-d), the coordinator penalizes agent 1 initially to discourage it from collecting the big reward. This is necessary, as a policy strongly biased towards the big reward is antithetical to both the "no left" constraint and KL-induced exploration objective. In Figure 4d, the coordinator penalizes agent NW from taking a left action in the cell immediately right of the big reward. In Figure 5d, the coordinator initially penalizes agent NW choosing to "Stay" seated on the big reward, however, once a more uniform occupancy measure has been induced, the coordinator adjusts its stimulus to spread out the occupancy measure elsewhere.

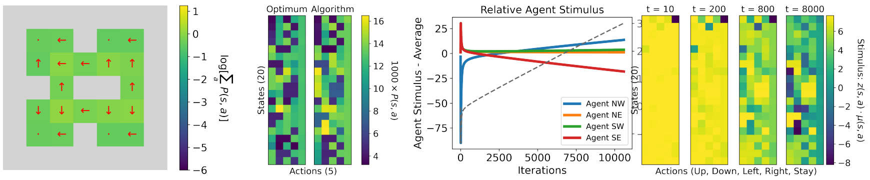
6 Conclusion and Discussion

In this work, we propose the RL problem of maximizing a nonlinear utility function in an MDP where the state space admits a natural partitioning. By leveraging the duality of the problem, we are led to



(a) (b) (c) (d)

Figure 4: Constrained MDP. (4a) Recovery of optimal policy. (4b) Convergence. (4c) Distribution of coordinator stimulus to agents. (4d) Coordinator discourages agent NW from collecting +10.



(a) (b) (c) (d)

Figure 5: KL Penalty. (5a) Recovery of optimal policy. (5b) Comparison to optimal occupancy. (5c) Distribution of coordinator stimulus to agents. (5d) Agent NW penalized for sitting on $+10$.

formulate its solution as a distributed multi-agent algorithm. A single leader efficiently coordinates between the agent owned partitions, injecting stimulus where needed to encourage collaboration and ensuring minimal leakage of private agent information across boundaries. Our proposed algorithm fits into both planning and learning scenario and guarantees an $\mathcal{O}(1/T)$ convergence rate.

REFERENCES

Jacob Abernethy, Kevin A Lai, and Andre Wibisono. Last-iterate convergence rates for min-max optimization. *arXiv preprint arXiv:1906.02027*, 2019.

M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A python package for convex optimization, 2021.

Dimitri P Bertsekas, WW Hager, and OL Mangasarian. *Nonlinear programming*. Athena Scientific Belmont, MA, 1998.

Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *TARK*, volume 96, pages 195–210. Citeseer, 1996.

Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

Qiang Cheng, Qiang Liu, Feng Chen, and Alexander T Ihler. Variational planning for graph-based MDPs. *Advances in Neural Information Processing Systems*, 26:2976–2984, 2013.

Jongeun Choi, Songhwai Oh, and Roberto Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802–2814, 2009.

Jerzy A Filar, Lodewijk CM Kallenberg, and Huey-Miin Lee. Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1):147–161, 1989.

Xiang Gao, Yang-Yang Xu, and Shu-Zhong Zhang. Randomized primal–dual proximal block coordinate updates. *Journal of the Operations Research Society of China*, 7(2):205–250, 2019.

Carlos E Guestrin and Geoffrey Gordon. Distributed planning in hierarchical factored MDPs. *arXiv preprint arXiv:1301.0571*, 2012.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

Bingsheng He and Xiaoming Yuan. On the acceleration of augmented lagrangian method for linearly constrained optimization. *Optimization online*, 3, 2010.

Junling Hu and Michael P Wellman. Nash Q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Ying Huang and Lodewijk CM Kallenberg. On finding optimal policies for Markov decision chains: a unifying framework for mean-variance-tradeoffs. *Mathematics of Operations Research*, 19(2):434–448, 1994.

Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang. An efficient augmented lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3):507–530, 2013.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, 8(10), 2007.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. *arXiv preprint arXiv:1705.10461*, 2017.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer, 1999.

Yi Tian, Jian Qian, and Suvrit Sra. Towards minimax optimal reinforcement learning in factored Markov decision processes. *Advances in Neural Information Processing Systems*, 33, 2020.

Yue Xu, Zengde Deng, Mengdi Wang, Wenjun Xu, Anthony Man-Cho So, and Shuguang Cui. Voting-based multi-agent reinforcement learning for intelligent IoT. *IEEE Internet of Things Journal*, 2020.

Yangyang Xu. Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM Journal on Optimization*, 27(3):1459–1484, 2017.

Bang Xiang Yong and Alexandra Brintrup. Multi agent system for machine learning under uncertainty in cyber physical manufacturing system. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 244–257. Springer, 2019.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. *arXiv preprint arXiv:2007.02151*, 2020.

Appendix

Roadmap: We provide the hyperparameters of our experiments in Appendix A and additional experiments in Appendix B. We discuss the related work in Appendix C. In Appendix D, we discuss the regime of teamwork RL with shared cost constraints. In Appendix E, we provide a detailed discussion of how to abstract the teamwork RL with joint utility problem as a multi-agent min-max optimization problem with a bilinear coupling term. In Appendix F we formulate the agents' private regularized MDPs and in appendix G, we show how to solve them by variational policy gradient methods.

A Hyperparameters

Hyperparamters for each case. We set $\gamma = 0.95$ for all experiments.

Figure 3:

- $\eta_1 = 0.5$ chosen from $[0.5, 0.75]$.
- $\alpha = 1.0$ chosen from $[1.0, 2.0]$.

Figure 4:

- $\eta_1 = 0.5$ taken from unconstrained setting above.
- $\alpha = 1.0$ taken from unconstrained setting above.
- $c(s,a) = \begin{bmatrix} -.001 & -.001 & 1 & -.001 & -.001 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -.001 & -.001 & 1 & -.001 & -.001 \end{bmatrix} \in \mathbb{R}^{20\times5}$ (see equation (10) in Appx. D).

Figure 5:

- $\eta_1 = 0.1$ chosen from $[0.1, 0.5, 0.8]$.
- $\eta_2 = 0.1$ chosen from $[0.01, 0.1, 1.0]$.
- $\alpha = 1.0$ chosen from $[0.5, 1.0, 2.0]$.
- KL coefficient: 1000.

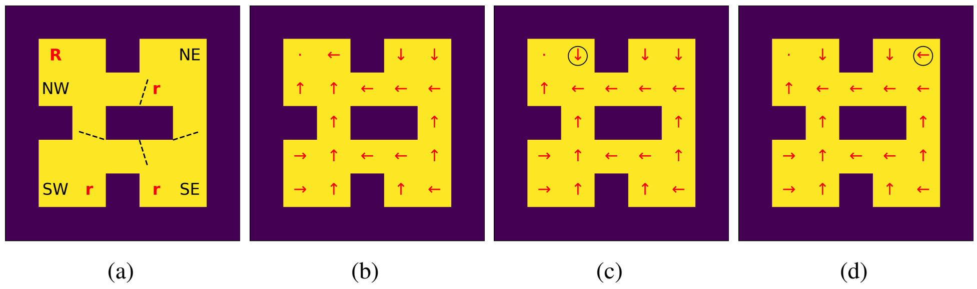
B Additional Experiments

Figure 6 examines three settings in sequence. We first confirm that our proposed algorithm recovers the globally optimal solution when neither the constraint nor the general, nonlinear ρ term is present. Next, we incorporate a minimal constraint and demonstrate the algorithm adjusts its learned occupancy measure to satisfy that constraint. Finally, we introduce the nonlinear $\rho = D_{KL}(\mu||\mu^*)$ where μ^* is given as the solution to the previous constrained problem, but with the measures for the “down" and “left" actions in a state (circled) swapped. In all cases, the algorithm recovers the globally optimal policy.

C Related Work

Augmented Lagrangian Method The Augmented Lagrangian Method (ALM), an algorithmic framework we leverage to handle the teamwork RL problem in this work, has been studied extensively in the literature, see e.g. Bertsekas *et al.* (1998); Nocedal and Wright (2006). Specifically, for an affinely constrained convex optimization problem, its convergence and complexity are provided under various situations He and Yuan (2010); Li *et al.* (2013); Xu (2017). When the decision variable consist of multiple blocks, ALM is naturally extended to the Alternating Direction Method of Multipliers—see Boyd *et al.* (2011) for a comprehensive overview of the method. Despite the rich body of theory on ALM for convex minimization, to the best of our knowledge, there is no existing ALM-type algorithm for the *affinely constrained* multi-block min-max problem.

Multi-agent Reinforcement Learning Multi-agent Reinforcement Learning (MARL) also considers a sequential decision-making problem with more than one agent involved. Common MARL frameworks include Markov games, also known as stochastic games, where multiple agents make decisions



(a) (b) (c) (d)

Figure 6: (6a) Each of the four distinct rooms, partitioned by the dotted lines (doors), is controlled by a single agent. One room contains a big reward (R=10) while the other rooms contain a small reward (R=1). (6b) The policy seeks out the largest reward in the top left room. This is the globally optimal team policy to maximize welfare. (6c) A linear constraint is enforced on the occupancy measure in the circled state to ensure the “Down" action contains 10x the mass of the other actions combined. (6d) A KL-penalty is added to the linear reward to penalize deviating from the optimal policy found in (6c), but with the measures for the “down" and “left" actions in a state (circled) swapped.

and earn individual rewards simultaneously in a public environment but possibly with a cooperative overarching goal Littman (1994); Boutilier (1996); Hu and Wellman (2003), and extensive-form games where agents are largely competitive, make decisions based on possibly incomplete information, and the primary representation is a game tree rather than a state-space Osborne and Rubinstein (1994). Our proposed teamwork RL is most similar to the Markov game setting. It differs in two aspects: 1) we consider a state-decomposition setting where each agent only has access to its own part of the state space; 2) we consider general non-linear team utilities while cooperative MARL only considers objectives based on individual agents’ cumulative rewards.

RL with Concave Utility In this work we consider reinforcement learning problems with general concave utilities beyond cumulative rewards. Early works Filar *et al.* (1989); Huang and Kallenberg (1994) consider concave utilities for tabular settings using dynamic programming approaches, but are not scalable to large problems. The work by Hazan *et al.* (2019) considers maximizing the entropy of a state visitation measure, a special case of the type of concave utilities we study, by proposing a model-based iterative policy update, which requires explicit knowledge of the environment’s transition probabilities. The variational policy gradient method developed by Zhang *et al.* (2020) is a model-free approach with parameterization over policies that handles general concave utilities.

State-Decomposition Setting In this work we consider a setting where the state set for a given MDP admits a natural partitioning into several disjoint state blocks, (e.g., in the task of assembling a product, or execution of a kitchen recipe), each handled by a block-specific (local) agent. Constructing a specific partitioning can be accomplished with spectral clustering and has previously been studied in MDPs in the context of proto-value functions Mahadevan and Maggioni (2007). Our setting is different from the decomposition settings discussed in the factored MDP or graph-based MDP literature Guestrin and Gordon (2012); Tian *et al.* (2020); Cheng *et al.* (2013), where the state space is factored in the form of a Cartesian product rather than a strict union.

D Teamwork RL with shared cost constraints

In this work, we consider the following two interesting scenarios within our proposed teamwork RL model, namely teamwork RL with a joint team utility and teamwork RL with shared cost constraints. We focus on the joint team utility scenario in the main text and discuss the shared cost constraints scenario here in the appendix. As a reminder, we give out the formulation of both scenarios first.

Teamwork RL with joint team utility: We consider the scenario where the individuals work together to maximize a general joint team utility:

$$\max_{\pi=\{\pi_i\}} R(\pi) = \rho(g_1(\pi_1;\pi_{-1}),\cdots,g_N(\pi_N;\pi_{-N})) \tag{8}$$

Here the mapping $g_i(\pi_i;\pi_{-i})$ denotes the contribution of the i-th agent to the team utility, which is defined to be a *linear* functional of the state-action occupancy measure restricted to block $\mathcal{S}_i$, and ρ is some general smooth concave utility coupling across agents. We remark such scenario indeed captures any smooth concave utility when we set mapping g_i to be identity. We are especially interested in the case where g_i's output low-dimensional vectors such that little private information needs to be revealed in the joint team utility.

Teamwork RL with shared cost constraints: We consider the task of maximizing the total expected discounted reward under a shared cost constraint across agents. In particular, given some cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we assume the total expected discounted cost incurred by the chosen policy is constrained from above:

$$\begin{aligned} \max_{\pi=\{\pi_i\}_{i=1}^N} R(\pi) = \sum_{i\in[N]} V_i(\pi_i; \pi_{-i}) \\ \text{s.t. } V_c(\pi) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \middle| \pi, s_0 \sim \Xi\right] \leq 0 \end{aligned} \tag{9}$$

where V_i denotes the local cumulative return of the i-th agent, defined as

$$V_i(\pi_i; \pi_{-i}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_i(s_t, a_t) \middle| \pi, s_0 \sim \Xi\right]$$

for some local reward function r_i.

In this section, we consider the teamwork RL with shared cost constraints scenario (9), to continue discussing how to use a simpler (existing) ALM type of algorithm to address it.

Without lose of generality, we consider the scenario where all agents act to maximize their cumulative returns subject to a shared cost constraint:

$$\begin{aligned} \max_{\mu\geq 0} & \sum_{s\in\mathcal{S}, a\in\mathcal{A}} r(s,a)\mu(s,a) \\ \text{s.t. } & \sum_{a} \mu(s,a) = \gamma \sum_{s',a'} \mu(s',a')P(s|s',a') + \Xi(s) \;\; \forall s \\ & \sum_{s\in\mathcal{S}, a\in\mathcal{A}} c(s,a)\mu(s,a) \leq 0 \end{aligned} \tag{10}$$

where $r(s, a)$ is the given reward function and $c(s, a)$ is the given cost function. This is a much simpler case of the general utility ρ in problem (2), and we provide a simpler alternative optimization method with primal-only updates and does not involve the Fenchel dual.

We follow the same approach as the joint utility case without introducing the min-max formulation due to the linearity of the objective. To handle the new cost constraint, we introduce a nonnegative dummy variable $\nu_{\text{cost}} \in \mathbb{R}_{\geq 0}$ and corresponding shadow price $\tau \in \mathbb{R}$, both handled by the team leader, to coordinate the agents through reshaping the local rewards. Thus, problem (10) can be reformulated as:

$$\begin{aligned} \max_{\mu\geq 0, \nu_{\text{cost}}\geq 0, \nu_{\text{in}}, \nu_{\text{out}}} & \sum_{s\in\mathcal{S}, a\in\mathcal{A}} r(s,a)\mu(s,a) \\ \text{s.t. } & \sum_{a\in\mathcal{A}} \mu(s,a) = \gamma \sum_{s'\in\mathcal{S}_i, a'\in\mathcal{A}} \mu(s',a')P(s|s',a') + \gamma \mathbb{1}_{\{s\in I_i\}}\nu_{\text{in}}(s) + \Xi(s) \;\; \forall s \in \mathcal{S}_i, \forall i \in [N] \\ & \nu_{\text{in}}(s) = \sum_{s'\in O_{-i}} \nu_{\text{out}}(s',u)P(s|s',u) \;\; \forall s \in I_i, \forall i \\ & \nu_{\text{out}}(s,u) = \mu(s,u) \quad \forall s \in O_i, i \in [N], u \\ & \sum_{s\in\mathcal{S}, a\in\mathcal{A}} c(s,a)\mu(s,a) + \nu_{\text{cost}} = 0 \end{aligned}$$

which can be abstracted as the following multi-agent convex minimization (linear programming) problem:

$$\begin{aligned} \min_{x \in \mathcal{X}} \; f(x) &= \sum_{i=1}^{N+1} f_i(x_i) \\ \text{s.t. } Ax &= \sum_{i=1}^{N+1} A_i x_i = b \end{aligned} \tag{11}$$

where the linear functions f_i denotes the cumulative reward of agent i and we include the shared cost constraint in $Ax = b$.

To handle the multi-agent optimization problem (11), we apply the primal-dual proximal block coordinate update (PDBCU) approach proposed in the work Gao *et al.* (2019). We introduce the following augmented Lagrangian:

$$\mathcal{L}_\eta(x, \lambda) = \sum_{i=1}^{N} f_i(x_i) - \lambda^\top (Ax - b) + \frac{\eta}{2}\|Ax - b\|_2^2$$

The alternating iterative scheme of the PDBCU can be outlined as

$$\begin{cases} \text{primal: } x^{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \mathcal{L}_\eta(x, \lambda^k) + \dfrac{1}{2}\|x - x^k\|_Q^2 \\ \text{dual: } \lambda^{k+1} = \lambda^k - \eta(Ax^{k+1} - b) \end{cases}$$

Again, by picking the predetermined proximal term $Q = \alpha I - \eta A^\top A \succeq 0$ with appropriate $\alpha > 0$, the primal update can be implemented by individual agents in parallel.

A Sub-linear Convergence guarantee As a corollary of Theorem 3.6 of Gao *et al.* (2019), we have the following sub-linear convergence result of the resulting method:

Theorem 3. *Assume there exists an optimal primal-dual solution* (x^*, λ^*) *to problem* (11)*. Choosing* $\eta > 0$ *and* $\alpha \geq \eta_1 \|A\|_2^2$*, we have:*

$$\max\{|f(\hat{x}^t) - f(x^*)|, \|A\hat{x}^t - b\|_2\} \leq \frac{1}{1+t}\left[\frac{1}{2}\|x^0 - x^*\|_Q^2 + \frac{\max\{(1 + \|\lambda^*\|)^2, 4\|\lambda^*\|_2^2\}}{2\alpha}\right]$$

where $\hat{x}^t = \frac{1}{1+t}\sum_{k=1}^{t+1} x^k$ *denotes the average of iterates.*

Now we present the concrete realization corresponding to the abstract PDBCU updates in Algorithm 2. Similar to the discussion of joint utility case in Section 4, the full algorithm of the shared cost constraints case involves the following participants and operations (for the completeness and the ease of reading, we repeat the same operations showing up in the joint utility case. Only those colored in blue are unique for the shared cost constraints case):

Team lead: The team lead is given the transition probability between state blocks and updates:
- τ: shadow price corresponds to the cost constraint
- λ, β: transition dynamics on connecting states corresponding to the boundary conditions
- ν_{out}: boundary dynamics on the outgoing states which help "reshape" agents' local rewards.

Local agent: Each local agent may have a private reward r_i and receives instructional information from the team lead. Based upon both local information and received information, the local agent will construct a regularized local MDP. The local agent solves the local MDP on its own, using any viable method.

Communication: All communications take place between a local agent and the team lead. Only information about connecting states and global utilities are to be communicated. Specifically, in each iteration the local agent $i \in [N]$ passes the following to the team lead:
- $\{\mu(s, u) | s \in O_i, u \in \mathcal{A}'\}, \nu_{\text{in},i}$: boundary dynamics of state block $\mathcal{S}_i$
- $\sum_{\mathcal{S}_i, \mathcal{A}} c(s, a)\mu^{(k)}(s, a)$: local sum of cost

And the team lead propagates the following information back to each agent i:
- $\nu_{\text{out},i}, \xi_i$: copied boundary dynamics of block $\mathcal{S}_i$
- τ, λ_i, β_i: dual variables that reshape and regularize the agent i's' local reward
- $\sum_{\mathcal{S}, \mathcal{A}} c(s, a)\mu^{(k)}(s, a)$: residual of cost

Local privacy: Each local agent $i \in [N]$ never needs to reveal its private reward or local policy. We emphasize that the following private information is never shared:

- $\{P(s'|s,a), s, s' \in \mathcal{S}_i, a \in \mathcal{A}\}$: internal transition probability within state block $\mathcal{S}_i$
- $\{\mu(s,a), s \in \mathcal{S}_i \backslash O_i, a \in \mathcal{A}\}$: transition dynamics within the internal states of block $\mathcal{S}_i$

Local agent only shares following the information to the team leader but not to other agents.

- $\{\mu(s,u) | s \in O_i, u\}, \nu_{\text{in},i}$: boundary dynamics of state block $\mathcal{S}_i$
- $\sum_{\mathcal{S}_i,\mathcal{A}} c(s,a)\mu^{(k)}(s,a)$: local sum of cost

Now, we present the primal updates, i.e., the subproblem each agent and team lead need to solve in each iteration. Again, we color the unique updates of the shared cost constraints case in blue.

Local Agent's Problem After Reward Shaping The primal update for agent $i \in [N]$ can be formulated as the following quadratic subproblem defined on block $\mathcal{S}_i$:

$$\begin{aligned}\{\mu_i^{(k+1)},\ \nu_{\text{in},i}^{(k+1)}\} = \underset{\mu_i \geq 0,\ \nu_{\text{in},i}}{\operatorname{argmax}}\ & r_i^{(k)\top}\mu_i - \beta_i^{(k)\top}\nu_{\text{in},i} + h_i^{(k)}(\mu_i, \nu_{\text{in},i}) \\ \text{s.t.} \sum_{a \in A}\mu(s,a) = \gamma & \sum_{s' \in S_i, a' \in A}\mu(s',a')P(s|s',a') + \gamma \mathbb{1}_{\{s \in I_i\}}\nu_{\text{in}}(s) + \Xi(s)\ \forall s \in \mathcal{S}_i\end{aligned} \tag{12}$$

where $r_i^{(k)}(s,a) = r(s,a) + c(s,a)\tau - \mathbb{1}_{\{s \in O_i\}}\lambda^{(k)}(s,a)$ for any $s \in \mathcal{S}_i, a \in \mathcal{A}$ and

$$\begin{aligned}h_i^{(k)}(\mu_i, \nu_{\text{in},i}) =& \eta_1 \sum_{s \in O_i, u}(\nu_{\text{out}}^{(k)}(s,u) - \mu^{(k)}(s,u))\mu(s,u) + \eta_1 \sum_{s \in I_i}(\xi_i^{(k)}(s) - \nu_{\text{in}}^{(k)}(s))\nu_{\text{in}}(s) \\ & -\eta(\sum_{s \in \mathcal{S}, a \in \mathcal{A}} c(s,a)\mu^{(k)}(s,a) + \mu_{\text{cost}}^{(k)})\sum_{\mathcal{S}_i,\mathcal{A}} c(s,a)\mu(s,a) - \frac{\alpha}{2}\sum_{s \in I_i}(\nu_{\text{in}}(s) - \nu_{\text{in}}^{(k)}(s))^2 \\ & - \frac{\alpha}{2}\sum_{\mathcal{S}_i,\mathcal{A}}(\mu(s,a) - \mu^{(k)}(s,a))^2\end{aligned}$$

Here $\xi_i^{(k)}(s) = \sum_{s' \in O_{-i}, u'} \nu_{\text{out}}^{(k)}(s',u')T(s|s',u')$ is the incoming measure for block i given by team lead.

Team Leader's Problem for Coordination The primal update for the team leader is a quadratic program:

$$\begin{aligned}\{\nu_{\text{cost}}^{(k+1)}, \nu_{\text{out}}^{(k+1)}\} = \underset{\nu_{\text{cost}} \geq 0, \nu_{\text{out}}}{\operatorname{argmax}} & -\frac{\alpha}{2}\|\nu_{\text{cost}} - \nu_{\text{cost}}^{(k)}\|_2^2 - \frac{\alpha}{2}\|\nu_{\text{out}} - \nu_{\text{out}}^{(k)}\|_2^2 + \sum_{i \in [N]}\sum_{s \in O_i}\lambda^{(k)}(s)\nu_{\text{out}}(s) \\ + \sum_{i \in [N]}\sum_{s \in I_i}\beta^{(k)}(s)\xi_i(s) + \tau^{(k)}\nu_{\text{cost}} & - \eta_1 \sum_{i \in [N]}\sum_{s \in O_i, u}(\nu_{\text{out}}^{(k)}(s,u) - \mu^{(k)}(s,u))\nu_{\text{out}}(s,u) \\ -\eta_1 \sum_{i \in [N]}\sum_{s \in I_i}(\xi_i^{(k)}(s) - \nu_{\text{in}}^{(k)}(s))\xi_i(s) & - \eta_1(\sum_{s \in \mathcal{S}, a \in \mathcal{A}} c(s,a)\mu^{(k)}(s,a) + \mu_{\text{cost}}^{(k)})\nu_{\text{cost}}\end{aligned} \tag{13}$$

where $\xi_i(s) = \sum_{s' \in O_{-i}, u'} \nu_{\text{out}}(s',u')T(s|s',u')$ denotes the incoming measure for block i.

Following the convergence result of Theorem 3, we have the following sublinear convergence rate:

Theorem 4. *Assume the occupancy measure under optimal policy π^* has full support on all states. By choosing appropriate parameters according to Theorem 3, the output policy $\pi^{(t)}$ of the proposed Algorithm 2 satisfy:*

$$R(\pi^*) - R(\pi^{(t)}) = O(\frac{1}{t}) \quad \textit{and} \quad V_c(\pi^{(t)}) = O(\frac{1}{t})$$

E TEAMWORK RL WITH JOINT TEAM UTILITY

In this section, we provide the details of abstracting problem (2) into a multi-agent min-max optimization problem with a bilinear coupling term. Note from the discussion in Section 3.1, we know

Algorithm 2 Teamwork RL with shared cost

```
**Initialize**: Individual agents and team leader's variables arbitrarily, parameter $\eta$, $\alpha$.
**for** $k = 1, 2, \cdots, t$ **do**
  **for** $i = 1, 2, \cdots, N+1$ perform following **in parallel**:
    **if** $i \in [N]$:
      The $i$-th agent updates its local dynamics $\{\mu_i^{(k)},\ \nu_{\text{in},i}^{(k)}\}$ by solving local subproblem (12)
  and
      passes following information to the team leader:
      • dynamics on its connecting states $\{\mu^{(k)}(s,u) | s \in O_i, u \in \mathcal{A}'\}$ and $\nu_{\text{in},i}^{(k)}$;
      • local sum of cost $\sum_{\mathcal{S}_i,\mathcal{A}} c(s,a)\mu^{(k)}(s,a)$.
    **if** $i == N+1$:
      Team leader updates copied connection dynamics and cost residual $\{\nu_{\text{out}}^{(k)}, \nu_{\text{cost}}^{(k)}\}$ by (13).
  Team leader updates the dual variables and passes the following to agent $i \in [N]$:
  • shadow price $\tau$: $\tau^{(k)} = \tau^{(k-1)} - \eta(\sum_{\mathcal{S},\mathcal{A}} c(s,a)\mu^{(k)}(s,a) + \nu_{\text{cost}}^{(k)})$
  • connection dynamics $\lambda, \beta$: $\beta_i^{(k)} = \beta_i^{(k-1)} - \eta(\xi_i^{(k)} - \nu_{\text{in},i}^{(k)}),\ \forall i \in [N]$;
                 $\lambda^{(k)}(s,u) = \lambda^{(k-1)}(s,u) - \eta(\nu_{\text{out}}^{(k)}(s,u) - \mu^{(k)}(s,u)),\ \forall s \in$
  $O_i, u \in \mathcal{A}'$
  • copied connection dynamics $\{\nu_{\text{out}}^{(k)}(s,u) | s \in O_i, u \in \mathcal{A}'\},\ \xi_i^{(k)}$;
  • residual of cost $\sum_{\mathcal{S},\mathcal{A}} c(s,a)\mu^{(k)}(s,a) + \nu_{\text{cost}}^{(k)}$.
**end for**
**for** $i = 1, 2, \cdots, N$ **do**
  The $i$-th agent calculates local policies by
      $$\pi_i^{(t)}(a|s) = \frac{\mu^{(t)}(s,a)}{\sum_{a\in\mathcal{A}} \mu^{(t)}(s,a)}, \forall s \in \mathcal{S}_i, a \in \mathcal{A}.$$
**end for**
**Output**: Individual policies $\{\pi_i^{(t)}\}_{i=1}^N$
```

problem (2) can be rewritten as

$$\begin{aligned}
\max_{\mu\geq 0,\nu_{\text{in}},\nu_{\text{out}}} \ \min_{z} \ & \sum_i \langle C_i\mu_i, z_i\rangle - \rho^*(z) \\
\text{s.t.} \ & \sum_{a\in\mathcal{A}} \mu(s,a) = \gamma \sum_{s'\in\mathcal{S}_i, a'\in\mathcal{A}} \mu(s',a')P(s|s',a') + \gamma \mathbb{1}_{\{s\in I_i\}}\nu_{\text{in}}(s) + \Xi(s) \quad \forall s \in \mathcal{S}_i, \forall i \in [N] \\
& \nu_{\text{in}}(s) = \sum_{s'\in O_{-i}} \nu_{\text{out}}(s')P(s|s') \quad \forall s \in I_i, \forall i \\
& \nu_{\text{out}}(s) = \sum_{a\in\mathcal{A}} \mu(s,a) \quad \forall s \in O_i, i \in [N]
\end{aligned}$$

Let the variable x_i for $i \in [N]$ represents the variable $\{\mu_i, \nu_{\text{in},i}\}$. Let the variable x_{N+1} represents ν_{out}. Define $x = [x_1^\top, \cdots, x_{N+1}^\top]^\top$. Note the first and the nonnegativity constraints are separable and only relates to $\{x_i, i \in [N]\}$. Define constraint set $\mathcal{X}_i$ to be the intersection of the first and the nonnegativity constraints with respect to x_i for $i \in [N]$. Let $\mathcal{X}_{N+1} = \mathbb{R}$ and $\mathcal{X} = \otimes_{i=1}^{N+1}\mathcal{X}_i$. We remark each $\mathcal{X}_i$ is closed and convex (but not bounded). The rest of constraints are linear with respect to x, which can be written as $Ax = b$ for some given constant matrix A and vector b. Lastly, the term $\langle C_i\mu_i, z_i\rangle$ in objective can be written as a bilinear term coupling x_i and z_i, which is denoted as $z_i^\top B_i x_i$ for some given constant matrix B_i. To summarize, we can abstract above problem as:

$$\min_{x\in\mathcal{X}} \max_{z} \ f(x,z) = \rho^*(z) - \sum_{i=1}^{N+1} z_i^\top B_i x_i$$

$$\text{s.t. } Ax = \sum_{i=1}^{N+1} A_i x_i = b$$

F Private MDPs

Theorem 5 (Equivalence between local subproblems and quadratic MDPs)**.** *The subproblem* (7) *of agent $i \in [N]$ is equivalent to an MDP $\mathcal{M}' = \{\mathcal{S}_i', \mathcal{A}_i', P_i', \gamma, \Xi'\}$ with utility $F_i'(\cdot)$, where*

$$\textit{state space: } \mathcal{S}_i' = \mathcal{S}_i \sqcup \{s_{\textit{start}}, s_{\textit{end}}\},$$

$$\textit{action space: } \mathcal{A}_i' = \begin{cases} \mathcal{A} & \textit{if } s \neq s_{\textit{start}} \\ I_i \sqcup s_{\textit{end}} & \textit{if } s = s_{\textit{start}} \end{cases},$$

$$\textit{initial distribution: } \forall s \in \mathcal{S}_i$$

$$\Xi_i'(s) = \begin{cases} \gamma(1-\gamma)\Xi_i(s) & \textit{if } s \in \mathcal{S}_i \\ 1 - \sum_{s \in S_i} \gamma(1-\gamma)\Xi_i(s) & \textit{if } s = s_{\textit{start}} \\ 0 & \textit{if } s = s_{\textit{end}} \end{cases},$$

$$\textit{transition probability: } \forall s \in \mathcal{S}_i'$$

$$P_i'(s'|s,a) = \begin{cases} P(s'|s,a) & \textit{if } s' \in S_i \\ 1 - \sum_{s' \in \mathcal{S}_i} P(s'|s,a) & \textit{if } s' = s_{\textit{end}} \\ 0 & \textit{if } s' = s_{\textit{start}} \end{cases},$$

$$P_i'(s'|s_{\textit{start}}, a) = \mathbb{1}\{s' = a\},\ P_i'(s'|s_{\textit{end}}, a) = \mathbb{1}\{s' = s_{\textit{end}}\}.$$

Denote the corresponding occupancy measure as $\mu_i' = \{\mu'(s,a), s \in \mathcal{S}_i', a \in \mathcal{A}_i'\}$. Then the quadratic utility can be written as

$$F_i'(\mu_i') = F_i(\{\frac{1}{\gamma(1-\gamma)}\mu'(s,a) | s \in \mathcal{S}_i, a \in \mathcal{A}\}, \\ \{\frac{1}{\gamma(1-\gamma)}\mu'(s_{\textit{start}}, a) | a \in I_i\}),$$

where function F_i is the original local objective in Eq. (7)*.*

G Variational Policy Gradient for Quadratic MDP

Theorem 5 reduces the individual's subproblem to a quadratic MDP problem. We remark that such result also holds for the share cost constraints case. Consider solving a general quadratic MDP $\mathcal{M}'$ discussed in Lemma 5. We note that the Bellman equation and dynamic programming all fail in this case due to the loss of additivity in the utility. Therefore, we apply the variational policy gradient method proposed in Zhang *et al.* (2020), which established the parametrized policy gradient for RL with general utilities.

Parameterizing the policy by $\pi = \pi(\theta), \theta \in \Theta$, we can write a quadratic MDP problem with corresponding parameterized utility measure $\mu = \mu(\pi(\theta)) = \mu(\theta)$ as:

$$\max_{\theta \in \Theta} R(\pi(\theta)) = \max_{\theta \in \Theta} F(\mu(\theta)) = \max_{\theta \in \Theta} b^\top \mu(\theta) - \frac{1}{2}\mu(\theta)^\top A \mu(\theta)$$

for some quadratic utility $F(\cdot)$ with $A = [a_1^\top; \cdots, ; a_n^\top] \succeq 0$. Then we have the following lemma for the variational policy gradient.

Lemma 1 (Variational policy gradient for Quadratic MDP)**.** *Denote $v(\theta; z)$ as the cumulative value of policy π_θ with reward function z, and assume $\nabla_\theta V(\theta; z)$ always exists. Then we have*

$$\nabla_\theta R(\pi(\theta)) = \nabla_\theta V(\theta; b) - \nabla_\theta V(\theta; V(\theta; A)). \tag{14}$$

Here $V(\theta; A)$ denotes the vector $[V(\theta; a_1), \cdots, V(\theta; a_n)]$.

Theorem 4.4 of work Zhang *et al.* (2020) established a sublinear convergence guarantee of the policy gradient ascent method using policy gradient shown in Lemma 1.

Lastly, we remark that the policy gradient (14) can be calculated via $V(\theta; z)$ and $\nabla V(\theta; z)$ for any given reward function z, which can be estimated given n i.i.d. episodes of sample trajectories with length K following π_θ, denoted as $\zeta_i = \{s_k^{(i)}, a_k^{(i)}\}_{k=1}^K$. Then according to the classic Policy Gradient Theorem Sutton *et al.* (1999), we can estimate $V(\theta; z)$ and $\nabla V(\theta; z)$ for any function z via

$$V(\theta; z) \approx \frac{1}{n}\sum_{i=1}^{n} V(\theta; z; \zeta_i) := \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} \gamma^k \cdot z(s_k^{(i)}, a_k^{(i)}),$$

$$\nabla V(\theta; z) \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta V(\theta; z; \zeta_i) := \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{a\in\mathcal{A}} \gamma^k Q^{\pi(\theta)}(s_k^{(i)}, a; z)\nabla_\theta \pi_\theta(a|s_k^{(i)}),$$

where $Q^{\pi(\theta)}(s, a; z) = \mathbb{E}_{\pi(\theta)}[\sum_t \gamma^t z(s_t, a_t)|s_0 = s, a_0 = a]$ denotes the Q-function.