Current Robotics Reports (2022) 3:77–84
https://doi.org/10.1007/s43154-022-00083-8

Graph-Based Multi-Robot Path Finding and Planning

Hang Ma[1]

Accepted: 18 May 2022 / Published online: 16 June 2022

Abstract

Purpose of Review Planning collision-free paths for multiple robots is important for real-world multi-robot systems and has been studied as an optimization problem on graphs, called multi-agent path finding (MAPF). This review surveys different categories of classic and state-of-the-art MAPF algorithms and different research attempts to tackle the challenges of generalizing MAPF techniques to real-world scenarios.
Recent Findings Solving MAPF problems optimally is computationally challenging. Recent advances have resulted in MAPF algorithms that can compute collision-free paths for hundreds of robots and thousands of navigation tasks in seconds of runtime. Many variants of MAPF have been formalized to adapt MAPF techniques to different real-world requirements, such as considerations of robot kinematics, online optimization for real-time systems, and the integration of task assignment and path planning.
Summary Algorithmic techniques for MAPF problems have addressed important aspects of several multi-robot applications, including automated warehouse fulfillment and sortation, automated train scheduling, and navigation of non-holonomic robots and quadcopters. This showcases their potential for real-world applications of large-scale multi-robot systems.

Keywords Multi-agent path finding · Multi-robot path planning · Multi-robot systems

Introduction

In many real-world multi-robot systems, robots have to plan collision-free paths to different locations to execute different tasks. Today, thousands of warehouse robots already navigate fully autonomously to relocate inventory pods in automated fulfillment centers [1, 2] or deliver parcels in sortation centers [3]. In the coming years, autonomous aircraft-towing vehicles will tow aircraft from the runways to the terminal gates (and vice versa) at airports. Other examples include autonomous intersection management [4], forklift robot fleets [5, 6], game characters in video games [7], object-transportation robots [8], patrolling robots [9], service robots [10, 11], swarms of differential-drive robots and quadcopters [12–14], robots in formations [15], and other multi-robot systems [16].

Solving the path planning problem optimally for multiple robots is computationally challenging, especially for a large number of robots. However, the above real-world applications require computing high-quality collision-free paths for a large number of robots in a short computation time since shorter paths result in higher throughput or lower operating costs (since fewer robots are required to achieve the same throughput) of the systems.

Multi-Agent Path Finding

Many recent works in the artificial intelligence, robotics, and operations research communities have modeled the path planning problem for multiple robots as a combinatorial optimization problem on graphs, called multi-agent path finding (MAPF) [17, 18••]. MAPF has also been studied under the name of multi-robot path planning on graphs [19]. A MAPF problem instance consists of a connected undirected graph and a set of robots. The vertices of the given graph correspond to locations and the edges correspond to connections between locations that the robots can move along. Each robot occupies one vertex at each discrete time

This article belongs to the Topical Collection on *Group Robotics*

✉ Hang Ma
hangma@sfu.ca

[1] School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby V5A 1B5, BC, Canada

step and is given a start vertex and a goal vertex. Between two consecutive time steps, each robot takes an action to either move to an adjacent vertex or wait at its current vertex. Two robots collide if they move to the same vertex or traverse the same edge in opposite directions at the same time. The problem of MAPF is to find collision-free paths for the robots from their start vertices to their goal vertices. The objective is to minimize either the makespan, defined as the maximum of the arrival times of all robots at their goal vertices, or the flowtime, defined as the sum of the arrival times of all robots at their goal vertices.

Finding a solution to any MAPF problem instance or deciding its unsolvability can be done in polynomial time [20]. However, it is NP-hard (namely, unlikely that a polynomial-time algorithm exists) to find a solution with the minimum makespan [21] or the minimum flowtime [22•] to a MAPF problem instance, even if the given graph is a planar graph [23] or a 2D 4-neighbor grid [24]. In addition, it is NP-hard to compute an approximate solution within any constant factor less than 4/3 to a MAPF problem instance [25].

On one hand, recent advances in MAPF solving have resulted in powerful MAPF algorithms that can compute collision-free paths for a large number of robots in a short runtime, despite the complexity of solving MAPF optimally. These advances have resulted in a number of achievements, including a MAPF software [26] that recently won the Flatland Challenge [27], a train-scheduling competition at NeurIPS 2020 (one of the top machine learning conferences). The MAPF solver has been demonstrated to be capable of computing high-quality paths (namely with small makespan or flowtime) for up to 3000 robots in minutes of runtime on a simulator. On the other hand, there are key challenges [28] that must be addressed in order to apply MAPF algorithms to real-world applications of multi-robot systems, which requires techniques beyond MAPF solving. The following sections of this review survey latest advances that enhance MAPF solving and extensions to MAPF solving that tackle the research challenges in generalizing it to real-world scenarios.

MAPF Algorithms

Recent MAPF algorithms can be categorized into reduction-based, rule-based, and search-based algorithms. In the following, we survey their methodologies and highlight their properties in terms of completeness (complete for all MAPF problem instances, complete for MAPF problem instances on graphs with special properties, or incomplete) and optimality (optimal, bounded-suboptimal, or suboptimal with respect to different objectives). A MAPF algorithm is complete for a class of MAPF problem instances if it guarantees to return a solution for any solvable MAPF problem instance in the class or correctly decide that the given MAPF problem instance in the class is unsolvable in finite time.

Reduction-Based MAPF Algorithms Reduction-based MAPF algorithms reduce MAPF to other well-studied combinatorial problems, such as Boolean Satisfiability [29], Integer Linear Programming [30], and Answer Set Programming [31, 32]. They are complete for all MAPF problem instances. They solve MAPF with the makespan objective optimally but can be modified to solve MAPF with other objectives optimally [30, 32, 33], bounded suboptimally (with the guarantee that the resulting solution is within a user-provided suboptimality factor from the optimal solution) [34], and suboptimally [35–37]. They perform well for MAPF problem instances on small-sized graphs with densely placed robots. For example, a state-of-the-art Integer Linear Programming–based MAPF solver can compute a solution with the minimum makespan for a MAPF problem instance on a 24 × 18 2D 4-neighbor grid with 60 robots in less than 15 s of runtime [37].

Rule-Based MAPF algorithms Rule-based algorithms solve MAPF using a set of primitive operations that specify the actions of the robots in different situations. They often guarantee completeness for only a restricted class of MAPF problem instances. Rule-based algorithms are often very efficient by simply following the predefined primitive operations but provide no guarantee on the solution quality (optimality). Push and Swap [38] and its extension [39] can compute a solution for 100 agents in less than 10 s of runtime but provide no completeness guarantee theoretically. One of their descendants, Push and Rotate [40], is complete for MAPF problem instances on graphs with at least two vertices that are unoccupied by robots. TASS [41] is complete for MAPF problem instances on "solvable" trees based on prior work on solving multi-robot motion planning on trees [42]. BIBOX [43] is complete for MAPF problem instances on bi-connected graphs with at least two vertices unoccupied by robots. Its descendant [44] works for strongly bi-connected directed graphs with at least two vertices unoccupied by robots. SAG [45] is complete for MAPF problem instances on grid-like "well-connected" graphs, runs in polynomial time, and provides a constant-factor approximation guarantee for minimizing the makespan on such graphs.

Search-Based MAPF Algorithms Search-based MAPF algorithms [46] solve MAPF with heuristic search techniques. The main computational challenge of optimally solving MAPF with a search algorithm is that the number of possible states of a MAPF problem instance grows exponentially in the number of robots.

- A*-based MAPF algorithms [47–49] plan paths with joint states but try to reduce the size of the state space they need to explore. They are complete for all MAPF problem instances and can be used for either makespan minimization or flowtime minimization.
- Decoupled MAPF algorithms [50–52] plan paths for robots one at a time according to a predefined or a dynamic total ordering on the robots. Path planning for each robot uses an A* search in vertex and time dimensions that treats already planned paths of other robots as moving obstacles. Decoupled MAPF algorithms are often efficient but provide no optimality or even completeness guarantee. PIBT [53] develops a scheme to decide a partial ordering on the robots dynamically but only guarantees that all robots reach their goal vertices at least once (but possibly not at the same time) in finite time on biconnected graphs. MAPF-LNS [54] is another recent decoupled MAPF algorithm that uses Large Neighborhood Search [55], a local search algorithm, to improve a suboptimal MAPF solution by repeatedly replanning paths for a subset of robots. It is one of the core elements of the MAPF software [26] that won the Flatland Challenge. Its descendant MAPF-LNS2 [56] uses Large Neighborhood Search to improve a MAPF plan (paths of all robots) with collisions by repeatedly replanning paths for a subset of robots to reduce the number of collisions, until a collision-free MAPF plan (a MAPF solution) is obtained.
- Hierarchical MAPF algorithms plan paths for robots individually on the low level and dynamically couple the resulting single-robot paths with a tree search on the high level. They are complete for all MAPF problem instances. Increasing Cost Tree Search [57] minimizes the flowtime. It performs a best-first tree search of all combinations of the arrival times of robots on the high level and checks whether collision-free paths exist for a combination of arrival times on the low level. Conflict-Based Search (CBS) [58••] is arguably the most popular optimal MAPF algorithm. It minimizes either the makespan or the flowtime. CBS first finds individually time-optimal paths for all robots (ignoring collisions). On the high level, it then performs a best-first search on a binary constraint tree. Each branching resolves one collision in the computed paths by imposing constraints on individual robots that forbid them from occupying a vertex or traversing an edge at a given time step. On the low level, CBS uses an A* search in vertex and time dimensions to replan for a robot that obeys the constraints. Many improvements to CBS have been proposed in recent years: Meta-Agent CBS [58••] dynamically groups multiple robots into a meta-agent on the high level and uses an A* search to plan paths for these robots with their joint states on the low level. ICBS [59] always first resolves collisions that result in child search nodes whose costs are larger than that of the current node, thus affording the high-level search of CBS pruning opportunities. CBSH [60] and its improvement [61] use an admissible heuristic to improve the high-level best-first search of CBS. Disjoint-Splitting CBS [62] expands each node in a way such that any solution is admitted by the subtree under only one but not both of its child nodes, thus reducing duplicate search effort of the high-level search of CBS. IDCBS [63] replaces the high-level best-first search of CBS with iterative-deepening depth-first searches. Symmetry-Breaking CBS [64–66•] and Mutex-Propagation CBS [67] add multiple constraints to a child node at a time to break symmetry in the high-level search of CBS. The best Symmetry-Breaking CBS variant has empirically been shown to compute optimal solutions for MAPF problem instances on a 256 × 257 2D 4-neighbor grid with 100 robots in seconds of runtime [66•]. ECBS [68] and its improvements [69, 70] perform a bounded-suboptimal search on the constraint tree, making CBS bounded-suboptimal. Recent research [71] has also developed an anytime version of the bounded-suboptimal search on the constraint tree for CBS. Another line of recent research also uses machine learning to learn a good branching policy for the high-level search of CBS for both the optimal [72] and bounded-suboptimal [73] settings.
- Hybrid MAPF algorithms combine several of the above search-based MAPF techniques or combine search-based MAPF techniques with reduction-based or rule-based MAPF techniques. SMT-CBS [74] replicates the high-level search of CBS with Satisfiability Modulo Theories, which is then solved by a Boolean Satisfiability solver, and minimizes either the makespan or the flowtime. Lazy CBS [75] replaces the high-level search of CBS with a Constraint Programming solver and minimizes the flowtime. BCP [76, 77] combines Branch-and-Cut-and-Price techniques for Mixed Integer Programming with symmetry-breaking techniques for MAPF and minimizes the flowtime. Priority-Based Search [78•] is a recent hierarchically decoupled that performs a depth-first search on a binary priority tree to explore all possible orderings on the robots. It is complete for only "well-formed" MAPF problem instances and has empirically been shown to compute close-to-optimal solutions for MAPF problem instances on a 481 × 530 2D 4-neighbor grid with 600 robots in half a minute of runtime. Some algorithms combine both primitive operations (rule-based MAPF techniques) and search. MAPP [79] explore different ways of combining paths of individual robots and is complete for MAPF problem instances on "slidable" graphs [79]. There is also a MAPF algorithm that uses a combination of A* searches on a graph abstraction, primitive operations, and reductions to Constraint Satisfaction Problems [80].

Recent research [81, 82] has used machine learning to select a MAPF algorithm among multiple candidate MAPF algorithms for a given MAPF problem instance.

MAPF Extensions and Related Problems

Recent studies have also generalized the standard definition of MAPF to different real-world scenarios.

MAPF with Deadlines MAPF with Deadlines [83, 84] aims to maximize the number of robots that reach their goal vertices within a given deadline. Its applications include robots that need to evacuate before a disaster and robots that need to finish tasks before a deadline.

MAPF with Delay Probabilities and Robust MAPF MAPF with Delay Probabilities (MAPF-DP) [85] generalizes MAPF to the case where the uncertainty of robot motion has to be considered during planning to ensure a collision-free execution of the plan. In MAPF-DP, the uncertainty of each robot is characterized by a given delay probability with which the robot stays in its current vertex whenever it intends to traverse an outgoing edge of its current vertex. The problem of MAPF-DP is to find a plan that consists of a path for each robot and a plan-execution policy that controls with GO or STOP commands how each robot proceeds along its path such that no collisions occur during plan execution. MAPF-DP has also been studied under the name MAPF with Uncertainty [86], where the paths are planned in the belief space of the robots and the execution of the resulting plan is not guaranteed to be collision-free. K-Robust MAPF [87] extends CBS to enforce K time steps for which a vertex must be unoccupied after it has been occupied by a robot during planning, which reduces the possibility of collisions during plan execution without using plan-execution policies. Recent research [88] has generalized Symmetry-Breaking CBS [66•] to the K-Robust MAPF setting. Probabilistic Robust MAPF [89] bounds the probability that any collision occurs during plan execution.

MAPF with Continuous Time or Kinematic Constraints MAPF with Continuous Time [90] extends CBS to planning paths on weighted graphs where the edge weights characterize the nonuniform traversal times of the edges. Other research [91, 92] has also studied MAPF on weighted graphs. MAPF for Large Agents [93] allows a robot to occupy more than one vertex at one time step according to its given shape and volume. Two robots collide if both of them occupy some vertex at the same time step. The resulting CBS-based algorithm has been applied to planning collision-free trajectories for quadcopters that take into account their ellipsoid shapes and downwash effects. MAPF-POST [94•, 95] is a polynomial-time algorithm that post-processes a MAPF solution to create a plan-execution schedule that works on non-holonomic robots, takes their kinematic constraints, such as the maximum and minimum translational and rotational velocity limits, into account, and provides a guaranteed safety distance between them, which avoids time-intensive replanning in many cases.

Reinforcement Learning for Distributed MAPF PRIMAL [96•] and its descendant [97] model MAPF as a multi-agent reinforcement learning task, where all robots follow the same learned single-agent policy to decide their actions at each time step based on their local observations. Recent research [98–100] uses a graph neural network to allow robots to communicate and also precomputed shortest path distances [99] or an online shortest path computation [97, 101] to assist training. We note that classic optimization-based collision-avoidance approaches [102–104] use a similar distributed MAPF setting but can be applied to robots moving in continuous space.

MAPF with Target Assignment Anonymous MAPF, also known as Permutation-Invariant MAPF or Unlabeled MAPF, does not assume predefined goal vertices for the robots and aims to find a one-to-one mapping from the given goal vertices to the robots and collision-free paths for the robots to their assigned goal vertices. Minimizing the makespan for Anonymous MAPF is polynomial-time solvable using a max-flow algorithm [105]. CBS-TA [106] searches all possible assignments of goal vertices to robots and minimizes the flowtime for Anonymous MAPF. Combined Target Assignment and Path Finding (TAPF) [107] partitions the robots into teams where the problem of each team is an Anonymous MAPF problem. CBM [107] combines the high-level search of CBS and a low-level min-cost max-flow algorithm to minimize the makespan for TAPF. Recent research [108, 109] has studied generalized TAPF problems where each robot needs to get assigned multiple goal vertices. MG-MAPF [110] assumes that each robot is preassigned multiple unordered goal vertices and aims to compute collision-free paths for the robots to visit all goal vertices. MG-TAPF [111] aims to find a one-to-one mapping from the given tasks that each consists of a sequence of ordered goal vertices and collision-free paths for the robots that visit the goal vertices of their assigned tasks in the specified order.

Online MAPF and Multi-Agent Pickup and Delivery Online MAPF [112, 113] assumes that each robot is assigned a new goal vertex by a black box once it reaches its current goal vertex. Recent research has conducted a theoretical study [113] on the competitiveness of online MAPF algorithms (namely the performance gap between online and optimal offline MAPF algorithms). RHCR [114] generalizes (offline) MAPF algorithms to online MAPF by repeatedly replanning paths for the robots. One version of RHCR that uses Priority-Based Search [78•] has been shown to compute paths for 1000 robots on a 37 × 77 4-neighbor grid in less than half a minute of runtime. Multi-Agent Pickup and Delivery (MAPD) [115•] is a combined multi-robot task-allocation and path-planning problem. MAPD has first been studied in an online setting where robots have to constantly get assigned a stream of incoming tasks that are added to the system at unknown release times and plan collision-free paths to the pickup and delivery vertices of the tasks. Online MAPD algorithms [115•] repeatedly apply task-assignment and MAPF algorithms to (re-)assign tasks to and (re-)plan paths for robots whenever a new task arrives or a robot becomes available for executing tasks. Recent research [116] has developed an online MAPD algorithm that considers kinematic constraints of robots directly during planning and shown experimentally that the algorithm can compute solutions for MAPD problem instances with 250 robots and 2000 tasks within a total runtime of 10 s. Offline MAPD [117] considers tasks that are known a priori. Recent research [3] has also considered an online TAPF/MAPD variant that aims to minimize the idle time of sorting stations—that is, when there are no warehouse robots servicing the sorting stations—in an automated sortation center. The resulting algorithm has been shown to compute solutions for 350 robots within 2 s of runtime on an industrial robot simulator.

Conclusions

Planning collision-free paths for multiple robots is a fundamental building block for many real-world applications of multi-robot systems. It has been studied as a graph-optimization problem under the name of MAPF by researchers from the artificial intelligence, robotics, and operations research communities. As the efficiency of MAPF algorithms improves, they will become increasingly viable for real-time path-planning operations of multi-robot systems. Current research has also addressed several challenges to adapt MAPF techniques to the requirements of real-world multi-robot systems. Future research directions include developing deeper theoretical understandings for MAPF variants such as Distributed MAPF and MAPD and combining MAPF techniques with considerations of complex real-world settings such as general temporal constraints and dependencies of tasks and high-order dynamic constraints of robots. Readers are referred to the MAPF information page [118] for a listing of MAPF researchers and links to their publications and software, tutorials, a mailing list, and other resources.

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant number RGPIN2020-06540.

Declarations

Conflict of interest The author declares no competing interests.

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

Papers of particular interest, published recently, have been highlighted as:
• Of importance
•• Of major importance

1. Wurman PR, D'Andrea R, Mountz M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Magazine. 2008;29(1):9–20.
2. Hönig W, Kiesel S, Tinka A, Durham JW, Ayanian N. Persistent and robust execution of MAPF schedules in warehouses. IEEE Robotics and Automation Letters. 2019;4(2):1125–31.
3. Kou NM, Peng C, Ma H, Kumar TKS, Koenig S. Idle time optimization for target assignment and path finding in sortation centers. In: AAAI conference on artificial intelligence; 2020. p. 9925–9932.
4. Dresner K, Stone P. A multiagent approach to autonomous intersection management. Journal of Artificial Intelligence Research. 2008;31:591–656.
5. Pecora F, Andreasson H, Mansouri M, Petkov V. A loosely-coupled approach for multi-robot coordination, motion planning and control. In: International conference on automated planning and scheduling; 2018. p. 485–493.
6. Salvado J, Krug R, Mansouri M, Pecora F. Motion planning and goal assignment for robot fleets using trajectory optimization. In: IEEE/RSJ international conference on intelligent robots and systems; 2018. p. 7939–7946.
7. Ma H, Yang J, Cohen L, Kumar TKS, Koenig S. Feasibility study: Moving non-homogeneous teams in congested video game environments. In: AAAI conference on artificial intelligence and interactive digital entertainment; 2017. p. 270–272.
8. Rus D, Donald B, Jennings J. Moving furniture with teams of autonomous robots. In: IEEE/RSJ international conference on intelligent robots and systems; 1995. p. 235–242.
9. Agmon N, Urieli D, Stone P. Multiagent patrol generalized to complex environmental conditions. In: AAAI conference on artificial intelligence; 2011. p. 1090–1095.
10. Ahmadi M, Stone P. A multi-robot system for continuous area sweeping tasks. In: IEEE international conference on robotics and automation; 2006. p. 1724–1729.
11. Khandelwal P, Zhang S, Sinapov J, Leonetti M, Thomason J, Yang F, et al. Bwibots: A platform for bridging the gap between AI and human-robot interaction research. International Journal of Robotics Research. 2017;36(5–7):635–59.
12. Hönig W, Kumar TKS, Ma H, Ayanian N, Koenig S. Formation change for robot groups in occluded environments. In: IEEE/RSJ international conference on intelligent robots and system; 2016. p. 4836–4842.

13. Preiss JA, Hönig W, Ayanian N, Sukhatme GS. Downwash-aware trajectory planning for large quadrotor teams. In: IEEE/RSJ international conference on intelligent robots and systems; 2017. p. 250–257.

14. Hönig W, Preiss JA, Kumar TS, Sukhatme GS, Ayanian N. Trajectory planning for quadrotor swarms. IEEE Transactions on Robotics. 2018;34(4):856–69.

15. Li J, Sun K, Ma H, Felner A, Kumar TKS, Koenig S. Moving agents in formation in congested environments. In: International conference on autonomous agents and multiagent systems; 2020. p. 726–734.

16. Ma H, Hönig W, Cohen L, Uras T, Xu H, Kumar TKS, et al. Overview: A hierarchical framework for plan generation and execution in multi-robot systems. IEEE Intelligent Systems. 2017;32(6):6–12.

17. Ma H, Koenig S. AI Buzzwords explained: Multi-agent path finding (MAPF). AI Matters. 2017;3(3):15–9.

18.•• Stern R, Sturtevant NR, Felner A, Koenig S, Ma H, Walker TT, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. In: International symposium on combinatorial search; 2019. p. 151–158. **This paper presents formalizations of various MAPF problems, defines a unifying terminology for MAPF research, and establishes common benchmarks and evaluation measures for evaluating MAPF algorithms. It is authored by some of the researchers working actively today in the field of MAPF.**

19. Yu J, LaValle SM. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. IEEE Transactions on Robotics. 2016;32(5):1163–77.

20. Yu J, Rus D. Pebble motion on graphs with rotations: Efficient feasibility tests and planning algorithms. In: Algorithmic Foundations of Robotics XI, Springer Tracts in Advanced Robotics. vol. 107. Springer; 2015. p. 729–746.

21. Surynek P. An optimization variant of multi-robot path planning is intractable. In: AAAI conference on artificial intelligence; 2010. p. 1261–1263.

22.• Yu J, LaValle SM. Structure and intractability of optimal multi-robot path planning on graphs. In: AAAI conference on artificial intelligence; 2013. p. 1444–1449. **This paper presents one of the first NP-hardness results on optimally solving MAPF.**

23. Yu J. Intractability of optimal multi-robot path planning on planar graphs. IEEE Robotics and Automation Letters. 2016;1(1):33–40.

24. Banfi J, Basilico N, Amigoni F. Intractability of time-optimal multirobot path planning on 2D grid graphs with holes. IEEE Robotics and Automation Letters. 2017;2(4):1941–7.

25. Ma H, Tovey C, Sharon G, Kumar TKS, Koenig S. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In: AAAI conference on artificial intelligence; 2016. p. 3166–3173.

26. Li J, Chen Z, Zheng Y, Chan SH, Harabor D, Stuckey PJ, et al. Scalable rail planning and replanning: winning the 2020 flatland challenge. In: International conference on automated planning and scheduling; 2021. p. 477–485.

27. Laurent F, Schneider M, Scheller C, Watson J, Li J, Chen Z, et al. Flatland competition 2020: MAPF and MARL for efficient train coordination on a grid world. In: NeurIPS 2020 Competition and Demonstration Track. vol. 133 of Proceedings of Machine Learning Research; 2021. p. 275–301.

28. Ma H, Koenig S, Ayanian N, Cohen L, Hönig W, Kumar TKS, et al. Overview: generalizations of multi-agent path finding to real-world scenarios. In: IJCAI-16 workshop on multi-agent path finding; 2016.

29. Surynek P. Towards optimal cooperative path planning in hard setups through satisfiability solving. In: Pacific Rim international conference on artificial intelligence; 2012. p. 564–576.

30. Yu J, LaValle SM. Optimal multi-robot path planning on graphs: Complete algorithms and effective heuristics. IEEE Transactions on Robotics. 2016;32(5):1163–77.

31. Erdem E, Kisa DG, Oztok U, Schueller P. A general formal framework for pathfinding problems with multiple agents. In: AAAI conference on artificial intelligence; 2013. p. 290–296.

32. Gómez RN, Hernández C, Baier JA. Solving sum-of-costs multi-agent pathfinding with answer-set programming. In: AAAI conference on artificial intelligence; p. 9867–9874.

33. Surynek P, Felner A, Stern R, Boyarski E. Efficient SAT approach to multi-agent path finding under the sum of costs objective. In: European conference on artificial intelligence; 2016. p. 810–818.

34. Surynek P, Felner A, Stern R, Boyarski E. Modifying optimal SAT-based approach to multi-agent path-finding problem to sub-optimal variants. In: International symposium on combinatorial search; 2017. p. 169–170.

35. Surynek P. Reduced time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally. In: International joint conference on artificial intelligence; 2015. p. 1916–1922.

36. Wang J, Li J, Ma H, Koenig S, Kumar TKS. A new constraint satisfaction perspective on multi-agent path finding. In: International conference on autonomous agents and multiagent systems; 2019. p. 417–423.

37. Han SD, Yu J. Integer programming as a general solution methodology for path-based optimization in robotics: Principles, best practices, and applications. In: IEEE/RSJ international conference on intelligent robots and systems; 2019. p. 1890–1897.

38. Luna R, Bekris KE. Push and Swap: Fast cooperative pathfinding with completeness guarantees. In: International joint conference on artificial intelligence; 2011. p. 294–300.

39. Sajid Q, Luna R, Bekris KE. Multi-agent pathfinding with simultaneous execution of single-agent primitives. In: International Symposium on Combinatorial Search; 2012.

40. de Wilde B, ter Mors AW, Witteveen C. Push and rotate: Cooperative multi-agent path planning. In: International conference on autonomous agents and multiagent systems; 2013. p. 87–94.

41. Khorshid M, Holte R, Sturtevant NR. A polynomial-time algorithm for non-optimal multi-agent pathfinding. In: International symposium on combinatorial search; 2011. p. 76–83.

42. Masehian E, Nejad AH. Solvability of multi robot motion planning problems on trees. In: IEEE/RSJ international conference on intelligent robots and systems; 2009. p. 5936–5941.

43. Surynek P. A novel approach to path planning for multiple robots in bi-connected graphs. In: IEEE international conference on robotics and automation; 2009. p. 3613–3619.

44. Botea A, Bonusi D, Surynek P. Solving multi-agent path finding on strongly biconnected digraphs. Journal of Artificial Intelligence Research. 2018;62:273–314.

45. Yu J. Average case constant factor time and distance optimal multi-robot path planning in well-connected environments. Autonomous Robots. 2020;44(3):469–83.

46. Felner A, Stern R, Shimony SE, Boyarski E, Goldenberg M, Sharon G, et al. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In: International symposium on combinatorial search; 2017. p. 29–37.

47. Standley TS. Finding optimal solutions to cooperative pathfinding problems. In: AAAI conference on artificial intelligence; 2010. p. 173–178.

48. Goldenberg M, Felner A, Stern R, Sharon G, Sturtevant NR, Holte RC, et al. Enhanced partial expansion A*. Journal of Artificial Intelligence Research. 2014;50:141–87.

49. Wagner G, Choset H. Subdimensional expansion for multirobot path planning. Artificial Intelligence. 2015;219:1–24.

50. Silver D. Cooperative pathfinding. In: AAAI conference on artificial intelligence and interactive digital entertainment; 2005. p. 117–122.

51. Sturtevant NR, Buro M. Improving collaborative pathfinding using map abstraction. In: AAAI conference on artificial intelligence and interactive digital entertainment; 2006. p. 80–85.

52. Bnaya Z, Felner A. Conflict-oriented windowed hierarchical cooperative A*. In: IEEE international conference on robotics and automation; 2014. p. 3743–3748.

53. Okumura K, Machida M, Défago X, Tamura Y. Priority inheritance with backtracking for iterative multi-agent path finding. In: International joint conference on artificial intelligence; 2019. p. 535–542.

54. Li J, Chen Z, Harabor D, Stuckey P, Koenig S. Anytime multi-agent path finding via large neighborhood search. In: International joint conference on artificial intelligence; 2021. p. 4127–4135.

55. Shaw P. Using constraint programming and local search methods to solve vehicle routing problems. In: International conference on principles and practice of constraint programming; 1998. p. 417–431.

56. Li J, Chen Z, Harabor D, Stuckey PJ, Koenig S. MAPF-LNS2: Repairing multi-agent path finding via large neighborhood search. In: AAAI conference on artificial intelligence; 2022. p. in press.

57. Sharon G, Stern R, Goldenberg M, Felner A. The increasing cost tree search for optimal multi-agent pathfinding. Artificial Intelligence. 2013;195:470–95.

58.•• Sharon G, Stern R, Felner A, Sturtevant NR. Conflict-based search for optimal multi-agent pathfinding. Artif Intell. 2015;219:40–66. **This article presents the vanilla version of Conflict-Based Search, argubly the most popular optimal MAPF algorithm.**

59. Boyarski E, Felner A, Stern R, Sharon G, Tolpin D, Betzalel O, et al. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In: International joint conference on artificial intelligence; 2015. p. 740–746.

60. Felner A, Li J, Boyarski E, Ma H, Cohen L, Kumar TKS, et al. Adding heuristics to conflict-based search for multi-agent pathfinding. In: International conference on automated planning and scheduling; 2018. p. 83–87.

61. Li J, Boyarski E, Felner A, Ma H, Koenig S. Improved heuristics for multi-agent path finding with conflict-based search. In: International joint conference on artificial intelligence; 2019. p. 442–449. **This paper presents admissible heuristics for the high-level search of CBS.**

62. Li J, Harabor D, Stuckey PJ, Felner A, Ma H, Koenig S. Disjoint splitting for conflict-based search for multi-agent path finding. In: International conference on automated planning and scheduling; 2019. p. 279–283.

63. Boyarski E, Felner A, Harabor D, Stuckey PJ, Cohen L, Li J, et al. Iterative-deepening conflict-based search. In: International joint conferences on artificial intelligence; 2021. p. 4084–4090.

64. Li J, Harabor D, Stuckey PJ, Ma H, Koenig S. Symmetry-breaking constraints for grid-based multi-agent path finding. In: AAAI conference on artificial intelligence; 2019. p. 6087–6095.

65. Li J, Gange G, Harabor D, Stuckey PJ, Ma H, Koenig S. New techniques for pairwise symmetry breaking in multi-agent path finding. In: International conference on automated planning and scheduling; 2020. p. 193–201.

66.• Li J, Harabor D, Stuckey PJ, Ma H, Gange G, Koenig S. Pairwise symmetry reasoning for multi-agent path finding search. Artif Intell. 2021;301:103574. **This article presents a version of CBS that combines several state-of-the-art techniques for solving MAPF optimally.**

67. Zhang H, Li J, Surynek P, Koenig S, Kumar TKS. Multi-agent path finding with mutex propagation. In: International conference on automated planning and scheduling; 2020. p. 323–332.

68. Barer M, Sharon G, Stern R, Felner A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In: International symposium on combinatorial search; 2014. p. 19–27.

69. Cohen L, Uras T, Kumar TKS, Xu H, Ayanian N, Koenig S. Improved solvers for bounded-suboptimal multi-agent path finding. In: International joint conference on artificial intelligence; 2016. p. 3067–3074.

70. Li J, Ruml W, Koenig S. EECBS: Bounded-suboptimal search for multi-agent path finding. In: AAAI Conference on Artificial Intelligence; 2021. p. 12353–12362.

71. Cohen L, Greco M, Ma H, Hernandez C, Felner A, Kumar TKS, et al. Anytime focal search with applications. In: International joint conference on artificial intelligence; 2018. p. 1434–1441.

72. Huang T, Koenig S, Dilkina B. Learning to resolve conflicts for multi-agent path finding with conflict-based search. In: AAAI conference on artificial intelligence; 2021. p. 11246–11253.

73. Huang T, Dilkina B, Koenig S. Learning node-selection strategies in bounded-suboptimal conflict-based search for multi-agent path finding. In: international conference on autonomous agents and multiagent systems; 2021. p. 611–619.

74. Surynek P. Unifying search-based and compilation-based approaches to multi-agent path finding through satisfiability modulo theories. In: International joint conference on artificial intelligence; 2019. p. 1177–1183.

75. Gange G, Harabor D, Stuckey PJ. Lazy CBS: Implicit conflict-based search using lazy clause generation. In: Proceedings of the international conference on automated planning and scheduling. vol. 29; 2019. p. 155–162.

76. Lam E, Le Bodic P, Harabor D, Stuckey PJ. Branch-and-cut-and-price for multi-agent pathfinding. In: International joint conference on artificial intelligence; 2019. p. 1289–1296.

77. Lam E, Le Bodic P. New valid inequalities in branch-and-cut-and-price for multi-agent path finding. In: International conference on automated planning and scheduling. vol. 30; 2020. p. 184–192.

78.• Ma H, Harabor D, Stuckey PJ, Li J, Koenig S. Searching with consistent prioritization for multi-agent path finding. In: AAAI conference on artificial intelligence; 2019. p. 7643–7650. **This paper presents several theoretical results on prioritized planning and a search-based prioritized-planning scheme for MAPF and other multi-robot path planning settings.**

79. Wang K, Botea A. MAPP : A scalable multi-agent path planning algorithm with tractability and completeness guarantees. Journal of Artificial Intelligence Research. 2011;42:55–90.

80. Ryan M. Constraint-based multi-robot path planning. In: IEEE international conference on robotics and automation; 2010. p. 922–928.

81. Kaduri O, Boyarski E, Stern R. Algorithm selection for optimal multi-agent pathfinding. In: International conference on automated planning and scheduling; 2020. p. 161–165.

82. Ren J, Sathiyanarayanan V, Ewing E, Senbaslar B, Ayanian N. MAPFAST: A deep algorithm selector for multi agent path finding using shortest path embeddings. In: International conference on autonomous agents and multiagent systems; 2021. p. 1055–1063.

83. Ma H, Wagner G, Felner A, Li J, Kumar TKS, Koenig S. Multi-agent path finding with deadlines: Preliminary results. In: International conference on autonomous agents and multiagent systems; 2018. p. 2004–2006.

84. Ma H, Wagner G, Felner A, Li J, Kumar TKS, Koenig S. Multi-agent path finding with deadlines. In: International joint conference on artificial intelligence; 2018. p. 417–423.

85. Ma H, Kumar TKS, Koenig S. Multi-agent path finding with delay probabilities. In: AAAI conference on artificial intelligence; 2017. p. 3605–3612.

86. Wagner G, Choset H. Path planning for multiple agents under uncertainty. In: International conference on automated planning and scheduling; 2017. p. 577–585.

87. Atzmon D, Stern R, Felner A, Wagner G, Barták R, Zhou N. Robust multi-agent path finding. In: International symposium on combinatorial search; 2018. p. 2–9.

88. Chen Z, Harabor D, Li J, Stuckey PJ. Symmetry breaking for k-robust multi-agent path finding. In: AAAI conference on artificial intelligence (AAAI); 2021. p. 12267–12274.

89. Atzmon D, Stern R, Felner A, Sturtevant NR, Koenig S. Probabilistic robust multi-agent path finding. In: International conference on automated planning and scheduling; 2020. p. 29–37.

90. Andreychuk A, Yakovlev K, Surynek P, Atzmon D, Stern R. Multi-agent pathfinding with continuous time. Artificial Intelligence. 2022; p. 103662.

91. Barták R, Švancara Jí, Vlk M. A scheduling-based approach to multi-agent path finding with weighted and capacitated arcs. In: International conference on autonomous agents and multiagent systems; 2018. p. 748–756.

92. Walker TT, Sturtevant NR, Felner A. Extended increasing cost tree search for non-unit cost domains. In: International joint conference on artificial intelligence; 2018. p. 534–540.

93. Li J, Surynek P, Felner A, Ma H, Kumar TKS, Koenig S. Multi-agent path finding for large agents. In: AAAI conference on artificial intelligence; 2019. p. 7627–7634.

94.• Hönig W, Kumar TKS, Cohen L, Ma H, Xu H, Ayanian N, et al. Multi-agent path finding with kinematic constraints. In: International conference on automated planning and scheduling; 2016. p. 477–485. **This paper presents an efficient method to transform a MAPF solution to a kinematically-feasible plan-execution schedule for non-holonomic robots.**

95. Hönig W, Kumar TKS, Cohen L, Ma H, Xu H, Ayanian N, et al. Summary: Multi-agent path finding with kinematic constraints. In: International joint conference on artificial intelligence; 2017. p. 4869–4873.

96.• Sartoretti G, Kerr J, Shi Y, Wagner G, Kumar TS, Koenig S, et al. Primal: Pathfinding via reinforcement and imitation multi-agent learning. IEEE Robot Autom Lett. 2019;4(3):2378–2385. **This article presents an algorithm based on multi-agent reinforcement learning for solving MAPF in a destributed setting.**

97. Damani M, Luo Z, Wenzel E, Sartoretti G. PRIMAL_2: Pathfinding via reinforcement and imitation multi-agent learning-lifelong. IEEE Robotics and Automation Letters. 2021;6(2):2666–73.

98. Li Q, Gama F, Ribeiro A, Prorok A. Graph neural networks for decentralized multi-robot path planning. In: IEEE/RSJ international conference on intelligent robots and systems; 2020. p. 11785–11792.

99. Ma Z, Luo Y, Ma H. Distributed heuristic multi-agent path finding with communication. In: IEEE international conference on robotics and automation; 2021. p. 8699–8705.

100. Li Q, Lin W, Liu Z, Prorok A. Message-aware graph attention networks for large-scale multi-robot path planning. IEEE Robotics and Automation Letters. 2021;6(3):5533–40.

101. Wang B, Liu Z, Li Q, Prorok A. Mobile robot path planning in dynamic environments through Globally Guided Reinforcement Learning. IEEE Robotics and Automation Letters. 2020;5(4):6932–9.

102. van den Berg J, Guy SJ, Lin MC, Manocha D. Reciprocal *n*-body collision avoidance. In: International symposium on robotics research. vol. 70 of Springer Tracts in Advanced Robotics;. p. 3–19.

103. Guy SJ, Chhugani J, Kim C, Satish N, Lin M, Manocha D, et al. Clearpath: highly parallel collision avoidance for multi-agent simulation. In: ACM SIGGRAPH/Eurographics symposium on computer animation; 2009. p. 177–187.

104. Zhou D, Wang Z, Bandyopadhyay S, Schwager M. Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells. IEEE Robotics and Automation Letters. 2017;2(2):1047–54.

105. Yu J, LaValle SM. Multi-agent path planning and network flow. In: Algorithmic foundations of robotics X, Springer Tracts in Advanced Robotics. vol. 86. Springer; 2013. p. 157–173.

106. Hönig W, Kiesel S, Tinka A, Durham JW, Ayanian N. Conflict-based search with optimal task assignment. In: International conference on autonomous agents and multiagent systems; 2018. p. 757–765.

107. Ma H, Koenig S. Optimal target assignment and path finding for teams of agents. In: International conference on autonomous agents and multiagent systems; 2016. p. 1144–1152.

108. Nguyen V, Obermeier P, Son TC, Schaub T, Yeoh W. Generalized target assignment and path finding using answer set programming. In: International joint conference on artificial intelligence; 2017. p. 1216–1223.

109. Henkel C, Abbenseth J, Toussaint M. An optimal algorithm to solve the combined task allocation and path finding problem. In: IEEE/RSJ International conference on intelligent robots and systems; 2019. p. 4140–4146.

110. Surynek P. Multi-goal multi-agent path finding via decoupled and integrated goal vertex ordering. In: AAAI conference on artificial intelligence; 2021. p. 12409–12417.

111. Zhong X, Li J, Koenig S, Ma H. Optimal and bounded-suboptimal multi-goal task assignment and path finding. In: IEEE international conference on robotics and automation; 2022. p. in press.

112. Švancara J, Vlk M, Stern R, Atzmon D, Barták R. Online multi-agent pathfinding. In: AAAI conference on artificial intelligence; 2019. p. 7732–7739.

113. Ma H. A competitive analysis of online multi-agent path finding. In: International conference on automated planning and scheduling; 2021. p. 234–242.

114. Li J, Tinka A, Kiesel S, Durham JW, Kumar TKS, Koenig S. Lifelong multi-agent path finding in large-scale warehouses. In: AAAI conference on artificial intelligence (AAAI); 2021. p. 11272–11281.

115.• Ma H, Li J, Kumar TKS, Koenig S. Lifelong multi-agent path finding for online pickup and delivery tasks. In: International conference on autonomous agents and multiagent systems; 2017. p. 837–845. **This paper presents the first study on combining task-assignment and MAPF in an online setting.**

116. Ma H, Hönig W, Kumar TKS, Ayanian N, Koenig S. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery. In: AAAI conference on artificial intelligence; 2019. p. 7651–7658.

117. Liu M, Ma H, Li J, Koenig S. Target assignment and path planning for multi-agent pickup and delivery. In: International conference on autonomous agents and multiagent systems; 2019. p. 2253–2255.

118. Koenig S.: MAPF information page. Accessed 27 Feb 2022. [Online]. Available from: http://mapf.info/.