

CRH*: A Deadlock Free Framework for Scalable Prioritised Path Planning in Multi-Robot Systems

James R Heselden¹[0000-0001-6494-4981] and
Gautham P Das¹[0000-0001-5351-9533]

University of Lincoln, Lincoln, Lincolnshire, UK {jheselden,gdas}@lincoln.ac.uk

Abstract. Multi-robot system is an ever growing tool which is able to be applied to a wide range of industries to improve productivity and robustness, especially when tasks are distributed in space, time and functionality. Recent works have shown the benefits of multi-robot systems in fields such as warehouse automation, entertainment and agriculture. The work presented in this paper tackles the deadlock problem in multi-robot navigation, in which robots within a common work-space, are caught in situations where they are unable to navigate to their targets, being blocked by one another. This problem can be mitigated by efficient multi-robot path planning. Our work focused around the development of a scalable rescheduling algorithm named Conflict Resolution Heuristic A* (CRH*)¹ for decoupled prioritised planning. Extensive experimental evaluation of CRH* was carried out in discrete event simulations of a fleet of autonomous agricultural robots. The results from these experiments proved that the algorithm was both scalable and deadlock-free. Additionally, novel customisation options were included to test further optimisations in system performance. Continuous Assignment and Dynamic Scoring showed to reduce the make-span of the routing whilst Combinatorial Heuristics showed to reduce the impact of outliers on priority orderings.

Keywords: Multi-robot path planning · Prioritised planning · Decoupled path planning · A* · Reservation tables.

1 Introduction

Autonomous mobile robotic technologies have matured over recent decades enabling their uses in many real-world applications. This has resulted in a push towards scaling up systems to large mobile robot fleets to improve operational efficiency, especially when tasks are inherently distributed in space, time or functionality. One of the primary requirements to ensure proper coordination among a fleet of autonomous mobile robots in a shared-workspace, is efficient path planning and allocation. Without effective coordination, interference between robots

¹ https://github.com/iranaphor/crh_star

can have a detrimental effect on the fleet operations. Congestion-free route planning and coordination is needed to address this interference.

In Multi-Robot Path Planning (MRPP), there are four main challenges to overcome: *Completeness*, is the guarantee that a route will be found if one exists; *Optimality*, is the guarantee to find a set of routes minimising some metric such as make-span (time the last robot arrives) or flow-time (sum of all route lengths) [19]; *Deadlocking*, is where a robot will be in a state preventing another from reaching a target [8, 5]; and *Scalability*, is the issue of processing resources thinning as the complexity of the joint state-space increases. Among these, optimality and scalability can be considered mutually exclusive, and are tackled well by Coupled and Decoupled approaches respectively. An example of a scenario where MRPP can be useful is a fleet of agricultural robots deployed in a polytunnel environment to execute a series of in-field logistics tasks [7]. These tasks are dynamic in nature and are dispersed over the environment. This paper addresses the MRPP problem to enable congestion and deadlock-free movement of robots in such a polytunnel environment, by proposing a novel algorithm called Conflict Resolution Heuristic A* (CRH*). By relying on decentralised and decoupled route planning to find sub-optimal solutions, the CRH* algorithm is complete and scalable, as demonstrated by empirical evaluations.

The rest of this paper is organised as follows: Section 2 provides an overview of related work in Prioritised Planning; Section 3 provides an overview of the implementation; Section 4 details the experimental setup, evaluation and results; and Section 5 concludes our findings.

2 Related Works

Decentralised and decoupled MRPP approaches are widely preferred to address the high computational complexity and low scalability of centralised approaches. Prioritised Planning (PP) is a decoupled MRPP approach which assigns priorities to each robot and then plans a route for each one sequentially, treating all prior robots as dynamic obstacles.

As the complexity of the joint state-space is proportional to the total configuration of priorities, finding the optimal assignment of priorities is a NP-Hard problem. Thus a full state-space search is unfeasible and smarter approaches must be taken [2]. When routes are unavailable due to blocking from higher priority agents, the robot can be considered to be deadlocked. Rescheduling can be used to optimise priority assignments and replan routes. CRH* tackles the area of rescheduling, making use of enhanced reservation tables and comprehensive replanning in a distributed decision-making topology.

Our approach focuses in the topological domain, in which the map is a finite set of discrete positions (nodes) and connections between them (edges) indicating possible paths from one node to another [10]. It is used as the common discretised environment representation over which all robots plan their routes. Such representations have low planning complexity and can easily detect possible conflicts compared to metric maps.

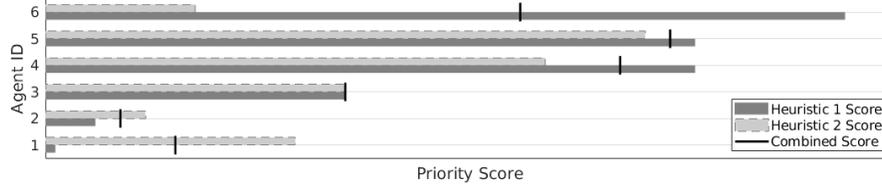


Fig. 1: Example of the potential of combinatorial heuristics, as the combination of two independent heuristics.

2.1 Heuristics

The effective use of heuristics works with the intent to give priority to the robot which needs it the most. In *Static Ordering* (SO) [9] scoring is made using the order the agents were added to the network, and in *Euclidean Distance* (ED) [17] scoring is based on the distance to the target.

These work well in metric maps which are very open and well-connected, however for agricultural environments in the scale of hectares, these are not viable as they are. Polytunnels and fields often include long isolated paths to traverse, which are better managed by approaches such as below, which follow the idea that agents struggle due to environmental constraints. ED can be adapted into *Optimal Path Length* (OPL) to become viable in these scenarios.

In *Planning Time* (PT) [18], the processing time to find a route is used for scoring, however this falls short in heterogeneous systems where platforms have differing processing hardware. In *Naive & Coupled Surroundings* [6, 19] agents are scored based on the cluttering of their workspace by counting the number of local obstacles. Where in *Naive Surroundings* (NS), obstacles are regarded as distinct and in *Coupled Surroundings* (CS) they are treated as effective, where effective obstacles are regions the agent is unable to navigate. In *Path Prospects* (PaP) [19] scoring is made based on the number of effective obstacles between the start and target to score on the number of paths available in the homology class of trajectories.

In PP, it is standard to use a single heuristic to assign priorities. We extend on this, exploring the potential of combining multiple heuristics to improve to handling of outliers demonstrated in Fig.1.

2.2 Rescheduling

Rescheduling works to reduce the impact of deadlocks by modifying the priority schema to optimise generated orderings [9]. In *Random rescheduling* [4] priorities are randomised and replanned whenever a deadlock occurs. The *Hill-Climbing* search [3] extends on this by randomly swapping pairs of priorities. *Continuous Enhancement* [15] allows agents to modify their own score if they are unable to find a route. This is extended by *Deterministic Rescheduling* [1], to award the maximum priority, following the idea that issues are caused by their local environment. *Local Priority Assignment* [14] works by exploring all local priority

configurations in proximity to an agent unable to find a route. *Priority Tuning* [13] optimises successful assignments by shuffling priorities between the least optimal agents, repeating until convergence. In our approach, we extend the ideas in Deterministic Rescheduling by integrating the intelligent overriding of higher priorities.

2.3 Path Finding

Path Finding algorithms are a core component within prioritised planning, being used to generate routes once priorities are assigned. As detailed in [11, 16], there are many categories of path finding algorithms. Being one of the most fundamental path finding algorithms, A^* [12] works only to identify a route. *Local Repair A^** (LRA*) [20], recalculates the remainder of its route when a collision is pre-empted. *Cooperative A^** (CA*) [16], uses three dimensional space-time reservation tables to mark off impassable regions. Our approach includes an extension to the reservation tables of CA* to override existing reservations; and an extension to LRA* to optimise replanning from a point of conflict.

3 Design and Development

3.1 Overview

The main architecture of CRH* (Fig.2), works in three stages, with the final two repeating together till convergence. In the first stage, the robot is assigned a target, plans a route, and informs the coordinator of its reservations. The coordinator, on receiving the route, adds the reservations to the global map, notifying any other robots of overturned reservations in the second stage. The robots each receive their failed reservations (FR) in the third stage, and perform replanning to identify new routes, passing these new routes to the coordinator.

Each message contains a list of reservations, of which each consist of an edge identifier, agent identifier, and a reservation start and end time.

On receiving an FR, the shortest of three replanning methods is returned to the coordinator. These three methods are: *Replanning from Start* where a new route is generated from scratch; *Replanning from Conflict* where the route beyond FR is replanned; and *Replanning with Delay* which adds a time delay (in which the robot will wait to use the FR once it is available) and replans from there.

3.2 A^* Adaptations

There were two key modifications to the motion planning algorithm to enable the enhancements for our approach, the first was the reservation override system (CRH Battle), the second was the deadlock management (FAILED List).

The CRH Battle was included within A^* where as each edge is initially investigated, a score (referred to as the CRH score) is generated to query against

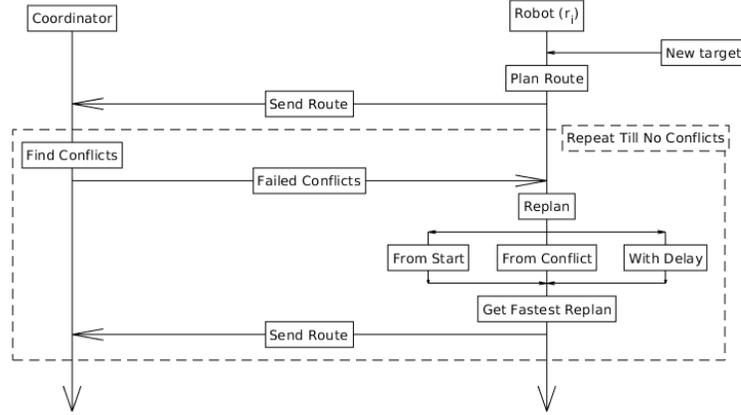


Fig. 2: Visual representation of robot-coordinator communication.

any existing reservations in the time period specified. Only if the robot’s CRH score is greater, will the override work. While this scoring system can utilise any single heuristic such as described in Section 2.1, it is also capable of taking the combination of multiple heuristics (e.g. ED , $\sum(ED, OPL)$, or $\prod(PT, NS, CS)$). For our experiments, we utilise a variety of scoring systems as the heuristics themselves are arbitrary to the aim of the research.

The deadlock management is implemented in the form of an additional flag for exploration. A* has two flags, OPEN and CLOSED, which define whether a node is on the frontier for exploration or has already been explored, where if the OPEN list is exhausted, the planning fails. In our approach, when reservation overrides fail, the edge is added to a new FAILED list, which is accessed once the OPEN list is exhausted to obtain the most optimal edge to override, boosting the CRH Score generated.

3.3 Framework Facilities

Continuous Assignment In PP, the handling of new assignments is done in one of two ways, in *Batch Assignment* replanning will wait till all have completed routes, whilst in *Full Replanning* all agents replan in the current state. In our approach, we propose a third method *Continuous Assignment* as a direct contrast to Batch Assignment in which rescheduling is applied without replanning the entire network, only updating affected agents. This allows for a reduction in make-span as shown in Fig.3.

Dynamic Scoring The standard method of score generation is *Static*, in which scores for each agent are assigned before routing. In our approach, we implement the concept of dynamic scoring, where every reservation uses locally relevant information to compute heuristics. As the routing gets closer to a target, each edge will be reserved with information relevant to that edge rather than information from the agents current location, for example with ED, each edge will

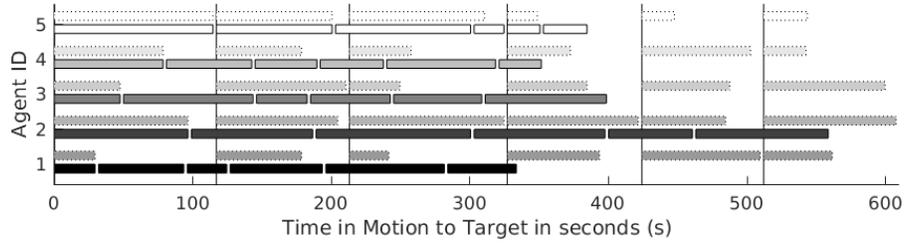


Fig. 3: Example of route times in batch (upper) and continuous (lower) assignment for a series of agents.

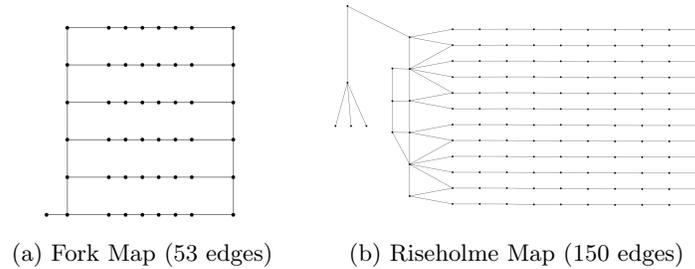


Fig. 4: Topological Maps used in the experiments.

be reserved with its distance to the target rather than the distance to the target from the start node as would happen with static scoring.

4 Experiments

Discrete Event Simulation To ensure a comprehensive evaluation, Discrete Event Simulation (DES) was utilised along with Monte-Carlo simulations, enabled through the deterministic nature of the approach. In this, time jumps to points of interest (POI) rather than evaluating every timestep. In our experiments, these POI consist of any timestep in which a route is completed; time jumps to the next route completion, performs any calculations, and repeats.

Experiment Maps Experimentation was completed on two topological maps Fork (Fig.4a) and Riseholme (Fig.4b), representing a single polytunnel and a pair of polytunnels respectively. The experimentation was completed on the Fork map for the majority of experiments, with the Riseholme map used also for the scalability test.

Heuristics In the following experiments, used to generate the CRH scores for resolving conflicts are the heuristics of: Optimal Path Length (OPL), Euclidean Distance (ED), Planning Time (PT), and Static Ordering (SO) of which we utilise the Agent's ID.

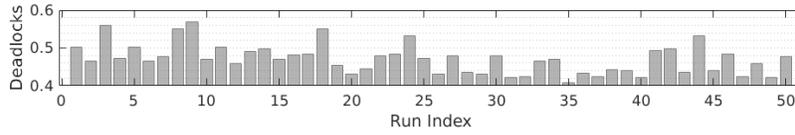


Fig. 5: Average number of deadlocks resolved per run. Completed with 10 agents, across 50 iterations of 1k targets in the Fork Map.

4.1 Evaluation of CRH*

Deadlocks The FAILED list, is used to handle every deadlock encountered, so by recording the activity of the FAILED list, we are able to identify how often potential deadlock situations occur. Fig.5 shows the average number of deadlocks encountered per target over 50 runs with 1000 targets per run. These results show on average 0.47 deadlocks for each given target, which if not managed as they are here, would cause many delays in navigation.

Scalability As the approach is decentralised in design, path planning is expected to run distributed, thus does not contribute to scalability concerns. The area of highest concern is thus communication, which consists of one message per replan to the coordinator. So to evaluate scalability, the Replans per Target (RT) is recorded, which accurately measures the load on the coordinator. The results in Fig.6 show upward trends initially before plateauing, avoiding exponential scaling and thus can be regarded as scalable.

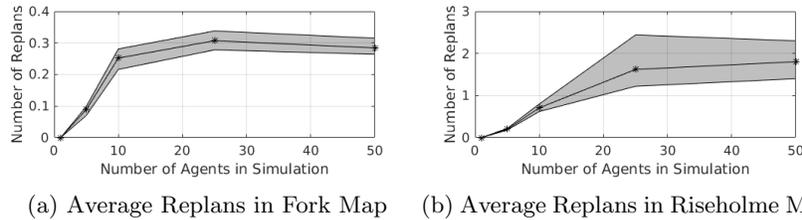


Fig. 6: Range-grouped area charts for average RT over five runs of 50 targets

4.2 Evaluation of Optimisation Improvements

Batch vs Continuous Assignment Continuous Assignment works to improve make-span by reducing idle time for agents waiting to replan. Fig.7a shows the resulting make-span distributions contrasting Batch and Continuous Assignment. The tests were run five times with 20 targets for each of 5, 10 and 20 agents using OPL as the CRH score to resolve conflicts. The results show Continuous Assignment improves make-span in both small and large simulations. This is evidenced by Fig.7b showing the same tests run for 1000 targets with make-span reductions of 47%, 38%, and 20% respectively to the number of agents.

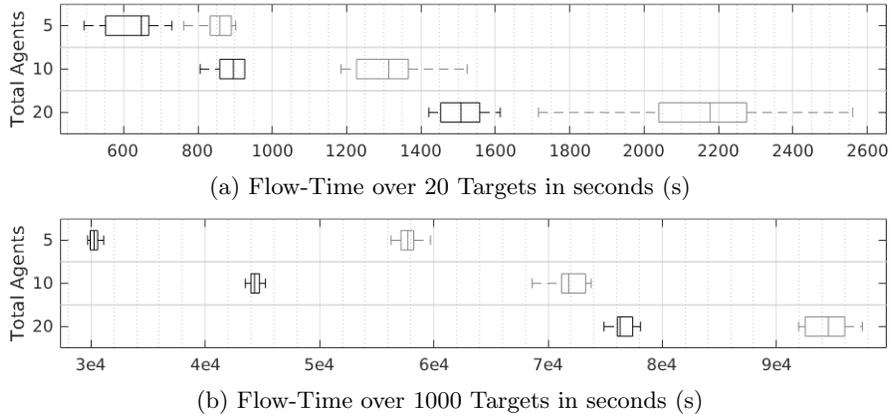


Fig. 7: Make-Span distribution for Continuous (left) and Batch (right) Assignment with simulations of 5, 10 and 20 agents.

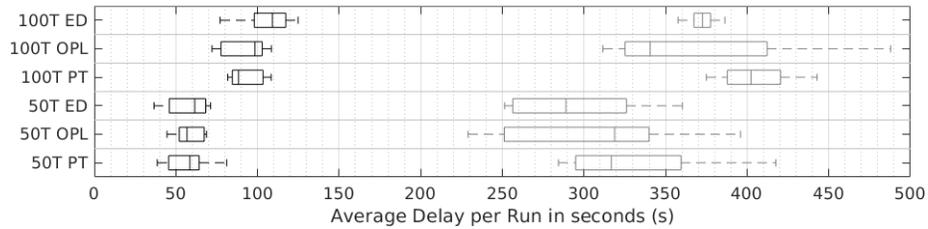


Fig. 8: Distribution of delays contrasting Dynamic (left) and Static (right) Scoring systems with 50 and 100 targets per agent.

Static vs Dynamic Scoring To test the efficacy of the optimisation, the decoupled optimal make-span is negated from the CRH* make-span to get the worst-case delay. This delay was recorded across two sets of experiments, the first testing static scoring, and the second with dynamic scoring. This was repeated across six categories using the heuristics of ED, OPL and PT, for each of 50 and 100 targets per agent. The results in Fig.8 show Dynamic Scoring offers a significant improvement, with the margin between Static and Dynamic Scoring growing larger proportionally to the total targets.

Base vs Combinatorial Heuristics To evaluate the efficacy of combinatorial heuristics, 10 tests were performed with 10 agents each, recording the average number of replans per run. Each test consisted of 10 runs with 20 targets, completed for each heuristic independently then combined together. This is repeated three times, the first comparing ED, OPL and PT; the second using SO and OPL; and the third comparing SO and ED. From Fig.9, the average range of outliers is reduced when using the combination of all heuristics as opposed to purely independent heuristics. In addition, the average number of replans decreases for combined heuristics when using a combination of heuristic and Agent ID.

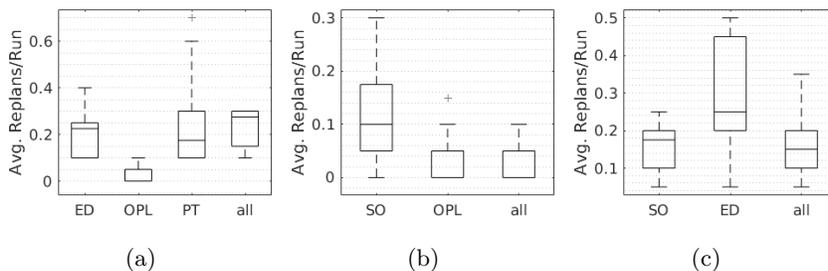


Fig. 9: Distribution of replans between independent heuristics, and their combination (all).

5 Conclusion

In this work, we have proposed the CRH* algorithm, a decentralised and decoupled enhancement to prioritised planning which aimed to improve deadlock avoidance with the use of reservation tables and deterministic rescheduling. We show through experimental evaluation that our approach is both scalable and deadlock-free as the complexity of the joint state-space increases with the size of the topological map and the number of agents.

We have also shown, clear make-span reductions of up to 47% with the use of Continuous Assignment and up to 82% with Dynamic Scoring. We have also shown prioritisation quality improvements with Combinatorial Heuristics reducing the impact of extreme outliers.

Due to the agnostic nature of the approach, the discrete event simulation and specific heuristics chosen are arbitrary to the evaluation of the developed features, however further work will include exploration with a wider sample of heuristics, and performance evaluations beyond discrete event simulation.

References

- [1] A Andreychuk and K Yakovlev. “Two techniques that enhance the performance of multi-robot prioritized path planning”. In: *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS* 3 (2018), pp. 2177–2179. ISSN: 15582914. arXiv: 1805.01270.
- [2] Kianoush Azarm and Günther Schmidt. “Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation”. In: *Proc. Int. Conf. Robot. Autom.* Vol. 4. IEEE, 1997, pp. 3526–3533.
- [3] M Bennewitz, W Burgard, and S Thrun. “Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots”. In: *Rob. Auton. Syst.* 41.2-3 (2002), pp. 89–99.
- [4] M Bennewitz, W Burgard, and S Thrun. “Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems”. In: *IEEE Int. Conf. Robot. Autom.* Vol. 1. 2001, pp. 271–276.

- [5] M Cap et al. “Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots”. In: *IEEE Trans. Autom. Sci. Eng.* 12.3 (2015), pp. 835–849. ISSN: 15455955. arXiv: 1409.2399.
- [6] C M Clark, T Bretl, and S Rock. “Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems”. In: *Proceedings, IEEE Aerosp. Conf.* Vol. 7. IEEE. 2002, p. 7.
- [7] G P Das et al. “Discrete Event Simulations for Scalability Analysis of Robotic In-Field Logistics in Agriculture – A Case Study”. In: *ICRA 2018 Work. Robot. Vis. Action Agric.* Brisbane, 2018.
- [8] Ram Kishan Dewangan, Anupam Shukla, and W Wilfred Godfrey. “Survey on prioritized multi robot path planning”. In: *2017 IEEE Int. Conf. smart Technol. Manag. Comput. Commun. Control. energy Mater.* IEEE. 2017, pp. 423–428.
- [9] M Erdmann and T Lozano-Pérez. “On multiple moving objects”. In: *Algorithmica* 2.1-4 (1987), pp. 477–521.
- [10] J P Fentanes et al. “Now or later? Predicting and maximising success of navigation actions from long-term experience”. In: *Proc. - IEEE Int. Conf. Robot. Autom.* Vol. 2015-June. June. Seattle, USA, 2015, pp. 1112–1117.
- [11] D Ferguson, M Likhachev, and A Stentz. “A guide to heuristic-based path planning”. In: *Proc. Int. Work. Plan. under Uncertain. Auton. Syst. Int. Conf. Autom. Plan. Sched.* 2005, pp. 1–10.
- [12] P.E. Hart, N.J. Nilsson, and B. Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Trans. Syst. Sci. Cybern.* 4.2 (1968), pp. 100–107.
- [13] B Hu and Z Cao. “Minimizing task completion time of prioritized motion planning in multi-robot systems”. In: *IEEE Int. Conf. Syst. Man Cybern.* Vol. 2019-Octob. IEEE, 2019, pp. 1018–1023.
- [14] H Ma et al. “Searching with Consistent Prioritization for Multi-Agent Path Finding”. In: *Proc. AAAI Conf. Artif. Intell.* 33 (2019), pp. 7643–7650. ISSN: 2159-5399. arXiv: 1812.06356.
- [15] R Regele and P Levi. “Cooperative multi-robot path planning by heuristic priority adjustment”. In: *2006 IEEE/RSJ Int. Conf. Intell. Robot. Syst.* IEEE. 2006, pp. 5954–5959.
- [16] D Silver. “Cooperative Pathfinding.” In: *Aiide* 1 (2005), pp. 117–122.
- [17] J P Van Den Berg and M H Overmars. “Prioritized motion planning for multiple robots”. In: *2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS* (2005), pp. 2217–2222.
- [18] P Velagapudi, K Sycara, and P Scerri. “Decentralized prioritized planning in large multirobot teams”. In: *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst.* 2010, pp. 4603–4609.
- [19] W Wu, S Bhattacharya, and A Prorok. “Multi-Robot Path Deconfliction through Prioritization by Path Prospects”. In: (2019), pp. 22–24. arXiv: 1908.02361.
- [20] A Zelinsky. “A mobile robot navigation exploration algorithm”. In: *IEEE Trans. Robot. Autom.* 8.6 (1992), pp. 707–717.