Research Project Proposal: Multi Agent Path Finding with Removable Obstacles

Matteo Bellusci, matteo.bellusci@mail.polimi.it

1. Introduction to the problem

We start from *Multi Agent Path Finding* (MAPF) [4] since our topic is a generalization of it. The topic lies at the intersection between *Artificial Intelligence* (AI), *Autonomous Agents and Multi-Agent Systems* (AAMAS), and *Autonomous Robotics*.

The task in MAPF problems is to find non-conflicting paths for multiple agents, each one with a unique start and goal position, in a shared discretized environment. Different algorithms are available for solving MAPF problems. A prominent approach is the *Conflict Based Search* (CBS) [12], a complete and optimal two-level solver that pursues the idea of planning for each agent individually and solving emerging conflicts in all the possible ways.

MAPF problems are based on strong assumptions concerning the environment. In fact, the entire environment is modeled as a fixed graph in classical MAPF formulations. This is a strong limitation because in real-world scenarios there are different types of environments and many of them cannot be modeled as a simple graph. For instance, in the field of autonomous navigation, several systems are designed to handle the problem of avoiding dynamic obstacles located in the environment [7], even applying some obstacles detection techniques if the environment is not fully-observable [6]. The classic *Frogger* arcade game [9] is an intuitive example of game that is set in an environment with dynamic obstacles: several moving objects are present in the map. In standard MAPF we deal only with static environments and therefore dynamicity in the environment are handled in opportunistic ways.

We want to overcome the current idea of dynamic environments by introducing the concept of environment reconfiguration. The idea is that something in the environment can be changed, at will, under certain conditions. In general, an environment that is mutable in this sense is called a *reconfigurable* environment. Plainly, different kinds of environments are included in this wide category. Speed up the agents moving process by finding the best arrangement for machinery in a warehouse environment or by choosing the best change to make to the environment, such as knocking down a wall, while remaining within a certain budget, are possible problems at the core of this topic. Whereupon, we focus on reconfigurable environments with destructible obstacles, so we assume that obstacles can just be removed from the environment.

Therefore, we introduce a new type of problem named *Multi-Agent Path Finding with Removable Obstacles* (RO-MAPF), in which a subset of obstacles can be removed, paying a cost, from the starting environment. For simplicity we focus on 4-neighbor grids, but the problem could be extended to arbitrary graphs. A removable obstacle is defined as a set of (blocked) cells and each removable obstacle has an associated cost called *removal cost*. The obstacle cells can be made free, intuitively, paying the corresponding removal cost. In RO-MAPF problems we are given a budget that can be spent on removing obstacles. Moreover, we assume that there are no cells shared by multiple obstacles. For optimally solving a RO-MAPF problem we mean returning the solution with the minimum cost and a set of removable obstacles to remove (within the budget) in order to allow agents to follow their paths without any collision. We keep the solution cost as the only priority. Hence, we are not interested in minimizing the budget usage. Obviously, other settings are possible. An example of RO-MAPF problem instance is shown in Figure 1.

RO-MAPF can model numerous real-world scenarios. Environments with removable obstacles are very recurrent in video games. For instance, first-person shooter (FPS) games are often set in destructible worlds in which, somehow, like using grenades, it is possible to destroy some physical objects [8]. In Kiva (Amazon Robotics)

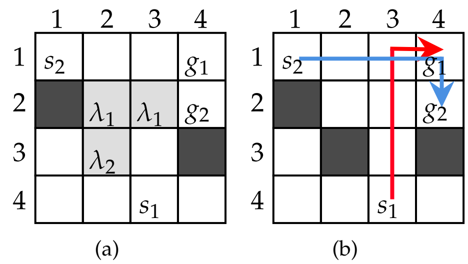

Figure 1: In (a) we show an example of RO-MAPF problem instance in which there are two agents and two removable obstacles: λ_1 composed by two cells and λ_2 composed by one only. We assume that the removal costs and the budget are unitary. In (b) we show the optimal solution associated with the problem (a) in which we removed the obstacle λ_1.

systems [14], movable shelves can be lifted by the agents and therefore can be treated as removable obstacles. Another example is a set in which military vehicles have to pass a barrier of other vehicles placed as perimeter of an area.

2. Main related works

The literature is missing a contribution explicitly devoted to RO-MAPF. Nevertheless, there are some important results concerning path planning in non-static environments.

A type of problem called *Multi-Agent Path Finding with Dynamic Obstacles* (MAPF-DO) [10] considers the presence of *dynamic obstacles* in the map, overcoming the assumption of static environments of classical MAPF. However, in MAPF-DO problems, dynamic obstacles follow given, potentially random, movement rules and there is no possibility to remove them or modify the environment.

One recent attempt to overcome the assumption of unchangeable environments in path planning comes from the computational geometry community. In fact, computing the best path for a single agent in environments with removable obstacles of polygonal shape is a recent topic [1]. However, we are interested in multiple agents. A first attempt to deal with MAPF and removable obstacles was made using a Theta*-based [11] algorithm [2]. As mentioned by the authors, Theta*-based algorithms cannot be directly applied to multi-robot path planning but they can be modified to become base blocks of multi-agent path-finders such as CBS. This last Theta*-based approach is really simple, our goal is to take a step forward and provide a better CBS-based approach.

3. Research plan

MAPF problem is NP-hard to solve optimally [15], even on grid-based environments [3]. RO-MAPF problem is NP-hard to solve optimally too since it is a generalization of the MAPF problem. Despite this intractability issue, our goal is to develop an effective CBS-based algorithm able to solve RO-MAPF problems optimally. Preliminarily, we want to introduce and clearly define the concept of environment reconfiguration in the MAPF context, illustrating also different possible cases that can be addressed in future works. Consequently, the nature of the research is hybrid and lies on the boundary between theory and experimentation.

The portion of work already done includes the analysis of the state of the art, the problem definition, and a large part of the algorithm design phase. We show in detail the main planned phases that are currently unfinished.

- *Algorithms design*: in this phase, we have to design our new CBS-based algorithm and other approaches that can be easily implemented starting from the literature in order to make an experimental comparison. In fact, we must demonstrate the efficiency of our approach and this cannot be done by implementing only our new algorithm. Moreover, we have to prove the completeness and the optimality of our new solver.

- *Implementation*: in this phase, we have to implement our new CBS-based algorithm and all the other algorithms defined in the previous phase. The ICBS algorithm [5] will probably be involved in the benchmark. The original implementation of ICBS is in `C#`, hence the programming language that will be adopted is probably `C#`.
- *Experimentation*: in this phase, we have to test our algorithms and their performance in a wide number of settings. We need a reliable comparison between our new approach and the other approaches. Finally, this phase includes the definition of some hand-crafted test environments because the current maps involved in MAPF benchmarks [13] are static.
- *Writing*: in this phase, we have to write down all the results gathered in the previous phases and the final document will be in the form of a scientific paper. To this end, we also need to produce clear and intuitive figures, using some LaTeX-oriented tools, to be included in the document.

A simple Gantt diagram of the entire research timeline is shown in Figure 2.

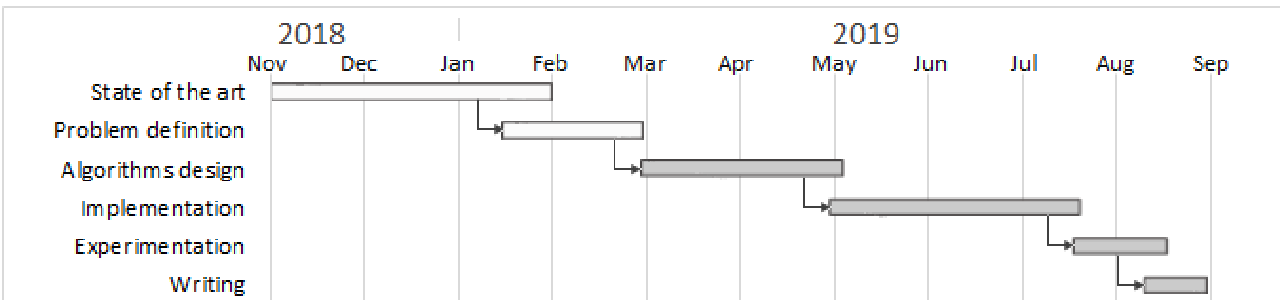

Figure 2: Gantt diagram of the task, starting from November 2018 to September 2019

.

The quality of the research output is strictly related to the performance of our new algorithm. Since all the approaches guarantee optimality, the main metric used to evaluate the outputs of the research will be the average runtime execution of the algorithm. Other secondary metrics will be considered, like the behavior of the algorithm in different types of environments [13].

References

[1] Agarwal, P. K., Kumar, N., Sintos, S., and Suri, S. Computing shortest paths in the plane with removable obstacles. In *Proceedings of the Scandinavian Symposium and Workshops on Algorithm Theory* (2018), pp. 5:1–5:15.

[2] Andreychuk, A., and Yakovlev, K. Path finding for the coalition of co-operative agents acting in the environment with destructible obstacles. In *Proceedings of the International Conference on Interactive Collaborative Robotics* (2018), pp. 13–22.

[3] Banfi, J., Basilico, N. G., and Amigoni, F. Intractability of time-optimal multirobot path planning on 2D grid graphs with holes. *IEEE Robotics and Automation Letters 2*, 4 (2017), 1941–1947.

[4] Bellusci, M. State of the art on: Multi agent path finding, 2019.

[5] Boyarski, E., Felner, A., Stern, R., Sharon, G., Tolpin, D., Betzalel, O., and Shimony, S. E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In *Proceedings of the International Joint Conference on Artificial Intelligence* (2015), pp. 740–746.

[6] Ferguson, D., Darms, M., Urmson, C., and Kolski, S. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium* (2008), pp. 1149–1154.

[7] Fulgenzi, C., Spalanzani, A., and Laugier, C. Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid. In *Proceedings of the IEEE International Conference on Robotics and Automation* (2007), pp. 1610–1616.

[8] Kruszewski, P. Navigation challenges in massively destructible worlds. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment* (2007), pp. 108–109.

[9] Levine, J., Bates Congdon, C., Ebner, M., Kendall, G., Lucas, S. M., Miikkulainen, R., Schaul, T., and Thompson, T. General video game playing, 2013.

[10] Majerech, O. Solving algorithms for multi-agent path planning with dynamic obstacles, 2017.

[11] Nash, A., Daniel, K., Koenig, S., and Felner, A. Thetaˆ*: Any-angle path planning on grids. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2007), pp. 1177–1183.

[12] Sharon, G., Stern, R., Felner, A., and Sturtevant, N. Conflict-based search for optimal multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2012), pp. 563–569.

[13] Sturtevant, N. R. Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games 4*, 2 (2012), 144–148.

[14] Wurman, P. R., D'Andrea, R., and Mountz, M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2007), pp. 1752–1759.

[15] Yu, J., and LaValle, S. M. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2013), pp. 1443–1449.