Second Smallest Eigenvalue and Empirical Hardness of Grid-based Multi-Agent Pathfinding Problem

Jingyao Ren[1], Eric Ewing[2], T. K. Satish Kumar[1], Sven Koenig[1], Nora Ayanian[2]

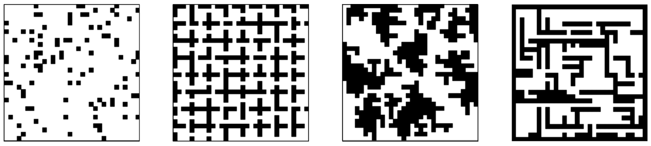
***Abstract*— We present an empirical study of the relationship between map connectivity and the empirical hardness of the multi-agent pathfinding (MAPF) problem. By analyzing the second smallest eigenvalue (commonly known as λ_2) of the normalized Laplacian matrix of different maps, our initial study indicates that maps with smaller λ_2 tend to create more challenging instances when agents are generated uniformly randomly. Although λ_2 does not exhibit a strict monotonic correlation with empirical hardness, we believe it can be considered as a simple and effective way of gaining deeper understanding on what makes a MAPF instance hard to solve.**

I. INTRODUCTION

Multi-agent pathfinding (MAPF) is the problem of finding collision-free paths for a team of agents on a map from a set of start positions to a set of goal positions [1]. Given an undirected map, an optimal MAPF algorithm computes the minimum path cost for all the agents such that no two agents occupy the same location or traverse the same edge at an identical time step. Although solving MAPF optimally is proven to be NP-Hard [2], many real-world MAPF instances can be solved within a reasonable time. While optimal MAPF algorithms can solve some instances with hundreds of agents, they can also struggle on instances with only a small number of agents [3], [4].

We are interested in understanding what features of MAPF instances make them hard to be solved optimally. We are also interested in finding an effective way of comparing the hardness of different maps when randomly generating MAPF instances on them. For example, when using uniform random sampling to generate agents and goals on two given maps, we seek to predict which map will have harder instances on average. This area of research is known as *empirical hardness*, which focuses on identifying features that determine how hard individual instances will be for particular algorithms to solve them [5]. Here, we present an empirical study that aims to elucidate the correlation between map connectivity and the empirical hardness of the multi-agent pathfinding problem.

There are two major components of a MAPF instance: the map topology and distribution of the agents. In this study, we focus on 2D grid-based MAPF problems, where a map can be viewed as a 4-way connected graph $G(V, E)$. A vertex $v_i \in V$ indicates a free map cell and an edge $e_{ij} \in E$ means v_i is connected to another vertex v_j. Our empirical results shows that the second smallest eigenvalue (henceforth referred to as λ_2) of the normalized Laplacian matrix of $G(V, E)$ is correlated with the empirical hardness of the MAPF instances when agents and goals are generated using uniform random sampling. The smaller λ_2 is, the less connected the graph will be, which tends to lead to harder instances. The most straightforward small λ_2 example is a map with many narrow corridors. Previous research has shown that various optimal algorithms have difficulty with such maps even with a small number of agents [3], which could be caused by the over-congestion and conflicts that narrow corridors bring.



random-32-32-10 214.1×10^{-5}

room-32-32-4 89.7×10^{-5}

frac-32-32-4 46.9×10^{-5}

maze-32-32-5 6.5×10^{-5}

Fig. 1. Example maps and their λ_2.

II. NORMALIZED LAPLACIAN MATRIX AND ITS EIGENVALUE

In spectral graph theory, the normalized Laplacian matrix $\bar{L}$ of a graph is defined by:

$$\begin{aligned} L &= D - A \\ \bar{L} &= D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2} \end{aligned} \tag{1}$$

where the D is the diagonal degree matrix and A is the adjacency matrix. The second smallest eigenvalue of the normalized Laplacian $\bar{L}$ defines the algebraic connectivity of the graph, describing how well the graph is connected (for more detail refer to [6]).

Figure 1 shows the value of λ_2 for several maps with different connectivity. The difference in λ_2 between a well-connected map, `random-32-32-10` on the far left and a less connected `maze-32-32-5` on the far right is significant. Generally, a relatively small λ_2 indicates the graph is poorly connected, whereas a large λ_2 implies strong connectivity. This suggests that λ_2 can be used as a quantitative method for characterizing the impacts of a map's features, such as narrow corridors, on the overall map connectivity. It may also enable a method for categorizing maps based on their connectivity rather than where the maps are originally collected from (e.g., game and city maps).

[1] University of Southern California, Department of Computer Science and Institute of Information Science, Los Angeles, CA 90089. `jingyaor@usc.edu`

[2] Brown University, Department of Computer Science, Providence, RI 02906

III. EXPERIMENT

As an initial proof of concept to show that the λ_2 value of a map has some correlation with the difficulty, or runtime, of MAPF instances on that map, we randomly generated MAPF instances on multiple maps with varying λ_2 values and compared runtimes.

A. *Simulation Setup*

To keep the test dataset diverse, we included maps from multiple data sources. Firstly, we have included all 32×32 maps (6 in total) from the MAPF benchmark dataset [7]. We have also included a fractal map generator based on a diffusion-limited aggregation method [4]. We slightly modified the generation rule of the fractal method such that it could generate maps of different styles (e.g., cave-like `frac-32-32-4` and maze-like `maze-32-32-5` in Fig. 1). We generated 30 fully-connected maps of size 32×32 using this approach.

When generating instances, we ensured that all the maps have the same agent-to-freespace ratio, where $r = \frac{\#agents}{\#free\ cells}$. The number of agents for our test ranged from 14 to 22. For each map, we generated 100 instances using uniform random sampling to determine the start and goal locations of agents. The feasibility of the generated instances was validated by using a sub-optimal MAPF algorithm ECBS with a relaxed bound ($w = 1.6$) [8].

Simulations were conducted on a PC with Ryzen 3950x CPU and 64GB RAM, and the runtime limit was set to 900 seconds.

For our initial analysis, we chose to use only one MAPF solver. We selected LazyCBS [9] as the MAPF algorithm, since it has been shown to be quite powerful according to previous benchmark analysis [4]. There are other optimal MAPF algorithms, such as BCP [10], CBSH [11] and its variants [12], which perform differently than LazyCBS, however, there was not enough time to do a thorough analysis of these solvers for this late breaking report. We plan to include these algorithms in a future publication.

B. *Simulation Result & Discussion*

The simulation results in Fig. 2 illustrate the relationship between the logarithm of average runtime and λ_2 of different maps. Given that λ_2 is not the only factor influencing empirical hardness, we are not surprised to see that the average runtime and λ_2 do not exhibit a strict monotonic correlation. Nonetheless, we still find our results to be convincing of the relationship between λ_2 and runtime and made several interesting observations. First, hard instances often appear on maps with smaller λ_2 (top left corner), whereas maps with larger λ_2 can be considerably easy (bottom right corner). Second, maps with smaller λ_2 could still have relatively easy instances. One possible reason for this could be the lack of constraints on the number of obstacles during map generation (i.e., imagine maps with a limited number of obstacles and small λ_2. Agents will have more free space to maneuver and avoid conflicts). Another reason might be the effectiveness of narrow corridors on 2D grid-map. For instance, increasing the width of a narrow corridor from 1-cell to 2-cell width will only change λ_2 slightly, but the corridors become less effective to create enough contested regions, thus the empirical hardness could drastically shift from hard to easy.

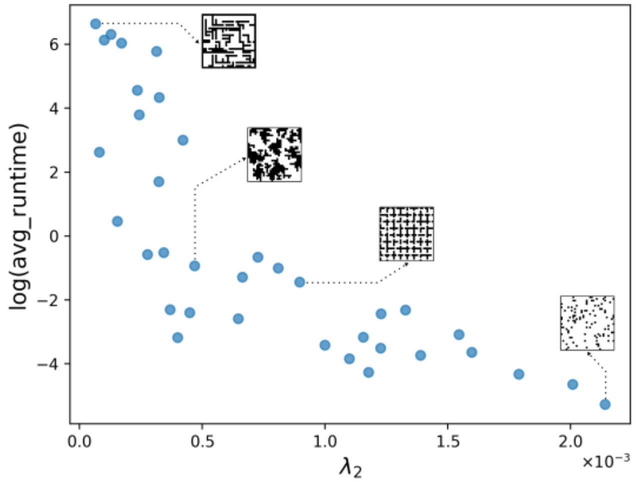


Fig. 2. Simulation result of λ_2 and the logarithm of average runtime for different maps. We have annotated the result with the maps shown in Fig. 1 to help better understand the correlation between map topology and the according empirical hardness.

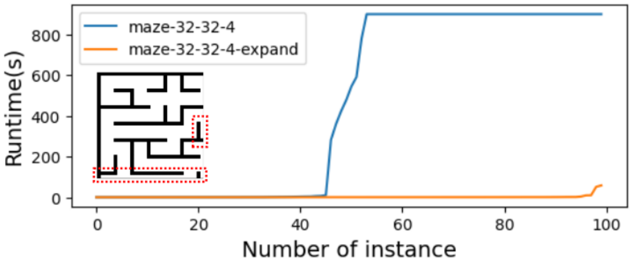


Fig. 3. Sorted runtime for `maze-32-32-4` and its expanded version by relaxing the width of the narrow corridor in red boxes from 1-cell to 2-cell.

To investigate this phenomenon, we manually changed the connectivity of maps without affecting the number of obstacles. This process can be viewed as a reconfiguration of the map layout. More specifically, as shown in Fig. 3, we expand some of the narrow corridors in `maze-32-32-4` from 1-cell to 2-cell width and observed that λ_2 changed from 10.1×10^{-5} to 15.4×10^{-5}. Although the change in λ_2 is small, there is a significant change of the empirical hardness, where instances on `maze-32-32-4-expand` are much easier. This demonstrates that even maps with small λ_2 can have easy instances. It also suggests that a 1-cell-width corridor is more likely to create contested regions and cause conflicts between agents, thus slowing down the algorithms (especially for conflict-based algorithms). These contested regions are significantly mitigated when increasing the corridor width, making the instances easier; in the meantime λ_2 exhibits minor change. This leads to another potential future work: using λ_2 as one of the metrics when evaluating different map layouts or reconfiguration strategies such that the instances tend to be easier than before.

In summary, even though λ_2 does not exhibit a strict monotonic correlation with empirical hardness, it still shows notable effectiveness, especially for very challenging instances associated with small λ_2. Considering the easy and straightforward nature of computing λ_2 and comparing them across different maps, we believe it can be a reasonably effective metric for analyzing MAPF empirical hardness. Future research will focus on combining other instance features such as obstacle density, number of agents, and a biased agent distribution with λ_2 and finding a more tightly correlated relationship with empirical hardness.

REFERENCES

[1] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzmon, L. Cohen, T. Kumar *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, no. 1, 2019, pp. 151–158.

[2] J. Yu and S. LaValle, "Structure and intractability of optimal multi-robot path planning on graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 27, no. 1, 2013, pp. 1443–1449.

[3] J. Ren, V. Sathiyanarayanan, E. Ewing, B. Senbaslar, and N. Ayanian, "Mapfast: A deep algorithm selector for multi agent path finding using shortest path embeddings," *arXiv preprint arXiv:2102.12461*, 2021.

[4] E. Ewing, J. Ren, D. Kansara, V. Sathiyanarayanan, and N. Ayanian, "Betweenness centrality in multi-agent path finding," in *International Conference on Autonomous Agents and Multiagent Systems*, 2022.

[5] K. Leyton-Brown, E. Nudelman, and Y. Shoham, "Empirical hardness models: Methodology and a case study on combinatorial auctions," *Journal of the ACM (JACM)*, vol. 56, no. 4, pp. 1–52, 2009.

[6] T. Vidick. (2018) Introduction to spectral graph theory. [Online]. Available: http://users.cms.caltech.edu/ vidick/notes/CMS139/spectral.pdf

[7] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, E. Boyarski, and R. Bartak, "Multi-agent pathfinding: Definitions, variants, and benchmarks," *Symposium on Combinatorial Search (SoCS)*, pp. 151–158, 2019.

[8] M. Barer, G. Sharon, R. Stern, and A. Felner, "Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 5, no. 1, 2014, pp. 19–27.

[9] G. Gange, D. Harabor, and P. J. Stuckey, "Lazy cbs: implicit conflict-based search using lazy clause generation," in *Proceedings of the international conference on automated planning and scheduling*, vol. 29, 2019, pp. 155–162.

[10] E. Lam, P. Le Bodic, D. Harabor, and P. J. Stuckey, "Branch-and-cut-and-price for multi-agent path finding," *Computers & Operations Research*, vol. 144, p. 105809, 2022.

[11] A. Felner, J. Li, E. Boyarski, H. Ma, L. Cohen, T. S. Kumar, and S. Koenig, "Adding heuristics to conflict-based search for multi-agent path finding," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 28, 2018, pp. 83–87.

[12] J. Li, A. Felner, E. Boyarski, H. Ma, and S. Koenig, "Improved heuristics for multi-agent path finding with conflict-based search." in *IJCAI*, vol. 2019, 2019, pp. 442–449.