Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling (ICAPS 2026)

Analyzing Planner Design Trade-Offs for MAPF Under ADG-Based Realistic Execution

Jingtian Yan[1], Zhifei Li[2], William Kang[1], Stephen F. Smith[1], Jiaoyang Li[1]
[1]Carnegie Mellon University,
[2]Massachusetts Institute of Technology
jingtianyan@cmu.edu, zhifeili@mit.edu, williamkang@cmu.edu, sfs@cs.cmu.edu, jiaoyangli@cmu.edu

Abstract

Multi-Agent Path Finding (MAPF) algorithms are increasingly deployed in industrial warehouses and automated manufacturing facilities, where robots must operate reliably under real-world physical constraints. However, existing MAPF evaluation frameworks typically rely on simplified robot models, leaving a substantial gap between algorithmic benchmarks and practical performance. Recent frameworks such as SMART combine kinodynamic modeling with execution based on the Action Dependency Graph (ADG), enabling realistic, large-scale MAPF evaluation. Building on this capability, this work investigates how key planner design choices influence performance under realistic execution settings. We systematically study three fundamental factors: (1) the relationship between solution optimality and execution performance, (2) the sensitivity of system performance to inaccuracies in kinodynamic modeling, and (3) the tradeoff between model accuracy and plan optimality. Empirically, we examine these factors to understand how these design choices affect performance in realistic scenarios. We highlight open challenges and research directions to steer the community toward practical, real-world deployment.

1 Introduction

Multi-Agent Path Finding (MAPF) has emerged as a critical challenge for coordinating robot fleets in automated warehouses, manufacturing facilities, and logistics centers (Ma et al. 2017; Hönig et al. 2019; Ho et al. 2022). As these systems continue to grow in scale and complexity, the demand for reliable and high-performance MAPF planners has intensified. Despite significant progress in algorithm design, a persistent challenge remains: most existing MAPF evaluation frameworks rely on overly simplified robot models, neglecting realistic kinodynamic constraints, execution-time variability, and other physical characteristics inherent to robotics systems. This gap limits the ability of researchers and practitioners to accurately assess how planner design choices translate into performance in realistic settings.

Some evaluation tools (Heuer et al. 2024; Yan et al. 2025) are proposed to evaluate the performance of MAPF methods in realistic settings. However, only limited work has studied how planner design choices translate into execution performance in realistic scenarios. To the best of our knowledge, Varambally, Li, and Koenig (2022) present the only study in this direction. Their work compares different MAPF models used during planning in automated warehousing environments, providing valuable initial insights. Nevertheless, their evaluation is limited to less than 50 robots, focuses solely on a warehouse layout, and does not systematically analyze how different planner design factors interact and collectively affect execution outcomes.

In this work, we investigate how key planner design choices influence MAPF performance under realistic settings using SMART (Yan et al. 2025), a state-of-the-art testbed. Built on the Action Dependency Graph (ADG) (Hönig et al. 2019), SMART is a physics-based simulation framework that enables execution-aware evaluation of MAPF planners in realistic settings and supports large-scale experiments with thousands of robots. Leveraging this capability, we systematically study the impact of three fundamental factors on execution performance: (1) the plan optimality, (2) the accuracy of the kinodynamic model used during planning, and (3) the trade-off between model accuracy and plan optimality. Through controlled experiments in SMART, we show how these factors influence execution behavior in practice. Our analysis uncovers key planner design trade-offs and outlines open challenges and research directions for advancing MAPF toward real-world deployment.

2 Background

In this section, we first introduce the MAPF problem and related work. Next, we discuss existing evaluation tools.

2.1 MAPF Problem

A *MAPF problem* (Stern et al. 2019) is typically defined on a discrete undirected graph $\mathcal{G}_M = (\mathcal{V}_M, \mathcal{E}_M)$, with a set of robots $\mathcal{R} = \{r_1, \ldots, r_I\}$ where each robot has a start and a goal location. A *MAPF plan* consists of collision-free paths for all robots to reach their goals. In the standard formulation, time is discretized into timesteps, and each robot can either move to an adjacent vertex or wait. Collisions occur if two robots occupy the same vertex or traverse the same edge in opposite directions at the same timestep.

2.2 MAPF Planners and Design Choices

Many MAPF methods have been proposed. Some methods focus on finding solutions with optimal or bounded-suboptimal guarantees (Sharon et al. 2015; Barer et al. 2014; Li, Ruml, and Koenig 2021), while others focus on scalability, solving MAPF problems involving thousands of robots (Okumura et al. 2022). Anytime algorithms quickly find an initial solution and then iteratively refine it as time progresses, balancing solution quality with computational efficiency (Li et al. 2021a; Huang et al. 2022). Additionally, some methods incorporate complex real-world factors, such as kinodynamics and unexpected delays, into the planning process (Cohen et al. 2019; Atzmon et al. 2020; Yan and Li 2024). Despite these advances, it remains unclear how such planner design decisions actually impact performance in realistic execution settings. Varambally, Li, and Koenig (2022) investigate how different assumptions in MAPF modeling affect practical performance in automated warehousing. However, its evaluation explores only a small set of MAPF modeling assumptions, focuses solely on warehouse layouts, and is limited to less than 50 robots.

2.3 Evaluating MAPF in Realistic Scenarios

Evaluating MAPF performance requires executing planned paths in settings that reflect real robotic behavior. Some tools have been developed to enable such assessment. MRP-Bench (Schaefer et al. 2023) utilizes Gazebo with integrated low-level controllers to account for kinodynamics and motion delays. The recently introduced SMART testbed (Yan et al. 2025) explicitly models robot kinodynamics, execution uncertainties, and communication delays. SMART scales to thousands of robots, making it well suited for evaluating modern MAPF planners.

The architecture of SMART integrates a physics-engine-based simulator, an execution monitoring server based on the Action Dependency Graph (ADG) (Hönig et al. 2019), and robot-specific executors. Given a MAPF plan, SMART first uses the execution monitoring server to convert the plan into an ADG, a directed graph consisting of two types of edges: (i) Type-1 (intra-robot) edges, which encode the temporal order of actions executed by the same robot, and (ii) Type-2 (inter-robot) edges, which encode precedence constraints between actions of different robots to prevent collisions. Then, the robot executors follow the ADG to issue motion commands to each robot while accounting for kinematic limits, delays, and disturbances. Finally, the simulator executes these commands in a physics-based simulation and returns realistic execution outcomes for evaluation.

3 From MAPF to Realistic Scenarios

In this section, we study research problems focusing on the gap between MAPF planning and robot execution. We use six simulation environments derived from the MovingAI benchmark (Stern et al. 2019) maps: `empty` (empty-32-32, size: 32×32), `random` (random-64-64-10, size: 64×64), `room` (room-64-64-16, size: 64×64), `den312d` (den312d, size: 65×81), `maze` (maze-32-32-4, size: 32×32), and `warehouse` (warehouse-10-20-10-2-1, size: 161×63). We use the Average Execution Time (AET) to measure the execution performance of a MAPF plan, which is the total simulated time that all robots take to complete their assigned paths divided by the number of robots. We adopt the

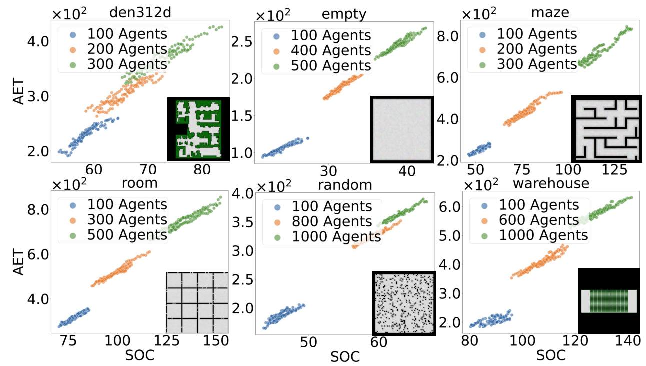

Figure 1: Relationship between SoC and AET. Each dot represents the SoC and AET of a single MAPF plan.

same robot configuration as used in the SMART paper: all robots are modeled as differential-drive platforms with circular footprints of 0.35 m in diameter and are driven by PID controllers for action execution. They can move forward and perform in-place rotations, with translational and rotational velocities constrained to [0, 0.5] m/s and 30°/s, respectively, and acceleration limited to $[-0.4, 0.4]$ m/s^2. We leave generalization across different robot kinodynamics to future work. The grid cell is standardized to 1.0×1.0 m. All experiments are conducted on an Ubuntu 20.04 server with a 64-core AMD 7980X CPU and 256 GB RAM.

3.1 Impact of Optimality

We first study the practical effectiveness of the objective used by MAPF algorithms. One of the most widely used objectives in MAPF research is the *Sum of Costs* (SoC), which measures the total number of timesteps for all robots to reach their goals. While extensive efforts have been made to develop algorithms that optimize SoC optimally or suboptimally, it is unclear, for example, whether reducing SoC by 1% translates to a similar degree of improvement in AET in a realistic simulation. In this section, we assess the correlation between improvements in SoC and AET. We also explore the possibility that there might be better objectives than SoC for evaluating the MAPF plan in realistic settings.

Experiment Setup We generate MAPF plans with varying SoC for a MAPF instance by using MAPF-LNS[1] (Li et al. 2021a), an anytime method that iteratively refines MAPF plans to improve SoC within a predefined runtime limit. For each map, we select three different numbers of robots and five sets of different start and goal locations for robots. For each MAPF instance, we record all plans generated in each iteration of MAPF-LNS within the runtime limit of 60 seconds. We then execute these plans in SMART.

Results and Analysis We first study how well the SoC objective measures the execution time of robots. As shown in fig. 1, SoC and AET show a strong positive linear correlation across all maps and all numbers of robots, indicating that SoC captures the trend of AET. However, the correlation

[1] Code at https://github.com/Jiaoyang-Li/MAPF-LNS

Feature	All Features	SoC + Rotations	SoC + Conflict pairs	Rotations + Conflict pairs	SoC	Type-1 edges	Type-2 edges	Rotations	Conflict pairs	Robots
MAPE	**0.0342**	0.070	0.162	0.164	0.1778	0.2414	0.2826	0.1793	0.3175	0.4317

Table 1: Quadratic regression results. “All Features” uses the full feature set, the next three columns use two-feature combinations, and the remaining columns use single features.

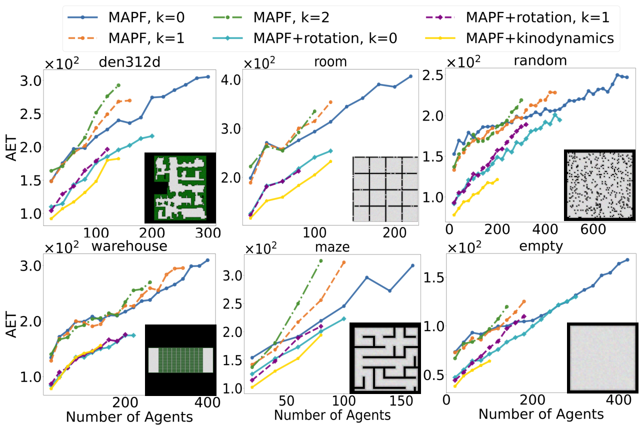

Figure 2: AET using different MAPF models.

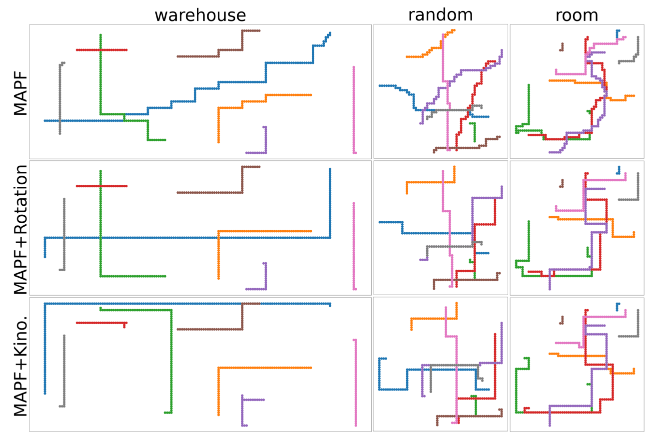

Figure 3: Paths generated on different maps.

is not perfectly monotonic. For the same SoC value, there are noticeable variations in AET, suggesting that SoC alone cannot account for all factors influencing execution time.

To explore other factors that influence execution time, we extract additional features from the ADG, including (1) *Type-1 edges*, the number of type-1 edges, (2) *Type-2 edges*, the number of type-2 edges, (3) *Rotations*, the number of rotation actions, (4) *Conflict pairs*, the number of robot pairs connected by type-2 edges, and (5) *Robots*, the number of robots. We analyze the relationship between these features and AET using a quadratic regression model and measure prediction accuracy with mean absolute percentage error (MAPE). As shown in tab. 1, SoC has the strongest predictive power among individual features, while other features also show correlation with AET. Among the tested feature pairs, *SoC + Rotations* performs best. Notably, the model using all features achieves a much lower MAPE than any single feature or feature combination, indicating that execution time is influenced by multiple factors.

3.2 Impact of Different MAPF Models

The standard MAPF problem uses a simplified model that ignores execution factors such as kinodynamics and potential delays, creating a gap between planning and execution. Thus, recent research has introduced more realistic MAPF models. An ideal MAPF model should capture the essential aspects of robot kinodynamics while omitting unnecessary details. Thus, we conduct an empirical comparison of different MAPF models to identify the most effective one.

Experiment Setup We consider the following MAPF models: (1) the standard MAPF model, (2) the MAPF model with rotation (Zhang et al. 2023), and (3) the MAPF model that accounts for the full kinodynamics of robots, including rotation, speed, and acceleration (Yan and Li 2025). To account for execution delays, we apply the k-robust delay model (Atzmon et al. 2020). This model introduces bounded temporal slack of up to k time steps during planning to tolerate potential execution delays. Combining these factors produces three variants: *MAPF,* $k = i$, *MAPF+rotation,* $k = i$, and *MAPF+kinodynamics*. We use PBS[2] (Ma et al. 2019) to handle conflicts between robots with a runtime limit of 60 seconds. We run these methods to generate MAPF plans for all maps using 25 “random scenarios” from the MAPF benchmark (Stern et al. 2019) with a progressive number of robots until the success rate is lower than 20%. Finally, these plans are executed in SMART.

Results and Analysis As shown in fig. 2, the method incorporating the full kinodynamics of robots achieves the best AET among all the models. Additionally, the planner considering robot rotations demonstrates much better AET than the standard MAPF model. These results indicate that incorporating more accurate MAPF models into planning can improve execution time. However, we also observe tradeoffs associated with these models. Incorporating more accurate MAPF models often reduces scalability. For example, modeling rotations during planning can reduce the maximum number of solvable agents by up to 40%. Meanwhile, the k-robust model performs comparably across different k values with fewer agents; however, its performance declines significantly as the number of robots increases, resulting in lower solution quality and reduced scalability. Interestingly, in the `warehouse` map, MAPF+kinodynamics does not provide significant improvements over MAPF+rotation. To better understand this, we visualize the MAPF plans in fig. 3.

[2]Code at https://github.com/Jiaoyang-Li/PBS

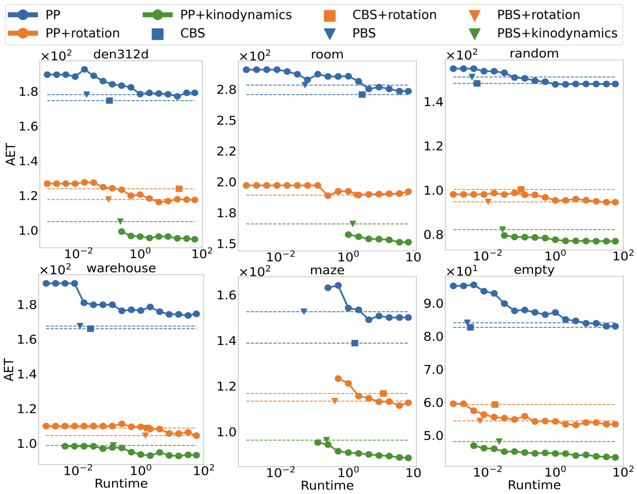

Figure 4: AET using different MAPF models and planners.

The results show that the paths in `warehouse` have fewer shared vertices than in the other two maps. We hypothesize that this minimizes the potential benefits of accounting for kinodynamics. In summary, while more accurate MAPF models generally improve execution time in realistic settings, their increased computational complexity can limit scalability. Additionally, the k-robust model is less efficient and may even harm performance, particularly in more congested scenarios.

3.3 Optimality and MAPF Models

From secs. 3.1 and 3.2, we show that both improving SoC and incorporating a more accurate MAPF model during planning can lead to better AET during execution. However, both of them require longer computation time. Considering the limited planning time available in real-world applications, this section explores the balance between these factors to provide practical insights.

Experiment Setup We generate MAPF plans with varying SoC for three MAPF models: (1) standard MAPF, (2) MAPF with rotation, and (3) MAPF with kinodynamics. To obtain optimal solutions, we use CBS[3] (Li et al. 2021b) for the standard MAPF model and CBS with rotation[4] (Zhang et al. 2023) for the rotation model. For suboptimal plans, we combine PBS with all three models, as described in sec. 3.2. To analyze the relationship between AET and SoC, we create a naive anytime planner by running Prioritized Planning (PP) (Erdmann and Lozano-Perez 1987) with random restarts and recording the MAPF plan with the lowest SoC until reaching a runtime limit of 60 seconds. We do not use existing anytime solvers, such as MAPF-LNS, because, to our best knowledge, no available implementation considers MAPF models with kinodynamic constraints.

Results and Analysis As shown in fig. 4, planners that use more accurate MAPF models consistently achieve better AET, even when less accurate models produce optimal solutions. Concretely, adding rotation reduces AET by 27–33% compared to the standard MAPF model, and adding full kinodynamic constraints provides an additional 17–20% improvement. This indicates that incorporating more accurate MAPF models into planning may be more important than obtaining optimal solutions with less accurate models. Notably, as discussed in sec. 3.1, since SoC is not a perfect metric, the optimal planner does not always achieve the best AET. Meanwhile, execution frameworks may also become a bottleneck. Since ADG adopts a conservative strategy, assuming robots can be delayed indefinitely, it compromises the solution quality of the MAPF plan in certain cases. Overall, achieving better AET depends on both more accurate MAPF models and more effective execution frameworks.

[3] Code at https://github.com/Jiaoyang-Li/CBSH2-RTC
[4] Code at https://github.com/YueZhang-studyuse/MAPF_T

4 Challenges and Future Directions

Our study highlights several challenges that motivate future directions for the deployment of MAPF in realistic settings.

Using Execution-Aware Objectives Although SoC, the most widely used objective, is a good approximation of execution performance, our results show that incorporating additional execution-related features can yield more accurate estimates of execution time. Thus, designing execution-aware objective functions and MAPF planners that optimize them represents an important research opportunity.

Developing Efficient Planners for More Accurate MAPF Models Our results show that using even slightly more accurate MAPF models during planning can significantly improve execution performance. However, this comes at the cost of substantially reduced scalability. Over the past decade, the MAPF community has made remarkable progress in scaling up planners for the standard MAPF model, yet planners that consider kinodynamics still struggle to scale. This underscores the need for better MAPF models and more efficient planners to solve them while maintaining the scalability required for industrial deployment.

Balancing Planner Optimality and Model Accuracy When computation time is limited, our results show that model accuracy often improves execution performance more than aggressive refinement of SoC. This highlights an underexplored trade-off between planner optimality and model accuracy. Moreover, it suggests the need for planners that can adaptively balance these factors based on time budgets.

Enhancing Execution Frameworks Another direction is to develop execution frameworks that operate less conservatively than ADG. Even when planning with accurate robot models, execution errors, like communication delays and controller variability, can accumulate and degrade execution performance. Recent methods, such as Switchable TPG (Berndt et al. 2024; Jiang, Lin, and Li 2025) and Bidirectional TPG (Su, Veerapaneni, and Li 2024), dynamically adjust passing orders between robots to mitigate such execution errors. Adapting these ideas to our execution framework and further advancing this direction could enhance execution performance while maintaining robustness in large-scale industrial deployments.

Acknowledgments

This work is in part supported by the National Science Foundation (NSF) under grant numbers #2328671 and #2441629, as well as a gift from Amazon.

References

Atzmon, D.; Stern, R.; Felner, A.; Wagner, G.; Barták, R.; and Zhou, N.-F. 2020. Robust Multi-Agent Path Finding and Executing. *Journal of Artificial Intelligence Research*, 67: 549–579.

Barer, M.; Sharon, G.; Stern, R.; and Felner, A. 2014. Suboptimal Variants of the Conflict-Based Search Algorithm for the Multi-Agent Pathfinding Problem. In *Proceedings of the International Symposium on Combinatorial Search*, volume 5, 19–27.

Berndt, A.; Van Duijkeren, N.; Palmieri, L.; Kleiner, A.; and Keviczky, T. 2024. Receding Horizon Re-ordering of Multi-Agent Execution Schedules. *IEEE Transactions on Robotics*, 40: 1356–1372.

Cohen, L.; Uras, T.; Kumar, T. K. S.; and Koenig, S. 2019. Optimal and Bounded-Suboptimal Multi-Agent Motion Planning. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, 44–51.

Erdmann, M.; and Lozano-Perez, T. 1987. On Multiple Moving Objects. *Algorithmica*, 2: 477–521.

Heuer, L.; Palmieri, L.; Mannucci, A.; Koenig, S.; and Magnusson, M. 2024. Benchmarking Multi-Robot Coordination in Realistic, Unstructured Human-Shared Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 14541–14547.

Ho, F.; Gonçalves, A.; Rigault, B.; Geraldes, R.; Chicharo, A.; Cavazza, M.; and Prendinger, H. 2022. Multi-Agent Path Finding in Unmanned Aircraft System Traffic Management With Scheduling and Speed Variation. *IEEE Intelligent Transportation Systems Magazine*, 14(5): 8–21.

Hönig, W.; Kiesel, S.; Tinka, A.; Durham, J. W.; and Ayanian, N. 2019. Persistent and Robust Execution of MAPF Schedules in Warehouses. *IEEE Robotics and Automation Letters*, 4(2): 1125–1131.

Huang, T.; Li, J.; Koenig, S.; and Dilkina, B. 2022. Anytime Multi-Agent Path Finding via Machine Learning-Guided Large Neighborhood Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9368–9376.

Jiang, H.; Lin, M.; and Li, J. 2025. Speedup Techniques For Switchable Temporal Plan Graph Optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 23212–23221.

Li, J.; Chen, Z.; Harabor, D.; Stuckey, P. J.; and Koenig, S. 2021a. Anytime Multi-Agent Path Finding via Large Neighborhood Search. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 4127–4135.

Li, J.; Harabor, D.; Stuckey, P. J.; Ma, H.; Gange, G.; and Koenig, S. 2021b. Pairwise Symmetry Reasoning for Multi-Agent Path Finding Search. *Artificial Intelligence*, 301: 103574.

Li, J.; Ruml, W.; and Koenig, S. 2021. EECBS: A Bounded-Suboptimal Search for Multi-Agent Path Finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 12353–12362.

Ma, H.; Harabor, D.; Stuckey, P. J.; Li, J.; and Koenig, S. 2019. Searching with Consistent Prioritization for Multi-Agent Path Finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7643–7650.

Ma, H.; Yang, J.; Cohen, L.; Kumar, T.; and Koenig, S. 2017. Feasibility Study: Moving Non-Homogeneous Teams in Congested Video Game Environments. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 13, 270–272.

Okumura, K.; Machida, M.; Défago, X.; and Tamura, Y. 2022. Priority Inheritance with Backtracking for Iterative Multi-agent Path Finding. *Artificial Intelligence*, 310: 103752.

Schaefer, S.; Palmieri, L.; Heuer, L.; Dillmann, R.; Koenig, S.; and Kleiner, A. 2023. A Benchmark for Multi-Robot Planning in Realistic, Complex and Cluttered Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 9231–9237.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-Based Search For Optimal Multi-Agent Path Finding. *Artificial Intelligence*, 219: 40–66.

Stern, R.; Sturtevant, N. R.; Felner, A.; Koenig, S.; Ma, H.; Walker, T. T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T. K. S.; Boyarski, E.; and Barták, R. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, 151–159.

Su, Y.; Veerapaneni, R.; and Li, J. 2024. Bidirectional Temporal Plan Graph: Enabling Switchable Passing Orders for More Efficient Multi-Agent Path Finding Plan Execution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17559–17566.

Varambally, S.; Li, J.; and Koenig, S. 2022. Which MAPF Model Works Best for Automated Warehousing? In *Proceedings of the International Symposium on Combinatorial Search*, volume 15, 190–198.

Yan, J.; and Li, J. 2024. Multi-Agent Motion Planning With Bézier Curve Optimization Under Kinodynamic Constraints. *IEEE Robotics and Automation Letters*, 9(3): 3021–3028.

Yan, J.; and Li, J. 2025. Multi-Agent Motion Planning For Differential Drive Robots Through Stationary State Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 23360–23368.

Yan, J.; Li, Z.; Kang, W.; Zheng, K.; Zhang, Y.; Chen, Z.; Zhang, Y.; Harabor, D.; Smith, S. F.; and Li, J. 2025. Advancing MAPF Towards the Real World: A Scalable Multi-Agent Realistic Testbed (SMART). *arXiv preprint arXiv:2503.04798*.

Zhang, Y.; Harabor, D.; Le Bodic, P.; and Stuckey, P. J. 2023. Efficient Multi Agent Path Finding with Turn Actions. In *Proceedings of the International Symposium on Combinatorial Search*, volume 16, 119–127.