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

Extended Abstract: Lifelong Path Planning with Kinematic Constraints for Multi-Agent Pickup and Delivery*

Hang Ma, Wolfgang Hönig, T. K. Satish Kumar, Nora Ayanian, Sven Koenig
Department of Computer Science
University of Southern California
{hangma, whoenig}@usc.edu, tkskwork@gmail.com, {ayanian, skoenig}@usc.edu

Abstract

This is an extended abstract of a previously published paper at AAAI 2019 (Ma et al. 2019). We study the Multi-Agent Pickup and Delivery (MAPD) problem where a large number of agents attend to a stream of incoming pickup-and-delivery tasks. We present an efficient and effective MAPD algorithm that can compute paths with continuous agent movements for hundreds of agents and thousands of tasks in seconds.

Introduction

In the Multi-Agent Pickup and Delivery (MAPD) problem (Ma et al. 2017), we are given a 2-dimensional 4-neighbor grid with cells of size $L \times L$ each and m agents that attend to a stream of incoming pickup-and-delivery tasks. Each task τ_j has a pickup cell s_j and a delivery cell g_j and is added to the system at an unknown time. The task set $\mathcal{T}$ is the set of unassigned tasks. An agent not executing a task can be assigned any task $\tau_j \in \mathcal{T}$. In order to execute τ_j, the agent has to move from its current cell via s_j to g_j. Applications include warehouse robots that move shelves (Wurman, D'Andrea, and Mountz 2008), aircraft towing robots that move planes (Morris et al. 2016), and office delivery robots that move packages (Veloso et al. 2015).

Existing MAPD algorithms repeatedly solve the multi-agent pathfinding (MAPF) problem (Ma and Koenig 2017) that computes collision-free paths for multiple agents assuming discrete agent movements and is NP-hard to solve optimally (Yu and LaValle 2013b; Ma et al. 2016b). MAPF algorithms include reductions to other problems (Yu and LaValle 2013a; Erdem et al. 2013; Surynek 2015) and dedicated algorithms (Wang and Botea 2011; Sharon et al. 2013; Wagner and Choset 2015; Sharon et al. 2015; Ma and Koenig 2016; Ma, Kumar, and Koenig 2017), as described in several surveys (Ma et al. 2016a; Felner et al. 2017).

In this paper, we present an efficient and effective MAPD algorithm based on the recent MAPD algorithm Token Passing (TP) (Ma et al. 2017). TP assumes discrete agent movements in the main compass directions with uniform velocity but can use MAPF-POST (Hönig et al. 2016a; 2016b) in a post-processing step to adapt its paths to continuous movements with given velocities. We propose to combine TP with Safe Interval Path Planning with Reservation Table (SIPPwRT), our contribution to improving Safe Interval Path Planning (SIPP) (Phillips and Likhachev 2011). The resulting MAPD algorithm TP-SIPPwRT directly computes continuous forward movements and point turns with given velocities, guarantees a safety distance between agents, and solves all well-formed MAPD instances (Cáp, Vokrínek, and Kleiner 2015), a realistic subset of MAPD instances that models automated warehouses and other real-world environments.

TP-SIPPwRT

TP (Ma et al. 2017) lets agents repeatedly plan paths for themselves using space-time A* (Silver 2005) considering the other agents as dynamic obstacles that follow their paths and with which collisions need to be avoided. It uses a token (a synchronized block of shared memory) that stores the task set $\mathcal{T}$ and the current paths. Each time an agent that is currently not following a path gets the token, it greedily assigns itself a task τ_j from $\mathcal{T}$. Then, it plans a time-minimal path from its current cell to s_j and then a time-minimal path from s_j to g_j that both avoid collisions with all other paths in the token. Finally, it releases the token and follows the planned paths.

We propose to replace space-time A* with SIPPwRT, that computes continuous forward movements and point turns with given velocities. We assume that each agent is a disk with radius $\leq L/2$ and always moves from the center of its current cell to the center of an adjacent unblocked cell via the following available actions, besides waiting: a point turn of $\pi/2$ rads (ninety degrees) in either clockwise or counter-clockwise direction with a given rotational velocity and a forward movement to the center of the adjacent cell with a given translational velocity.

Space-time A* and SIPP are two versions of A* with discrete agent movements: (1) Space-time A* operates on pairs of cells and time steps; and (2) SIPP operates on pairs of cells and *safe intervals*. A safe interval of a cell is the contiguous time steps during which a cell is not occupied. Therefore, it is always preferable for an agent to arrive at a cell earlier during the same safe interval since it can then

*Our research was supported by the National Science Foundation (NSF) under grant numbers 1409987, 1724392, 1817189 and 1837779 as well as a gift from Amazon.

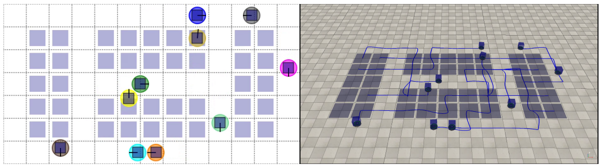

Figure 1: Left: Agent simulator. Right: Robot simulator.

simply wait at the cell. If SIPP has found a path that arrives at some cell at some time during some safe interval and then discovers a path that arrives at the same cell at a later time in the same safe interval, then it can prune the latter path without losing optimality.

However, SIPP represents the path of each dynamic obstacle as a chronologically ordered list of cells and is thus not efficient since it has to iterate through all these lists to calculate all safe intervals of a given cell. SIPPwRT improves upon SIPP using a novel data structure, called *reservation table*, that handles continuous agent movements with given velocities. Each reservation table entry of a given cell is a priority queue that contains all *reserved intervals* for that cell, each being a maximal contiguous interval during which the cell is occupied by some dynamic obstacle, in increasing order of their lower bounds. The reservation table thus allows SIPPwRT to efficiently (1) calculate all safe intervals of a given cell; (2) add reservation table entries after a new path has been calculated; and (3) delete reservation table entries that refer to irrelevant times in the past in order to keep the reservation table small.

We demonstrate the benefits of TP-SIPPwRT for automated warehouses using both an agent simulator with perfect path execution and a standard robot simulator with imperfect path execution resulting from unmodeled kinodynamic constraints and motion noise by the MAPD algorithms (Figure 1). We report our experimental results on a 2.50 GHz Intel Core i5-2450M laptop with 6 GB RAM: (1) The planning time of TP-SIPPwRT is less than 16 seconds for up to 250 agents and 2,000 tasks in the agent simulator; and (2) all robots follow their paths safely in the robot simulator. Videos of sample experiments can be found at

http://idm-lab.org/project-p.html

We refer the reader to the original paper (Ma et al. 2019) for the detailed description and theoretical analysis of SIPPwRT and more experimental insights, including comparisons to existing MAPD algorithms.

References

Cáp, 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.

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.

Hönig, W.; Kumar, T. K. S.; Cohen, L.; Ma, H.; Xu, H.; Ayanian, N.; and Koenig, S. 2016a. Multi-agent path finding with kinematic constraints. In *ICAPS*, 477–485.

Hönig, W.; Kumar, T. K. S.; Ma, H.; Ayanian, N.; and Koenig, S. 2016b. Formation change for robot groups in occluded environments. In *IROS*, 4836–4842.

Ma, H., and Koenig, S. 2016. Optimal target assignment and path finding for teams of agents. In *AAMAS*, 1144–1152.

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.; Li, J.; Kumar, T. K. S.; and Koenig, S. 2017. Lifelong multi-agent path finding for online pickup and delivery tasks. In *AAMAS*, 837–845.

Ma, H.; Hönig, W.; Kumar, T. K. S.; Ayanian, N.; and Koenig, S. 2019. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery. In *AAAI*.

Ma, H.; Kumar, T. K. S.; and Koenig, S. 2017. Multi-agent path finding with delay probabilities. In *AAAI*, 3605–3612.

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

Phillips, M., and Likhachev, M. 2011. SIPP: Safe interval path planning for dynamic environments. In *ICRA*, 5628–5635.

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. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219:40–66.

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

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

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.