Measuring the Vulnerability of a Multi-Agent Pathfinding Solution

Rotem Yoeli,[1] **Dor Atzmon,**[1] **Roni Stern**[1]
[1]Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
rotem.yoeli@gmail.com,{dorat,sternron}@post.bgu.ac.il

In the future, it is expected that autonomous agents will take a major role in our life. They will be delivering goods, serving as a transportation solution, and maintaining surveillance and security. A fundamental task such agents will have to perform is path planning, where agents aim to reach their goals as fast as possible but without colliding with each other. This is known as the Multi-Agent Pathfinding (MAPF) problem, and has been studied extensively in recent years (Sharon et al. 2015; Felner et al. 2017; Ma et al. 2019, inter alia).

However, communication between the agents and to/from a centralized controller (if exists) can be vulnerable to security breaches. For example, an unencrypted Wi-Fi used with a drone allows any individual to connect and hack the drone. We explore such a scenario, where one of the mobile agents was hacked and a malicious entity gained access to it. This malicious entity learns about the planned routes, and gains the ability to build a new routes and introduce a number of unplanned actions. In this work, we introduce the Maximum Damage Route (MDR) problem, which is the problem of assessing the maximum damage that the malicious entity can cause in this context. We formulate MDR as a heuristic search problem, and propose an A*-based algorithm that solves it optimally. Given a limited budget for protective measures, we show how solving MDR problems can help to choose which agent to protect.

Background and Problem Definition

A MAPF problem is defined by a tuple $\langle G, k, s, t\rangle$ where $G = (V, E)$ is an undirected graph, k is the number of agents, $s : [1, \ldots, k] \to V$ maps an agent to its start node, and $t : [1, \ldots, k] \to V$ maps an agent to its target (goal) node. A *plan* for an agent i is a sequence of actions such that if agent i is in $s(i)$ and executes these actions then it will end up in $t(i)$. There is a *conflict* between plan π_i for agent i and π_j for agent j if according to these plans the agents will occupy the same node or the same edge at the same time. A *joint plan* for a set of agents is a set of plans, one for each agent. A solution to MAPF problem is a *joint plan* in which no pair of plans has a conflict. The *makespan* of a solution $\Pi = \{\pi_1, \ldots, \pi_k\}$, denoted $\mathrm{M}(\Pi)$ is the max over the length of its constituent plans.

In our setting, there exists a malicious entity that takes control of one of the agents. It can direct that agent to perform actions that are different from those planned for it by the central controller. We refer to that agent as the *compromised agent* and refer to an action that differs from the planned action as an *abnormal action*. Without loss of generality, we assume that the compromised agent is agent 1.

We make the following assumptions about how the agents are controlled, the knowledge of the malicious entity, and what is can make the compromised agents do. The joint plan the agents are following is computed by a central controller. The controller always verifies that the plans it generates do not conflict. All non-compromised agents follow the joint plan unless a collision is about to occur in the next time step. In such a case, the agent avoids the collision by staying in its current location. The joint plan is then modified to avoid future conflicts by adjusting the plan of that agent and, if needed, the plans of other agents. The compromised agent may perform at most B of abnormal actions, where B is a parameter. We limit the number of abnormal actions a compromised agent may perform to reflect the malicious entity's desire to hide its intentions and its hold on the compromised agent. In addition, we limited abnormal actions to only deviate by at most 90 degrees from the planned action.

After the compromised agent performs an abnormal action, it immediately receives a new non-conflicting plan from the central controller. For a joint plan Π and an abnormal action a, we denote by $a(\Pi)$ the joint plan computed by the central controller after performing a. Note that $a(\Pi)$ and Π are the same for the first $t-1$ actions. An *interruption plan* $E = (e_1, \ldots e_b)$ is a sequence of abnormal actions ordered from earliest to latest. For a joint plan Π and an interruption plan E, we use the term *abnormal execution*, denoted $\textsc{Sym}(\Pi, E)$, to denote the sequences of actions the agents execute given that the abnormal actions in E are executed.

We assume the malicious entity can compute the abnormal execution for any joint plan Π and interruption plan E. The malicious entity's objective is to maximize the makespan of the abnormal execution.

Definition 1 (MDR). *An MDR problem is defined by the tuple* $P_{MDR} = \langle \Pi, B\rangle$ *where* Π *is a solution to a MAPF problem and* B *is the budget of allowed abnormal actions. A solution to* P_{MDR} *is an interruption plan with at most*

	Agents					
Runtime	5	6	7	8	9	10
B =1	0.19	0.22	0.24	0.35	0.36	0.36
B =2	1.04	1.46	1.57	1.89	2.16	2.27
B =3	12.54	20.85	21.97	25.13	28.48	31.03
Damage	5	6	7	8	9	10
B =1	1.12	1.14	1.04	0.95	0.89	0.86
B =2	2.50	2.51	2.37	2.28	2.19	2.12
B =3	4.02	4.08	3.85	3.58	3.51	3.40

Table 1: Results of MDR experiments on room maps.

B abnormal actions. An optimal solution is an interruption plan E such that for any other solution E' it holds that $\mathrm{M}(E) \geq \mathrm{M}(E')$.

MDR as a Search Problem

We solve MDR by formulating it as a search problem, and use the well-known A* algorithm (Hart, Nilsson, and Raphael 1968) to solve it. A *state* consists of the current time step t, the current joint plan Π, and the remaining number of abnormal actions $b \leq B$. The actions applicable in a state correspond to the compromised agent performing in the next time step either its planned action or an abnormal action. A *goal state* is a state in which all agents reached their goals or when the number of allowed abnormal actions is zero. The objective is to find a goal state with maximum cost.

To to solve MDR, we adapt A* as follows. Each state $n = \langle t, \Pi, b \rangle$ is associated with two values, $cost(n)$ and $f(n)$. $cost(n)$ is the makespan of the Π. $f(n)$ is equal to $cost(n)$ if n is a goal state. Otherwise, it is an *upper bound* to the cost of any goal state reachable from n. In particular, for a non-goal state $n = \langle t, \Pi, b \rangle$, we used the following heuristic: $f(n) = \mathrm{M}(\Pi) + b \cdot k$, where k is the number of agents. $f(n)$ is indeed an upper bound of all goals under n because an abnormal action can be repair by having all non-delayed agents wait one time step (Atzmon et al. 2018). In every iteration, A* for MDR expands the state n with the *highest* $f(n)$ value in the open list. The following is easy to prove.

Theorem 1. *When A* for MDR expands a goal state, it is guaranteed to have found an optimal solution.*

Experimental Results

We implemented A* for MDR and evaluated it experimentally on a room-like grid. Agents' starts and goals were selected from a pre-defined set of start vertices and goal vertices, respectively, such that the start and goal of each agent is adjacent to a start and goal of another agent, respectively. Next, we created 30 possible sets of initial routes of the agents by running prioritized planning with random priorities (Silver 2005). For every generated set of initial routes, and each agent i, we run our MDR algorithm assuming agent i is the compromised agent, having a budget B of 1, 2, and 3. We measured the runtime of our MDR algorithm and *damage*, which is the difference between the makespan of the initial plan and the abnormal execution.

Table 1 shows the average runtime in seconds (top) and the average damage (bottom) for different number of agents (columns) and different values of B (rows). As expected, larger B allows the compromised agent to cause more damage. Increasing B as well as adding more agents increase runtime, but increasing B has a much stronger impact. This is expected, since the size of the search space is exponential in B but not in k.

Conclusion and Future Work

We introduced the MDR problem, which is the problem of estimating the maximal damage a compromised agent may cause to the execution of a MAPF solution. We proposed an A*-based solution for this problem and demonstrate its feasibility experimentally. The high-level objective of this project is to protect autonomous agents against malicious entities. A possible application of this work is to identify the *weakest link* in a set of agents, i.e., the agent that may inflict the maximal damage if compromised. This can help focus protective measures such as monitoring various hardening techniques on these agents, making it harder for a malicious entity to control them (Wesson and Humphreys 2013). For example, one use MDR to compute the number of agents that must be protected in order to limit the maximum allowed damage, assuming that at most one agent can be compromised. Developing such preventive measures is a promising direction for future work.

References

Atzmon, D.; Stern, R.; Felner, A.; Wagner, G.; Barták, R.; and Zhou, N. 2018. Robust multi-agent path finding. In *International Symposium on Combinatorial Search (SOCS)*, 2–9.

Felner, A.; Stern, R.; Shimony, S. E.; Boyarski, E.; Goldenberg, M.; Sharon, G.; Sturtevant, N. R.; Wagner, G.; and Surynek, P. 2017. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In *the International Symposium on Combinatorial Search (SoCS)*, 29–37.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* SSC-4(2):100–107.

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 Conference on Artificial Intelligence*.

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

Silver, D. 2005. Cooperative pathfinding. *AIIDE* 1:117–122.

Wesson, K., and Humphreys, T. 2013. Unhackable drones: the challenges of securely integrating unmanned aircraft into the national airspace.