Map Connectivity 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]
[1]University of Southern California, Los Angeles, CA 90007, USA
[2]Brown University, Providence, RI 02912, USA
jingyaor@usc.edu, eric_ewing@brown.edu, tkskwork@gmail.com, skoenig@usc.edu, nora_ayanian@brown.edu

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. Additionally, we introduce a map generator based on Quality Diversity (QD) that is capable of producing maps with specified λ_2 ranges, offering a possible way for generating challenging MAPF instances. Despite the absence of a strict monotonic correlation with λ_2 and the empirical hardness of MAPF, this study serves as a valuable initial investigation for gaining a deeper understanding of what makes a MAPF instance hard to solve.

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 (Stern et al. 2019a). 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 (Yu and LaValle 2013), many real-world MAPF instances can be solved optimally 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 (Ren et al. 2021; Ewing et al. 2022).

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 to compare 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 (Leyton-Brown, Nudelman, and Shoham 2009). 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.

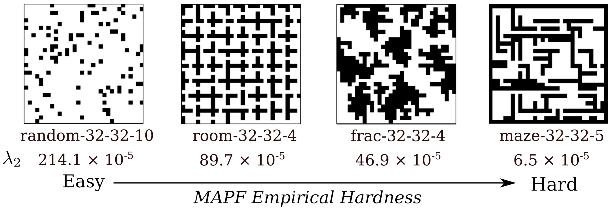


Figure 1: Example maps and their λ_2

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-connected graph $G(V, E)$. In spectral graph theory, the second smallest eigenvalue of the normalized Laplacian matrix (henceforth referred to as λ_2) of $G(V, E)$ serves as an algebraic measurement of graph connectivity. 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.

In this paper, we present empirical results that show the λ_2 of $G(V, E)$ is correlated with the empirical hardness of MAPF instances generated using uniform random sampling for agents and goals. While a smaller λ_2 value does not consistently yield challenging instances, instances characterized as difficult tend to occur more frequently when λ_2 is small. The most straightforward small λ_2 example is a map with many narrow corridors. Previous research has shown that various optimal MAPF algorithms have difficulty with such maps even with a small number of agents (Li et al. 2020; Ren et al. 2021), which could be caused by the over-congestion and conflicts that narrow corridors bring.

We also propose a map generator based on Quality Diversity (QD) (Mouret and Clune 2015) which provides the flexibility to generate maps within a desired range of λ_2. This provides an effective way to find maps that might be challenging for MAPF algorithms or generate benchmark dataset that covers a greater spectrum of connectivity.

Although λ_2 does not exhibit a strict monotonic correlation with empirical hardness, this study serves as a valuable initial study for understanding MAPF empirical hardness.

Preliminary

Normalized Laplacian and Cheeger's Inequality

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

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

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.

To get a better understanding of why λ_2 is related to the connectivity of graphs, we first introduce the *boundary* for a set of vertices $S \subset V$ of undirected graph $G(V, E)$:

$$\partial S = \{\{i, j\} \in E : i \in S, j \notin S\}. \tag{2}$$

The *conductance* of S is defined as:

$$\phi(S) = \frac{|\partial S|}{\min(d(S), d(V \backslash S))} \tag{3}$$

where $|\partial S|$ is the number of edges on the boundary and $d(S)$ denotes the number of edges with both endpoints (nodes) within S. $\phi(S)$ represents the ratio of the number of edges on the boundary of set S to the minimum of its internal and external edges.

The *conductance* of a graph $G(V, E)$ is subsequently defined as the smallest conductance over all cuts, where cuts refer to partitions of vertices:

$$\phi(G) = \min_{\emptyset \subsetneq S \subsetneq V} \phi(S) \tag{4}$$

The conductance represents how well-connected a graph is.

Theorem 1. (Cheeger's Inequality). Let λ_2 be the second smallest eigenvalue of the normalized Laplacian $\bar{L}$ of undirected graph $G(V, E)$, then:

$$\frac{\lambda_2}{2} \leq \phi(G) \leq \sqrt{2\lambda_2}. \tag{5}$$

Cheeger's inequality brings the graph connectivity and λ_2 together. This implies 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. Generally, a relatively small λ_2 indicates the graph is poorly connected, whereas a large λ_2 implies strong connectivity (for more detail please refer to (Vidick 2018)).

Conductance and MAPF Conflicts

Here we present an intuitive proof of how the maps with smaller conductance are more likely to generate more conflicts for MAPF instances. Consider a simple dumbbell graph G_d shown in Figure 2(a), where two partitions are only connected with a single edge. G_d is derived from a 4-connected map, thus the number of edges is roughly proportional to the number of vertices in its subgraphs. The size of the circle indicates different number of edges within the partition. Let's also assume this partition S^* has the smallest conductance of G_d, thus we have $\phi(G_d) = \phi(S^*)$. Next, consider another partition S of G_d shown in Fig. 2(b), where the two partitions are connected by more edges; thus, $|\partial(S)| > 1$ and $\phi(S^*) < \phi(S)$. Another observation is that S^* is a more balanced partition than S in terms of the number of edges within the partition, and we further have:

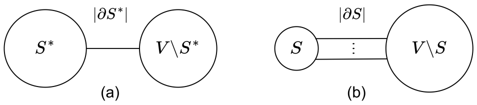


Figure 2: A dumbbell graph with two different partitions.

$$d(S^*) d(V \backslash S^*) > d(S) d(V \backslash S). \tag{6}$$

When uniformly and randomly sampling the start and goal locations on G_d, the shortest path will traverse a boundary edge only if the start and goal locations are on different sides of the boundary. The probability of the shortest path visiting a boundary edge of partition S is:

$$P(\partial S) = \frac{1}{|\partial S|} \frac{2d(S) d(V \backslash S)}{d(V)^2} \tag{7}$$

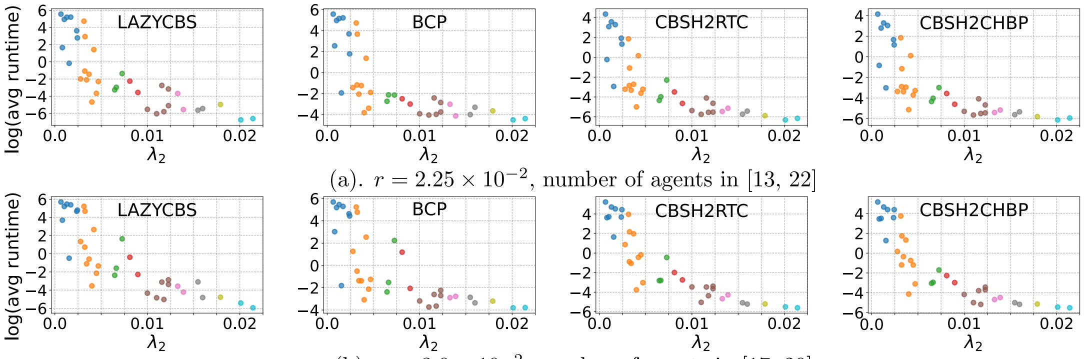
Given Eq. 6, we have:

$$\frac{2d(S^*) d(V \backslash S^*)}{d(V)^2} > \frac{2d(S) d(V \backslash S)}{d(V)^2} > \frac{1}{|\partial S|} \frac{2d(S) d(V \backslash S)}{d(V)^2}.$$

The left-hand side is $P(\partial S^*)$ since $|\partial S^*| = 1$ and the right-hand side is $P(\partial S)$. This indicates $P(\partial S^*) > P(\partial S)$. This inequality implies a higher likelihood of agents visiting the boundary edges of poorly connected cuts within the same graph, leading to increased potential conflicts, particularly in scenarios with more agents. Intuitively, one can think of these boundary edges as choke points that need to be traversed to get from one partition to the other. Relating this to the definition of $\phi(G)$ and Cheeger's inequality, we can loosely demonstrate that $P(\partial S^*) \propto \frac{1}{\phi(G)}$. This suggests that maps with smaller $\phi(G)$ or λ_2 may tend to exhibit more conflicts; thus MAPF instances on those maps are more likely to be challenging.

Quality Diversity Instance Generator

To investigate the relationship between λ_2 and the empirical hardness of maps, we developed a map generator that can produce maps with a given λ_2 value. The maps generated should provide as much diversity as possible in measures other than λ_2 to try and isolate the relationship between λ_2 and hardness. We used a Quality Diversity (QD) method based on the algorithm MAP-Elites (Mouret and Clune 2015). MAP-Elites is a search space illumination algorithm that seeks to find high quality solutions that are diverse along certain prescribed features. It maintains a container, which contains a set of potential solutions. New potential solutions are inserted into the container if they are the best solution found so far in a specific bin of feature space



(a). $r = 2.25 \times 10^{-2}$, number of agents in [13, 22]

(b). $r = 3.0 \times 10^{-2}$, number of agents in [17, 29]

Figure 3: Simulation results for the logarithm of average runtime and λ_2 for various maps, with distinct color coding denoting different ranges of λ_2.

with respect to some objective function. MAP-Elites typically utilizes evolutionary algorithms to alter existing solutions in the container and produce new potential solutions.

For our purposes, we sought to produce maps with specific λ_2 values that had diverse obstacle properties. We set the objective function to be the distance from a desired λ_2 value. We used features on the percentage of obstacles and the density of those obstacles (how many obstacles were adjacent to other obstacles). For each iteration, we took an existing map from the container and with equal probability we either “mutated” the map or “crossed” the map with another random map from the container. A mutation consisted of adding or removing up to five obstacles on the map uniformly at random. Crossing two maps involved randomly selecting regions of one map to add to the other map. We then checked for connectedness and added back the minimum number of vertices to reconnect any disconnected components. The new maps, either with randomly added or removed obstacles or the cross between existing maps, were then evaluated on closeness to the desired λ_2 value. If they were closer than any other map with similar features, they were kept and the other map in the container with those feature values was removed.

Previous work has explored generating MAPF maps with Quality Diversity algorithms to generate maps suitable for high-throughput online MAPF (Zhang et al. 2023). Zhang et al. trained a surrogate model DSAGE (Bhatt et al. 2022) that could help repair instances to meet constraints (e.g., number of shelves and connectivity) and predict the throughput of an instance. Our problem requires less sophisticated repair, since we have no hard constraint on the number of obstacles in an instance. Additionally, our objective function is relatively easy to compute, requiring only λ_2 for a generated map, and does not require any additional MAPF simulations. Our map generator is designed to create instances with a wide range of connectivity to use in the evaluation and benchmarking of MAPF algorithms.

Experiments

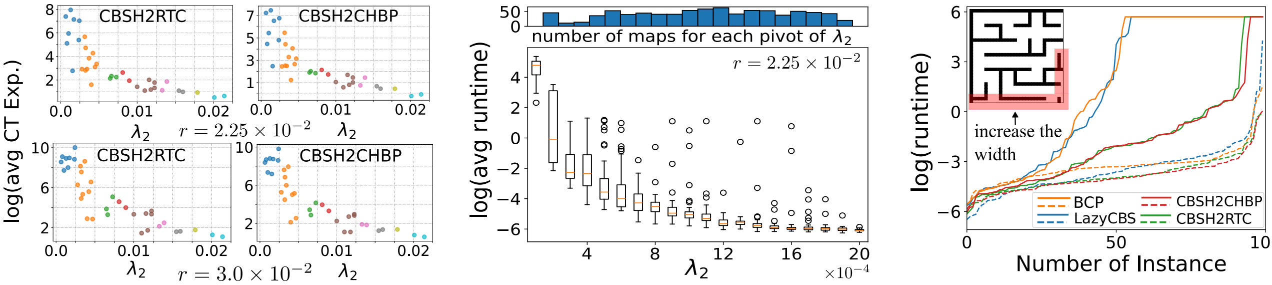
To thoroughly investigate the relationship between map connectivity and empirical hardness of MAPF, we have selected four different optimal MAPF algorithms which are proven to be quite powerful according to various benchmark analysis (Ewing et al. 2022; Shen et al. 2023a): **LazyCBS** (Gange, Harabor, and Stuckey 2019), **BCP** (Lam et al. 2022), **CBSH2-RTC** (Li et al. 2021) and **CBSH2-RTC-CHBP** (Shen et al. 2023b).

Simulation Setup. To ensure the diversity of our test dataset, we included maps from multiple data sources. Firstly, we have included all 32 × 32 maps (5 in total) from the MAPF benchmark dataset (Stern et al. 2019b). Additionally, apart from our QD map generator, we have also included a fractal map generator based on diffusion-limited aggregation method (Ewing et al. 2022). 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 Figure 1). We generated 31 fully-connected maps of size 32×32 using QD and fractal generator. The detailed map information and code are available at here[1].

When generating MAPF instances, we ensured that all the maps have the same agent-to-freespace ratio, where $r = \frac{\#agents}{\#free\ cells}$. This value is chosen based on our test such that the instances are neither excessively challenging nor overly easy so that we can still effectively compare the performance across different maps. 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 ($w = 1.6$) (Barer et al. 2014). Simulations were conducted on a PC with Ryzen 3950x CPU and 64GB RAM, with the runtime limit set to 300 seconds.

Experiment 1: Average Runtime and λ_2. As an initial proof of concept to show that the λ_2 value of a map has some correlation with the empirical hardness, or runtime, of MAPF instances on that map, we randomly generated MAPF instances on 36 maps with varying λ_2 values and compared runtimes. The simulation results in Figure 3 il-

[1] https://github.com/USC-ACTLab/MAPF-Lambda2



(a). (b). (c).

Figure 4: (a). The logarithm of average number of CT expansions and λ_2 and for different maps. (b). Boxplot for λ_2 and the logarithm of average runtime for the maps created by QD generator. (c). Sorted logarithm of runtime for `maze` and its expanded version `maze-e` by increasing the width of the narrow corridors in red boxes from 1-cell to 2-cell.

lustrate the relationship between the logarithm of average runtime and λ_2 of different maps. We have made several interesting observations on the results.

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). This pattern remains consistent across different algorithms and r settings. Additionally, it is noteworthy that CBSH2-based algorithms generally exhibit faster runtime than LazyCBS and BCP (notice the different scales on y-axis). Despite differences in absolute runtime, our results indicate within each algorithm, challenging instances happen more frequently on maps with smaller λ_2.

Second, maps with smaller λ_2 could still have relatively easy instances. 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. One possible reason might be the effect of narrow corridors on a 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 wider corridors are less likely to create enough contested regions, thus the empirical hardness could drastically shift from hard to easy. We further explore this in Experiment 4.

Experiment 2: Average Number of Constraint Tree (CT) Expansions and λ_2. Next, we illustrate the relationship of the average number of CT expansions and λ_2 for all instances on a map. For the CBSH2-based algorithms, the number of CT expansions is related to how many conflicts have been resolved during the searching process and reflects the hardness of instances (Gordon, Filmus, and Salzman 2021). From Figure 4(a), the trend for number of CT expansions is similar to the runtime trend. This correlation is believed to be caused by the poorly connected regions where conflicts are more likely to happen.

Experiment 3: More Tests Using QD Map Generator. Here we present additional tests on the runtime using CBSH2-RTC for maps generated by our QD map generator. We have generated 851 maps with a step size of 10^{-4} for λ_2 and the number of maps for each pivot is shown in Figure 4(b) ($r = 2.25 \times 10^{-2}$ and number of agents in $[19, 23]$). Different from fractal map generator, which lacks control over map connectivity, the QD map generator is able to generate maps with a well-distributed range of λ_2. This nice feature makes it a great choice to create benchmark dataset that requires a wider spectrum of map connectivity. Despite the fact that there are still many outliers in Figure 4(b), the relationship between λ_2 and empirical hardness still holds. This indicates that hard instances tend to happen around small λ_2, while large λ_2 generally result in easier instances.

Experiment 4: Expand the Width of Narrow Corridors. In Experiment 1, we mentioned that there are still many easy instances on maps with low λ_2. To investigate this phenomenon, we manually changed the connectivity of maps without affecting the number of obstacles. More specifically, we expand some of the narrow corridors (red boxes in Figure 4(c)) in a `maze` map 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 in empirical hardness, where instances on expanded version `maze-e` (shown with dashed lines) 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. We intend to develop a hybrid reasoning on both λ_2 and corridor width in future research.

Conclusion

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 simplicity and ease of comparing λ_2 across different maps, we believe it is a reasonably effective metric and great starting point for future research on MAPF empirical hardness. Another contribution of this work is the QD map generator which can generate maps with the desired range of λ_2. Future work includes developing MAPF instance generators for tunable empirical hardness and providing a more precise theoretical bound on the correlation of λ_2 and MAPF hardness.

Acknowledgments

The research at the University of Southern California was supported by the National Science Foundation (NSF) under grant numbers 1817189, 1837779, 1935712, 2121028, 2112533, and 2321786 as well as gifts from Amazon Robotics. The research at Brown University was supported by NSF grants 2311967 and 2330942, and an Amazon Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the U.S. government.

References

Barer, M.; Sharon, G.; Stern, R.; and Felner, A. 2014. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Proceedings of the International Symposium on Combinatorial Search*, volume 5, 19–27.

Bhatt, V.; Tjanaka, B.; Fontaine, M.; and Nikolaidis, S. 2022. Deep surrogate assisted generation of environments. *Advances in Neural Information Processing Systems*, 35: 37762–37777.

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

Gange, G.; Harabor, D.; and Stuckey, P. J. 2019. Lazy CBS: implicit conflict-based search using lazy clause generation. In *Proceedings of the international conference on automated planning and scheduling*, volume 29, 155–162.

Gordon, O.; Filmus, Y.; and Salzman, O. 2021. Revisiting the complexity analysis of conflict-based search: New computational techniques and improved bounds. In *Proceedings of the International Symposium on Combinatorial Search*, volume 12, 64–72.

Lam, E.; Le Bodic, P.; Harabor, D.; and Stuckey, P. J. 2022. Branch-and-cut-and-price for multi-agent path finding. *Computers & Operations Research*, 144: 105809.

Leyton-Brown, K.; Nudelman, E.; and Shoham, Y. 2009. Empirical hardness models: Methodology and a case study on combinatorial auctions. *Journal of the ACM (JACM)*, 56(4): 1–52.

Li, J.; Gange, G.; Harabor, D.; Stuckey, P. J.; Ma, H.; and Koenig, S. 2020. New techniques for pairwise symmetry breaking in multi-agent path finding. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, 193–201.

Li, J.; Harabor, D.; Stuckey, P. J.; Ma, H.; Gange, G.; and Koenig, S. 2021. Pairwise symmetry reasoning for multi-agent path finding search. *Artificial Intelligence*, 301: 103574.

Mouret, J.-B.; and Clune, J. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

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

Shen, B.; Chen, Z.; Cheema, M. A.; Harabor, D. D.; and Stuckey, P. J. 2023a. Tracking Progress in Multi-Agent Path Finding. *arXiv preprint arXiv:2305.08446*.

Shen, B.; Chen, Z.; Li, J.; Cheema, M. A.; Harabor, D. D.; and Stuckey, P. J. 2023b. Beyond Pairwise Reasoning in Multi-Agent Path Finding. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 338–347.

Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T.; et al. 2019a. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, 151–158.

Stern, R.; Sturtevant, N. R.; Felner, A.; Koenig, S.; Ma, H.; Walker, T. T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T. K. S.; Boyarski, E.; and Bartak, R. 2019b. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. *Symposium on Combinatorial Search (SoCS)*, 151–158.

Vidick, T. 2018. Introduction to Spectral Graph Theory.

Yu, J.; and LaValle, S. 2013. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, 1443–1449.

Zhang, Y.; Fontaine, M. C.; Bhatt, V.; Nikolaidis, S.; and Li, J. 2023. Multi-Robot Coordination and Layout Design for Automated Warehousing. In Elkind, E., ed., *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 5503–5511. International Joint Conferences on Artificial Intelligence Organization. Main Track.