Empirical Hardness in Multi-Agent Pathfinding: Research Challenges and Opportunities

Blue Sky Ideas Track

Jingyao Ren
University of Southern California
Los Angeles, CA, United States
jingyaor@usc.edu

Eric Ewing
Brown University
Providence, RI, United States
eric_ewing@brown.edu

T. K. Satish Kumar
University of Southern California
Los Angeles, CA, United States
tkskwork@gmail.com

Sven Koenig
University of California, Irvine
Irvine, CA, United States
sven.koenig@uci.edu

Nora Ayanian
Brown University
Providence, RI, United States
nora_ayanian@brown.edu

ABSTRACT

Multi-agent pathfinding (MAPF) is the problem of finding collision-free paths for a team of agents on a map. Although MAPF is NP-hard, the hardness of solving individual instances varies significantly, revealing a gap between theoretical complexity and actual hardness. This paper outlines three key research challenges in MAPF empirical hardness to understand such phenomena. The first challenge, known as algorithm selection, is determining the best-performing algorithms for a given instance. The second challenge is understanding the key instance features that affect MAPF empirical hardness, such as structural properties like phase transition and backbone/backdoor. The third challenge is how to leverage our knowledge of MAPF empirical hardness to effectively generate hard MAPF instances or diverse benchmark datasets. This work establishes a foundation for future empirical hardness research and encourages deeper investigation into these promising and underexplored areas.

KEYWORDS

Multi-Agent Pathfinding; Empirical Hardness; Algorithm Selection; Phase Transition; Backbone; Backdoor

ACM Reference Format:
Jingyao Ren, Eric Ewing, T. K. Satish Kumar, Sven Koenig, and Nora Ayanian. 2025. Empirical Hardness in Multi-Agent Pathfinding: Research Challenges and Opportunities: Blue Sky Ideas Track. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 5 pages.

1 INTRODUCTION

Multi-agent pathfinding (MAPF) is the problem of finding collision-free paths for a team of agents from their respective start to goal locations in a shared environment, such as a grid map [41]. This fundamental problem in artificial intelligence and robotics has numerous real-world applications, such as automated warehouses [13, 24, 25], trajectory planning for unmanned-aerial-vehicle (UAV) [14], and swarm control [23]. Solving MAPF efficiently is crucial for designing scalable and reliable real-world multi-agent systems.

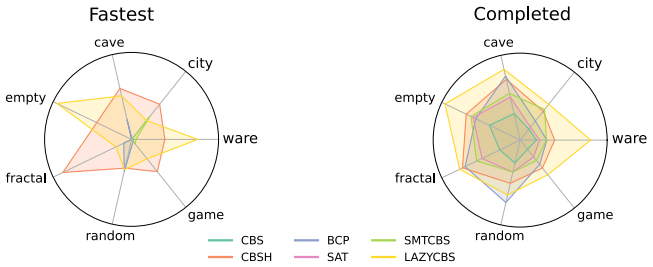

Figure 1: Number of instances where different MAPF algorithms are the fastest in runtime or successfully completed within the time limit across various map types.

Solving MAPF problems optimally for makespan or sum-of-cost is NP-hard [47], even for planar graphs [46] and grid-based problems [4]. This means that the worst-case running times grow exponentially with the problem size, such as the total number of agents. However, in practice, real-world MAPF instances are often solved fairly quickly by optimal algorithms [9, 38]. This discrepancy highlights a critical gap: while MAPF problems are challenging in theory, the hardness of individual instances varies significantly in practice. Understanding the factors that cause the significant variance of instance hardness is essential but remains poorly understood.

This gap has led to the research of *Empirical Hardness*, which studies the actual difficulty of solving specific problem instances. Empirical hardness is often measured by the runtime an algorithm takes to solve a given instance [1]. Empirical hardness research seeks to understand how problem features correlate with instance hardness and leverage these insights to improve algorithm performance. It has achieved significant success in many challenging problems such as propositional satisfiability (SAT) [5], traveling salesman problem (TSP) [12], and combinatorial auctions [21].

The empirical hardness of MAPF is a less studied topic. Most existing MAPF research has focused on algorithm design and heuristic

[1] Other metrics, like memory usage or algorithm-specific metrics (e.g., number of DP calls [26] or node expansions [32]), are also used to measure empirical hardness.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

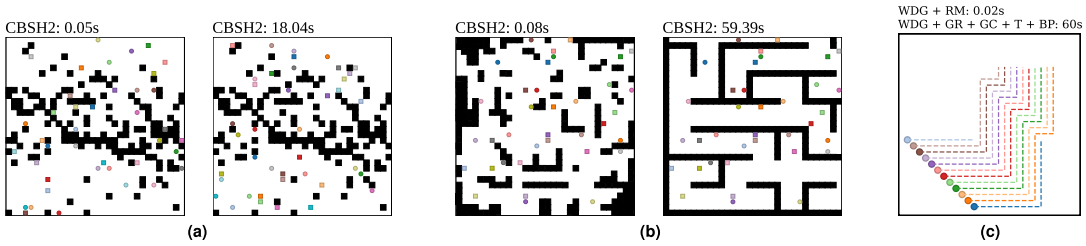

Figure 2: Runtime for CBSH2-RTC [22] algorithm for different MAPF instances. Circles and squares indicate start/goal locations, and obstacles are in black. (a) MAPF instances on the same map, the same number of agents, but significantly different runtimes. (b) MAPF instances with the same number of obstacles and same start/goal locations but dramatically different runtimes. (c) Runtime variation of CBSH2-RTC algorithm using different combinations of heuristics, dashed lines show planned paths.

enhancements. However, understanding instance-specific hardness variations is critical for developing new algorithms. As shown in Figure 1, MAPF algorithms exhibit varying strengths and weaknesses across different instances, with no single algorithm consistently outperforming others in all instances. How can we select the best algorithm for different instances? Furthermore, even the same algorithm can exhibit significant runtime differences for instances with the same size (e.g., Figure 2(a)(b)). What makes such differences in instance hardness? How can we make use of this knowledge? This paper aims to explore these questions, highlighting key challenges and opportunities in understanding and leveraging the empirical hardness of MAPF.

2 CHALLENGE I: WHEN TO CHOOSE WHICH ALGORITHM?

As shown in Figure 1, there is no single MAPF algorithm that consistently outperforms others in all cases [9, 17]. Thus, selecting the fastest algorithm on a case-by-case basis is crucial and can significantly reduce overall runtime [16, 33]. This area of research is known as algorithm selection and configuration.

2.1 Algorithm Selection

Given a specific instance, algorithm selection aims to choose the best algorithm from a predefined portfolio while minimizing performance objectives such as runtime [34]. Algorithm selection has been successfully applied to many computationally hard problems, including SAT [29] and Mixed Integer Programming (MIP) [45]. Modern approaches use machine learning techniques and treat algorithm selection as a classification or regression problem.

Algorithm selection has been successfully applied to MAPF problems as well. There are two major approaches to solving this problem. One is to formulate MAPF algorithm selection as an image classification problem [3, 33, 39]. By encoding the MAPF instances as RGB images, a convolutional neural network (CNN) can be trained to predict the fastest algorithm. The second approach manually selects MAPF instance features, such as obstacle density and agents' distance to goals [16], and trains a tree-based model based on these features [6]. A recent work [37] combined hand-crafted features with graph-based encodings and achieved notable improvements.

Challenges in Instance Encoding: The primary challenge in achieving effective algorithm selection lies in encoding MAPF instances accurately. Early CNN-based models encoded MAPF instances as RGB images with start and goal locations depicted as pixels in different colors [39]. However, this approach treated start and goal locations anonymously, failing to distinguish instances with different start-goal permutations and resulting in limited performance. One approach to address this limitation is to add single-agent shortest paths as an additional encoding, which captures both the agent distribution and map topology information [33]. This technique has been widely adopted by many CNN-based MAPF algorithm selectors [3, 7, 37]. However, there is still room for improvement in encoding strategies, as existing methods may not fully capture the complexity of MAPF instances.

Challenges in Feature Selection: For feature-based algorithm selectors, it is critical to include the most important instance features that affect the MAPF empirical hardness. But even identifying these instance features is a challenging task. For example, Kaduri et al. [16] proposed features like agent sparsity and average goal distance. However, these features fail to consider map topology, which also has major impacts on the empirical hardness of MAPF instances (e.g., Figure 2(b)). Moreover, there is no guarantee that these manually selected instance features cover all important factors.

2.2 Algorithm Configuration

Instead of selecting different algorithms, algorithm configuration optimizes the parameter settings for a given algorithm to minimize runtime or maximize solution quality [15]. These parameters can be in various forms such as categorical, boolean, or continuous, as long as they affect the algorithm's behavior and performance.

Challenges in Heuristic Configurations: Algorithm configuration remains unexplored in MAPF, as most MAPF solvers such as CBSH2-RTC [22] or BCP [20] have relatively fewer heuristics to consider compared to other combinatorial problems. However, heuristic configurations still play a crucial role. For example, Figure 2(c) demonstrates that different heuristic combinations for CBSH2-RTC [22] can lead to significantly varied runtimes. Understanding why certain configurations perform poorly in specific instances could also help develop better heuristics.

2.3 Future Directions

Future research should address two key areas:

Better Encoding Techniques: Chen et al. [7] found that similar encoding methods only bring limited performance gains in different

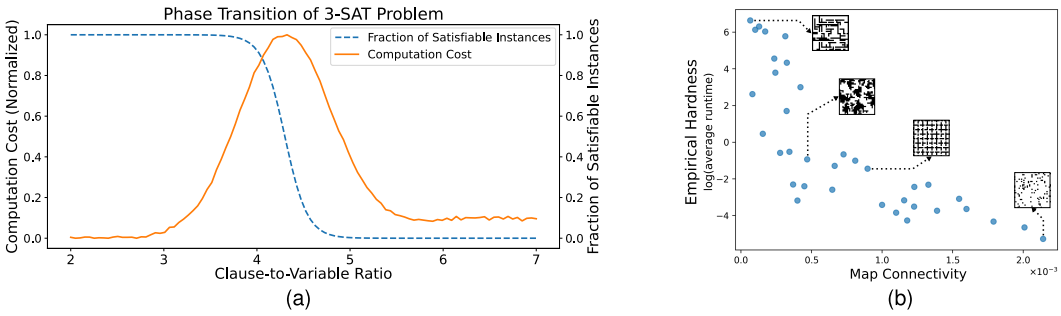

Figure 3: (a) Phase transition of 3-SAT problem. (b) The average empirical hardness of MAPF instances on different maps.

deep learning-based algorithm selectors. Future research should explore instance encoding techniques beyond single-agent shortest paths to better capture the inherent problem hardness.

Algorithm Configuration: Optimizing heuristic settings for MAPF solvers across various instances can improve performance. Additionally, analyzing poor configurations could inspire novel heuristics, enhancing solver capabilities and adaptability.

3 CHALLENGE II: UNDERSTANDING THE EMPIRICAL HARDNESS OF MAPF

Algorithm selection offers practical tools for predicting the best algorithm for each instance but does not explain the factors that cause the variance in instance hardness. For MAPF, understanding these factors is particularly challenging since both map topology and agent distribution will affect empirical hardness.

Empirical hardness is influenced by structural properties that are not immediately apparent from problem size alone. Research in related domains identifies phase transitions, backbone, and backdoors as critical predictors of instance hardness.

3.1 Phase Transition

Phase transition describes abrupt changes in system behavior or properties when critical parameters are altered [40]. This phenomenon is widely studied in physics and often describes the transition of different states for substances (e.g., water changes from liquid to gas). Phase transition also exists in many computationally hard problems, where instance hardness shifts sharply at specific thresholds [5]. For instance, as shown in Figure 3(a), the computation cost for 3-SAT instance exhibits an *easy-hard-easy* [5] pattern as the clause-to-variable ratio changes. The computation cost peaked as the fraction of satisfiable instances shifted from satisfiable to unsatisfiable. This discovery has enabled the systematic generation of challenging SAT instances by fine-tuning the clause-to-variable ratio [36]. Similar transitions have been observed in other NP-complete problems, such as graph coloring [1], Hamiltonian circuits [5], and traveling salesman (TSP) [12].

Phase transitions are particularly valuable for generating benchmark instances to evaluate algorithm performance. By tuning parameters near critical thresholds, researchers can systematically create hard instances [36]. Exploring whether phase transitions also exist in the MAPF problem is undoubtedly a fascinating research topic. However, studying phase transitions in non-decision problems, like optimization tasks, is more challenging since they lack clear solvability thresholds.

Challenges in MAPF Phase Transition: Studying the phase transition for MAPF problems is particularly challenging, as MAPF is more complex than SAT problems. It is very hard to find a single numerical parameter as clause-to-variable ratio in SAT that correlates with problem hardness. Using numerical parameters to capture map topology information is also very challenging.

Challenges in Map Topology and Agent Distribution: As shown in Figure 2(a)(b), both map topology and agent distribution can affect the empirical hardness. To address this, one approach is to assume agents are uniformly distributed and focus on map features and average empirical hardness. Ewing et al. [9] showed maps with higher betweenness centrality tend to have harder instances. Moreover, as shown in Figure 3(b), poorly connected maps generally exhibit higher average empirical hardness compared to well-connected ones [32]. While these studies provide a foundation for exploring MAPF phase transitions, removing the assumptions on agent distribution remains challenging.

3.2 Backbone and Backdoor

Backbone and backdoor are important concepts to understand empirical hardness, offering insights into problem instance structures.

Backbone: The backbone represents a set of variables whose values are identical across all possible solutions of an instance [27]. The backbone is often used as a key indicator for empirical hardness. In SAT problems, a large number of backbone variables often indicate an over-constrained instance and it is hard to find feasible solutions [2, 30]. Early identification and elimination of the backbones will help improve the algorithm's performance [8, 27].

Backdoor: The backdoor represents a set of variables that, when properly assigned, simplifies the problem [44]. The size of the backdoor is an effective indicator for estimating empirical hardness, as smaller backdoors indicate well-structured and easier instances [18, 35]. For example, setting the backdoor variables properly for SAT instances will make the remaining formula solvable in polynomial time by certain SAT solvers [43].

Challenges in Backbone/Backdoor: There is no existing research on the backbone/backdoor of MAPF problems. With MAPF's complex instance features compared to SAT, the main challenge is how to properly define the backbone/backdoor. In a broader concept, the backbone could be caused by specific structures that

dominate the map connectivity, such as narrow corridors or choker points. These structures might force agents to traverse certain locations in all possible solutions, causing more conflicts between agents, and making the instances harder. The backdoor could be a subset of agents causing the majority of conflicts. Planning their paths first may simplify the remaining problem.

3.3 Future Directions

Future research should focus on investigating phase transitions and backbone/backdoor, such as exploring different types of phase transitions for MAPF or providing formal definitions of backbone and backdoor. These studies can provide systematic insights into instance hardness and guide the development of novel algorithms.

4 CHALLENGE III: GENERATING HARD MAPF INSTANCES AND BENCHMARK DATASETS

Once the key instance features influencing MAPF empirical hardness are identified, the next challenge is how to effectively leverage this knowledge to generate hard instances. With the development of increasingly powerful algorithms [11, 19, 22], there is also a growing need for more challenging and diverse benchmark datasets to thoroughly evaluate their performance.

4.1 Generating Hard MAPF Instances

Generating hard instances is not an easy task, especially without a profound understanding of empirical hardness. For example, randomly generated SAT instances are generally easy to solve, but hard instances occur more frequently at specific clause-to-variable ratios [36]. The study of phase transition has inspired numerous methods of generating hard instances, as the most challenging instances often exist in phase transition regions. Selman et al. [36] demonstrated that challenging SAT instances can be generated by setting the clause-to-variable ratio to approximately 4.25.

Challenges in Hard Instance Generation: Since phase transition in MAPF has not been formally studied, it cannot be used to guide the generation of hard instances as it does in other research domains. Nevertheless, MAPF researchers have explored various alternative methods for creating challenging instances. The key challenge is that both agent distribution and map topology will affect the empirical hardness of MAPF instances. And it is very hard to study them together.

Some researchers make assumptions about the distribution of agents' start and goal locations and focus on the topology of maps. The most common assumption is that agents' start and goal locations are sampled uniformly at random. This scenario studies the average empirical hardness of the MAPF problem on different maps. For instance, Ewing et al. [9] showed that the empirical hardness is correlated with the betweenness centrality of the maps. Betweenness centrality measures how often a map cell is traversed by the shortest paths and a high betweenness centrality often indicates choker points. Similarly, Ren et al. [32] showed that hard instances are more common on poorly connected maps and demonstrated using the Quality Diversity (QD) method to generate maps with controllable connectivity. However, as discussed in Section 3.1, removing the assumptions on agent distribution remains challenging.

4.2 Generating Better Benchmark Datasets

Developing better benchmark datasets with a wide range of hardness is crucial for evaluating MAPF algorithms. Research that initially focused on simplifying instances has provided valuable insights. For instance, Zhang et al. [49] used QD methods to optimize warehouse throughput, which can be viewed as indirectly generating maps with lower average empirical hardness. This approach was improved with NCA [28] and CMA-MAE [10] for scalability [48]. Qian et al. [31] later proposed a more powerful framework for generating diverse MAPF datasets with varying hardness.

Challenges in Benchmarking: The key challenge is to choose the best metrics for quality diversity methods to generate diverse instances. Qian et al. [31] identified obstacle density and the Kullback–Leibler (KL) divergence of tile patterns as effective indicators for general hardness and spatial arrangement. While these metrics provide a foundation for benchmark generation and rigorous MAPF solver evaluation, many other potential metrics remain unexplored.

4.3 Future Directions

Generating hard MAPF instances without general agent distribution assumptions remains challenging. Possible directions are:

Reinforcement Learning: Train an instance generator with a reward function that encourages conflicts between agents, making instances harder for conflict-based algorithms.

Specialized Hard Instances: Extract MAPF instances from other problems. For example, one could extract challenging SAT instances from a MAPF problem [42]. The main challenge is identifying complex problems that can be effectively reduced to MAPF.

5 CONCLUSION

This paper identifies and presents three key research challenges in understanding and leveraging the empirical hardness of multi-agent pathfinding (MAPF). First, selecting the best-performing algorithm for a given instance remains an open problem, requiring advancements in better instance encoding techniques in algorithm selection and configuration. Second, understanding the factors that influence instance hardness, such as phase transitions and other structural features, is essential for building a deeper theoretical foundation. Finally, leveraging this knowledge to generate challenging MAPF instances and diverse benchmark datasets can lead to the development of more robust and efficient algorithms. Addressing these challenges will bridge the gap between theoretical complexity and practical hardness and ultimately advance our understanding of the true essence of the MAPF problem.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF) under grant 2330942 at Brown University, 2112533 at the University of Southern California, and 2434916, 2346058, 2321786, 2121028, and 1935712 at the University of California, Irvine, as well as gifts from Amazon Robotics. 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

[1] Dimitris Achlioptas and Ehud Friedgut. 1999. A sharp threshold for k-colorability. *Random Structures & Algorithms* 14, 1 (1999), 63–70.

[2] Dimitris Achlioptas, Carla Gomes, Henry Kautz, and Bart Selman. 2000. Generating satisfiable problem instances. *AAAI/IAAI* 2000 (2000), 256–261.

[3] Jean-Marc Alkazzi, Anthony Rizk, Michel Salomon, and Abdallah Makhoul. 2022. Mapfaster: A faster and simpler take on multi-agent path finding algorithm selection. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10088–10093.

[4] Jacopo Banfi, Nicola Basilico, and Francesco Amigoni. 2017. Intractability of Time-Optimal Multirobot Path Planning on 2d Grid Graphs with Holes. *IEEE Robotics and Automation Letters* 2, 4 (2017), 1941–1947.

[5] Peter C Cheeseman, Bob Kanefsky, William M Taylor, et al. 1991. Where the really hard problems are.. In *Ijcai*, Vol. 91. 331–337.

[6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proc. of the 22nd ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining*. 785–794.

[7] Weizhe Chen, Zhihan Wang, Jiaoyang Li, Sven Koenig, and Bistra Dilkina. 2024. No Panacea in Planning: Algorithm Selection for Suboptimal Multi-Agent Path Finding. *arXiv preprint arXiv:2404.03554* (2024).

[8] Olivier Dubois and Gilles Dequen. 2001. A backbone-search heuristic for efficient solving of hard 3-SAT formulae. In *IJCAI*, Vol. 1. 248–253.

[9] Eric Ewing, Jingyao Ren, Dhvani Kansara, Vikraman Sathiyanarayanan, and Nora Ayanian. 2022. Betweenness centrality in multi-agent path finding. In *International Conference on Autonomous Agents and Multiagent Systems*.

[10] Matthew Fontaine and Stefanos Nikolaidis. 2023. Covariance matrix adaptation map-annealing. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 456–465.

[11] Graeme Gange, Daniel Harabor, and Peter J Stuckey. 2019. Lazy CBS: implicit conflict-based search using lazy clause generation. In *Proceedings of the international conference on automated planning and scheduling*, Vol. 29. 155–162.

[12] Ian P Gent and Toby Walsh. 1996. The TSP phase transition. *Artificial Intelligence* 88, 1-2 (1996), 349–358.

[13] Wolfgang Hönig, Scott Kiesel, Andrew Tinka, Joseph W Durham, and Nora Ayanian. 2019. Persistent and robust execution of MAPF schedules in warehouses. *IEEE Robotics and Automation Letters* 4, 2 (2019), 1125–1131.

[14] Wolfgang Hönig, James A Preiss, TK Satish Kumar, Gaurav S Sukhatme, and Nora Ayanian. 2018. Trajectory planning for quadrotor swarms. *IEEE Transactions on Robotics* 34, 4 (2018), 856–869.

[15] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*. Springer, 507–523.

[16] Omri Kaduri, Eli Boyarski, and Roni Stern. 2020. Algorithm Selection for Optimal Multi-Agent Pathfinding. In *Proc. of the Intl Conf on Automated Planning and Scheduling*, Vol. 30. 161–165.

[17] Omri Kaduri, Eli Boyarski, and Roni Stern. 2021. Experimental evaluation of classical multi agent path finding algorithms. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 12. 126–130.

[18] Philip Kilby, John Slaney, Sylvie Thiébaux, Toby Walsh, et al. 2005. Backbones and backdoors in satisfiability. In *AAAI*, Vol. 5. 1368–1373.

[19] Edward Lam, Pierre Le Bodic, Daniel Harabor, and Peter J Stuckey. 2022. Branch-and-cut-and-price for multi-agent path finding. *Computers & Operations Research* 144 (2022), 105809.

[20] Edward Lam, Pierre Le Bodic, Daniel Damir Harabor, and Peter J Stuckey. 2019. Branch-and-Cut-and-Price for Multi-Agent Pathfinding. In *Proc. of the Intl Joint Conf on Artificial Intelligence*. 1289–1296.

[21] Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. 2006. Empirical hardness models for combinatorial auctions. *Combinatorial auctions* (2006), 479–504.

[22] Jiaoyang Li, Daniel Harabor, Peter J Stuckey, Hang Ma, Graeme Gange, and Sven Koenig. 2021. Pairwise symmetry reasoning for multi-agent path finding search. *Artificial Intelligence* 301 (2021), 103574.

[23] Jiaoyang Li, Kexuan Sun, Hang Ma, Ariel Felner, TK Kumar, and Sven Koenig. 2020. Moving agents in formation in congested environments. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 11. 131–132.

[24] Jiaoyang Li, Andrew Tinka, Scott Kiesel, Joseph W Durham, TK Satish Kumar, and Sven Koenig. 2021. Lifelong multi-agent path finding in large-scale warehouses. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11272–11281.

[25] Hang Ma, Wolfgang Hönig, TK Satish Kumar, Nora Ayanian, and Sven Koenig. 2019. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7651–7658.

[26] David Mitchell, Bart Selman, Hector Levesque, et al. 1992. Hard and easy distributions of SAT problems. In *Aaai*, Vol. 92. 459–465.

[27] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. 1999. Determining computational complexity from characteristic 'phase transitions'. *Nature* 400, 6740 (1999), 133–137.

[28] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. 2020. Growing neural cellular automata. *Distill* 5, 2 (2020), e23.

[29] Eugene Nudelman, Kevin Leyton-Brown, Holger H Hoos, Alex Devkar, and Yoav Shoham. 2004. Understanding random SAT: Beyond the clauses-to-variables ratio. In *Principles and Practice of Constraint Programming–CP 2004: 10th International Conference, CP 2004, Toronto, Canada, September 27-October 1, 2004. Proceedings 10*. Springer, 438–452.

[30] Andrew J Parkes. 1997. Clustering at the phase transition. In *AAAI/IAAI*. Citeseer, 340–345.

[31] Cheng Qian, Yulun Zhang, and Jiaoyang Li. 2024. A Quality Diversity Approach to Automatically Generate Multi-Agent Path Finding Benchmark Maps. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 17. 279–280.

[32] Jingyao Ren, Eric Ewing, TK Satish Kumar, Sven Koenig, and Nora Ayanian. 2024. Map Connectivity and Empirical Hardness of Grid-based Multi-Agent Pathfinding Problem. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 34. 484–488.

[33] Jingyao Ren, Vikraman Sathiyanarayanan, Eric Ewing, Baskin Senbaslar, and Nora Ayanian. 2021. MAPFAST: A deep algorithm selector for multi agent path finding using shortest path embeddings. *arXiv preprint arXiv:2102.12461* (2021).

[34] John R Rice. 1976. The Algorithm Selection Problem. In *Advances in Computers*. Vol. 15. Elsevier, 65–118.

[35] Yongshao Ruan, Henry A Kautz, and Eric Horvitz. 2004. The backdoor key: A path to understanding problem hardness. In *AAAI*, Vol. 4. 118–123.

[36] Bart Selman, David G Mitchell, and Hector J Levesque. 1996. Generating hard satisfiability problems. *Artificial intelligence* 81, 1-2 (1996), 17–29.

[37] Carmel Shabalin, Omri Kaduri, and Roni Stern. 2024. Algorithm Selection for Optimal Multi-Agent Path Finding via Graph Embedding. *arXiv preprint arXiv:2406.10827* (2024).

[38] Bojie Shen, Zhe Chen, Muhammad Aamir Cheema, Daniel D Harabor, and Peter J Stuckey. 2023. Tracking progress in multi-agent path finding. *arXiv preprint arXiv:2305.08446* (2023).

[39] Devon Sigurdson, Vadim Bulitko, Sven Koenig, Carlos Hernandez, and William Yeoh. 2019. Automatic Algorithm Selection in Multi-Agent Pathfinding. *arXiv preprint arXiv:1906.03992* (2019).

[40] Ricard V Solé. 2011. *Phase transitions*. Vol. 3. Princeton University Press.

[41] Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, and Roman Bartak. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. *Symposium on Combinatorial Search (SoCS)* (2019), 151–158.

[42] Pavel Surynek. 2019. Unifying search-based and compilation-based approaches to multi-agent path finding through satisfiability modulo theories. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 10. 202–203.

[43] Ryan Williams, Carla Gomes, and Bart Selman. 2003. On the connections between backdoors, restarts, and heavy-tailedness in combinatorial search. *structure* 23, 4 (2003).

[44] Ryan Williams, Carla P Gomes, and Bart Selman. 2003. Backdoors to typical case complexity. In *IJCAI*, Vol. 3. 1173–1178.

[45] Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2011. Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming. In *RCRA workshop on experimental evaluation of algorithms for solving problems with combinatorial explosion at the international joint conference on artificial intelligence (IJCAI)*. 16–30.

[46] Jingjin Yu. 2015. Intractability of optimal multirobot path planning on planar graphs. *IEEE Robotics and Automation Letters* 1, 1 (2015), 33–40.

[47] Jingjin Yu and Steven M LaValle. 2013. Structure and Intractability of Optimal Multi-Robot Path Planning on Graphs. In *Twenty-Seventh AAAI Conf on Artificial Intelligence*. 1444–1449.

[48] Yulun Zhang, Matthew Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. 2024. Arbitrarily scalable environment generators via neural cellular automata. *Advances in Neural Information Processing Systems* 36 (2024).

[49] Yulun Zhang, Matthew C Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. 2024. Multi-robot coordination and layout design for automated warehousing. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 17. 305–306.