

Multi-Agent Path Finding for Schedule Constrained Automation (Extended Abstract)*

Kareem Eissa,[†] Royal Prasad, Ankur Kapoor

Siemens Healthineers, Digital Technology and Innovation, Princeton, NJ, 08540, USA
{royal.prasad, ankur.kapoor}@siemens-healthineers.com

Abstract

In modern automation settings, jobs are processed across machines with interdependencies and are subject to limited equipment availability. When transportation between machines is considered, the problem evolves into a complex multi-agent routing task with operational constraints. Existing Multi-Agent Path Finding (MAPF) algorithms address challenges such as robustness and uncertainty, but practical applications involving scheduling constraints often require considerable manual effort for adaptation and heuristic design. In this paper, we introduce MAPF-SC, an extension of MAPF that incorporates scheduling constraints for continuous task flows. We explore the challenges of applying existing techniques to this problem, emphasizing the engineering effort involved in addressing these constraints. Our evaluation investigates the impact of temporal and topological variations on performance, highlighting key factors that influence real-world automation scenarios.

Introduction

In automation systems, jobs are transported to a sequence of processing machines and must adhere to an operating schedule. This means that the number of goal locations (i.e., machines) is finite, and agents have a time-window within which to visit each location, followed by a non-zero processing time at each of them. This can be seen in Fig. 1, where three jobs (i.e., agents) must visit three machines (i.e., goal locations) at certain defined time windows shown on the vertical axis.

Existing MAPF approaches mostly concentrate on finding collision-free paths for agents to reach their goals as quickly as possible. Search-based methods are the most popular, with state-of-the-art algorithms providing bounded sub-optimality guarantees and scalable heuristics for MAPF instances (Li et al. 2022). Decentralized Reinforcement Learning (RL) approaches have also shown promise (Damani et al. 2021), with adaptability to constraint variations through reward modeling. These methods use search-based methods through imitation learning to encourage agent cooperation.

*An early version of this paper was presented at WoMAPF25 (Workshop on Multi-Agent Path Finding at AAAI 2025)

[†]Work done while at Siemens Healthineers, Princeton, NJ
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

There is limited work, however, that includes temporal constraints with processing times into MAPF. While some recent search-based methods have incorporated these (Chong, Li, and Sycara 2024), they are mostly one-shot approaches and rely on heuristics and domain-specific data structures, making them difficult to generalize.

We formulate MAPF-SC, a lifelong variant of MAPF with scheduling constraints. We propose a Multi-Agent RL solution, building on existing work and introducing spatio-temporal encodings. We evaluate our method using temporal and topological variations based on real-world automation.

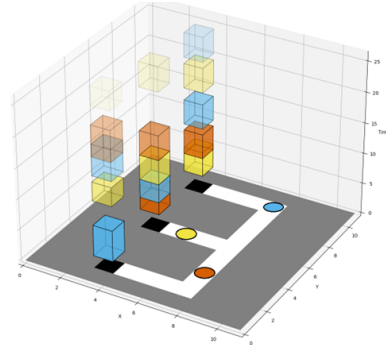


Figure 1: Example MAPF-SC instance. The bottom XY plane shows the agents as circles, goals as black squares, and gray areas as non-traversable. The added rectangles along the vertical time axis represent schedules for each agent, indicating the arrival and processing time at the goal location.

Problem Formulation

The MAPF-SC problem is defined by an undirected graph $G = (V, E)$, a set of m agents i_1, \dots, i_m , and a schedule S . In this lifelong setting, agent i is assigned a sequence of n goals j_1^i, \dots, j_n^i in accordance to S , with each goal j consisting of:

- Goal location vertex $\mathbf{v}(i, j) \in V$
- Earliest arrival time $\mathbf{A}_t(i, j)$ where an agent i is not expected at the goal location j before this time.
- Deadline $\mathbf{D}_t(i, j)$ where an agent i must be at the goal location j no later than this time.
- Goal processing time $\mathbf{P}_t(i, j)$ where the agent i is stationary at the goal location j for this fixed duration.

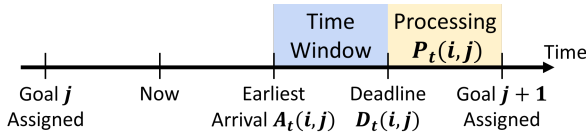


Figure 2: Different scheduling constraints in MAPF-SC.

Further, precedence constraints $C(v)$ require that multiple agents i_1, \dots, i_m assigned to the same v (with different scheduled time-windows) be processed in sequence. These are enforced implicitly, and agents can query their queue location. Fig. 2 shows the temporal components of each goal j . Various MAPF settings can easily be represented by adjusting the constraints – e.g., classical MAPF has $\forall i, j A_t(i, j) = D_t(i, j)$ and $P_t(i, j) = 0$, with no explicit precedence constraints ($C = \emptyset$).

Method

We propose an RL approach, modeling the MAPF-SC problem as a Partially Observable Markov Decision Process (POMDP) (S, O, A, P, R, γ) . We restrict our environments to 2D grids, and each agent observes a local field of view (FOV) that captures information of the environment in a limited grid around it (Damani et al. 2021). We aim to learn a homogeneous policy for decentralized execution by agents.

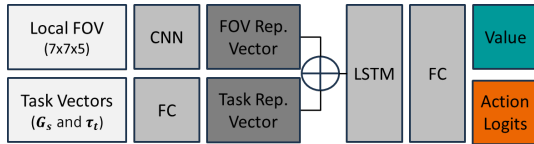


Figure 3: Architecture of our proposed RL-based method.

In addition to the FOV observation as shown in Fig. 3, we include two task vectors that describe the agent’s next goal:

- Goal vector G_s includes the relative in x/y -displacements and the total goal distance, similar to (Damani et al. 2021).
- Temporal vector τ_t includes the time until A_t , time until D_t , and an indicator marking the delay past D_t .

A key advantage of our RL formulation is its flexibility – by modifying the observations and rewards, we can train agents for new automation domains with arbitrary constraints. This is in contrast to search-based methods that provide better optimality with guarantees but require significant efforts to design heuristics and data structures for each new scenario.

Experimental Results

We hand-craft a set of layouts with atomic spatial variations such as padding cells and multi-path redundancy that helps us isolate spatial and temporal effects.

We create schedules with parametric temporal features (Mohanty et al. 2020) that ensure feasibility:

- Single-agent shortest distance (A^*) which is the minimum distance the agent travels assuming no collisions.
- Congestion estimate using the average of all A^* distances.

These influence A_t , and we then vary the time-windows ($D_t - A_t$) and P_t — the schedules are then translated to goal sequences with time and precedence constraints.

We use two types of schedules: tight schedules with limited A^* factor range and task durations, and relaxed schedules with both parameters expanded. Fig. 4 shows the layouts and average completed tasks for 100,000 training episodes. Note that across layouts, tight schedules have a higher absolute task count for a fixed horizon. Padding and multi-path layouts help improve task completion, although very restrictive layouts (highlighted in light/dark blue) that require higher degrees of coordination are bottlenecked by decentralized RL.

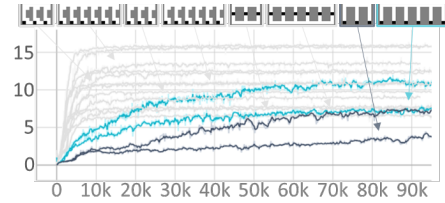


Figure 4: Average completed tasks (tight/relaxed schedules).

Conclusion and Future Work

In this work, we introduce a variant of MAPF to model scheduling and precedence constraints in real-world automation scenarios. Decentralized RL shows promising results with robustness to temporal schedule variations in the schedules. However, restricted layouts without redundancy are challenging as they need long-horizon coordination.

We will explore incorporating search-based methods similar to recent approaches (Jiang et al. 2025; Chong, Li, and Sycara 2024) to train stronger policies via imitation learning.

References

- Chong, Y. Q.; Li, J.; and Sycara, K. 2024. Optimal Task Assignment and Path Planning Using Conflict-Based Search with Precedence and Temporal Constraints. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS '24*, 2210–2212.
- Damani, M.; Luo, Z.; Wenzel, E.; and Sartoretti, G. 2021. PRIMAL\$ 2\$: Pathfinding Via Reinforcement and Imitation Multi-Agent Learning - Lifelong. *IEEE Robotics and Automation Letters*, 6(2): 2666–2673.
- Jiang, H.; Wang, Y.; Veerapaneni, R.; Duhan, T. H.; Sartoretti, G. A.; and Li, J. 2025. Deploying Ten Thousand Robots: Scalable Imitation Learning for Lifelong Multi-Agent Path Finding. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Li, J.; Chen, Z.; Harabor, D.; Stuckey, P. J.; and Koenig, S. 2022. Fast Repairing for Multi-Agent Path Finding via Large Neighborhood Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9): 10256–10265.
- Mohanty, S.; Nygren, E.; Laurent, F.; Schneider, M.; Scheller, C.; Bhattacharya, N.; Watson, J.; Egli, A.; Eichenberger, C.; Baumberger, C.; Vienken, G.; Sturm, I.; Sartoretti, G.; and Spigler, G. 2020. Flatland-RL : Multi-Agent Reinforcement Learning on Trains.