Constrained Multiagent Reinforcement Learning for Large Agent Population

Jiajing Ling[1], Arambam James Singh[2], Nguyen Duc Thien[1], and Akshat Kumar[1]

[1] School of Computing and Information Systems, Singapore Management University
{jjling.2018, akshatkumar}@smu.edu.sg, neverdie0000@gmail.com
[2] School of Computing, National University of Singapore
jamesa@nus.edu.sg

Abstract. Learning control policies for a large number of agents in a decentralized setting is challenging due to partial observability, uncertainty in the environment, and scalability challenges. While several scalable multiagent RL (MARL) methods have been proposed, relatively few approaches exist for large scale *constrained* MARL settings. To address this, we first formulate the constrained MARL problem in a collective multiagent setting where interactions among agents are governed by the aggregate count and types of agents, and do not depend on agents' specific identities. Second, we show that standard Lagrangian relaxation methods, which are popular for single agent RL, do not perform well in constrained MARL settings due to the problem of credit assignment—how to identify and modify behavior of agents that contribute most to constraint violations (and also optimize primary objective alongside)? We develop a fictitious MARL method that addresses this key challenge. Finally, we evaluate our approach on two large-scale real-world applications: maritime traffic management and vehicular network routing. Empirical results show that our approach is highly scalable, can optimize the cumulative global reward and effectively minimize constraint violations, while also being significantly more sample efficient than previous best methods.

Keywords: Multi-agent Systems · Multiagent reinforcemene learning · Constraint optimization.

1 Introduction

Sequential mutiagent decision making allows multiple agents operating in an uncertain, partially observable environment to take coordinated decision towards a long term goal [4].The decentralized partially observable MDP (Dec-POMDP) model [20] has emerged as a popular framework for cooperative multiagent control problems with several applications in multiagent robotics [2], packet routing in networks [12], and vehicle fleet optimization [17,32]. However, solving Dec-POMDPs optimally is computationally intractable even for a small two-agent system [4]. When the planning model is not known, multiagent reinforcement learning (MARL) for Dec-POMDPs also suffers from scalability

challenges. However, good progress has been made recently towards scalable MARL methods [21,24,34,36].

To address the complexity, various models have been explored where agent interactions are limited by design by enforcing various conditional and contextual independencies such as transition and observation independence among agents [16] and event driven interactions [3]. However, their impact remains limited due to narrow application scope. To address practical applications, recently introduced multiagent decision theoretic frameworks (and corresponding MARL algorithms) model the behavior of a population of nearly identical agents operating collaboratively in an *uncertain* and *partially observable* environment. The key enabling insight and related assumption is that in several urban environments (such as transportation, supply-demand matching) agent interactions are governed by the aggregate count and types of agents, and do not depend on the specific identities of individual agents. Several scalable methods have been developed for this setting such as mean field RL [26,27,28,36], collective Dec-POMDPs [17,18,35], anonymity based multiagent planning and learning [32,33] among others [10].

A key challenge in MARL is that of multiagent credit assignment, which enables different agents to deduce their individual contribution to the team's success, and is challenging in large multiagent systems [6,31]. Recently, there has been progress in addressing this issue for large scale MARL [8,19]. However, such previous methods address the credit assignment problem in a *constraint-free* setting. With the introduction of constraints, we need to perform credit assignment jointly both for primary objective and for the cost incurred by constraints, and deduce accurately the role of each agent in optimizing the primary objective, and lowering constraint violations. In our work, we develop novel techniques that address this issue for *constrained* collective MARL settings.

Constrained RL Most existing works focus on single agent constrained RL and deal with cumulative constraints (discounted and mean valued). The most common approach to solve this problem is the Lagrangian relaxation (LR) [5]. The constrained RL problem is converted to an unconstrained one by adding Lagrangian multipliers, and both Lagrange multipliers and policy parameters are updated iteratively [30]. Methods such as CPO [1] extend the trust region optimization to the constrained RL setting and solve an approximate quadratically constrained problem for policy updates. IPO [13] algorithm uses a logarithm barrier function as the penalty to the original objective to force the constraint to be satisfied. Forming a max-min problem by constructing a lower bound for the objective is used in [9].

Although the above mentioned approaches can solve the single agent constrained RL well, extending these approaches such as LR to multiagent constrained RL directly is not trivial. Since credit assignment for costs also remains unsolved, searching for a policy that satisfies the constraints becomes challenging. There are few works aiming to solve multi-agent constrained RL. [7] used the LR method and proposed to learn a centralized policy critic and penalty critics to guide the update of policy parameters and Lagrangian multipliers. However, centralized critics can be noisy since contributions from each individual agents are not clear. [14]

also proposed LR but in a setting where agents are allowed to communicate over a pre-defined communication network (in contrast, our method requires no communication during policy execution). The most recent work CMIX [12] combines the multi-objective programming and Q-mix framework [8]. However, scalability is still a big challenge since different Q-function approximators for each constraint and each agent are required. To summarize, LR is one of the most common approach to solve both single and multiagent constrained RL. However, how to decide the credit assignment with respect to constraint costs and how to scale to large-scale multiagent systems still remain challenging.

Our contributions First, we formulate the MARL problem for settings where agent interactions are primarily governed by the aggregate count and types of agents using the collective Dec-POMDP framework [17,18] augmented with constraints. Second, we develop a fictitious constrained MARL method which is also based on Lagrangian relaxation, but addresses the issue of credit assignment for both primary objective and constraints. Finally, we test on both real world and synthetic datasets for the maritime traffic management problem [25], and network routing problem [12]. We show that our method is significantly better in satisfying constraints than the standard LR method for MARL. Similarly, when compared against CMIX [12], our approach reduces both average and peak constraint violations to within the threshold using significantly lower number of samples, while achieving similar global objective.

2 Fictitious Constrained Reinforcement Learning

2.1 Collective $\mathbb{C}$Dec-POMDP

We consider the collective decentralized POMDP ($\mathbb{C}$Dec-POMDP) framework to model multi-agent systems (MAS) where the transition and the reward of each individual agent depends on the number (count values) of agents in different local states. $\mathbb{C}$Dec-POMDP MAS has a wide range of applications in many real world domains such as traffic control, transport management or resource allocation [17,25,35]. Formally, a $\mathbb{C}$Dec-POMDP model is defined by:

- A finite planning horizon H.
- The number of agents M. An agent m can be in one of the states in the state space S. We denote a single state as $i \in S$. We assume that different agents share the same state space S. Therefore, the joint state-space is $\boldsymbol{S} = S^M$.
- A set of actions A for each agent m. We denote an individual action as $j \in A$.
- $\boldsymbol{s}_t, \boldsymbol{a}_t$ denote the joint state and joint action of agents at time t.
- Let $(s_{1:H}, a_{1:H})^m = (s_1^m, a_1^m, s_2^m \ldots, s_H^m, a_H^m)$ denote the complete state-action trajectory of an agent m. We denote the state and action of agent m at time t using random variables s_t^m, a_t^m. We use the individual indicator function $\mathbb{I}(s_t^m = i, a_t^m = j) \in \{0; 1\}$ to indicate whether the agent m is in local state i and taking action j at time step t. Other indicators are defined similarly. Given different indicator functions, the count variables are defined as follows:
  - $\mathrm{n}_t(i, j, i') = \sum_{m=1}^{M} \mathbb{I}(s_t^m = i, a_t^m = j, s_{t+1}^m = i')\ \forall i, i' \in S, j \in A$

- $\mathrm{n}_t(i,j) \quad = \sum_{m=1}^{M} \mathbb{I}(s_t^m = i, a_t^m = j) \qquad \forall i \in S, j \in A$
- $\mathrm{n}_t(i) \quad = \sum_{m=1}^{M} \mathbb{I}(s_t^m = i) \;\; \forall i \in S$

When states and actions are not specified, we denote the state count table as $\mathrm{n}_t^{\mathrm{s}} = (\mathrm{n}_t(i)\ \forall i \in S)$, state-action count table as $\mathrm{n}_t^{\mathrm{sa}} = (\mathrm{n}_t(i,j)\ \forall i \in S, j \in A)$ and transition count table as $\mathrm{n}_t = (\mathrm{n}_t(i,j,i')\ \forall i, i' \in S, j \in A)$. For a given subset $S' \subseteq S$, we define the count table for agents in S' as $\mathrm{n}_t(S') = (\mathrm{n}_t(i,j,i')\ \forall i \in S', j \in A, i' \in S)$.

- The local transition function of an individual m is $P\big(s_{t+1}^m | s_t^m, a_t^m, \mathrm{n}_t^{\mathrm{sa}}\big)$. The transition function is the same for all the agents. Note that it is also affected by $\mathrm{n}_t^{\mathrm{sa}}$, which depends on the collective behavior of the agent population.
- Each agent m has a policy $\pi_t^m(j|i, \mathrm{n}_t^{\mathrm{s}})$ denoting the probability of agent m taking action j given its local state i and the count table $\mathrm{n}_t^{\mathrm{s}}$. Note that when agent cannot fully observe the whole count table, we can model an observation function $o(i, \mathrm{n}_t^{\mathrm{s}})$ as a non-trainable component of π. When agents have the same policy, we can ignore the index and denote the common policy with π.
- Initial state distribution, $b_o = (P(i) \forall i \in S)$, is the same for all agents.

We define a set of reward functions $r_l(\mathrm{n}_t)$, $l = 1 : L$ and a set of cost functions $c_k(\mathrm{n}_t)$, $k = 1 : K$ that depend on the count variables n_t. Our goal is to solve a collective constrained program:

$$\max_{\pi_\theta} \quad V(\pi_\theta) = \mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H}\sum_{l} r_l(\mathrm{n}_t)|\pi_\theta\Big] \tag{1a}$$

$$\text{s.t.} \quad \mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H} c_k(\mathrm{n}_t)|\pi_\theta\Big] \geq 0, \qquad \forall k \tag{1b}$$

Agents with types We can also associate different types with different agents to distinguish them (e.g., 4-seater taxi, 6-seater taxi). This can be done using a type-augmented state space as $S' = S \times \mathcal{T}$, where $\mathcal{T}$ is the set of possible agent types. The main benefit of the collective modeling is that we can exploit the aggregate nature of interactions among agents when the number of types is much smaller than the number of agents

Simulator for MARL In the MARL setting, we do not have access to the transition and reward function. As shown in [18], a count based simulator provides the experience tuple for the centralized learner as $(\mathrm{n}_t^{\mathrm{s}}, \mathrm{n}_t^{\mathrm{sa}}, \mathrm{n}_t, r_t)$. In other words, simulation and learning in the collective setting can be done at the abstraction of counts. This avoids the need to keep track of individual agents' state-action trajectories, and increases the computational scalability to large number of agents.

2.2 Individual Value Representation

Solving Problem(1) is difficult because the constraints are globally coupled with the joint counts $\mathrm{n}_{1:H}$. In many domains in practice, the reward and cost functions only involve the count variables over a subset of states S. For example, in

congestion domain, we have the penalty cost defined for a specific area/zone. We consider a general framework where we can define the subset $S_k \subseteq S$ that affects constraint k, and the subset $S_l \subseteq S$ that affects reward r_l. In extreme case where a function is non-decomposable, S_k can be set to S. Let $\mathrm{n}_t(S_l)$ denote count table that summarizes the distribution of agents in states $s \in S_l$ (as defined in section 2.1). Let $|\mathrm{n}_t(S_l)|$ denote the number of agents in S_l. We can re-write (1) as follows:

$$\max_{\pi_\theta} \quad V(\pi_\theta) = \mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H}\sum_{l} r_l(\mathrm{n}_t(S_l))|\pi_\theta\Big] \tag{2a}$$

$$\text{s.t.} \quad \mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H} c_k(\mathrm{n}_t(S_k))|\pi_\theta\Big] \geq 0, \qquad \forall k \tag{2b}$$

Furthermore, we show that we can re-write the global constrained program in the form of an individual agent's constrained program. For a specific function f_l, which can be either cost $f_l = c_l(S_l)$ or reward function $f_l = r_l(S_l)$, we define an auxiliary individual function:

$$f_l^m\big(s_t^m, a_t^m, \mathrm{n}_t(S_l)\big) = \begin{cases} \frac{f_l\left(\mathrm{n}_t(S_l)\right)}{|\mathrm{n}_t(S_l)|} & \text{if } s_t^m \in S_l \\ 0 & \text{otherwise} \end{cases}$$

We use $\mathbf{s}_{1:H}^m, \mathbf{a}_{1:H}^m$ to denote the state-action trajectory with length H of agent m, and use $\mathbf{s}_{1:H}, \mathbf{a}_{1:H}$ to denote the join state-action trajectory of all M agents in our system.

Proposition 1. *Consider any reward/cost component f_l. The global expected value of f_l is equal to a factor of individual value function:*

$$\mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H} f_l(\mathrm{n}_t(S_l))|\pi_\theta\Big] = M \times \mathbb{E}_{\mathbf{s}_{1:H},\mathbf{a}_{1:H}}\Big[\sum_{t=1}^{H} f_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l))\Big] \tag{3}$$

Proof. By applying the exchangeability theorem from [17], we can derive the individual function for reward/cost component f_l as:

$$\begin{aligned} &\mathbb{E}_{\mathbf{s}_{1:H},\mathbf{a}_{1:H}}[\sum_{t=1}^{H} f_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l))] \\ =&\sum_{t=1}^{H} \mathbb{E}_{\mathbf{s}_{1:H},\mathbf{a}_{1:H}}[f_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l))] \end{aligned} \tag{4}$$

We replace the joint probability $P(\mathbf{s}_{1:H}, \mathbf{a}_{1:H})$ with $P(\mathbf{s}_{1:t}^m, \mathbf{a}_{1:t}^m, \mathrm{n}_{1:t})$.

$$= \sum_{t=1}^{H} \sum_{\mathbf{s}_{1:t}^m, \mathbf{a}_{1:t}^m, \mathrm{n}_{1:t}} P(\mathbf{s}_{1:t}^m, \mathbf{a}_{1:t}^m, \mathrm{n}_{1:t}) f_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l)) \tag{5}$$

$$= \sum_{t=1}^{H} \sum_{\mathbf{s}_{1:t}^m, \mathbf{a}_{1:t}^m, \mathrm{n}_{1:t}} P(\mathbf{s}_{1:t}^m, \mathbf{a}_{1:t}^m, \mathrm{n}_{1:t}) \sum_{s' \in S_l} \mathbb{I}(s_t^m = s') \frac{f_l(\mathrm{n}_t(S_l))}{|\mathrm{n}_t(S_l|} \tag{6}$$

We now apply the exchangeability of agents with respect to the count variables n [17]:

$$= \sum_{t=1}^{H} \sum_{\mathrm{n}_{1:t}} P(\mathrm{n}_{1:t}) \sum_{s' \in S_l} \frac{\mathrm{n}_t(s')}{M} \frac{f_l(\mathrm{n}_t(S_l))}{|\mathrm{n}_t(S_l|} \tag{7}$$

$$= \frac{1}{M} \mathbb{E}_{\mathrm{n}_{1:H}} \Big[\sum_{t=1}^{H} f_l(\mathrm{n}_t(S_l)) | \pi_\theta \Big] \tag{8}$$

□

By applying Proposition 1 to reward and cost functions in (2), we have the following lemma.

Lemma 1. *Solving a collective constrained reinforcement learning problem defined in* (1) *is equivalent to solve the individual constrained reinforcement learning problem defined as follows:*

$$\max_{\pi_\theta} \quad V^m(\pi_\theta) = \mathbb{E}_{\mathbf{s}_{1:H}, \mathbf{a}_{1:H}} \Big[\sum_{t=1}^{H} \sum_{l} r_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l)) | \pi_\theta \Big] \tag{9a}$$

$$\text{s.t.} \quad \mathbb{E}_{\mathbf{s}_{1:H}, \mathbf{a}_{1:H}} \big[\sum_{t=1}^{H} c_k^m(s_t^m, a_t^m, \mathrm{n}_t(S_k)) \big] \geq 0, \qquad \forall k \tag{9b}$$

To solve Prolem (9), we apply fictitious-play [15] based constrained optimization (FICO) in which at each iteration, agent tries to optimize its own policy given the joint state-action samples and ignore the effect of its policy change on other agents. Amongst popular methods to solve constrained RL, in this work we apply the Lagrange relaxation method to solve FICO.

We also highlight that Problem (9) is the key to performing the credit assignment for primary objective and constraints. This problem clearly separates out the contribution of each agent m to the value function (or V^m) and each constraint k (or c_k^m). Therefore, the FICO method enables effective credit assignment for both primary objective and constraints.

2.3 Fictitious Collective Lagrangian Relaxation

We consider applying Lagrangian relaxation to solve FICO (9). The Lagrange dual problem is given as follows.

$$\min_{\lambda \geq 0} \max_{\pi_\theta} \mathbb{E}_{\mathbf{s}_{1:H}, \mathbf{a}_{1:H}} \Big[\sum_{t=1}^{H} \sum_{l} r_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l)) + \sum_{k} \lambda_k c_k^m(s_t^m, a_t^m, \mathrm{n}_t(S_k)) | \pi_\theta \Big] \tag{10}$$

To solve this dual Problem (10), we apply stochastic gradient ascent-descent to alternatively update parameters θ of the policy and the Lagrange multiplier λ following the two-time scale approximation [30].

Individual policy update To optimize π_θ, we first compute the modified reward as follows.

$$R(s_t^m, a_i^m, \mathrm{n}_t) = \sum_l r_l^m(s_t^m, a_t^m, \mathrm{n}_t(S_l)) + \sum_k \lambda_k c_k^m(s_t^m, a_t^m, \mathrm{n}_t(S_k))$$

Given the fixed Lagrange multipliers, parameters θ are optimized by solving the following problem,

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{s}_{1:H}, \mathbf{a}_{1:H}} \Big[\sum_{t=1}^{H} R(s_t^m, a_i^m, \mathrm{n}_t) | \pi_\theta \Big] \tag{11}$$

The benefit of the above representation is that now we can apply various techniques developed for collective Dec-POMDPs to optimize (11) using stochastic gradient ascent. To optimize (11), we consider a fictitious play approach to compute policy gradient for policy π_θ of agent m over all possible local state i and individual action j. Using the standard policy gradient [29] with respect to an individual agent m, we can perform the update for θ as follows:

$$\begin{aligned}\theta' =& \theta + \alpha_\theta \sum_t \sum_{i,j,\mathrm{n}_t} \mathbb{E}_{\mathbf{s}_{t:H}, \mathbf{a}_{t:H}} \Big[\mathbb{I}(s_t^m = i, a_t^m = j) \mathbb{I}(\mathrm{n}_t \sim \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \\ & \times \sum_{T=t}^{H} R(s_T^m, a_T^m, \mathrm{n}_T) | \pi_\theta \Big] \nabla_\theta \log \pi_\theta(j | i, \mathrm{n}_t)\end{aligned} \tag{12}$$

where $\mathbb{I}(\mathrm{n}_t \sim \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ is an indicator function for whether the count table of the joint transition $\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t$ is identical to n_t. Applying results from [17] for collective Dec-POMDPs, we can sample the counts (using the current policy) and use these counts to compute the gradient term in (12) as:

$$\begin{aligned}&\mathbb{E}_{\mathbf{s}_{t:H}, \mathbf{a}_{t:H}} \Big[\mathbb{I}(s_t^m = i, a_t^m = j) \mathbb{I}(\mathrm{n}_t \sim \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \sum_{T=t}^{H} R(s_T^m, a_T^m, \mathrm{n}_T) | \pi_\theta \Big] \\ =& \sum_{\boldsymbol{n}'_{1:H}} P(\boldsymbol{n}'_{1:H}) \mathbb{I}(\mathrm{n}'_t = \mathrm{n}_t) \frac{n'_t(i,j)}{M} \sum_{T=t}^{H} \sum_{s_T^m, a_T^m} \frac{n'_t(s_T^m, a_T^m)}{M} R\big(s_T^m, a_T^m, \mathrm{n}'_T\big)\end{aligned} \tag{13}$$

The above expected value can be estimated by a Monte-Carlo approximation $\hat{Q}_t^m(i, j, \mathrm{n}_t)$ with samples $\xi = 1, \ldots, K$ of counts [17]. For a given count sample $\mathrm{n}_{1:H}^\xi$:

$$V_H^\xi(i,j) = R_H(i, j, n_H^\xi(i)) \tag{14}$$

$$V_t^\xi(i,j) = R_t(i, j, n_t^\xi(i)) + \sum_{j'} \frac{n_t^\xi(i, j', i')}{n_t^\xi(i)} V_{t+1}^\xi(i', j') \tag{15}$$

$$Q_t^\xi(i,j) = \frac{n_t^\xi(i,j)}{M} \times V_t^\xi(i,j) \tag{16}$$

$$\hat{Q}_t^m(i, j, \mathrm{n}_t) = \frac{1}{K} \sum_{\xi | \mathrm{n}^\xi = \mathrm{n}_t} Q_t^\xi(i,j) \tag{17}$$

Continuous actions We highlight that even though we have formulated FICO for discrete action spaces, our method works for continuous action space also as long as the policy gradient analogue of (12) is available for continuous actions. Empirically, we do test on the maritime traffic control problem where action space is continuous, and policy gradient is derived in [25].
Lagrange multiplier update Given a fixed policy π_θ, the Lagrange multiplier λ_k for each constraint k is optimized by solving the following problem.

$$\min_{\lambda_k} \mathbb{E}_{\mathbf{s}_{1:H},\mathbf{a}_{1:H}}\Big[\sum_{t=1}^{H} \lambda_k c_k^m(s_t^m, a_t^m, \mathrm{n}_t(S_k))|\pi_\theta\Big] \tag{18}$$

We re-write the objective in the collective way as follows.

$$\begin{aligned}
&=\lambda_k \sum_{s'\in S_k}\sum_{t=1}^{H}\sum_{\mathbf{s}^m_{1:t},\mathbf{a}^m_{1:t},\mathrm{n}_{1:t}} \mathbb{I}(s_t^m = s')P(\mathbf{s}^m_{1:t},\mathbf{a}^m_{1:t},\mathrm{n}_{1:t})\frac{c_k(\mathrm{n}_t(S_k))}{|\mathrm{n}_t(S_k)|}\\
&=\lambda_k \frac{1}{M}\mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H} c_k(\mathrm{n}_t(S_k))|\pi_\theta\Big] \qquad (19)
\end{aligned}$$

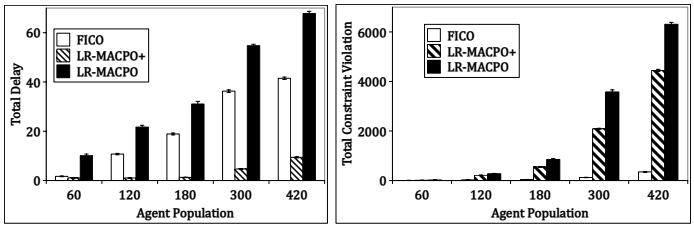
By applying gradient descent, we have $\lambda_k' = \lambda_k - \alpha_k \frac{1}{M}\mathbb{E}_{\mathrm{n}_{1:H}}\Big[\sum_{t=1}^{H} c_k(\mathrm{n}_t(S_k))|\pi_\theta\Big]$ where α_k is the learning rate for the update of λ_k.

3 Experiments

In the section, we evaluate our proposed approach FICO on two real-world tasks: Maritime traffic management (MTM) and vehicular network routing problem with a large scale of agent population. For the MTM problem, we compare FICO with two baseline approaches LR-MACPO and LR-MACPO+. LR-MACPO is a Lagrangian based approach without any credit assignment. The policy in this case is trained with global reward and global cost signal, similar to RCPO algorithm [30]. LR-MACPO+ is also a standard Lagrangian based approach with credit assignment only for the reward signal but not for the cost function in constraint. The detailed problem formulations for LR-MACPO and LR-MACPO+ are provided in the supplementary. We compare FICO with CMIX [12] in the vehicular network routing problem. Since CMIX only deals with discrete action space, we did not evaluate CMIX in the MTM problem where the action space is continuous. Our code is publicly available (link in supplementary).

3.1 Maritime traffic management

The main objective in the MTM problem is to minimize the travel delay incurred by vessels while transiting busy port waters and also to reduce the congestion developed due to uncoordinated movement of vessels. The previous formulations of the MTM problem in [23,25] involved unconstrained policy optimization—the



(a) Total delay (b) Total constraint violation

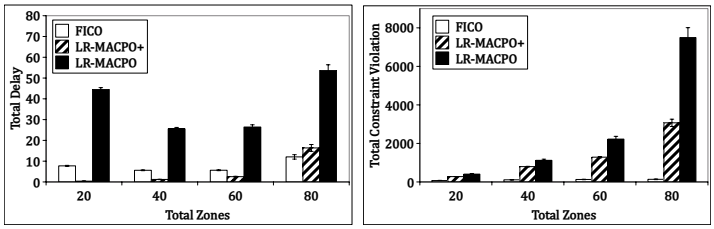
Fig. 1: Results on the map with 23 zones and different agent population. The lower is the better for both metrics.

objective is a weighted combination of the delay and congestion costs. This requires an additional tuning of the weight parameters in the objective. We propose a new MTM formulation using constrained MARL as in (1). In our formulation, we introduce constraints that the cumulative congestion cost should be within a threshold and minimize the travel delay as the main objective. The new formulation with constraint is more interpretable, and avoids the intensive search over weight parameters to formulate a single objective as in previous models [25]. Additional description of the formulation is in supplementary.

We evaluate our constrained based approach of the MTM problem in both synthetic and real-data instances. In synthetic data experiments we test the scalability and robustness of our proposed algorithm. Real-data instances are used to measure the effectiveness of the approach in a real-world problem.

Synthetic data instances For synthetic data experiments, we first randomly generate directed graphs (provided in supplementary) similar to the procedure described in [23]. The edge of the graph represents a zone, vessels move from left to right through the zones. Each zone has some capacity i.e the maximum number of vessels the zone can accomodate at any time. Each zone is also associated with a minimun and maximum travel time to cross the zone. Vessels arrive at the source zone following an arrival distribution, and its next heading zone is sampled from a pre-determined distribution. More details on the experimental settings are provided in supplementary.

We first evaluate the scalability of our approach with varying agent population size from 60 agents to 420 agents on the map with 23 zones. We show the results on total delay and total constraint violation respectively as in Fig. 1(a) and Fig. 1(b). Delay is computed as the difference between actual travel time and minimum travel time in the zone. Total violation computes the total constraint violations over all zones. X-axis denotes the agent population size in both the subfigures, and y-axis denotes the total delay and total constraint violation in (a) and (b) respectively. We observe LR-MACPO baseline perform poorly than other approaches in terms both the metric of delay and violation. This is because LR-MACPO is trained with global system reward and global cost function, which is without any credit assignment technique. LR-MACPO+'s performance on delay metric is superior than our approach FICO, but it suffers severely on



(a) Total delay (b) Total constraint violation

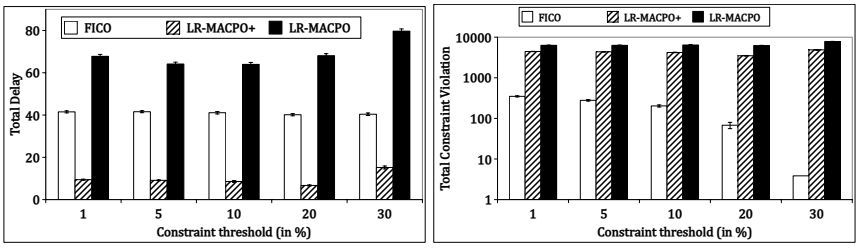
Fig. 2: Results on agent population 300 and different maps

violation metric in Fig. 1(b). This is an expected result because in LR-MACPO+ credit assignment is provided only for the reward signal not for the cost signal. This makes each agent's credit for the cost component to be noisy resulting in ineffective handling of the constraints. Also, low delay benefits from the high constraint violation in LR-MACPO+.

We next evaluate the robustness of our approach with different maps with varying number of zones (from 20 zones to 80 zones). As in Fig. 2, x-axis denotes different maps with varying number of zones in both the subfigures, and y-axis denoted the total delay and total constraint violation in (a) and (b) respectively. In this experiment the complexity of the problem increases with the increasing number of zones. We observe that our approach FICO is able to reduce the violation consistently for all the settings as shown in Fig. 2(b). In settings with less than 80 zones, LR-MACPO+ beats our approach in terms of total delay. However, it fails to satisfy the constraints poorly. We see that at the most difficult setting with 80 zones, our approach performs better than LR-MACPO+ in terms of both total delay and constraint violation.

Finally, we evaluate the robustness of our approach with different constraint thresholds on the map with 23 zones and 420 agents. The constraint threshold specifies the upper bound of cumulative resource violation over the horizons. The constraint threshold is defined as a percentage of the total resource violations over the horizons when agents are moving with the fastest speed. As shown in Fig. 3, x-axis denotes different constraint thresholds in both the subfigures, and y-axis denotes the total delay and total constraint violation in (a) and (b) respectively. We observe that FICO performs better than LR-MACPO baseline in terms of both the metric of delay and constraint violation. In Fig. 3(b), we see that FICO performs better than other baselines consistently over different constraint thresholds. With the increase of constraint threshold, the total constraint violation is decreasing. FICO is almost able to make the constraint satisfied with the loosest constraint threshold (30%). The constraint violation comes from the zone in the middle of the map which is the busiest zone.

Real data instances We also evaluated our proposed approach on real-world data instances from Singapore strait. The strait is considered to be one of the busiest in the world. It connects the maritime traffic of South China Sea and



(a) Total delay (b) Total constraint violation

Fig. 3: Results on agent population 420 and 23 zones with different constraint threshold

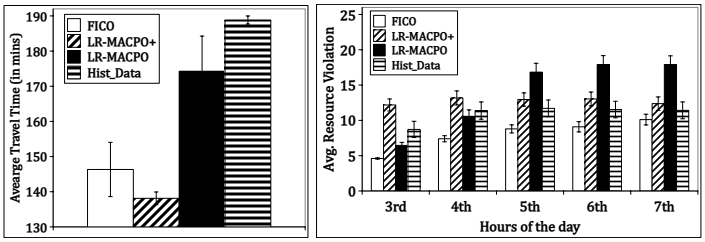
Indian Ocean. We use 6 months (2017-July - 2017-December) of historical AIS data of vessel movement in Singapore strait. Each AIS record consists of vital navigation information such as lat-long, speed over ground, heading etc, and is logged every 15 seconds. In our evaluation we mainly focus on tankers and cargo vessel types because majority of traffic belongs to these two types.

Fig. 6(a) shows the electronic navigation chart of Singapore strait. Vessels enter and leave the strait through one-way sea lanes called traffic separation scheme (TSS). It is created for easy transit of vessels in the strait and helps in minimizing any collision risks. TSS is further sub-divided into smaller zones for better management of the traffic. From the total datasets of 6 months (180 days), 150 days are used for training and 30 days for testing.

Training From the historical data, we first estimate the problem instance parameters such as capacity of each zones, minimum and maximum travel time in each zone. The simulator that we use is the same as in [25], and is publicly available at [22].

The capacity of a zone is computed as 60% of the maximum number of vessels present at any time in the zone overall all days. Each zone can have a different capacity value. We treat the physical sea space in a zone as a resource. Each vessel occupies 1 unit of resource of that zone. The constraint for each zone is expressed as the cumulative resource violation over time should be within a threshold. There are also other problem parameters which are specific to a particular day such as vessels' arrival time on the strait and initial count distribution of vessels present at the strait in beginning of the day. For each training day we estimate the two parameters. Our constrained based policy FICO is trained on varying scenarios of training days. From historical data we observe that there are peak hour periods of traffic intensity during 3rd - 7th hour of the day. Therefore, in our evaluation we focus on optimizing the peak hour periods.

Testing We test our trained policy on separate 30 testing days. Fig. 4(a) shows the results of average travel time of vessels crossing the strait averaged over 30 test days. We observe all the three baselines achieve better travel time than the historical data baseline Hist-Data. LR-MACPO performs poorly among the three. This is because LR-MACPO is trained with the system reward and cost



(a) Average travel time (b) Average resource violation

Fig. 4: Results on maritime real-data over 30 testing days. (lower is better for both metrics)

.

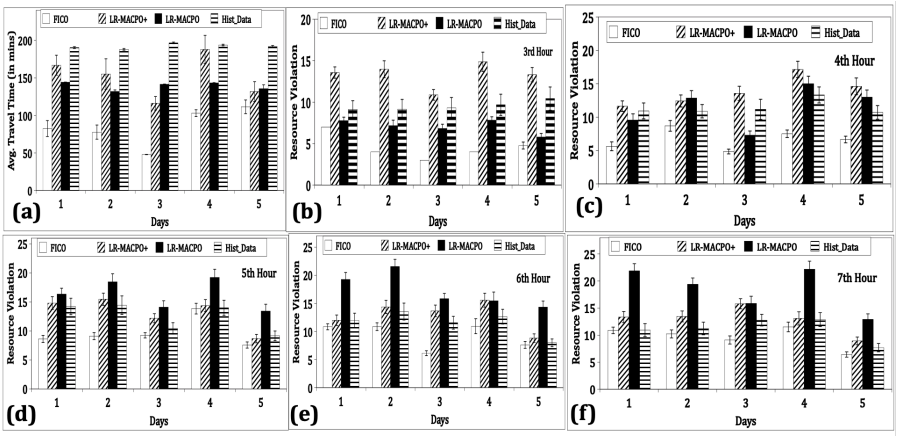
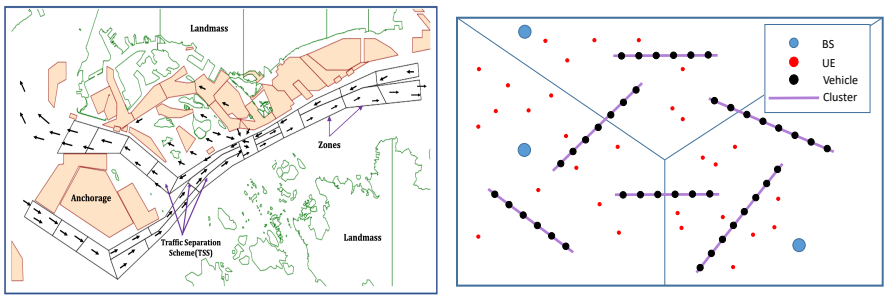


Fig. 5: (a) Average travel time(lower is better) (b-f) Resource violation for peak hours 3rd-7th (lower is better)

signals which are without any credit assignment techniques. We also observe that LR-MACPO+ baseline is able to further reduce the travel time slightly better than our approach FICO but at the expense of higher resource violation as seen in Fig. 4(b).

Results in Fig. 4(b) show the average violation of resource over 30 testing days. X-axis denotes the peak hour periods. During the peak hour period, FICO achieves reduced violation of resource among all the baselines. Since LR-MACPO+ lacks the credit assignment signal on the cost function it performs sub-optimally than FICO. The results in Fig. 4 validate the benefit of providing efficient credit assignment technique to both reward and cost function.

In Fig. 5 we show the results of top 5 busiest testing days and results for remaining 25 days are provided in supplementary. Fig. 5(a) shows the results for travel time, x-axis denotes days and y-axis denotes average travel time for crossing the strait. Fig. 5(b-f) show the results of resource violation during peak



(a) (b)

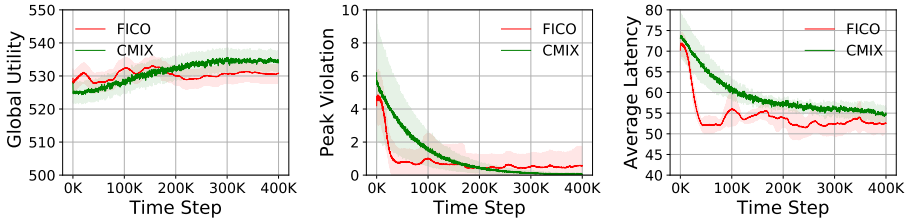
Fig. 6: (a) Electronic navigation chart(ENC) of Singapore Strait; (b)Vehicular network model (adapted from [11])

hour periods (3rd-7th hour), x-axis denotes the days and y-axis denotes the resource violation. In all 5 days and during the peak hour periods, our approach achieves an improved reduction in violation of resource while also reducing the travel time better than other baselines.

3.2 Vehicular Network Routing Problem

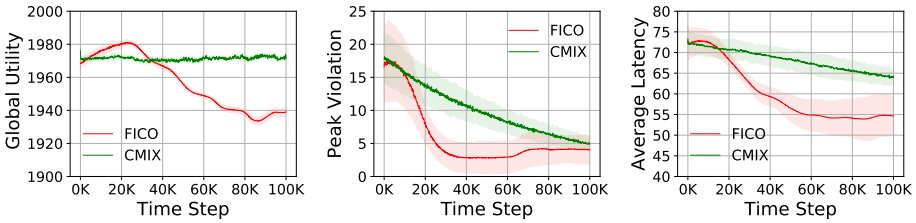
We compare our approach with CMIX [12] in their cooperative vehicular networking problem, which is their largest tested domain. Fig. 6(b) shows a network model with three cells and six clusters. There are two types of cluster - inter cluster (between two cells) and intra cluster. Each cluster contains well connected vehicles that can communicate with high throughput via V2V (Vehicle-to-Vehicle) links. The base station (BS) cell is shared by other mobile user equipments (UEs) and can communicate with vehicles via the direct V2I (Vehicle-to-Infrastructure) links. In this paper, we consider the problem of downlink data transmission where the data are transmitted from BSs to vehicles in the clusters. The objective for all vehicles/agents here is to find the network routes such that the total throughput is maximized (i.e., delivering high volumes of data to destination vehicles), while satisfying both the peak and average latency constraints. Peak constraint means that the latency due to the execution of an agent's action at any time step should be bounded. In CMIX, each agent requires an individual policy to perform action selection. In contrast, agents that belong to the same cluster share the same policy in our collective method. Time limit is set to 180 mins. Further details on hyperparams and neural network structure are in supplementary.

We first follow the same experimental settings as in the CMIX paper to evaluate the performance. There are total three cells and six clusters that are randomly distributed over cells. The number of vehicles in each cluster is randomly generated between a range $[5, 10]$ so that there will be total $30 \sim 60$ vehicles. The throughput and latency of these V2V and V2I links are also randomly generated. Fig. 7 shows the learning curves of global reward, peak violation and latency over time steps. The average latency over time steps in one episode is bounded by



(a) Global reward (b) Peak violation (c) Latency

Fig. 7: Convergence results over time steps with total $30 \sim 60$ agents



(a) Global reward (b) Peak violation (c) Latency

Fig. 8: Convergence results over time steps with total $150 \sim 180$ agents

60. The threshold for peak constraint is also 60. From Fig. 7(c), we can see that the average constraint is satisfied when convergence occurs in both approaches. However, our approach converged much faster and is more sample efficient than the CMIX. In Fig. 7(b), the peak constraint is almost satisfied in our approach. Only one agent's latency is greater than 60 in average. The reason is that we use shared policy for agents in the same cluster. And it is challenging for the policy to consider each agent's peak constraint. In Fig. 7(a), the global rewards in both approaches are increasing (higher is better), and converged to almost the same values (530 in our approach v.s. 535 in CMIX). It shows that our approach is also able to maximize the delivered volumes of data.

We next evaluate the scalability of our approach. We increase agents in each cluster to $[5, 10]$; total number of agents are between 150 and 180. CMIX is only able to train around 100K steps within the limit and our approach can finish 400K steps. Therefore, we show the learning process over 100K steps. Figures 8(b) and (c) show that peak violation and average latency are decreasing in both two approaches. However, the average latency and peak violation in FICO are decreasing with a much faster speed than CMIX. Also, our approach is able to find a policy to satisfy the latency constraint within 100K steps, confirming the effectiveness of our method for large scale problems. The average latency in CMIX is still greater than the threshold 60. The global rewards over time steps are almost unchanged in CMIX, and decreased slightly from 1970 to 1940 in our approach as our approach results in lower constraint violations versus CMIX.

4 Conclusion

We presented a new approach for solving constrained MARL for large agent population. We formulate the constrained MARL problem in a collective multiagent setting then propose to use the fictitious collective Lagrangian relaxation to solve the constrained problem. We developed a credit assignment scheme for both the reward and cost signals under the fictitious play framework. We evaluate our proposed approach on two real-world problems: maritime traffic management and vehicular network routing. Experimental results show that our approach is able to scale up to large agent population and can optimize the cumulative global reward while minimizing the constraint violations.

Acknowledgement

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-016). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and DSO National Laboratories.

References

1. Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: International Conference on Machine Learning. pp. 22–31 (2017)
2. Amato, C., Konidaris, G.D., Kaelbling, L.P., How, J.P.: Modeling and planning with macro-actions in decentralized pomdps. JAIR **64**, 817–859 (2019)
3. Becker, R., Zilberstein, S., Lesser, V.: Decentralized Markov decision processes with event-driven interactions. In: AAMAS. pp. 302–309 (2004)
4. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Mathematics of OR (2002)
5. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific (1999)
6. Chang, Y.H., Ho, T., Kaelbling, L.P.: All learning is local: Multi-agent learning in global reward games. In: NeurIPS. pp. 807–814 (2004)
7. Diddigi, R.B., Danda, S.K.R., Bhatnagar, S., et al.: Actor-critic algorithms for constrained multi-agent reinforcement learning. arXiv preprint:1905.02907 (2019)
8. Foerster, J.N., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: AAAI Conference on Artificial Intelligence (2018)
9. Gattami, A., Bai, Q., Agarwal, V.: Reinforcement learning for multi-objective and constrained markov decision processes. arXiv preprint arXiv:1901.08978 (2019)
10. Hüttenrauch, M., Šošić, A., Neumann, G.: Deep reinforcement learning for swarm systems. JMLR (2018)
11. Kassir, S., de Veciana, G., Wang, N., Wang, X., Palacharla, P.: Enhancing cellular performance via vehicular-based opportunistic relaying and load balancing. In: INFOCOM IEEE Conference on Computer Communications. pp. 91–99 (2019)
12. Liu, C., Geng, N., Aggarwal, V., Lan, T., Yang, Y., Xu, M.: Cmix: Deep multi-agent reinforcement learning with peak and average constraints. In: ECML PKDD (2021)

13. Liu, Y., Ding, J., Liu, X.: Ipo: Interior-point policy optimization under constraints. In: AAAI (2020)
14. Lu, S., Zhang, K., Chen, T., Basar, T., Horesh, L.: Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. In: AAAI (2021)
15. Meyers, C.A., Schulz, A.S.: The complexity of congestion games. Networks (2012)
16. Nair, R., Varakantham, P., Tambe, M., Yokoo, M.: Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In: AAAI Conference on Artificial Intelligence. pp. 133–139 (2005)
17. Nguyen, D.T., Kumar, A., Lau, H.C.: Collective multiagent sequential decision making under uncertainty. In: AAAI (2017)
18. Nguyen, D.T., Kumar, A., Lau, H.C.: Policy gradient with value function approximation for collective multiagent planning. In: NeurIPS. pp. 4322–4332 (2017)
19. Nguyen, D.T., Kumar, A., Lau, H.C.: Credit assignment for collective multiagent RL with global rewards. In: NeurIPS. pp. 8113–8124 (2018)
20. Oliehoek, F.A., Amato, C.: A Concise Introduction to Decentralized POMDPs. Springer Briefs in Intelligent Systems, Springer (2016)
21. Rashid, T., Samvelyan, M., de Witt, C.S., Farquhar, G., Foerster, J.N., Whiteson, S.: Monotonic value function factorisation for deep multi-agent reinforcement learning. JMLR **21**, 178:1–178:51 (2020)
22. Singh, A.J.: Multiagent decision making for maritime traffic management. `https://github.com/rlr-smu/camarl/tree/main/PG_MTM` (2019)
23. Singh, A.J., Kumar, A., Lau, H.C.: Hierarchical multiagent reinforcement learning for maritime traffic management. In: Proceedings of the 19th AAMAS (2020)
24. Singh, A.J., Kumar, A., Lau, H.C.: Learning and exploiting shaped reward models for large scale multiagent RL. In: ICAPS (2021)
25. Singh, A.J., Nguyen, D.T., Kumar, A., Lau, H.C.: Multiagent decision making for maritime traffic management. In: AAAI (2019)
26. Subramanian, J., Mahajan, A.: Reinforcement learning in stationary mean-field games. In: AAMAS. pp. 251–259 (2019)
27. Subramanian, S.G., Poupart, P., Taylor, M.E., Hegde, N.: Multi type mean field reinforcement learning. In: AAMAS (2020)
28. Subramanian, S.G., Taylor, M.E., Crowley, M., Poupart, P.: Partially observable mean field reinforcement learning. In: AAMAS. pp. 537–545 (2021)
29. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: NeurIPS (1999)
30. Tessler, C., Mankowitz, D.J., Mannor, S.: Reward constrained policy optimization. In: International Conference on Learning Representations (2018)
31. Tumer, K., Agogino, A.: Distributed agent-based air traffic flow management. In: AAMAS. pp. 255:1–255:8 (2007)
32. Varakantham, P., Adulyasak, Y., Jaillet, P.: Decentralized stochastic planning with anonymity in interactions. In: AAAI. p. 2505–2511 (2014)
33. Verma, T., Varakantham, P., Lau, H.C.: Entropy based independent learning in anonymous multi-agent settings. In: ICAPS. pp. 655–663 (2019)
34. Wang, J., Ren, Z., Liu, T., Yu, Y., Zhang, C.: QPLEX: duplex dueling multi-agent q-learning. In: ICLR (2021)
35. Wang, W., Wu, G., Wu, W., Jiang, Y., An, B.: Online collective multiagent planning by offline policy reuse with applications to city-scale mobility-on-demand systems. In: AAMAS (2022)
36. Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., Wang, J.: Mean field multi-agent reinforcement learning. In: ICML. vol. 80, pp. 5567–5576 (2018)