Investigation into Braess' Paradox and Crowd Management in Multi-Agent Pathfinding

Author
David Batonda,
Monash University,
david.batonda@monash.com

Primary Supervisor
David Dowe,
Monash University
david.dowe@monash.com

Co-Supervisor
Daniel Harabor,
Monash University,
daniel.harabor@monash.com

Abstract

This research paper explores the application of sub-optimal solvers in Multi-Agent Path-finding (MAPF) for efficient evacuation planning and crisis management. By strategically adding pillars to the environment, sub-optimal solvers exhibit enhanced capabilities in handling a larger number of agents within tight time constraints, leading to improved path-finding efficiency in complex and crowded spaces. Moreover, the study delves into the intriguing relationship between Braess' Paradox and optimal solvers in transportation networks. While closing and blocking additional paths have shown advantages in specific scenarios, their benefits diminish when dealing with randomly placed start and goal points. Understanding these dynamics is vital for developing navigation strategies and optimizing real-world transportation systems effectively.

1 Introduction

Sequential decision-making problems in robotics, logistics, transportation, and computer games often use modified versions of the optimal A* algorithm. They maintain an OPEN list of nodes to find minimum-cost paths from the initial node to a goal. A crucial role is played by admissible heuristics that estimate distances accurately. Multi-agent path-finding is another important area, involving collision-free paths for multiple agents in shared environments. Specialised algorithms and heuristics are developed to efficiently coordinate agents and optimize navigation in complex spaces, improving overall efficiency and reducing congestion.

2 Crowd Management on Multi-agent Path-finding

Through the strategic addition of pillars to the environment, sub-optimal solvers have demonstrated their capability to handle a larger number of agents within a 60-second time frame. Intentionally inputting obstacles optimizes path-finding efficiency, particularly in complex and crowded spaces [1], resulting in improved evacuation planning and crisis management. Take the orz900d[2] map for instance, there are several regions of congestion as pointed to in Figure 1.

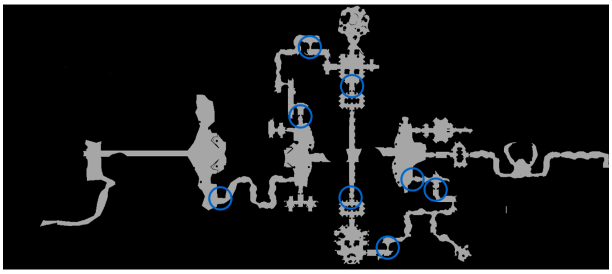

Figure 1: High Congestion regions on orz900d map circled in blue

By simply adding a pillar in one of these regions, the sub-optimal algorithms can find solutions for considerably more agents (for the given 1 minute time limit). Here, we added a pillar (as impassable regions on the map at (1064, 448) & (1065, 448)).

Figure 2: orz900d zoomed onto the rightmost congestion region from Fig .1. Pillar added is marked with red

Doing so enabled the LaCAM[3] algorithm to solve for 78 agents more, which is roughly 3.8% agents than what it could have previously solved.

	Maximum #agents LaCAM solves (in 1 minute)	Solution Cost for 2043 Agents (sum-of-costs)
Without Pillar	2043	4044939
With Pillar	2121	4058575

However, it is crucial to also highlight the trade-off between the planning cost. Whilst adding the pillar enabled LaCAM to solve for more agents, it also increased the cost of the solution

By leveraging these added physical constraints, MAPF algorithms can have lowered computation times, and hence, able to solve for more agents.

3 Braess' Paradox and optimal solvers

Braess' Paradox and optimal solvers, such as BCP-MAPF[4] and CBSH2-RTC[5] offer fascinating insights into transportation networks [1][6]. Closing and blocking additional paths appeared to aid optimal solvers in specific scenarios, but this advantage diminishes when dealing with randomly placed start and goal points. In many cases, blocking the additional road did not improve the solvers' ability to find optimal paths. Understanding these nuances is crucial for developing efficient navigation strategies and optimizing real-world transportation systems.

Here is an example where Braess' paradox takes place on roads

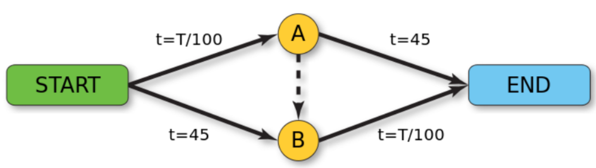

Figure 3: Braess' paradox[7]

Realistically, the fastest path is `Start-A-B-End`, however, as more cars/agents take this path the congestion forms on the `Start-A` road. And if all cars take this seemingly optimised route the total time would be slower than going `Start-A-End` or `Start-B-End`. Hence, blocking the path would ease the congestion.

Using this concept, a MAPF map with the `A-B` route open may look something like this:

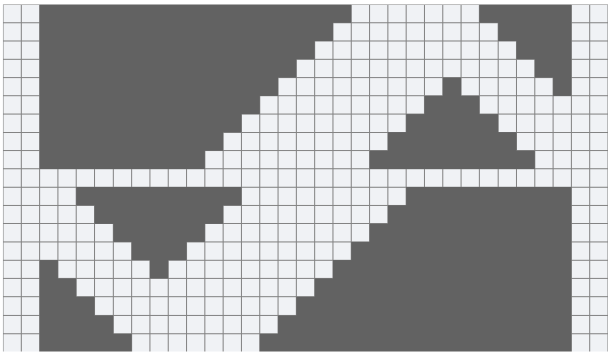

Figure 4: Sample map with with Braess' paradox and road opened[8]

And the same map with the `A-B` route closed would look like this:

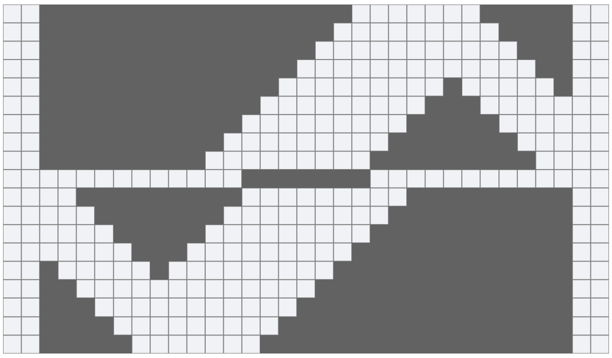

Figure 5: Sample map with with Braess' paradox and road closed[8]

There are three distinct scenario types when dealing with this sort of set up:

- Scenarios where all start points are on the left-hand side (anywhere on the first two columns) and all corresponding end points are on the right-hand side (anywhere on the last two columns), or vice-versa. We will call this scenario type **A**.

- Scenarios where start and end points are on either edge of the map (the first two columns or the last two columns). That is, start/end points can be on the left and/or right-hand side. This resulted in agents travelling from left to right and vice-versa. We will call this scenario type **B**.

- Scenarios where start/end points were placed randomly across the entire map without restrictions (besides for the impassable regions). We will call this scenario type **C**.

After experimenting 3 random scenarios from [8] of each scenario type and averaging the number of agents the optimal algorithms could solve for, the below table was formed:

	Random Scenario A	Random Scenario B	Random Scenario C
BCP-MAPF (A-B opened)	15.7	26.3	49.7
BCP-MAPF (A-B closed)	23.7	32.3	27.7
CBSH2-RTC (A-B opened)	10.3	10.3	38.7
CBSH2-RTC (A-B closed)	19.3	25	24

For Scenario types A & B, where start/end points only appear outside the main routes (on the map's edges), the optimal solvers were able to solve for significantly more agents. However, they seem to perform worse otherwise. Regarding type A & B, when the `A-B` route is open, the optimal path for every agent involves going through `Start-A-B-End`, which implies that all agents have a stricter time limit to reach their end goal. On the other hand, when the `A-B` route is closed, agents have a more relaxed time limit to reach their end goal, thus allowing the optimal MAPF algorithms to solve for more agents.

For scenario type C, closing the `A-B` results in agents on the top half requiring to go all the way around the map to reach the bottom, and vice-versa. This drastically increases their paths' costs, hence, the optimal MAPF algorithms solvers for fewer agents compared to when the `A-B` route is open.

Similar to the case of adding a pillar for crowd management strategies with LaCAM, closing the `A-B` often increases the solution cost. Take CBSH2-RTC with random_1a_scen.scen (from [4]) for instance:

	Maximum #agents CBSH2-RTC solves for `random-1a-scen.scen`	Solution Cost for 9 Agents (sum-of-costs)
CBSH2-RTC (A-B open)	9	364
CBSH2-RTC (A-B closed)	19	414

Whilst close the `A-B` route increase the amount of agents that can be solved optimally, the cost also dramatically increases, which highlights the trade-off of solving for more agents for a larger solution cost.

4 Levin Universal Search Algorithm

The potential to involve Universal Levin search in Multi-Agent Pathfinding (MAPF) through the exploration of Braess' Paradox and crowd management offers a promising avenue for enhancing pathfinding efficiency and optimizing navigation strategies. Universal Levin search, known for its strong theoretical foundations in solving combinatorial optimization problems, can be adapted to tackle MAPF scenarios where agents need to navigate through crowded environments with dynamically changing road conditions[9]. By harnessing insights from Braess' Paradox and crowd management strategies, Universal Levin search can intelligently adapt its exploration and search processes, leading to improved pathfinding solutions even in complex and time-sensitive situations. Integrating these concepts into MAPF algorithms has the potential to pave the way for more robust and efficient navigation strategies in real-world transportation systems and crisis management scenarios. [3][4].

5 Conclusion and Future Work

In summary, this study underscores the significance of modified A* algorithms and admissible heuristics in diverse applications, including robotics, logistics, and computer games. Specialized algorithms for multi-agent pathfinding have proven effective in coordinating agents within complex environments, optimizing navigation, and alleviating congestion. The strategic incorporation of pillars has showcased the potential of sub-optimal solvers, leading to improved evacuation planning and crisis management. However, it is essential to recognize the trade-off between solution cost and the number of agents handled by MAPF algorithms, requiring careful consideration for crowd management strategies and road blocking in real-world transportation systems.

In future work, exploring the integration of Universal Levin search in Multi-Agent Pathfinding (MAPF) using insights from Braess' Paradox and crowd management strategies shows promise in enhancing pathfinding efficiency and adaptability in complex and time-sensitive scenarios

6 Refrences

References

[1] Y. Zhao et al., "Optimal layout design of obstacles for panic evacuation using differential evolution," Physica A: Statistical Mechanics and its Applications, vol. 465, pp. 175–194, Jan. 2017, doi: https://doi.org/10.1016/j.physa.2016.08.021.

[2] "2D grid world maps," Movingai.com, 2021. https://movingai.com/benchmarks/mapf/index.html (accessed Jul. 22, 2023).

[3] K. Okumura, "LaCAM: Search-Based Algorithm for Quick Multi-Agent Pathfinding," Nov. 2022, doi: https://doi.org/10.48550/arxiv.2211.13432.

[4] E. Lam, Pierre Le Bodic, D. Harabor, and P. J. Stuckey, "Branch-and-cut-and-price for multi-agent path finding," vol. 144, pp. 105809–105809, Aug. 2022, doi: https://doi.org/10.1016/j.cor.2022.105809.

[5] J. Li, D. Harabor, P. J. Stuckey, H. Ma, G. Gange, and S. Koenig, "Pairwise symmetry reasoning for multi-agent path finding search," Artificial Intelligence, vol. 301, p. 103574, Dec. 2021, doi: https://doi.org/10.1016/j.artint.2021.103574.

[6] D. Braess, A. Nagurney, and T. Wakolbinger, "On a Paradox of Traffic Planning," Transportation Science, vol. 39, no. 4, pp. 446–450, Nov. 2005, doi: https://doi.org/10.1287/trsc.1050.0127.

[7] Wikipedia Contributors, "Braess's paradox," Wikipedia, Jan. 07, 2022. https://en.wikipedia.org/wiki/Braess

[8] D. Batonda, "Braess-MAPF/random_1a_scen.scen at main · dtonda8/Braess-MAPF," GitHub, Jul. 23, 2023. https://github.com/dtonda8/Braess-MAPF/blob/main/random_1a_scen.scen (accessed Jul. 22, 2023).

[9] M. Hutter, "THE FASTEST AND SHORTEST ALGORITHM FOR ALL WELL-DEFINED PROBLEMS," The Fastest and Shortest Algorithm for All Well-Defined Problems, vol. 13, no. 03, pp. 431–443, Jun. 2002, doi: https://doi.org/10.1142/s0129054102001199.