AI Buzzwords Explained: Multi-Agent Path Finding (MAPF)

Hang Ma (University of Southern California; hangma@usc.edu)
Sven Koenig (University of Southern California; skoenig@usc.edu)

Kiva Systems was founded in 2003 to develop robot technology that automates the fetching of goods in order-fulfillment centers. It was acquired by Amazon in 2012 and changed its name to Amazon Robotics in 2014. Amazon order-fulfillment centers have inventory stations on the perimeter of the warehouse and storage locations in its center, see Figure 1. Each storage location can store one inventory pod. Each inventory pod holds one or more kinds of goods. A large number of warehouse robots operate autonomously in the warehouse. Each warehouse robot is able to pick up, carry and put down one inventory pod at a time. The warehouse robots move inventory pods from their storage locations to the inventory stations where the needed goods are removed from the inventory pods (to be boxed and eventually shipped to customers) and then back to the same or different empty storage locations (Wurman, D'Andrea, & Mountz, 2008).[1]

These order-fulfillment centers raise a number of interesting optimization problems, such as which paths the robots should take and at which storage locations inventory pods should be stored. Path planning, for example, is tricky since most warehouse space is used for storage locations, resulting in narrow corridors where robots that carry inventory pods cannot pass each other. Warehouse robots operate all day long but a simplified one-shot version of the path-planning problem is the multi-agent path-finding (MAPF) problem, which can be described as follows: On math paper, some cells are blocked. The blocked cells and the current cells of n robots are known. A different unblocked cell is assigned to each of the n robots as its goal cell. The problem is to move the robots from their current cells to their goal cells in discrete time steps and let them wait there. The optimization objective is

[1]See the following YouTube video: https://www.youtube.com/watch?v=6KRjuuEVEZs

to minimize the makespan, that is, the number of time steps until all robots are at their goal cells. During each time step, each robot can move from its current cell to its current cell (that is, wait in its current cell) or to an unblocked neighboring cell in one of the four main compass directions. Robots are not allowed to collide. Two robots collide if and only if, during the same time step, they both move to the same cell or both move to the current cell of the other robot. Figure 2 shows an example, where the red and blue robots have to move to the red and blue goal cells, respectively.

There are also versions of the multi-agent path-finding problem with different optimization objectives than makespan (such as the sum of the time steps of each robot until it is at its goal cell) or slightly different collision or movement rules. For example, solving the eight-puzzle (a toy with eight square tiles in a three by three frame, see Figure 3) is a version of the multi-agent path-finding problem where the tiles are the robots.

Researchers in theoretical computer science, artificial intelligence and robotics have studied multi-agent path finding under slightly different names. They have developed fast (polynomial-time) algorithms that find solutions for different classes of multi-agent path-finding instances (for example, those with at least two unblocked cells not occupied by robots) although not necessarily with good makespans. They have also characterized the complexity of finding optimal (or bounded-suboptimal) solutions and developed algorithms that find them. A bounded-suboptimal solution is one whose makespan is at most a given percentage larger than optimal.

Interestingly, it is slow (NP-hard) to find optimal solutions (Yu & LaValle, 2013c; Ma, Tovey, Sharon, Kumar, & Koenig, 2016), although a slight modification of the multi-agent path-finding problem can be solved in polynomial time with flow algorithms, namely where n un-

Figure 1: Warehouse robot (left), inventory pods (center), and the layout of a small simulated warehouse (right). The left and center photos are courtesy of Amazon Robotics.
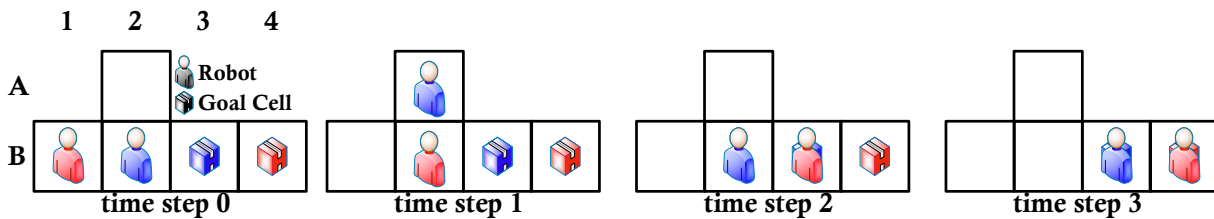

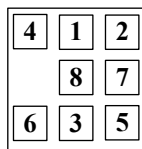Figure 2: A multi-agent path-finding instance with two robots.

Figure 3: The eight-puzzle.

blocked cells are given as goal cells but it is up to the algorithm to assign a different goal cell to each one of the n robots (Yu & LaValle, 2013a). Researchers have also studied versions of the multi-agent path-finding problem where goal cells require robots with certain capabilities (Ma & Koenig, 2016) or robots can exchange their payloads (Ma et al., 2016).

In principle, one can model the original multi-agent path-finding problem as a shortest-path problem on a graph whose vertices correspond to tuples of cells, namely one for each robot, as shown in Figure 4 (where the red path shows the optimal solution), but the number of vertices can be exponential in the number of robots and the shortest path thus cannot be found quickly. Instead, researchers have suggested to plan a shortest path for each robot independently (by ignoring the other robots), which can be done fast. If all robots can follow their paths without colliding, then an optimal solution has been found. If

not, then ...

- there are multi-agent path-finding algorithms that group all colliding robots together and find a solution for the group with minimal makespan (by ignoring the other robots), and then repeat the process. The hope is to find a solution before all robots have been grouped together into one big group (Standley, 2010; Standley & Korf, 2011).

- there are other multi-agent path-finding algorithms that pick a collision between two robots (for example, robots A and B both move to cell x at time step t) and then consider recursively two cases, namely one where robot A is not allowed to move to cell x at time step t and one where robot B is not allowed to move to cell x at time step t. The hope is to find a solution before all possible constraints have been imposed (Sharon, Stern, Felner, & Sturtevant, 2015).

These state-of-the-art multi-agent path-finding algorithms are currently not quite able to find bounded-suboptimal solutions for 100 robots in small warehouses in real-time. The tighter the space, the longer the runtime. Researchers have also suggested a variety of other multi-agent path-finding techniques (Silver, 2005; Sturtevant & Buro, 2006;
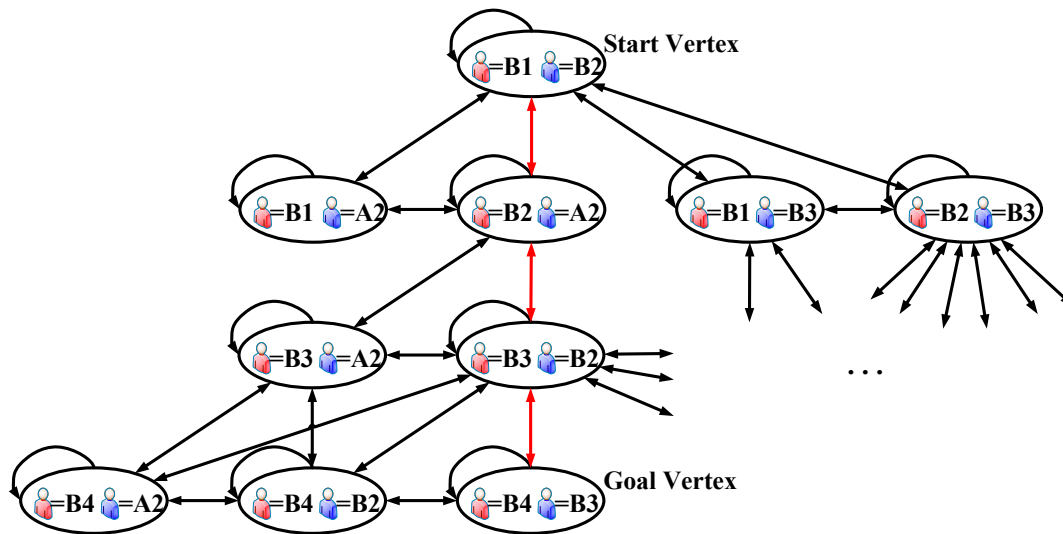
Figure 4: Partial graph for the multi-agent path-finding instance from Figure 2.

Ryan, 2008; Wang & Botea, 2008, 2011; Luna & Bekris, 2011; Sharon, Stern, Goldenberg, & Felner, 2013; de Wilde, ter Mors, & Witteveen, 2013; Barer, Sharon, Stern, & Felner, 2014; Goldenberg et al., 2014; Wagner & Choset, 2015; Boyarski et al., 2015; Ma & Koenig, 2016; Cohen et al., 2016), including some that transform the problem into a different problem for which good solvers exist, such as satisfiability (Surynek, 2015), integer linear programming (Yu & LaValle, 2013b) and answer set programming (Erdem, Kisa, Oztok, & Schueller, 2013). Researchers have also studied how to execute the resulting solutions on actual robots (Cirillo, Pecora, Andreasson, Uras, & Koenig, 2014; Hoenig et al., 2016).

Two workshops have recently been held on the topic, namely the AAAI 2012 Workshop on Multi-Agent Pathfinding[2] and the IJCAI 2016 Workshop on Multi-Agent Path Finding[3]. Recent dissertations include (Wang, 2012; Wagner, 2015; Sharon, 2016).

References

Barer, M., Sharon, G., Stern, R., & Felner, A. (2014). Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Annual symposium on combinatorial search* (pp. 19–27).

Boyarski, E., Felner, A., Stern, R., Sharon, G., Tolpin, D., Betzalel, O., & Shimony, S. (2015). ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In *International joint conference on artificial intelligence* (pp. 740–746).

Cirillo, M., Pecora, F., Andreasson, H., Uras, T., & Koenig, S. (2014). Integrated motion planning and coordination for industrial vehicles. In *International conference on automated planning and scheduling* (pp. 463–471).

Cohen, L., Uras, T., Kumar, T. K. S., Xu, H., Ayanian, N., & Koenig, S. (2016). Improved solvers for bounded-suboptimal multi-agent path finding. In *International joint conference on artificial intelligence* (pp. 3067–3074).

de Wilde, B., ter Mors, A., & Witteveen, C. (2013). Push and rotate: Cooperative multi-agent path planning. In *International conference on autonomous agents and multi-agent systems* (pp. 87–94).

Erdem, E., Kisa, D., Oztok, U., & Schueller, P. (2013). A general formal framework for pathfinding problems with mul-

---

[2]See the following URL:
http://movingai.com/mapf
[3]See the following URL:
http://www.andrew.cmu.edu/user/
gswagner/workshop/ijcai_2016
_multirobot_path_finding.html

tiple agents. In *AAAI conference on artificial intelligence* (pp. 290–296).

Goldenberg, M., Felner, A., Stern, R., Sharon, G., Sturtevant, N., Holte, R., & Schaeffer, J. (2014). Enhanced partial expansion A*. *Journal of Artificial Intelligence Research*, *50*, 141–187.

Hoenig, W., Kumar, S., Cohen, L., Ma, H., Xu, H., Ayanian, N., & Koenig, S. (2016). Multi-agent path finding with kinematic constraints. In *International conference on automated planning and scheduling* (pp. 477–485).

Luna, R., & Bekris, K. (2011). Push and swap: Fast cooperative path-finding with completeness guarantees. In *International joint conference on artificial intelligence* (pp. 294–300).

Ma, H., & Koenig, S. (2016). Optimal target assignment and path finding for teams of agents. In *International conference on autonomous agents and multiagent systems* (pp. 1144–1152).

Ma, H., Tovey, C., Sharon, G., Kumar, T. K. S., & Koenig, S. (2016). Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *AAAI conference on artificial intelligence* (pp. 3166–3173).

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

Sharon, G. (2016). *Novel search techniques for path finding in complex environment* (Unpublished doctoral dissertation). Ben-Gurion University of the Negev.

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

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

Silver, D. (2005). Cooperative pathfinding. In *Artificial intelligence and interactive digital entertainment* (pp. 117–122).

Standley, T. (2010). Finding optimal solutions to cooperative pathfinding problems. In *AAAI conference on artificial intelligence* (pp. 173–178).

Standley, T., & Korf, R. (2011). Complete algorithms for cooperative pathfinding problems. In *International joint conference on artificial intelligence* (pp. 668–673).

Sturtevant, N., & Buro, M. (2006). Improving collaborative pathfinding using map abstraction. In *Artificial intelligence and interactive digital entertainment* (pp. 80–85).

Surynek, P. (2015). Reduced time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally. In *International joint conference on artificial intelligence* (pp. 1916–1922).

Wagner, G. (2015). *Subdimensional expansion: A framework for computationally tractable multirobot path planning* (Unpublished doctoral dissertation). Carnegie Mellon University.

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

Wang, K. (2012). *Scalable cooperative multi-agent pathfinding with tractability and completeness guarantees* (Unpublished doctoral dissertation). Australian National University.

Wang, K., & Botea, A. (2008). Fast and memory-efficient multi-agent pathfinding. In *International conference on automated planning and scheduling* (pp. 380–387).

Wang, K., & 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., D'Andrea, R., & Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, *29*(1), 9–20.

Yu, J., & LaValle, S. (2013a). Multi-agent path planning and network flow. In E. Frazzoli, T. Lozano-Perez, N. Roy, & D. Rus (Eds.), *Algorithmic foundations of robotics x, springer tracts in advanced robotics* (Vol. 86, pp. 157–173). Springer.

Yu, J., & LaValle, S. (2013b). Planning optimal paths for multiple robots on graphs. In *Ieee international conference on robotics and automation* (pp. 3612–3617).

Yu, J., & LaValle, S. (2013c). Structure and intractability of optimal multi-robot path

planning on graphs. In *AAAI conference on artificial intelligence* (pp. 1444–1449).

Hang Ma is a Ph.D. student in computer science at the University of Southern California. He is the recipient of a USC Annenberg Graduate Fellowship. His research interests include artificial intelligence, machine learning and robotics. Additional information about Hang can be found on his webpages: `http://www-scf.usc.edu/~hangma`. Contact him at hangma@usc.edu.

Sven Koenig is a professor in computer science at the University of Southern California. Most of his research centers around techniques for decision making (planning and learning) that enable single situated agents (such as robots) and teams of agents to act intelligently in their environments and exhibit goal-directed behavior in real-time. Additional information about Sven can be found on his webpages: `http://idm-lab.org`. Contact him at skoenig@usc.edu.