machines

MDPI

Review

Heuristics and Rescheduling in Prioritised Multi-Robot Path Planning: A Literature Review

James R. Heselden [1,2,†] and Gautham P. Das [1,2,*,†]

[1] Lincoln Institute for Agri-Food Technology (LIAT), University of Lincoln, Riseholme Campus, Lincoln LN2 2LG, UK; jheselden@lincoln.ac.uk
[2] Lincoln Centre for Autonomous Systems (L-CAS), University of Lincoln, Brayford Pool Campus, Lincoln LN6 7TS, UK
* Correspondence: gdas@lincoln.ac.uk
† These authors contributed equally to this work.

Abstract: The benefits of multi-robot systems are substantial, bringing gains in efficiency, quality, and cost, and they are useful in a wide range of environments from warehouse automation, to agriculture and even extend in part to entertainment. In multi-robot system research, the main focus is on ensuring efficient coordination in the operation of the robots, both in task allocation and navigation. However, much of this research seldom strays from the theoretical bounds; there are many reasons for this, with the most-prominent and -impactful being resource limitations. This is especially true for research in areas such as multi-robot path planning (MRPP) and navigation coordination. This is a large issue in practice as many approaches are not designed with meaningful real-world implications in mind and are not scalable to large multi-robot systems. This survey aimed to look into the coordination and path-planning issues and challenges faced when working with multi-robot systems, especially those using a prioritised planning approach, and identify key areas that are not well-explored and the scope of applying existing MRPP approaches to real-world settings.

Keywords: coordination; prioritised planning; routing; MRPP

check for updates

Citation: Heselden, J.R.; Das, G.P. Heuristics and Rescheduling in Prioritised Multi-Robot Path Planning: A Literature Review. *Machines* **2023**, *11*, 1033. https://doi.org/10.3390/machines11111033

Academic Editor: Bao Kha Nguyen

Received: 27 September 2023
Revised: 7 November 2023
Accepted: 13 November 2023
Published: 20 November 2023

1. Introduction

Robotics, automation, and artificial intelligence (AI) have a great influence on many domains that traditionally relied on the human workforce. By replacing human-based roles with robotics and AI-based tools designed solely for the task, able to work with less error and fatigue and with enhanced communication speed and cooperation, the throughput of businesses can be increased along with the quality of the goods produced. With higher-quality automation and decreasing costs for robots, the potential for autonomous systems has soared. As these trends continue, more systems will see much more mass automation [1]. With these developments and the need to address wider spatial contexts, multi-robot systems (MRSs) have seen a significant increase in their use over the last decade. An example of the benefits of MRS was shown in [2], where strawberry harvesting logistics were improved by 20–30% with the use of autonomous robots to deliver fruit from pickers to storage. MRSs have already seen promising development in fully autonomous coordination such as in warehouses [3], agriculture [4], search and rescue [5], and drone shows [6].

A large factor impacting the effective deployment of MRSs is multi-robot path planning (MRPP), being the ability to define paths each robot should take through the environment from its location to its respective goal, which both minimises the total distance each robot takes through the network and, more importantly, avoids all potential collisions with other robots. MRPP research overlaps with multi-agent path finding (MAPF) in multi-agent system (MAS) research, which is defined as the problem of finding paths for multiple agents such that every agent reaches its goal and the agents do not collide. The discussions in this survey are relevant to both MRPP and MAPF.

Different environmental representations have been used in MRPP research. Continuous space metric map representations have been used for local trajectory planning of multi-robot systems [7,8]. However, the computational complexity of considering a continuous space for MRPP is prohibitively high for large environments, even for a small MRS. To enable efficient planning in large environments, most of the existing MRPP algorithms have used discretised environmental representations. A grid representation of the environment, with each cell connected to four neighbouring cells, has been used for different MRPP algorithms [9]. More recently, a discrete topological perspective of the environment is the most-commonly used discrete environment representation in MRPP. In this topological perspective, the environment is modelled as a graph or network with nodes identifying key points and edges describing how these points connect to one another, and this utility effectively describes the navigable connections between spaces in which the robot can move. With the utilisation of these representations, the complexity of the space is minimised to only the aspects of key importance and allows MRPP algorithms to work more efficiently such as in [10], though the planning can be performed without such a map, instead opting for the use or metric representations, such as in [11], despite a trade-off in scalability.

Even with the topological representation of the environment, MRPP has been proven to be NP-hard [12], wherein the joint state space grows exponentially as the complexity increases, with the complexity increasing alongside either the total number of robots or the size of the map. This means that finding optimal routes through a topology requires more processing resources or time, as there are more options and possibilities that must be explored. This restricts the scalability and, thus, deployment of large MRSs in potentially congested spaces or domains.

The purpose and contributions of this survey are focused on three aims. First, and primarily, is to consolidate the variety of prioritised multi-robot-path-planning approaches and the rationale behind them to form a single, referable source from which researchers can begin their study in this area. Second is to identify areas within the space of prioritised planning that warrant further exploration. Many approaches use simulations for their development; thus, there is potential for further development when considered in practice for physical robots. Third is to identify and provide solutions for common weaknesses in the evaluation methodologies, which otherwise limit the transferability of efficacy into actual deployments.

There has been extensive research on multi-robot systems and the efficient coordination among robots within a system, with some overlap in the focus areas of research between multi-robot systems and multi-agent systems, where multi-agent systems can consider agents as physical (such as robots or humans), software (as in simulated or projected robots), or general constructs (of mobile or stationary entities); these may not contain any robots at all. Multi-robot systems exclusively consider robots as the agents. In the context of this survey, consider the terms robot and agent to be synonymous. Similarly, as many of the MRPP approaches discussed in this survey rely on explicit communications between the robots, multi-robot networks and networks are used as synonymous with MRSs.

Similarly, domains that have vastly different contexts share the same or similar research challenges, and the algorithms developed for a specific context in one domain may also be applicable in another domain. For instance, agricultural transportation logistics [10] and pipe routing [13] are both capable of utilising versions of prioritised planning implementations to optimise path assignments to individual robots. Thus, this survey will be useful for research in domains where prioritised planning has a role to play.

The Structure of this paper: In the remainder of this paper, the wider MRPP landscape is presented initially to provide a holistic view of the challenges and the wide variety of existing MRPP algorithms. These are presented in Sections 2 and 3:

- In Section 2, the common challenges to MRPP are presented, looking in depth into the specifics of why the four main issues of completeness, deadlocks, scalability, and communication are challenging, along with what can be done to help manage them.

- In Section 3, a categorical classification of existing MRPP approaches is given, including a brief description of how each category fits within the two major classifications based on inter-agent communication and the decision-making topology.

This is followed by a focused review of prioritised MRPP algorithms and discussions on the use of heuristics and rescheduling in prioritised MRPP in Sections 4 and 5:

- In Section 4, a detailed analysis is given of prioritised planning approaches, where the two fundamental components of prioritised planning, heuristics and rescheduling, are detailed alongside a breakdown of motion planning algorithms, which are the cornerstones of routing.
- In Section 5, we discuss our findings of the commonly identified limitations with research across the literature in prioritised MRPP algorithms, namely the ideas of the static scoring of edges, the batch assignment of tasks, the drive for context generality, the lack of heuristic-focused works, and the lack of supporting topology manipulation.

We then summarise these findings in the conclusions in Section 6.

2. Core Challenges for Multi-Robot-Path-Planning Algorithms

While developing an MRPP algorithm, many design challenges need to be addressed. Often, these challenges are mutually exclusive, with one or more other constraints failing when the MRPP algorithm is designed to conform to one constraint. A short description of the five main challenges to MRPP is given below in this section.

2.1. Completeness

In MRPP solutions, the main aim is to find a set of routes that the robots within an MRS may use to reach their respective targets, subject to some constraints. Whilst it is often a guarantee that such a set of assignments may exist, it is not always possible to find it [12]. Completeness, the most-important challenge faced by MRPP approaches, is used to define whether an approach can guarantee that, if a solution exists that allows all agents to reach their targets, it will be identified.

2.2. Optimality

For any collection of robots, their start locations, and the targets, there is always an optimal collective route, that is the collection of routes that each robot can take in order to minimise a collective objective measured by a metric such as the total distance travelled by all robots through the network [11]. The metric chosen is most commonly either the make-span (time the last robot arrives) or flow-time (sum of all route lengths). The key idea is that there is some optimal assignment of routes that can be matched, but cannot be beaten.

Optimality refers to the ability of the MRPP algorithm to guarantee the identification of an optimal solution given enough computing time, i.e., the ability to find the optimal route from the start to the target given such a route exists. This is an issue on which much of the research on MRPP is focused, as the computational complexity of finding the optimal route is much higher.

2.3. Deadlocks

Deadlock situations are a significant concern for MRPP solutions and deployments. Deadlocks refer to a situation in which a robot is unable to find a route through the environment because it is trapped by another robot. Often, this is caused by planning inefficiencies such as the incorrect assessment of priorities or sub-optimal waiting points. In [14], the authors designed an abstract concept detailing the causes of deadlocks, which they used to evaluate the efficacy of a range of prioritised planning approaches. While their aim was to detail specifically prioritised planning issues, the theory can be extended beyond this. They summarised the causes of deadlocks into two key situations. In the first, the cause of the deadlock is the result of the high-priority robot lying in the path required by a low-priority robot, and in the second situation, the cause is a high-priority robot following behind a low-priority robot.

Papers such as [15] have attempted to resolve these common types of deadlocked situations using methods that discourage navigation around the initial movement areas of other agents. These methods, while simple, are effective at managing many causes of deadlocks.

While not as prevalent of an issue, approaches must also ensure they do not fall into the risks of *livelock* (i.e., robots circle around one another and are unable to move forward) and *starvation* (where resources, such as nodes of special interest, are locked up under different processes) as these can also be critical problems for a deployed system.

2.4. Scalability

The largest challenge to the successful deployment of an MRS is its scalability, where the performance is not degraded with an increase in the size of the MRS or the working environment. The grand aim of a MRS is to have many robots coordinating and working fully autonomously for long periods of time. There are three primary components that impact the scalability: the size of the map, the number/length of tasks, and the quantity of robots. As these grow, the complexity of the joint state space increases with them at an exponential rate due to more data requiring processing and more situations that need evaluation. This type of growth leads to a slower processing of the system overall [12].

Effective approaches to combat this are the utilisation of distributed and decentralised deployments. These can help unburden a centralised coordination system that would otherwise be handling all the processing by itself [16]. This, however, comes with other scalability problems such as communication or synchronisation overheads.

Decentralisation is effective at spreading the workload; however, it is not enough alone; the processing requirement is only divided by the total number of decentralised processors. So, if the number of robots exceeds the capacity for a decentralised unit, the coordination must be decentralised further so that the number of robots per processor is stable.

Distributed processing is used widely with autonomous robots, where even if the primary coordination system is centralised, the elements of the motion planning can be run locally for each agent. For instance, an agent can be given a simple goal or path from the coordination system, allowing it to compute its own motion trajectories. This type of offloading allows for scalability issues to be lessened as a minimal amount of information needs to be computed centrally. With the full utilisation of distributed coordination, robots can make all decisions themselves utilising technologies such as swarm behaviour or potential fields [17].

2.5. Communication

Communication is another challenge within MRSs that must be overcome. In any MRPP approach, there must be a method for robots to share information, either via a wireless data connection or via physical observations of their environment. Without such channels, agents in centralised or decentralised systems are unable to plan/share their routes, and agents in distributed systems are unable to share their intentions about their movements.

This challenge can have critical impacts on not just the efficiency of a system, but also on safety. If communications slow down too much and processing takes too long, hazards can appear and become critical before the coordinator has time to react and recalculate the routes. This is especially troublesome in MRSs involving both robots and humans sharing the same workspace, where even though local navigation can work to avoid direct collisions, robots suddenly moving in unexpected manners can cause accidents indirectly.

When utilising wireless data communication, which is the most-common approach, it is important to understand the coupling between the robots and the environment. Many aspects of the environment can impact the reliability of such a system. Many materials can have significant impacts on the quality of wireless signals: stone walls for caving/mining robots or metal infrastructure in warehouses/factories can significantly impact the reliability of wireless technologies, and the radiation in nuclear decommissioning can prevent any signals from passing through. In outdoor environments such as for agricultural robots, the reliability of 4G is not always guaranteed, and even in a well-suited environment such as an office or school, issues such as blackouts and bandwidth limitations can also occur [18]. In real-world settings, there are many elements that can cause communication issues and are easily overlooked when solely working in simulation.

Communication overhead scales linearly with the number of agents in a system where the agents are only required to communicate with the coordinator; however, when the robots are all communicating their locations in a distributed manner without the use of a central coordinator, this can result in much communication processing for each robot and each coordinator, and if they must process much information with each communication, there is also potential for error accumulation [17]. Many coordination issues can arise if a piece of information, such as a robot's location, is sent, but becomes delayed or corrupted or is inaccurate, even if it is eventually received.

3. General Classifications of Multi-Robot-Path-Planning Algorithms

There are two classification systems for which all MRPP approaches can be described, inter-agent communication (IAT) and decision-making topology (DMT). IAC describes how much communication occurs between robots during routing, while the DMT describes the number of processing units active throughout the planning and control.

3.1. Inter-Agent Communication

Within the path planning in an MRPP implementation, agents can present different levels of consideration to other agents in their proximity, and this is shown through their communication with one another. The complexity of their interactions can be described on a scale, where on one end is a coupled approach where all agents are considered together, and the other is a decoupled approach where agents do not use this communication directly, between which is a combination where communication is rarely used.

3.1.1. Coupled

In a coupled approach, the routes of all agents are planned together within the same configuration space. Due to the interdependence with the configuration space, coupled approaches are at risk of scalability issues. As described above, the time it takes to find an optimal collision-free trajectory is dependent on the amount of complexity within the planning space. Where the search space increases exponentially with the number of agents, this can lead to large MRPP situations, struggling to find an optimal route.

While coupled approaches can take longer to process, the results are often much better than decoupled approaches as searching the entire joint state space can lead to identifying more-complex and -optimal solutions that may not be so easily discoverable using decoupled methods as described in Section 3.1.2.

3.1.2. Decoupled

Decoupled approaches work by ignoring the links and connections between agents, instead making use of conflict resolution. This often works with agents each initially planning their routes independently either online (computed by the robots) such as in [11] or offline (computed by an external machine) such as in [10], with these routes then being passed through some comprehensive conflict resolution phase.

Due to the limited exploration of the state space and no consideration of other agents within the route-generation stage, decoupled approaches are not always able to guarantee the optimal set of trajectories for each agent; however, they are able to guarantee a reasonable set of trajectories where possible, within a polynomial time. As this type of approach can more quickly find a valid route, it is better suited for scenarios with more robots and for use in human–robot shared spaces. In these scenarios, swiftly calculating new routes to avoid collisions in a dynamic environment is key to ensuring both safety and efficiency. As much of the processing for decoupled MRPP algorithms is not reliant on a joint state space, these approaches are also much more effective than coupled approaches when the domain makes use of a distributed or decentralised topology.

3.1.3. Dynamic Coupling

While most approaches are either coupled or decoupled, there exist some methods that utilise both. In these approaches, agents may perform their own decoupled planning and re-couple the fleet to resolve issues. Alternatively, they may perform re-coupling to some agents in their proximity and decoupling to the rest of the agents in the system. There are many approaches that utilise this dynamic coupling behaviour, but the general principle is that the state of coupling is not explicitly fixed.

Depending on the scope of the re-coupling, there are two broad sets in which re-coupling approaches may fall. Total re-coupling (TRC) requires all agents in the system to reconnect for conflict resolution, and partial re-coupling (PRC) only requires agents that are in the local proximity to the conflict to join in with resolution.

Total Re-Coupling:

These types of approaches are commonplace in distributed decision-making topologies (see Section 3.2.3), where algorithms are designed so each robot can limit communication with the fleet without adverse effects [19].

Partial Re-Coupling:

Algorithms like the ones produced by [20,21] take advantage of PRC to identify the optimal route for all agents in a distributed topology. In their approaches, when agents are within a local distance from one another, they plan in a joint state space; however, once apart, they make use of a decoupled approach. This style of approach is useful for environments with limited network coverage.

3.2. Decision Making Topology

The decision-making topology (DMT) defines the structure of the decision-making systems, that is the number of processing units (PUs) responsible for generating the trajectories of the robots and performing the conflict resolution. There are many types of decision-making topologies that can be broadly categorised based on their level of decentralisation. A fully centralised (centralised) topology has a single processing unit for many robots, and a fully decentralised system (distributed) deploys each robot with its own processing unit. Between these are what can be referred to as decentralised approaches, where there are many processing units, but less than one per robot [22]. Centralised, decentralised, and distributed DMTs are shown in Figure 1a–c. These categories can be broken down further based on the coupling that happens between agents.

Inherently, these structures can be considered synonymous with the network communication topology, especially for the centralised and some types of decentralised topologies. However, it is important to note the distinction between the network communication topology and the decision-making topology. A DMT is focused on how the agents and servers implement control over one another. e.g., as in Figure 1d; while it is possible for Agent 1 to communicate with Agent 3 via Agent 2, the limits of coordination for Agent 1 would be only with Agent 2. Overall, the DMT defines the limits of how agents make decisions together, rather than the availability of communication.

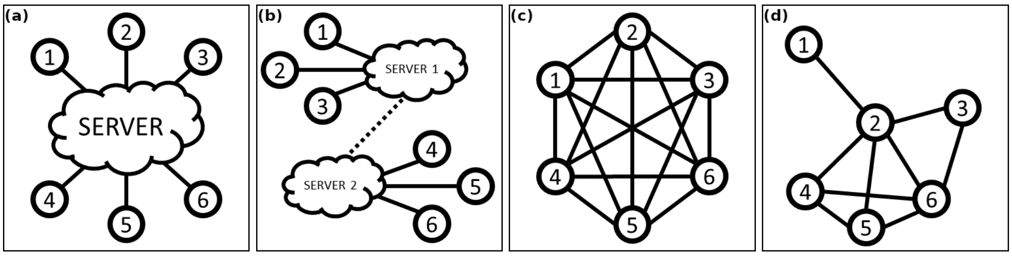

Figure 1. Communication structures: (**a**) Centralised: single central server; (**b**) Decentralised: localised servers communicating information; (**c**) Fully distributed: no servers; agents broadcast all information directly; (**d**) Unevenly distributed: each agent only communicate required information with a subset of other agents. The numbers 1–6 indicate the agents and the lines indicate communication links.

3.2.1. Centralised

The centralised topology consists of a communication network of a star, where a single processing unit (SPU) functions as the central coordinator. This SPU handles all decision-making processes related to task assignment, route planning, and conflict resolution.

Compared with decentralised and distributed topologies, centralised topologies have a significant reduction in communication overhead. Each agent communicates only with the SPU, meaning the amount of open connections is limited to the number of agents in the system. This allows for simple and straightforward approaches to be developed such as the prioritised planning method developed by [23].

The biggest risk of making use of a centralised approach is scalability. As the number of robots increases, having only one processing unit can lead to an overburdened system, which is slow to generate even the simplest of results. This can be especially risky when the SPU is responsible for many other critical functionalities such as navigation safety analysis.

3.2.2. Decentralised

Decentralised decision-making topologies are characterised by their use of multiple coordinators, each of which is responsible for a subset of the overall system. Two approaches that emphasise their unique benefits clearly are sub-fleet control and sub-region control.

In sub-fleet control, coordinators assume responsibility for a subset of robots. The coordinator in each group is responsible for calculating the routes and, then, communicating these with either a central coordinator or the network of other decentralised coordinators. Approaches such as platooning or virtual structures come under this definition. In leader–follower platooning, a single leader takes the lead and agents entering their vicinity relinquish their own control to follow the leader [17]; in virtual structures, agents will become a part of a temporary collective to maintain a geometric shape [17].

In sub-region control, the distribution of the coordinators is environment- or topology-based. In this structure, the coordinators take responsibility for managing defined regions within the network. This method can be seen in [24], where the decentralised coordinators each manage a defined junction, taking control of the autonomous vehicles that approach it and coordinating their movement through the junction. Decentralised approaches help manage the computational load away from a central coordinator, but may still struggle to keep up with demand as the number of agents distributed in the system increases.

3.2.3. Distributed

Distributed decision-making topologies, like decentralised methods, make use of a collection of processing units. However, unlike decentralised methods, distributed methods of MRPP do not risk scalability issues, whereas each robot works as its own processing unit, and the number of processing units scales proportionally with the number of robots.

In distributed topologies, algorithms are designed so that each robot is able to perform its own planning and organise its conflict resolution locally. Distributed approaches are often also decoupled approaches; however, this is not always the case: a coupled approach may make use of a distributed decision-making topology to offset elements of its processing that work independently. These types of distributed approaches can be grouped based on how much re-coupling occurs in the conflict resolution (see Section 3.1.3).

These types of topologies are distinct in their architecture from centralised and decentralised methods due to the absence of independent cloud-based/edge–server coordinators. However, not all distributed topologies are identical in structure. Depending on the application, environment, and robots, the topology may be either fully or partially connected. In a fully connected distribution, every robot will be in communication with every other robot, allowing for all robots to coordinate as a collective [25]; when conflict resolution is conducted for an individual robot, every agent is capable of contributing.

This is in contrast to a partially connected distribution, in which robots might only be allowed to connect to the resolution if they have a necessity to do so such as if they are in proximity to the agent needing replanning. In practice, these types of connected distributions are necessary due to communication constraints, where agents are unable to maintain a consistent or reliable connection to a single server and must instead act as relays for information [19,26].

4. Prioritised Multi-Robot Path Planning

Prioritised planning, developed by [23], is a commonly used decoupled approach to MRPP. It is a combination of two components, a prioritisation schema and a motion-planning system.

In traditional prioritised planning, each agent is assigned a priority, then their routes are planned in the priority order, with each subsequent agent planning around those with a priority higher than itself. These priorities are defined by a priority schema, which orders the agents based on some predefined metrics such as the distance to their respective targets (see Section 4.1). The routes themselves are calculated by the motion planning system (see Section 4.2), which makes use of a planning algorithm such as A*.

As the agents defined with the higher priorities plan first, they will often be planning on an empty map, i.e., so long as a valid route is possible, these higher-priority agents are guaranteed to reach their goal. In contrast, agents defined with a lower priority plan their routes in a more-restricted space navigating around the other robots, which are treated as dynamic objects. Due to the limitation on navigable routes, a route is not always guaranteed for the lower-priority robots. The complexity of the MRPP is, thus, reduced with prioritised planning with the decoupled and prioritised approach and can be used in large MRS deployments. However, it may also result in an incomplete MRPP in some conditions, as discussed above, and would require dynamically adjusting the underlying prioritisation schema or the motion planning system to obtain a complete solution.

More details on the prioritisation schema and the planning algorithms are discussed in the following subsections.

4.1. Prioritisation Schema

The prioritisation schema defines the approach to prioritise the agents, which is then used for ordering the agents in the decoupled path-planning phase.

4.1.1. Heuristics

The approach used to set the priority order of the robots is referred to as the heuristic. This subsection aims to highlight a broad range of heuristics found in the literature on prioritised planning. While some heuristics may work well generally, tailoring the chosen heuristic to the complexity and arrangement of the environment may lead to better results. These heuristics are summarised in Table 1 with brief descriptions and corresponding key publications.

Table 1. Overview of heuristics used in prioritised MRPP.

	Identifier	**Description**
[23]	Static Ordering 1	Prioritised based on order added to network
[27]	Hill-Climbing Search	Randomly swap priorities to search for better routes
	Random Ordering	Prioritised Randomly
[28]	Static Ordering 2	Agents given priorities based on their ID
[29]	Naive Surroundings	Counts number of other agents in local workspace
[30]	Road-Map Distance 1	Furthest robot is given priority
[31]	Road-Map Distance 2	Longest path first
[31]	Road-Map Distance 3	Ratio of context-aware plan to optimal path length
[32]	Coupled Ordering	Schema coupled with effective routing lengths
[33]	Continuous Enhancement	Blocked robots have scores increased
[34]	Planning Time	Total time to identify route in empty workspace
[11]	Path Prospects-R	Total number of distinct routes (with random tie-break)
	Path Prospects-LF	Total number of distinct routes (with Euclidean tie-break)
	Coupled Surroundings	Counts the number of objects in the local workspace that obstruct the robot
	Forwards Looking	Naive surroundings with restricted-search to low-route regions
[31]	Busiest Resource First	Agents using busiest resource receive higher scores

Static Ordering

The initial ordering for the priority schema in [23] used *the sequence in which agents were added to the network* as their heuristic. A similar heuristic was also used in [28], where the agents were assigned static scores based on their identity and always gave way to agents that had a higher priority. By itself, this ordering is not able to combat issues prevalent with larger and more-complex scenarios.

Road-Map Distance

The prioritisation heuristic used in [30] arranged the agents based on *their travel distance through a road-map to their respective targets*, with the furthest from their target having the highest priority. Conceptually, this was the first well-known use of knowledge-based heuristics in prioritised planning with a coupling to the environment (see Figure 2). This was similarly utilised in [31] with the heuristic of the longest path first being calculated as the ratio of the context-aware plan to the length of the shortest path.

A similar approach was utilised in [32], where the authors simultaneously performed task assignment and path planning. This results in the prioritisation schema being closely coupled to the resulting make-span and being more accurate to the reality of the routing than either the Euclidean distance or road-map distance.

Planning Time

Another approach that relied on network knowledge was designed by [34], in which agents were assigned a priority based on *the time taken to find a route* by them. Here, the intent was that agents that took a long time to find a route would benefit more from planning in an emptier map and should, thus, receive a higher priority. Just as with using the road-map distance, this worked well to support the prioritised planning and return priorities quickly, which is key for a scalable centralised approach. However, due to varying hardware on different robots in a distributed decision-making topology making use of heterogeneous robots, this can prove to be unreliable with the current implementation.

An alternative implementation could instead prioritise based on the total FLOPS during planning, which would be more consistent across evaluations of distributed processing, where each agent utilises unique hardware and sub-processes.

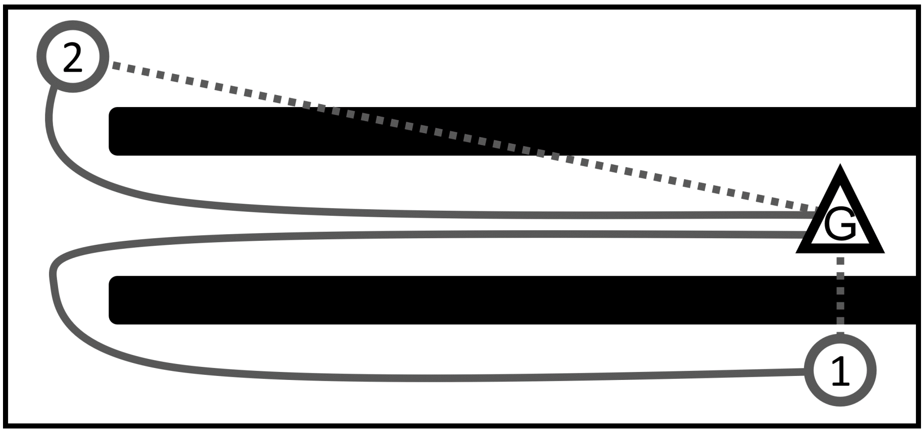

Figure 2. Impact of environmental constraints on optimal assignments; Robot 1 has a small Euclidean distance to the goal, but would take a longer route along a road-map than Robot 2.

Naive and Coupled Surroundings

In [29], the author developed a heuristic making use of knowledge about the local environment. Later titled naive surroundings by [11], this prioritisation heuristic scored agents based on *the number of other agents within its vicinity*. The intent behind this approach is that agents with a more-cluttered workspace would struggle more to find a route out of the workspace and may become trapped by other agents moving through it with higher priorities.

This is termed naive as it does not consider the effect of the agent on these objects. This is in contrast to the coupled surroundings heuristic developed by [11], which instead uses *the number of obstacles in the neighbourhood*. Here, an obstacle can be defined as a collection of objects that the robot cannot move between. If a robot is not being able to move between two objects, this approach classifies those objects as a single obstacle, and thereby coupling the robot to its environment.

Path Prospects

The authors of [11] worked to develop their own knowledge-based heuristic titled path prospects; in this, the agents are prioritised based on *the number of surrounding paths that are available to them*, calculated by counting the number of obstacles and the number of effective obstacles between them and their target by analysing the homology classes of the trajectories (the number of distinct routes that can be created to join two points). Obstacles are regarded as any obstruction, while effective obstacles are regarded as regions the robot is unable to navigate through due to the constraints of its footprint on the environment (see Figure 3). The researchers also evaluated the effectiveness of constraining the search algorithm to areas with more-promising minimal route options (forward-looking).

They evaluated a total of seven heuristics, comparing the two variants of path prospects (one that dealt with tie-breaks using random assignment and one that used the optimal route length), naive and coupled surroundings [29], the optimal route length adapted from [27], random assignment, and forward-looking. Their results showed using simulations comparing theirs to the benchmarks set out that they were able to achieve a higher global optimum with respect to completeness, which offered a good balance between make-span and flow-time.

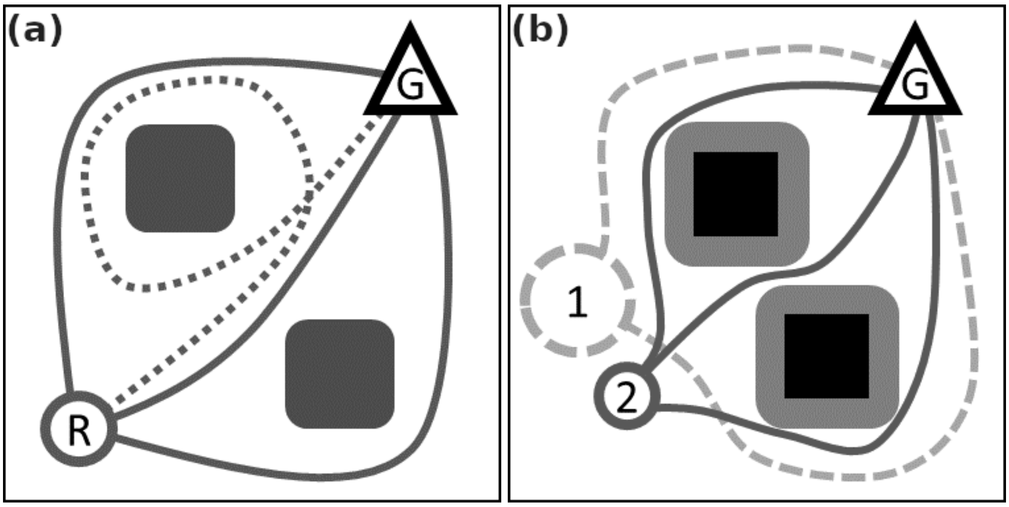

Figure 3. Generated homology class of trajectories [11]: (**a**) generation ignoring the coupling of a robot on the environment (obstacles); (**b**) generation considering environmental constraints of robot footprint (effective obstacles). In (**a**), R and G indicate the starting and target positions of the robot. In (**b**), 1 and 2 indicate two different starting positions of a robot and G indicate the target position.

Busiest Resource First

In [31], the authors utilised the direct information of the environment to improve the accuracy of the assignments, whereby they identified the resources within the topology that had the highest activity and raised the priority of agents utilising those facilities. This improved the coupling of the priorities to the environment and is one of very few heuristics that considers the structure of the topology as a motivation for improving assignments.

This approach went further by pairing the priorities of agents desiring the use of the resources in the same manner, e.g., two agents traversing an edge of the topology in the same direction. The motivation behind this is altruistic behaviours, i.e., if one agent is able to secure a section of a route, they secure it for all that need it.

4.1.2. Rescheduling

As discussed earlier, the original prioritised planning as defined by [23] is incomplete, i.e., the prioritisation of the robots can result in a situation where the robots with lower priority do not have a route to be found, thus failing the overall planning. Rescheduling is a modification to the priority schema, which exists to prevent such planning failures. However, it can also be extended to optimise the assignments to improve the ordering of agents. Inherently, decoupled approaches are not optimal as planning for each robot is performed independently. Some of the rescheduling approaches are able to guarantee a high global optimum; however, many others are unable to give a theoretical guarantee of optimality in polynomial time. Some of the commonly used rescheduling approaches are discussed below and summarised in Table 2.

Table 2. Summary of rescheduling approaches used to modify priority assignments.

	Identifier	Description
[23]	Prioritised Planning	Core algorithm
[35]	Full Search	Considers all possible configurations
[36]	Random Rescheduling	Random switching of priorities
[37]	All Opportunities	All agents occasionally given highest priority
[27]	Hill-Climbing Search	Randomly swap priorities to search for better routes
[38]	Deterministic Rescheduling	When an agent fails, it receives the max score
[39]	Local Priority Adjustment	Only conflict onlookers are required to replan
[40]	Priority Tuning	Robots with the least optimal routes have priorities increased and replan till convergence

Full Search

One of the first approaches to tackle optimality was completed by [35], where the author implemented *a full state space search by calculating routes with all combinations of priorities*. The result itself is able to guarantee optimality; however, not in a reasonable amount of time. As described by [35] at the time, the only way to guarantee optimality is to perform a complete search through every combination of priority orderings. However, as some approaches below show, a reasonably high global optimum can be reached while leaving some search spaces unexplored [27,40].

Random Rescheduling

The search for scalable optimality with the use of rescheduling first began in 2001 when [36] employed an approach making use of random rescheduling (RR). Here, *the priority scheme was randomised when a deadlock was detected*, and the plan was recomputed. Whilst simple, this was developed before the development of knowledge-based heuristics. This design was successful at avoiding many deadlocks; however, due to the random nature of the approach, it was not guaranteed to do so quickly.

All Opportunities

Randomisation in priorities was also used in the cooperative motion planner WHCA* [37], where *the priorities of agents were shuffled randomly* so each agent had an opportunity to be in full control of the environment. This allows agents that are stationary, having already completed their routes, to continue to cooperate with agents they are blocking.

Hill-Climbing Search

RR was later improved on by [27] with the use of a hill-climbing search. Here, initial priorities are randomly allocated and the set of routes calculated, then *random pairs of agents are selected and their priorities swapped*. This repeats iteratively until a more-optimal set of routes is found, restarting from scratch occasionally to prevent being stuck in local minima. This approach had a much-more-reasonable search space to explore than [35] and was still scalable; however, it took time to recompute routes each time an amendment was made.

Continuous Enhancement

In [33], the author proposed a more-complex solution to rescheduling called continuous enhancement (CE). In this approach, all robots start with a very low priority value, and rescheduling will be triggered when a robot cannot find a route after *updating the priority by two possible static values (a conservative value or a large value)*. While still quite simple, this was proven to work in many situations, and as the author described, “the algorithm is very robust against dynamic changes and erroneous robot behaviour”.

CE was later improved by [38], where maximum priorities were assigned when an agent failed to find any route. This approach, named deterministic rescheduling, is much simpler than its predecessor, *decreasing the total number of plans in attempting to find a route*. This approach uses the intuition that, if the routing fails, this is due to the complexity of the local environment, and this may not be resolved by moving it up one or two places. This follows the same concept as the naive surroundings by [27].

Local Priority Adjustment

In the approach by [39], local priority adjustment (LPA) was used. When a deadlock occurred, all agents with routes crossing within a certain radius of the deadlock were identified, *recomputing routes with all possible partial configurations of their priorities*. This method, while seemingly intensive, was able to perform similarly to the alternatives, being effective at improving the quality of priority assignments.

Local Neighbourhood Sdjustment

Similar to LPA, in [41,42], the authors utilised local neighbourhood adjustment, to adjust agent priorities in two manners: first, by *shuffling between agents being blocked and agents blocking* and, second, by *shuffling the priorities of agents utilising the most-active sections* within the topology.

Priority Tuning

A similar approach to LPA was taken in 2019, where [40] made use of a method called priority tuning. In this approach, regardless of whether a deadlock has occurred, once planning is completed, the *agents with the least-optimal routes would have their scores swapped with agents having slightly better scores*, then the agents would all replan their routes with the new orderings. This would continue until there was no considerable change in the results. This approach showed a gradual improvement in scores as more time was offered to the algorithm, enhancing towards the global optimum assignment, meaning it can also be classified as an anytime algorithm once all deadlocks have been eliminated. The authors also with this paper included an asynchronous approach to prioritised planning, which was employed to accelerate the tuning process. Their results showed that the asynchronous prioritised planning method was able to achieve higher convergence scores in a reasonable time.

4.2. Motion-Planning Algorithms

Motion-planning algorithms describe the process in which a route through a topology, from a start node to a goal node, is calculated. Specifically, this refers to topological routing, rather than trajectory planning (of a mobile manipulator or of a mobile robot within a continuous space). Motion-planning algorithms are a fundamental part of route planning that nearly all MRPP implementations build upon. In [43], the authors put together an assessment of the practical benefits and drawbacks of utilising various approaches of motion-planning algorithms, assessing both their real-world applicability and contexts in which they surpass one another.

4.2.1. Static Algorithms

Static algorithms are the most-fundamental component of motion planning. They are named as such because, once the optimal route is identified, any changes to the environment (e.g., a previously traversable area becomes untraversable due to the presence of another agent) mean the search must be restarted from the beginning.

A well-known and widely used basis for many static algorithms is Dijkstra [44], which identifies the shortest path to visit by iterating through edges connected from the start node and, at each node, computing the distance from the start. This approach spreads out in all directions until the target node is reached, and the path is determined by stepping backwards from the target.

An improvement to this was developed named A* [45], in which an additional computation was included, where a heuristic estimation of the distance (usually the Euclidean distance) to the target was added to the distance from the start node. This addition allowed the algorithm to prioritise searching in the direction of the target node, leading to a narrower search space.

These approaches can both be implemented as a forward or backward approach, by swapping the target and start nodes.

4.2.2. Replanning Algorithms

Replanning algorithms are designed to manage and modify identified routes, to ensure they remain optimal as the environment around the agent changes. Unlike static algorithms, these work without the need to restart the search. The algorithms Dynamic A* (D*) [46] and D* Lite [47] are the most-well-known replanning algorithms.

D* Lite is a modified version of a backwards A* approach. It works by performing replanning on small portions of the network when changes are identified, before propagating forwards towards the agent. It is designed to improve efficiency by limiting the impact on states not affected by the changes in the topology and restricting attention to states that are important for repairing the current solution; essentially, it only repairs parts that have become invalid. While D* Lite works on the entire route with global knowledge, LRA* [37] is designed to recalculate only the remainder of its route when a collision is pre-empted and is specifically useful for mobile platforms.

4.2.3. Anytime Algorithms

Anytime approaches aim to identify a valid solution first, then perform continuous improvement to attain a better optimality. The anytime repairing A* (ARA*) [48] is an effective anytime algorithm that identifies a sub-optimal route quickly using a loose bound defined based on the search time, then repeating elements of the search, tightening the bound, and optimising the route.

ARA* is designed to reuse sections of the results of its previous searches to decrease the computational requirement of the algorithm. Despite having a fast optimal route generation, implementations such as ARA* cannot respond to dynamic environments. Even if an optimal route has been found, if the topology of the environment is updated, the algorithm must start from scratch.

Anytime approaches work well for single-robot path planning, when there is no coupling; however, for conditions where many robots are restricted and have to plan around one another, this type of algorithm does not work well.

4.2.4. Replanning Anytime Algorithms

The replanning algorithm anytime dynamic A* (AD*) [49] was developed to use the section reuse component within ARA* and the invalidated-section-repairing component within D* Lite. The fusing of these components enables the algorithm to boast a strong efficiency in planning around dynamic obstacles while avoiding them.

4.2.5. Space–Time Search Algorithms

Space–time algorithms estimate the possible future conflicts in space and time and utilise both movements and waiting as viable actions to resolve those conflicts. CA* [37] is a prime example of this, in which the navigable space is a three-dimensional reservation table, in which the first two dimensions represent a 2D map and the third dimension is mono-directional, representing time. Here, waiting is considered a valid movement strategy for any time step, and reservation tables are used to mark off impassable regions. This is further extended in HCA* [37], which improves efficiency using search trees, and in CRH* [10], where the concept of waiting is combined with replanning strategies beginning at multiple time steps.

5. Discussion

In the literature analysed, there are a few key areas that affect the efficiency of the MRPP algorithms and need to be addressed. While the research as a whole is quite broad, in scope and evaluation, most work makes two key assumptions that define their algorithmic approach. The first assumption (which we named batch assignment) is that the planning is carried out in batches. At the point of planning, every agent is considered to be idle, and this could lead to unnecessary delays or route replanning for agents that are still in motion. The second assumption (which we named static scoring) is that the priority for each point along a reserved route is homogeneous, meaning a priority defined at the start of the route determines every point along it.

These assumptions have led to much of the existing research lacking accurate transferability when taken out of a single-route-planning instance and into practice in a physical deployment. Additionally, we identified a broader weakness in heuristic research as a whole (which we named context generality), in which the published works rarely tie the developed heuristics into the contextual domain directly. We also identified a shortcoming in the area of heuristic research, where only a fraction of papers actually had the developed heuristic as the focus of the paper.

Finally, we found a shortcoming in the exploration of cross-approach optimisations, where there has been little exploration in evaluating the utility of combining other approaches from the areas of topological manipulation, which should have a clear improvement on scalability for larger environments (due to the discretisation of the environment) and work well alongside prioritised planning.

5.1. Batch Assignment

The key concept here is that route planning is performed in batches. In a classical prioritised MRPP scenario, a system will start with all robots idle. Each robot will be assigned a priority, and they will plan their motion accordingly. Each robot would execute its navigation route by navigating to the goal. There are four different scenarios depending on how many goals are assigned to each robot and how the routes are planned for each goal:

- Scenario 1: All robots have only one target, so they will stop at $goal_1$. This approach is very broadly used [11,23,27,30,33,34].
- Scenario 2: Robots wait at $goal_n$ till all other robots have completed $goal_n$ before planning routes and starting to move to $goal_{n+1}$.
- Scenario 3: Each robot completes its $goal_n$ and receives a new, smaller priority, so as not to disrupt existing routes when moving to $goal_{n+1}$.
- Scenario 4: All robots that complete $goal_n$ receive $goal_{n+1}$. All robots delete existing routes, and planning restarts with one robot planning to its $goal_{n+1}$ and others still replanning to their $goal_n$.

Due to the efficacy in evaluating the quality of single-instance plans, Scenario 1 is used often. However, this has led to approaches not fully exploring the viability for practical deployments. As discussed in [32], minimising the flow-time can be thought of as trying to get agents to an idle state as quickly as possible so that they can take on new tasks; however, this does not fully resolve the underlying issues.

Scenario 2 is an extrapolation of how subsequent uses of Scenario 1 would result. It does not fully utilise the available resources, but does remove the likelihood of livelocks in the rescheduling phase.

For Scenario 3, the only way to guarantee a non-disruptive route is to utilise a priority value below the robots that already have routes; for any knowledge-based heuristics, this weakens their intended meaning.

Scenario 4 is a disruptive form of Scenario 2, where all resources are used, but with significantly increased processing due to unnecessary replanning. Routes being blocked can result in livelocked agents.

5.2. Static Scoring

Another issue identified here with the heuristics occurs when planning or replanning is carried out only for some robots while others are in motion (similar to Scenario 3 in batch processing). Once planning is completed for a robot, using a knowledge-based heuristic for the planning schema, every edge in the route can be thought of as being reserved with the score given by this heuristic. Although a robot moves through its reserved route, the score that had been calculated to reserve the edges in the route remains the same as when the routing began. This leads to inconsistencies with the results of replanning, where the robots that replanned their routes may have new scores computed, but these new accurate scores are being compared against historical existing scores to decide the winner to reserve the routes. As the purpose of knowledge-based heuristics is to bring relevant knowledge of the system into the planning schema, this is only partially achieved when the information being applied is historical and inaccurate for the new state of the robot.

It is only when priorities are integrated into live motion planning such as in [28,29,34], utilise reassignment as in [32,33], or apply to individual edges as in [10] that the scoring becomes reliable at all instances. For most approaches, however, this is not the case.

5.3. Context Generality

Prioritisation schema heuristics can be categorised into four levels based on their relation to context dependence and use. *Abstract heuristics* take no information from the state of the system, relying only on meta information such as what would be found in a configuration file. *System-based heuristics* can be categorised as taking general information from the system, applicable in nearly any context. *Context-based heuristics* can be described as taking information directly relevant to the deployed domain. *Combined heuristics* are simply the combination of multiple independent heuristics to improve assignments.

Due to the wide applicability of system-based heuristics, most research neglects potentially useful context information in favour of more-widely applicable approaches. It is understandable that applications using these types of prioritisation schema are less likely to garner wider attention due to their limited use cases. However, it is still worth noting their absence and potential. For example, in a warehouse environment, the payload weight/size could be a contributing factor to a viable heuristic due to the instability of the load if encountering many starts and stops during transit; in a horticultural fruit logistics setting, the time since the harvest could be a deciding factor as the shelf life of the fruit will be reduced if left in the heat after being harvested from the plant; in the domain of autonomous vehicles, an increase in the number of pedestrians in an area near school zones may work as a good prioritisation consideration. Table 3 shows how some of the heuristics discussed in Section 4.1.1 may be categorised based on the descriptions above.

Table 3. Examples and concepts of heuristics based on context generality.

Generality	Name/Description	
Abstract	Random Ordering	[27]
	Static Ordering	[23]
System-Dependent	Road-Map Distance	[30]
	Planning Time	[34]
	Path Prospects	[11]
Combined	Task-Priority Grading + Navigation Distance	[10]

5.4. Heuristic Focus

Throughout this review, it is evident that there is relatively little active research in prioritised MRPP focusing on improving the heuristics. The majority of the MRS research focused on optimising rescheduling for distributed fleet management. Of the heuristics discussed within this survey alone, only around half of the novel heuristics were developed as the sole focus of the paper.

5.5. Topology Manipulation

In order to reduce the MRPP problem's complexity, topology manipulation works by modifying the structure of the environment considered for planning. This can be performed in one of two ways.

Topology generation (generating internal structures within the topology) has been used in approaches like probabilistic road-maps [50], rapidly expanding random trees [51], and sub-dimensional expansion [20] to great effect, but have not been explored alongside prioritised planning.

Topology decomposition (decreasing the state space by combining components together) has been used in approaches such as road-map decomposition [52], component-based map decomposition [53], and junction-based map decomposition [24] to improve on scalability, with [53] being one of the very few works to utilise prioritised planning alongside topology manipulation to great effect.

6. Conclusions

As evidenced through this literature review, despite its general applicability and rapid conflict-resolution speed, prioritised MRPP is not under much active research. However, as shown in [10] and here, there is still potential for improving the generalised efficiency of prioritised MRPP approaches by focused research on heuristic developments and rescheduling approaches and framework developments to improve generalised efficiency.

Prioritised planning has been well explored on the surface in both the topological domain, in the form of online and offline path planning, and in the continuous domain, in the form of online motion planning. Due to the simple frameworks supporting heuristics and rescheduling, these are able to be applied quite broadly across many domains, which has undoubtedly helped in facilitating their interoperability. While they have been well explored over varying decision-making topologies, systems utilising dynamic coupling inter-agent communication have not received the same attention.

There is also much potential for developments in enhancing the global optimality in heterogeneous systems and new domains with combined heuristic research. Given the relatively small selection of available heuristics, the exploration of context-based heuristics may spark more creative solutions to system-based heuristics so the usefulness of context-based heuristics must not be understated. Further, rescheduling approaches have been generally limited to works utilising coupled and dynamic coupling inter-agent communications; however, they have not widely considered the impacts on communication challenges in this process, despite being a critical restriction on their utility.

Approaches have shown much potential when utilised alongside other innovations in the problem specifications such as the topological representations. However, further innovations such as topology manipulations are under-explored, although their combined utility seems to be an area for scalability optimisation.

Prioritised planning is a powerful architecture, with concepts applicable in both online and offline planning, and able to be utilised in both the topological and continuous domains. Yet, despite its conceptual simplicity, its potential is under-explored in a number of directions.

Author Contributions: Conceptualisation, J.R.H. and G.P.D.; validation, J.R.H.; investigation, J.R.H. and G.P.D.; writing—original draft preparation, J.R.H.; writing—review and editing, J.R.H. and G.P.D.; visualisation, J.R.H. and G.P.D.; supervision, G.P.D.; project administration, G.P.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Engineering and Physical Sciences Research Council [EP/S023917/1].

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data were created nor analysed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rizk, Y.; Awad, M.; Tunstel, E.W. Cooperative Heterogeneous Multi-Robot Systems: A Survey. *ACM Comput. Surv.* **2019**, *52*, 2. [CrossRef]
2. Das, G.P.; Cielniak, G.; From, J.; Hanheide, M. Discrete Event Simulations for Scalability Analysis of Robotic In-Field Logistics in Agriculture—A Case Study. In Proceedings of the ICRA 2018 Workshop on Robotic Vision and Action in Agriculture, Brisbane, Australia, 21–25 May 2018.
3. Fernandez Carmona, M.; Parekh, T.; Hanheide, M. Making the Case for Human-Aware Navigation in Warehouses. In *Towards Autonomous Robotic Systems*; Althoefer, K., Konstantinova, J., Zhang, K., Eds.; Springer: Cham, Switzerland, 2019; pp. 449–453.
4. Bechar, A.; Vigneault, C. Agricultural robots for field operations: Concepts and components. *Biosyst. Eng.* **2016**, *149*, 94–111. [CrossRef]
5. Drew, D.S. Multi-agent systems for search and rescue applications. *Curr. Robot. Rep.* **2021**, *2*, 189–200. [CrossRef]
6. O'Malley, J. There's no business like drone business [drone light shows]. *Eng. Technol.* **2021**, *16*, 72–79. [CrossRef]
7. Wen, T.; Wang, X.; Sun, Z.; Zheng, Z.; Kong, X. An Adaptive Local Path Planning Algorithm for Multi-robot Systems. In Proceedings of the 2022 IEEE International Conference on Unmanned Systems (ICUS), Guangzhou, China, 28–30 October 2022, pp. 817–822. [CrossRef]
8. Andreychuk, A.; Yakovlev, K.; Atzmon, D.; Stern, R. Multi-Agent Pathfinding with Continuous Time. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, Macao, China, 10–16 August 2019; pp. 39–45. [CrossRef]
9. Walker, T.T.; Sturtevant, N.R.; Felner, A. Extended Increasing Cost Tree Search for Non-Unit Cost Domains. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 13–19 July 2018; pp. 534–540. [CrossRef]
10. Heselden, J.R.; Das, G.P. CRH*: A Deadlock Free Framework for Scalable Prioritised Path Planning in Multi-robot Systems. In Proceedings of the Annual Conference Towards Autonomous Robotic Systems. Lincoln, UK, 8–10 September 2021; Springer: Cham, Switzerland, 2021; pp. 66–75.
11. Wu, W.; Bhattacharya, S.; Prorok, A. Multi-Robot Path Deconfliction through Prioritization by Path Prospects. *arXiv* **2019**, arXiv:1908.02361.
12. Yu, J.; LaValle, S. Structure and intractability of optimal multi-robot path planning on graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; Volume 27.
13. Belov, G.; Du, W.; De La Banda, M.G.; Harabor, D.; Koenig, S.; Wei, X. From multi-agent pathfinding to 3D pipe routing. In Proceedings of the Thirteenth Annual Symposium on Combinatorial Search, Online, 26–28 May 2020.
14. Dewangan, R.K.; Shukla, A.; Godfrey, W.W. Survey on prioritized multi robot path planning. In Proceedings of the 2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), IEEE, Chennai, India, 2–4 August 2017; pp. 423–428.
15. Cap, M.; Novak, P.; Kleiner, A.; Selecky, M. Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 835–849. [CrossRef]
16. Mao, W.; Liu, Z.; Liu, H.; Yang, F.; Wang, M. Research progress on synergistic technologies of agricultural multi-robots. *Appl. Sci.* **2021**, *11*, 1448. [CrossRef]
17. Soni, A.; Hu, H. Formation control for a fleet of autonomous ground vehicles: A survey. *Robotics* **2018**, *7*, 67. [CrossRef]
18. Prorok, A.; Malencia, M.; Carlone, L.; Sukhatme, G.S.; Sadler, B.M.; Kumar, V. Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems. *arXiv* **2021**, arXiv:2109.12343.
19. De Hoog, J.; Jimenez-Gonzalez, A.; Cameron, S.; de Dios, J.R.M.; Ollero, A. Using mobile relays in multi-robot exploration. In Proceedings of the Australasian Conference on Robotics and Automation (ACRA'11), Melbourne, Australia, 7–9 December 2011; pp. 7–9.
20. Wagner, G.; Choset, H. M*: A complete multirobot path planning algorithm with performance bounds. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 3260–3267.
21. Wagner, G.; Kang, M.; Choset, H. Probabilistic path planning for multiple robots with subdimensional expansion. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, IEEE, St Paul, MN, USA, 14–18 May 2012; pp. 2886–2892.
22. Blumenkamp, J.; Morad, S.; Gielis, J.; Li, Q.; Prorok, A. A framework for real-world multi-robot systems running decentralized GNN-based policies. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), IEEE, Philadelphia, PA, USA, 23–27 May 2022; pp. 8772–8778.
23. Erdmann, M.; Lozano-Pérez, T. On multiple moving objects. *Algorithmica* **1987**, *2*, 477–521. [CrossRef]
24. Zhao, W.; Liu, R.; Ngoduy, D. A bilevel programming model for autonomous intersection control and trajectory planning. In Proceedings of the Transportation Research Board 97th Annual Meeting, Washington, DC, USA, 7–11 January2018; pp. 1–25.

25. Patwardhan, A.; Murai, R.; Davison, A.J. Distributing collaborative multi-robot planning with gaussian belief propagation. *IEEE Robot. Autom. Lett.* **2022**, *8*, 552–559. [CrossRef]
26. Bayer, J.; Faigl, J. Decentralized topological mapping for multi-robot autonomous exploration under low-bandwidth communication. In Proceedings of the 2021 European Conference on Mobile Robots (ECMR), IEEE, Bonn, Germany, 31 August–3 September 2021; pp. 1–7.
27. Bennewitz, M.; Burgard, W.; Thrun, S. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Rob. Auton. Syst.* **2002**, *41*, 89–99. [CrossRef]
28. Clark, C.M.; Rock, S. Randomized motion planning for groups of nonholonomic robots. In Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS, Pasadena, CA, USA, 18–21 October 2001.
29. Clark, C.M.; Bretl, T.; Rock, S. Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems. In Proceedings of the IEEE Aerospace Conference, IEEE, Big Sky, MT, USA, 9–16 March 2002; Volume 7, p. 7.
30. Van Den Berg, J.P.; Overmars, M.H. Prioritized motion planning for multiple robots. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 2217–2222.
31. ter Mors, A. Evaluating heuristics for prioritizing context-aware route planning agents. In Proceedings of the 2011 International Conference on Networking, Sensing and Control, IEEE, Delft, The Netherlands, 11–13 April 2011; pp. 127–132.
32. Chen, Z.; Alonso-Mora, J.; Bai, X.; Harabor, D.D.; Stuckey, P.J. Integrated task assignment and path planning for capacitated multi-agent pickup and delivery. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5816–5823. [CrossRef]
33. Regele, R.; Levi, P. Cooperative multi-robot path planning by heuristic priority adjustment. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Beijing, China, 9–15 October 2006, pp. 5954–5959.
34. Velagapudi, P.; Sycara, K.; Scerri, P. Decentralized prioritized planning in large multirobot teams. In Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, Taipei, China, 18–22 October 2010; pp. 4603–4609.
35. Azarm, K.; Schmidt, G. Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 25–25 April 1997; Volume 4, pp. 3526–3533. [CrossRef]
36. Bennewitz, M.; Burgard, W.; Thrun, S. Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems. In Proceedings of the IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, Republic of Korea, 21–26 May 2001; Volume 1, pp. 271–276.
37. Silver, D. Cooperative pathfinding. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Marina Del Rey, CA, USA, 1–3 June 2005; Volume 1, pp. 117–122.
38. Andreychuk, A.; Yakovlev, K. Two techniques that enhance the performance of multi-robot prioritized path planning. In Proceedings of the AAMAS '18: The 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 10–15 July 2018; Volume 3, pp. 2177–2179.
39. Ma, H.; Harabor, D.; Stuckey, P.J.; Li, J.; Koenig, S. Searching with Consistent Prioritization for Multi-Agent Path Finding. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, Honolulu, HI, USA, 29–31 January 2019; Volume 33, pp. 7643–7650. [CrossRef]
40. Hu, B.; Cao, Z. Minimizing task completion time of prioritized motion planning in multi-robot systems. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2019), Bari, Italy, 6–9 October 2019; pp. 1018–1023.
41. Li, J.; Chen, Z.; Harabor, D.; Stuckey, P.J.; Koenig, S. Anytime multi-agent path finding via large neighborhood search. In Proceedings of the International Joint Conference on Artificial Intelligence 2021, Association for the Advancement of Artificial Intelligence (AAAI), Montreal, QC, Canada, 19–27 August 2021; pp. 4127–4135.
42. Li, J.; Chen, Z.; Harabor, D.; Stuckey, P.J.; Koenig, S. MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 20–27 February 2022; Volume 36, pp. 10256–10265.
43. Ferguson, D.; Likhachev, M.; Stentz, A. A guide to heuristic-based path planning. In Proceedings of the ICAPS '05 Workshop on Planning under Uncertainty for Autonomous Systems, Monterey, CA, USA, 5–10 June 2005; pp. 1–10.
44. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
45. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
46. Stentz, A. The Focussed D* Algorithm for Real-Time Replanning. In Proceedings of the IJCAI'95: The 14th International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 20–25 August 1995.
47. Koenig, S.; Likhachev, M. Improved Fast Replanning for Navigation in Unknown Terrain. *IEEE Trans. Robot.* **2002**, *21*, 354–363. [CrossRef]
48. Likhachev, M.; Gordon, G.; Thrun, S. ARA *: Anytime A * with Provable Bounds on Sub-Optimality. In Proceedings of the Advances in Neural Information Processing Systems 16, V ancouver and Whistler, BC, Canada, 8–13 December 2003; pp. 767–774.
49. Likhachev, M.; Ferguson, D.; Gordon, G.; Stentz, A.; Thrun, S. Anytime dynamic a*: An anytime, replanning algorithm. In Proceedings of the ICAPS 2005—The Fifteenth International Conference on Automated Planning and Scheduling, Monterey, CA, USA, 5–10 June 2005; pp. 262–271.

50. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
51. LaValle, S.M.; Kuffner, J.J. Randomized kinodynamic planning. *Int. J. Rob. Res.* **1999**, *20*, 378–400. [CrossRef]
52. Ryan, M. Multi-robot path-planning with subgraphs. In Proceedings of the 2006 Australasian Conference on Robotics and Automation, ACRA 2006, Auckland, New Zealand, 6–8 December 2006.
53. Ryan, M.R. Exploiting subgraph structure in multi-robot path planning. *J. Artif. Intell. Res.* **2008**, *31*, 497–542. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.