contains a target of one clause agent on its top row. We include an empty column at the right of each clause gadget to ensure accessibility of the latter target. Specifically, the empty columns allow each clause agent c to reach its target position even if all the targets in the respective clause gadget are occupied (note that in this case c's path will be longer than the shortest possible path).

The gadgets are arranged from left to right so that first we have variable gadgets and then clause gadgets. The order of gadgets of the same type and start/target positions within a gadget are arbitrary.

It is easy to verify that there exists a monotone motion for plan for M. Simply repeat the following for each variable gadget Ξ, going from the rightmost to the leftmost gadget: first move agents that are on Ξ's top path in right to left order and then do the same for Ξ's bottom path. At this point all literal agents are at their targets. Therefore, from now on there always exists a clause agent that may be moved to its target, until all have reaches their targets.

The following theorem proves the correctness of the construction.

Theorem 1. *M has a monotone motion plan with a distance cost of $d^*(M)$ if and only if ϕ is satisfiable.*

Proof. Assume that ϕ has a satisfying assignment $\mathcal{A}$. Let R^+ (resp. R^-) be the set of agents corresponding to literals that evaluate to true (resp. false) according to $\mathcal{A}$. Let P be the shortest path from the entrance of leftmost variable gadget to the exit of the rightmost variable gadget that passes through all the start positions of R^-; see Figure 2. Observe that for each variable gadget, R^- contains agents that are all either on the gadget's top path or bottom path. This means that P exists and that it is x-monotone.

We specify a monotone motion plan in which all agents move along P while traversing variable gadgets. The motion plan has three stages, which are illustrated in Figure 2. In each stage a group of agents move, starting with R^-, then the clause agents, and finally R^+. First, agents in R^- move in right to left order along P, which guarantees no collisions between literal agents. Observe that each agent in R^- can achieve the shortest path to its target, initially guided by P. Next, the clause agents move in the natural order that allows each of them to leave their initial room using the shortest path with no collisions. We have the following properties at this point: P contains only empty cells and each clause gadget's middle row must also contain an unoccupied target of an agent in R^+. The latter holds because $\mathcal{A}$ satisfies ϕ and the agents of R^+ have not yet moved. Therefore, each clause agent can also take an optimal path. Finally, R^+ can move optimally, guided by P, similarly to R^-.

For the other direction, we assume that there is a monotone motion plan for M with a distance cost of $d^*(M)$ and show that ϕ has a satisfying assignment. Let R^+ denote the agents that move after the last clause agent moves. For any variable $\alpha \in \phi$, R^+ cannot contain literal agents corresponding to both α and $\overline{\alpha}$, since then clause agents would not be able to reach their target positions. Therefore, we can define an assignment $\mathcal{A}$ in which the literals corresponding to R^+ evaluate to true. (If a variable does not have literals in R^+, then it can be assigned an arbitrary value.)

Let C be a clause in ϕ and let c be the corresponding clause agent, i.e., c has to go to C's clause gadget. There must be a target vertex v in the middle row of C's clause gadget that is unoccupied when c moves. Such a vertex v must exist in order for c to have the shortest possible path to its target $t(c)$ during its turn to move. Therefore, the literal agent r, with $t(r) = v$ must be in R^+ by definition, i.e., it must move after r. Hence, the literal corresponding to r evaluates to true by $\mathcal{A}$, which means that C is satisfied and we are done. □

It is easy to verify that the number of agents in M as well as the size of the resulting graph is linear in $|\phi|$. Therefore, we conclude the following.

Corollary 1. *Monotone Distance-Optimal MAPF is NP-hard and cannot be solved in sub-exponential time $2^{o(n)}$ or $2^{o(|V|)}$ unless ETH fails, even for a grid graph $G = (V, E)$ with 3 rows, where n is the number of agents.*

5 GENERAL DISTANCE-OPTIMAL MAPF

Our previous construction no longer holds once non-monotone motions are allowed. Since agents are not constrained by time, they can make intermediate stops, which adds a lot of possibilities to the motion plan. The main challenge is that literal agents can "cheat" by making intermediate stops in variable gadgets along their way. For example, in Figure 1 y_1 could position itself in the cell to the left of z_1, thereby creating a path through y's variable gadget that does not enforce an assignment to y.[1] Therefore, we introduce *blockers*, which are new agents that prevent undesirable intermediate stops, and prove that general distance-optimal MAPF is NP-hard on grid graphs.

As before, for a 3-SAT formula ϕ, we construct a distance-optimal MAPF instance M' that has a motion plan with a distance cost of $d^*(M')$ if and only if ϕ is satisfiable. An example of the new instance M' is illustrated in Figure 3. In general, M' is the same as M from Section 4 except for the following change: Each variable gadget now has a blocker agent that starts at the gadget's entrance and has to go to the gadget's exit. This ensures that all the agents passing through the gadget must use the same path within the gadget, thereby keeping the incoming and outgoing order of the traversing agents the same. This property prevents clause agents from "cheating" and bypassing literal agents, thereby mimicking the monotone case. The following lemma formally states the functionality of the blockers:

Lemma 1. *Let Ξ be a variable gadget in M'. Then, in any motion plan for M' with a cost of $d^*(M')$, all the agents that traverse Ξ must take the same path through Ξ.*

Proof. Let b denote the blocker agent that is initially at Ξ's entrance. Since $P(b)$ is an optimal path, it can be either the top path or the bottom path in Ξ, which we denote by P_1 and P_2, respectively; see Figure 5. Similarly, any agent r that traverses Ξ, must have either P_1 or P_2 be a subpath of its path, $P(r)$. However, we cannot have $P(b)$ be a subpath of $P(r)$, as that would necessarily lead to a collision between b and r. That is, r must somehow bypass b to exit Ξ, which is not possible if they take the same path in Ξ. This leaves r with exactly one path that it can take through Ξ, namely, the one not taken by b. Since this applies to any agent r, we have all the agents traversing Ξ take the same path through Ξ, as required. □

[1]Eliminating free cells in the two paths in z's variable gadget does not solve the problem. Observe that it is possible for both positive and negative literal agents to leave the gadget before any clause agent moves. If this happens, it would create space for the undesirable intermediate stops described.

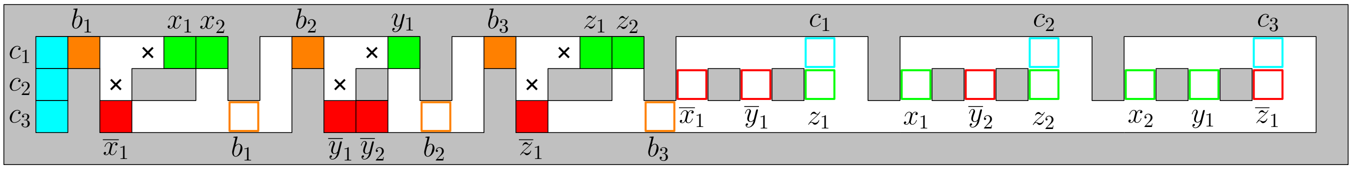

Figure 3: The instance M' modified from M in Figure 1. Blocker agents (and their target positions) are shown in orange.

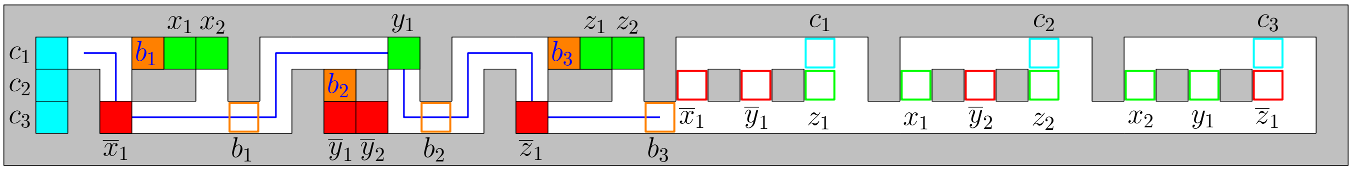

Figure 4: The path P (blue) corresponding to the assignment $x = T, y = F, z = T$ along with a snapshot after the first stage of the motion plan for M'.

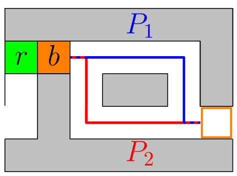

Figure 5: A variable gadget at some time point during a motion plan. The gadget's respective blocker agent b is at its start position and its target position is on the right. The agent r needs to traverse the gadget. The two possible shortest paths for b are shown as P_1 (blue) and P_2 (red). Note that these paths overlap near $s(b)$ and $t(b)$. Here we highlight that b and r cannot use the same path in the gadget.

We now prove the correctness of the modified construction.

Theorem 2. *M' has a motion plan with a distance cost of $d^*(M')$ if and only if ϕ is satisfiable.*

Proof. We adjust the motion plan defined for M in the proof of Theorem 1 to accommodate the blocker agents. Let $\mathcal{A}$ be a satisfying assignment and let P be as defined before. The new plan has two additional stages. First, blocker agents move to intermediate stops that are not on P (in arbitrary order). Figure 3 indicates the two possible intermediate stops for each blocker agent using crosses and Figure 4 shows an example of the blockers' positions after the first stage. Next, we perform the same motion plan as for M. Then, in the final stage, blockers move to their targets (again in arbitrary order, since each assignment only contains the blocker at this stage). It is easy to verify that the new plan is optimal: The intermediate stops always allow blocker agents to not block P, while also allowing them to eventually reach their targets using optimal paths. As for the rest of the agents, each agent is able to take the same path as in Theorem 1.

For the other direction, let us assume that there is a motion plan for M' with a cost of $d^*(M')$. By Lemma 1, all the clause agents follow the same path in each variable gadget. Therefore, let P denote the path that all the clause agents follow between the entrance of the leftmost variable gadget to the exit of the rightmost variable gadget. We define a satisfying assignment $\mathcal{A}$ using P as follows: A variable $\alpha \in \phi$ is assigned to be true (resp. false) if P passes though start positions of negative (resp. positive) literal agents in α's variable gadget. In other words, literals corresponding to literal agents that are initially located on P are assigned to be false (see Figure 4 for an example of the correspondence between P and $\mathcal{A}$). As before, let R^+ (resp. R^-) be the set of agents corresponding to literals that evaluate to true (resp. false) according to $\mathcal{A}$.

Let C be a clause in ϕ and let c be the corresponding clause agent, i.e., c has to go to C's clause gadget. We show that C is satisfied by $\mathcal{A}$. It suffices to show that c's path, $P(c)$, contains a target of an agent in R^+ in C's clause gadget. Let us assume for a contradiction that this does not hold. Then, since $P(c)$ is c's individually optimal path, it must still pass through a target in C's clause gadget. This target must be $t(r)$ of some $r \in R^-$. We will now claim that $P(r)$ must be a subpath of $P(c)$. Intuitively, this means that c cannot bypass r, and will ultimately be blocked by r once r reaches $t(r)$, thus yielding the contradiction.

Let Ξ be the variable gadget on which $s(r)$ lies. By definition, $s(r)$ lies on P, so r must follow P to exit Ξ using the shortest path. By Lemma 1, r must continue following P, the subpath shared by all clause agents, in all variable gadgets it traverses (after leaving Ξ). The remainder of $P(r)$ must also be a subpath of $P(c)$ since both paths are optimal and hence simply go right until reaching the cell below $t(r)$. Therefore, $P(r)$ as a whole is a subpath of $P(c)$. This is a contradiction since then r must reach $t(r)$ before c does, which would block c. In conclusion, we showed that C is satisfied, which holds for any clause, and so we are done. ◻

Observe that all our arguments hold regardless of whether parallel motion is allowed or not. For the case of synchronous rotations along cycles, by definition for such a rotation to occur, there has to be an agent moving left. As none of the individually optimal paths for agents ever require moving left, rotations cannot occur for a plan with cost $d^*(M')$. It is easy to verify that the number of agents in M' as well as the size of the resulting graph remains linear in $|\phi|$. Therefore, we conclude the following:

Corollary 2. *Distance-Optimal MAPF is NP-hard and cannot be solved in sub-exponential time $2^{o(n)}$ or $2^{o(|V|)}$ unless ETH fails, even for a grid graph $G = (V, E)$ with 3 rows, where n is the number of agents. This holds for both parallel and sequential motions.*

6 CONCLUSION

We have shown that distance-optimal MAPF is NP-hard on grid graphs with more than one empty vertex, settling the open problem by Banfi et al. [4]. Before discussing possible implications of our proof towards positive results, we advocate for more focus on achieving refined hardness results. Specifically, we believe that when proving hardness, one should strive towards the following two goals: The considered problem setting should be the most restricted, i.e., simplest, one that is still useful, while at the same time the hardness reduction should be kept simple as well, ideally with low blow-up. We note that these two goals can sometimes collide (e.g., compare the elegant proof by Goldreich [14] for general graphs versus the one for the grid [29], which was simplified only close to 30 years later [8]). Nevertheless, we believe that we have homed in on both goals in this paper, thereby improving the structural understanding of MAPF.

6.1 Implications of the hardness result

The previous hardness proof for distance-optimal MAPF on planar graphs by Yu [39] uses agents that need to move in opposite directions in order to emulate an assignment. As a result, Yu concluded that the hardness of the problem appears to arise from contention that occurs when two or more groups of agents want to move in opposite directions through the same set of narrow paths. From a practical standpoint, Yu suggests that environments with many robots would benefit from a design that minimizes path sharing among the robots.

Since in our construction all the agents move in the same general direction, we show that hardness remains even without opposite direction movement. In fact, we remark that our construction can be modified so that the problem is NP-hard even if agents can only move down and right. This requires two main modifications: The variable gadgets need to be arranged in a staircase-like manner, in which the exit of one gadget is on the same grid row as the entrance of its neighboring gadget. This modification eliminates the need for agents to go up between variable gadgets. The second modification is vertically mirroring the clause gadgets such that the clause agents' targets are on the bottom row each gadget.

Given that opposite direction movement does not play a role in our case, we provide another perspective for the source of difficulty of the problem. For the purpose of this discussion, when we say that a target vertex v is *fulfilled*, we mean that the agent r with $t(r) = v$ has reached v. Our construction's challenging aspect is the need of agents to negotiate through paths with many start and target vertices of other agents. This results in two conflicting goals: On the one hand each agent needs to pass target positions along its path before they become fulfilled (assuming that once they become fulfilled, the agent will have to take a longer path). This suggests that algorithms for distance-optimal MAPF can benefit from "prioritizing" agents that have targets along their optimal path that are close to becoming fulfilled. At the same time, each agent should aim not to force other agents to move in a manner that fulfills targets that other agents still need to pass through.

6.2 Future work

Our discussion on implications of the hardness results calls for the investigation of more parameters that affect the hardness of the problem. As we noted, agents in our construction have to pass through a large number of start and target positions of other agents. A natural question is whether the problem remains hard even if each agent has an optimal path that passes through a constant number of start and target positions. Another significant feature of our construction is that the agents' paths must largely overlap. Therefore, the case where the paths can overlap less, which requires a different layout than the "long and narrow" grid that we used, seems worthy of more study. Overall, we believe that more subtle underlying parameters need to be considered. Previous positive results that employ parameterized complexity in discrete motion planning problems [1, 17] provide some encouragement.

While our refined analysis has resulted in a concrete lower bound, we are not aware of any algorithm that matches (or nearly matches) it, i.e., has a running time of $O(2^n)$ or $O(2^{|V|})$. This brings to light a gap between lower and upper bounds, which we believe calls for additional refined analysis on both sides. On the upper bound side such work was recently done by Gordon et al. [16] for (time-optimal) Conflict-Based Search [31], which tightened the running time of the algorithm. Their improved bound is exponential in a few parameters, therefore it could be beneficial to simultaneously analyze multiple parameters on the lower bound side. We also remark here that existing hardness results for time-optimal MAPF on planar graphs [39] and 2D grid graphs [4, 7] use reductions that are not linear. Hopefully, tightening both lower and upper bounds will uncover areas for algorithmic improvements.

ACKNOWLEDGMENTS

This work has been supported in part by the Israel Science Foundation (grant no. 1736/19), by NSF/US-Israel-BSF (grant no. 2019754), by the Israel Ministry of Science and Technology (grant no. 103129), by the Blavatnik Computer Science Research Fund, and by the Yandex Machine Learning Initiative for Machine Learning at Tel Aviv University.

REFERENCES

[1] Pankaj K. Agarwal, Boris Aronov, Tzvika Geft, and Dan Halperin. 2021. On Two-Handed Planar Assembly Partitioning with Connectivity Constraints. In *SODA*. SIAM, 1740–1756.
[2] Meysam Aghighi, Christer Bäckström, Peter Jonsson, and Simon Ståhlberg. 2016. Refining complexity analyses in planning by exploiting the exponential time hypothesis. *Ann. Math. Artif. Intell.* 78, 2 (2016), 157–175.
[3] Christer Bäckström and Peter Jonsson. 2017. Time and Space Bounds for Planning. *J. Artif. Intell. Res.* 60 (2017), 595–638.
[4] Jacopo Banfi, Nicola Basilico, and Francesco Amigoni. 2017. Intractability of Time-Optimal Multirobot Path Planning on 2D Grid Graphs with Holes. *IEEE Robotics Autom. Lett.* 2, 4 (2017), 1941–1947.
[5] John F. Canny and John H. Reif. 1987. New Lower Bound Techniques for Robot Motion Planning Problems. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987.* 49–60.
[6] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized algorithms.* Vol. 5. Springer.
[7] Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Henk Meijer, and Christian Scheffer. 2019. Coordinated Motion Planning: Reconfiguring a Swarm of Labeled

Robots with Bounded Stretch. *SIAM J. Comput.* 48, 6 (2019), 1727–1762.
[8] Erik D. Demaine and Mikhail Rudoy. 2018. A simple proof that the (n^2-1)-puzzle is hard. *Theor. Comput. Sci.* 732 (2018), 80–84.
[9] Eduard Eiben, Jonathan Gemmell, Iyad A. Kanj, and Andrew Youngdahl. 2018. Improved Results for Minimum Constraint Removal. In *AAAI*. AAAI Press, 6477–6484.
[10] Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S.B. Mitchell. 2021. Computing Coordinated Motion Plans for Robot Swarms: The CG: SHOP Challenge 2021. *arXiv preprint arXiv:2103.15381* (2021).
[11] Ariel Felner, Roni Stern, Solomon Eyal Shimony, Eli Boyarski, Meir Goldenberg, Guni Sharon, Nathan R. Sturtevant, Glenn Wagner, and Pavel Surynek. 2017. Search-Based Optimal Solvers for the Multi-Agent Pathfinding Problem: Summary and Challenges. In *SOCS*. AAAI Press, 29–37.
[12] Michael R Garey and David S. Johnson. 1977. The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math.* 32, 4 (1977), 826–834.
[13] Tzvika Geft and Dan Halperin. 2021. Tractability Frontiers in Multi-Robot Coordination and Geometric Reconfiguration. *arXiv preprint arXiv:2104.07011* (2021).
[14] Oded Goldreich. 1984. Finding the Shortest Move-Sequence in the Graph-Generalized 15-Puzzle Is NP-Hard. laboratory for Computer Science, Massachusetts Institute of Technology, unpublished manuscript.
[15] Oded Goldreich. 2011. Finding the Shortest Move-Sequence in the Graph-Generalized 15-Puzzle Is NP-Hard. In *Studies in Complexity and Cryptography*. Lecture Notes in Computer Science, Vol. 6650. Springer, 1–5.
[16] Ofir Gordon, Yuval Filmus, and Oren Salzman. 2021. Revisiting the Complexity Analysis of Conflict-Based Search: New Computational Techniques and Improved Bounds. In *SOCS*. AAAI Press, 64–72.
[17] Siddharth Gupta, Guy Sa'ar, and Meirav Zehavi. 2020. The Parameterized Complexity of Motion Planning for Snake-Like Robots. *J. Artif. Intell. Res.* 69 (2020), 191–229.
[18] Russell Impagliazzo and Ramamohan Paturi. 2001. On the Complexity of k-SAT. *J. Comput. Syst. Sci.* 62, 2 (2001), 367–375.
[19] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.* 63, 4 (2001), 512–530.
[20] Wm Woolsey Johnson, William Edward Story, et al. 1879. Notes on the "15" puzzle. *American Journal of Mathematics* 2, 4 (1879), 397–404.
[21] Omri Kaduri, Eli Boyarski, and Roni Stern. 2020. Algorithm Selection for Optimal Multi-Agent Pathfinding. In *ICAPS*. AAAI Press, 161–165.
[22] David G. Kirkpatrick and Paul Liu. 2016. Characterizing minimum-length coordinated motions for two discs. In *Proceedings of the 28th Canadian Conference on Computational Geometry, CCCG 2016, August 3-5, 2016, Simon Fraser University, Vancouver, British Columbia, Canada*, Thomas C. Shermer (Ed.). Simon Fraser University, Vancouver, British Columbia, Canada, 252–259.
[23] Jiaoyang Li, Andrew Tinka, Scott Kiesel, Joseph W. Durham, T. K. Satish Kumar, and Sven Koenig. 2020. Lifelong Multi-Agent Path Finding in Large-Scale Warehouses. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, 1898–1900.
[24] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. 2011. Lower bounds based on the Exponential Time Hypothesis. *Bull. EATCS* 105 (2011), 41–72.
[25] Hang Ma, TK Satish Kumar, and Sven Koenig. 2017. Multi-agent path finding with delay probabilities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
[26] Hang Ma, Craig A. Tovey, Guni Sharon, T. K. Satish Kumar, and Sven Koenig. 2016. Multi-Agent Path Finding with Payload Transfers and the Package-Exchange Robot-Routing Problem. In *AAAI*. AAAI Press, 3166–3173.
[27] Sharada Prasanna Mohanty, Erik Nygren, Florian Laurent, Manuel Schneider, Christian Scheller, Nilabha Bhattacharya, Jeremy D. Watson, Adrian Egli, Christian Eichenberger, Christian Baumberger, Gereon Vienken, Irene Sturm, Guillaume Sartoretti, and Giacomo Spigler. 2020. Flatland-RL: Multi-Agent Reinforcement Learning on Trains. *arXiv preprint arXiv:2012.05893* (2020).
[28] Robert Morris, Corina S. Pasareanu, Kasper Søe Luckow, Waqar Malik, Hang Ma, T. K. Satish Kumar, and Sven Koenig. 2016. Planning, Scheduling and Monitoring for Airport Surface Operations. In *AAAI Workshop: Planning for Hybrid Systems (AAAI Workshops, Vol. WS-16-12)*. AAAI Press.
[29] Daniel Ratner and Manfred Warmuth. 1990. The (n^2-1)-puzzle and related relocation problems. *Journal of Symbolic Computation* 10, 2 (1990), 111–137.
[30] Oren Salzman and Roni Stern. 2020. Research Challenges and Opportunities in Multi-Agent Path Finding and Multi-Agent Pickup and Delivery Problems. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, 1711–1715.
[31] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artif. Intell.* 219 (2015), 40–66.
[32] Rahul Shome, Kiril Solovey, Andrew Dobson, Dan Halperin, and Kostas E. Bekris. 2020. dRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning. *Auton. Robots* 44, 3-4 (2020), 443–467.
[33] Kiril Solovey, Jingjin Yu, Or Zamir, and Dan Halperin. 2015. Motion Planning for Unlabeled Discs with Optimality Guarantees. In *Robotics: Science and Systems*.
[34] Pavel Surynek. 2010. An Optimization Variant of Multi-Robot Path Planning Is Intractable. In *AAAI*. AAAI Press.
[35] Hanlin Wang and Michael Rubenstein. 2020. Walk, Stop, Count, and Swap: Decentralized Multi-Agent Path Finding With Theoretical Guarantees. *IEEE Robotics Autom. Lett.* 5, 2 (2020), 1119–1126.
[36] Rui Wang, Kai Gao, Daniel Nakhimovich, Jingjin Yu, and Kostas E Bekris. 2021. Uniform Object Rearrangement: From Complete Monotone Primitives to Efficient Non-Monotone Informed Search. *arXiv preprint arXiv:2101.12241* (2021).
[37] Richard M Wilson. 1974. Graph puzzles, homotopy, and the alternating group. *Journal of Combinatorial Theory, Series B* 16, 1 (1974), 86–96.
[38] Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz. 2008. Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. *AI Mag.* 29, 1 (2008), 9–20.
[39] Jingjin Yu. 2016. Intractability of Optimal Multirobot Path Planning on Planar Graphs. *IEEE Robotics Autom. Lett.* 1, 1 (2016), 33–40.
[40] Jingjin Yu. 2018. Constant Factor Time Optimal Multi-Robot Routing on High-Dimensional Grids in Mostly Sub-Quadratic Time. *arXiv preprint arXiv:1801.10465* (2018).
[41] Jingjin Yu. 2020. Average case constant factor time and distance optimal multi-robot path planning in well-connected environments. *Auton. Robots* 44, 3-4 (2020), 469–483.
[42] Jingjin Yu and Steven M. LaValle. 2013. Structure and Intractability of Optimal Multi-Robot Path Planning on Graphs. In *AAAI*. AAAI Press.
[43] Jingjin Yu and Daniela Rus. 2014. Pebble Motion on Graphs with Rotations: Efficient Feasibility Tests and Planning Algorithms. In *Workshop on the Algorithmic Foundations of Robotics, WAFR*. 729–746.