Provably Learning Nash Policies in Constrained Markov Potential Games

Pragnya Alatur
Department of Computer Science
ETH Zurich and ETH AI Center
pragnya.alatur@ai.ethz.ch

Giorgia Ramponi*
Department of Computer Science
University of Zurich
giorgia.ramponi@uzh.ch

Niao He
Department of Computer Science
ETH Zurich
niao.he@inf.ethz.ch

Andreas Krause
Department of Computer Science
ETH Zurich
krausea@ethz.ch

ABSTRACT

Multi-agent reinforcement learning addresses sequential decision-making problems with multiple agents, where each agent optimizes its own objective. In many real-world scenarios, agents not only aim to maximize their goals but also need to ensure safe behavior. For example, in traffic routing, each vehicle (acting as an agent) seeks to reach its destination swiftly (an objective) while avoiding collisions (a safety constraint). Constrained Markov Games (CMGs) offer a natural framework for addressing safe MARL problems, but they are typically computationally challenging. In this work, we introduce and study *Constrained Markov Potential Games* (CMPGs), a significant subclass of CMGs. Initially, we demonstrate that Nash policies for CMPGs can be computed through constrained optimization. Then, we showed that Lagrangian-primal dual methods (one tempting approach to solve this optimization problem) cannot be used in this setting. In fact, unlike in single-agent scenarios, CMPGs do not satisfy strong duality, rendering such approaches inapplicable and potentially unsafe. To tackle the CMPG problem, we propose a novel algorithm **C**oordinate-**A**scent for **CMPGs** with **E**xploration (CA-CMPG-E), which provably converges to a Nash policy in tabular, finite-horizon CMPGs. The idea behind the algorithm is to solve for each agent a Constrained Markov Decision Process and update the joint policy in the direction of the steepest improvement. Furthermore, we provide the first sample complexity bounds for learning Nash policies in unknown CMPGs guaranteeing safe exploration.

KEYWORDS

Multi-agent Reinforcement Learning; Safe multi-agent learning

ACM Reference Format:
Pragnya Alatur, Giorgia Ramponi*, Niao He, and Andreas Krause. 2024. Provably Learning Nash Policies in Constrained Markov Potential Games. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) addresses sequential decision-making problems with *multiple agents*, where the decisions of individual agents may also affect others. In this work, we focus on a rich and fundamental class of MARL problems, known as *Markov Potential Games*, [MPGs, 20]. These problems find applications in crucial domains like traffic routing [3] and wireless communication [33]. The main characteristic of an MPG is the existence of an underlying *potential function*, which captures the agents' incentives to deviate between different policies. MPGs can model scenarios ranging from fully cooperative, where agents share a common objective, to settings where agents pursue individual goals, as long as a potential function aligns these objectives. For instance, in the context of traffic routing, each vehicle aims to identify the swiftest route to its destination. At the same time, there exists an underlying objective to minimize congestion, which is intrinsically tied to optimizing the choices made by each vehicle.

However, in many real-world applications, the standard MPG framework fails to incorporate additional safety requirements. Consider the aforementioned traffic routing scenario: here, we are not only interested in identifying the most efficient routes for individual vehicles but also in ensuring the safety of their journeys by preventing collisions. Combining these multiple objectives within a single reward function can be a tricky task, and a more intuitive approach involves the introduction of ad-hoc constraints. This necessitates a fresh perspective on the problem, leading us to introduce the innovative framework of Constrained Markov Potential Games (CMPGs).

In CMPGs, we address the critical issue of safety by introducing coupled constraints on the policies of the agents. These constraints introduce a layer of complexity since agents must cooperate to resolve them, while simultaneously striving to optimize their individual objectives independently. The significance of coupled constraints lies in their capacity to model essential requirements such as collision avoidance, which are inherently challenging to represent using unilateral constraints.

The objective of CMPGs is to identify a Nash Equilibrium policy [4, 27]. Such a policy represents a set of strategies where no individual agent has the incentive to deviate unilaterally while adhering to the constraints imposed. As previously mentioned, existing algorithms designed for tackling (unconstrained) MPGs, where each agent independently enhances its own objective, prove inadequate for addressing the constrained setting. This inadequacy stems from

[*] Work done at the ETH AI Center.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

the necessity for agents to coordinate their actions effectively to satisfy the imposed constraints. We delve deeper into a comprehensive discussion of prior work in Section 2.

In this paper, we consider tabular CMPGs in the finite-horizon setting, and our contributions are summarized as follows:

(1) First, we show that a Nash policy can in principle be recovered by solving a constrained optimization problem, which, however, becomes intractable as the number of agents increases (Section 4).
(2) Given tractable algorithms for unconstrained MPGs [cf. 18, 20], a tempting approach would be to utilizing Lagrangian duality to reduce the constrained problem to an unconstrained one, as done in previous works by [11, 28]. Unfortunately, we show that strong duality does not hold for our problem (Section 4), rendering such approaches *sub-optimal* and *unsafe*. This is in sharp contrast to the single-agent setting, for which strong duality does hold [29].
(3) Instead of solving the constrained optimization problem, we propose to directly search for a Nash policy. We present our algorithm – **C**oordinate-**A**scent for **CMPG**s (CA-CMPG) – which provably converges to an ε-Nash policy, assuming that the agents have full knowledge of the CMPG (Section 5).
(4) Finally, we provide a sample complexity bound for our algorithm CA-CMPG, when the agents do not know the CMPG beforehand (Section 6). With access to a generative model (Section 6.1), the agents converge to an ε-Nash policy with $\widetilde{O}\left(\frac{H^8}{\varepsilon^3\zeta^2}\right)$ samples, where ζ is the Slater constant of the CMPG and H is the horizon. On the other hand, if the agents do not have access to a generative model, but still want to ensure safe exploration, we obtain a sample complexity bound of $\widetilde{O}\left(\frac{H^{10}}{\varepsilon^5c^2}\right)$ (Section 6.2), where $c \in (0, \zeta]$ is a quantity related to the constraint set of the CMPG.

2 Related Work

In this section, we will focus on the results for MPGs in the tabular setting. Unless remarked, most work below focuses on *unconstrained* MPGs. MARL is a large area of research on its own. For a more comprehensive overview on MARL, we refer the reader to the surveys by Yang and Wang [34] and Zhang et al. [35].

Markov Potential Games: MPGs have become popular in recent years and have been studied for the tabular setting [9, 18, 20, 24, 25, 36, 37] and for state-action spaces with function approximation [10, 12]. For the tabular setting with *known* rewards and transitions, Leonardos et al. [20] prove that independent policy gradient (IPG) converges to an ε-Nash policy in $O(1/\varepsilon^2)$ iterations. If rewards and transitions are *unknown*, Mao et al. [25] prove that IPG with access to a stochastic gradient oracle converges to an ε-Nash policy with a sample complexity of $O\left(1/\varepsilon^{4.5}\right)$.

In these IPG algorithms, the agents improve their own objectives *independently*. It is challenging to apply these algorithms with coupled constraints, as the agents may need to coordinate to satisfy those constraints, at least during the learning process. Song et al. [31] present a different approach for tabular MPGs with unknown rewards and transitions, in which the agents *coordinate* to compute an ε-Nash policy with a sample complexity of $\widetilde{O}(1/\varepsilon^3)$. Maheshwari et al. [24] present a different approach with asymptotic convergence to a Nash policy, whereas we target finite-time convergence. Note that MPGs are only one way to model MARL problems, and for a more comprehensive overview on MARL, we refer the reader to the surveys by Yang and Wang [34] and Zhang et al. [35].

Constrained Markov Decision Processes: A common approach to constrained *single-agent* RL are *Constrained Markov Decision Processes* [CMDPs, 2]. CMDPs are widely studied, and a comprehensive survey is given by Gu et al. [19]. In CMDPs, the agent optimizes a reward function subject to constraints. Lagrangian duality is a common approach for constrained optimization and Paternain et al. [29] proved that CMDPs possess the *strong duality property*, giving theoretical justification for the use of Lagrangian dual approaches.

Constrained Markov Games: One of the common approaches to constrained multi-agent RL are *Constrained Markov Games* [CMGs, 4]. CMGs restrict the policies of the agents, which can be used to model safety objectives. Note that CMPGs are one class of CMGs. In cooperative CMPGs[1], where the agents have one common reward function, the CMPG objective very much resembles the CMDP formulation. Furthermore, Diddigi et al. [11] and Parnika et al. [28] demonstrate good experimental results for cooperative CMPGs with Lagrangian dual approaches, but provide no theoretical guarantees. We prove in our work, however, that strong duality does not hold in general for CMPGs (cf. Section 4), rendering Lagrangian dual approaches inapplicable in those cases. Furthermore, we demonstrate that the dual might even return unsafe solutions. Lu et al. [23] use a different approach and prove convergence to first-order stationary points in cooperative CMPGs, which are a weaker notion of the (generalized) Nash equilibria considered in our work.

Cai et al. [8], Elsayed-Aly et al. [16] propose shield-type mechanisms, where a shield prevents the agents' from taking unsafe actions. They empirically demonstrate that the agents satisfy the safety constraints and obtain high rewards, however, they do not provide any theoretical guarantees on whether the agents converge to an equilibrium. Shalev-Shwartz et al. [30] model the autonomous driving task as a CMDP and propose a different approach, which separates learning a good policy from learning to satisfy the constraints. They do not provide any guarantees on whether agents will reach an equilibrium if they employ their learning algorithm. We instead focus on the general notion of CMPGs and focus on learning Nash equilibria in those games.

3 Background and Problem Definition

Notation: For any $n \in \mathbb{N}$, we use the short-hand notation $[n]$ to refer to the set of integers $\{1, ..., n\}$. For any finite set X, we denote by Δ_X the probability simplex over X, i.e., $\Delta_X = \{v \in [0,1]^{|X|} \mid \sum_{x \in X} v(x) = 1\}$.

3.1 Markov Potential Games

An n-agent *Markov Potential Game* (MPG) is a tuple $\mathcal{G} = (\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, H, \{\mathcal{P}_h\}_{h=1}^H, \{\{r_{i,h}\}_{h=1}^H\}_{i=1}^n, \mu)$, where $\mathcal{S}$ is the state space, $\mathcal{A}_i$ is agent i's action space. We denote by $\mathcal{A} \triangleq \times_{i=1}^n \mathcal{A}_i$ the joint action space, $H \in \mathbb{N}_{>0}$ the horizon. $\mathcal{P}_h : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the environment's

[1] Note that cooperative games are a strict subclass of CMPGs, as CMPGs are able to model non-cooperative settings too.

transition function at time $h \in [H]$ and $\mathcal{P}_h(s'|s,a)$ denotes the probability of moving to state s' from state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ at step $h \in [H]$, $r_{i,h} : \mathcal{S} \times \mathcal{A} \to [0,1]$ is agent i's reward function at step $h \in [H]$ and $\mu \in \Delta_{\mathcal{S}}$ denotes the initial state distribution. We assume $\mathcal{S}$ and $\mathcal{A}$ to be finite.

Policies: For every agent $i \in [n]$, we define its policy space as

$$\Pi^i \triangleq \left\{ \{\pi_{i,h}\}_{h=1}^H \mid \pi_{i,h} : \mathcal{S} \to \Delta_{\mathcal{A}_i}, \forall h \in [H] \right\}.$$

If agent i follows a policy $\pi \in \Pi^i$, it means that at step $h \in [H]$ and state $s \in \mathcal{S}$, the agent samples its next action from $\pi_h(\cdot|s)$. We denote by $\Pi \triangleq \left\{ \boldsymbol{\pi} = (\pi_1, ..., \pi_n) \,|\pi_i \in \Pi^i, \forall i \in [n] \right\}$ the set of *joint* policies. For any policy $\boldsymbol{\pi} \in \Pi$ and agent $i \in [n]$, we denote by $\boldsymbol{\pi}_{-i}$ the policy of the *other* $n-1$ agents.

Value Function: For any policy $\boldsymbol{\pi} \in \Pi$ and agent $i \in [n]$, the value function $V^{r_i}(\boldsymbol{\pi})$ measures the expected, cumulative reward of agent i, and is defined as follows:

$$V^{r_i}(\boldsymbol{\pi}) \triangleq \mathop{\mathbb{E}}_{\substack{s \sim \mu, \\ a_h \sim \boldsymbol{\pi}_h(\cdot|s_h), \\ s_{h+1} \sim \mathcal{P}_h(\cdot|s_h, a_h)}} \Big[\sum_{h=1}^H r_{i,h}(s_h, a_h) | s_0 = s \Big]. \tag{1}$$

Potential Function: An MPG possesses an underlying potential function $\Phi : \Pi \to \mathbb{R}$ such that:

$$\begin{gathered} V^{r_i}(\pi_i, \boldsymbol{\pi}_{-i}) - V^{r_i}(\pi_i', \boldsymbol{\pi}_{-i}) = \Phi(\pi_i, \boldsymbol{\pi}_{-i}) - \Phi(\pi_i', \boldsymbol{\pi}_{-i}) \\ \forall \pi_i' \in \Pi^i, \forall \boldsymbol{\pi} \in \Pi, \forall i \in [n]. \end{gathered} \tag{2}$$

This is an adaptation of the potential function defined in [20] to the finite-horizon setting. Instead of defining a per-state potential function, we directly consider the potential function with respect to the initial distribution μ.

Remark: Note that the potential function is a property of the MPG and is typically not known to the agents. In a cooperative game, the agents have one shared reward function r such that $r_i \equiv r, \forall i \in [n]$. In this case, the potential function is simply the value function of the agents, i.e., $\Phi = V^r$. Note, however, that cooperative games are a *strict* subset of MPGs, and MPGs have the ability to express non-cooperative scenarios, such as traffic congestion. In Section 7, we describe different instances in detail.

3.2 Constrained Markov Potential Games

An n-agent *Constrained Markov Potential Game* (CMPG) is an MPG $\mathcal{G} = (\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, H, \{\mathcal{P}_h\}_{h=1}^H, \{\{r_{i,h}\}_{h=1}^H\}_{i=1}^n, \mu)$ with constraints $\{(\{c_{j,h}\}_{h=1}^H, \alpha_j)\}_{j=1}^k$, where $c_{j,h} : \mathcal{S} \times \mathcal{A} \to [0,1]$ denotes the j-th cost function at step $h \in [H]$ and $\alpha_j \in [0,H]$ is the constraint threshold.[2]

Feasible Policies: We call a policy $\boldsymbol{\pi} \in \Pi$ *feasible*, if it satisfies the following constraints:

$$V_\mu^{c_j}(\boldsymbol{\pi}) \triangleq \mathop{\mathbb{E}}_{\substack{s \sim \mu, \\ a_h \sim \boldsymbol{\pi}_h(\cdot|s_h), \\ s_{h+1} \sim \mathcal{P}_h(\cdot|s_h, a_h)}} \Big[\sum_{h=1}^H c_{j,h}(s_h, a_h) \Big| s_0 = s \Big] \le \alpha_j, \quad \forall j \in [k].$$

In the rest of the paper, we use Π_C to refer to the set of *feasible* policies. For every agent i and policy $\boldsymbol{\pi}_{-i}$ of the other $n-1$ agents, we define $\Pi_C^i(\boldsymbol{\pi}_{-i}) \triangleq \left\{ \pi_i \in \Pi^i | (\pi_i, \boldsymbol{\pi}_{-i}) \in \Pi_C \right\}$. We refer to this type of constraints as *coupled* constraints, as the values of the constraints depend on the *joint* actions of the agents. If we wish to model an intersection in a traffic scenario, an important constraint to incorporate would be collision avoidance. To decide whether a certain set of actions causes a collision or not, we need to take the actions of *all* agents at the intersection into account.

In a CMPG, each agent i aims to maximize its own value function V^{r_i}. Since the rewards and transitions depend on the *joint* policy, it may not be possible to find a policy that is globally optimal for all value functions simultaneously. Instead, the agents typically need to settle for an equilibrium policy, at which no agent has an incentive to deviate unilaterally. Many different types of equilibria exist in the literature, such as the Nash equilibrium [27], correlated equilibrium [5] or Stackelberg equilibrium [6]. In this work, our goal is to obtain a *Nash equilibrium policy* [4, 27] in a CMPG. We define a relaxed notion in the following paragraph.

ε-Nash Equilibrium Policy: For any $\varepsilon \ge 0$, a policy $\boldsymbol{\pi}^* = (\pi_1^*, ..., \pi_n^*) \in \Pi_C$ is a *ε-Nash equilibrium policy*, if it is the ε-best-response policy for each agent, i.e.,[3]:

$$\max_{\pi_i \in \Pi_C^i(\boldsymbol{\pi}_{-i}^*)} V^{r_i}(\pi_i, \boldsymbol{\pi}_{-i}^*) - V^{r_i}(\boldsymbol{\pi}^*) \le \varepsilon, \qquad \forall i \in [n]. \tag{3}$$

We call $\boldsymbol{\pi}^*$ a *Nash equilibrium policy*, if Eq. (3) holds with $\varepsilon = 0$. In the rest of the paper, we refer to the Nash equilibrium policy as *Nash policy.*

3.3 Constrained Markov Decision Processes

A *Constrained Markov Decision Process* (CMDP) is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \{\mathcal{P}_h\}_{h=1}^H, \{r_h\}_{h=1}^H, \mu, \{(\{c_{j,h}\}_{h=1}^H, \alpha_j)\}_{j=1}^k)$. In a CMDP, there is a *single* agent. However, the individual elements in $\mathcal{M}$ carry the same meaning as in CMPGs. Furthermore, the policy sets Π, Π_C and the value functions $V^r : \Pi \to \mathbb{R}$ (reward), $V^{c_j} : \Pi \to \mathbb{R}, j \in [k]$ (costs) are defined in the same way as for CMPGs. In a CMDP, the agent aims to find a policy π^*, that satisfies:

$$\pi^* \in \arg\max_{\pi \in \Pi_C} V^r(\pi). \tag{4}$$

In the following section, we prove that a Nash policy in a CMPG can be found by maximizing the potential function with respect to the given constraints, similar to Eq. (4). We will show that Lagrangian duality, a common approach for constrained optimization, will not work in general for CMPGs.

4 Duality for Constrained Markov Potential Games?

For an MPG with potential function Φ, a globally optimal policy $\boldsymbol{\pi}^* \in \arg\max_{\boldsymbol{\pi} \in \Pi} \Phi(\boldsymbol{\pi})$ is also a Nash policy [20]. We show in Proposition 4.1 that this property generalizes to CMPGs. We defer the proofs for all theoretical results in this section to Appendix A.

Proposition 4.1. *Define the following constrained optimization problem:*

$$\boldsymbol{\pi}^* \in \arg\max_{\boldsymbol{\pi} \in \Pi_C} \Phi(\boldsymbol{\pi}). \tag{5}$$

Then, $\boldsymbol{\pi}^*$ is a Nash policy for a CMPG with potential function Φ.

[2] Even though we define our problem in the finite-horizon setting, our results can be easily extended to the discounted, infinite-horizon setting.

[3] This is an extension of the *generalized Nash equilibrium* [17] to CMPGs.

Solving Eq. (5) directly is not trivial; even if the agents know the rewards and transitions, the potential function is usually not known to the agents. Moreover, the fact that we have *coupled* constraints makes solving Eq. (5) directly intractable. This problem does not arise in the unconstrained setting because each agent can solve part of the problem independently.

Nevertheless, a common approach for solving constrained optimization problems is *Lagrangian duality*, which, in our case, turns the CMPG into an (unconstrained) MPG with modified rewards (Proposition 4.2). This would enable the use of scalable algorithms that have been developed for unconstrained MPGs [20]. Furthermore, in previous works [11, 21], Lagrangian duality was used for cooperative CMPGs and showed promising experimental results. This makes Lagrangian duality a tempting approach for CMPGs. For this, we define the *Lagrangian* $\mathcal{L} : \Pi \times \mathbb{R}_+^k \to \mathbb{R}$ and the primal[4] and dual problems for Eq. (5) as follows:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda}) \triangleq \Phi(\boldsymbol{\pi}) + \sum_{j=1}^{k} \lambda_j \left(\alpha_j - V^{c_j}(\boldsymbol{\pi})\right) \qquad \text{(Lagrangian)}$$

$$P^* = \max_{\boldsymbol{\pi} \in \Pi} \min_{\boldsymbol{\lambda} \in \mathbb{R}_+^k} \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda}) \qquad \text{(Primal)}$$

$$D^* = \min_{\boldsymbol{\lambda} \in \mathbb{R}_+^k} \max_{\boldsymbol{\pi} \in \Pi} \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda}). \qquad \text{(Dual)}$$

As a first step, in Proposition 4.2, we prove that the dual problem does indeed correspond to an (unconstrained) MPG.

Proposition 4.2. *For any $\boldsymbol{\lambda} \in \mathbb{R}_+^k$, $\mathcal{L}(\cdot, \boldsymbol{\lambda})$ is a potential function for an MPG with reward functions $\tilde{r}_{i,h} \triangleq r_{i,h} - \sum_{j=1}^{k} \lambda_j c_{j,h}, \forall i \in [n], \forall h \in [H]$.*

Then, weak duality guarantees that $D^* \geq P^*$ holds. Unfortunately, in the following proposition, however, we show that *strong duality*, i.e., $D^* = P^*$, *does not hold* in general for CMPGs.

Theorem 4.3. *There exists a CMPG, for which strong duality does not hold, i.e., for which $P^* \neq D^*$.*

Discussion: To give an intuition on Theorem 4.3, consider a cooperative CMPG with $\Phi \equiv V^r$, i.e., the potential function is equal to the shared value function V^r. Note that, in this case, the primal problem very much resembles the CMDP objective (Eq. (5)) and it is tempting to solve the CMPG as a CMDP with a large action space $\mathcal{A} = \times_{i=1}^n \mathcal{A}_i$. Recall also, that strong duality does indeed hold for CMDPs [29] and CMDPs can be solved via primal-dual algorithms. By solving this large CMDP, we obtain a solution $\boldsymbol{\pi}^*$ that specifies distributions over the *joint* action space $\mathcal{A}$. To obtain a solution for the original CMPG, however, we require a policy that can be factored into a set of independent policies $\{\pi_i^*\}_{i \in [n]}$ such that $\boldsymbol{\pi}_h^*(a|s) = \prod_{i=1}^n \pi_{i,h}^*(a_i|s), \forall (s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$. We show in Appendix A that unfortunately, this property is not always guaranteed, implying that also for a simple class of CMPGs, strong duality may not always hold.

5 Solving Constrained Markov Potential Games

In this section, we propose an efficient algorithm to compute Nash policies in CMPGs[5]. Similar to the work on unconstrained MPGs by Song et al. [31], in our algorithm **C**oordinate-**A**scent for **CMPG**s (CA-CMPG), agents take turns to solve a *Constrained Markov Decision Process* (CMDP), i.e., a single-agent reinforcement learning problem, in every iteration. To do this, the agents need to coordinate, such that when one agent is solving the CMDP, the others provide a stationary environment to that agent by keeping their policies fixed. There are some technical challenges compared to the unconstrained MPG setting. The main difference is that in the CMPG setting, to ensure the convergence to a Nash policy, we need also to ensure that the intermediate policies remain *feasible* (see remark at the end of this section). Our algorithm CA-CMPG is described in Algorithm 1.

Algorithm 1 CA-CMPG (Known Transitions)

```
Require: $\varepsilon > 0$ (approximation error), $\boldsymbol{\pi}^S \in \Pi_C$ (feasible policy), $T$ (number of iterations)
$\boldsymbol{\pi}^0 \leftarrow \boldsymbol{\pi}^S$
for $t = 1, \dots, T$ do
    for agent $i = 1, \dots, n$ do
        Agent $i$ computes $\hat{\pi}_i{}^t$ such that Eq. (6) is satisfied.
        $\varepsilon_i^t \leftarrow V^{r_i}(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1}) - V^{r_i}(\boldsymbol{\pi}^{t-1})$.
    if $\max_{i \in [n]} \varepsilon_i^t > \varepsilon/2$ then
        Set $\boldsymbol{\pi}^t = (\hat{\pi}_j^t, \boldsymbol{\pi}_{-j}^{t-1})$, where $j = \arg\max_{i \in [n]} \varepsilon_i^t$, break ties arbitrarily.
    else
        break
```

We assume for now that the agents know their own reward functions, the cost functions as well as the transition model. As a starting point for CA-CMPG, the agents require access to a feasible, initial policy, which we state in the following assumption:

Assumption 1. *Given a CMPG, the agents have access to a feasible policy $\boldsymbol{\pi}^S \in \Pi_C$.*

This type of assumption is common in the single-agent CMDP setting [7, 22]. While finding such a policy in the *multi-agent* setting may be computationally expensive, we provide two examples here and explain, how such a policy can be computed by the agents. In both cases, we assume that the CMPG is feasible, i.e., $\Pi_C \neq \emptyset$.

Example 5.1 (Single Constraint). Consider the problem $\min_{\boldsymbol{\pi} \in \Pi} V^{c_1}(\boldsymbol{\pi})$. Since the constraint set is feasible, we must have that $\min_{\boldsymbol{\pi} \in \Pi} V^{c_1}(\boldsymbol{\pi}) \leq \alpha_1$. Note that this is an unconstrained Markov decision process (MDP) with state space $\mathcal{S}$ and action space $\mathcal{A}$. It is well-known that MDPs always possess at least one *deterministic*, optimal policy, which can be computed using dynamic programming techniques. Thus, we compute a deterministic policy $\boldsymbol{\pi}^C \in \arg\min_{\boldsymbol{\pi} \in \Pi} V^{c_1}(\boldsymbol{\pi})$, s.t. for every state $s \in \mathcal{S}$ and step $h \in [H]$, there is exactly one action $a = (a_1, \dots, a_n) \in \mathcal{A}$, for which $\boldsymbol{\pi}_h^C(a|s) = 1$ and $\boldsymbol{\pi}_h^C(a'|s) = 0, \forall a' \neq a$. Then, for every agent $i \in [n]$, we set $\pi_{i,h}^C(a_i|s) = 1$ and $\pi_{i,h}^C(a_i'|s) = 0$, for all $a_i' \neq a_i$. It is easy to verify that $\boldsymbol{\pi}^C = \prod_{i=1}^n \pi_i^C$.

Example 5.2 (Independent Transitions and Composite Constraints). Consider a CMPG with per-agent state spaces $\mathcal{S}_1, \dots, \mathcal{S}_n$ and transition models $\mathcal{P}_1, \dots, \mathcal{P}_n$, where $\mathcal{P}_{j,h}(s'|s, a)$ is the probability that agent j transitions to state $s' \in \mathcal{S}_j$ from state-action pair $(s, a) \in \mathcal{S}_j \times \mathcal{A}_j$ at step $h \in [H]$. We denote by $\mathcal{S} \triangleq \times_{i=1}^n \mathcal{S}$ the joint

[4] Note that the primal is equivalent to Eq. (5).

[5] Note that we may not find a Nash policy that solves Eq. (5) though.

state space and define $\mathcal{P}_h(s'|s,a) \triangleq \prod_{i=1}^{n} \mathcal{P}_h^i(s_i'|s_i,a_i)$ as the joint probability of transitioning to state $s' \in \mathcal{S}$ from state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ at step $h \in [H]$. Furthermore, assume that for each $j \in [k]$, the constraint function c_j can be written as $c_{j,h}(s,a) \triangleq \sum_{i=1}^{n} c_{j,h}^i(s_i,a_i)$. Due to this, the cumulative constraints can be written as $V^{c_j}(\boldsymbol{\pi}) = \sum_{i=1}^{n} V^{c_j^i}(\pi_i), \forall j \in [k]$. To find a feasible policy, each agent $i \in [n]$ computes $\pi_i \in \left\{\pi \in \Pi^i \middle| V^{c_j^i}(\pi) \leq c_i^*, \forall j \in [k]\right\}$, where $c_i^* \triangleq \min_{c \in \mathbb{R}} \left\{\exists \pi \in \Pi^i \middle| V^{c_j^i}(\pi) \leq c, \forall j \in [k]\right\}$. Assuming that the constraint set is feasible, it is easy to see that $\boldsymbol{\pi}^S = (\pi_1^S, ..., \pi_n^S)$ must be feasible.

5.1 Algorithm

In CA-CMPG, the agents start with the feasible policy $\boldsymbol{\pi}^S$. In every iteration, the agents take turns to maximize their own value function. While one agent is maximizing its value function, the other agents keep their policy fixed (Line 4); therefore, that agent is essentially solving a CMDP. We defer the exact description of the CMDP that agent i faces in iteration t to Appendix B. Let us recall the CMDP objective from Eq. (4). In practice, we can only solve Eq. (4) *approximately*. Given $\varepsilon > 0$, we assume that in every iteration t, agent $i \in [n]$ can efficiently compute a policy $\hat{\pi}_i^t \in \Pi_C^i(\boldsymbol{\pi}_{-i}^{t-1})$ such that it satisfies the following conditions[6]:

$$\max_{\pi \in \Pi_C^i(\boldsymbol{\pi}_{-i}^{t-1})} V^{r_i}(\pi, \boldsymbol{\pi}_{-i}^{t-1}) - V^{r_i}(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1}) \leq \varepsilon/2. \tag{6}$$

Due to the potential property (Eq. (2)), if agent $i \in [n]$ improves its own value function, it implicitly also improves the potential function. To prove that the potential function can be increased only a finite number of times, implying termination of CA-CMPG, we require the potential function to be bounded.

Lemma 5.3. *Fix an arbitrary base policy $\boldsymbol{\pi}^B \in \Pi$. Then, for every $\boldsymbol{\pi} \in \Pi$, the potential function can be bounded as: $\Phi(\boldsymbol{\pi}) \leq nH + \Phi(\boldsymbol{\pi}^B)$.*

We defer the proofs of all theoretical results in this section to Appendix B.

Theorem 5.4. *Suppose that Assumption 1 holds. Then, given $\varepsilon > 0$, if we invoke CA-CMPG with $T = \frac{2nH}{\varepsilon}$, it converges to an ε-Nash policy.*

Remark: What if we relax the feasibility requirement in Eq. (6) and allow the CMDP solver to return an *ε-feasible* policy $\boldsymbol{\pi}$ such that $V^{c_j}(\boldsymbol{\pi}) \leq \alpha_j + \varepsilon, \forall j \in [k]$, for an $\varepsilon > 0$? In that case, the intermediate policies might not be feasible and CA-CMPG may get stuck in an infeasible policy, which is not a Nash policy.

6 Learning in Unknown Constrained Markov Potential Games

In this section, we assume that the agents do not know the transition model beforehand. For simplicity, we assume that they do know the rewards and costs[7]. Our objective is to establish a *sample complexity* bound for learning in CMPGs. Concretely, we want to construct an algorithm, such that, given any $\varepsilon > 0, \delta \in (0,1)$, the algorithm returns an ε-Nash policy with probability at least $1-\delta$, using at most $\mathcal{F}(\varepsilon,\delta)$ *samples* from the transition model $\mathcal{P}$. Before we proceed, we define an important quantity related to the constraint set, which also contributes to the final sample complexity.

Algorithm 2 CA-CMPG-E (Unknown Transitions)

```
Require: $\varepsilon > 0$ (approximation error), $\delta \in (0,1)$ (confidence), $\boldsymbol{\pi}^S \in \Pi_C$ (feasible policy), $T$ (number of iterations), $M > 0$ (number of samples per policy)
1:  $\boldsymbol{\pi}^0 \leftarrow \boldsymbol{\pi}^S$
2:  for $t = 1, ..., T$ do
3:      Execute policy $\boldsymbol{\pi}^{t-1}$ for $M$ episodes and estimate $\hat{V}^{r_1}(\boldsymbol{\pi}^{t-1}), ..., \hat{V}^{r_n}(\boldsymbol{\pi}^{t-1})$.
4:      for agent $i = 1, ..., n$ do
5:          Agent $i$ computes $\hat{\pi}_i{}^t$ such that Eq. (7) is satisfied.
6:          Execute policy $(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1})$ for $M$ episodes and estimate $\hat{V}^{r_i}(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1})$.
7:          $\varepsilon_i^t \leftarrow \hat{V}^{r_i}(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1}) - \hat{V}^{r_i}(\boldsymbol{\pi}^{t-1})$.
8:      if $\max_{i \in [n]} \varepsilon_i^t > \varepsilon/2$ then
9:          Set $\boldsymbol{\pi}^t = (\hat{\pi}_j{}^t, \boldsymbol{\pi}_{-j}^{t-1})$, where $j = \arg\max_{i \in [n]} \varepsilon_i^t$, break ties arbitrarily.
10:     else
11:         break
```

Definition 6.1 (Slater constant). Given a feasible CMPG $\mathcal{G}$, we define its Slater constant ζ as follows:

$$\zeta \triangleq \min_{j \in [k]} \min_{i \in [n]} \min_{\boldsymbol{\pi}_{-i} \in \Pi \backslash \Pi^i} \max_{\pi \in \Pi^i} \{\alpha_j - V^{c_j}(\pi, \boldsymbol{\pi}_{-i})\}.$$

We call $\mathcal{G}$ *strictly* feasible if and only if $\zeta > 0$.

In the rest of this section, we assume that the agents face an unknown, strictly feasible CMPG with Slater constant $\zeta > 0$. Next, we discuss which parts of CA-CMPG need to be adapted for this setting.

(1) In every iteration t, each agent $i \in [n]$ needs to solve the CMDP described in Section 5 (Line 4). To solve this CMDP, we assume access to a *sample-efficient* CMDP solver, which has the following guarantees: Given $\varepsilon > 0, \delta \in (0,1)$, the solver uses at most $\mathcal{F}_C\left(|\mathcal{S}|, |\mathcal{A}_i|, H, \zeta, \delta, \frac{\varepsilon}{4}\right)$ samples and returns a policy $\hat{\pi}_i^t \in \Pi_C^i(\boldsymbol{\pi}_{-i}^{t-1})$ such that it satisfies the following, with probability at least $1-\delta$:

$$\max_{\pi \in \Pi_C^i(\boldsymbol{\pi}_{-i}^{t-1})} V^{r_i}(\pi, \boldsymbol{\pi}_{-i}^{t-1}) - V^{r_i}(\hat{\pi}_i{}^t, \boldsymbol{\pi}_{-i}^{t-1}) \leq \varepsilon/4, \tag{7}$$

Compared to the setting with known transitions, we have a stricter bound on the approximation error of $\varepsilon/4$ here. We discuss in Appendix C, why we require this.

(2) To compute ε_i^t in step t, agent i needs to estimate the value functions $V^{r_i}(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1})$ and $V^{r_i}(\boldsymbol{\pi}^{t-1})$. For the former, the agents execute the policy $(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1})$ for $M > 0$ episodes[8] and agent i estimates $\hat{V}^{r_i}(\hat{\pi}_i^t, \boldsymbol{\pi}_{-i}^{t-1})$ with the average of the observed, cumulative rewards. For the latter, similarly, the agents execute $\boldsymbol{\pi}^{t-1}$ for M episodes, but these observations

[6] This can be achieved using state-of-the-art primal-dual methods, such as the work by Ding et al. [15], Paternain et al. [29].

[7] In general, learning the transitions is harder than learning rewards and costs. Concretely, this also means that learning rewards and costs will not add any dominating terms to the overall sample complexity (see Vaswani et al. [32]).

[8] Each episode is a sequence of H steps. At the beginning of each episode, the initial state is freshly sampled from μ.

can be used to estimate $V^{r_1}(\boldsymbol{\pi}^{t-1}), ..., V^{r_n}(\boldsymbol{\pi}^{t-1})$ simultaneously[9].

The resulting algorithm **C**oordinate-**A**scent for **CMPGs** with **E**xploration (CA-CMPG-E) is described in Algorithm 2.

Theorem 6.2. *Given a strictly feasible CMPG $\mathcal{G}$ with Slater constant $\zeta > 0$, suppose that the agents have access to an initial feasible policy (cf. Assumption 1). Furthermore, assume that the agents have access to a sample-efficient CMDP solver (Eq. (7)). Then, for any $\varepsilon > 0$, $\delta \in (0, 1)$, CA-CMPG-E invoked with $M = \frac{32H^2}{\varepsilon^2}\log\left(\frac{32n^2H}{\varepsilon\delta}\right)$ and $T = \frac{4nH}{\varepsilon}$ returns an ε-Nash policy with probability at least $1-\delta$, using the following number of samples:*

$$\mathcal{F}(\varepsilon,\delta) \triangleq \sum_{t=1}^{T}\sum_{i=1}^{n}\mathcal{F}_C\left(|S|, |\mathcal{A}_i|, H, \zeta, \frac{\varepsilon\delta}{8n^2H}, \frac{\varepsilon}{4}\right) + \frac{256n^2H^4}{\varepsilon^3}\log\left(\frac{32n^2H}{\varepsilon\delta}\right).$$

Discussion The previous result is similar to the result obtained by previous works on unconstrained MPGs. In fact, for unconstrained MPGs, without knowing the transition dynamics, the best-known sample complexity for learning an approximate Nash equilibrium is of order $O\left(\frac{1}{\varepsilon^3}\right)$ [10] (or worse, $O\left(\frac{1}{\varepsilon^5}\right)$, when independent learning is considered [14]). On the other hand, the sample complexity can increase due to the CMDP solver.

In the next two sub-sections, we will instantiate CA-CMPG-E with two different state-of-the-art CMDP solvers and state the resulting sample complexity bounds. Both algorithms are designed for CMDPs with a *single* constraint. Due to this, we set $k = 1$ and denote our cost function by $\{c_h\}_{h=1}^{H}$ and refer to the constraint parameter as α. Note that this is due to a limitation of the existing CMDP algorithms and not of CA-CMPG-E.

6.1 Generative model

In this section, we assume that the agents have access to a *generative model*, i.e., they can directly query an oracle to obtain samples from the transition model $\mathcal{P}_h(\cdot|s, a)$, for any state-action pair $(s, a) \in \mathcal{S}\times\mathcal{A}$ and any $h \in [H]$ Similar to previous results in CMDPs [32] we propose a novel algorithm for finite-horizon CMDPs and describe it in Algorithm 4 (cf. Appendix D). Lemma D.1 (cf. Appendix D) establishes the sample complexity for the generative model setting (Algorithm 4[10], Appendix D).

Corollary 6.3. *Given a strictly feasible CMPG $\mathcal{G}$, assume that its Slater constant $\zeta > 0$ is known. Furthermore, assume that the agents invoke Algorithm 4 with $\varepsilon' = \frac{\varepsilon}{4}$, $\delta' = O\left(\frac{\varepsilon\delta}{n^2H}\right)$ and parameters set as in Lemma D.1 to solve Eq. (7). Then, for any $\varepsilon > 0, \delta \in (0, 1)$, CA-CMPG-E invoked with $M = O\left(\frac{H^2}{\varepsilon^2}\log\left(\frac{nH}{\varepsilon\delta}\right)\right)$ and $T = \frac{4nH}{\varepsilon}$, returns an ε-Nash policy with probability at least $1-\delta$ with an overall sample complexity of:*

$$\mathcal{F}(\varepsilon,\delta) \leq \widetilde{O}\left(\frac{n|\mathcal{S}|H^8\log\left(\frac{1}{\varepsilon\delta}\right)\sum_{i=1}^{n}|\mathcal{A}_i|}{\varepsilon^3\zeta^2} + \frac{n^2H^4\log\left(\frac{1}{\varepsilon\delta}\right)}{\varepsilon^3}\right).$$

Discussion: Compared to the result for *unconstrained* MPGs [31, Theorem 7], our Corollary 6.3 has an additional dependence on $\frac{1}{\zeta^2}$ and a worse dependence on the horizon H. These are due to the fact that our CMDP solver must always return a *feasible* policy. Finally, the sample complexity result in Song et al. [31] explicitly depends on $\Phi_{max} \triangleq \max_{\boldsymbol{\pi}\in\Pi}\Phi(\boldsymbol{\pi})$, whereas we substituted $\Phi_{\max} \leq nH$ (Lemma 5.3).

6.2 Safe exploration without a generative model

We now consider the more challenging setting where the agents do not have access to a generative model, but can only explore by executing policies and observing the transitions. Moreover, during the learning process, we want to ensure that the agents explore *safely*, i.e. we are not violating the constraints during the all learning process. We need then to use a CMDP algorithm with zero-constraint violations. Existing algorithms with safe exploration for CMDPs [7, 22] have guarantees on the *regret*, but no sample complexity guarantees. To address this, we derive a sample complexity bound for the algorithm by Bura et al. [7] (Algorithm 5 [11] in Appendix E) in Lemma E.2.

However, to apply this CMDP solver in CA-CMPG-E, we need to ensure that in every iteration, the agents have access to a *strictly* feasible policy [12]. The additional assumption is described below.

Assumption 2. *There exists $c \in (0, \zeta]$ s.t. for any agent $i \in [n]$ and policy $\boldsymbol{\pi}_{-i} \in \Pi_C \setminus \Pi^i$ of the other agents, the agent can obtain a strictly feasible policy $\pi \in \Pi^i$ s.t. $V^c(\pi, \boldsymbol{\pi}_{-i}) \leq \alpha - c$.*

This is a stronger assumption than in Section 6.1, as we additionally require access to a strictly feasible policy for every CMDP that is solved in CA-CMPG-E.

Corollary 6.4. *Suppose that Assumption 2 holds. Given $\varepsilon > 0, \delta \in (0, 1)$, assume that we invoke CA-CMPG-E with $M = O\Big(\frac{H^2}{\varepsilon^2}\log\left(\frac{nH}{\varepsilon\delta}\right)\Big)$ and $T = \frac{4nH}{\varepsilon}$. Furthermore, assume that we use Algorithm 5 as CMDP solver with $\varepsilon' = \frac{\varepsilon}{4}$, $\delta' = O\left(\frac{\varepsilon\delta}{n^2H}\right)$ and parameters set as in Lemma E.2. Then, CA-CMPG-E returns an ε-Nash policy with probability at least $1-\delta$ with an overall sample complexity of:*

$$\mathcal{F}(\varepsilon,\delta) \leq \widetilde{O}\left(\frac{n|\mathcal{S}|^2H^{10}\log\left(\frac{1}{\varepsilon\delta}\right)\sum_{i=1}^{n}|\mathcal{A}_i|}{\varepsilon^5c^2} + \frac{n^2H^4\log\left(\frac{1}{\varepsilon\delta}\right)}{\varepsilon^3}\right).$$

Discussion: Note that to satisfy Assumption 2, any $c \in (0, \zeta]$ is a valid choice. A large c yields a better sample complexity for Corollary 6.4, but restricts the set of strictly feasible policies for the CMDP solver. A smaller c increases the sample complexity, but gives more flexibility, as it allows for a larger set of strictly feasible

[9] This holds because we assumed that the reward functions are known.

[10] Algorithm 4 is the same as Algorithm 2 with a sample-efficient algorithm with access to a generative model.

[11] Algorithm 5 is the same as Algorithm 2 with a sample-efficient algorithm without access to a generative model.

[12] This assumption is related to the CMDP solver used, and could be not necessary for different CMDP solvers.

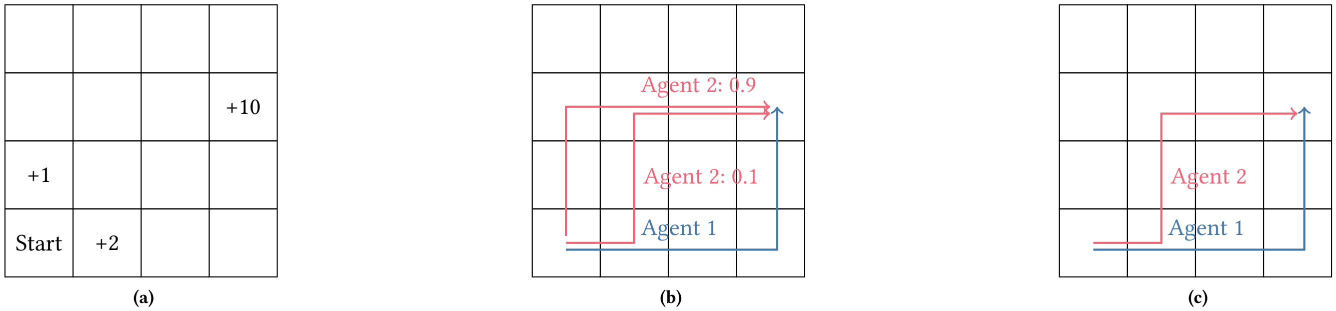


Figure 1: Grid world experiment: Fig. 1a illustrates the state space that the agents navigate in. Both agents start from the bottom left state and their goal is to maximize the sum of their individual rewards. The numbers on the states indicate the rewards associated with those states. Fig. 1b displays the policies with their corresponding probabilities returned by CA-CMPG. If the agents were to solve the dual problem directly (Section 4), they might obtain the policy illustrated in Fig. 1c, which is not feasible.

policies. Comparing the two corollaries, we observe that safe exploration without a generative model leads to a worse dependence of $|S|$, H, and ε.

The obtained sample complexity is worse compared to the results in the unconstrained setting [31]. On the other hand, it is comparable with new results on MPGs with independent learning [13]. We need also to notice that exploring safely is a harder task with respect to the usual MPG setting and that a worse sample complexity can be expected.

7 Experiments

Grid world: We consider a cooperative CMPG with two agents, in which the agents navigate in a 4x4 grid world (more details can be found in Appendix F). Each cell in the grid represents a state and in every state, each agent can choose to move *up*, *right*, *down*, or *left*. State transitions are deterministic and if an agent selects an action that would make it leave the grid, it remains in the current state. Fig. 1a illustrates the rewards that an agent can obtain in the individual states. Both agents start from the bottom left state and their goal is to reach the target state, which is the state with a reward of 10. To model this as a cooperative game, we set the agents' joint objective to be the sum of their individual rewards. Whenever the agents are in the same state, excluding the start and target states, they *collide* and incur a cost of 1. The agents must keep the expected cost below a pre-defined threshold $\alpha \in [0, 1]$.

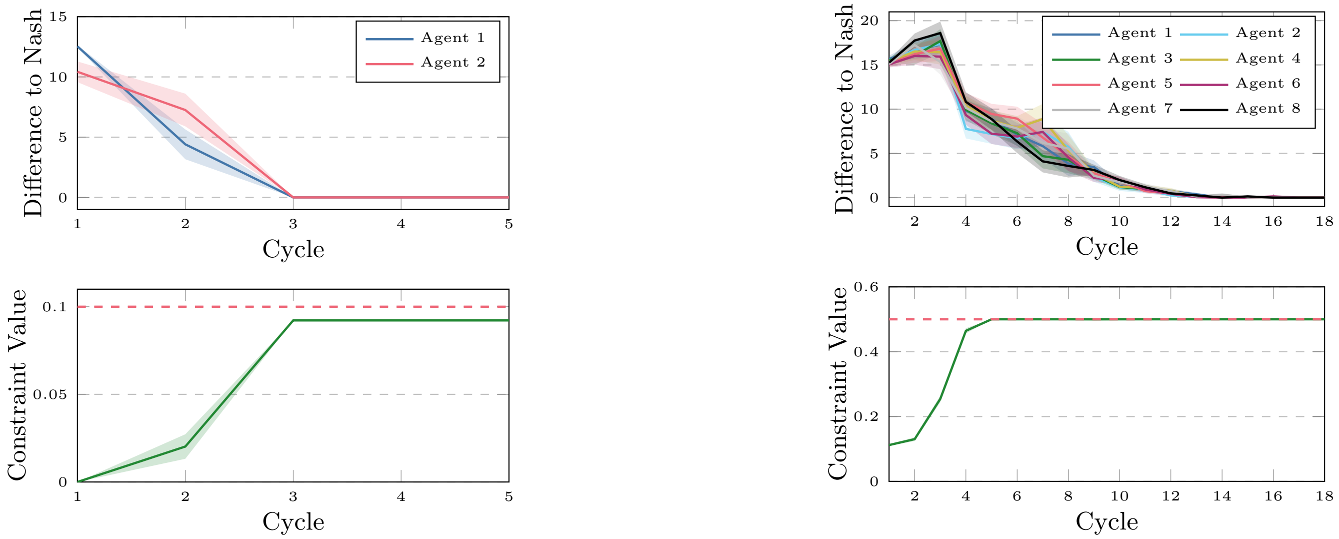
The resulting policies with the corresponding probabilities are shown in Fig. 1b. With this, the agents collide once with probability 0.1, thus, satisfying the constraint of the experiment. On the other hand, if we solve the Lagrangian dual problem directly (Section 4), one of the returned policies is illustrated in Fig. 1c. In this case, they always have one collision, which does not satisfy the constraint of the experiment, leading then to an infeasible policy.

We evaluate our algorithm CA-CMPG with known transitions and use a primal-dual algorithm as CMDP solver. We set the horizon to $H = 6$ and use a threshold of $\alpha = 0.1$. Fig. 2a (top row) displays the reward differences between the current policy and the new policy for both agents and after every cycle of the algorithm, averaged over 20 runs. One cycle corresponds to one full iteration of Algorithm 2, i.e. all agents solving their CMDPs. When the reward differences reach zero for both agents, this implies that the agents have converged to a Nash policy. The bottom row tracks the cost over the cycles of Algorithm 2. The agents start from a strictly feasible policy with a cost of 0, and converge to a policy with a cost close to α.

Congestion game: We consider a finite-horizon version of the setup described in Leonardos et al. [20] in which every state is a congestion game[13]. The game (cf. Appendix F) consists of two states $\mathcal{S} = \{\texttt{safe}, \texttt{unsafe}\}$, N agents and action space $\mathcal{A} = \{A, B, C, D\}$ for every agent. Each action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ has a weight $w_a^s > 0$ associated with it. In the safe state, an agent that selects action $a \in \mathcal{A}$, receives a reward of $k_a \cdot w_a^{\texttt{safe}}$, where k_a denotes the number of agents that selected action a. In the unsafe state, the reward structure is similar, however, we subtract an offset $c \geq 0$, resulting in a reward of $k_a \cdot w_a^{\texttt{unsafe}} - c$. In both states $s \in \mathcal{S}$, the weights follow the order $w_A^s < w_B^s < w_C^s < w_D^s$. Thus, in both states, the agents prefer to take the action that is chosen by most agents. Furthermore, for every action $a \in \mathcal{A}$, $k_a \cdot w_a^{\texttt{safe}} \gg k_a \cdot w_a^{\texttt{unsafe}} - c$ s.t. the agents prefer to stay in the safe state. In the safe state, if more than $N/2$ agents choose the same action, the system transitions to the unsafe state. To get back to the safe state from the unsafe state, the agents must equally distribute themselves among the four actions (see more details in Appendix F).

We evaluate our algorithm CA-CMPG with $N = 8$ agents and a horizon of $H = 2$. Furthermore, we assume that the transitions are known and use a linear program to solve the CMDPs [2]. For the initial state, we set $\mu(\texttt{safe}) = \mu(\texttt{unsafe}) = 0.5$. At step $h = 1$, in the unsafe state, if more than $N/2$ agents select the same action, the agents incur a cost of 1. Their goal is to keep the cost below a threshold $\alpha = 0.5$. Fig. 2b (top row) displays, as before, the reward differences between the current policy and the new policy, for each agents and averaged over 50 runs. When this difference reaches zero, this implies that the agents have converged to a Nash policy. The bottom plots track the cost over the cycles of Algorithm 2. The agents start from a strictly feasible policy with a cost of 0, and converge to a value close to α.

[13] Note that every congestion game is also a potential game and vice versa [26]; however, the setup considered here may not necessarily be MPGs as pointed out in Leonardos et al. [20].



(a) Grid world

(b) Congestion game

Figure 2: These plots illustrate the results of the grid world (Fig. 2a) and congestion game (Fig. 2b) experiments. One cycle on the x-axis corresponds to one full iteration of Algorithm 2, i.e. all N agents solving their CMDPs. The top row displays, for each agent, an average of their reward difference between the current and new policy. When the difference reaches zero, they converge to a Nash policy. The bottom row tracks the averaged cost over the cycles of Algorithm 2 (green, solid line). The agents start from a strictly feasible policy and converge to a cost close to α (red, dashed line). In all plots, we additionally plot the standard error.

8 Conclusion

In this paper, we made a significant contribution by demonstrating that strong duality *does not always hold* in the context of Constrained Markov Potential Games (CMPGs). This shows how the problem of safety in multi-agent systems is intrinsically harder with respect to the single-agent one. Moreover, this finding opens up an intriguing avenue for future research, which revolves around the exploration of the specific conditions under which primal-dual methods might offer effective solutions for CMPGs. Unraveling these conditions could shed light on how to adapt and refine optimization techniques for this class of complex problems.

In our quest to address the challenges posed by CMPGs, we have introduced our novel algorithm, **C**oordinate-**A**scent for **CMPG**s with **E**xploration (CA-CMPG-E). This algorithm is not just a theoretical construct but a practical tool that has been proven to converge to an ε-Nash policy in the finite-horizon setting. It's worth noting that the applicability of our algorithm extends beyond this setting. In particular, it can be readily adapted to handle the discounted, infinite-horizon setting by leveraging a suitable Constrained Markov Decision Process (CMDP) solver as a sub-routine. This adaptability underscores the versatility and robustness of our proposed solution, making it applicable in various scenarios.

Furthermore, we have gone a step further in advancing the research in Constrained Markov Games by establishing the *first sample complexity* bound for learning within CMPGs. Our work has shown that in the context of CA-CMPG-E, exploration primarily occurs within the CMDP sub-routines. This observation sparks an interesting avenue for future exploration: studying whether it is possible to optimize the sample complexity bound further in the case of the generative model setting (as elaborated in Sec.6.1) by shifting the exploration process outside the CMDP sub-routines. This intriguing question paves the way for potentially enhancing the efficiency and effectiveness of our algorithm, pushing the boundaries of what is achievable in the realm of CMPGs.

Another interesting future research direction would be to study a lower bound for the sample complexity of the setting without access to the generative model. In fact, our result shows a sample complexity of the order $O\left(\frac{1}{\varepsilon^5}\right)$ and would be interested to understand if this result can be improved or it is due to the complexity of the considered setting.

To conclude, our work establishes the first theoretical guarantees for learning Nash equilibria in CMPGs, which form an important class of safe MARL problems. However, more research needs to be done in order to provide safe and efficient algorithms that can be applied to real-world applications.

Supplementary material: The supplementary material can be found in Alatur et al. [1].

ACKNOWLEDGMENTS

We thank Daniil Dmitriev, Manish Prajapat, Vignesh Ram Somnath and the anonymous reviewers for their valuable comments on the paper. This research was primarily supported by the ETH AI Center. Pragnya Alatur has been funded in part by ETH Foundations of Data Science (ETH-FDS). Giorgia Ramponi is partially funded by Google Brain.

ETHICAL AND SOCIETAL IMPACT

While our work is theoretical and we hope that it will inspire the development of safe MARL algorithms, we do not expect any direct ethical implications from our work.

REFERENCES

[1] Pragnya Alatur, Giorgia Ramponi, Niao He, and Andreas Krause. 2023. Provably Learning Nash Policies in Constrained Markov Potential Games. arXiv:2306.07749 [cs.LG]

[2] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC Press.

[3] Eitan Altman, Thomas Boulogne, Rachid El-Azouzi, Tania Jiménez, and Laura Wynter. 2006. A survey on networking games in telecommunications. *Computers & Operations Research* 33, 2 (2006), 286–311.

[4] Eitan Altman and Adam Shwartz. 2000. Constrained markov games: Nash equilibria. In *Advances in dynamic games and applications*. Springer, 213–221.

[5] Robert J Aumann. 1987. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society* (1987), 1–18.

[6] Michele Breton, Abderrahmane Alj, and Alain Haurie. 1988. Sequential Stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications* 59, 1 (1988), 71–97.

[7] Archana Bura, Aria Hasanzadezonuzy, Dileep Kalathil, Srinivas Shakkottai, and Jean-Francois Chamberland. 2022. DOPE: Doubly Optimistic and Pessimistic Exploration for Safe Reinforcement Learning. https://openreview.net/forum?id=U4BUMoVTrB2

[8] Zhiyuan Cai, Huanhui Cao, Wenjie Lu, Lin Zhang, and Hao Xiong. 2021. Safe Multi-Agent Reinforcement Learning through Decentralized Multiple Control Barrier Functions. *CoRR* abs/2103.12553 (2021). arXiv:2103.12553 https://arxiv.org/abs/2103.12553

[9] Dingyang Chen, Qi Zhang, and Thinh T. Doan. 2022. Convergence and Price of Anarchy Guarantees of the Softmax Policy Gradient in Markov Potential Games. https://openreview.net/forum?id=pe2ZGTUxVvJ

[10] Qiwen Cui, Kaiqing Zhang, and Simon Du. 2023. Breaking the curse of multiagents in a large state space: Rl in markov games with independent linear function approximation. In *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, 2651–2652.

[11] Raghuram Bharadwaj Diddigi, Sai Koti Reddy Danda, Prabuchandran K. J., and Shalabh Bhatnagar. 2019. Actor-Critic Algorithms for Constrained Multi-agent Reinforcement Learning. *CoRR* abs/1905.02907 (2019). arXiv:1905.02907 http://arxiv.org/abs/1905.02907

[12] Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. 2022. Independent Policy Gradient for Large-Scale Markov Potential Games: Sharper Rates, Function Approximation, and Game-Agnostic Convergence. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 5166–5220. https://proceedings.mlr.press/v162/ding22b.html

[13] Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. 2022. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *International Conference on Machine Learning*. PMLR, 5166–5220.

[14] Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. 2021. Provably efficient safe exploration via primal-dual policy optimization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3304–3312.

[15] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo R Jovanovic. 2020. Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes.. In *NeurIPS*.

[16] Ingy Elsayed-Aly, Suda Bharadwaj, Chris Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. 2021. Safe Multi-Agent Reinforcement Learning via Shielding. In *Adaptive Agents and Multi-Agent Systems*. https://api.semanticscholar.org/CorpusID:231719044

[17] Francisco Facchinei and Christian Kanzow. 2010. Generalized Nash equilibrium problems. *Annals of Operations Research* 175, 1 (2010), 177–211.

[18] Roy Fox, Stephen M Mcaleer, Will Overman, and Ioannis Panageas. 2022. Independent natural policy gradient always converges in Markov potential games. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4414–4425.

[19] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. 2023. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. arXiv:2205.10330 [cs.AI]

[20] Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. 2022. Global Convergence of Multi-Agent Policy Gradient in Markov Potential Games. In *International Conference on Learning Representations*. https://openreview.net/forum?id=gfwON7rAm4

[21] Chenyi Liu, Nan Geng, Vaneet Aggarwal, Tian Lan, Yuan Yang, and Mingwei Xu. 2021. CMIX: Deep Multi-agent Reinforcement Learning with Peak and Average Constraints. In *Proceedings of the 2021 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2021), Virtual Conference*. 13–17.

[22] Tao Liu, Ruida Zhou, Dileep Kalathil, PR Kumar, and Chao Tian. 2021. Learning Policies with Zero or Bounded Constraint Violation for Constrained MDPs. *arXiv preprint arXiv:2106.02684* (2021).

[23] Songtao Lu, Kaiqing Zhang, Tianyi Chen, Tamer Başar, and Lior Horesh. 2021. Decentralized Policy Gradient Descent Ascent for Safe Multi-Agent Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 10 (May 2021), 8767–8775. https://doi.org/10.1609/aaai.v35i10.17062 Number: 10.

[24] Chinmay Maheshwari, Manxi Wu, Druv Pai, and Shankar Sastry. 2022. Independent and Decentralized Learning in Markov Potential Games. (2022). arXiv:2205.14590 http://arxiv.org/abs/2205.14590

[25] Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Basar. 2022. On Improving Model-Free Algorithms for Decentralized Multi-Agent Reinforcement Learning. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 15007–15049. https://proceedings.mlr.press/v162/mao22a.html ISSN: 2640-3498.

[26] Dov Monderer and Lloyd S Shapley. 1996. Potential games. *Games and economic behavior* 14, 1 (1996), 124–143.

[27] John F Nash et al. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36, 1 (1950), 48–49.

[28] P. Parnika, Raghuram Bharadwaj Diddigi, Sai Koti Reddy Danda, and Shalabh Bhatnagar. 2021. Attention Actor-Critic Algorithm for Multi-Agent Constrained Co-Operative Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1616–1618.

[29] Santiago Paternain, Luiz FO Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. 2019. Constrained reinforcement learning has zero duality gap. *arXiv preprint arXiv:1910.13393* (2019).

[30] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2016. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. arXiv:1610.03295 [cs.AI]

[31] Ziang Song, Song Mei, and Yu Bai. 2021. When Can We Learn General-Sum Markov Games with a Large Number of Players Sample-Efficiently? (2021). arXiv:2110.04184 http://arxiv.org/abs/2110.04184

[32] Sharan Vaswani, Lin Yang, and Csaba Szepesvari. 2022. Near-Optimal Sample Complexity Bounds for Constrained MDPs. https://openreview.net/forum?id=ZJ7Lrtd12x_

[33] Koji Yamamoto. 2015. A Comprehensive Survey of Potential Game Approaches to Wireless Networks. *IEICE Transactions on Communications* E98.B, 9 (2015), 1804–1823. https://doi.org/10.1587/transcom.E98.B.1804

[34] Yaodong Yang and Jun Wang. 2021. An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective. http://arxiv.org/abs/2011.00583 arXiv:2011.00583 [cs].

[35] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control* (2021), 321–384.

[36] Runyu Zhang, Jincheng Mei, Bo Dai, Dale Schuurmans, and Na Li. 2022. On the Global Convergence Rates of Decentralized Softmax Gradient Play in Markov Potential Games. https://openreview.net/forum?id=X1oVDZIABwF

[37] Runyu Zhang, Zhaolin Ren, and Na Li. 2021. Gradient play in stochastic games: stationary points, convergence, and sample complexity. http://arxiv.org/abs/2106.00198 arXiv:2106.00198 [cs, math].