Distributed Constrained Multi-Agent Reinforcement Learning with Consensus and Networked Communication

Santiago Amaya-Corredor
Universitat Pompeu Fabra
santiagoesteban.amaya@upf.edu

Anders Jonsson
Universitat Pompeu Fabra
anders.jonsson@upf.edu

Miguel Calvo-Fullana
Universitat Pompeu Fabra
miguel.calvo@upf.edu

Abstract

Our research addresses scalability and coordination challenges inherent to distributed multi-agent systems (MAS) executing under operation constraints. We introduce a novel Constrained Multi-Agent Reinforcement Learning (CMARL) algorithm that integrates a consensus mechanism to ensure agent coordination. Our decentralized approach allows each agent to independently optimize its local rewards while adhering to global constraints evaluated via secondary rewards. These secondary rewards act as a coupling mechanism, penalizing non-cooperative behaviors. Agents operate within a communication network modeled as an undirected graph, exchanging information solely with immediate neighbors to dynamically update dual variables. Our algorithm is validated through its application to the economic dispatch problem within smart grid management, demonstrating its scalability and practical utility in optimizing energy distribution under operational constraints. Experimental results show that our method effectively balances the global and local objectives, proving its robustness in real-world, distributed settings. Key contributions of this work include: (i) the development of a CMARL algorithm that achieves long-term constraint satisfaction and agent consensus, (ii) an enhanced scalability of policy training through problem factorization based on observed state distributions, and (iii) the successful application of our algorithm in a smart grid management use case, highlighting its practical applicability and effectiveness in managing distributed energy resources.

1 Introduction

In recent years, reinforcement learning (RL) has achieved significant success in solving diverse and complex decision-making tasks [Silver et al., 2017, Orr and Dutta, 2023, Brown and Sandholm, 2019]. Many of these successes involve multiple agents and can be characterized as multi-agent RL (MARL). Generally, MARL addresses a sequential problem where a set of autonomous agents make decisions and interact in a shared environment to maximize a reward. However, MARL problems can quickly become intractable as the number of agents increase, since the number of possible interactions and the space of possible states can grow exponentially in the number of agents. Moreover, as all agents navigate and learn simultaneously, the environment may become non-stationary, invalidating many of the single-agent RL assumptions. In realistic scenarios, conflicting objectives often need to be balanced to achieve satisfactory solutions. This issue is exacerbated when increasing the number of

17th European Workshop on Reinforcement Learning (ewrl 2024).

autonomous agents, whose specific goals are not commonly aligned. Finding optimal strategies in multi-agent systems (MAS) usually require at least some level of coordination and communication.

Our work addresses MAS with constrains and continuous operation by adopting a distributed approach. Concretely, we consider a special case of MARL in which each agent operates based on its local states and individual rewards, and selects actions independently. A key contribution of our approach is that we only need to train one policy for all agents as long as they share the same Markov Decision Process (MDP). For example, if all agents observe states from the same distribution, a single policy suffices regardless of the number of agents. However, when agents have different MDPs, separate policies must be trained for each unique MDP. This significantly reduces the complexity of policy training compared to typical MARL approaches. The multi-agent component of the problem consists in achieving a common constraint on a secondary reward function, through the communication of a single variable during execution. This design allows the complexity of execution to increase only linearly with the number of agents and actions, since each agent only needs to compute its own actions and share a single value with its neighbors. Such decentralization is common in real-world scenarios where global state information is unavailable or the size of the problem makes centralized approaches unfeasible. To ensure coordination between these decentralized agents, we introduce a consensus mechanism using a CMARL framework. Here, each agent independently optimizes its part of the problem by maximizing a main reward while adhering to an average bound on a secondary reward. The secondary reward is evaluated globally, acting as a coupling mechanism i.e., a way to allow for the dynamics of the problem to be weakly affected by every agents' actions, and penalizing behaviours that prevent coordination.

We consider that agents are part of a communication network. Each agent only communicates with its immediate neighbors, sharing the necessary information to update their policies and dual variables dynamically. Communication is essential to ensure that agents, while operating independently, coordinate to satisfy the global constraint. We develop a novel CMARL algorithm and apply it to a tailored problem within smart grid management; which refers to the use of advanced information and communication technologies to improve the efficiency, reliability, and sustainability of electricity production, distribution, and consumption [Dileep, 2020]. Our goal is to optimize the energy distribution on an energy district while satisfying operational constraints. Our experiments validate that our methodology is scalable by demonstrating its performance across different network configurations with varying levels of complexity and agent demand heterogeneity. Specifically, we test the algorithm on multiple communication networks with increasing numbers of agents and more intricate connectivity patterns. The results show that our algorithm maintains coordination, even as the number of agents grow and the problem complexity increases. This indicates that our decentralized approach, with its dynamic consensus mechanism, can effectively handle larger and more complex systems and should be easily scalable to even larger systems in real-world, distributed settings. Our methodology is practical and capable of handling real-world tasks through distributed learning and dynamic coordination. We can summarize the key contributions of this work as follows:

1. Introduction of a CMARL distributed algorithm: We develop a novel CMARL algorithm that ensures consensus among agents and achieves constraint satisfaction over extended operational periods.
2. Problem factorization and policy management: Our method significantly enhances the scalability of policy training by factorizing the problem based on observed state distributions across agents. This approach allows for efficient training of a limited number of agent-specific policies, which can be then applied to manage the collective task effectively.
3. Experimental validation in smart grid management: We applied our CMARL algorithm to the economic dispatch problem within smart grid management, demonstrating its practical utility in optimizing energy distribution with adherence to operational constraints.

2 Related Work

Constrained Reinforcement Learning. Our work builds upon the field of CRL, which focuses on solving single-agent sequential decision-making problems subject to constraints on some expected rewards or costs. Traditional CRL approaches include Lagrangian methods [Altman, 2021, Borkar, 2005], where constraints are incorporated into the objective function via Lagrange multipliers. More recently, Calvo-Fullana et al. [2023] proposed a state-augmented CRL framework that augments

the state space with Lagrange multipliers, enabling the agent to learn optimal policies that adapt to changes in the multipliers, overcoming limitations of fixed multiplier approaches. Some works guarantee safe exploration through training with near-constraint satisfaction at each iteration [Achiam and Amodei, 2019, Chow et al., 2019, Achiam et al., 2017], while others, like ours, focus on the satisfaction of the constraints on average in the long-term [Liang et al., 2018, Paternain et al., 2022].

Multi-Agent Reinforcement Learning. Multi-Agent Reinforcement Learning (MARL) problems can be categorized as cooperative, competitive, or mixed. We focus on cooperative MARL, utilizing joint rewards (sum of individual rewards). Centralized control can simplify MARL to single-agent RL, allowing for centralized training with decentralized execution [Kraemer and Banerjee, 2016, Rashid et al., 2018, Sunehag et al., 2017, Lowe et al., 2017]. Our distributed algorithm, however, supports decentralized training and execution, similar to Zhang et al. [2018]. We enhance this with a dynamic consensus mechanism to update dual variables, ensuring coordination and adherence to global constraints over time.

Constrained Multi-Agent Reinforcement Learning. Ensuring safety, risk management, and constraint satisfaction in autonomous multi-agent systems is critical. Centralized approaches with generalized Lagrangian optimization have shown efficiency in online safe RL [Ding et al., 2023]. However, the field of MARL is still in its early stages, and many results that apply to single-agent CRL have not yet been formally extended to the CMARL setting [Chen et al., 2024]. Specifically, for primal-dual methods, while strong duality holds for CRL [Paternain et al., 2019], this is not necessarily the case for all CMARL. Some scenarios, such as constrained Markov potential games, do not satisfy strong duality [Alatur et al., 2023].

Distributed Optimization and Consensus Mechanisms. In large-scale multi-agent systems, centralized approaches quickly become intractable due to the exponential growth of the state and action spaces. To address this, our work draws upon techniques from distributed optimization and consensus mechanisms [Nedić and Ozdaglar, 2009, Olfati-Saber et al., 2007]. These approaches enable decentralized computation and coordination among agents by leveraging local information exchanges and iterative updates to reach consensus on global objectives or constraints. Our proposed methodology integrates elements from these different areas, combining constrained reinforcement learning with multi-agent systems and distributed optimization techniques. Similar to the use of a gossiping algorithm for the multi-agent assignment problem in Agorio et al. [2024], we build upon the state-augmented CRL framework of Calvo-Fullana et al. [2023] and incorporate a consensus mechanism to enable scalable and coordinated learning in distributed multi-agent systems operating under constraints.

3 Preliminaries

Typically, CMARL is studied using the Markov Games framework [Littman, 1994]; an extension of game theory to environments where the dynamics can be modelled using a Markov Decision Process (MDP). Markov games model interactions among multiple agents whose decisions influence a shared environment. In our distributed constrained setting, the Markov game is defined by the tuple $\langle N, \{S^i\}_{i=1}^N, \{A^i\}_{i=1}^N, \{P^i\}_{i=1}^N, \{r_0^i\}_{i=1}^N, \{r_1^i\}_{i=1}^N \rangle$, where N is the number of agents, $S^i \subset \mathbb{R}^m$ and $A^i \subset \mathbb{R}^d$ are compact sets denoting the states and actions of agent i, $P^i : S^i \times A^i \to \Delta(S^i)$ is the state transition probability function for agent i, where $\Delta(S^i)$ is the probability simplex on S^i, $r_0^i : S^i \times A^i \to \mathbb{R}$ is the reward function for the main objective of agent i, and $r_1^i : S^i \times A^i \to \mathbb{R}$ is the reward function for the secondary objective of agent i, which is subject to a constraint. In these games, each agent i observes the current state of its local environment and selects an action $a^i \in A^i$ according to its policy $\pi^i : S^i \to \Delta(A^i)$. At time t, given a joint state $s_t = (s_t^1, \ldots, s_t^N)$ and a joint action $a_t = (a_t^1, \ldots, a_t^N)$, the system transitions to a new state $s_{t+1} = (s_{t+1}^1, \ldots, s_{t+1}^N)$ with probability

$$P(s_{t+1}|s_t, a_t) = \prod_{i=1}^{N} P^i(s_{t+1}^i|s_t^i, a_t^i), \tag{1}$$

and each agent i receives two rewards, $r_0^i(s_t^i, a_t^i)$ and $r_1^i(s_t^i, a_t^i)$. Note that equation (1) means that we assume that the actions of the agents do not affect the states or rewards of others. The Markov property ensures that the dynamics only depend on the last state and action, i.e. $P(s_{t+1}|s_0, a_0, \ldots, s_t, a_t) = P(s_{t+1}|s_t, a_t)$. In our formulation, the rewards are conflicting, with r_0 acting as the main objective

and r_1 as the secondary objective. Specifically, we aim to maximize the sum of all agents' long-term average rewards for $r_0^i(s_t, a_t)$, while ensuring that the sum of all agents' long-term average rewards for $r_1^i(s_t, a_t)$ exceeds a given threshold c. We use this formulation since we are interested in systems that are subject to continuous operation and real world scenarios. This constrained optimization problem can be expressed as

$$\underset{\pi}{\text{maximize}} \quad \sum_{i=1}^{N} \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{s,a\sim\pi^i} \left[\sum_{t=0}^{T} r_0^i(s_t, a_t) \right] \tag{2a}$$

$$\text{subject to} \quad \sum_{i=1}^{N} \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{s,a\sim\pi^i} \left[\sum_{t=0}^{T} r_1^i(s_t, a_t) \right] \leq c. \tag{2b}$$

By defining value functions as the long-term average of each reward,

$$V_j^i(\pi^i) \triangleq \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{s,a\sim\pi^i} \left[\sum_{t=0}^{T} r_j^i(s_t^i, a_t^i) \right], \tag{3}$$

we can rewrite the maximization problem in (2) in the following more concise manner:

$$\underset{\pi}{\text{maximize}} \quad \sum_{i=1}^{N} V_0^i(\pi^i) \quad \text{subject to} \quad \sum_{i=1}^{N} V_1^i(\pi^i) \leq c. \tag{4}$$

In what follows, we formalize two assumptions that are central to our algorithm. These assumptions ensure the tractability and effectiveness of our distributed approach.

Assumption (A1). Each agent i selects an action taking into account only its local state according to its policy $\pi^i : S^i \to \Delta(A^i)$, where $\Delta(A^i)$ is a probability distribution over its actions.

This first assumption allows each agent to operate based solely on local information, which is essential for scalability and decentralization, as it reduces the need for global state information and minimizes communication.

Assumption (A2). The actions of one agent do not affect the states or rewards of others, resulting in a transition probability defined by $P^i(s_{t+1}^i | s_t^i, a_t^i)$. Since we are in a constrained setting, the actions taken by the agents result in a collection of several rewards $r_j^i : S^i \times A^i \to \mathbb{R}$ for $j = 0, \ldots, 1$ that only depend on the state and action of the specific agent.

The second assumption ensures that the interactions between agents are structured in a way that the global objectives can be achieved through local decisions. This facilitates the use of the consensus mechanism to maintain coordination and adherence to global constraints, as the decoupling of agents' actions means that each agent can update its policy and dual variables independently. The consensus mechanism synchronizes these updates across the network, ensuring that the global constraints are respected collectively. Together, these assumptions provide a foundation that supports the feasibility and robustness of our algorithm in managing complex, distributed multi-agent systems.

4 Methodology

We start by formulating the Lagrangian of the optimization problem in (4). This involves introducing Lagrange multipliers to transform the constrained optimization problem into a form where the constraints are incorporated into the objective function as penalty terms.

$$\mathcal{L}(\pi, \lambda) = \sum_{i=1}^{N} V_0^i(\pi^i) + \lambda \left(c - \sum_{i=1}^{N} V_1^i(\pi^i) \right) = \sum_{i=1}^{N} \left(V_0^i(\pi^i) + \lambda \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right), \tag{5}$$

where $\lambda \in \mathbb{R}^+$ is the Lagrange multiplier, interchangeably called the dual variable, associated with the inequality constraint of problem (2), hereinafter refereed to as the global multiplier. Further, we have rewritten the Lagrangian as the sum of the individual agent components to maintain the distributed nature of the formulation. This decomposition allows each agent to optimize its local objective independently while contributing to the global constraint satisfaction, facilitating decentralized

computation and coordination. We can now consider the dual function $d(\lambda) \triangleq \max_\pi \mathcal{L}(\pi, \lambda)$ and state the dual problem as

$$\begin{aligned}\min_\lambda d(\lambda) &= \min_\lambda \left[\max_\pi \sum_{i=1}^N \left(V_0^i(\pi^i) + \lambda \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right) \right] \\ &= \min_\lambda \left[\sum_{i=1}^N \max_{\pi^i} \left(V_0^i(\pi^i) + \lambda \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right) \right], \end{aligned} \tag{6}$$

where the last equality follows from Assumptions (A2) and (A1), meaning that each agent independently maximizing its portion of the Lagrangian with respect to π^i, is equivalent to maximizing it jointly. The next step involves the dual optimization process, where we aim to find the optimal value of the dual variable λ, which is crucial for balancing the primary and secondary objectives across all agents. We seek a value of λ that minimizes the dual function $d(\lambda)$, which is now expressed as the sum of the independently maximized terms from each agent. However, problem (6) still exhibits coupling between agents through λ. To address this, we weaken this coupling by allowing each agent i to have its own Lagrangian multiplier λ^i, while requiring that these individual multipliers are continuously aligned closer and closer to each other through a consensus process. This modification means that the problem we solve is not entirely the same as the one in (6) unless we achieve perfect consensus. However, in practice, it is sufficient to perform a single consensus step at each iteration.

We consider a communication network among the agents, given by an undirected graph $G = (V, E)$, where V is the set of vertices (agents) and $E \subset N \times N$ is the set of edges (communication links between vertices). The neighborhood of a node $i \in V$, denoted by $\mathcal{N}^i$, is the set of nodes that are directly connected to node i by an edge; i.e., $\mathcal{N}^i = \{j \in V \mid (i, j) \in E\}$. In other words, $\mathcal{N}^i$ consists of all nodes j such that there exists an edge (i, j) in the graph G, indicating that node j can directly communicate with node i. Under this model, we can equivalently rewrite the dual problem (6) in distributed dual consensus form

$$\underset{\lambda^1,\ldots,\lambda^N}{\text{minimize}} \sum_{i=1}^N \max_{\pi^i} \left(V_0^i(\pi^i) + \lambda^i \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right) \tag{7a}$$

$$\text{subject to } \lambda^i = \frac{1}{\mathcal{N}^i} \sum_{n \in \mathcal{N}^i} \lambda^n, \quad i = 1, \ldots, N. \tag{7b}$$

We can then write the Lagrangian of this distributed dual problem as

$$\mathcal{L}_D(\lambda^i, \mu^i) = \min_{\lambda^i} \left[\sum_{i=1}^N \max_{\pi^i} \left(V_0^i(\pi^i) + \lambda^i \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right) + \sum_{i=1}^N \mu^i \left(\lambda^i - \frac{1}{\mathcal{N}^i} \sum_{n \in \mathcal{N}^i} \lambda^n \right) \right].$$

Where $\mu \in \mathbb{R}$ is the Lagrange multiplier, associated with the inequality constraint of problem (7), hereinafter refereed to as the consensus multiplier. The resulting dual problem of this dual Lagrangian results in a completely distributed problem across agents $i = 1, \ldots, N$, written as

$$\max_{\mu^i} \left\{ \min_{\lambda^i} \left[\max_{\pi^i} \left(V_0^i(\pi^i) + \lambda^i \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right) + \mu^i \left(\lambda^i - \frac{1}{\mathcal{N}^i} \sum_{n \in \mathcal{N}^i} \lambda^n \right) \right] \right\}. \tag{8}$$

Given the structure of (8), one can address the problem by using three distinct optimization steps to iteratively produce actions and update the dual variables for each agent i. First, each agent takes actions from the policy that maximizes its local objective, given the current values of the dual variables λ^i. Namely,

$$a_t^i \sim \pi^{i,\star}(s_t^i, \lambda_k^i) \quad \text{with} \quad \pi^{i,\star} = \underset{\pi^i}{\text{argmax}} \left(V_0^i(\pi^i) + \lambda_k^i \left(\frac{c}{N} - V_1^i(\pi^i) \right) \right). \tag{9}$$

Second, the central loop updates the dual variables of the global constraint given the current value of the consensus multiplier and the current policy, using a gradient descent approach:

$$\lambda_{k+1}^i \leftarrow \left[\lambda_k^i - \alpha^i \left(\frac{c}{N} - V_1^i(\pi^{i,\star}, \lambda_k^i) + \mu_k^i\right)\right]_+ . \tag{10}$$

Here, α^i is the learning rate, and $[\cdot]_+$ denotes the projection onto the non-negative orthant to ensure that $\lambda \geq 0$. Finally, the outer loop updates the consensus multiplier μ^i given the current values of the policy and the versions of the global multiplier of each agent in the neighborhood of i, using a gradient ascent approach:

$$\mu_{k+1}^i \leftarrow \mu_k^i + \beta^i \left(\lambda_k^i - \frac{1}{\mathcal{N}^i} \sum_{n \in \mathcal{N}^i} \lambda_k^n\right), \tag{11}$$

where β^i is the learning rate. The communication protocol is simple, at every timestep, each agent shares its current value of the global multiplier λ^i with all the agents in its neighborhood. Note that solving (8) is equivalent to solving the original problem if, and only if, all the λ^i values are equal across agents. Achieving this would require running the consensus step until full convergence for each agent. However, to make the problem more tractable, we assume that performing a single consensus step per iteration is sufficient for the algorithm to converge to a solution that is close to the original problem's solution. While this assumption simplifies the algorithm, we acknowledge that it lacks formal theoretical guarantees at this stage. In future work, we plan to rigorously analyze this assumption and provide theoretical guarantees to support it.

The proposed methodology has several important implications. First, *scalability* is achieved by decomposing the global problem into independent local problems, allowing our approach to efficiently scale with the number of agents. Second, *flexibility* is a key advantage, as the dual formulation permits the handling of various types of constraints, making the framework applicable to a wide range of multi-agent reinforcement learning problems. Lastly, the *coordination* is ensured through the consensus mechanism, which is crucial for achieving the global objectives in distributed settings.

5 Algorithm

Each agent optimizes its policy by maximizing the Lagrangian given the current values of the dual variables. One could attempt to perform this step resorting to some form of policy gradient method. However, the primal-dual nature of the algorithm can result in a lack of policy convergence. To address this limitation, we resort to augment the state space of the environment with the Lagrange multipliers of the constraints. This augmentation allows us to learn a policy $\pi^{i,*}(s_t^i, \lambda_t^i)$ that considers the dual variables as part of the state. The policy optimization can be achieved through any RL algorithm.

The dual variables, λ^i, are updated to reflect the current policy's performance concerning the global constraints. This step uses a gradient descent approach to adjust the dual variables based on the difference between the global constraint and the observed performance. The update rule is given by (10). Finally, to ensure consistency across agents, we employ a consensus mechanism where each agent's dual variable is updated based on the average of its neighbors' dual variables. This is achieved using the gradient ascent approach in (11).

Combining these steps, we summarize the execution process in Algorithm 1. The algorithm first learns the optimal policy, and then iteratively updates the dual variables and consensus multipliers to ensure that the global constraints are met while optimizing the local objectives. From Calvo-Fullana et al. [2023, Theorem 1], we know that if we have an optimal policy for a given set of multipliers $\pi^\star(s, \lambda)$, and we *continuously update* these multipliers as in (10) and (11), then the state-action trajectories generated satisfy the constraints in (2).

Algorithm 1 Updating multipliers during execution of trained policy

Input: Trained policies $\pi^{i,\star}(s,\lambda)$, learning rates α^i, β^i, requirement c
Output: Trajectories satisfying the constraints
for $k = 0, 1, \dots$ **do**
  Initialize: Dual variables $\lambda_0^i = 0$, $\mu_0^i = 0$ for $i = 1, \dots, N$
  for each agent i **do**
    Take actions from policy $a_t^i \sim \pi^{i,\star}(s_t^i, \lambda_k^i)$ to obtain $V_1^i(\pi^{i,\star}, \lambda_k^i)$
    Update λ^i using:

$$\lambda_{k+1}^i = \left[\lambda_k^i - \alpha^i \left(\frac{c}{N} - V_1^i(\pi^{i,\star}, \lambda_k^i) + \mu_k^i\right)\right]_+$$

    Update μ^i using:

$$\mu_{k+1}^i = \mu_k^i + \beta^i \left(\lambda_k^i - \frac{1}{|\mathcal{N}^i|} \sum_{n \in \mathcal{N}^i} \lambda^n\right)$$

  end for
end for

6 Use Case: Smart Grid Management

We use an example taken from the smart grid management setting; Demand Response (DR) in a district of independently controlled buildings that have access to solar energy and battery storage units. We want to minimize the prices of energy consumption for each building, while avoiding critical energy peaks that could cause instabilities in the grid. To do so, we will use strategies such as energy storage and load shifting. At each time step, the agent controlling each building's consumption observes a continuous variable for the current demand, the charge of the battery and the price of the energy from the grid. The agent then takes a continuous action, deciding how much of the demand it will supply with energy from the grid and how much with energy from the battery. The demand that is not met is automatically postponed. We have the following local objective for a given agent i:

$$\max_{\pi} \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{s,a} \left[\sum_{t=0}^{T} -e_{\text{grid}}(s_t^i, a_t^i) p_t\right], \tag{12}$$

where $e_{\text{grid}}(s_t^i, a_t^i)$ is the amount of energy consumed from the grid and p_t is the price of consuming an unit of energy at time t. Note that we maximize the negative of the cost. The agents are subject to the following global constraint:

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{s,a} \left[\sum_{t=0}^{T} \sum_{i=1}^{N} e_{\text{grid}}(s_t^i, a_t^i)\right] \leq c, \tag{13}$$

where c is set to a percentage of the maximum peak demand of the whole energy grid, i.e., we require the smart controllers to reduce the maximum peak demand of the grid to a given percentage of the original one, on average over time. This guarantees that the consumption for the whole district is maintained at an acceptable range for grid stability. We let the agents shift the load by deciding whether or not to fulfill the entire demand at a given time step. The demand that is not met, will be postponed for later. The battery is charged automatically when solar energy is available for generation. We add a final local constraint that ensures the demand is met:

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{s,a} \left[\sum_{t=0}^{T} \left(d_t^i - e_{\text{grid}}(s_t^i, a_t^i) - e_{\text{battery}}(s_t^i, a_t^i)\right)\right] = 0, \tag{14}$$

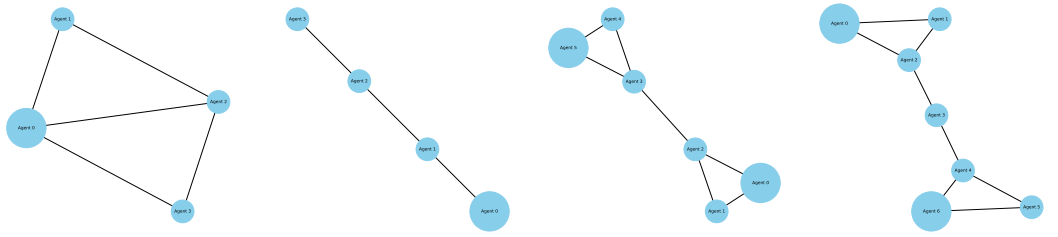
where d_t^i is the demand af the i-th agent at time t and $e_{\text{battery}}(s_t^i, a_t^i)$ is the energy consumed from the battery at time step t. The training of the policy is performed following the state augmented procedure described in Section 5 and the updates of the global constraint and the consensus multipliers are performed as shown in Algorithm 1. For the handling of the local constraint we just add another term to the Lagrangian which only needs the addition of the following update

$$\nu_{k+1}^i = \nu_k^i - \eta^i \left(d_k^i - e_{\text{grid}}(s_k^i, a_k^i) - e_{\text{battery}}(s_k^i, a_k^i)\right). \tag{15}$$

We take the data on energy prices ($/kWh), energy demand (kWh) and solar generation (W/kW) from an open source Farama Foundation Gymnasium environment called City Learn which is used for the implementation and benchmarking of MARL for demand response in cities [Vázquez-Canteli et al., 2019, Vazquez-Canteli et al., 2020].

7 Experimental Results

We run the experiment for the different network configurations shown in Figure 1 which present increasing levels of complexity with respect to the number of direct connections (edges) a node has to other nodes, and also with respect to how many different demands distributions are present in the network. The less connected the nodes are in the network, the harder it is for the agents to reach consensus. Moreover, by allowing agents to have different demands, we can engineer scenarios where the problem is unfeasible for individual agents without coordination but feasible for a coordinated group. We present the results obtained for the configuration in Figure 1d which we consider to be complex enough to validate the strengths of our algorithm. Specifically, this configuration has two weakly connected groups of agents, each with one agent that has double the demands of the others. We then train two policies; one for agents with normal demand, and one for agents with double demand. The policies were trained using the Proximal Policy Optimization (PPO) algorithm. To be precise, we utilize single-agent training with multi-agent execution. By leveraging assumptions A1 and A2, along with the introduction of individual Lagrange multipliers (λ^i) for each agent, we enable a single-agent policy to be trained while ensuring coordination during execution. Consensus over λ^i is crucial, as it serves as the key mechanism that links single-agent training to multi-agent execution, ensuring global constraint satisfaction. Each training episode consisted of 80 timesteps, which corresponded to approximately 3 days of energy demand in our dataset. At the beginning of each episode, we randomly sampled the global multiplier λ and the local multiplier μ from a uniform distribution, and trained for around 10,000 episodes. During training, we allowed for values of $\lambda \in \{0, 15\}$ and $\mu \in \{-20, 20\}$. We selected the range of values for the multipliers by gradually extending their limits to include values that allowed for a solution under specific constraint requirements, while trying to increase the state space size as little as possible. The dataset contained around 3,000 hours of information, and for each episode, we randomly selected the starting point within this data.



(a) Circular connections, one agent with double demand

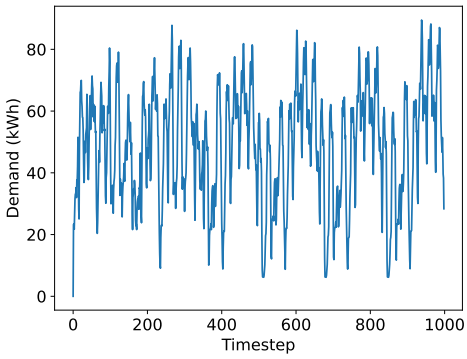
(b) Linear connections, one agent with double demand

(c) Two clusters, two agents with double demand

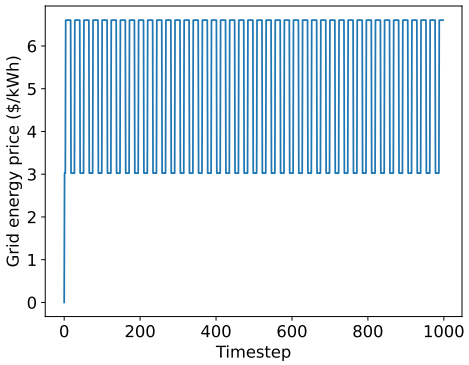
(d) Two clusters, two agents with double demand

Figure 1: Different Communication Networks and agent demands

The grid demand and grid unit energy price for the selected configuration can be observed in Figures 2a and 2b respectively. We set the requirement c to 27% of the total peak demand as we find it to be a good value to obtain a difficult but feasible problem. We run the trained policies independently for each agent during 3,000 time-steps, while continuously updating their respective multipliers as shown in Algorithm 1 to obtain trajectories that optimize the objective and fulfill the constraint on average over time. To validate the importance of coordination, we compare two versions of the algorithm; one that reaches consensus and one that does not. In Figure 3, we see that the algorithm that reaches consensus and the one that does not, have almost identical grid consumption, with an average that is below the required percentage of the total peak demand. However, the algorithm without consensus achieves this at the expense of indefinitely postponing the demand.



(a) Energy demand of the grid composed by the 7 buildings of the selected configuration



(b) Price per unit of energy bought from the grid

Figure 2: States of the smart grid use case

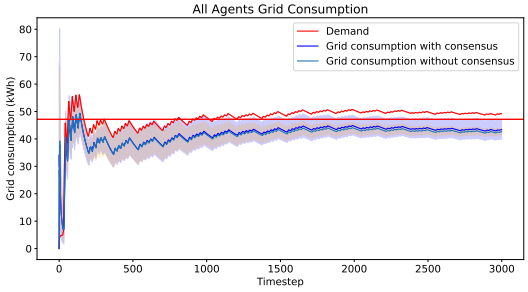
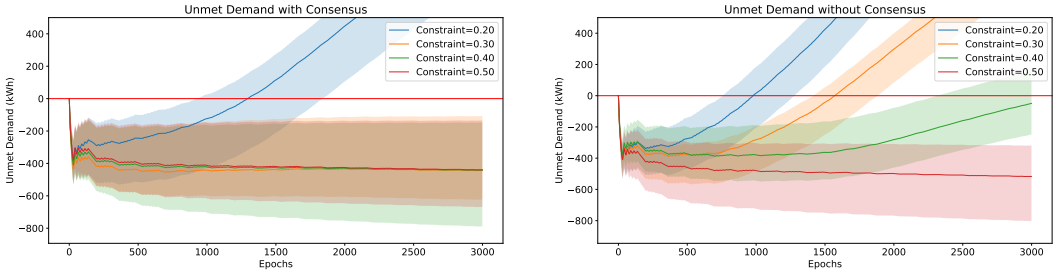


Figure 3: Comparison of the total grid consumption (on average) between algorithm where the global constraints reach consensus and algorithm where it does not.

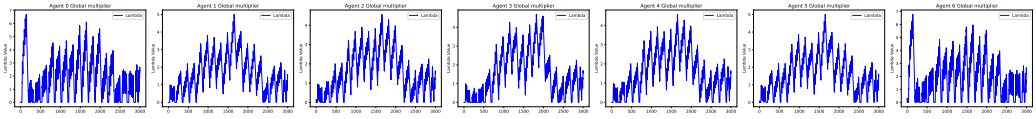
To see the influence of consensus on demand satisfaction we consider four constraint levels $c \in \{0.2, 0.3, 0.4, 0.5\}$, and plot the cumulative average unmet demand over time for each of them. In Figure 4a we can see the unmet demand for the consensus algorithm. When the problem is feasible i.e., $c \in \{0.3, 0.4, 0.5\}$, the unmet demand stabilizes at a point, which is more negative the less stringent the constraint. This makes sense as the agents are less penalized from grid energy usage. We see that for $c = 0.2$ the unmet demand keeps increasing, which indicates that a solution is not reached, likely because at this level the problem becomes unfeasible even with coordination. For the algorithm that does not reach a consensus, the problem becomes unfeasible for less strict values of the constraint, i.e. $c \in \{0.3, 0.4\}$ (Figure 4b). We see then, that for the original constraint of 27% even though both algorithms consume the same amount of energy from the grid, the non-consensus version is actually not solving the problem, since the Lagrange multipliers never converge (Figure 5), and thus the optimal solution is not found. In Figure 5a, we observe that our methodology successfully achieves consensus for the global multipliers. This indicates that the agents are able to coordinate their actions effectively, ensuring that the global constraints are met collectively. Conversely, when the consensus mechanism is not used, the multipliers for the agents with double demand fail to converge and instead reach the maximum allowed value of 15 (Figure 5b). This maximum value is imposed for practical reasons, as allowing the multipliers to grow indefinitely is not feasible in real-world applications. The fact that the multipliers hit this upper limit indicates that they have not converged within the permissible range, suggesting that the optimal values have not been found. Consequently, this implies that the problem is not being solved adequately in this scenario. The failure to achieve convergence within the allowed range of the multipliers demonstrates the importance of the consensus mechanism in ensuring the successful coordination and optimization of the agents' actions in meeting the global constraints.



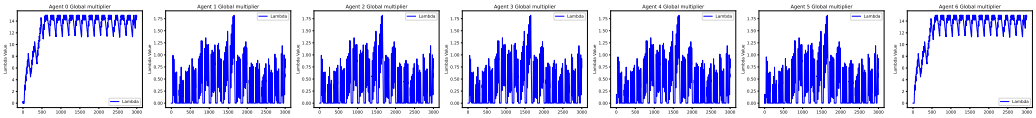
(a) Cumulative unmet demand on average over time for the algorithm that reaches consensus for different constraint levels

(b) Cumulative unmet demand on average over time for the algorithm that does not reach consensus for different constraint levels

Figure 4: Comparison between algorithms for local constraint satisfaction, i.e., unmet demand



(a) Global multipliers for algorithm that reaches consensus



(b) Global multipliers for algorithm that does not reach consensus.

Figure 5: Comparison of global constraint multipliers for the algorithm that achieves consensus (a) and for the algorithm that does not (b).

8 Conclusion

In this paper we have presented a novel CMARL algorithm that addresses some of the scalability and coordination challenges inherent to distributed multi-agent systems (MAS) operating under constraints. Our proposed method integrates a consensus mechanism within a decentralized framework, enabling each agent to independently optimize its local rewards while adhering to global constraints. The application of our algorithm to the problem in smart grid management has demonstrated its practical utility and effectiveness in optimizing energy distribution under operational constraints. Future work includes the study of theoretical analysis of the proposed algorithm to establish formal guarantees on performance, convergence and robustness.

9 Acknowledgements

Santiago Amaya-Corredor is part of the Maria de Maeztu Units of Excellence Programme CEX2021-001195-M, funded by MICIU/AEI /10.13039/501100011033.

Anders Jonsson is partially supported by AGAUR SGR and Spanish grant PID2019-108141GB-I00.

Miguel Calvo-Fullana is partially supported the Spanish Agencia Estatal de Investigación under grant RYC2021-033549-I.

References

Joshua Achiam and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. 2019. URL https://api.semanticscholar.org/CorpusID:208283920.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

Leopoldo Agorio, Sean Van Alen, Miguel Calvo-Fullana, Santiago Paternain, and Juan Andres Bazerque. Multi-agent assignment via state augmented reinforcement learning. *arXiv preprint arXiv:2406.01782*, 2024.

Pragnya Alatur, Giorgia Ramponi, Niao He, and Andreas Krause. Provably learning nash policies in constrained markov potential games. In *Sixteenth European Workshop on Reinforcement Learning*, 2023. URL https://openreview.net/forum?id=1EusBrDDrOK.

Eitan Altman. *Constrained Markov decision processes*. Routledge, Boca Raton, December 2021.

V.s Borkar. An actor-critic algorithm for constrained markov decision processes. *Systems & Control Letters*, 54:207–213, 03 2005. doi: 10.1016/j.sysconle.2004.08.007.

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456): 885–890, 2019. doi: 10.1126/science.aay2400. URL https://www.science.org/doi/abs/10.1126/science.aay2400.

Miguel Calvo-Fullana, Santiago Paternain, Luiz FO Chamon, and Alejandro Ribeiro. State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards. *IEEE Transactions on Automatic Control*, 2023.

Ziyi Chen, Yi Zhou, and Heng Huang. On the hardness of constrained cooperative multi-agent reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wFWuX1Fhtj.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Mohammad Ghavamzadeh, and Edgar A. Duéñez-Guzmán. Lyapunov-based safe policy optimization for continuous control. *ArXiv*, abs/1901.10031, 2019. URL https://api.semanticscholar.org/CorpusID:59336201.

G. Dileep. A survey on smart grid technologies and applications. *Renewable Energy*, 146:2589–2625, 2020. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2019.08.092. URL https://www.sciencedirect.com/science/article/pii/S0960148119312790.

Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo R. Jovanović. Provably Efficient Generalized Lagrangian Policy Optimization for Safe Multi-Agent Reinforcement Learning. *arXiv e-prints*, art. arXiv:2306.00212, May 2023. doi: 10.48550/arXiv.2306.00212.

Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.01.031. URL https://www.sciencedirect.com/science/article/pii/S0925231216000783.

Qingkai Liang, Fanyu Que, and Eytan H. Modiano. Accelerated primal-dual policy optimization for safe reinforcement learning. *ArXiv*, abs/1802.06480, 2018. URL https://api.semanticscholar.org/CorpusID:3349200.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 157–163. Morgan Kaufmann, San Francisco (CA), 1994. ISBN 978-1-55860-335-6. doi: https://doi.org/10.1016/B978-1-55860-335-6.50027-1. URL https://www.sciencedirect.com/science/article/pii/B9781558603356500271.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv e-prints*, art. arXiv:1706.02275, June 2017. doi: 10.48550/arXiv.1706.02275.

Angelina Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009. ISSN 0018-9286. doi: 10.1109/TAC.2008.2009515. Funding Information: Manuscript received May 25, 2007; revised October 29, 2007. Current version published January 14, 2009. This work was supported in part by the National Science Foundation CAREER Grants CMMI 07-42538 and DMI-0545910, and by the DARPA ITMANET Program. Recommended by Associate Editor D. Henrion.

Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007. doi: 10.1109/JPROC.2006.887293.

James Orr and Ayan Dutta. Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors*, 23(7), 2023. ISSN 1424-8220. doi: 10.3390/s23073625. URL `https://www.mdpi.com/1424-8220/23/7/3625`.

Santiago Paternain, Luiz F. O. Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained Reinforcement Learning Has Zero Duality Gap. *arXiv e-prints*, art. arXiv:1910.13393, October 2019. doi: 10.48550/arXiv.1910.13393.

Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Safe policies for reinforcement learning via primal-dual methods. *IEEE Transactions on Automatic Control*, 68 (3):1321–1336, 2022.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv e-prints*, art. arXiv:1803.11485, March 2018. doi: 10.48550/arXiv.1803.11485.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, Oct 2017. ISSN 1476-4687. doi: 10.1038/nature24270. URL `https://doi.org/10.1038/nature24270`.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-Decomposition Networks For Cooperative Multi-Agent Learning. *arXiv e-prints*, art. arXiv:1706.05296, June 2017. doi: 10.48550/arXiv.1706.05296.

José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. Citylearn v1.0: An openai gym environment for demand response with deep reinforcement learning. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '19, page 356–357, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450370059. doi: 10.1145/3360322.3360998. URL `https://doi.org/10.1145/3360322.3360998`.

Jose R Vazquez-Canteli, Sourav Dey, Gregor Henze, and Zoltan Nagy. CityLearn: Standardizing Research in Multi-Agent Reinforcement Learning for Demand Response and Urban Energy Management. *arXiv e-prints*, art. arXiv:2012.10504, December 2020. doi: 10.48550/arXiv.2012.10504.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents. *arXiv e-prints*, art. arXiv:1802.08757, February 2018. doi: 10.48550/arXiv.1802.08757.