Multi-Agent Path Finding on Real Robots

Roman Barták, Ivan Krasičenko, Jiří Švancara
Charles University, Faculty of Mathematics and Physics
Malostranské nám. 25, Praha, Czech Republic
bartak@ktiml.mff.cuni.cz

Abstract

Multi-agent path finding (MAPF) deals with the problem of finding a collision-free path for a set of agents in a graph. It is an abstract version of the problem to coordinate movement for a set of mobile robots. This demo presents software guiding through the MAPF task, starting from the problem formulation and finishing with execution of plans on real robots. Users can design grid-like maps, specify initial and goal locations of robots, generate plans using various abstract models implemented in the Picat programming language, simulate and visualise execution of these plans, and translate the plans to command sequences for Ozobots, small robots developed for teaching programming.

There exists a widely-accepted uniform abstract model of *multi-agent path finding* (MAPF) consisting of an undirected graph describing allowed locations and movements of agents and two possible abstract actions: *move* for moving to a neighbouring node and *wait* for waiting at the current node. The MAPF task is finding a plan, i.e., a collision-free path from a start node to a destination node, for each agent. The research question is if this abstract model is appropriate for problems with real robots.

We present software for experimental evaluation of various MAPF abstract models by executing the obtained plans on real robots. The software provides a visual editor to state the MAPF problems on a grid map, interface for MAPF solvers written in the Picat language, visualisation of plans and plan execution, transformation of plans to control procedures for the Ozobot robots, and tools supporting execution of plans. The software is intended as a research tool for testing various abstract models of the MAPF problem on real robots. The initial results comparing several abstract models were already published (Barták et al. 2018).

Background on MAPF

The MAPF problem is defined by a graph $G = (V, E)$ and a set of agents $a_1, \ldots, a_k$, where each agent a_i is associated with starting location $s_i \in V$ and goal location $g_i \in V$. A grid map with a unit length of each edge is often used to represent the environment (Ryan 2008), Figure 1 shows an example of such a map. The task is to find a collision-free path for each agent from its starting location to its goal location. There exist versions of the MAPF problem, for example, the k-robust version, that is particularly interesting for real robots as the plans are supposed to be robust to possible delays during execution. Formally, k-robust plans require for each vertex of the graph to be unoccupied for at least k time steps before another agent can enter it (Atzmon et al. 2018). Perhaps due to many practical applications in areas such as automated warehouses, interest in MAPF increased in recent years and many solving techniques have been proposed. Our system uses a reduction-based solver in the Picat programming language (Barták et al. 2017) as it is easy to encode versions of the MAPF model there. Picat then solves the problems by translating them to SAT problems.

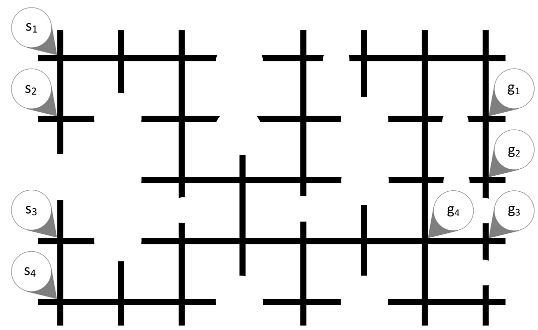

Figure 1: A grid map for MAPF. Agents follow the black line, the gray circles indicate starting and goal locations.

The abstract plan outputted by MAPF solvers is a sequence of locations that the agents visit (or equivalently a sequence of move and wait operations). Before execution on a real robot, the abstract plan needs to be translated to a sequence of actions that the physical robot can perform. Our system supports the Ozobot robots (Ozobot & Evollve, Inc. 2018), see Figure 2, that provide high-level actions such as turn left and right and move forward so it is not necessary to deal with low-level control. By concatenating these actions, the agent can perform all the required steps from the abstract plan. This translates to five possible actions at each time step - (1) wait, (2) move forward, (3,4) turn left/right and move, and (5) turn back and move. As the mobile robot can-not move backward directly, turning back is implemented as two turns right (or left).

System Capabilities

The presented system supports the whole process of solving MAPF problems. The user can define a grid map, put obstacles there by removing vertices and edges, and specify initial and goal locations of agents. The map can be printed for usage with Ozobots or it can be displayed on the computer screen and robots can move on the screen directly. The system provides encodings of several MAPF models in the Picat programming language including the classical model (Barták et al. 2017), the 1-robust model (Atzmon et al. 2018), and a model that includes turning actions in addition to move and wait actions (Barták et al. 2018). There is also an interface for adding other models. Problem solving can be directly realized from the software, which generates the problem specification for the solver from the map drawn by the user. The generated plans can then be visualized as a timeline of actions for each robot (Gantt chart). The system can also visualize execution of plans. Finally, the plans can be exported for execution on Ozobots; the system allows users to specify durations of actions for execution. As we already mentioned, the robots can be then placed on a printed map to execute the plans (the map can be printed from the application) or the robots can run on the computer screen with the map displayed there. In this second case, the system also shows where the robots are supposed to be so the user can see how the real plan execution corresponds to expected execution. Figure 3 shows the integrated user interface of the software. Video presenting the system is available at (Švancara and Krasičenko 2019).

Conclusions and Future Steps

The presented system is intended to study various abstract models of the MAPF problem from the perspective of plan execution on real robots Ozobots. The initial empirical evaluation (Barták et al. 2018) showed that there is indeed a gab between widely-used theoretical frameworks for MAPF and deployment of solutions in real environments. A wider experimental study is necessary to understand better the relations between abstract models and real environments. For example, the ratio between the length of edges and the size of robots seems important. The presented system allows

Figure 2: Ozobot Evo from Evollve. Picture is taken from (Ozobot & Evollve, Inc. 2018).

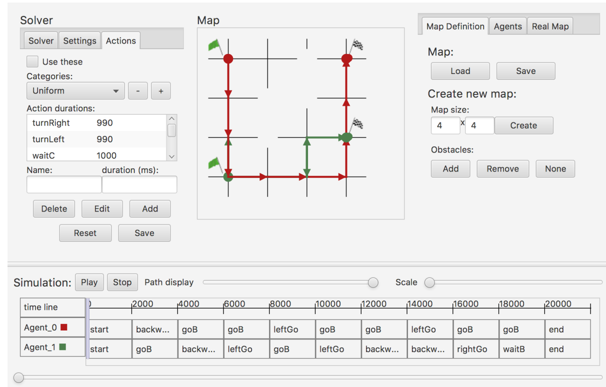

Figure 3: User interface of the MAPF system.

users to define the length of edges so such studies can be realized in future. Similarly, the system allows users to define own abstract models of MAPF so other abstractions can be studied in future. Currently, blind execution of plans is assumed, which means that sensors are not used during execution. It would be interesting to look at plan-execution policies that assume communication between agents and exploit information from sensors. The system allows users to modify the execution strategy by using different command sequences so more advanced execution strategies can be implemented in future.

The presented system provides to the MAPF community a tool for bridging the abstract models and plan execution on real robots. Thanks to using a standard platform of Ozobots, no specific expertise in robotics is necessary.

Acknowledgements

Research is supported by the Czech Science Foundation under the project P103-19-02183S and by the Charles University Grant Agency under the project 90119.

References

Atzmon, D.; Stern, R.; Felner, A.; Wagner, G.; Barták, R.; and Zhou, N. 2018. Robust multi-agent path finding. In *AAMAS 2018*, 1862–1864.

Barták, R.; Zhou, N.-F.; Stern, R.; Boyarski, E.; and Surynek, P. 2017. Modeling and solving the multi-agent pathfinding problem in picat. In *ICTAI 2017*, 959–966. IEEE Computer Society.

Barták, R.; Švancara, J.; Škopková, V.; and Nohejl, D. 2018. Multi-agent path finding on real robots: First experience with ozobots. In *IBERAMIA 2018*, 290–301. Cham: Springer.

Ozobot & Evollve, Inc. 2018. *Ozobot — Robots to code, create, and connect with.* https://ozobot.com/.

Ryan, M. R. K. 2008. Exploiting subgraph structure in multi-robot path planning. *J. Artif. Intell. Res.* 31:497–542.

Švancara, J., and Krasičenko, I. 2019. *MAPF Scenario video demo*. Charles University. https://drive.google.com/file/d/1mkHq_Xgkp8QH3H60I7R27g4Q6RWzIfY7/view.