Adapted Conflict Detection for Conflict Based Search

Kollakidou, Avgi; Bodenhagen, Leon

Published in:
Proceedings of the 16th International Conference on Agents and Artificial Intelligence - (Volume 1)

DOI:
10.5220/0012438000003636

Publication date:
2024

Document version:
Final published version

Adapted Conflict Detection for Conflict Based Search

Avgi Kollakidou[a] and Leon Bodenhagen[b]
Mærsk Mc-Kinney Møller Institute, University of Southern, Denmark

Keywords: Conflict Based Search, Multi-Agent Path Finding, Navigation.

Abstract: Mobile robots are increasingly deployed in various applications, including autonomous vehicles and logistics. Conflict-Based Search (CBS) is a promising approach for Multi-Agent Path Finding (MAPF), but has limitations when applied to real-world scenarios. This paper explores the challenges of adapting CBS to real-world mobile robotics, focusing on additional conflicts caused by imperfect navigation. We propose an Adaptive Conflict Detection (ACD) approach that proactively identifies conflicts within a rolling time window, making CBS more suitable for real-world applications. Both virtual and real robots are used to evaluate the importance of an adaptation to CBS if adapted to real scenarios. Experimental results show that ACD outperforms traditional CBS when penalties for conflict resolution are applied, demonstrating its potential for improved performance and reliability in practical multi-agent path planning applications.

1 INTRODUCTION

Mobile robots have found their way into an ever-expanding array of applications, from autonomous vehicles and traffic management to automated warehouse logistics and robotics. With their increasing deployment in real-world scenarios in particular unstructured environments, we encounter a host of complex challenges that push the boundaries of conventional navigation algorithms.

One of the fundamental problems in multi-robot systems is Multi-Agent Path Finding (MAPF), which involves finding collision-free paths for a group of agents operating simultaneously within a shared environment. The input to the MAPF problem is an undirected, unweighted graph $G = (V,E)$, and a set of k agents with a starting vertex s_i and target vertex t_i each. The MAPF solution $S = \{\pi_i, ..., \pi_k\}$ is a set of action sequences for each agent $\pi_i = (\alpha_1, ..., \alpha_n)$, with all sets able to be carried out concurrently without conflicts. The complexity of MAPF increases with the number of agents and the intricacy of the environment they must navigate. The search for optimal solutions in such scenarios is notoriously time-consuming and computationally demanding.

Conflict Based Search (CBS) (Sharon et al., 2015) has emerged as a promising two-stage strategy for tackling the MAPF problem. CBS revolves around the identification and resolution of conflicts that arise during the path planning process. While CBS provides an effective framework for MAPF, it is not without its shortcomings, especially when confronted with the realities of the real world.

One significant limitation of CBS becomes apparent when we attempt to adapt it to the complexities of real-world mobile robotics. In the virtual world of CBS, units of time and velocity are abstract concepts that often do not seamlessly translate into the physical reality of real robots. In the real world, factors such as acceleration, deceleration, dynamic obstacles, and varying terrains introduce complexities that challenge the assumptions made by traditional CBS algorithms, as can also be seen in fig. 1.

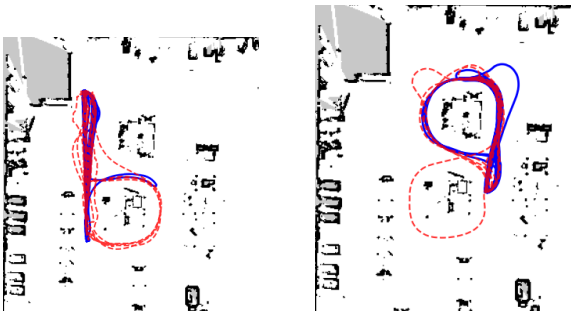
These disparities indicate the necessity of reevaluating and enhancing CBS for real-world mobile robotics applications. We must address the limitations stemming from the mismatch between the perfect model and the physical world. Our goal is to develop solutions that account for the nuances of real robots and unpredictable real-world scenarios.

In this work, we delve into the inherent limitations of CBS when applied to real-world scenarios, particularly focusing on the challenge of additional conflicts occurring due to imperfect navigation.

[a] https://orcid.org/0000-0002-0648-4478
[b] https://orcid.org/0000-0002-8083-0770

Kollakidou, A. and Bodenhagen, L.
Adapted Conflict Detection for Conflict Based Search.
DOI: 10.5220/0012438000003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 1*, pages 367-373
ISBN: 978-989-758-680-4; ISSN: 2184-433X



(a) Mission 3. (b) Mission 5.

Figure 1: Robot trajectories in lab experiments (2 out of 6 missions). Blue: single robot runs; Red: multi-robot runs.

2 RELATED WORK

The field of Multi-Agent Path Finding (MAPF) has evolved considerably in recent years, driven largely by the expanding role of mobile robotics across various domains, such as warehouse logistics (Wurman et al., 2008) or video games (Ma et al., 2017). In this context, Conflict-Based Search (CBS) has been a method for addressing the MAPF problem. This section examines the current scope of CBS, its limitations, and the emerging research trends that emphasise the necessity to tailor CBS for the intricate demands of real-world mobile robotics as well as strategies for real time conflict avoidance.

CBS (Sharon et al., 2015) is an optimal solution to the MAPF problem and is divided into two levels each having a distinct task: individual path planning, and constraint definition from detected conflicts. On the low level, each agent searches for a path individually and independently, ignoring all other agents' positions or plans, using the A* algorithm. A list of constraints, derived from previously existing conflicts, is provided by the upper-level. For the first low level search, an empty list of constraints exists. If a node is generated where agent α_i needs to occupy vertex v at time t and a constraint exists that involves the same configuration, that node is discarded. Likewise, if agent α_i is to traverse edge e at time t and a constraint exists involving that edge, the node is also discarded. However, the method encounters substantial challenges when transposed into real-world scenarios.

Primary among these challenges is the abstraction of unit time and velocity in CBS. In theoretical models, actions are assumed to occur in fixed, discrete time steps, a simplification that does not translate well to the real world, where mobile robots exhibit varied velocities and motions are continuous. This discrepancy limits CBS's direct applicability in practical settings, where factors like acceleration, deceleration, and dynamic obstacles play a significant role. Several works have attempted to bridge this gap by expanding CBS into continuous-time frameworks e.g. with Increasing Cost Tree Search (Walker et al., 2021) or by calculating safe intervals representing time periods when an agent can occupy a vertex without colliding with moving obstacles (Andreychuk et al., 2021). These, however, have not been adequately tested in the real world.

Planning for multiple robots while considering synergies and their temporal dependencies was also considered with promising results (Jiang et al., 2019). The paper assumes that robots can communicate continuously and cost-free, which presents challenges for practical implementation in the real world.

Traditional navigation techniques, delegate the tackling of any agents that might be encountered when navigation to their local planners. Local planning is of course a large research field with many approaches. For example, the Dynamic Window Approach (DWA) is known for its fluid movement in trajectory planning, though it faces challenges in efficiency due to limited perception of obstacle density (Fox et al., 1997). The Enhanced Vector Field Histogram, a refined version of the Vector Field Histogram (VFH), demonstrates notable effectiveness in obstacle avoidance, with enhancements like VFH+ ensuring smoother trajectories and greater reliability (Borenstein et al., 1991; Ulrich and Borenstein, 1998).

Furthermore recent works use learning-based strategies in multi-robot navigation. Integration of deep learning with traditional robotic control strategies to enhance efficiency and adaptability in dynamic environments is proposed in (Han et al., 2022). This is a distributed approach combining the reciprocal velocity obstacle (RVO) concept with deep reinforcement learning, offering a solution for reciprocal collision avoidance in complex scenarios was introduce. A combined model utilising convolutional neural networks and graph neural networks, enabling effective communication and decision-making among robots was suggested for decentralised path planning (Li et al., 2020). (Bae et al., 2019) proposes a reinforcement learning-based path planning method using Deep Q-learning combined with CNNs.

CBS provides a solid theoretical foundation for MAPF, but its idealised assumptions necessitate significant modifications to effectively address the challenges posed by real-world conditions such as continuous-time motion, dynamic environments, and complex terrains. The ongoing research in this field suggests that bridging the theoretical-practical divide in CBS is an area ripe for exploration, offering

substantial opportunities for advancements in mobile robotics deployed in real-world settings.

In the following sections, we delve deeper into the inherent limitations of CBS in practical scenarios, particularly emphasising the additional conflicts arising due to imperfect navigation. Our methodology aims to address these limitations, and our experimental findings offer insights into developing more robust and adaptable solutions for real-world mobile robotics.

3 METHODOLOGY

In this section, we delve into the real-world limitations of Conflict-Based Search (CBS) as applied to Multi-Agent Path Finding (MAPF). CBS relies on certain assumptions which often diverge significantly from the complexities of the real world. To assess the impact of these deviations, we conducted experiments with MiR200 robots, exploring how multiple robots assigned different assembly missions affect task completion times. We also introduced noise elements to replicate real-world irregularities and propose an adaptive conflict resolution method. This approach considers a time window to proactively detect and address conflicts, making CBS more suitable for real-world scenarios. Furthermore, we discuss the possibility of dynamically determining the rolling window size based on map topology, path cost, and the number of robots, to optimise coordination in various settings.

3.1 CBS Limitations in the Real World

MAPF depends on several assumptions: Time is discretised in unit time steps during which, an agent can occupy one vertex and each vertex can accommodate only a single agent. Two types of actions can be performed with unit cost: move to an adjacent vertex, or wait in the current vertex. Reality, however, differs substantially. Robots do not move with constant velocity or at unit time. Additionally, the grid occupancy of an agent is usually larger than a single grid cell, as not only is its footprint larger, but also a buffer area around the robot is considered for safety. If another robot or an obstacle is within this area, the robots in the real world will engage the emergency break and stop moving.

To assess how much robots deviate from their planned trajectories in the real world, we conducted a series of experiments to investigate the navigational performance of agents in controlled lab conditions. The experimental environment was designed to replicate a flexible assembly setting featuring multiple

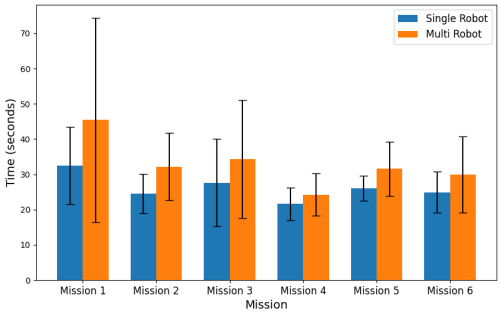


Figure 2: Mean of mission completion times for single and multi-robot runs with their respective standard deviations.

workstations. We used MiR200 robots equipped with the 3.12 software version. The primary objective was to investigate how the introduction of multiple robots, each assigned different assembly missions, influenced the overall running time of the tasks. Our experimental setup involved two key scenarios: one with a single robot executing all missions sequentially, and another with three robots, each independently assigned missions. Both scenarios were repeated 10 times to allow for deviations, and the average time was calculated. The aim was to assess the impact of multiple robots working simultaneously within the same space.

For the second scenario, we introduced three robots, with each robot assigned random assembly missions. These missions were randomly assigned, and the planning of the path to accomplish each mission remained consistent with the single robot scenario. We monitored and documented the time taken by each robot to complete their respective missions.

The trajectories of an agent during single-robot (SR) and multi-robot (MR) runs can be seen in fig. 1 for two different missions. Trajectories for SR runs are very similar but not identical which can then lead to conflicts as timings vary. Trajectories of MR runs include larger deviations as well as detours. The effect of such deviations can be seen on the mission completion times fig. 2.

3.2 Noise Introduction

In order to replicate real-world irregularities in simulation, noise was introduced in the agents' paths. This noise was incorporated by introducing waiting actions. At each step along the path a vertex had the possibility of being repeated with a specified probability. For the purposes of our experiments, we set this probability at 10%. As the identical goal vertex is inserted in the plan, the robot waits for one time-step at said vertex. This intentional inclusion of noise allowed us to more accurately replicate the unpredictable nature of real-world scenarios and assess

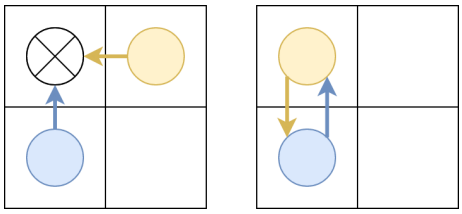


Figure 3: Conflicts definitions: Vertex conflict (left), Edge conflict (right).

the robots' planning in challenging, dynamic environments.

3.3 Adaptive Conflict Resolution

Considering the previous experiments, where robots encountered conflicts despite optimal path planning, we recognise the need to adapt existing coordination and conflict resolution methods to better address real-world scenarios. In particular, we propose an adaptive approach based on Conflict-Based Search (CBS), which considers the inherent uncertainties and dynamic nature of robot coordination in practical settings.

In traditional CBS, conflict detection typically occurs for a single time instance, and assumes perfect execution of robot plans. There are several types of MAPF conflicts discussed in the literature (Stern et al., 2019b). In this work, we consider two types: vertex and edge conflicts (fig. 3). A vertex conflict arises when two agents, a_i and a_j, are planned to occupy the same vertex, v, during the same time step and is defined as $C_v = \langle a_i, a_j, v, t \rangle$. An edge conflict occurs when two agents are planned to traverse the same edge, connecting vertex v_1 and v_2, during the same time step and is defined as $C_e = \langle a_i, a_j, v_1, v_2, t \rangle$. While this approach is well-suited for perfect scenarios, it often falls short in addressing real-world complications, such as minor deviations from planned paths, variations in robot speed, or dynamic environmental factors.

3.3.1 Rolling Window Conflict Detection

To enhance the CBS framework's adaptability to real-world conditions, we introduce the concept of Rolling Window Conflict Detection. This method involves observing the robots' plan within a time window, rather than at a single instance. The rolling window allows us to monitor the robots' movements and identify potential conflicts that might arise due to imperfect plan execution.

By increasing the time period where conflicts are considered, we can proactively detect conflicts and initiate conflict resolution strategies in advance. This adaptability to real-time conditions mitigates the impact of conflicts and should minimise delays upon execution. Additionally, reducing the conflicts that the robots will encounter upon execution, will increase reliability and consistency. Execution consistency is a well sought after attribute for mobile robots that should be deployed along humans, as they increase the trust in the agent by the user as well as bystanders.

Assuming:

s – is the window size.

α_i – is the agent of interest.

v – is the vertex where agent α_i is positioned.

t – is the current time.

We expand the search for vertex conflicts as follows:

$$C\langle \alpha_i, v, t, s \rangle = \bigcup_{\tau=t-\frac{s-1}{2}}^{t+\frac{s-1}{2}} C_v = \langle a_i, a_j, v, t \rangle \quad (1)$$

Where:

τ – is a time variable that iterates through the time interval from $t - \frac{s-1}{2}$ to $t + \frac{s-1}{2}$.

Conflicts(α_i, v, τ) – represents the set of conflicts that may arise at time τ between agent.

3.3.2 Dynamic Rolling Window Size

We propose a dynamic determination of the rolling window size to the specific characteristics of the robot coordination scenario. This adaptability is achieved by considering a combination of map topology, path cost, and conflicts arising from the inclusion of noise in the robot's plan.

- **Map Topology:** The rolling window size is related to the map's size. In scenarios with larger and sparser workspaces, a smaller rolling window may be preferable, as robots should come into fewer conflicts. Conversely, in more intricate or congested workspaces, a larger rolling window might be suitable, as robots encounter a higher density of potential collision points. By aligning the rolling window size with the map's complexity, we can optimise robot coordination in different environments.
- **Path Cost:** The cost of the planned paths also plays a crucial role in determining the rolling window size. Higher-cost paths may necessitate a larger rolling window, as they can pose a greater risk of conflict due to longer traversal times. Conversely, lower-cost paths may allow for smaller rolling windows, as the paths are inherently less likely to result in conflicts. By adapting the rolling

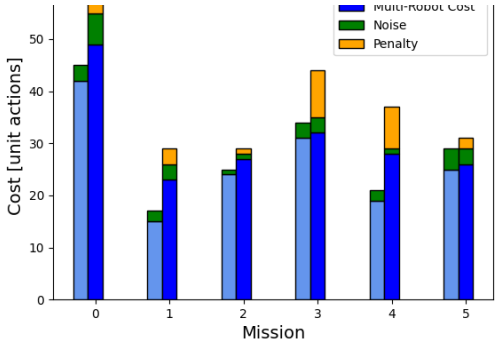


Figure 4: Effects of noise addition and conflict penalties after post-noise conflict detection in multi-robot runs (mean values).

window size to the path cost, we strike a balance between efficiency and conflict prevention.

- **Number of Robots:** in the environment is a crucial factor influencing the rolling window size. In cases with a larger number of robots, a larger rolling window may be beneficial, as it enhances conflict detection in crowded settings. Conversely, with fewer robots, a smaller rolling window can be employed, allowing for less lengthy conflict checks and facilitating smoother coordination among a smaller group.

4 RESULTS

4.1 Noise and Penalties

Following the introduction of noise, the revised paths underwent a comparison to identify any emergent conflicts. This enabled us to replicate real-world scenarios where conflicts may arise as a result of imperfect velocities or time delays. In the event a conflict was detected, a conflict penalty was assigned to the overall cost, to emulate the evading manoeuvre or waiting actions the agent would have to employ to overcome the detected conflict. For the purpose of this evaluation, the conflict penalty value was set to 3.

Figure 4 shows the effect the inclusion of noise has on the path costs of agents. Six simulated missions were used, with varying lengths and complexities as seen in fig. 5. The missions were executed 10 times and the reported numbers represent their mean. The optimal path cost for each mission is presented, along with the cost for the same agent when navigating in the environment along other agents. The included noise for both sessions as well as the deduced conflict penalty is shown to demonstrate the importance of a more robust CBS solution when intended for the real world. The mean penalties assigned amount to up to 21.6% of the final path cost (minimum: 3.5%).

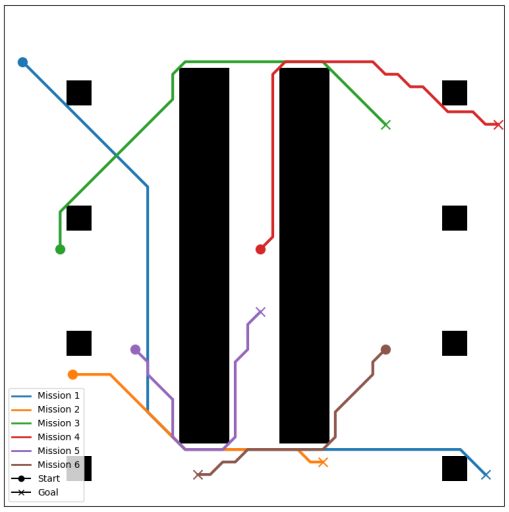


Figure 5: Missions used for evaluation and optimal paths derived with A*.

4.2 Adaptive Conflict Detection

In order to evaluate the impact of the Adaptive Conflict Detection approach on multi-agent path planning, a series of experiments were conducted on a dataset of simulated missions. The experiments were conducted on a simplified map scenario produced by us as well as the MAPF dataset (Stern et al., 2019a), which provides a diverse set of benchmark scenarios for MAPF. These scenarios encompass a wide range of complexities, map topologies, and mission characteristics, making them a valuable resource for assessing the performance of path planning algorithms in diverse and challenging real-world-inspired environments. In this section, we present and discuss the results of these experiments, and potential benefits of the Adaptive Conflict Detection (ACD) approach in real-world MAPF applications.

Initially, the ACD approach showed, as expected, a higher path cost compared to traditional CBS, primarily due to the increased number of detected conflicts. The rolling window approach yielded higher path costs at a maximum of 18% increase. However, with the introduction of noise, and the penalties resulting from the subsequent search for arising conflicts, the ACD approach demonstrated superior performance. Conflict penalties applied on the multi-robot path costs resulting from simple CBS were up to a mean of 27.9% of the initial path cost for mission 1 and a minimum mean of 18.6% for mission 2, whereas the maximum penalty applied on a path

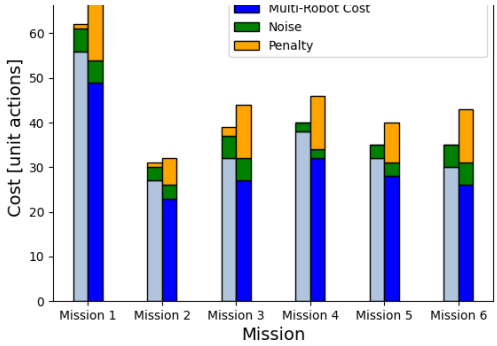


Figure 6: Comparison of CBS with rolling window conflict detection against basic CBS when noise is included.

resulting from the ACD approach was 5.1% for mission 3 (minimum 0%) (fig. 6). This outcome suggests that the ACD approach, with its adaptability and early conflict detection capabilities, is well-suited for real-world scenarios where deviations from planned paths and unexpected factors can impact multi-agent coordination.

4.3 Conflict Penalty Size

In the evaluation of the Adaptive Conflict Detection approach, an essential aspect was the determination of an appropriate conflict penalty size. Three distinct considerations were taken into account to make an informed decision regarding penalty values.

Firstly, a conservative conflict penalty of 3 was considered in simulation to avoid collisions during a conflict as the agent would need to a) detect the conflict, b) move to avoid it and c) return to its original path.

Secondly, the effect a conflict has on mission completion times in simulation was considered. This was calculated by comparing the increased mission completion time of agents when considering the multi-robot scenarios against the optimal solution of single-agent runs, divided by the amount of conflicts that occurred.

Finally, the time taken in real-world experiments to overcome conflicts were calculated. The variance of the additional actions required to complete a mission in case of a conflict was very large, spanning from 15% increase of the path length to 100% doubling the distance travelled by the agent.

With all three considerations taken, the conservative conflict penalty was used to avoid inflating the advantages of ACD, but it is evident that such a penalty in the real world would be larger.

5 DISCUSSION

The effectiveness of our Adapted CBS in real-world scenarios is depended on map topology. The complexity and layout of the environment, such as narrow passageways, play an important role in both the performance and applicability of the algorithm. In densely populated or cluttered spaces the frequency and complexity of conflicts increase, thereby increasing the need for conflict resolution and requires larger window sizes. On the other hand, simpler maps with large open spaces, would, in theory, feature fewer conflicts and therefore the need of adaptation might be smaller. A consideration for applying the method locally in critical areas might be worthwhile.

The mission selection is another critical factor that could potentially influence our results. The inherent complexity of the chosen missions - whether they are relatively straightforward or particularly challenging, clustered together in certain areas, or spread out through the map - directly affects the performance of the adapted CBS methodology. Simple missions, with direct paths and minimal obstacles, may not adequately test the robustness and adaptability of the algorithm, potentially leading to an overestimation of its effectiveness in more complex scenarios. On the other hand, excessively difficult missions could push the system to its limits, potentially highlighting weaknesses that are less pronounced in standard operational conditions. In our study, we attempted to choose missions that were neither too easy nor too hard. This way, we could properly test our system without pushing it too far. Our goal was to see how well the algorithm works under realistic conditions, but without making the tasks overly complicated.

6 CONCLUSIONS

In this study, we conducted an evaluation of an Adaptive Conflict Detection (ACD) approach for multi-agent path planning, aimed at addressing the complexities and uncertainties of real-world scenarios. Our experiments revealed insights into the adaptability and effectiveness of the ACD approach.

The introduction of noise in our simulations, mimicking real-world irregularities, initially led to an increased path cost as conflicts were detected more frequently. However, our approach, with its ability to adapt and proactively identify conflicts within a rolling time window, demonstrated its true potential when penalties for conflict resolution were applied. These penalties, chosen through a combination of practical considerations, showed that the

ACD approach outperformed traditional Conflict-Based Search (CBS) in real-world-inspired scenarios, where deviations from planned paths and unexpected factors could influence multi-agent coordination.

In our exploration of penalty size, we considered multiple factors, including conservative values, simulated mission impact, and insights from real-world experiments. This comprehensive approach to penalty selection ensures that the penalties are not exaggerated purposefully to magnify the approach's advantages but show a fair if not understated estimate of the method's effect.

In summary, the results of our experiments suggest that the Adaptive Conflict Detection approach is a promising candidate for addressing the challenges of real-world multi-agent path planning. By proactively detecting and resolving conflicts, this approach showcases its potential for improved performance and reliability in practical applications.

ACKNOWLEDGEMENTS

This work is supported by the Innovation Fund Denmark for the project DIREC (9142-00001B) and the SDU Industry 4.0Lab.

REFERENCES

Andreychuk, A., Yakovlev, K., Boyarski, E., and Stern, R. (2021). Improving continuous-time conflict based search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35.

Bae, H., Kim, G., Kim, J., Qian, D., and Lee, S. (2019). Multi-robot path planning method using reinforcement learning. *Applied sciences*, 9(15):3057.

Borenstein, J., Koren, Y., et al. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE transactions on robotics and automation*, 7(3):278–288.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Han, R., Chen, S., Wang, S., Zhang, Z., Gao, R., Hao, Q., and Pan, J. (2022). Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. *IEEE Robotics and Automation Letters*, 7(3):5896–5903.

Jiang, Y., Yedidsion, H., Zhang, S., Sharon, G., and Stone, P. (2019). Multi-robot planning with conflicts and synergies. *Autonomous Robots*, 43:2011–2032.

Li, Q., Gama, F., Ribeiro, A., and Prorok, A. (2020). Graph neural networks for decentralized multi-robot path planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11785–11792. IEEE.

Ma, H., Yang, J., Cohen, L., Kumar, T., and Koenig, S. (2017). Feasibility study: Moving non-homogeneous teams in congested video game environments. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 13.

Sharon, G., Stern, R., Felner, A., and Sturtevant, N. R. (2015). Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219.

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. (2019a). Multi-agent pathfinding: Definitions, variants, and benchmarks. *Symposium on Combinatorial Search (SoCS)*.

Stern, R., Sturtevant, N. R., Felner, A., Koenig, S., Ma, H., Walker, T. T., Li, J., Atzmon, D., Cohen, L., Kumar, T. S., et al. (2019b). Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Twelfth Annual Symposium on Combinatorial Search*.

Ulrich, I. and Borenstein, J. (1998). Vfh+: Reliable obstacle avoidance for fast mobile robots. In *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, volume 2, pages 1572–1577. IEEE.

Walker, T. T., Sturtevant, N. R., Felner, A., Zhang, H., Li, J., and Kumar, T. S. (2021). Conflict-based increasing cost search. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31.

Wurman, P. R., D'Andrea, R., and Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1).