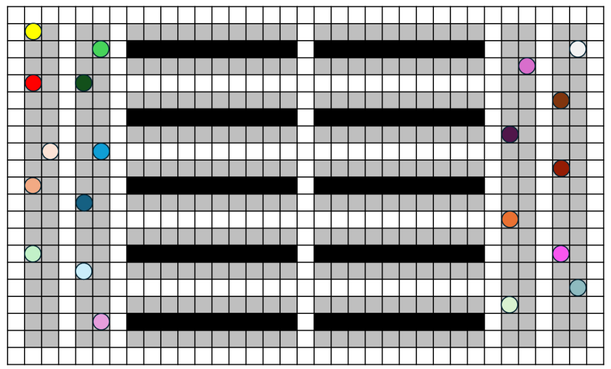
STAR-GNN: Scalable Graph Neural Reinforcement Learning for Collision-Aware Online Multi-Robot Task Allocation

Yifan Yang, Jiaqi Tan, Hang Ma

Autonomous Intelligence & Robotics Lab @ Simon Fraser University

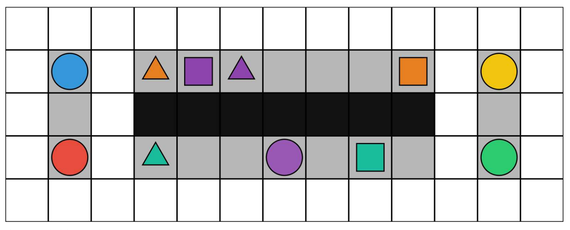
1. Multi-Agent Pickup and Delivery



What is MAPD?

In Multi-Agent Pickup and Delivery (MAPD), robots repeatedly receive dynamically arriving pickup-and-delivery tasks. Each task has a pickup location, a delivery location, and a release time. Robots must execute assigned tasks while maintaining collision-free paths.

At each event time, the system assigns pending tasks to available or soon-to-be-available robots, then calls a Multi-Agent Path Finding (MAPF) planner to compute collision-free paths.



circle: agent triangle: pickup square: delivery

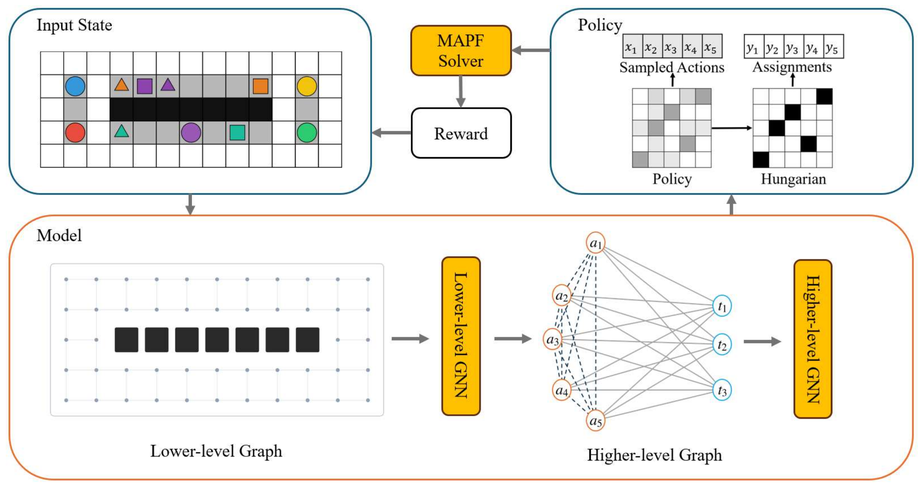
Why Is Task Allocation Hard?

Most scalable MAPD systems use a two-stage pipeline:
Task allocation -> **collision-free path planning**

The assignment step is often based on shortest-path distance. However, shortest paths do not capture congestion, shared corridors, or future collision-avoidance delays.
Planner-evaluated assignment costs are more accurate, but repeatedly querying a planner during assignment search can be expensive.

We train assignment as a planner-feedback RL problem: the policy selects a feasible agent-task matching and receives reward from the solver's completion cost.

2. Planner-Aware STAR-GNN



Algorithm 1: Online MAPD Paradigm

```
t ← 0, T ← ∅;
while not all tasks completed do
    if ∃τ_j : t_j^rel = t or ∃ agent becomes free at t then
        T ← T ∪ {τ_j | t_j^rel = t};
        TaskAssignment(T);
        PathPlanning();
    t ← t + 1;
    Each agent moves one step; remove each τ_j from T if
      it starts execution;
```

Key idea

STAR-GNN operates inside the standard event-driven online MAPD loop. A new decision is triggered when a task is released or an agent becomes free. The system updates the pending task pool, assigns tasks to candidate agents using STAR-GNN, replans collision-free paths with PBS, and executes the resulting paths until the next event.

Instead of relying only on shortest-path distance, STAR-GNN predicts assignment scores that reflect map topology, task distribution, traffic, and inter-agent interactions.

Lower-level graph: learns congestion-aware spatial representations from the warehouse map and current traffic.
Higher-level graph: scores feasible agent-task matchings while accounting for inter-agent interactions.
Decoding: uses Hungarian matching to produce a one-to-one assignment.
Reward: Reduction in planner-estimated total completion time.

$$r_k = -\Big(\hat{L}(s_k, u_k) - \hat{L}(s_{k-1}, u_{k-1})\Big).$$

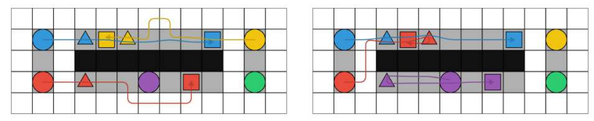
3. Training

1. **Spatial pretraining:** learn local traffic and task-density representations.
2. **Hungarian warm start:** initialize assignment scores from feasible distance-based matching.
3. **Planner-based RL fine-tuning:** improve assignments using solver-evaluated reward.

4. Results

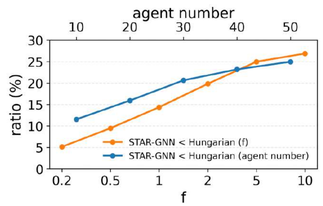
λ	M	**STAR-GNN**	CENTRAL	LNS-PBS	LNS-wPBS	RMCA	HBH+MLA*	Hun+PBS
0.2	10	27.14	29.77	27.41	27.49	**26.74**	28.79	27.79
	20	24.93	26.70	24.60	24.84	**24.28**	24.98	24.61
	30	23.29	25.56	23.51	24.11	**23.27**	23.77	23.51
	40	23.03	25.46	23.26	24.06	**22.62**	23.20	23.23
	50	22.65	25.05	22.70	23.49	**22.37**	22.86	22.77
0.5	10	105.49	109.71	**99.81**	102.67	101.62	156.78	106.94
	20	25.84	27.99	26.30	26.52	**25.44**	29.49	26.62
	30	23.79	26.23	24.22	25.54	**23.66**	26.08	24.24
	40	**22.56**	25.39	23.51	24.20	22.73	25.18	23.52
	50	**22.19**	24.94	23.19	23.68	22.44	24.65	23.14
1	10	**243.73**	285.75	252.10	247.85	269.76	288.03	249.95
	20	63.21	75.13	63.83	60.26	**59.12**	93.82	64.30
	30	**25.14**	31.41	27.68	26.96	25.59	48.55	27.56
	40	**23.02**	28.33	24.41	25.22	23.67	29.12	24.65
	50	**22.66**	27.38	23.49	24.79	23.01	27.59	23.54
2	10	**328.95**	388.21	334.02	334.65	371.27	363.59	334.73
	20	**130.51**	162.0	130.66	130.59	146.81	162.91	132.27
	30	71.03	85.89	71.39	**70.13**	77.75	96.62	71.44
	40	**40.22**	57.53	44.52	42.38	43.49	67.98	43.69
	50	**27.96**	41.43	29.43	28.86	28.88	53.18	29.19
5	10	**387.41**	455.16	395.13	391.86	435.70	421.44	394.27
	20	**182.36**	229.55	186.69	185.10	209.55	205.71	187.53
	30	**117.17**	147.76	121.07	117.89	132.06	138.15	121.10
	40	**85.46**	108.28	87.34	86.39	96.81	105.45	87.71
	50	**67.60**	86.90	68.63	68.37	74.32	87.57	69.92
10	10	**410.28**	478.17	411.51	412.31	458.23	438.40	412.96
	20	**203.94**	242.18	207.16	205.71	228.90	227.18	207.59
	30	**136.65**	165.13	139.56	137.87	154.28	157.27	140.46
	40	**102.43**	128.39	105.87	102.55	115.04	122.83	106.89
	50	**82.77**	106.70	87.17	83.61	94.29	102.73	85.90

STAR-GNN achieves the best or near-best average service time in most warehouse setting. The advantage is strongest when task release frequency and fleet size increase.



(a) Hun+PBS **(b)** STAR-GNN

Hun+PBS selects locally short assignments based on shortest-path costs, but these assignments create stronger downstream interference after planning. STAR-GNN selects a different assignment and reduces total planned path length from 26 to 23.



The fraction of decision steps where STAR-GNN improves over Hun+PBS increases with task frequency and fleet size, showing that learned planner-aware assignment is most useful in dense traffic.