Learning Cooperation for Partially Observable Multi-Agent Path Finding

M.Sc. Thesis Defence

Qiushi Lin

Simon Fraser University

Motivations

Automated Warehouses[1]

Sorting Centers[2]

Video Games[3]

Drone Swarm Systems[4]

---

[1] https://www.amazon.science/latest-news/how-amazon-robots-navigate-congestion

[2] https://www.wired.com/story/amazon-warehouse-robots

[3] https://upload.wikimedia.org/wikipedia/commons/0/02/3_cossacks_european_wars.jpg

[4] https://futureoflife.org/wp-content/uploads/2019/04/Why-ban-lethal-AI-1030x595.jpg

Challenges

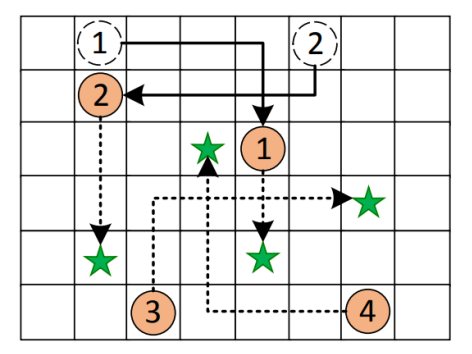
- Planning paths for multiple agents are usually modelled as **NP-hard** combinatorial search problems in discretized worlds with discretized time steps
- Search-based planning algorithms are **computationally expensive** and **cannot generalize well among instances**
- Learning-based methods provide solvers that are computationally efficient but usually have **poor solution quality**
- Most multi-agent reinforcement learning (MARL) methods can only deal with instances with small numbers of agents (usually 5-10 agents) and are usually **not scalable** when it comes to congested environments
- It is not clear in the literature how various types of cooperation and objectives among agents can be learned

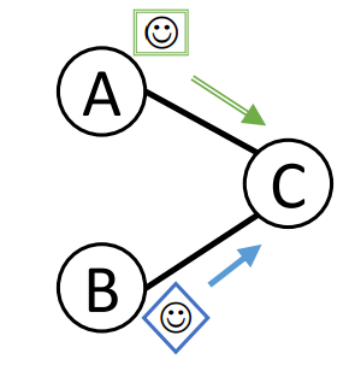
Outlines

- Introduction
  - Problem Definitions
  - Related Works
- Single-Objective Cooperation
  - Methodology (SACHA)
  - Empirical Evaluation
- Bi-Objective Cooperation
  - Methodology (MFC-EQ)
  - Empirical Evaluation
- Conclusion and Future Work

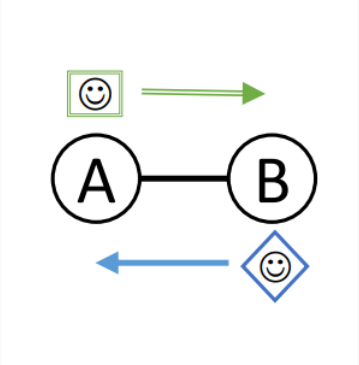
Introduction

Problem Definitions: Single-Objective Cooperation





Vertex Collision



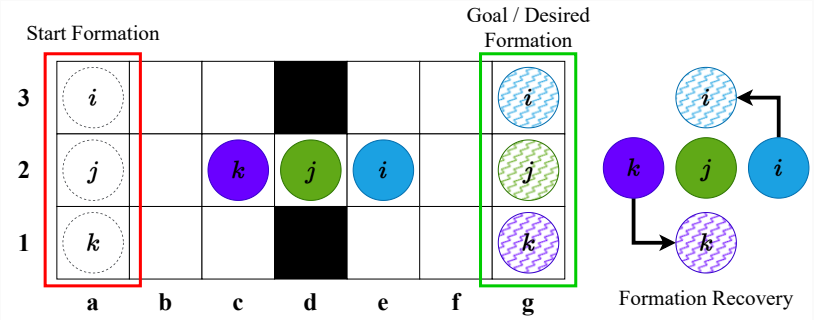
Edge Collision

Multi-Agent Path Finding (MAPF)[5]

- a connected and undirected graph and a set of M agents
- Each agent has a unique start vertex and a unique goal vertex.
- For each time step, each agent can either **move** to one of its adjacent vertices or **wait** at its current vertex.
- Collisions: vertex collisions and edge collisions
- The goal is to find a set of **collision-free** path, one for each agent, while minimizing the **flowtime** (i.e., sum of all path length)

[5] Stern et al., "Multi-agent pathfinding: Definitions, variants, and benchmarks". In SoCS, 2019.

Problem Definitions: Bi-Objective Cooperation



Moving Agents in Formation (MAiF)[6]

- maintain close adherence to a designated formation (specified by goal locations) while moving towards the goals
- The formation deviation $\mathscr{F}_t$ quantifies the least effort required to transform from the current formation to the desired formation:

$$\mathscr{F}_t := \min_{\boldsymbol{\Delta}} \sum_{i=1}^{M} \|\boldsymbol{u}^i - (\boldsymbol{v}^i + \boldsymbol{\Delta})\|_1,$$

  where $\boldsymbol{\Delta}$ is the element-wise median of $\{\boldsymbol{u}^i - \boldsymbol{v}^i\}_{i\in[M]}$
- The goal is to minimize (i) both the total (average) **formation deviation** (summing up $\mathscr{F}_t$ over t) and the **makespan** (i.e., maximum of all path length) or (ii) a linear combination of them

[6] J. Li et al., "Moving agents in formation in congested environments". In AAMAS, 2020

Related Works: Planning

MAPF (Single-Objective Cooperation)

- Conflict-Based Search[7](CBS): A two-level search algorithm. The high level constructs a **constraint search tree** by adding constraints to different nodes, while the low-level plans paths w.r.t these constraints via **A* search**
- Priority-Based Search[8](PBS): A two-level search algorithm. The high level constructs a **priority search tree** by adding **partial orderings** to different nodes, while the low-level plans paths w.r.t these partial orderings via **prioritized planning**

MAiF (Bi-Objective Cooperation)

- SWARM-MAPF[9](SWARM): A two-phase method combining swarm-based formation control with MAPF algorithms. Phase 1 selects a **leader**, plans its path, and then partitions it into segments; Phase 2 runs **conflict-based search** to plan paths for other agents to follow the leader in those congested segments
- Scalarized Prioritized Planning (SPP)[10]: Plan agents' paths one by one using a specific (or random) ordering and optimize over the scalarized objective

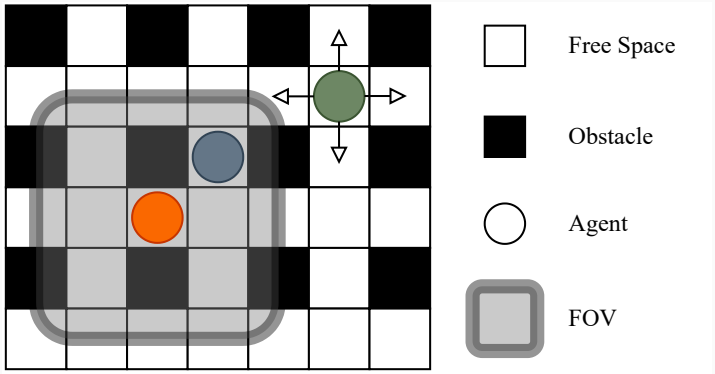
[7] Sharon et al., "Conflict-based search for optimal multi-agent pathfinding". In Artificial Intelligence, 2015.
[8] H. Ma et al., "Searching with consistent prioritization for multi-agent path finding". In AAAI, 2019.
[9] J. Li et al., "Moving agents in formation in congested environments". In AAMAS, 2020.
[10] Silver, "Cooperative pathfinding". In AAAI, 2015.

Related Works: Learning Environments

There are some universal settings that learning-based methods follow which we adopt as our learning environments

Reward Function Design

Action	Reward
Move (up / down / left / right)	-0.075
Wait (on goal, away goal)	0, -0.075
Collision (obstacles or agents)	-0.5
Reaching Goal	3



- 2-Dimensional 4-neighbor Grid Worlds: 2D grids environments where agents are only allowed to move along 4 cardinal directions
- Reward Design: we penalize each move with a small negative reward to incentivize agents to reach their goals as fast as possible
- Partially Observable Environments: each agent can only observe its surrounding $\mathcal{L} \times \mathcal{L}$ area, namely field-of-view (FOV)
- Homogeneous Multi-Agent Systems: each agent shares the same policy but makes different decisions based on their observations

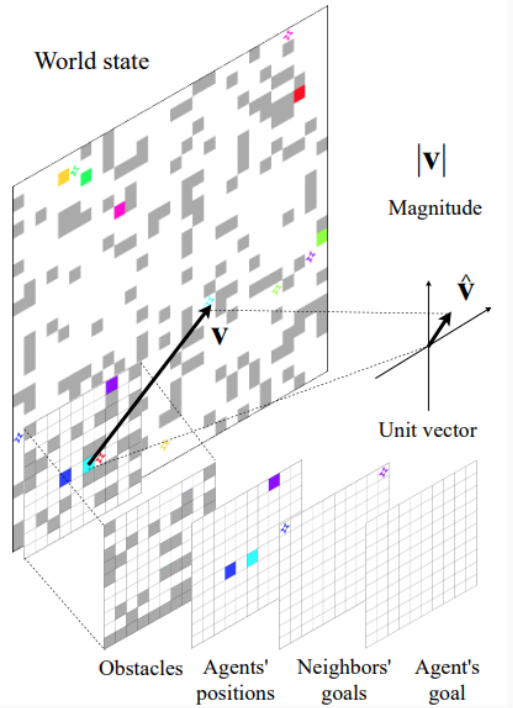
Related Works: Learning

PRIMAL[11]

- Mixture of A3C (RL) and Behavior Cloning (IL)
- Only observe **goal direction** as path planning guidance without considering the obstacles
- Use neighbouring agents' **goal directions** as cooperative guidance (not informative for complex cooperation)

DHC[12] and DCC[13]

- Independent *Q*-Learning (IQL)
- Embed **communication** models
- Utilize **single-agent heuristic maps** as path planning guidance (equivalent to obstacle avoidance)
- No explicit guidance for cooperation

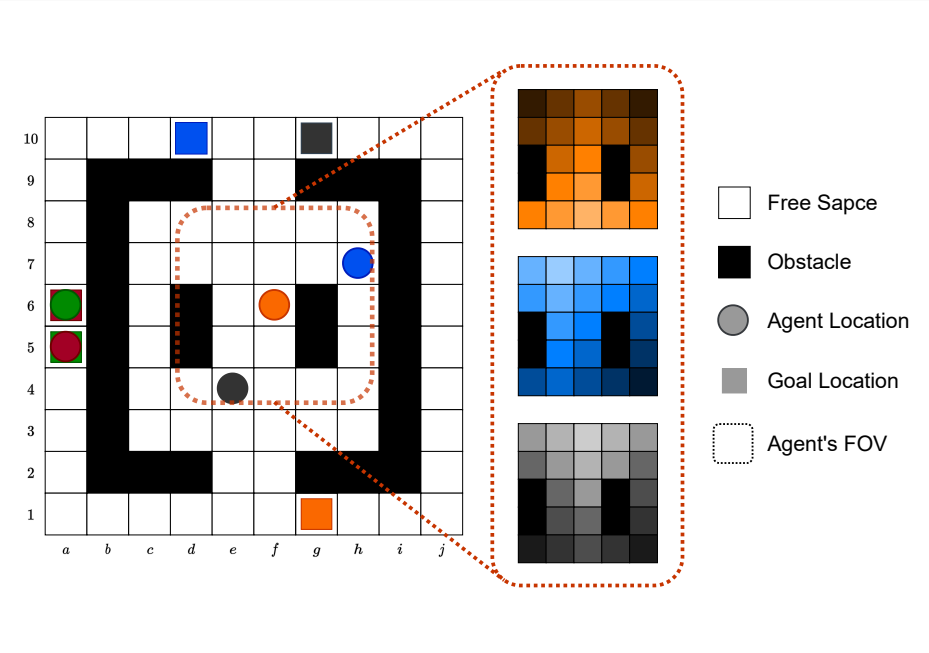


System design from PRIMAL

[11] Sartoretti et al., "Primal: Pathfinding via reinforcement and imitation multi-agent learning". In IEEE RA-L, 2019.
[12] Z. Ma, Luo, and H. Ma, "Distributed heuristic multi-agent path finding with communication". In ICRA, 2021.
[13] Z. Ma, Luo, and Pan, "Learning selective communication for multi-agent path finding". In IEEE RA-L, 2021.

Single-Objective Cooperation

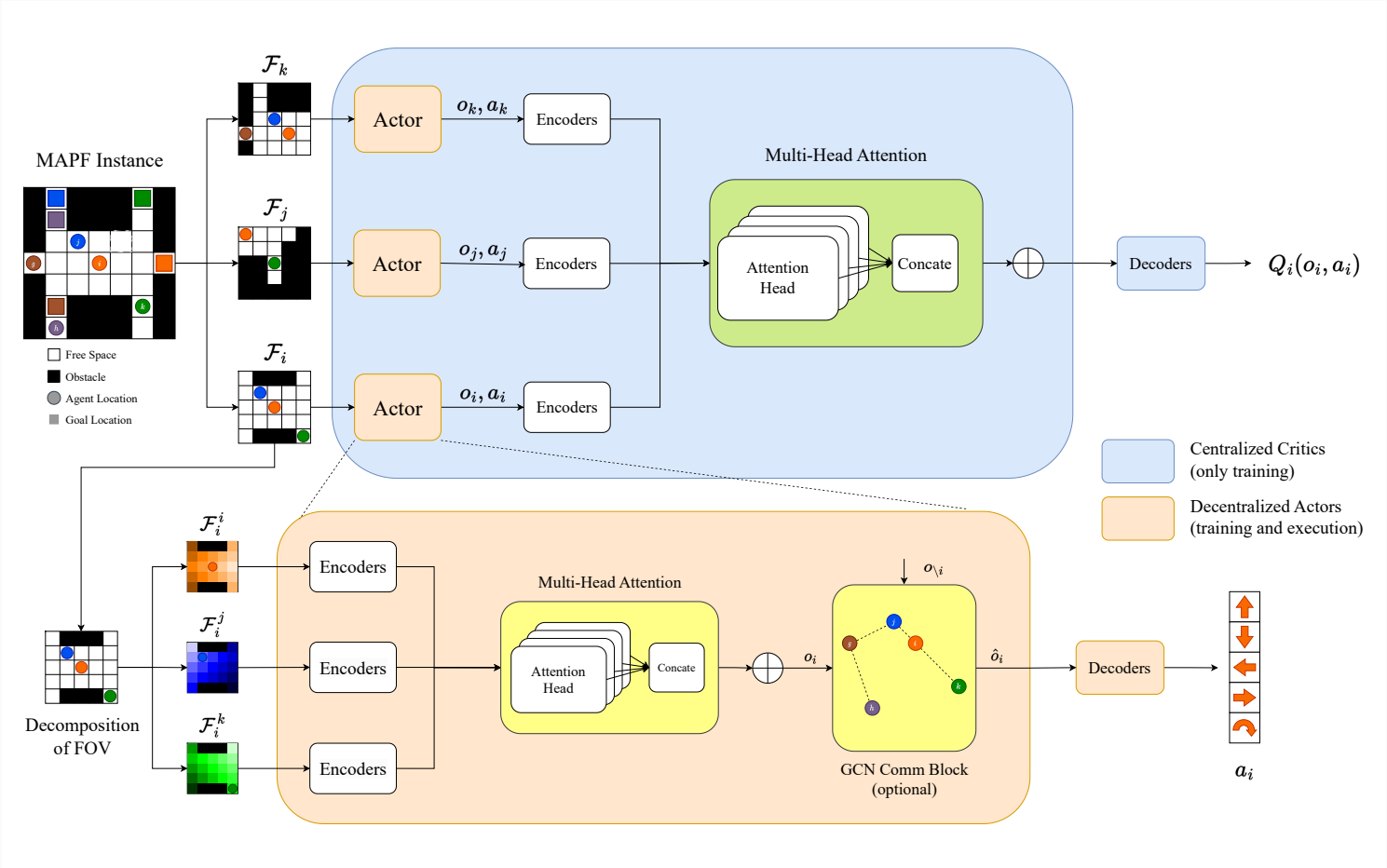
SACHA: Multi-Agent Heuristic Maps



Example of multi-agent heuristic maps. A darker shade means a larger heuristic.

- Make use of heuristic maps from not only the center agent but also its neighboring agents
- Each cell holds a heuristic value proportional to its shortest path distance to the goal, and these heuristic maps inform each agent about its paths and neighboring agents' potential plans
- These heuristics can be pre-computed before execution with polynomial-time algorithms (e.g., Dijkstra) and will stay constant during execution

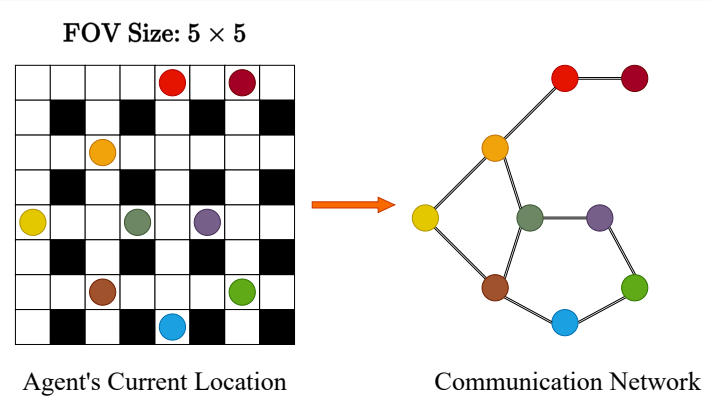
SACHA: Soft Actor-Critic and Heuristic-Based Attention



Centralized Training and Decentralized Execution (CTDE) and Multi-Agent Actor-Critic

- Policy network with heuristic-based attention for greater cooperative potentials
- Partially centralized attention critic network for better credit assignment

SACHA: Graph-Based Communication



We also proposed another communication-based variant, named SACHA(C)

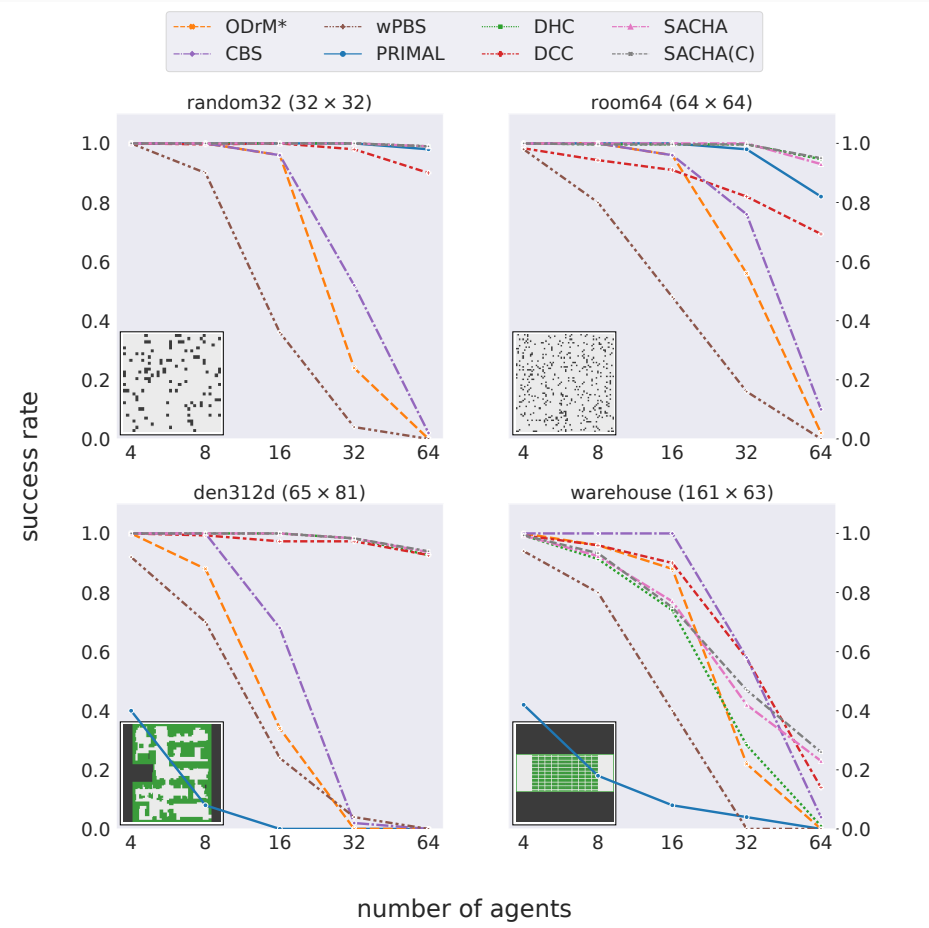
- We establish a dynamic communication network G_t that depends on agents' current positions: each vertex represents one agent, and each edge means two connecting agents lie within each other's FOV
- We run two-layer GCN[14]to encode, re-normalize, and pass messages along the communication network
- It has been shown by other works (e.g., Li et al.[15]) that this technique can encourage communication among agents

[14]Kipf and Welling, "Semi-supervised classification with graph convolutional networks". In arXiv, 2016.
[15]Li et al., "Graph neural networks for decentralized multi-robot path planning". In IROS, 2020.

SACHA: Empirical Evaluation

- Besides learning-based methods (PRIMAL, DHC, and DCC), we also compare our methods with some planning algorithms with runtime limits: CBS (120s), PBS(120s), and OrDM*(20s)

Comparison of success rate in different maps



SACHA: Empirical Evaluation

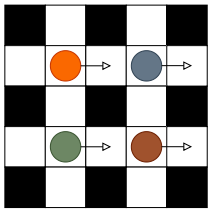
Comparison of solution quality

Map	Agents	Average Step Per Agent				
		PRIMAL	DHC	DCC	SACHA	SACHA(C)
random32	4	32.96	35.70	32.83	**29.93**	31.03
	16	45.12	48.67	43.56	41.71	**41.30**
	64	69.40	**66.05**	88.79	76.47	74.48
random64	4	67.82	71.04	70.80	**65.47**	67.10
	16	89.22	94.22	102.27	83.74	**82.17**
	64	105.12	120.68	154.72	99.02	**96.42**
den312d	4	196.54	86.56	82.99	**78.33**	81.43
	16	~~256.00~~	109.24	108.29	97.86	**96.74**
	64	~~256.00~~	153.17	145.21	**140.79**	142.97
warehouse	4	355.80	146.12	135.89	**131.43**	134.59
	16	492.04	281.37	208.72	**192.30**	198.72
	64	~~512.00~~	~~512.00~~	473.92	449.83	**437.29**

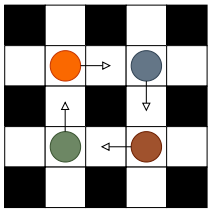
- PRIMAL has the worst performance since behavior cloning from experts (planning algorithms) hinders the trained model from generalizing toward unseen environments
- Our methods can outperform DHC and DCC in most cases

Bi-Objective Cooperation

MFC-EQ: Mean Field Formation Control



$\bar{a} = [0, 0, 0, 1, 0]$



$\bar{a} = [0.25, 0.25, 0.25, 0.25, 0]$

Mean Field Reinforcement Learning[16]

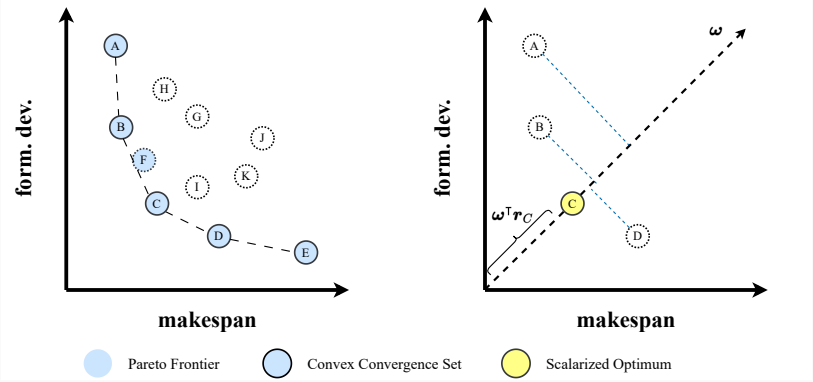
- Approximate interactions within agents by those between every single agent and the average effect from the overall population

$$Q^j(s, \{a^k\}_{k\in[M]}) = \frac{1}{M}\sum_{k=1}^{M} Q^j(s, a^j, a^k) \approx Q^j(s, a^j, \bar{a}),$$

  where $\bar{a} = \sum_{k=1}^{M} a^k$ is the mean action
- Avoid the exponential growth of agents' interactions (the curse of dimensionality) and thus enhance scalability
- Most importantly, **mean action can reflect on the formation change**

[16] Y. Yang et al., "Mean field multi-agent reinforcement learning". In ICML, 2018

MFC-EQ: Envelop Q-Learning



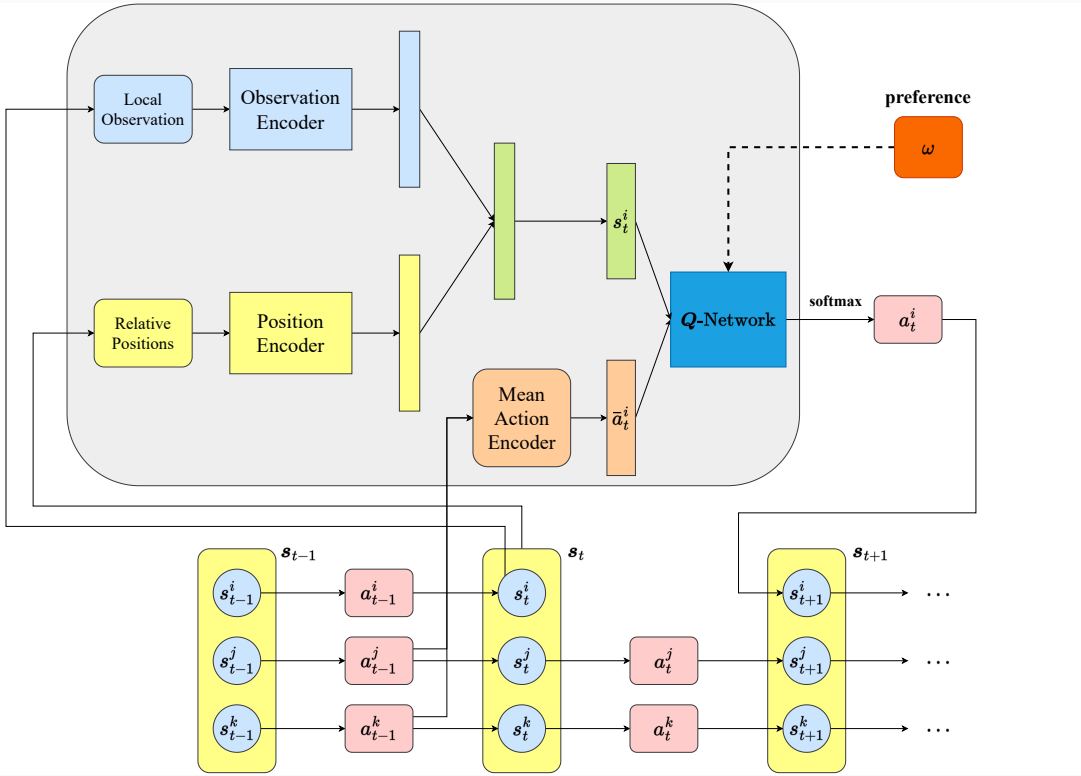
- Reward: $r_t^j = (c_t^j, \mathscr{F}_t^j)^\intercal$, where c_t^j is the moving cost and $\mathscr{F}_t^j$ is agent j's contribution to the formation deviation
- Goal: learn a universal model to minimize $\sum_{t=0}^{T} \gamma^t \boldsymbol{\omega}^\intercal (\sum_{j\in[M]} r_t^j)$ for **any given linear preference** $\boldsymbol{\omega} = (\lambda, 1-\lambda)^\intercal \in \Omega$,
- We adopt the Envelope Q-Learning[17] to tackle the bi-objective optimization in the multi-agent settings (optimistic approach)

$$(\mathcal{T}\boldsymbol{Q})(\boldsymbol{s},\boldsymbol{a},\boldsymbol{\omega}) := \boldsymbol{r}(\boldsymbol{s},\boldsymbol{a}) + \gamma\, \mathbb{E}_{s'\sim\mathcal{P}(\cdot|\boldsymbol{s},\boldsymbol{a})}\, \arg_Q \left\{ \max_{\boldsymbol{\omega}'\in\Omega} \max_{\boldsymbol{a}'} \boldsymbol{\omega}^\intercal \boldsymbol{Q}(\boldsymbol{s}',\boldsymbol{a}',\boldsymbol{\omega}') \right\}$$

where $\arg_Q$ takes the Q-value that corresponds to the maximal $\boldsymbol{\omega}^\intercal \boldsymbol{Q}$

[17] R. Yang, Sun, and Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation". In NeuIPS, 2019.

MFC-EQ: Model Design



- Observation: encode agents' observation inside the FOV
- Position: encode agents' relative positions with others for formation control
- Mean Action: $\bar{a}$ from mean field reinforcement learning
- Preference: $\boldsymbol{\omega} = (\lambda, 1-\lambda)^\intercal$ from envelope Q-learning

MFC-EQ: Empirical Evaluation

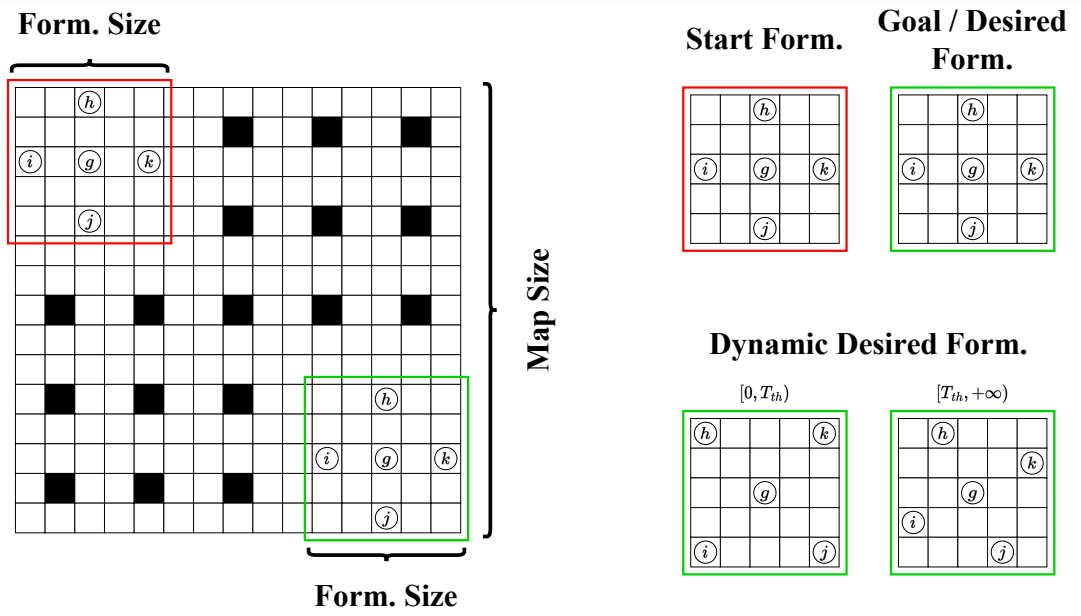


Illustration of experimental settings for MAiF

- The start position is located at the top-left corner, and the goal / desired formation lies at the bottom-right corner
- Agents travel from the top-left corner to the bottom-right corner while minimizing makespan and formation deviation
- We also conduct the dynamic formation experiment in which we alter the desired formation after a certain time threshold T_{th}

MFC-EQ: Empirical Evaluation

- Define the *MIX* metric: $MIX(\lambda) = \lambda T + (1-\lambda) \cdot \frac{\sum_{t=0}^{T} \mathscr{F}_t}{M}$
- We first test the model by setting $\omega = (0.5, 0.5)^\intercal$

Comparison of solution quality in different sizes of maps

		Success Rate			MIX(0.5)		
Map Size	M	SPP	SWA-RM	MFC-EQ	SPP	SWA-RM	MFC-EQ
32	10	1.00	1.00	1.00	39.05	32.70	**32.30**
×	20	1.00	0.99	0.99	40.73	37.78	**35.21**
32	30	0.79	0.96	0.90	46.90	39.87	**37.46**
48	10	1.00	0.99	0.99	67.18	53.14	**49.56**
×	20	0.95	0.99	0.96	76.26	66.27	**62.89**
48	30	0.74	0.94	0.88	90.48	**68.91**	72.30
64	10	1.00	0.99	0.99	105.65	79.77	**76.79**
×	20	1.00	0.97	0.93	114.04	94.64	**84.80**
64	30	0.22	0.98	0.90	111.62	**99.98**	103.47

MFC-EQ: Empirical Evaluation

- Test for adaptability towards various linear preferences

$\omega(\lambda)$	Makespan	Form. Dev.	MIX(0.1)	MIX(0.3)	MIX(0.5)	MIX(0.7)	MIX(0.9)
0.1	106.33	14.67	**23.84**	42.17	60.50	78.83	97.16
0.3	101.14	15.37	23.95	**41.10**	58.26	75.41	92.56
0.5	98.64	16.84	25.02	41.38	**57.74**	74.10	90.46
0.7	96.74	19.16	26.92	42.43	57.95	**73.47**	88.98
0.9	96.42	21.75	29.22	44.15	59.09	74.02	**88.95**

- Dynamic Formations: require agents to change formation midway

	Success Rate			MIX(0.5)		
M	SPP	SWA-RM	MFC-EQ	SPP	SWA-RM	MFC-EQ
10	1.00	0.98	0.96	88.05	115.70	**80.47**
15	1.00	1.00	1.00	90.50	137.34	**85.95**
20	0.97	1.00	1.00	95.41	135.94	**88.75**
25	0.72	1.00	1.00	100.09	133.46	**92.25**
30	0.90	0.98	0.93	98.41	123.20	**96.12**
35	0.48	0.94	0.87	107.58	123.76	**98.89**
40	0.25	0.81	0.74	110.66	109.59	**101.31**

Conclusion and Future Work

Conclusion and Future Work

Conclusions

- **SACHA and SACHA(C)**[18]address the issues of learning single-objective cooperation in partial observable multi-agent path finding (**MAPF**), with a focus on **generalizability** among different environments
- **MFC-EQ**[19]tackles the challenges of learning bi-objective cooperation in decentralized moving agents in formation (**MAiF**), with a focus on **scalability** towards large-scale instances and **adaptability** towards any linear preferences

Future Work

- To improve generalizability, a more generalized scheme is to consider meta-learning, in which one can pre-train a model as initialization and fine-tune to different environments
- To solve multi-object path planning tasks, we can apply multi-objective RL algorithms (e.g., Pareto Q-learning) to directly approach the Pareto frontier
- Design algorithms/frameworks for more sophisticated types of cooperation in multi-agent path planning

[18]published paper in IEEE Robotics and Automation Letters 2023

[19]submitted paper under review

The End

Thank you for listening!