Overview: Generalizations of Multi-Agent Path Finding to Real-World Scenarios

Hang Ma
Simon Fraser University
32 PUBLICATIONS **361** CITATIONS
SEE PROFILE

Liron Cohen
University of Southern California
26 PUBLICATIONS **180** CITATIONS
SEE PROFILE

Wolfgang Hoenig
California Institute of Technology
17 PUBLICATIONS **225** CITATIONS
SEE PROFILE

Hong Xu
University of Southern California
25 PUBLICATIONS **162** CITATIONS
SEE PROFILE

Overview: Generalizations of Multi-Agent Path Finding to Real-World Scenarios

Hang Ma, Sven Koenig, Nora Ayanian, Liron Cohen
Wolfgang Hönig, T. K. Satish Kumar, Tansel Uras, Hong Xu
Department of Computer Science
University of Southern California
{hangma,skoenig,ayanian,lironcoh,whoenig,turas,hongx}@usc.edu, tkskwork@gmail.com

Craig Tovey
School of ISyE
Georgia Institute of Technology
ctovey@isye.gatech.edu

Guni Sharon
Department of Computer Science
The University of Texas at Austin
gunisharon@gmail.com

Abstract

Multi-agent path finding (MAPF) is well-studied in artificial intelligence, robotics, theoretical computer science and operations research. We discuss issues that arise when generalizing MAPF methods to real-world scenarios and four research directions that address them. We emphasize the importance of addressing these issues as opposed to developing faster methods for the standard formulation of the MAPF problem.

1 Introduction

Multi-agent path finding (MAPF) has been well-studied by researchers from artificial intelligence, robotics, theoretical computer science and operations research. The task of (standard) MAPF is to find the paths for multiple agents in a given graph from their current vertices to their targets without colliding with other agents, while at the same time optimizing a cost function. Existing MAPF methods use, for example, reductions to problems from satisfiability, integer linear programming or answer set programming [Yu and LaValle, 2013b; Erdem *et al.*, 2013; Surynek, 2015] or optimal, bounded-suboptimal or suboptimal search methods [Silver, 2005; Sturtevant and Buro, 2006; Ryan, 2008; Wang and Botea, 2008; Standley, 2010; Standley and Korf, 2011; Wang and Botea, 2011; Luna and Bekris, 2011; Sharon *et al.*, 2013; de Wilde *et al.*, 2013; Barer *et al.*, 2014; Goldenberg *et al.*, 2014; Wagner and Choset, 2015; Boyarski *et al.*, 2015; Sharon *et al.*, 2015].

We have recently studied various issues that arise when generalizing MAPF to real-world scenarios, including Kiva (Amazon Robotics) warehouse systems [Wurman *et al.*, 2008] (Figure 1) and autonomous aircraft towing vehicles [Morris *et al.*, 2016]. These issues can be categorized into two general concerns: 1. Developing faster methods for the standard formulation of the MAPF problem is insufficient because, in many real-world scenarios, new structure can be exploited or new problem formulations are required. 2. Studying MAPF or its new formulations only as combinatorial optimization problems is insufficient because the resulting MAPF solutions also need to be executed. We discuss four research directions that address both concerns from different perspectives:

1. In many real-world multi-agent systems, agents are partitioned into teams, targets are given to teams, and each agent in a team needs to get assigned a target from the team, before one finds paths for all agents. We have formulated the *combined target assignment and path finding* (TAPF) problem for teams of agents to address this issue. We have also developed an optimal TAPF method that scales to dozens of teams and hundreds of agents [Ma and Koenig, 2016].
2. In many real-world multi-agent systems, agents are anonymous (exchangeable), but their payloads are non-anonymous (non-exchangeable) and need to be delivered to given targets. The agents can often exchange their payloads in such systems. We have formulated the *package-exchange robot routing* (PERR) problem as a first attempt to tackle more general transportation problems where payload transfers are allowed [Ma *et al.*, 2016]. In this context, we have also proved the hardness of approximating optimal MAPF solutions.
3. In many real-world multi-agent systems, the consistency of agent motions and the resulting predictability of agent motions is important (especially in work spaces shared by humans and agents), which is not taken into account by existing MAPF methods. We have exploited the problem structure of given MAPF instances in two stages: In the first stage, we have developed a scheme for finding paths for the agents that include many edges from user-provided highways, which achieves consistency and predictability of agent motions [Cohen *et al.*, 2015]. In the second stage, we have developed methods that automatically generate highways [Cohen *et al.*, 2016].
4. MAPF is mostly motivated by navigation or motion planning for multi-robot systems. However, the

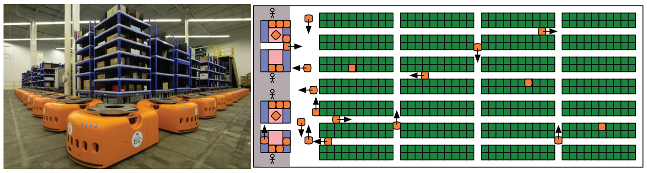

Figure 1: Autonomous drive units and storage pods that contain products and can be moved by the drive units (left) and the layout of a typical Kiva warehouse system (right) [Wurman *et al.*, 2008].

optimality or bounded-suboptimality of MAPF solutions does not necessarily entail their robustness, especially given the imperfect plan-execution capabilities of real-world robots. We have developed a framework that efficiently postprocesses the output of a MAPF method to create a plan-execution schedule that can be executed by real-world multi-robot systems [Hönig *et al.*, 2016].

We now showcase the practicality of these research directions to demonstrate that, in order to generalize MAPF methods to real-world scenarios, addressing both concerns is as important, if not more, than developing faster methods for the standard formulation of the MAPF problem.

2 Combined Target Assignment and Path Finding (TAPF) for Teams of Agents

Targets are often given to teams of agents. Each agent in a team needs to get assigned a target given to its team so that the paths of the agents from their current vertices to their target optimize a cost function. For example, in a Kiva warehouse system, the drive units that relocate storage pods from the inventory stations to the storage locations form a team because each of them needs to get assigned an available storage location. Previous MAPF methods assume that each agent is assigned a target in advance by some target-assignment procedure but, to achieve optimality, we have formulated TAPF, which couples the target-assignment and the path-finding problems and defines one common objective for both of them. In TAPF, the agents are partitioned into teams. Each team is given the same number of unique targets as there are agents in the team. The task of TAPF is to assign the targets to the agents and plan collision-free paths for the agents from their current vertices to their targets in a way such that each agent moves to exactly one target given to its team, all targets are visited and the makespan (the earliest time step when all agents have reached their targets and stop moving) is minimized. Any agent in a team can get assigned a target of the team, and the agents in the same team are thus exchangeable. However, agents in different teams are not exchangeable. TAPF can be viewed as a generalization of (standard) MAPF and an anonymous variant of MAPF:

- **(Standard) MAPF** results from TAPF if every team consists of exactly one agent and the number of teams thus equals the number of agents. The assignments of targets to agents are pre-determined and the agents are thus non-anonymous (non-exchangeable).
- **The anonymous variant of MAPF** (also called goal-invariant MAPF) results from TAPF if only one team exists (that consists of all agents). The agents can get assigned any target and are thus exchangeable. It can be solved optimally in polynomial time with flow-based MAPF methods [Yu and LaValle, 2013a; Turpin *et al.*, 2014].

The state-of-the-art optimal TAPF method, called the Conflict-Based Min-Cost Flow [Ma and Koenig, 2016], combines search and flow-based MAPF methods. It generalizes to dozens of teams and hundreds of agents.

3 Package-Exchange Robot Routing (PERR) and New Complexity Results for MAPF

Agents are often anonymous but carry payloads (packages) that are assigned targets and are thus non-anonymous. For example, in a Kiva warehouse system, the drive units are anonymous but the storage pods they carry are assigned storage locations and are thus non-anonymous. If each agent carries one package, the problem is equivalent to (standard) MAPF. In reality, the packages can often be transferred among agents, which results in more general transportation problems, for example, ride-sharing with passenger transfers [Coltin and Veloso, 2014] and package delivery with robots in offices [Veloso *et al.*, 2015]. We have formulated PERR as a first step toward understanding these problems [Ma *et al.*, 2016]. In PERR, each agent carries one package, any two agents in adjacent vertices can exchange their packages, and each package needs to be delivered to a given target. PERR can thus be viewed as a modification of (standard) MAPF:

- Packages in PERR can be viewed as agents in (standard) MAPF which move by themselves.
- Two packages in adjacent vertices are allowed to exchange their vertices in PERR but two agents in adjacent vertices are not allowed to exchange their vertices in (standard) MAPF.

K-PERR is a generalization of PERR where packages are partitioned into K types and packages of the same type are exchangeable. Since, in TAPF, agents are partitioned into teams and agents in the same team are exchangeable, K-PERR can be viewed as a modification of TAPF with K teams in the same sense that PERR can be viewed as a modification of (standard) MAPF. We have proved the hardness of approximating optimal PERR and K-PERR solutions (for $K \geq 2$). One corollary of our study is that both MAPF and TAPF are NP-hard to approximate within any factor less than 4/3 for makespan minimization, even when there are only two teams for TAPF. We have also demonstrated that the addition of exchange operations to MAPF does not reduce its complexity theoretically but makes PERR easier to solve than MAPF experimentally. There is a continuum of problems that arise in different real-world scenarios: "One agent with many packages" yields the classic rural postman problem; "as many agents as packages" yields MAPF, TAPF or PERR. Understanding both extremes helps

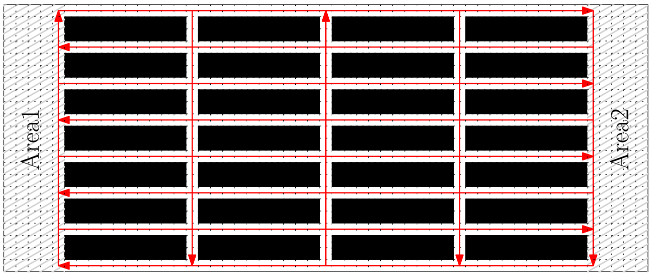

Figure 2: User-provided highways in a simulated Kiva warehouse system.

one to attack the middle, as required by many other real-world scenarios.

4 Exploitation of Problem Structure and Predictability of Motions

Agents share their work spaces with humans, and the consistency of their motions and the resulting predictability of their motions are important for the safety of the humans, which existing MAPF methods do not take into account. This motivates us to exploit the problem structure of given MAPF instances and design a scheme that encourages agents to move along user-provided sets of edges (called highways) [Cohen *et al.*, 2015]. We use highways in the context of a simple inflation scheme based on the ideas behind experience graphs [Phillips *et al.*, 2012] to derive new heuristic values that encourage MAPF methods to return paths that include the edges of the highways, which avoids head-to-head collisions among agents and achieves consistency and predictability of their motions. For example, in a Kiva warehouse system, we can design highways along the narrow passageways between the storage locations as shown by the arrows in Figure 2. We have demonstrated in a simulated Kiva warehouse system that such highways accelerate MAPF methods significantly while maintaining the desired bounded-suboptimality of the MAPF solution costs. The problem structure of TAPF and PERR instances can also be exploited with the same methods. We have also developed methods that automatically generate highways that are competitive with user-provided ones in our feasibility study [Cohen *et al.*, 2016].

5 Dealing with Imperfect Plan-Execution Capabilities

State-of-the-art MAPF or TAPF methods can find collision-free paths for hundreds of agents optimally or with user-provided sub-optimality guarantees in a reasonable amount of computation time. They perform well even in cluttered and tight environments, such as Kiva warehouse systems. However, agents often have imperfect plan-execution capabilities and are not able to synchronize their motions perfectly, which can result in frequent and time-expensive replanning. Therefore, we have proposed a framework that makes use of a simple temporal network to postprocess a MAPF solution efficiently to create a plan-execution schedule that works for non-holonomic robots, takes their maximum translational and rotational velocities into account, provides a guaranteed safety distance between them and exploits slack (defined as the difference of the latest and earliest entry times of locations) to absorb imperfect plan executions and avoid time-intensive replanning in many cases [Hönig *et al.*, 2016]. This framework has been evaluated in simulation and on real robots. TAPF and PERR methods can also be applied in the same framework. Issues to be addressed in future work include adding user-provided safety distances, additional kinematic constraints, planning with uncertainty and replanning.

6 Conclusions

We discussed four research directions that address issues that arise when generalizing MAPF methods to real-world scenarios and exploit either the problem structure or existing MAPF methods. Our goal was to point out interesting research directions for researchers working in the field of MAPF.

7 Acknowledgments

The research at USC was supported by ARL under grant number W911NF-14-D-0005, ONR under grant numbers N00014-14-1-0734 and N00014-09-1-1031, NASA via Stinger Ghaffarian Technologies and NSF under grant numbers 1409987 and 1319966. 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.

References

[Barer *et al.*, 2014] M. Barer, G. Sharon, R. Stern, and A. Felner. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Annual Symposium on Combinatorial Search*, pages 19–27, 2014.

[Boyarski *et al.*, 2015] E. Boyarski, A. Felner, R. Stern, G. Sharon, D. Tolpin, O. Betzalel, and S. E. Shimony. ICBS: improved conflict-based search algorithm for multi-agent pathfinding. In *International Joint Conference on Artificial Intelligence*, pages 740–746, 2015.

[Cohen *et al.*, 2015] L. Cohen, T. Uras, and S. Koenig. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding. In *Annual Symposium on Combinatorial Search*, pages 2–8, 2015.

[Cohen *et al.*, 2016] L. Cohen, T. Uras, T. K. S. Kumar, H. Xu, N. Ayanian, and S. Koenig. Improved bounded-suboptimal multi-agent path finding solvers. In *International Joint Conference on Artificial Intelligence*, 2016.

[Coltin and Veloso, 2014] B. Coltin and M. Veloso. Scheduling for transfers in pickup and delivery problems with very large neighborhood search. In *AAAI Conference on Artificial Intelligence*, pages 2250–2256, 2014.

[de Wilde *et al.*, 2013] B. de Wilde, A. W. ter Mors, and C. Witteveen. Push and rotate: Cooperative multi-agent path planning. In *International Conference on Autonomous Agents and Multi-agent Systems*, pages 87–94, 2013.

[Erdem *et al.*, 2013] E. Erdem, D. G. Kisa, U. Oztok, and P. Schueller. A general formal framework for pathfinding problems with multiple agents. In *AAAI Conference on Artificial Intelligence*, pages 290–296, 2013.

[Goldenberg *et al.*, 2014] M. Goldenberg, A. Felner, R. Stern, G. Sharon, N. R. Sturtevant, R. C. Holte, and J. Schaeffer. Enhanced partial expansion A*. *Journal of Artificial Intelligence Research*, 50:141–187, 2014.

[Hönig *et al.*, 2016] W. Hönig, T. K. S. Kumar, L. Cohen, H. Ma, H. Xu, N. Ayanian, and S. Koenig. Multi-agent path finding with kinematic constraints. In *International Conference on Automated Planning and Scheduling*, pages 477–485, 2016.

[Luna and Bekris, 2011] R. Luna and K. E. Bekris. Push and swap: Fast cooperative path-finding with completeness guarantees. In *International Joint Conference on Artificial Intelligence*, pages 294–300, 2011.

[Ma and Koenig, 2016] H. Ma and S. Koenig. Optimal target assignment and path finding for teams of agents. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1144–1152, 2016.

[Ma *et al.*, 2016] H. Ma, C. Tovey, G. Sharon, T. K. S. Kumar, and S. Koenig. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *AAAI Conference on Artificial Intelligence*, pages 3166–3173, 2016.

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

[Phillips *et al.*, 2012] M. Phillips, B. J. Cohen, S. Chitta, and M. Likhachev. E-graphs: Bootstrapping planning with experience graphs. In *Robotics: Science and Systems*, 2012.

[Ryan, 2008] M. Ryan. Exploiting subgraph structure in multi-robot path planning. *Journal of Artificial Intelligence Research*, 31:497–542, 2008.

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

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

[Silver, 2005] D. Silver. Cooperative pathfinding. In *Artificial Intelligence and Interactive Digital Entertainment*, pages 117–122, 2005.

[Standley and Korf, 2011] T. S. Standley and R. E. Korf. Complete algorithms for cooperative pathfinding problems. In *International Joint Conference on Artificial Intelligence*, pages 668–673, 2011.

[Standley, 2010] T. S. Standley. Finding optimal solutions to cooperative pathfinding problems. In *AAAI Conference on Artificial Intelligence*, pages 173–178, 2010.

[Sturtevant and Buro, 2006] N. R. Sturtevant and M. Buro. Improving collaborative pathfinding using map abstraction. In *Artificial Intelligence and Interactive Digital Entertainment*, pages 80–85, 2006.

[Surynek, 2015] P. Surynek. Reduced time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally. In *International Joint Conference on Artificial Intelligence*, pages 1916–1922, 2015.

[Turpin *et al.*, 2014] M. Turpin, K. Mohta, N. Michael, and V. Kumar. Goal assignment and trajectory planning for large teams of interchangeable robots. *Autonomous Robots*, 37(4):401–415, 2014.

[Veloso *et al.*, 2015] M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal. CoBots: Robust Symbiotic Autonomous Mobile Service Robots. In *International Joint Conference on Artificial Intelligence*, pages 4423–4429, 2015.

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

[Wang and Botea, 2008] K. Wang and A. Botea. Fast and memory-efficient multi-agent pathfinding. In *International Conference on Automated Planning and Scheduling*, pages 380–387, 2008.

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

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

[Yu and LaValle, 2013a] J. Yu and S. M. LaValle. Multi-agent path planning and network flow. In E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, editors, *Algorithmic Foundations of Robotics X, Springer Tracts in Advanced Robotics*, volume 86, pages 157–173. Springer, 2013.

[Yu and LaValle, 2013b] J. Yu and S. M. LaValle. Planning optimal paths for multiple robots on graphs. In *IEEE International Conference on Robotics and Automation*, pages 3612–3617, 2013.