Decentralized Mean Field Games

To Appear in AAAI-22

Authors: Sriram Ganapathi Subramanian, Matthew E. Taylor, Mark Crowley, Pascal Poupart

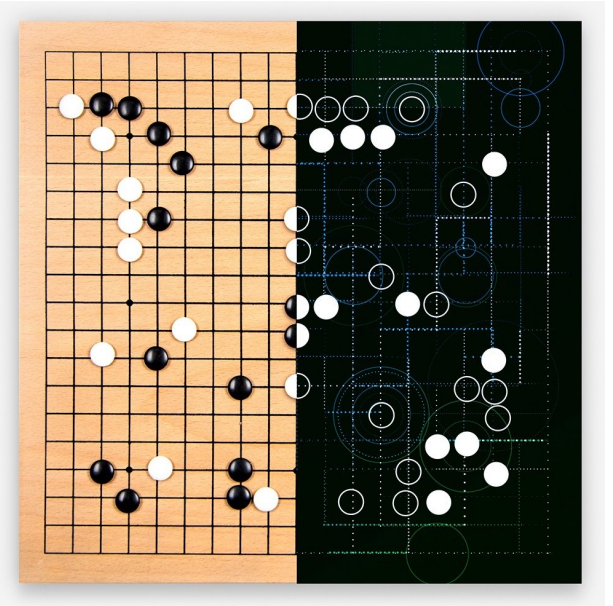
Introduction

- **Problem**: The previous works on multi-agent reinforcement learning either does not **scale** well with big number of agents, or only learns in a **centralized** fashion with the assumption that all the agents are identical and have the same policy, which are impractical

- This paper introduces a new method to tackle the multi-agent reinforcement learning problem in a **decentralized** way, i.e. each agent can learn to cooperate or compete with other agents **individually**, without the global information

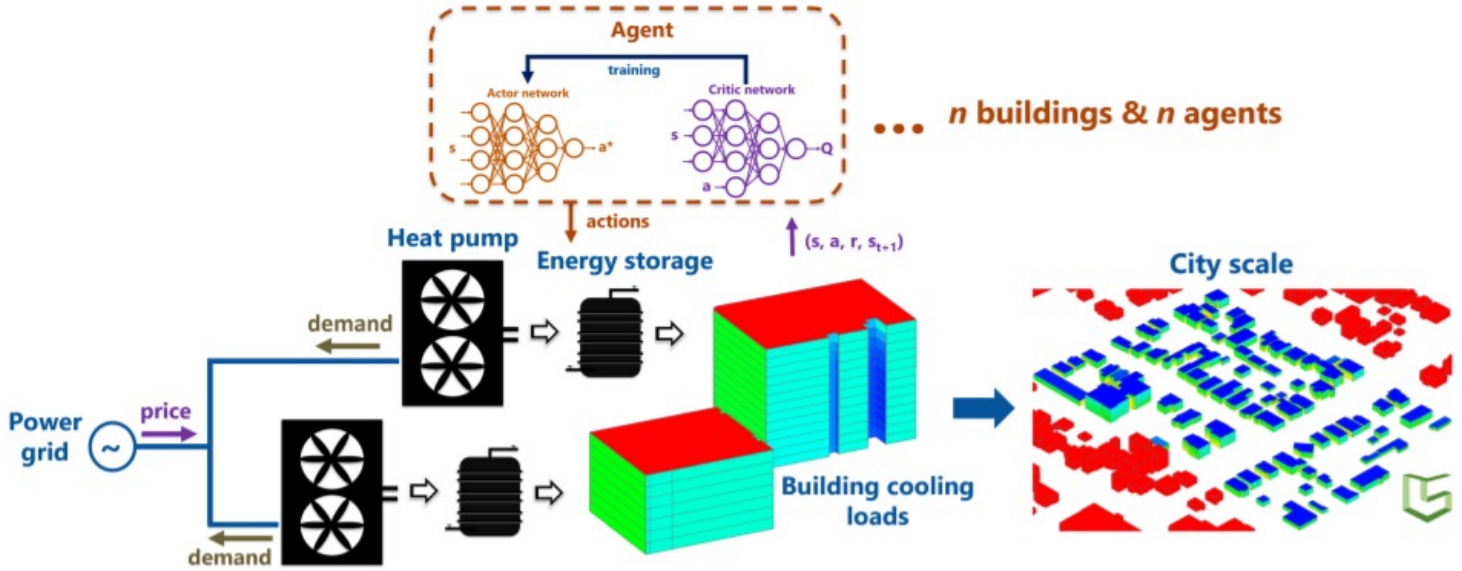
Background: Multi-Agent Reinforcement Learning

- **Background**: How can **multiple agents** interact with the environment and one another to complete tasks. More specifically, how can they **cooperate** with each other to complete a common task, or **compete** with each other to complete their own task, or a mix of the two.



Background: Multi-Agent Reinforcement Learning

- **Problem**: The previous multi-agent RL algorithms are hard to scale up. They require the knowledge of all agents' states and actions, if the number of agents is too big, the problem becomes **intractable**.



Background: Mean Field Game

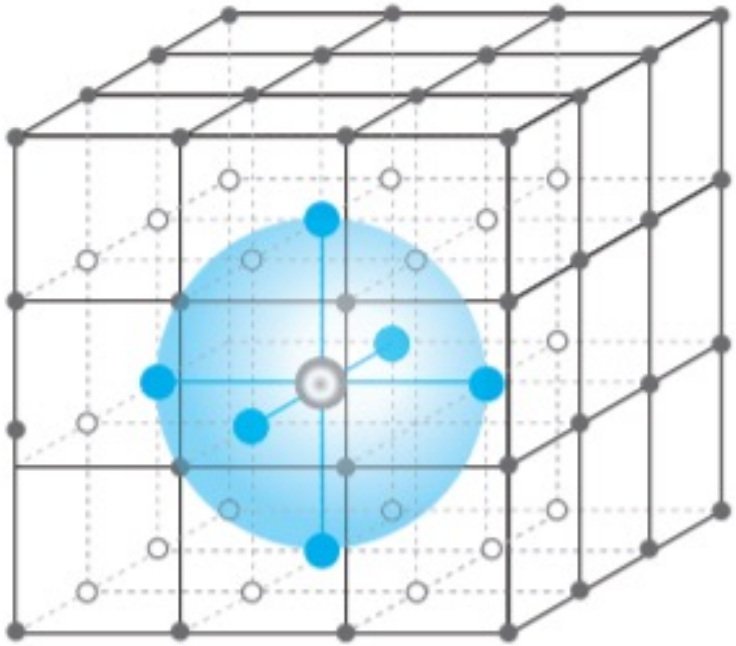
- **Solution**: **Mean-Field Game (MFG)** – Approximate the effects of all other agents in the environment into one agent, which captures the essential aspects of the system, and thus makes the multi-agent problem a two-agent problem.

- **Assumptions**
  1) Each agent has access to accurate **global** information, but not the local information
  2) All agents in the environment are **identical** and **indistinguishable**, i.e. all agents share the same state/action spaces and reward function, and have the same objectives (motivates **centralized** training methods)
  3) All agents maintain interactions with others only through the mean field

Previous Work: Mean Field Reinforcement Learning

- **Mean Field Reinforcement Learning** (Yang et al, 2018) extends Q-Learning into mean field game by using the empirical mean action to update the Q-function

- **Additional Assumptions**:
  1) Assumes each agent's sphere of influence is restricted by certain neighborhoods
  2) Use the previous mean field to update the current action to avoid the "chicken and egg" problem (next slide)



Previous Work: Mean Field Reinforcement Learning

- **Mean Field Reinforcement Learning** (Yang et al, 2018) Algorithm

$$Q^j(s_t, a_t^j, \mu_t^a) = (1-\alpha)Q^j(s_t, a_t^j, \mu_t^a) + \alpha[r_t^j + \gamma v^j(s_{t+1})]$$

where

$$v^j(s_{t+1}) = \sum_{a_{t+1}^j} \pi^j(a_{t+1}^j | s_{t+1}, \mu_t^a) Q^j(s_{t+1}, a_{t+1}^j, \mu_t^a)$$

$$\mu_t^a = \frac{1}{\mathcal{N}} \sum_j a_t^j, a_t^j \sim \pi^j(\cdot | s_t, \mu_{t-1}^a)$$

$$\pi^j(a_t^j | s_t, \mu_{t-1}^a) = \frac{\exp(-\hat{\beta} Q^j(s_t, a_t^j, \mu_{t-1}^a)}{\sum_{a_t^{j'} \in A^j} \exp(-\hat{\beta} Q^j(s_t, a_t^{j'}, \mu_{t-1}^a))}$$

The value function ***v*** is the product of the probability of choosing an action and the Q-value

μ is an average of one-hot encoded policies of the neighboring agents

Boltzmann Policy

This Paper: Decentralized Mean Field Game (DMFG)

- **Assumptions of DMFG**
  1) Does **not** assume the agents are **indistinguishable** and **homogeneous**
  2) Does **not** assume each agent can access the global mean field of the system, instead each agent has full accurate information in its **neighborhood** only
  3) Assumes all agents formulate **responses only to the mean field** of the system (since each agent's impact on the environment is infinitesimal)
  4) Assumes each agent's sphere of influence is restricted by its **neighborhood** (in line with Yang et al. 2018)

- **Goal** (Assume the mean field of the system $\boldsymbol{\mu}$ can be represented by $\boldsymbol{\mu^j}$ after finite time, this paper uses the mean field of state distribution)

$$J^j_{\boldsymbol{\mu}}(\pi^j) \triangleq \mathbb{E}^{\pi^j}\left[\sum_{t=0}^{\infty} \beta^t r^j(s^j_t, a^j_t, \mu_t)\right].$$

Decentralized Mean Field Equilibrium

- The decentralized mean field equilibrium of an agent *j* is represented as a pair (π_*^j, μ_*^j), π_*^j is the best response to μ_*^j and μ_*^j is the best mean field estimate of agent *j* when it plays π_*^j

- How can we make sure such an equilibrium can be achieved under the assumptions of decentralized mean field games?

Theoretical Results

- **Theorem 1**. For any mean field, $\mu \in M$, and an agent $j \in \{1, \ldots, N\}$, we have

$$\sup_{\pi^j \in \Pi^j} J^j_{\boldsymbol{\mu}}(\pi^j) = \sup_{\pi^j \in \Pi^j_M} J^j_{\boldsymbol{\mu}}(\pi^j)$$

- Π^j_M denotes the Markov policies for agent *j*. Restricting the policies to be Markovian will not lose any optimality

Theoretical Results

- **Theorem 2**. An agent j ∈ {1, …, N} in the DMFG admits (has) a decentralized mean field equilibrium $(\pi_*^j, \mu_*^j) \in \Pi^j$ x M

Theoretical Results

- **Theorem 3**. The decentralized mean field operator *H* is well-defined, i.e., this operator maps *C* x *P(S)* to itself.

$$H : \mathcal{C} \times \mathcal{P}(\mathcal{S}) \ni (Q^j, \boldsymbol{\mu}^j)$$

$$\rightarrow (H_1(Q^j, \boldsymbol{\mu}^j), H_2(Q^j, \boldsymbol{\mu}^j)) \in \mathcal{C} \times \mathcal{P}(\mathcal{S})$$

where

$$H_1(Q^j, \boldsymbol{\mu}^j)(s_t^j, a_t^j) \triangleq r^j(s_t^j, a_t^j, \mu_t)$$

$$+\beta \int_{\mathcal{S}} Q^j_{\max_{a^j}}(s_{t+1}^j, a^j, \mu_{t+1}^j) p(s_{t+1}^j | s_t^j, a_t^j, \mu_t)$$

$$H_2(Q^j, \boldsymbol{\mu}^j)(\cdot)$$

$$\triangleq \int_{\mathcal{S}\times\mathcal{A}^j} p\Big(\cdot\,|s_t^j, \pi^j(s_t^j, Q_t^j, \mu_t^j), \mu_t\Big)\mu_t^j(s)$$

Theoretical Results

- **Theorem 4**. Let *B* represent the space of bounded functions in *S*. Then the mapping $H : C \times P(S) \rightarrow C \times P(S)$ is a contraction in the norm of *B(S)*

- Since H is a contraction, an update algorithm (Q-learning) can converge to a fixed point representing DMFE

Theoretical Results

- **Theorem 5**. Let the Q-updates in Algorithm 1 converge to (Q_*^j, μ_*^j) for an agent $j \in \{1, \ldots, N\}$. Then, we can construct a policy π_*^j from Q_*^j using the relation

$$\pi_*^j(s^j) = \arg \max_{a^j \in \mathcal{A}^j} Q_*^j(s^j, a^j, \mu_*^j)$$

Then the pair (π_*^j, μ_*^j) is a DMFE.

Algorithm 1: Q-learning for DMFG

1: For each agent $j \in \{1, \ldots, N\}$, start with initial Q-function Q_0^j and the initial mean field state estimate μ_0^j
2: **while** $(Q_n^j, \mu_n^j) \neq (Q_{n-1}^j, \mu_{n-1}^j)$ **do**
3: $(Q_{n+1}^j, \mu_{n+1}^j) = H(Q_n^j, \mu_n^j)$
4: **end while**
5: Return the fixed point (Q_*^j, μ_*^j) of H

Detailed Algorithm

$$Q^j(s_t^j, a_t^j, \mu_t^{j,a}) = (1-\alpha)Q^j(s_t^j, a_t^j, \mu_t^{j,a}) + \alpha[r_t^j + \gamma v^j(s_{t+1}^j)] \quad (12)$$

where

$$v^j(s_{t+1}^j) = \sum_{a_{t+1}^j} \pi^j(a_{t+1}^j | s_{t+1}^j, \mu_{t+1}^{j,a}) Q^j(s_{t+1}^j, a_{t+1}^j, \mu_{t+1}^{j,a}) \quad (13)$$

$$\mu_t^{j,a} = f^j(s_t^j, \hat{\mu}_{t-1}^{j,a}) \quad (14)$$

and $\pi^j(a_t^j | s_t, \mu_t^{j,a}) = \frac{\exp(-\hat{\beta} Q^j(s_t, a_t^j, \mu_t^{j,a}))}{\sum_{a_t^{j'} \in A^j} \exp(-\hat{\beta} Q^j(s_t^j, a_t^{j'}, \mu_t^{j,a}))}$

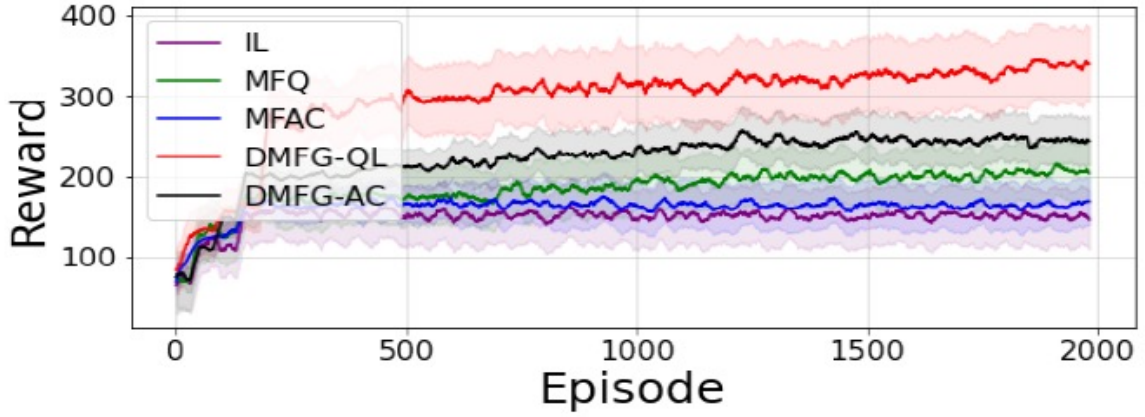
f is a mean field network trained on observed mean action $\hat{u}_t^{j,a}$, eliminate the "chicken-and-egg" problem

Experiment and Results

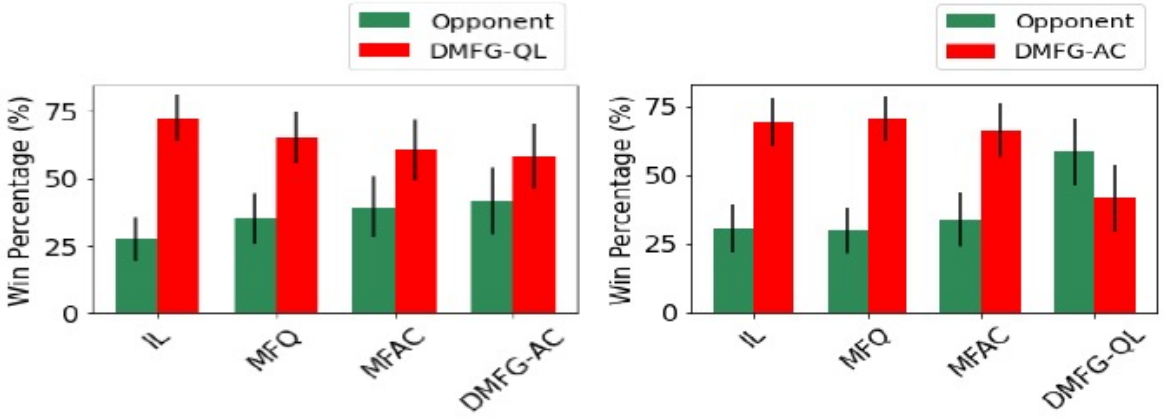
- **Conduct the experiments in 2 phases**: Training and Execution, repeat each experiment 30 times and report mean and std
- **Training**: all agents train against other agents playing the same algorithm for 2000 games
- **Execution**: The trained agents then execute the learned policies for 100 games, where algorithms may compete against each other.
- **3 Baselines**: Independent Q-learning (**IL**), mean field Q-learning (**MFQ**), mean field actor-critic (**MFAC**), implemented in a **decentralized** fashion with only local information

Experiment and Results

- First experiment: **Mixed cooperative-competitive** Battle game, where 2 teams of 25 agents compete with each other (the environment is not zero-sum), **DMFG-QL performs best** while IL performs worst.



(a) Training

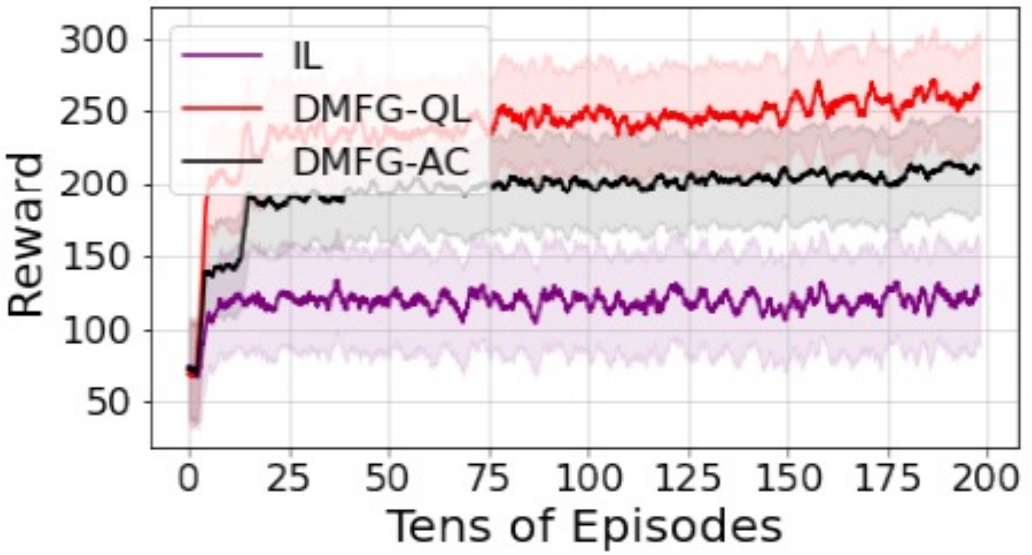


(b) Execution vs. DMFG-QL

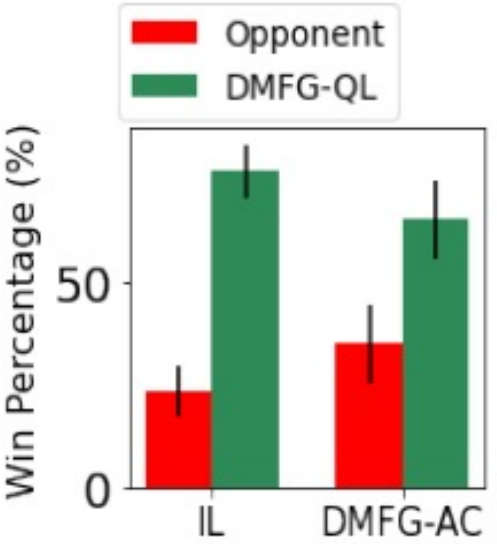
(c) Execution vs. DMFG-AC

Experiment and Results

- Second experiment: **Mixed cooperative-competitive** similar to Battle, except each team has 15 ranged and 10 melee agents. MFQ and MFAC are removed since both require homogeneous agents. **DMFG-QL performs best.**



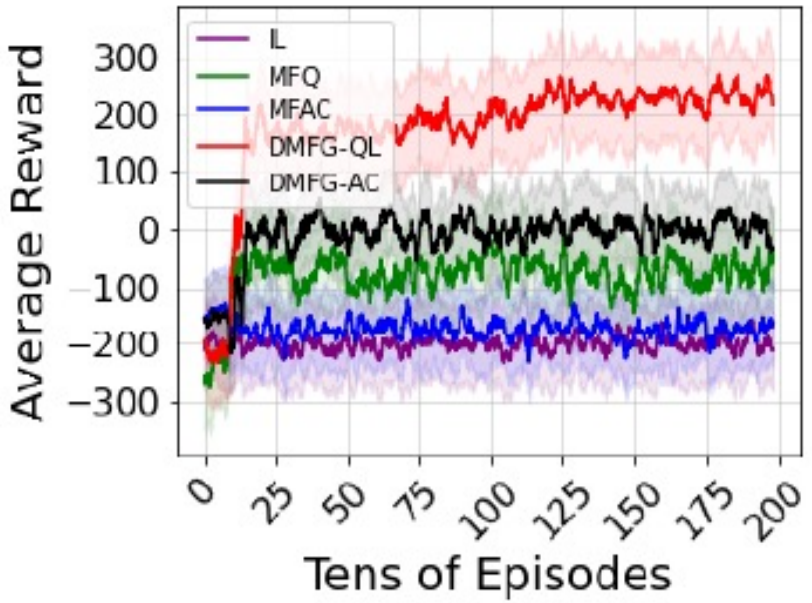
(a) Training



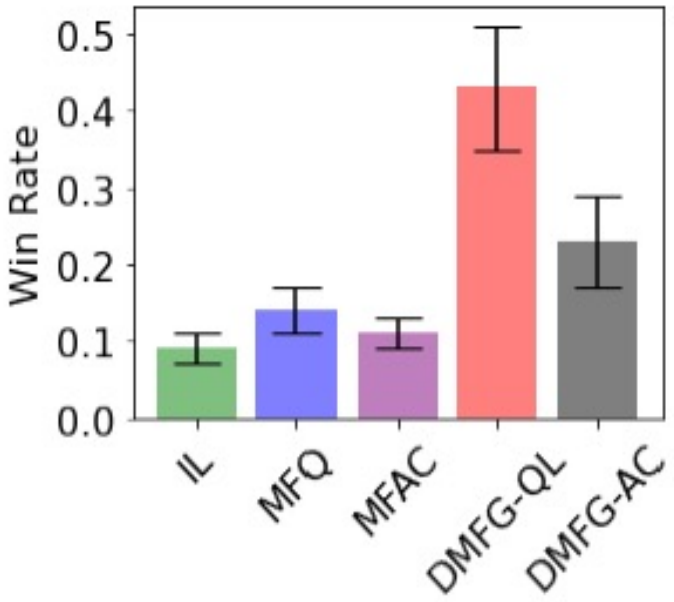
(b) Execution

Experiment and Results

- Third experiment: **Fully competitive** Gather environment. 30 agents compete against each other to capture limited food and could resort to killing others. **DMFG-QL performs best** by far. Actively formulating the best individual strategy is crucial in competitive environments.



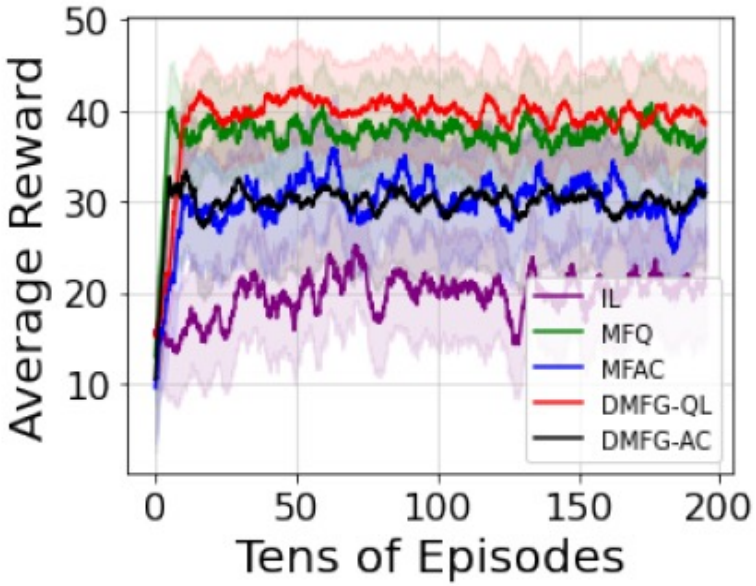
(a) Training



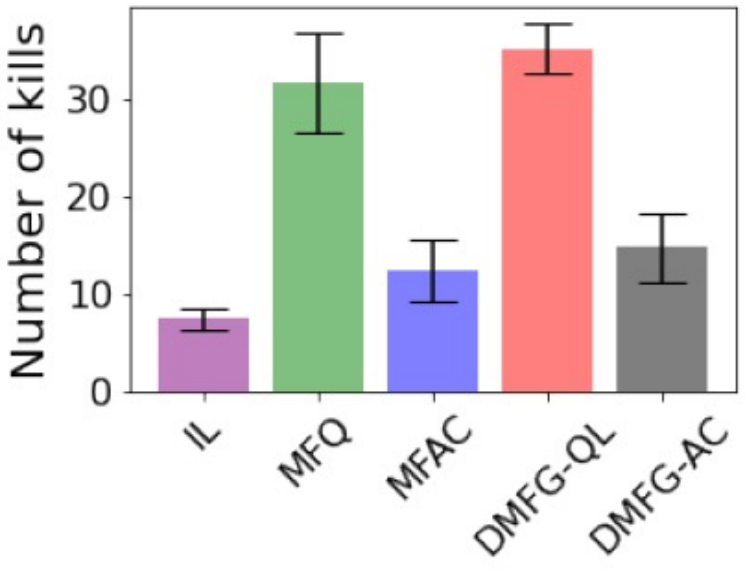
(b) Execution

Experiment and Results

- Fourth experiment: **Fully cooperative** Tiger-Deer environment. Deers are part of the environment and at least 2 tigers need to attack a deer together to gain large rewards. DMFG and MFG algorithms perform similarly.



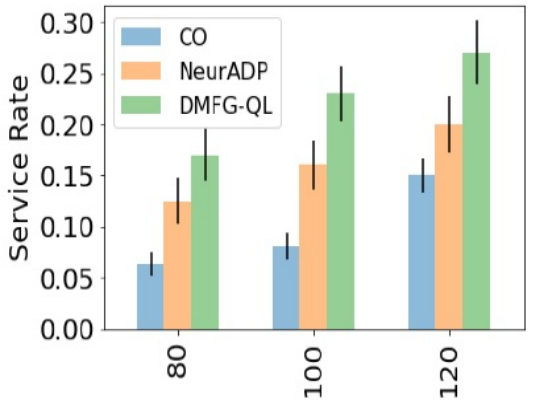
(a) Training



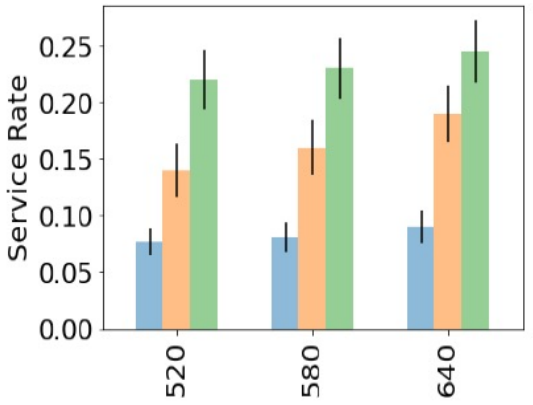
(b) Execution

Experiment and Results

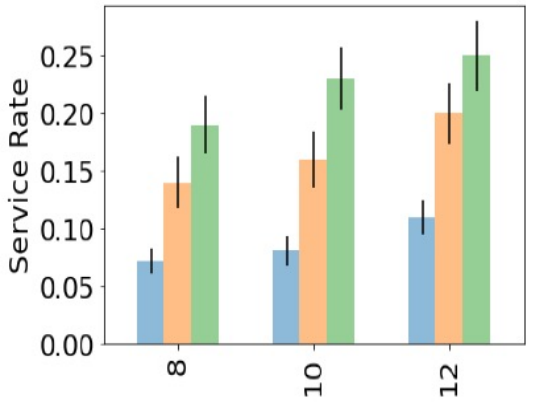
- Fifth experiment: **real-world** Ride-pool Matching Problem – tries to improve the efficiency of vehicles satisfying ride request. Baselines are 2 previous methods using constrained optimization and centralized DQN. Service rate=percentage of requests served.



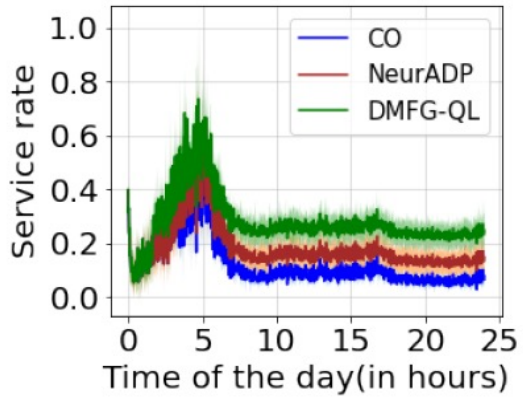
(a) # of Vehicles



(b) Maximum Pickup Delay



(c) Capacity



(d) Single Day Test

Potential Drawbacks

- The neighborhoods of the mean field estimate are artificially defined, which may not be optimal in practice
- Sometimes the training becomes difficult if the number of agents is too large (since each agent learns its own policy), however in practice the training can be decentralized (like experiment five the ride match problem)

Conclusion

- This paper relaxed two strong assumptions from the previous work on using mean field method in RL, and under the relaxed assumptions it introduces the **Decentralized Mean Field Game** (DMFG) framework, where agents do not have global information and are not homogeneous, and learn in a decentralized fashion
- Proved the Q-learning based algorithm will find the DMFE
- Addressed the “chicken-and-egg” problem with a mean field network
- Demonstrated the superior performances in various scenarios

Future Work

- Theoretically, extend the theoretical analysis to the function approximation setting and analyze the convergence of policy gradient algorithm
- Empirically, consider other real-world applications like autonomous driving and demand and supply optimization