Proceedings of the Twelfth International
Symposium on Combinatorial Search (SoCS 2019)

# Extended Abstract: Searching with Consistent Prioritization for Multi-Agent Path Finding*

**Hang Ma,[1] Daniel Harabor,[2] Peter J. Stuckey,[2] Jiaoyang Li,[1] Sven Koenig[1]**
[1]University of Southern California
[2]Monash University
hangma@usc.edu, {daniel.harabor,peter.stuckey}@monash.edu, {jiaoyanl,skoenig}@usc.edu

## Abstract

This is an extended abstract of a previously published paper at AAAI 2019 (Ma et al. 2019). We generalize prioritized planning for Multi-Agent Path Finding from planning with a fixed total priority ordering of all agents to planning with all possible partial priority orderings. We present new theoretical results on its limitations in terms of completeness and optimality. We also present a novel prioritized algorithmic framework and demonstrate state-of-the-art solution qualities and success rates, often with runtimes similar to those of existing prioritized algorithms.

## Introduction

In Multi-Agent Path Finding (MAPF) (Ma and Koenig 2017), we are given a connected undirected graph $G = (V, E)$ and $M$ agents $\{a_i \mid i \in [M]\}$ ($[M] = \{1, \ldots, M\}$). Each $a_i$ is given a unique start vertex $s_i \in V$ and a unique target vertex $t_i \in V$ and either moves to an adjacent vertex or waits at the same vertex at each discrete time $t = 0, \ldots, \infty$. Let $\pi_i(t)$ denote the vertex occupied by $a_i$ at $t$. A *plan* consists of a set of paths, one path $\pi_i = \langle \pi_i(0), \ldots, \pi_i(T_i), \pi_i(T_i + 1), \ldots \rangle$ for each $a_i$, where $\pi_i(0) = s_i$ and $\pi_i(t) = t_i$ for all times $t = T_i, \ldots, \infty$. The *arrival time* $T_i$ of agent $a_i$ at $t_i$ is the earliest time when it has reached $t_i$ and stops moving. A *vertex collision* is a tuple $\langle a_i, a_j, v, t \rangle$ where $a_i$ and $a_j$ occupy the same $v$ at the same $t$. An *edge collision* is a tuple $\langle a_i, a_j, u, v, t \rangle$ where $a_i$ and $a_j$ traverse $(u, v) \in E$ in opposite directions at the same $t$. A solution is a plan that consists of collision-free paths for all agents. Its quality is measured by the *flowtime* $\sum_{i \in [M]} T_i$, the sum of the arrival times of all agents.

MAPF arises in many applications, such as for aircraft-towing vehicles (Morris et al. 2016), warehouse and office robots (Wurman, D'Andrea, and Mountz 2008; Veloso et al. 2015), game characters (Ma et al. 2017b), and other multi-agent systems (Ma et al. 2017a). MAPF is NP-hard to solve optimally (Yu and LaValle 2013b; Ma et al. 2016b) and can be solved with reductions to other well-studied combinatorial problems (Surynek 2015; Yu and LaValle 2013a; Erdem et al. 2013) and dedicated MAPF algorithms (Standley and Korf 2011; Luna and Bekris 2011; Goldenberg et al. 2014; Sharon et al. 2013; Wagner and Choset 2015; Sharon et al. 2015), as described in several surveys (Ma et al. 2016a; Felner et al. 2017).

Prioritized MAPF algorithms (Silver 2005; Sturtevant and Buro 2006) use the following simple prioritized-planning scheme (Erdmann and Lozano-Pérez 1987): Each $a_i$ is given a unique priority and computes, in priority order, a minimum-cost path from $s_i$ to $t_i$ that avoids collisions with the (already planned) paths of all agents with higher priorities. Existing (standard) prioritized MAPF algorithms are often used as parts of MAPF solvers (Velagapudi, Sycara, and Scerri 2010; Wang and Botea 2011; Cáp, Vokrínek, and Kleiner 2015). However, they use a predefined total priority ordering of the agents and can thus result in solutions of bad quality or even fail to find any solutions for solvable MAPF instances, where a different total priority ordering could have resulted in solutions of higher quality. In this paper, we thus consider a generalized form of prioritized planning with all possible total priority orderings. We discuss the limitations of prioritized planning. We also develop two prioritized MAPF algorithms, Conflict-Based Search with Priorities (CBSw/P) and Priority-Based Search (PBS), that systematically explore "good" priority orderings.

## Theoretical Results

We summarize the theoretical results: (1) Some MAPF instances that are solvable are not solvable with prioritized planning. (2) Some MAPF instances that are solvable with prioritized planning are only solvable with prioritized planning for a single total priority ordering. (3) Some MAPF instances that are solvable with prioritized planning are not optimally solvable with prioritized planning for any total priority ordering. (4) Even worse, some MAPF instances that are optimally solvable with prioritized planning require prioritized planning not only to use the correct total priority ordering but also break ties correctly when planning paths for the agents, which—if done incorrectly—can prevent priori-

*The research at the University of Southern California was supported by the National Science Foundation (NSF) under grant numbers 1409987, 1724392, 1817189 and 1837779 as well as a gift from Amazon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

tized planning from finding any solution. We refer the reader to the original paper (Ma et al. 2019) for the detailed theorems and proofs.

## Algorithms

Conflict-Based Search with Priorities (CBSw/P) is an adaptation of Conflict-Based Search (CBS) (Sharon et al. 2015) to prioritized planning. It explores the space of all total priority orderings lazily using a systematic best-first search: It introduces an ordered pair of agents only when their paths collide. Priority-Based Search (PBS) explores the space of all total priority orderings lazily using a systematic depth-first search: It takes a user-specified partial priority ordering as input, dynamically adds new ordered pairs of agents to it, and plans paths that are consistent with the resulting partial priority ordering. Standard prioritized MAPF algorithms are special cases of PBS.

We compare CBSw/P and PBS to a state-of-the-art implementation of CBS (Felner et al. 2018) and several PBS variants that simulate standard prioritized MAPF algorithms with different total priority orderings on a 2.50 GHz Intel Core i5-2450M laptop with 6 GB RAM. We find that CBSw/P often computes optimal or near-optimal solutions and is more efficient than CBS. PBS also computes near-optimal solutions and is much more efficient than CBSw/P. Moreover, PBS finds solutions for many MAPF instances where standard prioritized MAPF algorithms cannot and solves MAPF instances with six hundred agents on a video game map in $35.18$ seconds on average.

We refer the reader to the original paper (Ma et al. 2019) for a detailed description and theoretical analysis of CBSw/P and PBS and more experimental insights.

## References

Čáp, M.; Vokrínek, J.; and Kleiner, A. 2015. Complete decentralized method for on-line multi-robot trajectory planning in well-formed infrastructures. In *ICAPS*, 324–332.

Erdem, E.; Kisa, D. G.; Oztok, U.; and Schueller, P. 2013. A general formal framework for pathfinding problems with multiple agents. In *AAAI*, 290–296.

Erdmann, M. A., and Lozano-Pérez, T. 1987. On multiple moving objects. *Algorithmica* 2:477–521.

Felner, A.; Stern, R.; Shimony, S. E.; Boyarski, E.; Goldenberg, M.; Sharon, G.; Sturtevant, N.; Wagner, G.; and Surynek, P. 2017. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In *SoCS*, 29–37.

Felner, A.; Li, J.; Boyarski, E.; Ma, H.; Cohen, L.; Kumar, T. K. S.; and Koenig, S. 2018. Adding heuristics to conflict-based search for multi-agent path finding. In *ICAPS*, 83–87.

Goldenberg, M.; Felner, A.; Stern, R.; Sharon, G.; Sturtevant, N.; Holte, R. C.; and Schaeffer, J. 2014. Enhanced Partial Expansion A*. *Journal of Artificial Intelligence Research* 50:141–187.

Luna, R., and Bekris, K. E. 2011. Push and Swap: Fast cooperative path-finding with completeness guarantees. In *IJCAI*, 294–300.

Ma, H., and Koenig, S. 2017. AI buzzwords explained: Multi-agent path finding (MAPF). *AI Matters* 3(3):15–19.

Ma, H.; Koenig, S.; Ayanian, N.; Cohen, L.; Hönig, W.; Kumar, T. K. S.; Uras, T.; Xu, H.; Tovey, C.; and Sharon, G. 2016a. Overview: Generalizations of multi-agent path finding to real-world scenarios. In *IJCAI-16 Workshop on Multi-Agent Path Finding*.

Ma, H.; Tovey, C.; Sharon, G.; Kumar, T. K. S.; and Koenig, S. 2016b. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *AAAI*, 3166–3173.

Ma, H.; Hönig, W.; Cohen, L.; Uras, T.; Xu, H.; Kumar, T. K. S.; Ayanian, N.; and Koenig, S. 2017a. Overview: A hierarchical framework for plan generation and execution in multi-robot systems. *IEEE Intelligent Systems* 32(6):6–12.

Ma, H.; Yang, J.; Cohen, L.; Kumar, T. K. S.; and Koenig, S. 2017b. Feasibility study: Moving non-homogeneous teams in congested video game environments. In *AIIDE*, 270–272.

Ma, H.; Harabor, D.; Stuckey, P. J.; Li, J.; and Koenig, S. 2019. Searching with consistent prioritization for multi-agent path finding. In *AAAI*.

Morris, R.; Pasareanu, C.; Luckow, K.; Malik, W.; Ma, H.; Kumar, T. K. S.; and Koenig, S. 2016. Planning, scheduling and monitoring for airport surface operations. In *AAAI-16 Workshop on Planning for Hybrid Systems*.

Sharon, G.; Stern, R.; Goldenberg, M.; and Felner, A. 2013. The increasing cost tree search for optimal multi-agent pathfinding. *Artificial Intelligence* 195:470–495.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219:40–66.

Silver, D. 2005. Cooperative pathfinding. In *AIIDE*, 117–122.

Standley, T. S., and Korf, R. E. 2011. Complete algorithms for cooperative pathfinding problems. In *IJCAI*, 668–673.

Sturtevant, N., and Buro, M. 2006. Improving collaborative pathfinding using map abstraction. In *AIIDE*, 80–85.

Surynek, P. 2015. Reduced time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally. In *IJCAI*, 1916–1922.

Velagapudi, P.; Sycara, K.; and Scerri, P. 2010. Decentralized prioritized planning in large multirobot teams. In *IROS*, 4603–4609.

Veloso, M.; Biswas, J.; Coltin, B.; and Rosenthal, S. 2015. CoBots: Robust symbiotic autonomous mobile service robots. In *IJCAI*, 4423–4429.

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

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

Wurman, P. R.; D'Andrea, R.; and Mountz, M. 2008. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine* 29(1):9–20.

Yu, J., and LaValle, S. M. 2013a. Planning optimal paths for multiple robots on graphs. In *ICRA*, 3612–3617.

Yu, J., and LaValle, S. M. 2013b. Structure and intractability of optimal multi-robot path planning on graphs. In *AAAI*, 1444–1449.