ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

INGEGNERIA BIOMEDICA, ELETTRICA E DEI SISTEMI

Ciclo 37

Settore Concorsuale: 09/G1 - AUTOMATICA

Settore Scientifico Disciplinare: ING-INF/04 - AUTOMATICA

HEURISTIC ALGORITHMS FOR PLANNING AND SCHEDULING DISASTER RESPONSE OPERATION WITH AN HETEROGENEOUS MULTI-ROBOT AUTONOMOUS SYSTEM

Presentata da: Matteo Lanzarini

Coordinatore Dottorato	**Supervisore**
Michele Monaci	Lorenzo Marconi

Esame finale anno 2026

Contents

Abstract

The demand for advanced systems capable of managing heterogeneous Multi-Robot Systems (MRS) is rapidly increasing in parallel with the growth of autonomous vehicles and robotics technologies. This research addresses this need by presenting a comprehensive system designed to coordinate and manage teams of heterogeneous unmanned land and aerial robots operating in highly dynamic and unpredictable environments, such as those encountered in Disaster Response (DR) scenarios. A central focus of our work lies in the development of a robust and adaptable task generation, allocation, and scheduling framework. We propose a greedy heuristic algorithm to tackle the dynamic task allocation problem, ensuring that the system can effectively assign tasks while considering the key constraints faced by robotic fleets. These constraints include limited resources, task priorities, and varying capabilities among heterogeneous agents.

Our approach is grounded in the decomposition of the overarching problem into smaller, more manageable sub-problems, thus optimizing each component individually, ensuring an efficient, scalable, and flexible solution. The proposed framework is particularly suited for complex scenarios where timely and accurate decisions are crucial. By optimizing the allocation of resources and tasks, our system enables heterogeneous teams of robots to operate collaboratively and autonomously, minimizing delays and maximizing mission success rates. Through this research, we aim to contribute to the development of resilient and intelligent MRS solutions, capable of addressing real-world challenges in disaster response and beyond.

Introduction

In 2019, natural disasters affected more than 90 million people globally [1]. Events such as storms, floods, wildfires, and volcanic eruptions continue to pose significant threats to human lives and livelihoods, disrupting communities and causing widespread devastation. Between 2000 and 2014, over 1.5 million individuals lost their lives as a direct result of these catastrophic events [2], [3], underscoring the devastating human toll of natural disasters. More recently, the earthquake that struck Turkey and Syria in February 2023 caused immense destruction, claiming the lives of more than 40,000 people and displacing millions. The increasing severity and frequency of these disasters, exacerbated by climate change, have been emphasized in the World Disaster Report, which projects a future with even greater threats to life, property, and infrastructure. This alarming trend underlines the critical importance of developing advanced Disaster Response (DR) technologies that can enhance the speed, efficiency, and effectiveness of emergency operations. Robotic systems hold significant promise as transformative tools for disaster response. These systems can serve as surrogates for human responders in hazardous and inaccessible environments, where the risks to human life are too great. Robots can enhance reconnaissance capabilities by providing real-time data through advanced sensors, enabling quicker assessment of disaster-stricken areas. Additionally, they can perform labor-intensive tasks, such as clearing debris, transporting equipment, and evacuating injured victims, which would otherwise require substantial human effort. For instance, as noted in [4], research competitions like RoboCup have showcased diverse robotic platforms designed for rescue challenges, including bipedal and crawling mechanisms, all equipped with a range of sensors and actuators. The variety of robotic capabilities demonstrated in these controlled environments highlights the potential for developing sophisticated multi-robot teams to address real-world disaster scenarios. Despite these advancements, the practical deployment of robotic systems in disaster response operations remains relatively limited. Since 2001, at least 40 documented instances of robot-assisted disaster interventions have occurred [5], [6], [7], [8], [9], [10]. However, only four of these cases have involved the use of Multi-Robot Systems (MRS), and just two featured autonomous robotic teams capable of independent decision-making and coordination. This limited adoption can be attributed to various challenges, including technological constraints, lack of robust task allocation strategies, and difficulties in ensuring seamless cooperation among heterogeneous robotic agents. As highlighted in [11], while significant progress has been made in MRS research over the last decade, a considerable gap remains between the capabilities currently available and those required for effective disaster response operations. Key challenges include improving the scalability and adaptability of multi-robot systems, ensuring their reliability in highly dynamic environments, and developing algorithms that enable autonomous teams to coordinate effectively under uncertain conditions. Addressing these challenges requires innovative approaches that integrate advancements in robotics, artificial intelligence, and human-robot collaboration. The urgency to bridge this gap is evident as disasters become increasingly complex and demanding. Unlocking the full potential of autonomous multi-robot systems has the potential to revolutionize disaster response efforts, enabling faster, safer, and more effective interventions in scenarios where human capabilities alone may fall short.

Literature

For a Multi-Robot System (MRS) to be effectively deployed in Disaster Response (DR) scenarios, several critical requirements must be met to ensure its operational utility. As noted in [11], [12], the solution proposed in this thesis takes into account the following key considerations:

- **Scalability:** Each disaster scenario is inherently unique, often varying significantly in scope, severity, and environmental complexity. Consequently, an effective MRS solution must be highly scalable, and capable of adapting to both small-scale interventions and large-scale operations involving diverse quantities and types of robots. This scalability is essential to meet the demands of different disaster scenarios, ranging from localized incidents requiring a few agents to complex operations necessitating extensive robotic teams.

- **Reliability:** Unpredictability is a defining characteristic of DR operations. This arises not only from incomplete or imprecise situational information but also from the potential for rapid environmental changes, such as aftershocks, collapsing structures, or secondary hazards following the primary disaster event. Additionally, the possibility of hardware malfunctions or damage to robotic agents further complicates operations. To address these challenges, the MRS must exhibit high reliability, enabling it to maintain functionality even in the face of unforeseen events and failures within the system. This includes the ability to compensate for individual agent malfunctions while maintaining overall mission progress.

- **Autonomy:** One of the primary advantages of robotic teams in DR scenarios lies in their capacity for autonomous operation. By reducing the need for direct human intervention in hazardous environments, autonomous MRS solutions enhance safety for human operators. Furthermore, a high degree of autonomy allows human responders to focus on critical tasks beyond robot supervision, such as strategic decision-making or medical interventions. As such, the autonomy of the MRS must extend to high-level decision-making, dynamic replanning, and self-coordination among agents.

- **Mobility:** A heterogeneous MRS often includes robots with varying mobility capabilities, such as aerial drones, ground vehicles, or even aquatic agents. Each of these platforms faces unique operational constraints and environmental challenges. For instance, an optimal path for a ground robot may be inaccessible to an aerial drone and vice versa. Therefore, a robust MRS solution must address these diverse mobility requirements, ensuring seamless collaboration between heterogeneous agents while accommodating their operational constraints.

Task allocation and scheduling represent foundational components of MRS planning activities, encompassing the determination of actions, their timing, and their assignment to specific agents. As highlighted in [13], task allocation is critical for leveraging the advantages of MRS over single-agent interventions. Effective task allocation ensures balanced workload distribution, minimizes operational redundancy, and enhances the overall efficiency of the mission. Unlike single-agent systems, autonomous multi-robot teams introduce additional complexities, such as the need for system-level autonomy achieved through real-time information sharing and collaborative decision-making. Early formal analyses by Gerkey and Mataric [14] provided a taxonomy of multi-robot task allocation (MRTA), while Dias et al. [15] surveyed market-based approaches that leverage distributed auctions for scalability and decentralization. Later works introduced coalition formation [16] and detailed taxonomies [17], showing the growing complexity of MRTA. Optimization-based strategies, such as mixed-integer linear programming (MILP), offer globally optimal allocations, though at the cost of exponential runtime. Recent contributions emphasize that MRTA and path planning are inherently coupled: obstacle avoidance and travel costs must be explicitly considered during allocation. The field of task scheduling in Multi-Robot Systems (MRS) has seen significant advancements. This field addresses the complexities of coordinating multiple robots to perform tasks efficiently. Recent literature provides insights into various approaches and methodologies developed to optimize task allocation and scheduling in MRS. Another comprehensive systematic literature review [18] delves into the recent advancements in Multi-Robot Task Allocation (MRTA) problems. The study categorizes existing approaches and highlights the challenges and future directions in MRTA research. In the realm of task and motion planning, a survey [19] examines recent contributions to the field, categorizing works based on their underlying methodologies. The survey provides a detailed analysis of the integration of task and motion planning in multiple mobile robots, emphasizing the importance of cohesive strategies in complex environments. Optimization techniques have been pivotal in enhancing MRTA problems. In [20] the authors discuss various optimization methods applied to MRTA, focusing on their applicability and effectiveness in real-world scenarios. The paper provides a critical analysis of different optimization strategies, offering insights into their strengths and limitations.

Resilience and robustness are critical requirements in real-world missions. Kalra et al. [21] introduced incremental re-planning methods that enable teams to recover from unexpected events. Calvo and Capitán [22] extend this with resilient MRTA formulations that handle recharging and failures in heterogeneous teams. Complementary approaches explore robust navigation under uncertainty [23] and metaheuristics such as NSGA-III [24] or particle swarm optimization [25] for scalable dynamic adaptation. David [26] presents SPADES, an integrated framework for simultaneous path planning and task allocation in dynamic environments. Online task allocation and scheduling in multi-manipulator systems have been addressed in recent research. In [27] specifically tackles the online allocation problem within multi-task scenarios, proposing solutions to enhance the efficiency and responsiveness of multi-manipulator systems. The study emphasizes the importance of real-time decision-making in dynamic environments. Furthermore, dynamic task allocation frameworks have been developed to enhance coordination in heterogeneous robot teams. A recent study [28] presents DTA-HMR-TT, a dynamic task allocation framework designed to improve task scheduling and coordination in multi-robot systems. The framework addresses the challenges associated with heterogeneous teams and dynamic task environments. In summary, the current state of task scheduling in MRS encompasses a diverse range of approaches, including optimization techniques, online allocation strategies, and dynamic frameworks. These developments aim to address the inherent complexities of coordinating multiple robots and enhancing efficiency, adaptability, and performance in various applications. The DR context imposes further constraints, including the need for low failure rates under some of the most challenging environmental conditions robots currently face. Quick response times and the capacity for dynamic replanning in the event of unexpected developments are also essential. Existing task allocation solutions for specific MRS cases have been explored in works such as [29], [30], [31], [32], [33], [34]. Still, the unique demands of DR scenarios require planning systems that go beyond standard frameworks. In particular, these scenarios may require presenting a preplanned schedule to a supervisor to verify the feasibility and safety of the mission. Unlike traditional online allocation approaches, disaster response situations demand a preliminary understanding of the intended actions to enable accurate predictions regarding outcomes and resource requirements. Consequently, it becomes essential to equip multi-robot systems with a planning tool capable of real-time operation during missions, particularly in response to unexpected events, malfunctions, newly identified obstacles, or emerging risks to operators and assets. In [35], a scheduling algorithm for

MRS supervision is presented, offering insights into long- and mid-term mission adjustments and predictions. Also in [36] they proposed a framework specifically for search and rescue operations, focusing on victim clustering and needs, that addresses the challenges of task planning and scheduling with heterogeneous teams of robots. While effective in their domain, the solution could benefit from enhancements in computational speed, generality, and versatility. Collectively, these works validate the importance of re-planning, fault tolerance, and adaptive mission repair, all of which directly we will try to assess in this work.

Is worth noting that Path planning Algorithms have been a cornerstone of robotics since the early formulation of the A* algorithm [37], which introduced heuristic search for shortest paths, and its dynamic extension D* [38], which efficiently supports re-planning in partially known environments. The comprehensive work of LaValle [39] further consolidated both graph-based and sampling-based planning methods into a unified theoretical framework. In multi-agent contexts, Conflict-Based Search (CBS) has been proposed as an exact method for multi-agent path finding (MAPF), guaranteeing optimality by systematically resolving conflicts between agents [40]. However, its exponential complexity highlights the scalability challenge of MAPF. Other studies, such as [41], [42], extended pathfinding to cooperative and any-angle formulations, while sampling-based variants like RRT and its extensions continue to offer probabilistic completeness in high-dimensional spaces [43]. These results establish the foundation on which our Tactical Planner builds: the iterative generation of alternative routes is consistent with the literature on k-shortest paths and dynamic replanning under obstacle constraints.

Contributions

Building on our prior work [46], we present a meta-heuristic algorithm integrated into a versatile tool for task planning and allocation in Multi-Robot Systems. This tool accounts for the operational area's morphology and is designed to address a wide range of scenarios with heterogeneous autonomous agents. It combines adaptability, computational efficiency, and reliability, making it suitable for real-time applications and enhancing the planning and execution capabilities of robotic teams in dynamic and unpredictable disaster response environments. The proposed tool generates real-time operational plans that can either be reviewed by human supervisors or autonomously deployed in the field. Its design ensures robust performance even under adverse conditions, such as sudden mission changes or unforeseen challenges. By leveraging advanced algorithms, it balances automation with human oversight while enabling efficient and reliable operation in high-risk scenarios. Additionally, the tool is resilient to disruptions, allowing for continuous monitoring and rapid updates to task allocation and planning strategies as conditions evolve or new information becomes available. This unified approach bridges the gap between theoretical research and practical application, offering a robust and responsive solution for DR and other complex operational contexts. To further expand the proposed tool's capabilities, we have developed and tested an initial version of an algorithm designed to automate task generation. This advancement enhances the system's overall autonomy and ensures greater adaptability in dynamic and unpredictable environments, such as those encountered during disaster response operations. The system can quickly and effectively respond to unexpected challenges or impediments by automating the task creation process, and dynamically adjusting its operational plan to accommodate new conditions. This capability allows for higher-level replanning, enabling the system to reassess and reorganize its priorities, resource allocation, and strategy without requiring significant human intervention. Such a feature is particularly valuable in scenarios where real-time decision-making is crucial, as it reduces delays and increases the system's responsiveness to emerging situations. Furthermore, the automated task generation algorithm contributes to the robustness and scalability of the Multi-Robot System by facilitating seamless adjustments across a wide range of operational scales and configurations. This marks an important step toward fully autonomous systems capable of managing complex missions with minimal external supervision.

Chapter 1

Multi-Robot Systems for Disaster Response

Disaster Response refers to the organized set of actions undertaken immediately following a disaster to minimize its adverse impacts on human life, property, and the environment. It is a critical phase within the broader disaster management cycle, which encompasses mitigation, preparedness, response, and recovery. The primary goals of disaster response are to ensure the safety of affected populations, provide essential services, and facilitate the transition to recovery and reconstruction. Disaster response can be broadly categorized into three main activities:

- Emergency Assistance: This involves search and rescue operations, first aid, and the provision of basic necessities such as food, water, shelter, and medical care. For example, during natural disasters like earthquakes or hurricanes, immediate efforts focus on locating survivors, treating the injured, and addressing urgent needs [47].
- Resource Deployment and Coordination: Efficient disaster response requires the mobilization and allocation of resources, including personnel, equipment, and supplies, to areas of need. Advanced systems, such as Geographic Information Systems (GIS), play a vital role in mapping disaster-affected regions and optimizing resource distribution.
- Communication and Information Management: Effective disaster response relies heavily on robust communication networks to share real-

time information among responders, decision-makers, and affected communities. This is crucial for maintaining situational awareness and coordinating actions in dynamic and high-pressure environments.

Disaster response operations are inherently complex, requiring rapid decision-making and efficient resource allocation in unpredictable environments. Recent advancements in technology have significantly enhanced the capabilities of disaster management agencies, enabling more effective preparedness, response, and recovery efforts. This chapter is dedicated to the study of autonomous multi-agent systems, providing a comprehensive analysis of their applications, use cases, benefits, limitations, and the essential characteristics required for their effective deployment.

1.1 Applications of MRS

Multi-robot systems (MRS) represent a significant and evolving field within robotics [48], focusing on the coordinated operation of multiple autonomous robots to achieve shared objectives. Formally, an MRS comprises two or more autonomous mobile robots working collaboratively, often referred to as teams or societies of mobile robots. In such systems, individual robots, which may have limited capabilities on their own, coordinate their actions to accomplish well-defined goals, leveraging the power of cooperation to enhance overall system performance. Research in MRS is inherently interdisciplinary, drawing upon concepts and techniques from robotics, control theory, computer science, and artificial intelligence. This interdisciplinary approach addresses complex challenges such as coordination, communication, and collective decision-making among multiple robots. The study of MRS is relatively recent and ongoing, with many interesting developments and opportunities anticipated in the future. The applications of MRS are diverse and impactful across various sectors:

- Disaster Response: In search and rescue operations, MRS can be deployed to navigate hazardous environments, locate survivors, and deliver essential supplies, thereby reducing risks to human responders.
- Environmental Monitoring: Teams of robots can efficiently monitor large areas for environmental data collection, pollution tracking, and wildlife observation, providing comprehensive and timely information.

- Agriculture: MRS can perform tasks such as planting, harvesting, and monitoring crop health, increasing efficiency and precision in agricultural practices.

- Security and Surveillance: Coordinated robot teams can be deployed to conduct surveillance over extensive areas, enhancing security measures in public spaces, borders, and critical infrastructures.

- Space Exploration: MRS are utilized in space missions to explore planetary surfaces, construct habitats, and perform maintenance tasks, leveraging their ability to operate in challenging extraterrestrial environments.

Despite the promising applications, deploying MRS in real-world scenarios presents challenges, including scalability, reliability, autonomy, and mobility [49]. Addressing these challenges requires ongoing research and development to enhance the robustness and adaptability of MRS in dynamic and unpredictable environments. In summary, Multi-Robot Systems represent a transformative approach in robotics, enabling collaborative efforts among autonomous robots to perform complex tasks across various domains. As research progresses, MRS are expected to play an increasingly vital role in advancing technology and addressing real-world challenges. Our primary focus in this work is the application of Multi-Robot Systems (MRS) for Disaster Response scenarios. However, the tools and approaches we propose could easily be adapted for use in other domains. Consequently, we will examine the application of these systems within the context of disaster response, while also highlighting their broader potential for deployment in diverse fields.

1.1.1 Disaster Response Scenario

Multi-robot systems have become increasingly integral to disaster response operations, offering enhanced efficiency and safety in complex and hazardous environments. These systems are deployed in various scenarios, including search and rescue missions, environmental monitoring, and infrastructure assessment. For instance, collaborative ground-aerial MRS have been utilized to assess damage, analyze environmental stability, and allocate resources effectively in disaster-stricken areas. In such operations, unmanned aerial vehicles (UAVs) provide rapid area mapping and real-time data transmission, while unmanned ground vehicles (UGVs) conduct detailed inspections and deliver aid to inaccessible locations [50]. Despite their advantages, deploying MRS in disaster response presents several challenges. One significant difficulty is ensuring reliable communication and coordination among heterogeneous robots operating in dynamic and unpredictable environments. The complexity of task allocation and scheduling increases with the diversity and number of robots involved, necessitating sophisticated algorithms to manage these processes effectively [36]. Another challenge is the limited autonomy and energy constraints of individual robots, which can impede prolonged operations in disaster zones. Robots must navigate complex terrains, avoid obstacles, and adapt to unforeseen changes in the environment, all while managing limited power resources. Enhancing the autonomy and energy efficiency of robots is crucial for improving the effectiveness of MRS in disaster response [51]. Security is also a critical concern, as MRS can be vulnerable to cyber-attacks and system faults. Ensuring the robustness and resilience of these systems against such threats is essential to maintain operational integrity during critical missions. In summary, while MRS offer significant potential in enhancing disaster response capabilities, addressing the challenges of communication, coordination, autonomy, energy efficiency, and security is vital for their successful deployment in real-world scenarios.

Robot for Disaster Response

Robots have become increasingly integral to disaster response operations, offering enhanced efficiency and safety in complex and hazardous environments. These systems are deployed in various scenarios, including search and rescue missions, environmental monitoring, and infrastructure assessment.

- Unmanned Aerial Vehicles: commonly referred to as drones, have proven to be indispensable in disaster response scenarios due to their ability to quickly survey large areas and access locations that are otherwise inaccessible. For instance, UAVs equipped with high-resolution cameras and thermal sensors were extensively used during the 2010 Haiti earthquake to map affected areas and identify survivors. Similarly, during the Fukushima nuclear disaster in 2011, UAVs were employed to measure radiation levels and assess structural damage while minimizing human exposure to hazardous conditions [52]. A study [53] demonstrated a collaborative approach using both aerial and ground robots to assess the damage and allocate resources effectively in disaster zones.

- Unmanned Ground Vehicles: particularly effective in navigating debris-filled or structurally unstable environments. For example, the PackBot by iRobot has been used in several disaster scenarios, including search-and-rescue operations after the 9/11 attacks and the Fukushima nuclear disaster. Its compact design, mobility, and ability to carry cameras and sensors make it suitable for inspecting collapsed buildings and detecting hazardous materials [54]. Another example is the RoboCue, a snake-like robot developed by Tohoku University, which was deployed in disaster response exercises. Its flexible body enables it to crawl through narrow spaces in collapsed structures to locate trapped individuals [4].

- Maritime Robots: In flood or tsunami scenarios, maritime robots such as the EMILY (Emergency Integrated Lifesaving Lanyard) robotic buoy have been used to assist in water rescues. EMILY can be remotely controlled to deliver life jackets or ropes to stranded individuals in fast-moving water, as demonstrated in rescue operations during Hurricane Harvey in 2017.

Despite these advancements, challenges remain in deploying robots for disaster response. Ensuring reliable communication and coordination among heterogeneous robots in dynamic environments is complex. Additionally, the limited autonomy and energy constraints of individual robots can impede prolonged operations in disaster zones. Addressing these challenges is crucial for improving the effectiveness of robotic systems in disaster response. Leveraging the unique capabilities of different robotic platforms can significantly enhance disaster response operations. However, further advancements in autonomous decision-making, energy efficiency, and inter-robot communication are essential to maximizing their effectiveness in real-world scenarios.

Chapter 2

Framework and Analysis

This chapter introduces the mathematical framework and problem formulation for the development of a planning and scheduling system tailored to complex and dynamic operational environments. Specifically, we focus on the generality of the approach, while keeping the attention on disaster response requirements, where task scheduling and resource allocation must be conducted under strict temporal, spatial, and operational constraints. The chapter aims to provide a structured representation of the problem, capturing the interplay between agents, tasks, resources, and the environment while addressing the unique challenges posed by the application domain. The first section outlines the mathematical formulation of the problem, detailing the representation of agents, their capabilities and resources, tasks, and the constraints governing their execution. This formalization ensures clarity in the problem definition and serves as the foundation for designing algorithms to address task scheduling and allocation. By incorporating temporal dependencies, resource limitations, and cooperative task requirements, the framework reflects the real-world complexities encountered in DR scenarios. The second section focuses on analyzing the specific requirements of the problem, with an emphasis on the unique characteristics of DR applications. These include the need for real-time adaptability to unforeseen events, such as equipment failures or emerging hazards, and the ability to prioritize tasks based on mission-critical objectives. Additionally, the analysis highlights the importance of balancing computational efficiency with solution robustness, enabling the system to operate effectively in time-sensitive and resource-constrained settings.

2.1 Framework

In this section, we will present a parametrization for an MRS in a generic mission in a Disaster Response scenario, that we will use for our algorithm proposal. Let denote the set of agents with A, which could be composed both by robots, such as UGV and UAV, and human operators. In our case study, we will assume every agent is a robot, without the loss of generality. Due to the heterogeneity of the team, each agent could have a different set of capabilities, and thus we denote with $Ca^j = \{Ca^j_1, \dots, Ca^j_c\}, j \in A$ the vector that stores the information of agent j, where c is the total number of distinct capabilities available in the team. This set represents capabilities as sensing equipment and every type of ability that a robot in the group has that has no associated limited resources, e.g., an infrared sensor or an actuated robotic appendage to remove debris. Ca^j_k is a binary variable, that represents the presence (1) or absence (0) of capabilities k on the agent j. Then we have to introduce variables for those resources that will diminish during operation. The set of resources is denoted with $Ra^j = \{Ra^j_1, \dots, Ra^j_r\}$ where r is the total number of distinct types of resources available to the team. Ra^j_k is an integer variable that count the amount of resource k available to agent j. In a DR scenario, a possible resource could be a set of first aid kits carried by a ground agent, that has to deliver them to people in need or fire extinguishing exploding balls; in general, could be every kind of equipment that has limited usage in quantities.

Let's denote with T the set of tasks in a mission. Mirroring the variables for the agents, each task could require a set of capabilities $Ct^i = \{Ct^i_1, \dots, Ct^i_c\}, i \in T$ and some resources $Rt^i = \{Rt^i_1, \dots, Rt^i_r\}$ in any possible combination, where Ct^i_k represents the need of capability k for task i if equal to 1, and Rt^i_k is the number of the resource k needed for that same task. A task i could be done only by those agents that have enough resources and the desired capabilities set specified by Rt^i and Ct^i. Regarding the management of task allocation and the assessment of their completion, we utilized a 2D map representation. This choice aligns with our focus on planning and travel time estimation rather than the complexities of three-dimensional navigation or dynamic environmental modelling. Each task is assigned a specific location in an xy-plane, reflecting its geographical position within the operational area. To account for the time required to perform each activity, an estimated duration D^i is also associated with each task. Additionally, tasks are assigned a priority value p_i, which serves as a key parameter for establishing a hierarchy of importance. This prioritization enables the system to favour the execution of high-priority tasks, often critical to mission success, over secondary or less urgent operations. By integrating this prioritization into the task allocation framework, the system ensures that time-sensitive or life-critical objectives are addressed promptly, while lower-priority tasks are deferred until the most pressing needs have been met. This approach is particularly advantageous in scenarios such as Disaster Response, where the ability to dynamically adapt to evolving conditions and resource availability is crucial. For instance, higher-priority tasks, such as rescuing individuals or extinguishing a spreading fire, must be addressed immediately, whereas tasks like debris removal or area reconnaissance may hold secondary importance. By incorporating estimated durations and priority values, the system provides a structured methodology for task sequencing, enhancing operational efficiency and responsiveness. Furthermore, the use of a 2D map not only simplifies computational complexity but also facilitates intuitive visualization for human supervisors, enabling them to monitor the planning process, validate task feasibility, and make informed decisions. This dual emphasis on computational efficiency and human interpretability ensures the practical applicability of the proposed framework in real-world scenarios. To address the diverse requirements and considerations necessary for planning an effective intervention in a Disaster Response scenario, we have incorporated four additional relative start and end constraints into our framework. These constraints enhance the planning model by accounting for interdependencies among tasks, ensuring operational coherence and mission feasibility under dynamic and complex conditions. The constraints are defined as follows:

- End-after: A task cannot conclude until another specified task has been completed. This constraint reflects scenarios where the completion of one task directly depends on the conclusion of another. For instance, a mobile charging station must remain operational and accessible until all dependent agents in the mission, such as UAVs or UGVs, have been successfully recharged. This ensures that essential resources remain available for ongoing operations.
- Start-while: A task cannot commence until another task is already

underway. This captures situations where tasks are inherently interdependent in their execution. For example, aerial reconnaissance by a UAV might need to begin concurrently with a UGV performing ground operations to ensure complementary data collection and situational awareness.

- Cooperative tasks: Certain tasks require simultaneous execution by multiple agents, reflecting the collaborative nature of some operations in Disaster Response. For example, aerial reconnaissance by a UAV must be conducted concurrently with ground debris removal by a UGV to provide real-time data and enhance the efficiency of ground operations. Such cooperation ensures task synchronization and optimizes resource usage.
- Start-after: A task is constrained by a fixed lower bound on its start time, meaning it cannot begin earlier than a predetermined moment. This constraint is particularly useful for scheduling tasks that depend on external factors, such as environmental conditions or the availability of specific resources. For instance, initiating a debris-clearing operation might need to be delayed until the completion of preliminary fire suppression in the area.

By integrating these constraints, the framework achieves a higher degree of realism and flexibility, aligning task scheduling with the inherent complexities and interdependencies of Disaster Response scenarios. These constraints also ensure that task execution respects the temporal and operational dependencies critical to mission success while providing a robust structure for dynamic replanning in the face of unforeseen challenges or changes in the operational environment.

2.2 Problem Analysis

As formulated, this problem could be represented as a highly complex variant of the dynamic programming approach to the heterogeneous vehicle routing problem (HVRP), with additional synchronization and precedence constraints challenges. The Vehicle Routing Problem (VRP) (Appendix B) is a well-known combinatorial optimization problem extensively studied in operations research, transportation logistics, and robotics. It is a generalization of the Traveling Salesman Problem (TSP), where multiple vehicles must serve a set of locations while optimizing specific objectives such as minimizing travel distance, cost, or time. The VRP is crucial in fleet management, logistics, and autonomous robotic systems. Due to its NP-hard nature, exact solutions are computationally expensive for large instances, necessitating the use of heuristics and metaheuristics for real-world applications. The problem has many variations, including capacitated VRP (CVRP), VRP with time windows (VRPTW), and heterogeneous VRP (HVRP), each addressing different constraints encountered in practical scenarios. The multi-robot task scheduling problem shares many similarities with VRP but introduces additional complexities such as heterogeneous agent capabilities, synchronization, precedence, and real-time path estimation. By leveraging techniques from VRP research, we aim to develop a scalable and adaptive scheduling algorithm for autonomous systems operating in dynamic environments. Our approach builds upon heuristic and metaheuristic methods, balancing computational efficiency and solution quality, ensuring mission success even in highly unpredictable disaster response scenarios. Their evolving nature makes it impossible to compute a fixed solution a priori, as the operational environment changes dynamically during mission execution. Additionally, the necessity to compute solutions online in real-time, while agents are operating in hazardous areas, adds significant computational and operational constraints. Rapid scheduling is essential to ensure that the team is not left without clear directives or critical task assignments, minimizing exposure to risk and inefficiencies. A major challenge in this context is dealing with uncertainty and the potential for unexpected events that can disrupt precomputed plans. These disruptions can arise from environmental changes, such as blocked pathways or the appearance of new hazards, or from agent-specific issues, such as malfunctions or resource depletion. Consequently, reliance on an offline, globally optimal approach is infeasible, as such methods are typically computationally intensive and prone to obsolescence the moment the environment deviates from initial conditions. Instead, we propose a meta-heuristic greedy approach that prioritizes fast computation and satisfactory solutions. This approach embraces a pragmatic trade-off between performance and optimality, aiming to provide timely and actionable plans that can be continuously adapted as the scenario evolves. A foundational formulation of the Combined Vehicle Routing Problem (CVRP) with synchronization and precedence constraints is provided in [55]. This framework addresses the assignment of multiple vehicles to serve various locations while adhering to constraints such as the simultaneous presence of vehicles at specific locations and defined time windows. In our case, while time windows are less relevant (apart from delayed starting times), we introduce new constraints such as capacity limitations, resource availability, and relative execution dependencies. For instance, certain tasks may require concurrent execution, such as monitoring a hazardous area with a UAV while a UGV performs ground-level operations. Conversely, some tasks cannot terminate until others are completed, such as a UAV continuing surveillance until a ground agent concludes a rescue operation. The heterogeneity of agents further complicates the problem, requiring consideration of distinct travel times, risks, and operational capabilities for each agent. For example, UAVs are faster and less constrained by obstacles than UGVs, but they may have limited operational endurance or payload capacity. Additionally, the dynamic nature of the mission environment means that maps, travel times, risks, and agent positions evolve continuously, necessitating individualized evaluation of travel times and risks for each agent from multiple starting points. To address these complexities, we incorporate differentiated weighting mechanisms during task assignments. For instance, the framework prioritizes the use of more resilient or robust agents, such as those with higher endurance or better protection, over more fragile ones. This approach enhances mission reliability and minimizes potential failures caused by agent vulnerabilities in high-risk areas. Furthermore, our framework accounts for task interdependencies, such as the need for simultaneous or sequential execution. Cooperative operations, such as reconnaissance by aerial agents combined with ground-level interventions, are modelled explicitly to ensure proper coordination. Precedence relationships and task prioritization are integrated into the scheduling mechanism, ensuring that high-priority tasks are addressed promptly while maintaining operational feasibility. Overall, this dynamic framework seeks to optimize the coordination of heterogeneous agents in disaster response scenarios by balancing computational efficiency, operational safety, and adaptability to evolving conditions.

Chapter 3

Scheduling and Allocation

Our proposed solution addresses a complex problem by introducing a framework capable of generating a feasible schedule that satisfies numerous operational constraints. The primary objective is to maximize task throughput while prioritising high-priority tasks, minimizing risk exposure, and ensuring mission completion in the shortest possible time. The framework is designed to be fast and efficient, capable of adapting to the unpredictability and rapid evolution of hazardous environments. This adaptability is critical in scenarios such as disaster response, where delays in decision-making could leave the team vulnerable and uninformed, potentially compromising both safety and operational success. Recent literature has emphasized that real-world disaster scenarios commonly produce tasks online with hard or soft deadlines; consequently, scheduling algorithms must operate under time-critical constraints rather than relying on offline, static optimization. In particular, online matching and bipartite allocation strategies have been proposed that process task arrivals in real time and assign them to available agents with bounded computational cost [56]. These methods achieve performance levels close to offline ILP formulations on average while offering orders-of-magnitude improvements in runtime, which is essential when decisions must be taken within seconds. From a methodological viewpoint, online matching approaches focus on maintaining feasibility and improving throughput under arrival uncertainty, often using greedy or competitive-ratio guarantees. This contrasts with offline MILP/ILP solutions that aim for global optimality but suffer from prohibitive computation times as the mission size grows. For our framework, this distinction justifies the choice of a greedy/metaheuristic pre-optimization stage: it provides timely, near-feasible assignments that can be re-evaluated as new tasks appear, while offline optimal solvers remain useful as benchmarks or for small problem instances. Empirically, works in flood response and UAV tasking contexts report that online matching attains task completion rates comparable to optimal offline methods while reducing latency significantly [56]. Unlike traditional scheduling problems, where the scenario remains static and predictable, our problem space is characterized by a highly dynamic environment that is subject to frequent and unexpected changes. For example, new hazards may arise, danger zones may dissipate, or new opportunities for task execution may be discovered. These unpredictable factors make it infeasible to evaluate all possible combinations of path travel times, task allocations, and agent assignments in advance. Consequently, the scheduling process requires a dynamic approach integrating real-time scenario updates into the decision-making pipeline. To address this, we propose the integration of a path estimation procedure directly into the scheduling process. Rather than relying on precomputed static plans, the path-finding algorithm operates iteratively, utilizing the most up-to-date information about the environment to generate coherent paths and realistic travel time estimates. This approach ensures that the schedule remains adaptable to the evolving scenario, enabling agents to respond effectively to changes while maintaining a focus on the overall mission objectives. By continuously updating path estimations and task allocations, the system can maintain alignment with real-world developments, avoiding the inefficiencies and risks associated with outdated plans. The proposed allocation and scheduling framework relies on a comprehensive set of inputs to function effectively. These include the set of tasks that compose the mission, each characterized by its spatial position, estimated completion time, required sensors and equipment, priority level, and precedence constraints. Additionally, the system requires continuous updates on the status of each agent, including their location, remaining resources, operational capabilities, and current assignments. A detailed representation of the evolving scenario is also necessary, capturing elements such as the emergence of new hazards, changes in terrain accessibility, and shifting task priorities. One key component of this framework is its ability to perform forward analysis and simulate scenario evolution as the mission progresses. This predictive capability allows the system to anticipate future changes, such as the resolution of existing hazards. By incorporating these predictions into the scheduling process, the framework can proactively adjust allocations and task assignments to maintain operational efficiency and safety.

3.1 Maximizing the Throughput of Tasks

Before the actual assignment process begins, a pre-optimization phase is required to ensure that the system assigns tasks effectively and efficiently. The objective of this phase is to maximize the number of tasks assigned at each step while prioritizing tasks based on their importance and urgency. This strategy not only improves task throughput but also ensures that critical tasks are addressed promptly, aligning with the mission's overarching objectives.

3.1.1 Ready Agents and Tasks

The process begins by identifying the subset of free agents $A_f \subseteq A$, which includes all agents that are not currently occupied with ongoing tasks. This step is essential to prevent task interruptions and ensure that only available agents are considered for assignment. By focusing exclusively on free agents, the system minimizes conflicts during task execution and reduces the likelihood of disruptions caused by reassignments. Starting from the full list of tasks T, a pool of tasks ready to be assigned is then created. This pool, denoted as $T_r \subseteq T$, comprises all tasks that are eligible for immediate execution. A task is considered ready for assignment if it satisfies the following criteria:

- No Unfulfilled Precedence Requirements: Tasks that do not depend on the completion of other tasks or those whose prerequisite tasks have already been completed.
- Not Already Assigned: Tasks that have not been allocated to any agent.

The pool T_r thus represents the set of tasks that can be executed without delay, ensuring that the system avoids idle time and maximizes operational efficiency.

3.1.2 Pre-Assignment Problem

From this pool of ready tasks, a subset $T_a \subseteq T_r$ is computed. This subset represents the largest group of tasks that can be feasibly assigned to the team of agents at the current step, taking into account both task priority and agent capabilities. Priority is a critical factor in this selection process, as it ensures that high-value tasks are prioritized over those of lower importance. This prioritization is particularly crucial in scenarios where resources are limited, and the timely execution of critical tasks can significantly impact mission success. The selection of T_a is carried out through a pre-assignment procedure. This procedure involves solving a mixed-integer optimization problem designed to maximize the number of tasks assigned while adhering to the following constraints:

- Each task in T_a must be executable by at least one agent in A_f, based on the agent's capabilities and available resources.
- All precedence and synchronization requirements associated with the tasks in T_a must be satisfied.
- Task assignments must respect agent-specific limitations, such as energy reserves, equipment availability, and current operational conditions.

The outcome of this pre-assignment procedure is a subset of tasks T_a that can be immediately executed by the team, ensuring that resources are allocated effectively and mission priorities are addressed. This pre-optimization phase is an integral part of the scheduling process, as it sets the foundation for efficient and adaptive task allocation in dynamic and uncertain environments. By focusing on ready tasks and leveraging real-time information about agent availability and task requirements, the system can maintain high levels of efficiency and responsiveness throughout the mission.

$$\mathbf{x}_{ij} = \begin{cases} 1 & \textit{if agent } j \textit{ is assigned to task } i \\ 0 & \textit{otherwise} \end{cases}$$

$$\begin{aligned} \max_{\mathbf{x}_1,\dots,\mathbf{x}_N} \quad & \sum_{i \in T_r} \sum_{j \in A_f} p_i s_j(\mathbf{x}_{ij}) \\ \text{subj. to} \quad & \mathbf{x}_{ij} \in [0,1], \end{aligned} \tag{3.1}$$

$$
\begin{aligned}
&\sum_{i \in T_r} \mathbf{x}_{ij} \leq 1, \quad j \in A_f \\
&\sum_{j \in A_f} \mathbf{x}_{ij} \leq 1, \quad i \in T_r
\end{aligned}
\tag{3.2}
$$

$$
\begin{aligned}
&\mathbf{x}_{ij}(Ca_k^j - Ct_k^i) \geq 0, \quad k = 1, \ldots, c, \\
&\mathbf{x}_{ij}(Ra_k^j - Rt_k^i) \geq 0, \quad k = 1, \ldots, r,
\end{aligned}
\tag{3.3}
$$

$$
S_{kl}\Big(\sum_{j \in A_f} \mathbf{x}_{kj} - \sum_{j \in A_f} \mathbf{x}_{lj}\Big) = 0, \quad \forall k, l \in T_r, k \neq l \tag{3.4}
$$

The objective function of the problem assigns a fixed cost to each task, ensuring that tasks with higher priority are given greater importance in the optimization process. This is achieved by introducing a priority cost p_i for each task i, where a higher p_i indicates a more critical task that should be prioritized over others. Additionally, we introduce a preference factor s_j for each agent j, which determines how desirable it is to assign that agent to a task. A value of s_j closer to 1 signifies a stronger preference for using agent j, whereas a lower value indicates that the agent should be utilized only when necessary. The distinction between p_i and s_j serves different purposes within the scheduling process. While p_i is task-dependent and directly influences the order in which tasks are executed, s_j controls agent selection based on external considerations. This mechanism is particularly useful when certain agents, such as fragile robotic units or human operators, should only be deployed when no other viable alternatives exist. By assigning a lower s_j to such agents, we effectively discourage their selection unless mission-critical conditions require their involvement, ensuring their safety and preserving their operational availability for essential tasks. The first set of constraints enforces the unique assignment condition, ensuring that an agent can be assigned to no more than one task at a given time and that each task is executed by at most one agent. These constraints, formulated in (3.2), are standard within assignment optimization problems and form the foundation of the scheduling framework.

Next, we introduce problem-specific constraints to account for the compatibility between agents' capabilities and task requirements, as defined in (3.3). As previously discussed, Ca_k^j and Ct_k^i describe the presence of capability k on agent j and the required capability for task i, respectively. An agent j can be assigned to task i only if, for every capability $k = 1, \ldots, c$, the condition $Ca^j_k \geq Ct^i_k$ holds, meaning the agent possesses all necessary skills and equipment to complete the task. Similarly, resource constraints are enforced to ensure that agents have sufficient reserves to complete their assigned tasks. Another crucial aspect of the scheduling process is managing tasks that must be executed synchronously. Some tasks require multiple agents to operate in coordination, and failing to enforce synchronization constraints could result in deadlock situations where agents indefinitely wait for other tasks to commence. To prevent this, we introduce the synchronization constraints in (3.4), ensuring that interdependent tasks are allocated simultaneously. To formalize this, we define a synchronization matrix S, where an entry $S_{kl} \in \{0, 1\}$ indicates whether task k must wait for task l. The following cases are considered: - If $S_{kl} = S_{lk} = 1$, then tasks k and l must be assigned together or not at all within the same allocation round. - If $S_{kl} = 0$ and $S_{lk} = 1$, task l must wait for task k to be assigned and executed first, but not vice versa. This allows task k to be scheduled independently while delaying l until a later allocation round. By incorporating these synchronization constraints, we ensure that cooperative tasks requiring multiple agents, such as simultaneous UAV reconnaissance and UGV debris removal are correctly scheduled without causing unnecessary delays or execution bottlenecks. Once the optimization problem is solved (3.1), the system identifies the most valuable batch of tasks $T_a \subseteq T_r \subseteq T$ that can be assigned at the current step. This solution respects all scheduling constraints, prioritization criteria, and agent-task compatibility requirements. To solve this maximization problem, we use the Simplex Algorithm (Appendix A.1), granting that the solutions will be integer, by creating the constraint matrix in such a fashion to obtain a Totally Unimodular (TUM) one (Appendix A.2). This is done with some attention: slack variables are introduced to transform inequalities constraints to equality ones and pivoting techniques are used to align related constraints and variables, minimizing structural overlaps that could result in submatrices with determinants outside the range of $\{-1, 0, 1\}$. Also to simplify the problem, all variables related to unfeasible agent-task pair are removed, to make the computation faster. After determining the optimal task assignments, the next step is to evaluate the travel costs associated with executing them. This is performed using a path-finding algorithm that computes the most efficient paths between free agents and their assigned tasks. To enhance computational efficiency, path computations are restricted to agent-task pairs that are deemed compatible based on capability and resource constraints. Through the integration of optimization in task assignment alongside real-time path evaluation, the system guarantees that agents are assigned the most appropriate tasks while also enabling them to navigate to their destinations in an efficient manner that minimizes risk. This iterative methodology upholds adaptability within dynamic environments, facilitating ongoing reassessment and realignment of mission strategies in response to evolving conditions.

3.2 Travel Time and Risk Estimation

A natural disaster can affect vast geographic areas, creating obstacles and hazards that significantly impact mobility and operational efficiency. Consequently, during mission planning, it is essential to evaluate the spatial distribution of tasks and agents to determine the most effective allocation strategy. Risk-awareness in scheduling is increasingly recognized as a first-order concern for MRS deployed in hazardous, uncertain environments. Recent approaches incorporate probabilistic models of failure or environmental uncertainty into the allocation process, either by embedding chance constraints into MILP formulations or by coupling allocation with online statistical tests that enforce satisfaction thresholds for mission-critical tasks [57]. For instance, methods that integrate sequential hypothesis testing during allocation can bound the probability of task failure below a designer-specified level by triggering redundant allocations or by preferring higher-reliability agents for critical tasks. For our influence-map-based cost function, these findings suggest a natural extension: augment the travel/risk cost with a probabilistic penalty term that reflects not only the expected hazard intensity along a route but also the uncertainty of successful completion (e.g., probability of interruption or agent failure). In practice this can be implemented by computing for each agent-task candidate a risk-expectation (or worst-case quantile) and using a weighted combination of travel-time and failure-probability in the assignment cost. Such an approach preserves the fast, online nature of our scheduler while explicitly managing mission-level risk as done in prior work. This requires an assessment that accounts for the travel time of each agent based on its speed, manoeuvrability, and resistance to environmental hazards. To address this challenge, we implemented a hierarchical path-finding algorithm integrated with influence maps, designed to dynamically adapt to evolving disaster conditions. The approach is structured into two distinct phases:

1. Offline Preprocessing Phase: This phase is computationally intensive but can be performed before the mission begins. It provides a preliminary mapping of the environment, optimizing traversal costs by incorporating static terrain characteristics, known obstacles, and estimated hazard levels.

2. Online Execution Phase: This phase operates in real-time and is invoked repeatedly throughout the scheduling process. It rapidly updates travel cost estimations by incorporating newly discovered obstacles, dynamic environmental changes, and agent-specific constraints, ensuring an adaptive and responsive decision-making process.

By leveraging this dual-phase approach, our method balances computational efficiency and adaptability, allowing agents to navigate disaster-affected environments with optimized route selection while maintaining the flexibility required for real-time mission adjustments.

3.2.1 Offline Phase

Given a two-dimensional operational area, we generate a set of points distributed according to a uniform random distribution. The density of these points is a configurable parameter that directly influences the quality of path estimation. A higher density results in more accurate path length estimations and improved risk minimization, especially for large-scale maps. However, increasing point density also elevates the computational cost of the path-finding algorithm during the online phase. This trade-off must be carefully calibrated for each mission to achieve an optimal balance between precision and execution time. Once the points are placed, we apply the Delaunay Triangulation Algorithm [58], which constructs a triangular mesh by connecting the points to form a set of non-overlapping triangles. The vertices of these triangles serve as nodes in an undirected graph, where edges represent feasible traversal paths between points [Fig. 3.1]. This representation offers multiple advantages:

- Reduced Data Complexity: Compared to traditional grid-based or cell decomposition approaches, the Delaunay Triangulation significantly reduces the amount of information needed to describe the environment while maintaining a high level of accuracy.
- Improved Path Efficiency: The triangulation ensures that paths avoid unnecessary deviations while preserving navigability across complex terrains.
- Adaptive Scalability: The approach can be fine-tuned to different map resolutions by adjusting the point density.

Once the graph is generated, it remains fixed and precomputed, allowing for efficient reuse in each call of the online phase. While the underlying graph structure remains unchanged, additional mechanisms are employed to manage environmental evolution and account for agent-specific constraints. This separation between static map representation and dynamic environmental modelling ensures that the system maintains both computational efficiency and adaptability in rapidly evolving disaster response scenarios.

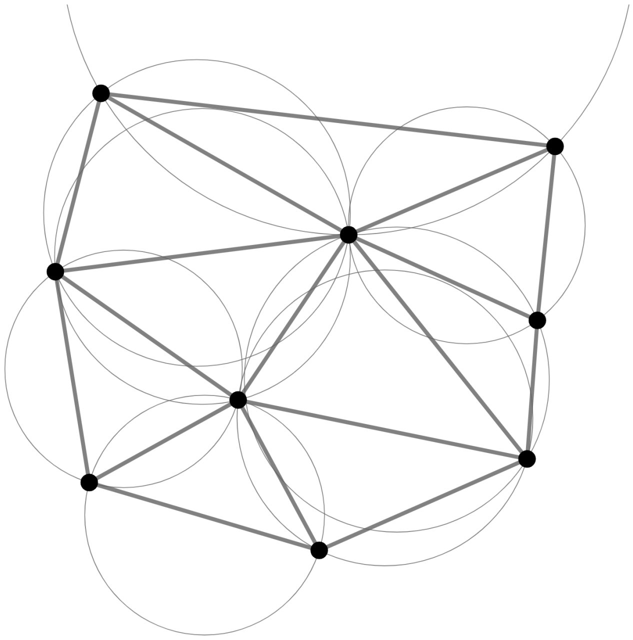


Figure 3.1: *Example of Delaunay Triangulation for a random set of points.*

3.2.2 Online Phase

In the online phase, we integrate an influence map over the precomputed Delaunay triangulated graph to dynamically adapt path planning to environmental hazards and agent-specific constraints. The influence map is represented as a grid, where each cell encodes a traversal cost associated with that region. These costs can be determined based on multiple environmental factors, such as:

- Terrain difficulty, e.g., rough or uneven surfaces that impede UGV mobility.
- Hazard intensity, such as the heat and flames of a wildfire could affect an agent's safety.
- Operational constraints, including no-fly zones or restricted areas must be avoided.

Since the influence map is stored as a bi-dimensional matrix, its computational overhead remains minimal. Each agent generates its version of the map, assigning different traversal costs to various obstacles and hazards based on its mobility capabilities and resistance to environmental risks. This ensures that each agent optimizes its path according to its specific constraints and limitations. To integrate the influence map into the path-planning process, we overlay the triangulated graph onto the cost grid [Fig. 3.2]. We then apply the A^* (Appendix C) search algorithm [59] over the weighted graph, where the cost of each edge is computed as the sum of the influence map values corresponding to the cells it traverses. This approach allows for a flexible and adaptive cost function that considers not only path length but also risk factors and terrain difficulty. A key advantage of using the Delaunay-based graph representation is that it requires fewer nodes compared to a classical grid-based approach while still preserving sufficient accuracy for path estimation. This reduces computational complexity, enabling A^* to execute efficiently in real-time applications, making it feasible for online execution even in large-scale environments. To further enhance path quality, we implement an after-smoothing technique (Fig.3.3, 3.4, 3.5) that refines the initial trajectory produced by A^*. This technique analyzes the generated waypoints and merges consecutive segments whenever possible, provided that the direct path between them does not introduce a higher

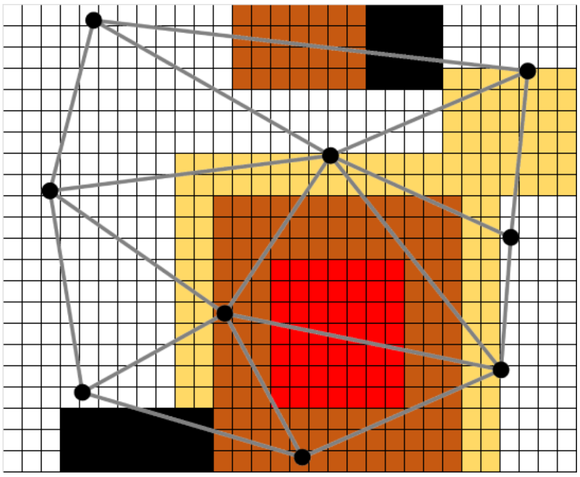


Figure 3.2: *An example of the graph computed by the Delaunay triangulation algorithm overlapped with the influence map. Black regions are physical objects or no-fly-zone and coloured squares represent a different degree of risk or travelling hardness. Edges that cross black areas are not traversable.*

traversal cost than the one already incurred. This smoothing step reduces unnecessary detours and generates a path closer to the true optimal trajectory, improving both efficiency and feasibility in real-world applications. One of the key strengths of the influence map-based method is its flexibility in defining traversal costs. By adjusting the way costs are assigned, the approach can be used to compute:

- Shortest paths (pure geometric distance).
- Estimated travel time for a specific agent, factoring in terrain resistance and movement speed.
- Risk-aware paths, where hazardous regions contribute an augmented cost, discourage agents from crossing dangerous areas unless necessary.

By leveraging this adaptive cost modelling, the influence map approach provides a versatile and computationally efficient solution for real-time path planning, ensuring that agents navigate their environment with optimized decision-making based on their specific mission objectives and constraints.

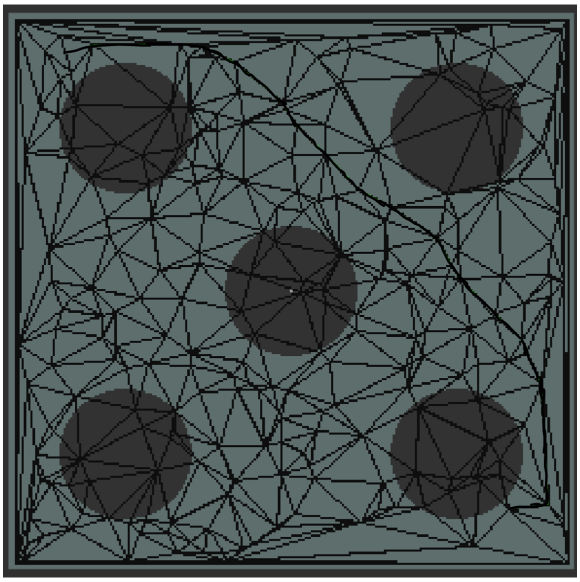


Figure 3.3: *Online Phase: Influence map overlap with Delaunay Triangulation.*

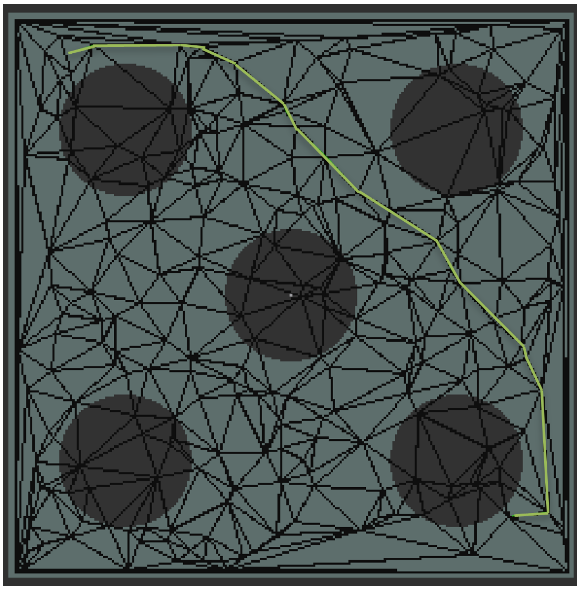


Figure 3.4: *Online Phase: A^* search on the graph.*

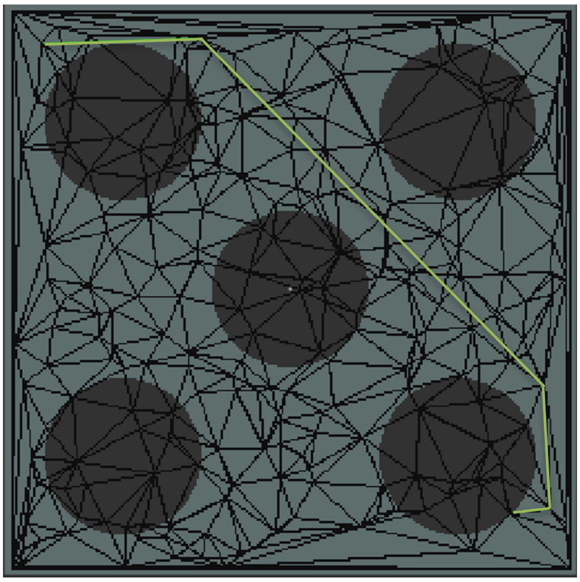
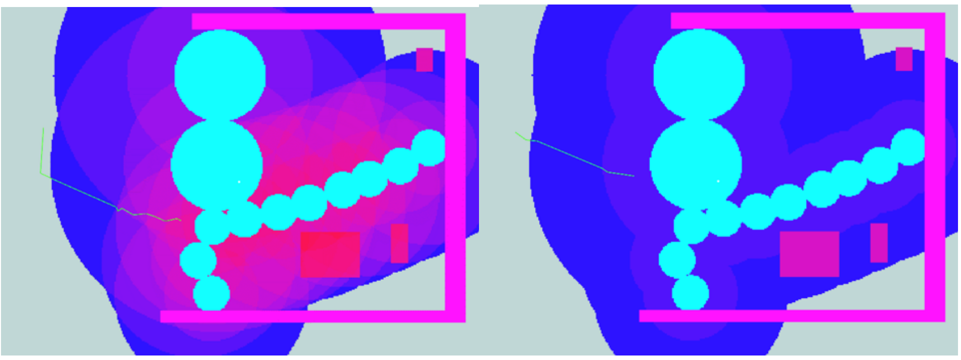


Figure 3.5: *Online Phase: Path Smoothing and rectification for improved estimation.*



(a) (b)

Figure 3.7: *Example of differentiated influence map. They could represent maps for different agents.*

3.3 Task-to-Agent Assignment Optimization

At this stage, we focus on assigning the previously selected ready-to-be-assigned tasks T_a to the free agents of the team, denoted as A_f. The objective is to ensure that each task in T_a is assigned to a distinct agent in a way that minimizes the overall operational cost. The problem formulation (3.5) follows a structure similar to the previous assignment step; however, in this phase, the assignment constraints (3.6) are stricter, requiring that each task be allocated to a different agent. Additionally, the deadlock avoidance constraint (3.4) is no longer necessary, as it has already been satisfied in the previous scheduling stage. Synchronization requirements and heterogeneous resource constraints pose significant challenges for MRTA: tasks may demand simultaneous presence of agents with different skill-sets (coalition formation), and agents vary in endurance, payload, or survivability. Several recent frameworks explicitly model these coalition requirements and propose scalable heuristics that jointly decide membership and timing of multi-agent tasks [60]. These methods typically trade exactness for scalability by using incremental coalition formation, market-based bidding, or greedy selection of candidate coalitions that satisfy required skill-sets. An alternative and complementary direction exploits decentralized learning: multi-agent reinforcement learning (MARL)[61] methods have been shown to learn coordination patterns enabling synchronous task execution with low runtime overhead [62]. However, MARL approaches tend to require careful reward shaping and significant training data; they are most effective when a representative set of scenarios can be used for offline training. For the scheduler proposed in this thesis, a pragmatic hybrid is recommended: keep a central, fast heuristic for coalition feasibility checks and synchronization enforcement (as currently implemented), and consider a learned policy to suggest promising coalition candidates or to tune heuristic hyperparameters online. This hybrid yields the scalability of heuristics together with the adaptivity of learned controllers.

3.3.1 Optimization Criteria

The optimization objective at this step is to minimize the path length and risk cost associated with completing the tasks in T_a. The assignment cost, denoted as w_{ij}, represents the cost of allocating task $i \in T_a$ to agent $j \in A_f$. This cost can be defined in multiple ways, depending on the mission requirements:

- Path Length (d_{ij}): The total distance that agent j must travel to reach and complete task i.
- Risk Cost (r_{ij}): The accumulated risk incurred by agent j while travelling to and executing task i, which may include factors such as en-

vironmental hazards or proximity to restricted areas.

- Weighted Combination ($\alpha d_{ij} + \beta r_{ij}$): A customized cost function that balances path length and risk, where α and β are weighting factors determined based on mission priorities.

The values of w_{ij} are obtained through the path-finding algorithm, which computes the optimal traversal route for each agent-task pair while incorporating environmental constraints and agent-specific parameters. By adjusting the weights in the cost function, we can prioritize different operational strategies, such as minimizing travel time, avoiding high-risk areas, or achieving a trade-off between efficiency and safety. Since each task must be assigned to a different agent, this problem can be formulated as a minimum-cost assignment problem, which can be efficiently solved using combinatorial optimization techniques such as the Hungarian Algorithm or Integer Linear Programming (ILP). Given the dynamic nature of the environment, real-time adjustments may be necessary if unexpected obstacles or new mission constraints arise. By leveraging this structured assignment approach, we ensure that agents are allocated tasks in an optimal and computationally efficient manner, ultimately enhancing the adaptability and robustness of the mission execution process.

$$\mathbf{x}_{ij} = \begin{cases} 1 & \textit{if agent } j \textit{ is assigned to task } i \\ 0 & \textit{otherwise} \end{cases}$$

$$\begin{aligned} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_N} \quad & \sum_{i \in T_a} \sum_{j \in A_f} w_{ij}(\mathbf{x}_{ij}) \\ \text{subj. to} \quad & \mathbf{x}_{ij} \in [0,1], \end{aligned} \tag{3.5}$$

$$\begin{aligned} \sum_{i \in T_r} \mathbf{x}_{ij} = 1, \quad j \in A_f \\ \sum_{j \in A_f} \mathbf{x}_{ij} = 1, \quad i \in T_a \end{aligned} \tag{3.6}$$

$$\begin{aligned} \mathbf{x}_{ij}(C_k^j - C_k^i) \geq 0, \quad k = 1, \ldots, c, \\ \mathbf{x}_{ij}(R_k^j - R_k^i) \geq 0, \quad k = 1, \ldots, r. \end{aligned} \tag{3.7}$$

3.4 Post-Assignment Evaluation and Scheduling Update

Once the task assignment process is completed, the next step involves evaluating how this allocation affects the state of the system. This phase ensures that all scheduling constraints are properly enforced and that the synchronization dependencies between tasks are maintained. The scheduling process begins by analyzing how long each assigned agent requires to reach and complete its designated task. Since certain tasks have temporal dependencies, specific adjustments are made to align task execution with predefined constraints. The following conditions are checked and enforced:

- End-After Constraint: If task T_i is required to conclude only after another task T_j has been completed, the system checks the estimated completion time of T_j. If T_j finishes later than the nominal estimated conclusion time of T_i, the execution duration of T^i is extended accordingly to ensure both tasks conclude simultaneously.
- Start-While Constraint: If task T_i must start while another task T_j is in progress, but the current schedule places T_i too early, the system introduces idle time for the agent assigned to T_i so that both tasks can commence together.
- Cooperative Task Constraint: When two tasks must start and end simultaneously, the End-After and Start-While constraints are applied reciprocally to both tasks, ensuring they are scheduled to begin and conclude at the same time.
- Start-After Constraint: If a task is scheduled to begin before a predefined time step, idle time is introduced so that the assigned agent arrives at the task location precisely when execution is permitted.

3.4.1 Environmental State Update and Simulation Progression

After enforcing synchronization constraints, the system **identifies the task with the earliest estimated completion time** and checks whether its execution modifies the environment. The simulation progresses to this time step, and the following updates are applied:

1. Agent State Updates: Each agent's attributes, such as remaining resources, current position, and availability status (free or occupied), are updated based on travel and execution time estimations computed by the path-finding module.

2. Environmental Modifications: The influence map is updated to reflect changes introduced by completed tasks. For example, if a task involves fire suppression, the risk values in the corresponding area are reduced to reflect the new, safer conditions.

3. Task Iteration and Continuation: The scheduling loop repeats, following these steps:

    - Selection of ready-to-be-assigned tasks.
    - Maximization of task throughput based on priority and feasibility.
    - Path cost evaluation.
    - Task allocation and synchronization adjustment.
    - System state update and simulation progression.

This iterative process continues until either all tasks are assigned and completed, or an infeasibility condition is encountered, e.g., when no available agent possesses the required resources or capabilities to execute a remaining task.

3.4.2 Handling Dynamic Environmental Changes and Scheduling Reinvocation

Given the unpredictable nature of real-world operations, the environment may change unexpectedly due to external factors such as new hazards emerging (e.g., sudden fires, floods, or structural collapses), resource depletion requiring strategic reassignment, or Weather changes affecting agent manoeuvrability and task feasibility. Whenever such unforeseen alterations occur, or when modifications to the task list or optimization parameters are introduced, the scheduling process can be dynamically reinvoked. The new scheduling instance uses updated environmental data and the real-time status of the agent team as the initial conditions for rescheduling. This ensures that the system remains adaptive and responsive to mission-critical changes, maintaining operational efficiency even in highly dynamic and uncertain environments.

3.5 Test and Results

To rigorously evaluate the performance of our algorithm, we conducted extensive tests across various scenarios, incorporating different environmental conditions, agent configurations, and task distributions. The primary objectives of these experiments were to assess the robustness, versatility, and computational efficiency of our approach compared to existing methods. The family of heuristic and metaheuristic algorithms applied to MRTA is broad and includes market-based auctions, evolutionary multi-objective optimizers, and swarm-based methods. Recent hybrid designs combine auction mechanisms with multi-objective metaheuristics (e.g., NSGA-III) to select coalitions and trade-off competing metrics such as makespan, energy, and risk exposure; these hybrid solutions show favorable Pareto fronts in large-scale simulation studies [63]. Similarly, multi-phase particle swarm optimizers (PSO) and modified swarm-based metaheuristics have been applied to dynamic aggregation problems where tasks require temporally coordinated multi-point rendezvous of heterogeneous agents [64]. Compared to pure sequential auction-based allocation, batch-based and hybrid metaheuristic strategies can find richer combinations of tasks to be executed concurrently, thereby reducing deadlocks and improving global efficiency. However, metaheuristics typically demand more tuning and can be computationally heavier than strictly greedy algorithms; thus, an effective system often exposes tunable modes (fast/greedy vs. improved/metaheuristic) that the operator can select according to mission urgency. One of the most relevant benchmarks for evaluating our method is the auction-based allocation algorithm presented in [35]. This approach assigns tasks sequentially, processing one task at a time in a greedy fashion. While effective in structured settings, this method struggles in scenarios that require optimizing task dependencies, cooperative execution, and dynamic reallocation. Our algorithm significantly improves computational efficiency and solution quality by adopting a batch-based allocation approach, where multiple tasks are assigned at each step. This approach enables better task combinations, reduces total mission execution time, avoids deadlocks, and ensures a more effective allocation of cooperative tasks. Benchmark experiments designed with strong precedence constraints and cooperative execution requirements, such as UAV reconnaissance supporting UGV ground intervention, show that our approach could reduce total mission time by up to 30% compared to sequential auction-based methods, particularly in high-density task environments.

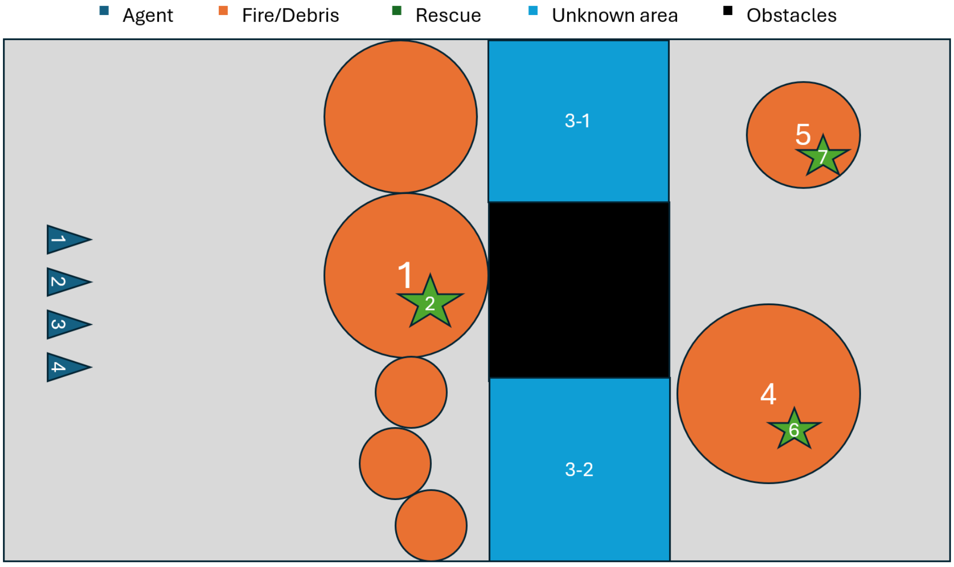


Figure 3.8: *Example scenario: Four agents have to reach 3 goals while managing environmental hazards. The numbers define correlations between elements and tasks.*

To further assess the scalability of our method, we tested its performance in increasingly complex missions, varying the number of tasks from ten to over a hundred while also adjusting the composition of the agent team. The results confirm that our algorithm maintains near-linear growth in computational time as mission complexity increases. Unlike auction-based methods, which exhibit exponential growth due to sequential allocation, our approach remains efficient, making it viable for large-scale operations. Another key strength of our solution is the integration of Delaunay Triangulation with the A* pathfinding algorithm. Traditional grid-based methods, while highly precise, require significant computational resources, making them impractical for large-scale applications. By leveraging a Delaunay-based approach, we achieved a substantial reduction in execution time, cutting computational costs by up to 40% while maintaining effective path estimation. The efficiency of this method is particularly evident in dynamic scenarios, where rapid path recomputation is required. Additionally, our system allows for configurable trade-offs between precision and computational speed, making it adaptable to different mission requirements. Higher accuracy can be prioritized when necessary, while faster execution can be selected when real-time performance is critical. The experimental results highlight several key strengths of our approach. The batch-based task allocation mechanism optimally distributes tasks among agents, significantly improving mission efficiency compared to sequential methods. The scheduling framework scales effectively, ensuring computational feasibility even in missions involving large numbers of tasks and multiple heterogeneous agents. The integration of an optimized pathfinding strategy reduces computational overhead while maintaining navigational accuracy, ensuring that agents operate efficiently in complex environments. Furthermore, the system demonstrates strong robustness in dynamic conditions, as it can adapt to unforeseen changes without requiring a full recomputation of previous allocations. Overall, our algorithm has shown clear advantages over conventional approaches, particularly in handling large-scale, dynamic, and cooperative task scheduling problems. The combination of batch-based allocation, adaptive pathfinding, and real-time reallocation ensures that the system remains both scalable and robust. These characteristics make our method well-suited for mission-critical autonomous systems operating in unpredictable environments. Future research will focus on further optimizing real-time scheduling performance and exploring advanced multi-objective cost functions to enhance adaptability across different application domains.

Table 3.1: *Test with A^* algorithm on a 256x256km grid-map.*

Number of cells	Cell Size [km]	Mean execution time [ms]	Path Length [km]
256	16	3	358
1024	8	16	347
4096	4	54	337
16384	2	183	336
65536	1	685	335

It is essential to clarify that the primary purpose of the proposed pathfinding tool is to provide an estimation framework for mission planning rather than serving as the actual pathfinding or trajectory-planning mechanism for individual agents. The tool is designed to assess the relative proximity between agents and their assigned tasks, while also evaluating the obstacles and risks that must be considered during mission execution. By offering an approximate yet computationally efficient representation of the operational environment, this method enables optimized task allocation and scheduling, ensuring that mission objectives are met effectively. However, each agent will rely on its dedicated navigation system to compute fine-grained trajectories during execution, allowing for more precise real-time adjustments based on local conditions.

Table 3.2: *Test with A^* algorithm with Delaunay triangulation in the same conditions as in Tab.3.1. We compared the result of this approach against the highlighted result of Tab.3.1.*

Number of nodes	Mean execution time [ms]	Speed up factor	Path degradation
256	6	30.5	6.8%
512	13	14.1	6.5%
4096	21	8.7	2.0%
16384	54	3.4	1.2%
65536	72	2.6	0.8%

To illustrate the applicability of our approach, we present an example scenario depicted in Fig.3.8, representing a disaster response environment where a rescue robot team is deployed. The orange circular areas indicate critical obstacles that must be mitigated to reach individuals in need of rescue, represented by green stars. Impassable barriers, such as collapsed structures or hazardous terrain, are denoted in black, while light blue areas highlight unexplored regions where no prior information is available. The mission objectives in this scenario consist of rescuing three individuals and exploring unknown areas while strategically managing environmental hazards, such as fire or debris zones, which must be cleared to facilitate access to victims. Each operational step, whether it involves extinguishing a fire or performing an extraction, follows a structured process: first, a preliminary analysis of the target area is conducted to assess feasibility; then, the operation is executed; finally, a validation step is required to confirm successful task completion. A detailed overview of agent-task compatibility for this specific mission is provided in Table 3.3, outlining which agents are best suited for each operation based on their capabilities. Additionally, Fig. 3.9 and

3.10 illustrate two experimental trials performed on this scenario, involving a total of 23 diverse tasks, executed by four heterogeneous agents, under two distinct scheduling parameterizations. The first configuration optimizes overall travel distance, aiming to minimize energy consumption and maximize operational endurance. The second configuration focuses on minimizing mission completion time and prioritizing task concurrency and efficiency to ensure rapid intervention. The flexibility of the proposed algorithm allows for adaptability to different mission requirements. Depending on operational priorities, the system can be reconfigured to emphasize different objectives, such as identifying safer routes that minimize risk exposure, enforcing task prioritization strategies, or preserving specific agents by restricting their involvement in hazardous areas unless necessary. This adaptability ensures that the algorithm can be tailored to various real-world scenarios, enhancing the reliability and effectiveness of autonomous multi-agent systems in disaster response and other mission-critical operations. The total computational time required for scheduling and path optimization is inherently dependent on both hardware capabilities and mission complexity. The primary factors influencing execution time include the number of nodes used for the Delaunay Triangulation and the number of iterations necessary to generate a complete schedule. Since these elements scale with the size of the operational environment and the number of tasks and agents involved, the algorithm remains adaptable to a wide range of scenarios while maintaining real-time feasibility. In our experimental setup, assuming an average pathfinding evaluation time of approximately 60 milliseconds per query, combined with the rapid resolution of both the minimization and maximization-based task allocation problems, a mission of complexity comparable to the example scenario presented was computed in an average time of approximately 2000 milliseconds. This result was obtained from tests conducted on a portable workstation equipped with an AMD Ryzen 4700U processor, demonstrating the computational efficiency of the proposed approach even on consumer-grade hardware. These performance metrics emphasize the suitability of our algorithm for dynamic, time-critical environments where rapid decision-making is essential. The ability to continuously update task assignments and recompute paths in response to environmental changes ensures adaptability in unpredictable conditions, making the approach highly relevant for applications that require multi-agent coordination in real-time operations.

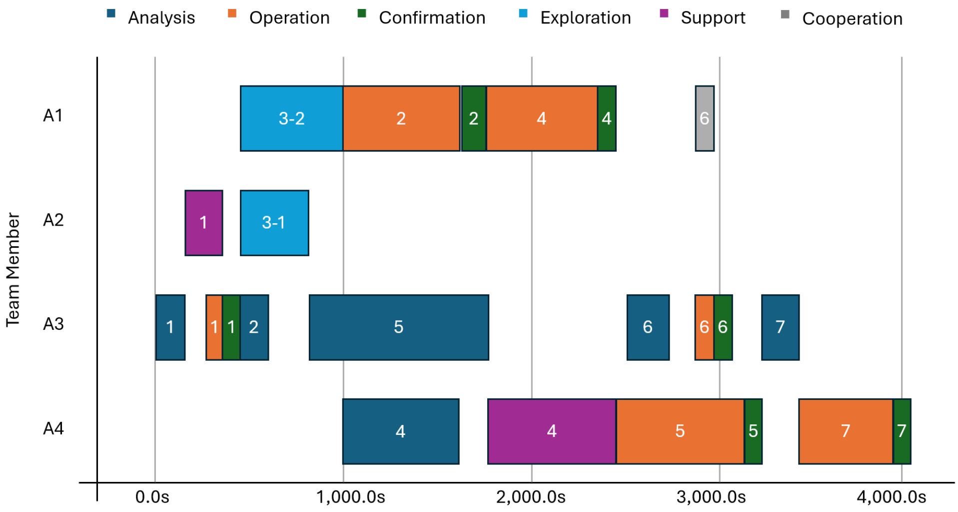


Figure 3.9: *Scheduling minimizing overall distance travelled.*

The efficiency and robustness of the proposed method make it well-suited for disaster response missions, where quick deployment and adaptive reallocation are crucial for maximizing operational effectiveness. Furthermore, its applicability extends to search and rescue, surveillance, logistics, and autonomous fleet management, where optimizing agent collaboration and minimizing response times are fundamental objectives.

Table 3.3: *Agent-Task compatibility.*

	Analysis	Operation 1-2-4-6	Operation 5-7	Confirmation	Support	Cooperation	Exploration
A1	x	x		x	x	x	x
A2	x			x	x		x
A3	x	x		x		x	x
A4	x		x	x	x		

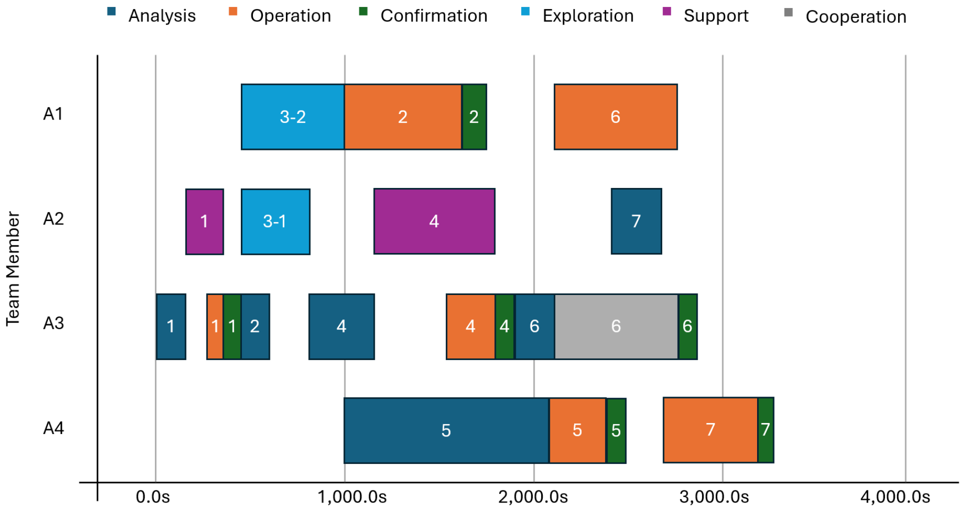


Figure 3.10: *Scheduling minimizing completion time.*

Chapter 4

Tactical Planning for MRAS

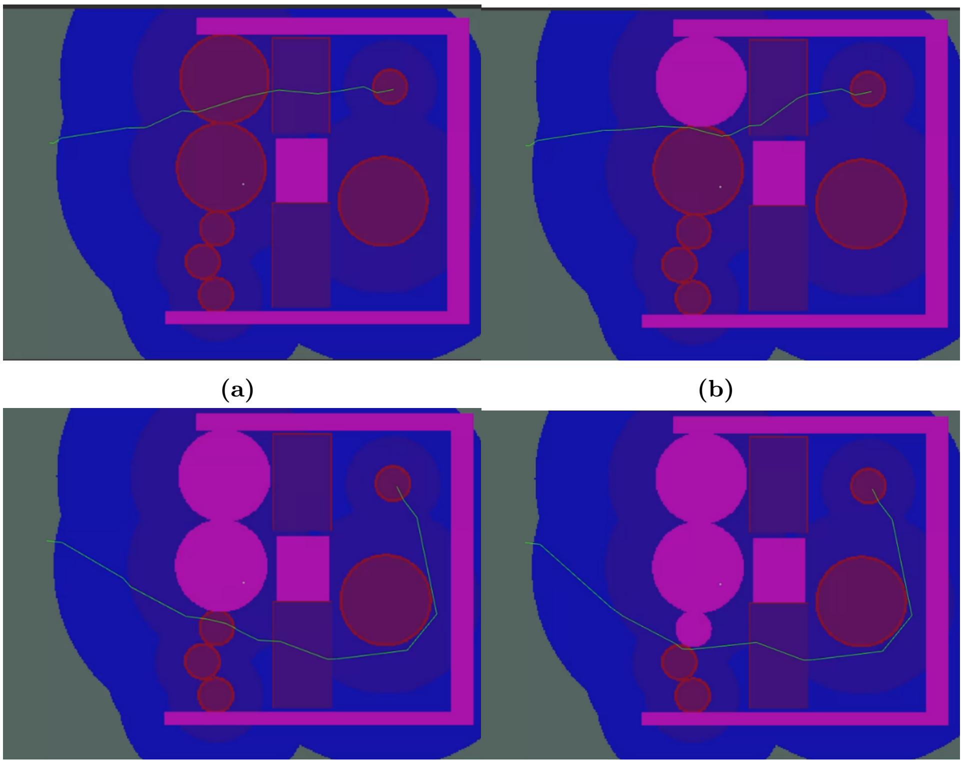
Between high-level task allocation and low-level control lies the tactical layer, which refines abstract mission goals into executable plans. Behavior Trees (BTs) have emerged as a widely adopted representation for modular and reactive control [65]. Heppner [44] extends this paradigm with runtime, capability-aware task allocation, enabling heterogeneous robots to dynamically bind to BT nodes. Other tactical approaches focus on hybrid refinements: Sung et al. [66] formulate asynchronous task?motion refinement, while Hustiu et al. [45] exploit Petri nets to integrate mission logic and robot behavior in a formal framework. More recently, Cai et al. [67] propose MRBTP, an algorithm to automatically generate multi-robot BTs with guarantees, even exploring the use of large language models for subtree synthesis. These works demonstrate that the tactical layer is increasingly central to bridging scheduling and execution ? precisely the niche targeted by our Tactical Planner Algorithm. The solution we propose is a sophisticated software framework designed to enhance mission planning and execution in complex and dynamic environments. Building upon the principles of the widely used A^* pathfinding algorithm (Appendix C), it extends traditional pathfinding capabilities by incorporating risk assessment, resource optimization, and real-time adaptability. This system's primary objective is to identify the optimal route to reach a specific goal while minimizing operational risks and resource consumption, ensuring efficient mission execution across diverse scenarios. Unlike conventional pathfinding approaches, which focus solely on finding the shortest path, the Tactical Planner Algorithm integrates a dynamic analysis of environmental challenges and mission-specific constraints. This allows for the formulation of highly adaptive and strategic plans, making it suitable for applications ranging from disaster response and surveillance to military operations and autonomous robotic navigation.

4.1 Design and Principles

The Tactical Planner is designed as an advanced support tool for the autonomous management of missions involving Multi-Robot Autonomous Systems (MRASs). One of its fundamental capabilities is the automated generation of tasks, which is essential for ensuring a smooth and rapid response to unexpected events that necessitate mission re-planning. Our approach proposes the development of a flexible tool, adaptable to specific operational scenarios, that enables the selection and optimization of mission objectives by dynamically structuring the task allocation process. Building upon a variation of the Path Planning algorithm introduced earlier 3.2, the Tactical Planner is designed to identify “access routes” to mission targets. The algorithm computes multiple alternative paths connecting the mission’s starting point to each objective while simultaneously identifying the obstacles that must be addressed along the way. The optimization criteria focus on minimizing both the travel distance and the operational effort required to reach each target. For instance, in a post-earthquake rescue scenario, if the objective is to reach a victim trapped under rubble, the Tactical Planner prioritizes a route that requires the least amount of debris removal, thereby accelerating the rescue process and reducing unnecessary workload. To ensure adaptability to specific operational constraints, the Tactical Planner allows for the parameterization of task generation. This feature is particularly relevant when the execution of mission objectives requires a strict sequence of actions. In scenarios where debris removal, reconnaissance, and victim extraction must be carried out in a coordinated manner, the planner structures its output accordingly. The result is an ordered task list, complete with precedence constraints and resource requirements, allowing for the automatic generation of a comprehensive mission plan from start to finish. Once generated, the task output is processed by the scheduling algorithm presented earlier, ensuring that the tasks are optimally assigned to available agents while accounting for mission priorities, resource availability, and environmental constraints. In the event of significant environmental changes, the

Tactical Planner is capable not only of re-scheduling tasks but also of replanning them entirely, dynamically restructuring mission objectives in response to real-time conditions. This ability to seamlessly integrate task generation, scheduling, and real-time adaptation ensures that the MRAS can operate autonomously and efficiently in highly dynamic and unpredictable environments. Just as our pathfinding algorithm used in the scheduling phase can be tailored to different objectives by adjusting the risk and distance weighting factors, the Tactical Planner also offers a high degree of flexibility in its decision-making process. By modifying the interaction costs and complexity factors associated with obstacles and mission targets, the system can generate a wide range of solutions tailored to specific operational requirements. The algorithm can be configured to prioritize obstacle avoidance, even if this results in longer and more circuitous routes, thereby minimizing risk exposure. Alternatively, it can be adjusted to favor more direct routes, optimizing for shortest travel distances while proactively engaging with encountered obstacles. This adaptability enables mission planners to balance efficiency, risk mitigation, and resource expenditure according to mission constraints and priorities. As illustrated in Fig.(4.1), the system generates a set of alternative paths, allowing for a broader decision space for the mission supervisor, whether human or autonomous. The final selection of the optimal path is determined based on the resources required to traverse each route, weighed against the capabilities available within the team. This multi-path approach enhances strategic flexibility, ensuring that mission execution remains resilient and adaptable to evolving conditions.

While the Tactical Planner already provides cost-aware task generation and adaptive pathfinding, an interesting direction for future development is to combine its outputs with Behavior Tree (BT) structures. In this view, the Tactical Planner would focus on evaluating the cost of each task, identifying the supporting subtasks required, and defining the associated constraints (e.g., precedence or synchronization). The resulting evaluations could then serve as inputs to a mission-level BT, which would assemble the planner?s results into an executable sequence of actions with explicit ordering, concurrency, and fallback mechanisms [68]. Such an integration would allow the BT to act as a reactive execution layer, enforcing the Tactical Planner's optimization objectives while providing built-in adaptability through fallback and recovery nodes. This synergy could strengthen the bridge be-



(a) **(b)**

(c) **(d)**

Figure 4.1: Different paths created by the Path Planner module. Each one crosses a different set of obstacles.

tween strategic mission planning and robust low-level execution, especially in domains where resilience to unexpected events is critical.

4.2 Algorithm Description

The fundamental operation of the Tactical Planner is straightforward: it iteratively searches for a path from the starting point to a target, records the list of encountered obstacles, marks the first encountered obstacle as "to be avoided," and recalculates the path to explore alternative access routes. The tool is configured to search for a selectable maximum number of alternative paths and will terminate early if, at iteration n, no further alternative access routes are found. This process is repeated for all mission targets. Once all alternative path lists for all targets are obtained, a selection and ordering process is required. This involves analyzing tasks shared among multiple targets to minimize their number, evaluating which tasks are more onerous and complex, and assessing whether the total resources required by the tasks are covered by the team's capabilities. Currently, as an initial iteration of the tool, we employ a simple system that seeks only the minimum number of tasks necessary to reach all targets. Ongoing considerations aim to evaluate the creation of the task list more accurately, delving into complex study areas and combinatorial optimization. At this stage, the algorithm is relatively simple, acting as a proof-of-concept. Despite its simplicity, it is designed to systematically calculate and optimize paths and tasks for reaching multiple mission targets while considering obstacles and constraints. For each target, the path planner calculates N alternative paths. Along each path, a list of obstacles encountered is generated in a sequential order: obstacles are listed from closest to the starting point to farthest, ending with those nearest to the target. Maintaining this order is crucial because, in later stages, it ensures that tasks corresponding to these obstacles are executed in the correct sequence. For instance, if the path to a target includes obstacles A, B, and C in that order, the list will reflect this sequence. Once the planner generates alternative paths for all targets, these obstacle lists must be combined into a unified set of tasks. However, this process is not straightforward and involves:

- Consolidating Redundant Obstacles: Some obstacles may appear in multiple paths (e.g., if two targets share a portion of their routes). To optimize the plan, these duplicate obstacles are compacted into a single instance to avoid repeated tasks.
- Tracking the Original Lists: Each individual list must be preserved to ensure that any specific precedence requirements for tasks (based on the obstacle order within the original list) are respected.

Each obstacle is assigned a cost that reflects the complexity of overcoming it. The cost depends on various factors, such as:

- Obstacle Type: Is it a physical barrier, a hostile zone, or a no-fly area?
- Management Complexity: Does it require specialized equipment, additional time, or specific agent capabilities?
- Risk Level: How dangerous is it to approach or manage the obstacle?

These costs allow the algorithm to quantify the difficulty of a mission plan. If for instance, we set the cost for each obstacle to 1, then the computed solution will valorize only the number of obstacles over their hardness. Once the costs of individual obstacles are calculated, the total cost of each mission plan (i.e., the combined obstacle lists) is computed, and then the algorithm selects the mission plan with the lowest total cost, prioritizing efficiency and feasibility. Using the information from the combined lists, the algorithm generates macro-tasks to guide the mission. A macro-task refers to a high-level task that has not yet been decomposed into simpler, actionable tasks that can be directly assigned to individual agents. It acts as an overarching objective that will later be divided into smaller, more specific tasks, each tailored to the capabilities and roles of the agents involved. For example, in a search and rescue scenario, a macro-task such as 'extinguish fires in a designated area" might involve several sub-tasks, including: Scanning the area for hotspots using aerial drones equipped with thermal cameras; deploying robots equipped with firefighting tools to extinguish flames in localized zones; coordinating logistics to deliver firefighting equipment and resources to human operators in the field. This hierarchical approach enables mission planners to define broad objectives (macro-tasks) and then refine them into executable tasks that align with the mission's operational constraints and the team's capabilities. This process ensures that the system can adapt to complex scenarios while maintaining clarity and coherence in task allocation and execution. These macro-tasks are arranged in the proper sequence, respecting the obstacle order to ensure the mission unfolds logically and efficiently. It is important to maintain the proper order to abide by the precedence constraints generated by the path-finder algorithm. Those macro-task that are not bounded by precedence constraints can be executed in parallel with other tasks. While the algorithm is still in its early stages, these steps lay the groundwork for a powerful and adaptable mission planning tool capable of addressing real-world scenarios with a high degree of efficiency and reliability.

4.3 Future Development

The Tactical Planner plays a crucial role in autonomous mission management, providing a structured and adaptive approach to task generation, pathfinding, and resource-aware decision-making. While the current implementation already offers a flexible, multi-objective planning framework, several key areas could be further explored to enhance its capabilities. Below are some proposed advancements that would extend its applicability and improve overall system efficiency, to further enhance the autonomy of MRAS in Disaster Response scenarios.

4.3.1 Integration of Machine Learning for Dynamic Path Optimization

One potential enhancement involves incorporating machine learning techniques to improve path selection and mission adaptability. By training the Tactical Planner on historical mission data, agent performance, and environmental conditions, the system could refine its path-selection criteria over time. A reinforcement learning approach, for instance, could allow the planner to learn optimal engagement strategies for different types of obstacles, dynamically adjusting its weight parameters based on mission outcomes. This would reduce reliance on manually defined heuristic functions and improve autonomous decision-making in previously unseen environments. A Machine Learning approach could also enhance the task generation part of the tool, optimizing the resources of the team.

4.3.2 Probabilistic Risk Assessment for Uncertain Environments

Currently, the Tactical Planner evaluates risks and obstacles using predefined costs and deterministic models. However, in real-world scenarios, environmental conditions and threat levels are often uncertain. Incorporating probabilistic models would allow the system to quantify uncertainty and compute risk-adjusted mission plans. For example, Bayesian networks or Monte Carlo simulations could be used to estimate the likelihood of obstacles “changing position” (e.g., landslides, fire spread) or becoming “more hazardous” over time, enabling the planner to proactively adjust mission routes.

4.3.3 Human-in-the-Loop Decision Support Systems

Although the Tactical Planner is designed to operate autonomously, integrating a human-in-the-loop (HITL) mechanism would enhance its usability in critical missions. A real-time decision support interface could allow human operators to intervene, modify task priorities, or override automated choices based on situational awareness. By leveraging explainable AI (XAI) techniques, the system could also justify its recommendations, offering clear visualizations of why a particular path or task sequence was chosen, thus improving transparency and trust in AI-driven decision-making.

4.4 Consideration

Several studies confirm the effectiveness of generating multiple alternative routes for robotic navigation. For instance, Gautier et al. [69] emphasize that in multi-agent systems it is common practice to compute the k-shortest paths, thereby maintaining a diverse set of routing options. This strategy closely resembles the operation of the Tactical Planner, which iteratively explores new access routes by marking obstacles to be avoided and recalculating paths accordingly. From an algorithmic perspective, the classical A* algorithm [37], [70] and its dynamic extension D* [38] remain well-established methods for obstacle-aware path planning, with D* particularly suited for recomputing trajectories online as new obstacles are discovered. Other techniques, such as space decomposition approaches, can similarly produce multiple alternative solutions by connecting free-space cells [71]. Obstacle avoidance itself has long been recognized as a fundamental capability: Katona et al. [72] stress that reliable obstacle-avoidance methods are essential to ensure that robots reach their goals without collision. In this respect, the Tactical Planner adopts a consistent approach by systematically combining the generation of alternative routes with explicit recording of encountered obstacles. The subsequent step (selecting and ordering tasks required to cover all mission targets) naturally leads to a combinatorial optimization problem. Literature widely acknowledges that the integration of multi-robot task allocation and path planning is NP-hard, requiring heuristics or approximate solutions. Xu et al. [73] specifically highlight how obstacle avoidance must be incorporated into the evaluation of inter-task routes in multi-robot settings. Similar issues are addressed in frameworks that combine centralized task allocation with distributed trajectory refinement, such as SPADES [74], which minimizes mission completion time while adapting to dynamic environments. The current Tactical Planner design, while simplified, follows this rationale by attempting to minimize redundant tasks shared among different mission targets and ordering them according to obstacle precedence. This mirrors the basic principles of task allocation problems, in which tasks must be efficiently distributed and sequenced under resource constraints. Although still operating as a proof-of-concept, the methodology aligns with findings in related studies. As in [73] is pointed out, multi-objective planning with environmental constraints is inherently complex and typically addressed through heuristics or combinatorial optimization. The Tactical Planner's iterative generation of alternative paths and reduction of redundant tasks provides a pragmatic entry point to these problems. More advanced approaches in the literature employ genetic algorithms, mathematical programming, or learning-based strategies to handle computational load and adapt to dynamic conditions. In summary, the adopted methodology is consistent with well-established practices in robotics: using alternative path generation to overcome obstacles and applying combinatorial reasoning to minimize shared tasks. Enhancing the Tactical Planner through machine learning, probabilistic modelling, and HITL integration would significantly improve its effectiveness across autonomous and semi-autonomous mission scenarios. These proposed advancements would increase resilience, reduce mission execution time, and enhance adaptability to evolving operational conditions. As autonomous multi-robot systems continue to evolve, integrating these features will be crucial for developing next-generation mission planning frameworks capable of handling increasingly complex and unpredictable environments.

Chapter 5

Consideration and Conclusion

The primary objective of this research was to develop a decision-making framework capable of supporting complex, high-risk operations, particularly in search and rescue and disaster response scenarios. The proposed solution was designed to be robust against unforeseen events, allowing for real-time adaptability in highly dynamic environments. By leveraging an advanced scheduling framework and path-planning mechanisms, the system ensures that task allocation remains efficient, optimizing the coordination of autonomous agents even in rapidly evolving mission conditions. Extensive simulation-based evaluations have confirmed the system's ability to efficiently allocate resources, prioritize critical tasks, and adjust dynamically to environmental disruptions, demonstrating its potential applicability in real-world deployment scenarios. However, achieving full autonomy in multi-robot systems requires the integration of additional intelligent components to enhance both decision-making capabilities and operational independence. While the current system effectively manages task allocation and scheduling, a critical area for future development lies in the automation of the task-generation process. At present, task generation is predefined or relies on external inputs, but for a multi-robot system to operate autonomously in disaster scenarios, tasks must be identified, classified, and prioritized dynamically based on real-time sensor data, situational awareness, and mission objectives. To address this challenge, the development of an adaptive task-generation module is proposed, which would be fully integrated with the existing tactical planner algorithm. This module would leverage sensor fusion techniques, anomaly detection, and predictive modeling to autonomously identify mission-critical objectives. For instance, if an aerial drone detects a structural collapse in a disaster zone, the system should automatically generate a sequence of tasks, including damage assessment, victim localization, and debris removal, prioritizing them based on urgency and available resources. By bridging the gap between task generation and scheduling, the system has the potential to significantly enhance autonomous decision-making in multi-robot system applications. The integration of machine learning-based predictive algorithms could further refine this process, enabling the system to anticipate operational constraints and adjust planning strategies accordingly. Future research will focus on developing advanced AI-driven methodologies capable of extracting task requirements from multi-sensor data, forecasting mission-critical needs, and dynamically optimizing the planning strategy to ensure maximum efficiency in life-saving operations. Although the primary contribution of this thesis lies in a heuristic/metaheuristic framework, recent advances in learning-based scheduling are worth noting as a complementary research avenue. Deep reinforcement learning (DRL) and meta-learning approaches have been used to synthesize policies that produce near-optimal scheduling decisions online after an offline training phase [75], [76]. These methods are particularly attractive when problem structure is stable enough to allow effective training, and when online inference latency is the critical metric: trained agents can produce allocations in milliseconds even for large instances, whereas exact or even complex heuristic solvers may still be orders of magnitude slower. Nevertheless, learning approaches also face limitations: they can generalize poorly to out-of-distribution scenarios (e.g., drastically different hazard maps or agent failures), and they require a careful design of rewards to balance multiple objectives (safety, makespan, energy). Therefore, we propose treating DRL/meta-learning techniques as an augmentation to the present scheduling pipeline ? for example, DRL policies can provide initial seeds or parameter tuning for faster metaheuristic runs, or can be used to prioritize batches for the ILP pre-check stage. This combined strategy leverages fast inference and provable fallback behaviors, aligning with recent proposals in the literature. The implementation of these enhancements would mark a significant step toward a fully autonomous mission planning system, eliminating the need for human intervention in the early stages of operation and allowing for faster and more adaptive deployment in critical scenarios. Through the seamless integration of real-time task generation, dynamic scheduling, and adaptive resource allocation, the proposed system aims to redefine the role of autonomous robotic teams in disaster response and other mission-critical domains.

Appendix A

Mathematical Foundations of Integer Linear Programming and Optimization Methods

This appendix provides an overview of key mathematical concepts used in Integer Linear Programming (ILP) and optimization, focusing on three essential topics: the Simplex Algorithm, sufficient conditions for a Totally Unimodular (TUM) matrix, and the conditions under which an Integer Linear Programming (ILP) formulation guarantees integer vertices in its polytope representation. These theoretical foundations play a crucial role in ensuring the efficiency and correctness of optimization algorithms, particularly in combinatorial and discrete optimization problems [77], [78].

A.1 The Simplex Algorithm

The Simplex Algorithm [79], is a widely used method for solving Linear Programming (LP) problems. It operates by iteratively improving a feasible solution along the edges of the feasible polytope until an optimal solution is reached. Given an LP problem in the standard form:

$$\min c^T x$$

$$\text{subj. to} \quad Ax \geq b, \quad x \geq 0,$$

where A is an $m \times n$ matrix, x is a vector of decision variables, b is a constraint vector, and c represents the cost coefficients, the Simplex Algorithm proceeds as follows:

- Initialization: Identify an initial basic feasible solution (BFS) at a vertex of the feasible region.
- Pivoting Step: Select an entering variable (one that improves the objective function) and a leaving variable (to maintain feasibility).
- Iterative Improvement: Move along the polytope edges to the adjacent BFS that improves the objective function.
- Termination: Stop when no further improvement is possible, meaning that the optimal vertex has been found.

The Simplex Algorithm is computationally efficient in practice, despite its worst-case exponential time complexity, because real-world problems often exhibit a structure that allows for rapid convergence. However, in ILP problems, where variables must take integer values, the Simplex method alone is insufficient. Instead, techniques such as Branch and Bound or Branch and Cut are used in combination with LP relaxations. It is still possible to use the simplex algorithm to solve ILP if the matrix A is Totally Unimodular A.2: in that case the solution of the simplex algorithm, i.e., the vertex of the polytope, are all integer values A.2.1.

A.2 Sufficient Condition for a Totally Unimodular Matrix

A matrix A is Totally Unimodular (TUM) if every square submatrix has a determinant of 0, +1, or -1. That is, for every $k \times k$ submatrix B of A, we have:

$$det(B) \in \{-1, 0, 1\}$$

This property is crucial in ILP, as it guarantees that the solution to a linear programming relaxation of an integer problem remains integer-valued, eliminating the need for additional combinatorial techniques. A sufficient condition for a matrix A to be totally unimodular is that:

1. Each column of A contains at most two nonzero elements.
2. The rows of A can be partitioned into two disjoint sets I_1 and I_2 such that:
    - If a column contains two nonzero elements of the same sign, the corresponding rows belong to different sets.
    - If a column contains two nonzero elements of opposite signs, the corresponding rows belong to the same set.

If these conditions are met, then A is totally unimodular.

A.2.1 Integer Linear Programming and Conditions for Integer Vertices in a Polytope

In ILP, a fundamental question is whether a linear program's feasible region guarantees integer solutions. This is essential for problems like scheduling, assignment, and routing, where solutions must be discrete. The significance of TUM matrices in ILP is that they ensure that the vertices of the feasible polytope are always integer-valued, allowing the problem to be solved efficiently using the Simplex Algorithm without the need for additional integer programming techniques.

Appendix B

Vehicle Routing Problem

In this chapter, we will introduce the classical VRP and its formal mathematical representation before highlighting its extensions relevant to the multi-robot task scheduling problem addressed in this research. Our scheduling problem extends VRP by integrating heterogeneous agents, precedence constraints, synchronization conditions, and dynamic reallocation, making it more complex than standard VRP formulations.

B.1 Graph Representation

The classical VRP can be represented as a graph optimization problem. Let:

- $G = (N, A)$ be a directed graph, where:
  - $N = \{0, 1, \ldots, n\}$ is the set of nodes, with 0 representing the depot and the remaining n nodes representing customers or task locations.
  - $A = \{(I, j) | (I, j \in N, i \neq j\}$ is the set of arcs representing possible travel paths between nodes.
- k is the set of vehicles, each starting and ending at the depot.
- c_{ij} is the travel cost (e.g., distance or time) associated with traversing arc (I, j).

The objective is to design routes R_k for each vehicle $k \in K$, minimizing a cost function subject to feasibility constraints. A route is formally defined as a sequence of nodes $R_k = (i_1, i_2, \ldots, i_m)$, ensuring:

1. Each customer is visited exactly once:

$$\sum_{k\in K}\sum_{(I,j)\in A} x_{ijk} = 1, \quad \forall i \in N \setminus \{0\} \tag{B.1}$$

   where $x_{ijk} = 1$ if vehicle k travels from node i to node j, and 0 otherwise.

2. Vehicle routes start and end at the depot:

$$\sum_{j\in N} x_{0jk} = 1, \sum_{i\in N} x_{i0k} = 1, \quad \forall k \in K \tag{B.2}$$

3. Capacity constraints (if applicable):

$$\sum_{i\in R_k} d_i \leq Q_k, \quad \forall k \in K \tag{B.3}$$

   where d_i is the demand at node i and Q_k is the vehicle's maximum capacity.

4. Subtour elimination constraints (preventing disconnected Cycles):

$$\sum_{(i,j)\in S} x_{ijk} \leq |S| - 1, S \subseteq N, 2 \leq |S| \leq |N| - 1 \quad \forall k \in K \tag{B.4}$$

The objective function is typically:

$$\min \sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x_{ij}. \tag{B.5}$$

B.2 VRP Variants

Several extensions of VRP align closely with our problem formulation. Below, we analyze key variants and how they relate to our multi-robot scheduling framework.

B.2.1 Heterogeneous VRP (HVRP

Unlike classical VRP, where all vehicles are identical, HVRP considers vehicles with different capabilities, travel speeds, and constraints. In the multi-robot system (MRS) scheduling problem, agents (UGVs, UAVs) have different movement capabilities, sensor configurations, and payload capacities. The resource allocation problem in MRS is analogous to HVRP, requiring vehicles (agents) to be assigned tasks they can feasibly perform.

B.2.2 VRP with Time Windows (VRPTW)

Time windows restrict the allowable visit time for nodes. Task scheduling in dynamic missions often requires meeting deadlines or respecting synchronization constraints (e.g., UAVs providing real-time reconnaissance while UGVs execute ground tasks). The Start-While and End-After constraints in our scheduling problem are there to express in a task-oriented way the same constraints as the Time Windows of a VRPTW.

B.2.3 VRP with Precedence Constraints

Certain tasks in MRS require sequential execution. For example, reconnaissance should precede debris removal in disaster response missions. This introduces precedence constraints, where task T_i cannot begin before T_j is completed. This problem extends classical VRP by integrating temporal dependencies between nodes.

B.2.4 Dynamic VRP (DVRP)

In disaster response missions, the environment is highly dynamic, requiring real-time scheduling and replanning. Unlike classical VRP, where all information is known a priori, DVRP incorporates new tasks, vehicle failures, and evolving constraints. Our proposed approach extends DVRP by integrating on-the-fly path estimation, and continuously updating routes based on new information.

B.3 Complexity and Solution Approaches

Since VRP and its variants are NP-hard problems, exact solutions are impractical for large-scale instances. Common solution methods include:

1. Exact Methods (MIP, Branch-and-Bound)
    - Solves small-scale VRP optimally.

  - Computationally expensive; infeasible for large-scale dynamic problems.

2. Heuristic Approaches (Greedy, Local Search)
    - Provides fast and near-optimal solutions.
    - Applied when real-time performance is needed.
3. Metaheuristic Algorithms (Genetic Algorithms, Tabu Search, Ant Colony Optimization)
    - Explores a wider solution space.
    - Suitable for large-scale problems.
4. Hybrid Approaches
    - Combine heuristics with machine learning or reinforcement learning.
    - Adaptive strategies for dynamic environments.

Appendix C

A^* Algorithm

The A^* algorithm is a widely utilized pathfinding and graph traversal method, renowned for its efficiency in identifying optimal paths within various domains, including robotics, gaming, and network routing. Developed as an extension of Dijkstra's algorithm, A^* introduces heuristic estimations to enhance search efficiency by guiding the exploration process towards the target node. A^* operates by maintaining two primary sets of nodes: the open set, containing nodes pending evaluation, and the closed set, comprising nodes already assessed. The algorithm iteratively selects the node from the open set with the lowest $f(n)$ value, where:

$$f(n) = g(n) + h(n)$$

Here:

- $g(n)$ represents the cost from the start node to the current node n. - $h(n)$ denotes the heuristic estimate of the cost from n to the goal node.

The choice of the heuristic function $h(n)$ is pivotal, as it influences the algorithm's performance and accuracy. A^* is guaranteed to find the shortest path if the heuristic is admissible (never overestimates the true cost) and consistent (satisfies the triangle inequality).

C.1 Comparative Analysis with Dijkstra's Algorithm

While Dijkstra's algorithm explores all possible paths uniformly, A^* enhances efficiency by incorporating heuristic information to prioritize paths that appear more promising [80]. This targeted approach often results in faster convergence to the optimal path, especially in large or complex search spaces. Studies have demonstrated that A^* can outperform Dijkstra's algorithm in terms of computational efficiency, particularly when an effective heuristic is employed.

C.2 Applications and Enhancements

A^* has been extensively applied in various fields requiring efficient pathfinding solutions. In robotics, it facilitates autonomous navigation by enabling robots to plan optimal paths in dynamic environments. In video games, A^* is employed to manage character movements, ensuring realistic and efficient navigation. Moreover, A^* serves as a foundational algorithm in network routing protocols, optimizing data transmission paths. To address specific challenges and improve performance, numerous enhancements to the standard A^* algorithm have been proposed. For instance, the $EBS - A^*$ algorithm [81] introduces expansion distance, bidirectional search, and path smoothing to enhance path planning efficiency and robustness. Additionally, combining A^* with greedy algorithms has been explored to develop multi-objective path planning strategies [82], balancing factors such as path length and computational resources. The A* algorithm remains a cornerstone in pathfinding research and applications, valued for its balance between optimality and computational efficiency. Ongoing research continues to refine and adapt A*, ensuring its relevance and effectiveness in increasingly complex and dynamic problem domains.

Bibliography

[1] IFRC, "Come heat or high water," *World Disaster Report 2020*, 2021.

[2] IFRC, "Urban risk," *World Disaster Report 2010*, 2010.

[3] IFRC, "World disasters report," *World Disasters Report 2015*, 2015.

[4] Z. Xuexi, A. Yuming, F. Genping, L. Guokun, and L. Shiliu, "Survey on key technology of robocup rescue robot," in *2019 Chinese Control Conference*, 2019, pp. 4746–4750.

[5] R. R. Murphy, S. Tadokoro, and A. Kleiner, "Disaster robotics. in: Siciliano, bruno and khatib, oussama," in *Springer Handbook of Robotics.* Cham: Springer International Publishing, 2016, pp. 1577–1604.

[6] I. Kruijff-Korbayova et al., "Deployment of ground and aerial robots in earthquake-struck amatrice in italy (brief report)," *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 278–279, 2016.

[7] O. Fernandes, R. Murphy, J. Adams, and D. Merrick, "Quantitative data analysis: Crasar small unmanned aerial systems at hurricane harvey," *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6, 2018.

[8] J. Adams, L. Hart, J. McBride, D. Merrick, and R. Murphy, "Use of small unmanned aerial systems for tactical response during kilauea volcano lower east rift zone event," *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–2, 2018.

[9] J. Whitman, N. Zevallos, M. Travers, and H. Choset, "Snake robot urban search after the 2017 mexico city earthquake," *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6, 2018.

[10] R. R. Murphy et al., “Use of a small unmanned aerial system for the sr-530 mudslide incident near oso, washington,” *Journal of field Robotics*, vol. 33, no. 4, pp. 476–488, 2016.

[11] D. S. Drew, “Multi-agent systems for search and rescue applications,” *Current Robotics Reports*, vol. 2, pp. 189–200, 2021.

[12] M. B. Dias, M. Zinck, R. Zlot, and A. T. Stentz, “Robust multirobot coordination in dynamic environments,” *Proceeding of the 2004 IEEE International Conference on Robotics & Automation*, 2004.

[13] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot task allocation: A review of the state-of-the-art,” *Cooperative Robots and Sensor Networks*, 2015.

[14] B. P. Gerkey and M. J. Mataric, “A formal analysis and taxonomy of task allocation in multi-robot systems,” *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004. DOI: `10.1177/0278364904045564`

[15] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006. DOI: `10.1109/JPROC.2006.876939`

[16] Y. Zhang and L. E. Parker, “Multi-robot task scheduling for executable coalitions in multirobot teams,” *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1151–1163, 2013. DOI: `10.1177/0278364913483188`

[17] G. A. Korsah, A. Stentz, and M. B. Dias, “A comprehensive taxonomy for multi-robot task allocation,” *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013. DOI: `10.1177/0278364913496484`

[18] A. K A, D. U. J, and U. Subramaniam, “A systematic literature review on multi-robot task allocation,” *ACM Comput. Surv.*, vol. 57, no. 3, Nov. 2024, ISSN: 0360-0300.

[19] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, “Multiple mobile robot task and motion planning: A survey,” *ACM Comput. Surv.*, vol. 55, no. 10, Feb. 2023.

[20] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, "Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art," *Robot. Auton. Syst.*, vol. 168, no. C, Oct. 2023.

[21] N. Kalra, D. Ferguson, and A. Stentz, "Incremental replanning for multi-robot systems," pp. 3401–3406, 2005. DOI: `10.1109/ROBOT.2005.1570634`

[22] R. Calvo and J. Capitán, "Long-endurance heterogeneous multi-robot task allocation with resilience and plan repair," *Autonomous Robots*, 2024. DOI: `10.1007/s10514-024-10123-9`

[23] Y. Pan, C. Wang, and L. Zhang, "Robust decentralized navigation under dynamic uncertainty," 2024.

[24] X. Yang, R. Chen, and H. Zhao, "Hybrid task allocation and coalition coordination for uav sar missions using nsga-iii," *The Journal of Supercomputing*, vol. 81, pp. 11 234–11 259, 2025.

[25] Y. Dai, Q. Wu, and W. Zhou, "Multi-phase particle swarm optimization for dynamic multi-robot task aggregation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2025, early access.

[26] J. David and F. Valencia, "Simultaneous path planning and task allocation in dynamic environments (spades)," *Robotics*, vol. 14, no. 2, p. 17, 2025. DOI: `10.3390/robotics14020017`

[27] X. Qin, Z. Liao, C. Liu, and Z. Xiong, "Online task allocation and scheduling in multi-manipulator system considering collision constraints and unknown tasks," *Robot. Comput.-Integr. Manuf.*, vol. 90, no. C, Dec. 2024.

[28] A. Verma, A. Gautam, V. S. Shekhawat, and S. Mohan, "Dta-hmr-tt: Dynamic task allocation for a heterogeneous team of mobile robots with task transfer," *IEEE Access*, 2024.

[29] D. Budaev, K. Amelin, G. Voschuk, P. Skobelev, and N. Amelina, "Real-time task scheduling for multi-agent control system of uav′s group based on network-centric technology," *Control, Decision and Information Technologies*, Apr. 2016.

[30] A. da Fonseca Braga, P. D. M. Plentz, and E. R. D. Pieri, "Development of a generic framework for multi-robot systems," *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020.

[31] T. Sakamoto, S. Bonardi, and T. Kupbota, "A routing framework for heterogeneous multi-robot teams in exploration tasks," *IEEE Robotics and Automation Letters*, vol. 5, 2020.

[32] M. Badreldin, A. Hussein, and A. Khamis, "A comparative study between optimization and market-based approaches to multi-robot task allocation," *Advances in Artificial Intelligence*, 2013.

[33] A. R. Mosteo and L. Montano, "Simulated annealing for multi-robot hierarchical task allocation with minmax objective," *Workshop on Network Robot Systems: Toward intelligent robotic system integrated with environments, IROS 2006*, 2006.

[34] A. M. Khamis, A. M. Elmogy, and F. O. Karray, "Complex task allocation in mobile surveillance systems," *Journal of Intelligent & Robotic Systems*, 2011.

[35] G. Galati, S. Primatesta, and A. Rizzo, "Auction-based task allocation and motion planning for multi-robot systems with human supervision," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 2, p. 24, 2023.

[36] H. Osooli, P. Robinette, K. Jerath, and S. R. Ahmadzadeh, *A multi-robot task assignment framework for search and rescue with heterogeneous teams*, 2023. arXiv: `2309.12589 [cs.RO]`.

[37] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[38] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1994, pp. 3310–3317. DOI: `10.1109/ROBOT.1994.351061`

[39] S. M. LaValle, *Planning Algorithms.* Cambridge University Press, 2006, ISBN: 978-0-521-84440-4.

[40] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, “Conflict-based search for optimal multi-agent path finding,” *Artificial Intelligence*, vol. 219, pp. 40–66, 2015. DOI: `10.1016/j.artint.2014.11.006`

[41] D. Silver, “Cooperative pathfinding,” in *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, 2005, pp. 117–122.

[42] K. Daniel, A. Nash, S. Koenig, and A. Felner, “Theta*: Any-angle path planning on grids,” *Journal of Artificial Intelligence Research (JAIR)*, vol. 39, pp. 533–579, 2010. DOI: `10.1613/jair.2994`

[43] T. Xu et al., “Recent advances in rapidly-exploring random tree: A review,” *Review / survey (2024)*, 2024, survey of RRT variants (recommended instead of an unverified 'Zhou2021').

[44] G. Heppner, “Behavior tree capabilities for dynamic multi-robot task allocation,” *arXiv preprint arXiv:2402.02833*, 2024, ros_bt_py framework; capability-aware BT allocation (arXiv).

[45] C. Hustiu, M. Nguyen, and J. De Schutter, “Nets-within-nets for multi-robot mission and motion planning,” 2023.

[46] M. Lanzarini and L. Marconi, “Ai-enabled disaster response planning forâ multi-robot andâ autonomous systems viaâ task scheduling andâ path-finding,” C. Secchi and L. Marconi, Eds., pp. 258–262, 2024.

[47] D. Alexander, “Principles of emergency planning and management,” *Oxford University Press google schola*, vol. 2, pp. 229–238, 2002.

[48] A. Gautam and S. Mohan, “A review of research in multi-robot systems,” *2012 IEEE 7th International Conference on Industrial and Information Systems, ICIIS 2012*, Aug. 2012.

[49] J. J. Roldán-Gómez and A. Barrientos, “Special issue on multi-robot systems: Challenges, trends, and applications,” *Applied Sciences*, vol. 11, no. 24, p. 11 861, 2021.

[50] S. Rajapakshe et al., *Collaborative ground-aerial multi-robot system for disaster response missions with a low-cost drone add-on for off-the-shelf drones*, 2023. arXiv: `2304.06992 [cs.RO]`.

[51] E. D′Amato, *Advanced solutions and challenges in multi-drone systems for search and rescue missions: Technical approaches and unresolved matter.*

[52] R. R. Murphy, *Disaster robotics*. MIT press, 2017.

[53] S. Rajapakshe et al., *Collaborative ground-aerial multi-robot system for disaster response missions with a low-cost drone add-on for off-the-shelf drones*, 2023. arXiv: `2304.06992 [cs.RO]`.

[54] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, 2003.

[55] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European Journal of Operational Research*, vol. 191, no. 1, pp. 19–31, 2008.

[56] P. Ghassemi and S. Chowdhury, "Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints," *Robotics and Autonomous Systems*, vol. 147, p. 103 905, Oct. 2021.

[57] J. Park, A. Messing, H. Ravichandar, and S. Hutchinson, "Risk-tolerant task allocation and scheduling in heterogeneous multi-robot teams," Oct. 2023, pp. 5372–5379.

[58] S. Sloan, "A fast algorithm for constructing delaunay triangulations in the plane," *Advances in Engineering Software*, vol. 9, no. 1, pp. 34–55, 1978.

[59] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of a* pathfinding," *Procedia Computer Science*, vol. 179, pp. 507–514, 2021, 5th International Conference on Computer Science and Computational Intelligence 2020, ISSN: 1877-0509.

[60] A. Aswale and C. Pinciroli, *Heterogeneous coalition formation and scheduling with multi-skilled robots*, 2023. arXiv: `2306.11936 [cs.RO]`.

[61] D. Huh and P. Mohapatra, *Multi-agent reinforcement learning: A comprehensive survey*, 2024. arXiv: `2312.10256 [cs.MA]`.

[62] W. Dai, U. Rai, J. Chiun, Y. Cao, and G. Sartoretti, "Heterogeneous multi-robot task allocation and scheduling via reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2654–2661, 2025.

[63] M. Yang, B. Zhang, Z. Shi, and J. Li, "A hybrid task allocation approach for multi-uav systems with complex constraints: A market-based bidding strategy and improved nsga-iii optimization," *The Journal of Supercomputing*, vol. 81, Feb. 2025.

[64] S.-H. Dai, Y.-H. Jia, W.-n. Chen, Y. Mei, and Q. Yang, "Multistage particle swarm optimization for heterogeneous multipoint dynamic aggregation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, pp. 1–15, Jan. 2025.

[65] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction.* CRC Press, 2018, ISBN: 9781138593733.

[66] Y. Sung, J. Kim, and J. Lee, "Asynchronous taskâmotion plan refinement for multi-robot systems," 2024.

[67] Z. Cai, L. Wang, and Y. Li, "Mrbtp: Multi-robot behavior tree planning with guarantees and llm acceleration," *Autonomous Robots*, 2025, in press.

[68] Y. Cai et al., *Mrbtp: Efficient multi-robot behavior tree planning and collaboration*, 2025. arXiv: `2502.18072 [cs.RO]`.

[69] A. Gautier, A. Stephens, B. Lacerda, N. Hawes, and M. Wooldridge, "Negotiated path planning for non-cooperative multi-robot systems," pp. 472–480, 2022.

[70] A. Prasad, K. Chaudhary, B. Sharma, and A. S. Ali, "Robot path planning and motion control: A systematic review," *Engineered Science*, vol. 33, p. 1354, 2025.

[71] S. Debnath et al., "Different cell decomposition path planning methods for unmanned air vehicles-a review," in Jan. 2021, pp. 99–111.

[72] K. Katona, H. A. Neamah, and P. Korondi, "Obstacle avoidance and path planning methods for autonomous navigation of mobile robot," *Sensors*, vol. 24, no. 11, 2024.

[73] G. Xu et al., "Multi-robot task allocation and path planning with maximum range constraints," Sep. 2024.

[74] J. David and R. Valencia, "Simultaneous path planning and task allocation in dynamic environments," *Robotics*, vol. 14, no. 2, 2025.

[75] J. Zhang, Q. Jia, S. Zhang, and G. Chen, "Dynamic and prioritized task scheduling of heterogeneous multi-robot systems using deep reinforcement learning," *Neurocomputing*, vol. 638, p. 130 184, 2025, ISSN: 0925-2312.

[76] P. Song, H. Chen, K. Cui, J. Wang, and D. Shi, "Meta-learning for dynamic multi-robot task scheduling," Jan. 2024.

[77] A Schrijver, *Theory of linear and integer programming.* John Wiley & Sons, 1998.

[78] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization.* John Wiley & Sons, 1999.

[79] G. B. Dantzig, "Maximization of a linear function of variables subject to linear inequalities," *Activity analysis of production and allocation*, vol. 13, pp. 339–347, 1951.

[80] Y. Yan, "Research on the a star algorithm for finding shortest path," *Highlights in Science, Engineering and Technology*, vol. 46, pp. 154–161, Apr. 2023.

[81] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu, and T. Liu, "The ebs-a* algorithm: An improved a* algorithm for path planning," *PLoS One*, 2022.

[82] D. Xiang, H. Lin, J. Ouyang, and D. Huang, "Combined improved a* and greedy algorithm for path planning of multi-objective mobile robot," *Scientific Reports*, 2022.