Design and Implementation of Coordination Algorithms for Multi-Robot System Under Constrained Communication Range

THESIS

Submitted in the partial fulfilment of the requirements of

DOCTOR OF PHILOSOPHY

by

ANKIT SONI

(2018PHXF0016P)

Under the Supervision of

Dr. AVINASH GAUTAM

and

Under the Co-supervision of

Prof. VIRENDRA SINGH SHEKHAWAT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI- 333031 (RAJASTHAN) INDIA

2024

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
PILANI CAMPUS, RAJASTHAN, INDIA

CERTIFICATE

This is to certify that the thesis entitled, **"Design and Implementation of Coordination Algorithms for Multi-Robot System Under Constrained Communication Range"** and submitted by **Mr. Ankit Soni**, ID No. **2018PHXF0016P** for the award of Ph.D. of the Institute embodies original work done by him under our supervision.

Signature of the Supervisor
Dr. Avinash Gautam
Assistant Professor
Department of CSIS, BITS Pilani,
Pilani Campus, India, 333031

Signature of the Co-Supervisor
Prof. Virendra Singh Shekhawat
Associate Professor
Department of CSIS, BITS Pilani,
Pilani Campus, India, 333031

Date: $01/May/2024$

This thesis is dedicated

To

My beloved

Parents and other family members

ACKNOWLEDGEMENTS

First and foremost, I wish to thank my thesis supervisor, Dr. Avinash Gautam, and co-supervisor, Dr. Virendra Singh Shekhawat, for their guidance and support throughout my Ph.D. work. Their continued support and guidance at each step helped me to complete my thesis work.

I have been blessed to have a group of friendly and cheerful fellow research scholars and undergraduate students who have offered emotional and academic support during this work. Special thanks to my friends, Adapa Dharmateja Kaliprasad Hemasri, Nivedan Yakolli, Ashish Verma, Dr. Geetika Arora, Dr. Upendra Singh, Chirag Dasannachary, and Tanishq Duhan.

I express my deepest gratitude to Prof. J P Misra and Prof. Sudeept Mohan, who continuously provided me with critical inputs during my Ph.D. work. Many thanks to Prof. Bijay K Rout of the Department of Mechanical Engineering for willingly sharing his knowledge and experience in the field of robotics with me.

I am also indebted to Prof. Mukesh Kumar Rohil, Prof. Poonam Goyal, and Dr. Vandana Agarwal for their constructive comments during my end-semester research presentations.

I would also like to thank Professor Navneet Goyal, Head of the Department of Computer Science and Information Systems, who helped us at different points and made this research possible. Thanks to Prof. V. Ramgopal Rao, Vice-Chancellor, and Prof. Sudhir Kumar Barai, Director, BITS Pilani (Pilani campus), for their constant support and concern. I gratefully acknowledge Prof. Navin Singh, Associate Dean, SWD, for the help.

I would also like to thank Dr. Yashvardhan Sharma and Dr. Amit Dua from the Department of Computer Science and Information Systems, Dr. Rajdeep Chowdhary (HoD of the Department of Biological Sciences), and Dr. Syed Mohammad Zafaruddin of the Department of Electrical and Electronics Engineering who encouraged me to play cricket and badminton for physical and mental fitness.

I cannot conclude this acknowledgment without mentioning my parents. It would not have been possible to complete this thesis without their love, encouragement, support, and blessings.

Place: BITS Pilani, Pilani Campus, India
Date: $01/May/2024$ **(Ankit Soni)**

ABSTRACT

When multiple mobile robots work cooperatively, one is able to accomplish a task that is difficult for a single robot to accomplish. Creating solutions involving multiple robots has a number of advantages. Some of the advantages are as follows:

- Tasks like unknown area exploration and terrain coverage have high complexity; therefore, it is not feasible to employ a single robot. Instead, a well-coordinated team of robots can perform the same task more efficiently.
- The tasks, like sweeping, lawn mowing, harvesting crops, patrolling, and monitoring war fields, are boring, repetitive, and potentially dangerous. Additionally, these tasks demand redundancy in terms of robots as they are required to be completed quickly.
- Multi-robot systems generally offer high resilience and fault tolerance.

Unknown area exploration and terrain coverage are two fundamental applications in multi-robot systems. However, there is a thin line of separation between these two applications. Unknown area exploration requires the robots to build a map of the environment without visiting/ traversing the entire navigable region. On the other hand, the terrain coverage application entails the entire free space to be physically traversed by at least one robot. In order to accomplish any task using a team of autonomous mobile robots, inter-robot communication is vital for coordination, information dissemination, and resource sharing. Many state-of-the-art approaches work with the premise that communication is omnipresent. This particular assumption may not be feasible in real-world scenarios. In situations where communication is restricted within a certain range, achieving coordination and expecting the robot team to have cohesive behaviour is non-trivial. In this thesis, we have designed and implemented coordination algorithms for multi-robot systems to solve the identified problem. We have shown that a well-coordinated group of robots is able to improve performance by making better use of the system's resources, specif-

ically the robots themselves, even when there is only sporadic communication between the robots.

Chapter 3 of this thesis began with a study and simulation of five cutting-edge approaches to online terrain coverage. These are given below -

1. Backtracking Spiral Approach – Cooperative Multi-Robot (BSA-CM) [1]
2. Spiraling and Selective Backtracking (SSB) [2]
3. Boustrophedon and Backtracking mechanism (BoB) [3]
4. Multiple Depth First Search (MDFS) [4]
5. Brick and Mortar (BnM) [4]

We chose these algorithms because they are designed for scenarios with uninterrupted communication without any range restrictions. However, we dropped the assumption of global communication and conducted experiments where we varied both the communication range and the number of robots. This investigation aimed to shed light on how the performance of these algorithms could be affected when communication is limited. MDFS and BnM have incomplete coverage for communication ranges less than 12 meters and are excluded from the comparison. For communication ranges up to 4 meters, BSA CM and SSB approaches have a redundant coverage exceeding BoB by 4% and 11%, respectively. However, for larger communication ranges, SSB and BSA CM outperform BoB. For a team of 4 robots, SSB and BSA-CM outperform BoB by 14.8% and 9.60% in redundant coverage. For a team of 6 robots, SSB and BSA CM perform even better at 20.8% and 18.93% respectively. These findings proved instrumental in the development of more efficient multi-robot exploration algorithms for unknown terrain, which are presented in Chapter 4 and Chapter 5 of this thesis.

We have suggested algorithms (both centralized and distributed) for exploring uncharted areas using a team of mobile robots in an environment with limited communication. This thesis introduces an innovative strategy for simultaneous robot exploration of an uncharted region, viz., Multi-Robot Unknown Area Exploration Using Frontier Trees (MRFTE). The frontier tree data structure used in single-robot exploration stores information on the frontiers, their locations, the explorer's current status, and the map itself. Inquiries into this tree could be used to plan future investigations. MRFTE extends this idea for multi-robot exploration by introducing a new abstraction that is a *group*, which is designed to share data via a shared frontier tree, group-level operations, and a means of assigning goals to numerous robots. A group of robots is a collection of machines whose collective map of the world covers a continuous area. During the exploration assignment, the robots are in sync with one another because each group has only one tree. Once groups' maps intersect, we provide methods for merging their frontier trees. We conclude by recommending a strategy for designating and allocating exploration goals to individual robots via the selection of nodes from the frontier tree. Simulation of MRFTE outperforms seven state-of-the-art methods.

1. Nearest Frontier(NF) [5]
2. Information Gain Based Heuristic (D+IG) [6]
3. Cost+Utility (C+U) [7]
4. Voronoi Graph-Based Segmentation (VGS) [8]
5. Multiple Rapidly Exploring Random Trees (M-RRT) [9]
6. Information-Driven RRT (ID-RRT) [10]
7. Goal Assignment Using Distance Cost (GADC) [11]

During our simulation on a cluttered map with 8 robots, we discovered that MRFTE takes less time than other algorithms. Specifically, it takes 62.16%, 53.33%, 41.66%, 28.81%, 56.25%, 58.82%, and 46.15% less time than NF, D+IG, C+U, VGS, M-RRT, ID-RRT, and GADC respectively. Additionally, we found that MRFTE travels less distance than other algorithms. Specifically, it travels 39.47%, 32.35%, 28.12%, 4.16%, 17.85%, 28.12%, and 23.33% less distance than NF, D+IG, C+U, VGS, M-RRT, ID-RRT, and GADC respectively. However, this approach relies on the assumption that all robots can communicate with each other without any restrictions. To get over this limitation, we developed a decentralized method for multi-robot systems and were able to get results on par with those of the centralized method.

To conclude this thesis, we designed a novel Decentralized Relay-Based Approach for Multi-Robot Unknown Area Exploration (D-MRFTE) for exploring uncharted territory with a fleet of mobile robots with limited communications. Meetups assure eventual coherence and completeness of the scattered copies of exploration data used by the high-latency multi-robot system's, decentralized network formed by the relay robots. Whenever the multi-robot network gets defragmented, the periodic meetups ensure that data is transferred at regular intervals to restore network stability. Relay robots provide for easier communication between units about the status of the robots and their current exploration. To maintain consistency of the distributed information in the robot team, The robots use timestamps and version vectors. To maintain a steady stream of explorer robots, the relays organize get-togethers with the other relays they come into contact with. This method outperformed two state-of-the-art algorithms in terms of both task completion time and robot travel distance when using either the Disk-based or Line-of-Sight communication models. After analyzing the impact of different approaches on two metrics (cumulative distance travelled and frequency of disconnections), we discovered that D-MRFTE+0R (a multi-explorer system without relay assistance for information exchange) performed the poorest compared to all other approaches. The situation improved with the introduction of one relay, but frequent disconnections were still observed. However, this approach performed less favourably than all others on both metrics. Despite this, it helped to reduce redundancy. Interestingly, when two relays were introduced (D-MRFTE+2R), the performance improved significantly. With six and eight explorer robots, it surpassed VGS by 13.4% and 14.2%, respectively. However, it fell short of MRFTE for Disk-based communication. For Line-of-sight communication, it surpassed VGS by 9.09% with eight explorer robots. The cumulative distance travelled yielded a similar conclusion.

Keywords: Multi-Robot Systems, Online Terrain Coverage, Unknown Area Exploration, Frontier Exploration, Inter-Robot Communication, Constrained Communication, Player/ Stage and ROS-based simulation

Table of Contents

LIST OF ABBREVIATIONS

2D: Two-Dimension
3D: Three-Dimension
AGV: Automated Guided Vehicle
AMR: Autonomous Mobile Robot
AMRs: Autonomous Mobile Robots
AutoML: Autonomous Machine Learning Algorithms
AVR: Automatic voltage regulator
BFS: Breadth-First Search
BnM: Brick and Mortar
BoB: Boustrophedon and Backtracking mechanism
BP: Backtracking Point
BSA-CM: Backtracking Spiral Approach – Cooperative Multi-Robot
C+U: Cost Utility Based Heuristic
CDT: Cumulative Distance Travelled
CS: Control Station
D-MRFTE : Decentralized MRFTE
D-MRFTE : Decentralized MRFTE
D-MRFTE+1R : Decentralized MRFTE with one relay
D-MRFTE+2R : Decentralized MRFTE with two relays
DFS: Depth-First Search
DHBA: Decentralized Hungarian Based Algorithm
DS: Dispersion Score
ECT: Exploration Completion Time
ED: Euclidean Distance
FB-V: Firebird 5

FB-VI:	Firebird 6
FFD:	Fast Frontier Detector
FOV:	Field of View
GADC:	Goal Assignment Using Distance Cost
GVG:	Generalized Voronoi Graph
IA:	Instantaneous Assignment
ID-RRT:	Information Driven RRT
IG:	information Gain Based Heuristic
IoRT:	Internet of Robotic Things
IR:	Infrared
IROS:	International Conference on Intelligent Robots and Systems
JPS:	Jump Point Search
LIDARs:	Light Detection and Ranging
LoS:	Line of Sight
LOSC:	Line-of-sight based Communication
M-RRT:	Multiple Rapidly Exploration Random Trees
MARS:	Multi-Agent Robotic Systems
MDFS:	Multiple Depth First Search
MR-CPP:	Multi-Robot Coverage Path Planning
MRFTE :	Multi-Robot Unknown Area Exploration Using Frontier Trees
MROTC:	Multi-Robot Online Terrain Coverage
MRS:	Multi-Robot System
MRT:	Multi-Robot Task
MRTA:	Multi-Robot Task Allocation
MST:	Minimum Spanning Tree
MT:	Multi Task
NF:	Nearest Frontier Heuristic
OLS:	Opposite Lateral Side

OTC :	Online Terrain Coverage
PID:	Proportional-Integral-Derivative
PropEM-L:	Propagation Environment Modelling and Learning
RDA:	Rolling Dispersion Algorithm
RF:	Radio Frequency
RLS:	Reference Lateral Side
ROS2:	Robot Operating system version-2
ROS:	Robot Operating system
RRS:	Received Signal Strength
RRT:	Rapidly Exploration Random Trees
rviz:	ROS visualization
SEA:	Sweep Exploration Algorithm
SLAM:	Simultaneous Localization and Mapping
SoTA:	State-of-the-Art
SRS:	Singe Robot System
SSB:	Spiraling and Selective Backtracking
ST:	Single Task
TA:	Time-extended Assignment
TEA:	Train Exploration Algorithm
TM-RRT:	Temporal Memory-based Exploration Random Trees
US:	Ultrasound
VG:	Voronoi Graph
VGS:	Voronoi Graph-Based Segmentation
WFD:	Wave Front Detector
WSN:	Wireless Sensor Network

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 MULTI-ROBOT SYSTEMS (MRS)

Multi-robot systems are a paradigm shift in the field of robotics, offering a wide range of advantages across industries and applications. These systems provide a number of benefits, including increased efficiency through parallelization, flexibility, adaptability, redundancy, collaboration, and cost-effectiveness. The potential for multi-robot systems to revolutionize industries and redefine the boundaries of automation remains compelling and promising as technology continues to advance. Multi-robot systems hold the key to unlocking new levels of productivity and innovation, whether it's transforming manufacturing processes, revolutionizing disaster response, or re-imagining urban infrastructure. Some of the benefits of MRS are as follows:

(i) **Enhanced Efficiency through Parallelization**

Parallelization capability is one of the most important benefits of multi-robot systems. In conventional single-robot configurations, tasks are performed sequentially, which frequently results in bottlenecks and slower overall execution. Multi-robot systems, on the other hand, permit simultaneous execution of tasks, thereby reducing the time necessary to complete complex operations. This parallelization is especially evident in manufacturing [15], where multiple robots can collaborate on different production stages, optimizing efficiency and accelerating manufacturing. Moreover, multi-robot systems are ideally suited for tasks requiring continuous operation, including environment monitoring [16], unknown area exploration [17], and surveillance [18]. In scenarios such as perimeter security [19], a team of robots can maintain constant vigilance by patrolling in shifts, thereby reducing the likelihood of blind spots and enhancing the efficacy of surveillance efforts.

(ii) **Flexibility and Adaptability**

Multi-robot systems excel in environments requiring adaptability and flexibility. Unlike single robots, which may struggle to adapt to dynamic and unpredictable conditions, a team of robots can more easily adapt to environmental changes. This adaptability is crucial in industries such as agriculture, where robots equipped with a variety of sensors can navigate difficult terrain [20], evaluate crop health [21], and perform more precise targeted interventions. Multi-robot systems excel in disaster response scenarios due to their versatile capabilities. These systems may include robots specializing in various tasks, including search and rescue [22, 23], hazard detection [24], and communication [25]. These robots can effectively address the complex challenges presented by natural disasters and other emergencies by cooperating.

(iii) **Redundancy and Fault Tolerance**

Multi-robot systems have inherent redundancy and fault tolerance. Multiple robots capable of performing the same task serve as a safety net in situations where failure is not an option. If one robot encounters a malfunction or an obstacle, the operation can continue with the assistance of other robots. This feature is especially advantageous for tasks involving hazardous or remote environments, such as space exploration [26] or deep-sea exploration [27]. Moreover, the distributed nature of multi-robot systems mitigates the effect of individual robot failures [28]. When a single robot becomes incapacitated, the system's overall performance is largely unaffected, ensuring that the mission or operation can continue with minimal interruption.

(iv) **Collaboration and Collective Intelligence**

The capacity of multi-robot systems to exhibit collective intelligence through cooperation is one of their most intriguing characteristics. When robots communicate and share information, they can optimize their actions collectively to achieve the desired outcome. This emergent behaviour is observable in swarm robotics, in which a large number of simple robots collaborate to accomplish complex tasks. Applications of swarm robotics range from environmental monitoring to construction. For instance, a swarm of drones can rapidly map and assess disaster-stricken areas, providing disaster response teams with vital information [29,30]. Combining the digital and physical realms, robots can assemble complex structures with high precision in the construction industry.

(v) **Cost Efficiency**

While setting up a multi-robot system may require a larger initial investment than purchasing a single robot. However, long-term cost savings are substantial. Multi-robot systems can cover larger areas and perform tasks more effectively, resulting in an increase in productivity and a decrease in operational expenses over time. Multiple robots can optimize inventory management, order fulfillment, and distribution in industries such as logistics and warehousing [31, 32], where they can work in concert to optimize inventory management, order fulfillment, and distribution. Moreover, multi-robot systems can be reconfigured and repurposed for various tasks, making them more adaptable investments. Businesses can maximize the utility of their robotic assets by reprogramming or retraining existing robots rather than purchasing brand-new robots for specific applications.

1.2 APPLICATIONS OF MRS

As discussed earlier, some of the benefits of multi-robot systems over single-robot systems are increased efficiency, scalability, fault tolerance, and adaptability. As technology advances and robots become more capable, the use of multi-robot systems and their applications continues to expand. Some of the application areas where MRS are useful are discussed below:

(i) **Automation in Industry and Manufacturing**

Multi-robot systems are utilized extensively in industrial automation and manufacturing. By executing tasks in parallel and performing repetitive actions with high precision, these systems have the capacity to improve production processes significantly. Multi-robot systems can assemble products, weld components, and conduct quality control checks in manufacturing facilities [33]. They can work in unison, decreasing production time and increasing output overall.

(ii) **Search and Rescue**

Multi-robot systems have demonstrated enormous potential for use in search and rescue missions, especially in hazardous or disaster-stricken environments where human participation is hazardous [22]. Robots equipped with sensors and cameras can navigate through debris, rubble, and hazardous materials to locate survivors. By collaborating, these robots can cover larger areas in less time, increasing the likelihood of locating individuals who have become trapped. In addition, MRS can share information wirelessly and optimize their search patterns based on real-time data, resulting in more effective and coordinated efforts [23].

(iii) **Precision Agriculture**

Multi-robot systems are transforming traditional farming practices in the agricultural sector. These systems can perform tasks such as plantation, irrigation delivery of nutrients, fertilizers, and harvesting. With sensors and GPS technology, farmers can collect data on soil quality, moisture levels, and crop health, allowing them to make informed decisions regarding resource allocation. By collaborating, these robots can efficiently cover vast fields, resulting in optimized resource utilization, increased yields, and decreased environmental impact [34, 35].

(iv) **Unknown Environment Exploration**

Additionally, multi-robot systems pave the way for the efficient exploration of unknown environments. Clusters of robots can be deployed for exploration of buildings that are earthquake-hit but visually appear intact [36]. The robots can collect data, map terrain, and evaluate damage. MRS also benefits underwater exploration, as robots equipped with underwater sensors can navigate complex underwater environments and conduct surveys for scientific research, environmental monitoring, and resource exploration [37, 38].

(v) **Logistics and Storage**

Modern logistics and e-commerce rely heavily on the timely and accurate movement of goods. Multi-robot systems offer a means to streamline warehouse and distribution center operations [31, 39]. Collaborating robots with picking arms can locate and transport items from shelves to packaging stations. These robots can optimize their routes to minimize congestion and maximize efficiency thanks to sophisticated coordination algorithms. This application expedites order processing and reduces the likelihood of human error.

As technology advances, the applications for multi-robot systems continue to diversify and expand. From industrial automation to search-and-rescue missions, precision agriculture to medical advancements, these systems improve efficiency, precision, and adaptability in a variety of domains. As researchers continue to develop more complex coordination algorithms, improved communication protocols, and advanced sensor technologies, the potential of multi-robot systems to address complex problems and revolutionize industries remains promising. Future opportunities for innovation and integration of these systems into our daily lives are promising.

1.3 MRS Definitions And Taxonomies

The literature on MRS cooperation is substantial, and various definitions have been presented. Some of the definitions are as follows:

(a) The first definition is due to [40], which says that cooperation is *"joining together for doing something that creates a progressive result such as increasing performance or saving time."* This definition aims to make optimal use of the available system resources. This definition drives the vast majority of multi-robot task allocation(MRTA) research.

(b) The second definition is due to [41], which states that cooperation is *"joint collaborative behavior that is directed towards some goal in which there is a common interest or reward."* This quantitative definition is suggested with profit maximization in mind, which means minimizing the use of system resources. Furthermore, it leads to evaluation of performance in terms of the earliest possible time to complete a given task.

(c) The third definition given in [42] contrasts with the first two by describing cooperation as a form of interaction that is typically based on communication. This definition encompasses inter-robot communication for synchronization of actions and information exchange related to the robot's state, for example, the robot's pose and its beliefs about its workspace.

In a multi-robot system, one of the vital software components is the algorithm that breaks down a complex problem T into a set of simple sub-tasks representatives as $T_i = \{t_1, t_2, ..., t_n\}$ that can be handled by a single robot. These simple tasks are then strategically apportioned to the individual robots. Individual robots then simultaneously perform/execute the assigned tasks. Most of the time, robots perform independent tasks; however, whenever two or more activities overlap, robots must find a way to negotiate. The task *T's* completion is the result of recombining the outcome of the execution of all the tasks T_i. The system's entire behaviour, including termination of task *T*, is governed by the recombination of $T_i' s$.

It is important to properly evaluate individual and collective performance in unknown area exploration using multi-robot systems. Task completion time, distance travelled by the robot team, the computational complexity of the algorithm, and redundant exploration are key metrics that can be used to evaluate the performance of the system. Task definition, system group architecture (whether centralized, weakly centralized, or distributed), team composition (whether the multi-robot team is homogeneous or heterogeneous), and communication structure (a given robot's ability to recognize and model the intentions, beliefs, actions, and capabilities of other robots) all influence the performance of the robot team. Coherent and cooperative behavior among the robots is crucial to the existence of a superior solution. In MRS, robots are able to accomplish this mission by exchanging information with one another. Explicit communication occurs when robots convey messages to one another, while implicit communication occurs when robots detect environmental cues and locate one another. The complexity of designing and deploying multi-robot solutions stems from the wide variety of factors that must be considered simultaneously. The major features of MRS, such as group organization, communication structure, control, group composition, learning, and conflict resolution methods, have been addressed by researchers who have produced solutions in an integrated fashion. Multi-robot systems due to their interconnected nature defy simple categorization based on a single set of features. For example, in [12], we see a classification given by Dudek et al., which is (shown in Figure 1.1) based on factors including team size, communication structure, self-organizing team capability, computing capability, and group makeup.

Farinelli et al., [13] presents a taxonomy of coordinated multi-robot systems as shown in Figure 1.2. The authors have defined four distinct stages: collaboration, information, coordination, and management. Cooperative methods are differentiated from non-cooperative methods at the highest level. Depending on their level of knowledge, robots may or may not be aware of the presence of other robots in their surroundings. Strongly coordinated, weakly coordinated, and uncoordinated multi-robot systems exist on the

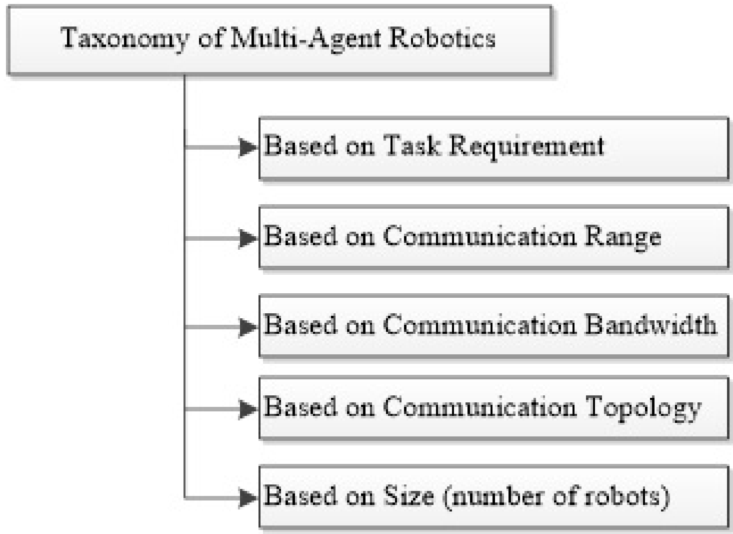


Figure 1.1: MRS Taxonomy Based on Available System Resources [12]

spectrum of coordination. The robots in a system with high levels of coordination, work together more closely and make decisions through communication. The efficiency with which the job is done is undoubtedly enhanced by sharing of information. The robots in a weakly coordinated system are less invested and use ad-hoc group communication to keep the noise level down. When all the robots do a set of specified activities in response to varying environmental stimuli, intelligent behavior will likely occur if the robots are self-aware and not coordinated. Individual robots' freedom to choose their own behaviour is a major focus at the organizational level. An elected leader is typically in charge of and accountable for all robots in a centralized organization. In a distributed organization, on the other hand, each robot is independent and makes its own decisions. Further refinement of this classification has led to the proposal of a weakly centralized organization in which leaders are eligible to be re-elected in the event of the death or voluntary resignation of the already elected leader. Each robot also has some degree of independence from the others.

In [14], Gerkey et al. present a categorization scheme that uses MRTA and coordination methods. This taxonomy of MRTA problems is not specific to any specific MRTA application. What the authors mean by *tasks* is *sub-goals* that must be accomplished in order to complete the overall objective. The designer of a multi-robot system are required

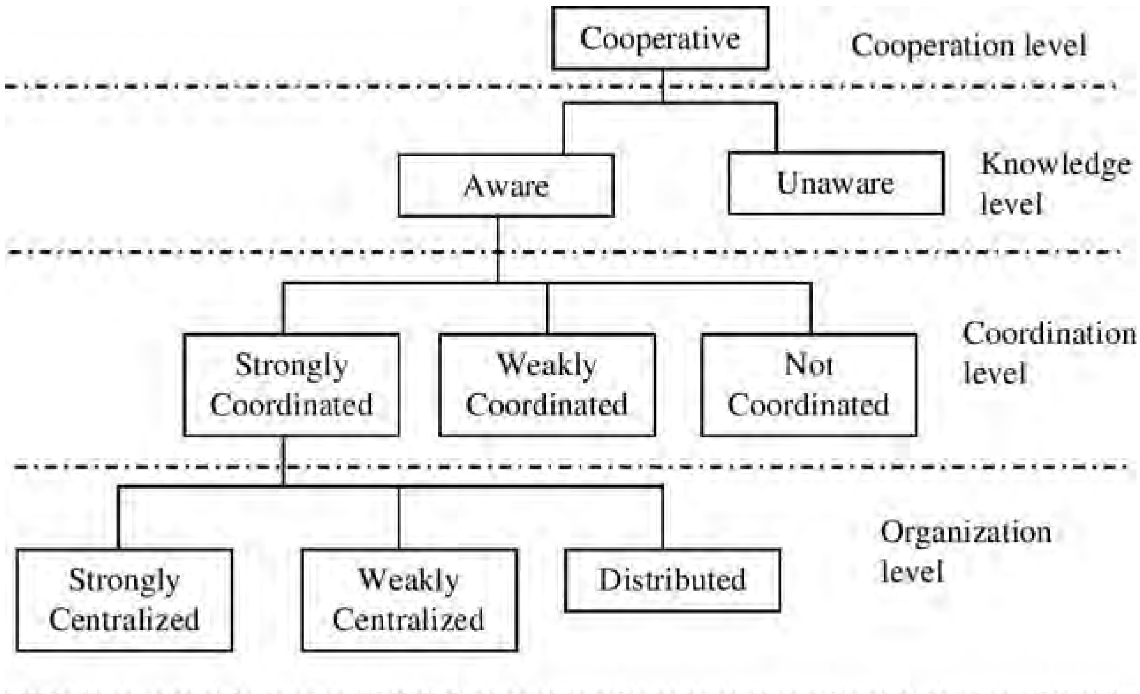


Figure 1.2: MRS Taxonomy Based on Coordination and System Dimensions [13]

to understand the nature of the task allocation problem at hand. For the same reason, the three dimensions of classification indicated in Figure 1.3 are established. Robots can be classified as either single-task (ST) or multi-task (MT), with the latter being able to do multiple tasks simultaneously. While a single robot can complete a single-robot task (SR), multiple robots are needed to complete a multi-robot task (MR). In the case of instantaneous assignment (IA), we can only plan one step or action in the future. For time-extended assignments (TA), the robots can plan for the future, taking into account more than one action or a series of events in the future, due to the wealth of data at their disposal. Sensor and actuator data, task specifications (including what robot capabilities are needed to complete the task, the number and location of tasks, and the arrival pattern of tasks), and environmental characteristics (including the geometry of obstacles) make up the bulk of the data.

Communication is one of the most crucial aspects of the successful deployment of MRS in the real world in accordance with the taxonomies presented in [12–14]. Communications in MRS have been reviewed in [43,44] wherein the authors have highlighted the importance of designing multi-robot coordination methods tailored to address different real-world communication challenges, for example, intermittent communication between the robot peers due to range restrictions. Designing coordination algorithms under the presence of communication range restrictions is the central theme of our research work in this thesis.

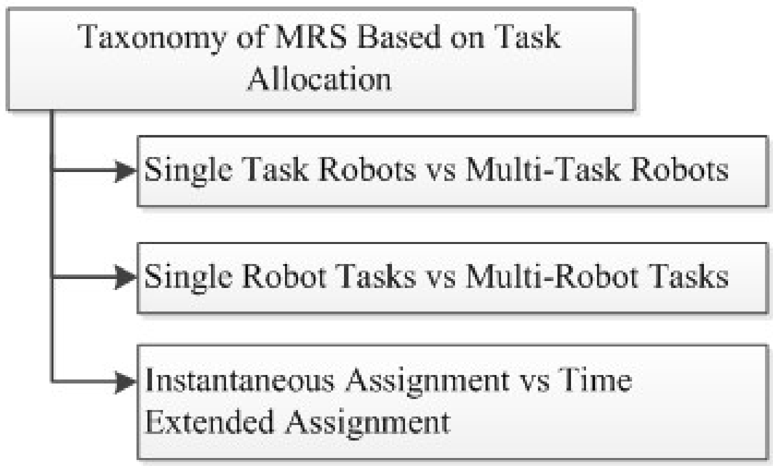


Figure 1.3: MRS Taxonomy Based on Domain Independent MRTA Problems [14]

1.4 MRS LIMITATIONS AND RESEARCH GAPS

Multi-robot systems hold immense potential for transforming various industries, but they face several limitations that must be carefully considered during system design and operation. Overcoming these limitations requires a multidisciplinary approach that involves advances in robotics, artificial intelligence, communication, and human factors to create efficient, reliable, and ethically sound multi-robot systems.

(i) **Communication and Coordination Complexity -** One of the primary challenges in multi-robot systems is coordinating the actions of multiple robots effectively. Communication among robots is essential for sharing information, avoiding collisions, and task synchronization. However, as the number of robots increases, the complexity of communication and coordination also rises exponentially. The need to establish reliable communication links while mitigating interference and latency can hinder system scalability. There is no dearth of literature on multi-robot coordination wherein researchers have ignored the communication complexity and developed methods that work on the premise that communication is omnipresent and the channels are noise-free [45–47]. *Furthermore, in the real world, the communication range of the robots is usually restricted. Maintaining consistency of distributed information is a non-trivial task and requires careful planning. Designing such approaches is the primary objective of this thesis.*

(ii) **Task Allocation and Coordination -** Assigning tasks to robots and coordinating their actions efficiently is a fundamental challenge. The task allocation problem becomes more intricate in large environments where tasks and priorities change over time. Ensuring that robots collaborate without duplicating efforts or leaving tasks unattended requires sophisticated algorithms. Few state-of-the-art approaches have successfully dispersed the robot team while opportunistically allowing the robot peers to communicate with each other. Those who have attempted to maintain continuous connectivity were not able to globally disperse the robots, thereby achieving sub-standard performance [5, 48, 49]. Moreover, some of them resulted in a deadlock, which is resolved with the costly deadlock detection and resolution algorithm [50]. *The approaches suggested in this thesis address the task allocation problem in the context of multi-robot unknown area exploration without restricting the robots to explore locally. At the same time, we have achieved global dispersion without losing connectivity with the teammates for a long duration.*

(iii) **Limited Local Perception -** Many multi-robot systems rely on local perception capabilities to gather information about their surroundings. This limited field of view can lead to incomplete or inaccurate situational awareness. Robots might struggle to make well-informed decisions, especially in environments with complex obstacles. *This thesis addresses the information dissemination problem with a limited field of view of the robots.*

(iv) **Scalability -** While the idea of deploying numerous robots to perform tasks more efficiently is appealing, the scalability of multi-robot systems remains a challenge. As the number of robots increases, the potential for conflicts, congestion, and inefficiencies in resource allocation rises. Additionally, the algorithms and protocols that work well with a small number of robots might not scale appropriately to larger groups, leading to reduced overall system performance.

(v) **Heterogeneity -** Multi-robot systems often consist of robots with varying capabilities, including sensing, computation, and mobility differences. Integrating heterogeneous robots into a cohesive system introduces task allocation, resource management, and communication challenges. Ensuring that each robot contributes effectively to the system's goals while accommodating their differences is a complex problem.

(vi) **Fault Tolerance and Reliability -** individual robots' reliability significantly impacts a multi-robot system's overall reliability. As more robots are involved, the probability of individual robot failures or malfunctions increases. Developing fault-tolerant strategies that allow the system to continue functioning despite the failures of some robots is essential, but it adds another layer of complexity to system design.

(vii) **Interference and Competition -** In scenarios with limited resources or tightly interconnected tasks, robots might compete with each other for access to those resources or tasks. This competition can lead to inefficiencies, conflicts, and even deadlock, where robots are unable to proceed due to mutual dependencies.

(viii) **Security and Privacy -** Multi-robot systems often involve sharing sensitive information among robots to facilitate coordination. Ensuring the security and privacy of this information is crucial to prevent unauthorized access or malicious attacks that could compromise the system's integrity.

(ix) **Cost and Maintenance -** Deploying a multi-robot system involves significant upfront costs, including robot hardware, communication infrastructure, and software development. Maintenance, repairs, and upgrades can also be costly and time-consuming, impacting the system's long-term viability.

1.5 OBJECTIVES

The primary aim of this thesis is to develop effective multi-robot coordination algorithms for addressing the problem of intermittent connectivity while carrying out unknown environment coverage and exploration. The subsequent enumeration presents a list of objectives achieved in this thesis.

1. **Empirical Analysis and Comparison of Various Online Terrain Coverage Algorithms under Communication Range Restrictions**

    (a) Design a Robot Operating System(ROS) based framework for implementing different state-of-the-art decentralized methods and communication range restrictions for online terrain coverage and unknown area exploration tasks.

    (b) Implement some state-of-the-art approaches that assume noise-free continuous communication within the developed framework and expose them to communication range restrictions.

    (c) Establish an experimental test bed comprising of mobile robots to test the efficacy of the selected state-of-the-art algorithms.

2. **Multi-Robot Unknown Area Exploration**

    (a) Design a Player/Stage-based simulation framework.

    (b) Design and implement a multi-robot coordination algorithm for unknown area exploration that are efficient in terms of reduced coverage/exploration completion time and cumulative distance travelled.

    (c) Experimentally compare the proposed algorithm(s) with seven state-of-the-art research works by implementing them in simulation.

3. **Multi-Robot Unknown Area Exploration in a Communication Restricted Environment**

   (a) Design and implement a decentralized multi-robot coordination algorithm under communication range restrictions, which efficiently reduces the redundancy in exploration while minimizing completion time and distance travelled compared to some state-of-the-art approaches.

1.6 SCOPE OF THE THESIS

Some aspects related to multi-robot systems and coordination algorithms that are not addressed at present in this thesis are as follows:

1. *Heterogeneity* - refers to a group or team of autonomous robots that have different physical characteristics, capabilities, or functionalities. These systems consist of robots that are not identical and may vary in terms of their hardware, sensors, software, or tasks they are designed to perform. This heterogeneity can be intentional, as it allows the team to benefit from the complementary strengths of each robot, making them more versatile and adaptable for a wider range of tasks.

2. *Machine learning* - This domain focuses on the concept of autonomous robots learning and adapting collectively. Multiple robots collaborate to enhance their performance through shared knowledge and data. They exchange information, optimize their behaviors, and make collective decisions by leveraging techniques like reinforcement learning, deep learning, and swarm intelligence. This results in an overall improved ability of the robots to carry out tasks efficiently.

3. *3D exploration* - It is an emerging field that extends robotics into complex, three-dimensional environments. In this context, a team of autonomous robots collaborates to navigate, map, and explore intricate, multi-level spaces like underground tunnels, caves, or skyscrapers. These robots employ advanced sensing technologies such as LiDAR and stereo cameras to build 3D maps and identify obstacles,

ensuring safe and efficient exploration.

4. *Fault Tolerance* - These systems are designed to maintain functionality even when one or more robots in the team experience failures or malfunctions. These systems are crucial for ensuring the reliability and robustness of robot teams in complex, real-world environments. They employ redundancy in both hardware and software, enabling unaffected robots to compensate for those with faults.

1.7 STRUCTURE OF THE THESIS

This thesis is structured into four chapters. Following is the overview of the research work carried out in the individual chapters of the thesis:

In Chapter 2, we have briefly surveyed the literature on multi-robot systems. It entails a comprehensive evaluation of several research publications, categorized according to the communication taxonomy presented in [43]. Through the examination of these specific attributes, we are able to discern deficiencies within the current body of study, thereby pinpointing areas that warrant our dedicated research endeavours. This facilitates the attainment of the objectives stated in our thesis scope. Our focus is on the advancement of online terrain coverage (OTC) for multiple robots, taking into account limitations in communication range. This has led to the development of two approaches: a centralized multi-robot frontier tree approach and a decentralized relay-based approach. Both approaches aim to explore unknown areas while considering the constraints imposed by the communication range.

In Chapter 3, we investigated the impact of communication range on multi-robot Online Terrain Coverage (OTC) approaches. Terrain coverage has various real-life applications, ranging from small-scale tasks such as floor cleaning, lawn mowing, and harvesting to large-scale missions like hazardous terrain inspection and battlefield surveillance. However, these tasks are often tedious, time-consuming, and dangerous, necessitating the use of more robots with effective communication capabilities to achieve optimal coordination. While communication is crucial in enabling effective coordination in multi-robot systems, many advanced approaches overlook intermittent connectivity issues arising due to communication range limitations. Therefore, this chapter departs from the assumption of global communication and instead restricts the robots to communicate within a specified range, as in realistic scenarios. To compare the performance of different approaches, we use decentralized algorithms with a team of homogeneous robots, assessing the performance of five state-of-the-art approaches that assume omnipresent communication. We vary the communication range with teams of varying sizes in simulations and on a physical multi-robot test bed. The impact of communication range restrictions on the system's performance has been investigated.

Chapter 4 presents an innovative strategy for multi-robot exploration of unknown areas. Recently, [51], the frontier tree data structure has been utilized in single robot exploration to memorize frontiers, their positions on the map, and the exploration state. One could query this tree to determine the next step in the exploration process. In this chapter, we take the concept one step further for multi-robot exploration by proposing a new abstraction we call the "group." The group is intended to share information through a common frontier tree, requisite operations at the group level, and a method to assign goals to multiple robots. A group is a collection of robots, the explored regions of which, when added together, form a continuous area (a single connected region in a topological sense). The robots are in the same state regarding the exploration task because each group has exactly one tree. When two groups' maps intersect, we propose various methods for merging the frontier trees of both groups. Lastly, we propose a technique that can be used to select nodes from the frontier tree to designate and delegate exploration goals to the various robots. The proposed method outperforms seven research works considered state-of-the-art regarding simulation.

Chapter 5, we proposed a novel approach to multi-robot exploration. This method differs from MRFTE in some ways. This method is based on the robot's independent motion—the robot's classification into explorer and relay roles. Explorers are in charge of investigating the environment, while relays help to disseminate information to other explorers. Conditions may arise in this approach in which explorers cannot communicate because they are not in communication range with one another. The relay then assumes responsibility for transferring information or messages from one explorer to another. We created an algorithm introducing a rendezvous point to improve communication between relays and explorers. This point facilitates the successful exchange of information by allowing relays and explorers to meet at a designated location known as the rendezvous. This algorithm is critical in ensuring effective task allocation. We developed two models for restricting communication range for this disk (range)-based and line-of-sight communication model. We also developed three approaches in this method: multi-explorer with no relay, multi-explorer with one relay, and multi-explorer with multiple relays. Finally, we compared the area redundancy for all three methods while exploring the environment. We discovered that combining relays with an explorer for area exploration in a communication range-restricted environment can reduce redundancy and complete exploration tasks faster than multiple explorers without a relay.

Finally, the chapter concludes the thesis by summarizing the research work presented in the previous chapters and the scope for future work. It also enumerates the publications based on the research work in this thesis.

Chapter 2

Literature Review

2.1 Introduction

MRS is a group of robots that collaborate on the same or different tasks. However, problems arise when robots interact with the physical world, requiring them to modify current strategies to deal with uncertainty, obtain environmental information, and model incomplete knowledge. The development of robot teams capable of cooperating to accomplish a specific goal is a top priority for researchers, who recognize that MRS outperforms single-robot strategies in various applications. MRS is more time-efficient and less prone to a single point of failure, resulting in a more effective solution to a given problem. Researchers have examined natural systems, such as swarms of bees, ants, and humans, to gain a better understanding of how different entities can collaborate to accomplish a task. These early studies led to the use of MRS in a variety of fields, such as object transportation [52–54], foraging [55–57], cooperative manipulation [13, 58, 59], surveillance [45, 60, 61] and search and rescue [62–64], to name a few.

The following essential components of MRS enable multiple robots to work together effectively toward a common task.

Control and Coordination: Effort and time are required to keep track of the numerous components in MRS. To effectively control these diverse robots, the controller must fully utilize their unique capabilities. In scenarios like swarm exploration, where robots are tasked with investigating a predetermined set of targets, it would be inefficient for all the robots to examine the same target. Similarly, for a group of mobile manipulators to lift and transport an object, they must move in unison. The control of the entire system can either be centralized with a single master exercising authority or decentralized. Coordination and proper control are essential for activities such as robot soccer tournaments.

Communication: To ensure the successful functioning of the system, it is crucial to establish effective communication between robot peers. In some robotics applications, communication can be implicit, with information transmission occurring through the environment, such as when insects use pheromone trails to mark explored routes. However, communication can pose significant challenges during exploration tasks, particularly when robots operate away from one another. Proper communication is also necessary for activities such as robot soccer, where the ability to pass the ball between robots is vital. Similarly, proper communication is essential in exploring unknown environments with multiple robots to avoid redundant exploration of the same area.

Localization and mapping: In many robotic applications, robots must build a map of their environment to compute paths, move between targets, avoid collisions, and locate objects. This process is commonly referred to as mapping, while finding one's position in the environment is known as localization. In MRS, each robot must know others' locations to avoid collisions and complete specific tasks. On the other hand, depending on the team members' perception abilities, they can assist each other in collecting data about the environment or improving their localization.

MRS properties can be categorized using the PEAS(Performance, Environment, Actuators, and Sensors) representation model. In this model, *Performance* pertains to the standards employed to evaluate the MRS's effectiveness, such as its speed, accuracy, or resilience. Its metrics could comprise the distance traveled and completion time. *Environment* denotes the physical and operational conditions under which the MRS will function, encompassing factors such as the size of the area to be covered, obstacles present, and lighting and weather conditions. *Actuators* refers to the mechanisms that regulate the movements and actions of robots, such as motors, servos, and grippers. *Sensors* are devices used to perceive the environment and obtain information about the task. Examples of sensors that could be utilized in MRS include cameras, microphones, sonar, and lidar. In the following sections, we provide chapter chapter-specific literature survey of the state-of-the-art on multi-robot online terrain coverage and exploration under limited communication.

2.2 ONLINE TERRAIN COVERAGE APPROACHES

During the last two decades, research on MRS has attracted considerable interest. MRS's primary advantage over one single monolithic robot is that MRS is more robust and fault-tolerant [65]. Individual robots in MRS are less capable when compared with a single monolithic robot, but their innate nature of coordination with their teammates makes them more robust and fault-tolerant [66]. Therefore, it is viable to use MRS for tasks that can be parallelized, e.g., area exploration, terrain coverage, pick and place operations, etc. Fault tolerance is particularly important in unknown environments wherein the robots are susceptible to partial and/or complete failure. Several approaches for online terrain coverage (OTC) were not designed to deal with real-world situations like communication disruptions, but they are frequently used for comparison in simulation. Therefore, it is necessary to test the applicability of these algorithms in real-world situations to make more significant comparisons.

Moreover, some algorithms are designed on the premise that the large-sized robot team is available, the global map of the environment is available apriori, and communication is omnipresent. These are strong assumptions in the real world and should be relaxed when designing OTC algorithms using multiple robots. In [67], the authors deployed a team of more than a hundred simple robots to achieve real-world military missions. The coverage task is classified into three categories, i.e., blanket, barrier, and sweep. The objective of blanket coverage is to maximize the area covered [68, 69]. Barrier coverage keeps track of the variation on the borders while setting up a perimeter around the area of interest. On the other hand, sweep coverage is nothing but a moving barrier. Early research concentrated on surveillance and patrolling missions and therefore employed blanket and barrier coverage [50, 70]. With a small team of robots, these approaches are not capable of completing the coverage task. Sweeping robots traverse the terrain while ensuring that the obstruction-free region is visited at least once. The robots coordinate with their teammates to share coverage information and progressively cover the terrain while avoiding redundant coverage [71, 72]. Patrolling robots repeatedly traverse a designated region in order to obtain the latest state of the dynamic environment. Sweeping robots can complete the OTC task even with a smaller-sized multi-robot system on the premise that at least one robot survives until task completion.

Decentralized group organization of MRS is generally preferred to take complete advantage of the redundancy offered by the system and design algorithms that are scalable, robust, and fault-tolerant [45]. However, centralization has its advantages when the team size of MRS is small. For example, in [73, 74], the robots form a communication bridge with a fixed base station for delivering information in a multi-hop manner. In [71], the robots reduce the odometry error while localizing w.r.t, another stationary robot that acts as a beacon. These approaches demand exact and expensive sensors for the global coverage map to be centrally constructed [75–77]. Some approaches (both centralized and decentralized) are designed to achieve global dispersion such that the robot team quickly gets dispersed in the unknown environment for exploration, and when they meet up, they merge their maps to build a standard, consistent global map [3, 77, 78]. Nevertheless, it is a non-trivial task because of intermittent network connectivity.

In the real world, both the communication range and the bandwidth are limited [43]. Most of the centralized OTC algorithms work with the premise that the individual robots can always communicate with the central planner/base-station [3, 50, 74, 79, 80]. In the outdoor environment, the robots may travel to in far-off regions while conducting the coverage task, or due to disaster, the communication infrastructure may break down. In an indoor environment, thick concrete walls, metal surfaces, and mirrors act as physical barriers to wireless communication. These are some of the possible reasons for the multi-robot network to get partitioned. Therefore, research interest has shifted to achieving OTC under communication restrictions [81]. Most of the robotic systems are equipped with some form of local communication like Wi-Fi, Bluetooth, and Zigbee modules. It is possible to communicate with the robot peers in a specified communication range or when the robots are in line-of-sight [82, 83] of each other using short-range wireless communication technologies. In applications like battlefield surveillance, the robot team must cover the terrain as fast as possible in search of landmines and minimize redundant coverage.

After the terrain map is discovered completely in applications like patrolling, repeated coverage is desirable for monitoring the terrain [84]. In some research works, the robots deploy beacons/smart tags in the terrain while performing OTC tasks. The robots store the coverage information in the beacons to avoid redundant coverage [4, 85]. The beacons communicate with each other, and the coverage information propagates in the network of beacons. The robots themselves do not share the coverage information. Instead, they obtain and update the coverage information using these beacons. However, the beacons themselves are prone to failure. Small living species like ants inspire these approaches [70,86,87]. Two solutions exist for addressing intermittent connectivity. First is rendezvous-based approaches [88], and the second is to design multi-robot coordination algorithms that are connectivity aware [89]. The rendezvous-based approaches may not be suitable when a faster task completion time is desirable. However, they can still be used along with dispersion techniques [90], which will speed up the task. The connectivity-aware multi-robot coordination algorithms are viable in situations when the terrain is bounded. For unbounded terrains, moving base stations can be deployed [91].

2.3 UNKNOWN AREA EXPLORATION - CENTRALIZED APPROACHES

Area exploration is crucial in various fields such as planetary exploration, search and rescue, agriculture, cleaning, and dangerous locations like mined lands and radioactive zones. To achieve efficiency and robustness, exploration is usually carried out by multi-robot systems. During exploration, robots encounter various geometrically challenging regions due to impediments such as furniture, and they must navigate around them. The robots need to explore these environments independently without relying on pre-designed maps created explicitly for them. This autonomy enables them to adapt to their surroundings and operate effectively.

The idea of the frontier for unknown area exploration is presented in [5]. Over the course of time, the frontier-based methodology was combined with a number of other heuristics. Research efforts have been focused on heuristic-based techniques, emphasizing the interaction between the cost of the distance, the utility, and the amount of information gained. In [92], a utility function was used to change the utility of the frontiers given to other robots for exploration. In spite of the fact that maintaining constant contact throughout the exploratory process may potentially simplify things, it became clear that the robots needed a shared map. Target points were allotted to each individual robot based on the calculated costs and utilities associated with them. Using a method known as the "next-best view," which involved balancing the costs of traveling with the amount of information gained, [93] explored a new point of view. When traversing unexplored terrains, the utilization of several robots was undertaken primarily with the intention of achieving comprehensive environmental coverage. With this line of research, a new method was proposed in [94]. This method incorporated a distance cost measure obtained from solving the Traveling Salesman Problem by employing the Chained Lin-Kernighan heuristic. An algorithm for the online exploration of repetitive tasks was suggested in [95]. This algorithm considered the current expenses and the prospective future rewards of exploration efforts. The algorithm used a greedy strategy, which selected the most effective exploratory action for each task repetition by striking a balance between the costs and benefits of the action in its immediate environment. Various techniques have been explored in these studies to coordinate robots during exploration. In [96], robot coordination is achieved through environment segmentation, where each robot is confined to exploring within its designated segment. This segmentation is achieved by initiating wavefronts from the robots towards the frontiers, stopping their growth upon collision. However, these methods require a shared map and centralized planning. Most algorithms rely on graphs or trees, such as the Voronoi graph, for graph-based exploration and Rapidly Exploring Random Trees (RRT) and frontier trees for trees.

In [8], a Voronoi graph depicting the environment is created, and critical spots at doorways and narrow corridors are obtained. The graph is then partitioned into multiple parts, each allocated to a separate robot using the Hungarian approach. This method decreased the overhead associated with the recurrent allocation of frontier cells. However, successful map segmentation cannot be obtained if the graph has only a few crucial points. To enhance this approach, [97] extends the segmentation algorithm by merging several partitions, further optimizing the exploration process by reducing the overall travel distance. Furthermore, the approach is centralized and needs to be more resilient to failure. K-means clustering is utilized to segment the landscape in [98]. Frontier cells in the inner and exterior of the robot's partition are allocated distinct charges. This method necessitates many calculations to repeatedly cluster frontier cells in the unknown region, particularly when the environment is large. Voronoi partitions are created using a distributed approach in a recent work [99]. Each robot respects the territorial bounds of the Voronoi partitions into which it falls and does not cross them. Each robot explores the boundaries of its zones to identify inaccessible patches owing to impediments. After that, the patches are auctioned off to other robots. In arbitrarily complicated terrains, this approach ensures comprehensive, non-overlapping coverage. However, this strategy does not guarantee a fair workload allocation in a multi-robot system; for example, some robots may explore extensive territories while others lie inactive. This strategy does not account for one or more robot failures. Instead of assigning vast unknown segments to the robots, [79] assigns subsets of tasks made up of known frontier cells grouped using the Geodesic K-means algorithm [100] to the robots using the Hungarian technique [101]. In this approach, robot dispersion is an emergent behavior, meaning that when fewer frontier cells are known, the robots are locally dispersed at the start of the coverage task. As the map unfolds and more frontier cells become visible, the robots disperse from each other. [6]

suggests a simple yet straightforward landscape-covering approach. This method is based on structured trajectories. Therefore, a single robot can cover simple regions using spiral paths. A backtracking method detects and connects other sections because the spiral paths end in the middle of the rectangular zone. Structured patterns, it has been proposed, are more efficient since they sweep or cover lengthy contiguous segments and require robots to take fewer turns. [1] proposes a multi-robot adaptation of [6]. However, this strategy employs a basic allocation scheme, increasing coverage redundancy. This approach still needs to be improved when dealing with robot breakdowns.

The Frontier-based Rapidly Exploring Random Tree (RRT) exploration method is useful for exploring unknown environments. However, its greedy nature results in multiple robots focusing on the same high-revenue zone, leading to significant overlap. To address this issue, a group of researchers has proposed a new exploration technique called the Temporal Memory-based RRT (TM-RRT) [102], which involves multiple robots collaborating in the exploration process. This technique uses adaptive duration and income to determine the value of each frontier based on the robot's position, resulting in more efficient and robust exploration of new environments. In their paper [103], the authors proposed a method for exploring uncharted territories involving Frontier and a tree-like task assignment structure. It is known as the Frontier Tree algorithm because each frontier is stored in a tree-like data structure. One of the disadvantages of this algorithm is the increased computational effort, which is tolerable in comparison to the increase in travel distance toward the end of exploration when greedy algorithms update only a small percentage of the map. They employ a single robot to investigate an uncharted environment.

2.4 UNKNOWN AREA EXPLORATION - DECENTRALIZED APPROACHES

In [104], a topological map-based exploration method for multi-agent exploration is suggested, viz., MR-TopoMap in communication-constrained situations. Each robot has its own local grid map for path planning, which is not shared. Regarding the exploration assignment, a robot's ability to select an appropriate trajectory depends on the locations of the other robots and the unexplored regions. The authors have shown that MR-TopoMap increases exploration efficiency by 23–77%, and in comparison to the occupancy grid map scheme, reduces data transmission by 84% to 90%. In [105], a method for Propagation Environment Modelling and Learning (PropEM-L) is developed that predicts signal intensity by learning attenuation in complicated, communication-restricted situations. The authors have compared their neural network-based online learning algorithm with the standard received signal strength(RSS) prediction methods and demonstrate PropEM-L on a dynamic network of exploring robots and stationary antennae in various communication-restricted, underground locations. This algorithm is shown to improve RSS prediction and adapt to novel settings. A novel distributed controller for multi-robot exploration is presented in [106] that autonomously decides at each time step, based on the current system state, how to weigh network reliability, connectivity, localizability, and information gain. The presented distributed controller achieves greater coverage than the state-of-the-art while reducing localization uncertainty. However, this approach works with the premise that the robot's movement is error-free. GVGExp [30], a recurrent connection exploration approach for multi-robot systems, explores unforeseen environments under communication constraints. This approach successfully minimizes the communication events between robots and path interference. Generalized Voronoi Graph (GVG) is created by robots in a progressive manner and is used by the robots to traverse and determine the topology of the unknown environment. This method, however, produces suboptimal results because the method used for portioning the area does not guarantee balanced workload distribution for the individual robots. In [107], the authors have suggested a multi-robot exploration framework that comprises a mission-based protocol to address the absence of global communication. This protocol allows robots to independently explore the environment and reach pre-specified places at the scheduled time for information exchange with their peers. It is a rendezvous-based approach that may take a longer time and distance to be traveled by the robot team to complete exploration if the meeting point is not decided carefully. Another research work, [108], presents a behavior-based approach for multi-agent exploration of the unknown non-convex environment. It is not mandatory for the agents to always maintain continuous connectivity. This approach is based on the behavior of ants laying pheromone trails by using artificially placed markers in the environment. Using these markers, the agents avoid redundant exploration. Robots exchange information pertaining to the partial maps they have discovered and the markers with their peers whenever they are inside the communication range of each other. In general, ant-based approaches have a limitation in that they need special hardware for trail laying and sensing. It is also possible that the markers get damaged, rendering the suggested approach in [108] ineffective. In [109], the authors proposed an approach for multi-robot systems to learn and simultaneously update the communication map while exploring the unknown environment. The multi-robot system constructs a communication graph incrementally in a communication-restricted environment. As the robots explore the environment, they keep updating the communication map, i.e., the vertices represent the locations visited by the robots, and the edges represent the strength of the radio frequency signals. The robot team uses this communication map to predict the possibility of communication with the base station without using relays. In this work, the authors have attempted to ensure connectivity between robots and the base station. However, in many real-world situations, like an exploration of a building that looks visually intact from the outside but is earthquake hit, building a multi-robot network that ensures continuous connectivity is not possible.

Subsequently, in this chapter, the literature survey on MRS is conducted in accordance with the *characteristics of the environment* in which MRS has deployed the composition of the robot team, followed by inter-robot communication and coordination algorithms. We have recently come across several research papers on coordination algorithms. These algorithms can be divided into four phases: *task decomposition*, *task allocation*, *task exploration*, and *task termination* upon completion. According to *MRS characteristics*, these algorithms' control architectures can be centralized or decentralized. In centralized control architectures, robots remain connected to the base station, so communication criteria are not a major concern. However, in the decentralized approach, connectivity issues between robots and the base station may arise. Therefore, in the *communication characteristics* section, we have discussed research papers that focus on communication-based solutions for the decentralized approach.

2.5 ENVIRONMENT CHARACTERISTICS

This section outlines several essential environmental attributes and their corresponding sub-attributes, as shown in Table 2.1.

Table 2.1: Environment Characteristics

Environment Attributes	Sub Attributes	Related Papers
Knowledge of the environment	Known	[50, 75, 110–115]
	Unknown	[5, 8, 11, 19, 29, 38, 58, 60, 62, 63, 74, 88, 89, 91–93, 96, 108, 109, 116–140, 140–176]
	Known bounds	[91, 96, 177–179]
	Known bounds with holes	[180–182]

2.5.1 Knowledge of the Environment

The environment in which the robot team operates can be known beforehand, or it can be a completely unknown environment, with the exception that sometimes only the bounds are known. Some examples of the known environment are that of warehouses [116, 183], surveillance [18, 184] and search-and-rescue operations for unknown environments [19, 185]. We first delve deeper into the attributes of the known environment and explain how a robot can have complete knowledge of all possible outcomes of its actions. In contrast, we also discuss the challenges faced by the robots when they incrementally acquire knowledge of the unknown environment before making any decision. The known environment has two sub-types: known bounded [172, 186] and known bounds with holes [181, 182]. In a known bounded environment, the robot is aware of the overall boundary, but unexpected obstacles may be present that the robot must navigate around [187]. For instance, we consider a scenario in which a robot is tasked with delivering medication to patients on a specific floor or ward of a hospital [188]. On the other hand, in a known bound with holes environment [180], the overall layout and boundaries are known, but certain areas are restricted. For example, we explore a library where a robot collects books from various shelves. These restricted areas, or holes, present a challenge for the robot. Nonetheless, by incorporating the knowledge of these holes, the robot can determine the external boundaries and internal obstacles it must avoid.

2.6 MRS CHARACTERISTICS

This section outlines some essential attributes of MRS characteristics and their sub-attributes. Table 2.2 summarizes the characteristics discussed in the literature.

Table 2.2: MRS Characteristics

MRS Attributes	Sub Attributes	Related paper
MRS composition	Homogeneous	[5, 8, 11, 19, 38, 50, 58, 62, 63, 74, 88, 89, 91–93, 96, 110–125, 127, 128, 130, 131, 134–138, 140, 141, 143–145, 147, 148, 150–159, 161–165, 167, 171, 177–180, 186, 189, 190]
	Heterogeneous	[29, 75, 83, 129, 133, 139, 151, 191]
MRS control architectures	Centralized	[11, 38, 50, 58, 63, 75, 83, 89, 92, 112, 114, 115, 123, 124, 129, 134, 136, 145, 150, 162, 177, 189]
	Decentralized	[5, 8, 19, 29, 58, 62, 88, 91, 96, 110, 113, 116, 118–122, 125, 127, 128, 130, 131, 133, 137, 139, 140, 140, 141, 143–145, 147–149, 151, 152, 155–159, 161, 163, 165, 167, 171, 178–180, 186, 190, 191]

The main objective of MRS is to achieve proper coordination among the robot peers [23, 47, 92, 165]. During movement, robots share information like their pose in both local and global frames. When examining the characteristics of MRS, the initial position of each robot is a crucial aspect to consider. It is essential to determine whether all robots start from the same location, such as a base station [89], or different locations [153]. If the robots have a common starting point [50, 192–194], then it remains easier for them to discover the pose of their peers. Furthermore, with known poses, in tasks involving environment monitoring and mapping, the information dissemination within the group is more accurate. On the other hand, when the robots start from different locations [153], determining the pose of the peer robots is complex, and information integration is less accurate. The size of the environment is also critical since it affects the success of MRS exploration, along with other important factors like the type of robots, sensors, speed, computation speed, and simulation software used. These factors significantly impact the navigation, communication, coordination, and task completion of MRS. Therefore, selecting an appropriate robot and environment configuration is necessary to ensure efficient navigation and task completion performance.

2.6.1 MRS Composition

A homogeneous MRS [157, 186] comprises robots with identical capabilities, sensor suites, and hardware. They have similar abilities and can complete similar tasks. These robots can efficiently coordinate their actions since they are equally capable of performing tasks. According to [195], this composition has several advantages, including simplicity, reduced complexity, and lower cost. Whereas, A heterogeneous MRS [196, 197] comprises varying robots with differing capabilities, sensor suites, and hardware. They possess different skills and can execute different duties. This makes coordinating their actions more challenging since not all can execute similar duties. According to [198, 199], Some advantages of this mixture include increased versatility, efficiency, and strength. Most researchers prefer homogeneous MRS due to its simplicity and affordability compared to heterogeneous MRS, as shown in Table 2.2.

2.6.2 MRS Control Architectures

In a *centralized control architecture*, a single central controller manages the actions of multiple robots [200]. This controller receives sensory data from the robots and sends commands to direct their actions. Although this structure is simple to implement, it is vulnerable to failure due to its reliance on a single point. According to [201], the centralized approach is similar to the leader-follower approach. The commonly used algorithms for centralized techniques in task allocation are the Hungarian approaches [202]. In the Hungarian-based approach, a cost matrix based on distance is utilized to assign tasks to robots. The aim of this approach is to minimize costs and obtain an optimal solution. However, it is important to note that centralized task allocation algorithms have limitations in terms of robustness and scalability. On the other hand, in a *decentralized control architecture*, there is no central controller, and the robots make decisions independently. While they share information and coordinate actions, this control architecture is more complex. Each robot works to optimize its objectives within its specific boundaries while interacting with other robots to achieve a common global goal. In the decentralized auction-based task allocation method [203], a bidding method is used to assign tasks to robots. Both approaches aim to minimize costs and obtain an optimal solution. In their article [204], the authors discuss the optimal control architecture to utilize based on the specific application requirements and desired system performance. Centralized control is easy to implement but prone to a single point of failure. The decentralized control is more robust but challenging to set up.

In the field of MRS research, researchers have traditionally relied on a centralized control architecture [124, 129]. This approach has been effective for simpler systems with either known or unknown surroundings. However, as the complexity of the system and the number of robots increases, managing and coordinating communication becomes more challenging. To address this issue, researchers have started to adopt a decentralized approach [131,133]. However, decentralization of the robots poses significant challenges, such as fault tolerance, scalability, collaboration, and cooperation.

When designing a multi-robot system (MRS), several critical factors need careful consideration. These factors include the speed of the robots, the type of sensor used, the range and field of view (FOV) of the sensor, and the specific robot type being employed. High-speed robots are capable of executing tasks faster, while different sensor types, ranges, and FOVs affect how well robots perceive their surroundings and interact with each other. Choosing the right robot for the MRS has a direct impact on the system's performance and capabilities. To optimize MRS performance, it is crucial to select robots and sensors carefully and design efficient algorithms and control strategies that account for each robot's species, sensors, and capabilities. Robots working in a team can explore their environment faster by communicating and coordinating with each other. In the future, we will focus on the characteristics of communication and coordination when robots work in a centralized or decentralized manner. To ensure success, it is crucial to establish a reliable communication system. Effective communication is vital in robotics, as it involves transmitting and receiving information among various components of the robot, such as the control station operator to the robot, robot control device to robot hardware, robot running node to running node, and so on. All these examples of communication fall under this category. Mobile robots commonly use wireless communications, such as Bluetooth, Wi-Fi, and cellular networks, due to their numerous benefits [205]. Establishing an effective decentralized strategy requires determining the transmission modes of communication, which can be either *half-duplex or full-duplex*. In *half-duplex communication*, robots can only send or receive flags, which can limit communication and coordination. An example of inter-process communication using a Zigbee modem [206] is available. Meanwhile, robots can achieve *full-duplex communication* by transmitting and receiving flags simultaneously, which is exemplified by Bluetooth technology [207]. In their paper [208], the authors present several practical applications that utilize two modes of communication for robots. The first mode is long-range and low-bandwidth, utilizing technologies such as XBee. The second mode is a shorter range but high capacity, using WiFi. Additionally, the authors introduce three algorithms: network formation, which leverages subroutines cluster-connection and star-connection. In the study [168], the authors provide clear evidence that their proposed exploration technique, which utilizes a fixed Bluetooth chain-like team, outperforms existing approaches in obstacle-free scenarios where limited connection to the base station is an issue. They accomplish this by creating a static networking architecture that is not ad hoc.

2.7 COMMUNICATION CHARACTERISTICS

This section summarizes the key qualities of effective communication and their sub-attributes. See Table 2.3 for a summary.

Table 2.3: Communication Characteristics

Communication Attributes	Sub Attributes	Related paper
Connectivity Model	No Connectivity	[75,118,209]
	Event-based	[46,210]
	Continuous	[91,173,211]
	Time-triggered	[180,212]
Communication model	No Communication	[112,148,150,163,180]
	Line of Sight based	[29,83,89,121,148,170,171,191]
	Range based	[5,8,19,29,50,58,63,74,75,75,83, 91,92,110,111,113,113–115,118–120, 122, 123, 125–127, 129, 131, 133–135, 137, 139, 140, 143, 144, 148,149,152,155,158,163,164,170, 177–179,189,191]
	Signal based	[170,171]
	Traces-based	[96,116,141,163,167,171]
Communication necessity or need	Periodic	[19,83,140,147–149,152,155,158, 159,163,164,166,170,171,186]
	Aperiodic	[5,8,75,113,118,123,127,135,137, 139,141,177–179,191]
	Continuous	[11, 19, 38, 50, 58, 63, 74, 88, 89, 91,92,110,111,114–116,119–122, 124–126, 129, 132–134, 140, 143, 144,162,167,189]
Inter-Robot network	Fragmentation	[8,38,50,74,75,111,119,121,125, 130–132,189,213]

2.7.1 Connectivity Model

Effective coordination is crucial for autonomous robots to work together towards shared objectives in tasks like search and rescue or swarm robotics. To achieve proper coordination, robots need to communicate with each other by exchanging data and sharing information. Connectivity plays a vital role in providing reliable links between robots, which is essential for seamless interaction and teamwork. Improper communication methods can lead to uncoordinated systems, resulting in ineffective performance. The robots' communication networks facilitate proper coordination between the team of robots. Multiple categories of connectivity methods are explained below [43, 176].

- **No connectivity:** No connectivity in robotics occurs when robots can't communicate due to infrastructure failure, physical obstructions, or distance [5]. This means robots rely on their sensors and decision-making abilities to complete tasks, which can result in sub-optimal performance or mission failure if they can't complete tasks independently.

- **Event-based connectivity:** Event-based connectivity is a communication model used in MRS where robots communicate based on specific events or triggers instead of continuous communication [171]. It is particularly useful in dynamic environments where communication needs change frequently. The robots are only connected when communication is necessary, helping to save energy and reduce communication overhead.

- **continuous connectivity:** Robots in MRS communicate continuously using continuous connectivity. This allows robots to collaborate in real-time, improving performance. Continuous connectivity allows a group of robots mapping an unknown environment to share sensor data and coordinate their movements to cover more ground and create a detailed map. It takes a reliable infrastructure and nearby robots to use wireless or wired communication to achieve this model. If communication infrastructure fails, the entire system may go down. Continuous connectivity is bandwidth- and power-intensive. In [89], a group of robots explores an unknown 2D environment while maintaining continuous connectivity with a single base station. In [214], They propose a distributed algorithm for continuous robot-to-base station connectivity. In this algorithm, swarm members form a network topology and role switch to deploy repeaters for base station connectivity.

- **Time-triggered connectivity:** Time-triggered connectivity is a communication model used in MRS where robots communicate at predetermined intervals rather than continuously [29, 212]. This means that the robots are only connected at specific times. Regular communication intervals, such as once per hour, can be established when deploying robots for surveillance in a large area.

In their paper [215], the authors address two communication challenges related to connectivity when using ROS as a middleware. The first challenge is a time-based physics-based ROS simulator toolkit, while the second challenge is an event-based network simulator. There are a few papers that focus on the connectivity between MRS. In [216], They compare the bandwidth and speed restrictions of local and global connectivity maintenance on an MRS. In [217], They also proposed several local and global connectivity maintenance algorithms for distance-dependent communication topology. In [30], they generate a graph-based connectivity between the multiple robots. They also suggest that robots can conflict when they explore the same area, so communication between robots becomes important. In [218], They provide a definition of connectivity that requires agents to maintain line-of-sight, which is guaranteed by obstacles. When computing the graph topology, each agent considers its environment task, and the tasks are calculated based on the Minimum Spanning Tree (MST). Simultaneously, some researchers will focus more on network connectivity. In their study [219], the authors evaluate the effectiveness of simultaneous and iterative link removal techniques on various types of network connectivity. In their study [220], researchers conducted experiments on preserving network connectivity in a multi-robot system. The study showed that a group of three robots could successfully preserve network links. The researchers also noted that wireless communication and sensor observation networks can have varied topologies, which makes preserving network links a challenging task. Most strategies for preserving network connections assume similar topologies and rely on robots communicating with their neighbors. However, this assumption is false since some sensors, such as cameras, are only available in the line of sight. Nevertheless, wireless communication can still occur through diffraction or reflection, even if the robot is out of sight. Therefore, each robot must simultaneously observe its relevant neighbors' relative positions and network IDs to interact wirelessly.

Whereas continuous connectivity [211] can limit the freedom of movement for robots in exploration strategies. However, despite continuous connectivity, limited-range communication can prevent robots from reaching certain parts of the environment. Another important factor is the control station (CS) to which the gathered information must be periodically delivered. This factor is particularly useful in exploration missions that involve search and rescue operations. During an exploration mission, a mobile robot can communicate with its teammates (and potentially a CS) through a radio channel, such as Wi-Fi [205] or Zigbee. However, radio channels can be degraded as the distance between robots and physical obstructions increases. In this context, the ”communication model” refers to a robot's communication capabilities, which decide where it should go next. The communication model predicts that robots will communicate within a specific range when mutually visible, but this may only be true when robots are near each other [221]. This conservative method can be used to construct robust multi-hop chains [19, 46] between robots to achieve tasks such as live video streaming between a location of interest and the CS. It is considered a very safe method because it rarely produces false positives. However, in some scenarios, the communication model is only assumed [201, 222]. In these cases, robots do not determine which locations to visit based on the possibility of inter-robot communication. As a result, communication can only occur when it is convenient. A list of the most common types of communication models [43, 169, 173] used in MRS organizations:

2.7.2 Communication Model

The term ”communication range restriction” in a MRS refers to the limit on the distance over which robots can communicate. This limitation is due to factors such as communication device power or environmental obstacles, and it means that robots can only exchange information when they are in close proximity to each other. This constraint is important because it affects how robots collaborate, form teams, and distribute tasks. Robots that are within the communication range can share data and coordinate with each other, while those that are beyond it must operate independently. When designing MRS algorithms, engineers need to take this limitation into account. The communication model that they choose will determine how the robots connect and share information, and it is directly affected by the communication range. Researchers should consider these aspects to ensure that the MRS operates efficiently. There are different communication models available, including:

- **No communication**: In MRS, it is possible to have a type of system called an "uncoordinated" or "asynchronous" system, where there is no direct communication between the robots. In such a system, each robot acts independently, without direct communication with the other robots [5, 112, 180, 223]. MRS only sometimes requires communication between robots. There are several situations where robots can perform their tasks without communicating with each other. For example, when robots are assigned to different locations or performing tasks that do not require coordination, such as exploring an unknown environment. Furthermore, in some environments, the communication channel might be unreliable or unavailable due to external factors, such as noise or jamming. There are several advantages of having no communication in MRS. One of the benefits is simplicity since there is no need for communication infrastructure or coordination. Another advantage is robustness because the system can operate even if the communication channel is disrupted. Finally, the need for more communication makes the system cost-effective.

- **Line of Sight based communication**: In MRS, Line of Sight (LoS) based communication is a type of wireless communication in which the communication link between the robots is established through a direct, unobstructed path. This means no physical obstacles, such as buildings or trees, should block the signal between the robots [89, 170, 171, 173]. Examples of LoS-based communication in MRS include microwave or millimeter wave communication systems or infrared communication. These systems typically have a limited range and require a clear path between the robots for reliable communication. Low latency, high data rates, reliability, security, and accuracy are advantages of LoS-based MRS communication.

- **Range-based communication**: In MRS, Circle or Range-based communication refers to a type of wireless communication where robots can only communicate with each other if they are within a certain range or radius [46,91,177]. This means that only robots located near each other can establish a communication link. Several wireless technologies, such as WiFi, Zigbee, Bluetooth, ultrasonic or infrared communication, etc., have limited communication ranges. In beacon-based localization [160], robots periodically broadcast their location, and other robots can listen and determine their relative position based on the received signal strength. The use of Circle or Range-based communication in MRS has several advantages. Firstly, it helps reduce the amount of noise or interference in the system since communication is only established between robots located close to each other. Secondly, it is a more efficient approach as it limits the number of robots that need to be coordinated. Lastly, this approach allows for independent coordination as small groups of communication are created between the robots.

- **Signal based communication**: Wireless communication in MRS that relies on transmitting signals between robots is called signal-based communication. This communication model enables the robots to exchange information, such as their location, actions, and sensor data [78, 224]. Examples of signal-based communication include Wi-Fi, Zigbee, Infrared (IR), Ultrasound (US), and Wireless Sensor Networks (WSNs) [171, 176]. Real-time communication between the robots and a central control system is possible through signal-based communication, allowing for coordinated actions.

- **Traces-based communication**: Traces-based communication in MRS is wireless communication that relies on the robots leaving behind a trace or a record of their actions and movements [167, 171, 223]. Other robots can then detect and read these traces to gain information about the actions and movements of other robots. Robots leave behind a trail of pheromones [225], markers [226] that other robots can detect and follow to reach a certain location. They broadcast their location and actions, which other robots can use to update their map of the environment. Traces-based communication in MRS offers several advantages, including direct communication between robots, even in noisy or jammed environments. This communication method is commonly utilized in swarm robotics [227], where robots can collaborate to achieve a common goal without a central controller. The primary objective of this collaboration is usually to accomplish cooperative tasks.

The paper [228] presents three algorithms for exploration tasks under communication range restrictions. In the Rolling Dispersion Algorithm (RDA), Wi-Fi is used for communication between robots, and This algorithm uses beacons as markers to mark the explored path. Sweep Exploration Algorithm (SEA) For chemical signals or line-of-sight with cameras and LEDs are used in communication between robots. The train Exploration Algorithm (TEA) is similar to the Sweep Exploration Algorithm (SEA), but here, robots remain in the group like a train moving on a track. On comparing all of these algorithms, SEA is 1.35 times faster than RDA in achieving full coverage and works better than RDA in environments with long paths.

2.7.3 Communication Need

In MRS, there are different types of communication needs, including:

- **Periodic communication**: This type of communication is characterized by the need for robots to communicate at regular intervals [163, 164]. For example, robots may need to periodically exchange their positions or sensor data to coordinate their actions. This type of communication is helpful for tasks such as formation control, where the robots need to maintain a specific configuration [229].

- **Aperiodic communication**: This type of communication is characterized by the need for robots to communicate only when necessary [8, 75]. For example, robots may need to communicate only when they detect an obstacle or reach a specific location. This type of communication is helpful for tasks such as exploration, where the robots need to adapt to environmental changes.

- **Continuous communication**: This type of communication is characterized by the need for robots to communicate constantly [88, 150]. For example, robots may need to communicate in real-time to control a shared resource or to maintain a shared map. This type of communication is helpful for tasks such as navigation and localization, where the robots need to know the positions of other robots in real-time.

The type of communication needed will depend on the specific task and the requirements of the MRS. For example, in a search and rescue task [230], aperiodic communication is probably more suitable because the robots can move independently and only need to communicate when they detect something interesting. On the other hand, in a swarm robotic task [231], continuous communication is probably more suitable because the robots need to work together in real time to achieve the task.

2.7.4 Fragmentation

Fragmentation [148, 191] in MRS refers to the situation where the network of robots becomes disconnected, which means that some robots can no longer communicate with others. This can happen due to various factors, such as obstacles blocking communication paths, the limited communication range of the robots, or the failure of some robots. Fragmentation can significantly impact the performance and efficiency of MRS as disconnected robots cannot share information or coordinate their actions. This can result in sub-optimal performance or even failure of the overall design. Fragmentation can occur in different ways in MRS, such as physical (caused by obstacles), logical (caused by the failure of some robots), and topological fragmentation (changes in network topology due to robot motion).

In [232] a decentralized approach to evaluate inter-robot communication. The study found that communication can improve performance in tasks that require less communication. Three communication models were introduced and tested in three different environments. The results suggest that low-level communication strategies are more effective than complex ones. In [233], They provide a new idea for a lightweight open-source system called Robofleet that provides inter-robot communication, remote monitoring, and remote tasking for a heterogeneous fleet of ROS-enabled service-mobile robots. They also mention that it is better than the ROS when there are many robots in the environment. They used ROS2 to solve their problem. In [234], A strategy for collision avoidance in communication-based multi-agent navigation was presented. The strategy involves improving agent navigation by communicating hidden state information. A self-attention model is employed to encode neighbor observation and link prediction for inter-agent communication. In various circumstances, the approach outperformed other learning-based baselines in simulation.

There are few research also exist that work on aerial, underwater, and ground robots or a combination of them. In their paper [197], the authors presented a technique that enables decentralized multi-robot systems to establish ad-hoc communication links with specific target robots based on their surroundings. They used Markov chains to model a spatially targeted communication protocol between aerial and ground robots in a swarm to achieve this. The technique involves a swarm of marXbots combined with several aerial robots that work together to climb a hill. In their paper [170], the authors proposed a distributed exploration method for 3D environments. The robots always maintained communication range with one another and exchanged limited data - only the movement direction of each robot. The method was computationally efficient due to a heuristic function and a greedy computation strategy. The authors also presented an efficient deadlock recovery strategy. The exploration algorithm was tested on a simulator for autonomous underwater vehicles.

In their paper [234], the authors propose a collision avoidance strategy for multi-agent navigation based on communication. They developed a self-attention model to encode neighbor observations and predict links for inter-agent communication.

2.8 COORDINATION CHARACTERISTICS

Effective coordination among multiple robots that share a common objective is crucial for exploring unknown environments [162]. This coordinated approach has potential applications that span from surveillance and sea rescue to space exploration. When the area is too vast for a single robot, multiple robots can increase efficiency and coverage, which is particularly important given the significant time and energy required to map new terrain and adhere to constraints. However, coordinating numerous robots presents a significant challenge that requires the development of sophisticated algorithms for coordination, control, and communication to ensure optimal outcomes. The primary objective is to optimize performance by identifying the key factors that contribute to successful coordination.

In the field of path planning, recent works have made significant progress toward guaranteeing complete coverage of free space. This is especially important in applications such as de-mining, where it is critical to cover the entire environment. To achieve this, the most effective method is to use cellular decomposition. Three types of cellular decomposition are commonly used: approximate, semi-approximate, and exact.

2.8.1 Task Decomposition

Task decomposition is a method of breaking down a complex task into smaller, more manageable subtasks. This approach allows for more efficient and effective completion of the task by distributing the workload among multiple robots. The subsequent methods are associated with task decomposition. In our case, we are considering the exploration of an unknown environment as the task to be accomplished.

An approximate cellular decomposition is a representation of the free space using a fine grid of cells that are all the same size and shape, also known as binary cell decomposition. In semi-approximate cellular decomposition [65, 235], the cells are of fixed width but can have varying top and bottom shapes. The exact cellular decomposition is also known as trapezoidal decomposition [236], where robots move back and forth to cover the space completely.

Table 2.4: Coordination Characteristics

Coordination Attributes	Sub Attributes	Related paper
Task decomposition	Local frontier identification	[5, 8, 19, 38, 62, 75, 83, 88, 89, 92, 110–113, 116, 118, 119, 125, 127, 135, 137, 143, 145, 148, 152–155, 158, 159, 162–164, 167, 179, 186, 189, 190, 237]
	Voronoi based	[50, 114, 122, 139, 140, 142, 147, 180]
	Bidding based	[129, 131, 136, 157, 161]
Task allocation	Depth-First search	[5, 151, 160, 180, 238]
	Breath-First search	[180, 238–241]
	Greedy based	[5, 19, 29, 88, 110, 190]
	Bidding based	[113, 118, 242]
	Hungarian method	[111, 114]
	Heuristic based	[62, 117, 129–131, 134, 135, 137, 138, 153, 154, 157, 161, 179, 189]
	Rendezvous based	[143, 148, 149, 152, 155, 158, 163, 164]
Task exploration	Frontier based	[5, 8, 11, 29, 38, 58, 62, 63, 74, 75, 88, 91, 92, 110–113, 115, 117–127, 129–131, 143, 145, 148, 150, 153, 155, 161–163, 170, 177, 179, 190]
	Segment based	[8, 52, 114, 158, 173]
	Target based	[91, 110, 135, 178]
	Role based	[29, 152, 155, 158, 164]
Task termination	No visible frontiers	[5, 8, 11, 19, 29, 29, 38, 58, 62, 63, 74, 75, 88, 89, 91, 92, 110–113, 115, 117–127, 129–131, 140, 143, 145, 148, 150, 152, 153, 155, 158, 161–164, 170, 170, 177, 179]
	No bids available	[129, 131, 131, 136, 157, 161]
	All Voronoi cells covered	[50, 114, 121, 122, 139, 140, 142, 147, 180]
	Return to BS	[29, 88, 149]

Exploring a map involves gathering new information by moving through an environment. There are various methods to accomplish this, starting with *local frontier detection*. We already know that the frontier is the boundary between the explored and unexplored areas. Identifying these frontiers is crucial for enabling robots to explore and map their operational environment fully. Various approaches can be employed to identify local frontiers in MRS. One such approach is occupancy grid mapping, The robots can then use this map to identify unexplored areas and navigate toward them to explore them. Another approach involves using visual sensors and cameras to detect unexplored areas. The robots can capture images using these sensors and use them to identify areas that have not been visited and then navigate toward these areas to explore them [243].

One of the techniques used in this approach is Distance-based frontier identification [94, 98], which is widely used in multi-robot systems. This technique utilizes the distance to different points (robot's location and centroid of the frontier) in the environment to identify potential frontiers that can be further explored using path-planning algorithms. According to [5], robots detect closed frontiers and move towards them, while [92] considers the trade-off between the distance cost of reaching a target and its utility of that target. Future [118] approach uses a metric to determine the value of a frontier, taking into account the information gained and the travel cost of observing from the selected pose. According to [114], cell decomposition may not be effective in larger environments. To address this issue, a new approach based on area partitioning, known as Voronoi diagrams, was introduced.

Voronoi diagrams have numerous applications due to their useful properties. Voronoi-based frontier identification provides a dependable approach for multi-robot systems to navigate unfamiliar surroundings. The coordination of team members can be achieved through Voronoi diagrams, which divide the environment into various segments. Each unknown segment acts as a Voronoi region [244]. Using the Voronoi diagram, this technique divides the space into regions based on the proximity to the nearest point. The method identifies frontiers using sensor data and robot locations by selecting the edges of the closest regions. Path-planning algorithms are then used to explore these frontiers effectively. Numerous studies, such as [245, 246], have investigated the applications of

Voronoi diagrams. The Voronoi graph is widely used in various fields such as path planning, computer graphics, biology, VLSI design [247], and sensor networks. In [248], they used Voronoi diagrams for developing the roadway-based approach.

The strategy involves having each teammate robot move toward the frontier cell that's closest to them. However, this approach doesn't take into account the placement of all the team members throughout the environment. To address this, a *bidding-based algorithm* was introduced to enhance the frontier-based method by balancing the costs of reaching the frontier cells with their utilities [75]. Bidding-based approaches have been the subject of various studies exploring decentralized control strategies among robots competing for tasks or resources. These approaches aim to distribute functions and resources efficiently and equitably among the robots.

In [249] providers, the multiple *Auction-based methods* that contain the Parallel Auction, G-Prim [250], Sequential Auction [251], Repeated Parallel Auction [252], Repeated G-Prim [252], and Combinatorial Auction [253]. Implementing these strategies can be challenging and require a robust communication infrastructure. The effectiveness of these approaches can be impacted by factors such as the accuracy of robots' information, bid quality, and communication range. It is crucial to carefully tailor bidding-based approaches to meet the application's demands and system performance objectives, considering the number of robots, task requirements, communication infrastructure, and available resources. Multiple studies have explored this approach differently, including [165, 254]. According to [255], an MRS task is assigned in bidding-based approaches by maximizing cell utility and minimizing cost.

2.8.2 Task Allocation

Task allocation refers to the process of assigning particular subtasks to individual robots based on their skills, abilities, and the requirements of the sub-task. The methods that are used in this process are associated with task allocation.

An efficient way to explore and map an environment is to use a Multi-Robot System (MRS) that employs *Depth-First Search (DFS) or Breadth-First Search (BFS) algo-*

rithms. In these methods, each robot is given a unique ID and starting location (vertex), and follows either a DFS or BFS approach while maintaining a stack of vertices to be visited. In order to avoid collisions and ensure that they do not visit the same vertex simultaneously, the robots communicate with each other. However, careful coordination and planning are necessary to overcome challenges such as collision avoidance. There are several algorithms proposed for exploring unknown terrains using a DFS approach. One of the algorithms, presented in [180], involves dividing the terrain into smaller regions, constructing a tree with each region as a node, and using a centralized DFS algorithm for robot exploration. Another algorithm proposed in [113] divides the obstacle-free area into cells and assigns a single robot to explore each cell using DFS. Moreover, in [77], each robot follows a DFS-like procedure to create spanning trees required for the exploration algorithm when multiple robots explore an area divided into cells. Finally, [4] proposes an improved DFS approach by introducing a new method that employs multiple DFS searches. However, BFS is an efficient method for finding optimal paths and systematically exploring an environment. In [108], BFS was used to calculate the Flooding Distance. The BALANCE algorithm, described in [239], uses the lower bound values of DFS and BFS. Moreover, [240] introduced two algorithms, the Wave Front Detector (WFD) and Fast Frontier Detector (FFD), which are based on BFS.

Greedy-based task allocation is a popular approach for assigning robot tasks in a MRS. In this method, each robot maintains a priority queue of available tasks, sorted based on a greedy criterion, such as distance or remaining time to complete the task. The robot selects and performs the task with the highest priority from its queue. Despite its efficiency and simplicity, this approach may not always result in optimal task allocation. The process is repeated until all tasks are completed. When choosing the greedy criterion, carefully considering the system's goals and requirements is important. One such approach, described in [256], is the Nearest-Based Frontier Allocation Strategy, where robots move toward the closest frontier on the map. In [257], each robot calculates the distance to all tasks available for auction. Research in [238] has shown that for single-robot systems, the greedy approach of guiding the robot to the closest frontier is reasonably effective compared to the optimal solution.

The Kuhn-Munkres algorithm, also known as *the Hungarian method*, is used to solve the MRS assignment problem. The assignment problem involves optimally assigning tasks to a group of robots. The Hungarian method is a combinatorial optimization algorithm that minimizes the cost of the assignment to obtain an optimal solution. It can handle multiple tasks and robots and address constraints. However, it can be computationally expensive, unsuitable for real-time applications, and not handle uncertainty well. In [64], a Decentralized Hungarian Based Algorithm (DHBA) was developed to address these limitations. Additionally, [8] divided the map into segments using a Voronoi graph (VG) [258] to represent individual rooms or areas of a large corridor. These segments were then assigned to robots using the Hungarian method.

Heuristic-based task allocation method allows robots to select tasks based on predefined rules or heuristics rather than an optimal solution, providing coordination of the robots' actions and allowing for workload balance and adaptation to changing environmental conditions. However, it may not guarantee the optimal global solution, fair allocation among robots, or consider the long-term impact on the mission. Different heuristics have been studied for task allocation in previous research. For instance, [92] built a global map using a heuristic that combined distance cost and utility for target selection. Similarly, [110] developed a utility-based heuristic and introduced a "belief measure" for allocating the next target to the robot. In [174], a cost-utility-based approach was used where cost was defined as the distance between the robot and the target, and utility was based on communication constraints. [259] defined a cost-utility approach, where cost was the distance between the robot's location and the frontier's centroid, and the area covered by the frontier defined the utility. [132] combined information gain and distance cost of the frontiers in their heuristic. Finally, [139] developed a new function based on distance cost to select the target that provided the maximum information gain.

The potential field is an AI approach that can be utilized for task allocation in MRS. The method works based on the principle that each task generates a corresponding potential field, and robots are attracted to tasks with the highest potential. This decentralized coordination strategy distributes the workload among robots and allows for adaptation to environmental changes. However, the potential field method may not always provide the optimal global solution and may not adequately address real-time constraints and uncertainty. In [120], the authors use a decentralized frontier-based exploration approach to evaluate the cost-utility ratio of navigating towards target waypoints, utilizing the A* algorithm. They implement the potential field method to control the robots' motion to avoid obstacles. On the other hand, [135] employs a behavior-based approach called social potential fields to determine the robots' heading and move them toward unexplored areas. The system is optimized by adjusting each robot's angle and velocity. In [115], the potential field method is used for decentralized robot control, and a frontier-based approach is used to escape potential field minima.

Rendezvous points-based task allocation is a method used in MRS to coordinate the actions of robots by agreeing on a common location, called a rendezvous point, where they can meet and exchange information about the available tasks. The main aim of this technique is to enable efficient and rapid allocation of functions among the robots. It allows for decentralized coordination of the robots' actions, facilitating workload balancing and adaptation to changing environmental conditions. However, it depends on a reliable communication infrastructure and may not be suitable for real-time constraints. Robots are divided into explorers and relays in the system proposed by [46]. Explorers are tasked with discovering new areas of the environment while relaying the shuttle back and forth to transmit the data collected by the explorers to the central station. The explorers and relays coordinate by selecting appropriate rendezvous points. However, it is important to note that periodic meetings at these points may limit the exploration process.

2.8.3 Task Exploration

Task exploration identifies and evaluates possible subtasks and determines their relative importance and dependencies. This involves analyzing the overall task and determining which subtasks are necessary to achieve the goal and how they can be best accomplished. The subsequent methods are associated with task exploration.

Frontier-based exploration [129, 130, 179] in MRS is a method where robots explore an unknown environment by identifying and visiting the frontiers, or the boundaries between known and unknown areas. An approach for this is the Wavefront Frontier Detector (WFD) algorithm [124]. The main advantage of the WFD algorithm over the original is that it only scans the known regions of the occupancy grid instead of scanning the entire grid every time the algorithm runs. This makes the process much more efficient and faster. In [260], they proposed an extended frontier-based exploration method that relies on a new bidding function called ”span.” The ”span” parameter represents the distance between the frontier cell and the other robot.

Segment-based exploration [52, 158] in MRS is a method where robots explore an unknown environment by dividing it into smaller segments and assigning each segment to a robot or a group of robots for exploration. The algorithm in [261] uses range finders to recognize the environment. They segment the map. That approach was described in [262] for efficient robot exploration. To start segmentation, robots load an environment map into memory. Machines then segmented the space. Sharing section positions. Guess which geographic region is under-explored. One robot searches the less-explored region. Choosing the nearest robot to aimed parts.

Role-based exploration [29, 149] in MRS is a method where robots are assigned specific roles and responsibilities based on their capabilities and resources and coordinate their actions to explore an unknown environment. In their study [248], the authors distinguish between Behavior-based and Role-based systems for MRS. It is also known as the relays-explorers; MRS is a method where one robot, the leader, is assigned to plan and coordinate the exploration. In contrast, the other robots, the explorers, gather information and map the environment. It allows for efficient exploration of unknown environments by utilizing the robots’ capabilities and resources. [149] presents a different approach to using rendezvous points in an unknown environment. This method involves role-based exploration, where some robots (referred to as ”relays”) relay information between other robots and a central command center while others (called ”explorers”) continue to explore the environment using frontier exploration. When an explorer and a relay meet, they share information about the environment. The explorer chooses the next rendezvous point by placing it deep into its next choice of frontier while ensuring that the point has a strong communication range.

Efficient exploration of unknown environments is made possible through effective utilization of the robots' capabilities and resources. This allows for a balanced workload distribution among the robots and adaptability to changing environmental conditions. However, it is important to have a good communication infrastructure and a robust localization system to ensure success. Real-time constraints and uncertainty may pose challenges in its implementation.

2.8.4 Task Termination

Task termination, conversely, refers to ending or completing a task assigned to a group of robots. This can include shutting down the robots, releasing them for other tasks, or returning them to a home base. Factors such as task completion, detecting an error or failure, or changing operating conditions can trigger task termination. The goal of task termination is to ensure that the robots stop working as soon as the task is completed, to avoid unnecessary work, and to save energy. It is also possible for some robots to reach their base station after completing all their assigned tasks.

Initially, most researchers studying multi-robot exploration utilized frontier-based methodologies as described in [5, 263]. The necessary components for conducting decentralized exploration are outlined, highlighting robot coordination's importance in efficiently exploring unknown environments. In [92], the focus shifts to global exploration, with an emphasis on task allocation and evaluating the suitability of a new target pose for an individual robot. Other approaches, such as the one proposed in [114, 139], use scenario segmentation and Voronoi diagrams to divide the environment into zones and compute a function that assesses the cost of exploring a particular area for a given robot. The Hungarian Algorithm is then used to assign a goal area for each robot. Combining a centralized approach and the Hungarian Method as a task allocator in structured environments like offices can yield positive results, as demonstrated in [111].

The future approach to minimize repetitive visits involved segmenting the partial map into separate rooms and prioritizing unexplored cells within the same area as the robot's current location. To accomplish this segmentation, the method outlined in [264] was implemented, which entailed dividing map regions at local minima or critical points in the Voronoi diagram of the map's unoccupied space. While effectively extracting the environment's structure from grid maps, this algorithm could result in numerous partitions, particularly in long corridors. To tackle this issue, [8] enhanced the algorithm by limiting critical points to those with a degree of 2 and adjacent to a junction node in the Voronoi diagram. This modification significantly reduced the number of segments and allowed for coordination among multiple autonomous exploring robots. The majority of research in this area is centered on the Voronoi-based segmentation approach, as demonstrated in Table 2.4.

A fleet of n identical robots (R1, R2,..., Rn) is exploring an unexplored region, with each robot equipped with sensing, localization, mapping, and short-range communication capabilities. A simple coordination algorithm must be developed to ensure the reliable and efficient exploration of the entire environment. To guarantee mission success even in case of robot shutdowns or communication link failures, a fully decentralized algorithm for mission coordination has been devised for achieving reliability. To achieve time and energy efficiency, a bidding mechanism is used to select the best movement from the bids submitted by a set of robots, as described in [136]. A simple, fully distributed coordination algorithm presented in [242] is used, where all robots make their movement decisions asynchronously based on a bidding mechanism. The robot selects the best bid from competing robots and sets the winning bid's destination as the next stop. The robot executes the following three operations sequentially: (i) sensing and mapping, (ii) bidding, and (iii) traveling.

Simulations are the primary method used to investigate the behavior of MRS in complex or large environments with a large number of robots. Simulations offer advantages such as cost-effectiveness, controllability, scalability, and safety, making them a valuable tool for optimizing the performance of MRS, evaluating different configurations and scenarios, and minimizing the risk of errors and accidents. The papers cited, including [103, 109, 265], represent only a small subset of research studies focused on simulations of MRS.

In addition to simulations, some researchers also conduct real-time experiments to observe and analyze the behavior of robots as they interact with their environment and each other. This information can be leveraged to optimize system performance, identify and rectify any issues or errors, and improve overall efficiency. Real-time experimentation also allows for system robustness and scalability evaluation, which are important factors for real-world deployment. The papers cited, including [74, 118], represent only a small portion of research studies focused on real-time experiments in multi-robot systems. It's worth noting that some studies use both simulation and real-time experimentation, and the papers cited, including [5, 50, 91, 138], represent only a fraction of the research in this area.

2.9 RESEARCH GAPS

The literature survey conducted in this chapter unearthed research gaps related to multi-robot unknown terrain coverage and area exploration under communication constraints. Specifically, we identified the following:

- In recent years, researchers embarked on a mission to tackle the intricate challenge of limited-range communication [46, 107, 266]. Simultaneously, there is no dearth of literature assuming that the communication network is always available. These research works can be adapted to function within the constraints of restricted communication environments. However, as the communication range shrinks, the performance of these approaches gets compromised. The consequence is a slower completion of the coverage task, along with the increase in redundant coverage. This redundancy is a direct result of robots being unable to share vital information, such as an evolving map of the environment they have covered so far. Hence, it becomes imperative to delve into the impact of communication range restrictions on these approaches and evaluate their effectiveness in real-world scenarios.

- Sometimes, when gathering data, there may be time constraints, requiring the information to reach the base station within some specified time [50, 89, 267]. Maintaining continuous connectivity with the base station is not always feasible due to various factors like issues with wireless connectivity over the last mile, battery depletion, or ongoing tasks. Such situations often occur in scenarios like unknown area exploration, search and rescue missions, or disasters, where ground or Unmanned Aerial Vehicles (UAVs) navigating may become disconnected from the network and thus unable to communicate with their command center. Therefore, it becomes crucial to provide temporary last-mile connectivity to some wirelessly isolated robots.

In Chapters 3, 4, and 5 of this thesis, we have addressed these research gaps.

Chapter 3

Multi-Robot Online Terrain Coverage under Communication Range Restrictions – An Empirical Study

3.1 Introduction

The use of autonomous mobile robots in real-world applications [65, 151] has become a more feasible and enticing choice due to improvements in the capabilities of these machines and a reduction in their costs. Online terrain coverage (OTC) is particularly interesting for applications that range from mundane and repetitive tasks like harvesting and insecticide spraying to potentially life-threatening missions like hazardous substance detection and battlefield surveillance for landmine detection, amongst others. Some examples of these applications include harvesting, spraying, hazardous substance detection, battlefield surveillance for land mine detection, and other applications. The urgency and seriousness of the situation are the primary factors in determining the specific criteria that must be met for these applications. However, the overarching goal of these applications is to achieve complete coverage of the environment, leaving no part of the surrounding environment that can be navigated uncovered in the process of accomplishing this goal [67]. As the terrain map is not currently available, it is impossible to calculate the robot's path across the entire landscape. Instead, the movement of the robot team determines as the mission continues and the map begins to take shape. Many different heuristics for decision-making have been suggested [66, 151] for use with an SRS. Despite this, MRS complicates the coordination process. This relates to the question of what information and how it should be shared with the other team members. As the map unfolds, this coordination is required to assign coverage tasks to individual robots. This is done to improve important measurements like the time it takes to finish coverage, the total distance traveled, the total energy used, and so on.

Before we begin our discussion of some of the methods developed in recent years, it is important first to distinguish between terrain coverage and terrain exploration tasks. Both of these tasks are necessary to understand the terrain fully. While terrain coverage requires the robots to traverse the unknown terrain in a shorter time, exploration requires the robots to construct a quality global map of the unknown region. The robots do not have to travel throughout the region to complete the exploration process; once a region has been scanned, the robots head in the direction of unexplored territories. Despite being two distinct applications, there is a significant overlap between robot activities. These activities are:

- The robots cannot access a map of the area before the mission begins.
- An approach called approximate cell decomposition is utilized to break up the landscape into grid cells of consistent dimensions. Although this is not the only technique for terrain decomposition, we have focused solely on frontier exploration methods for this discussion.
- Robots need to possess the capability of self-localization. Without a global map,

techniques such as simultaneous localization and mapping (SLAM) [268] are utilized to construct the map.

- The robots determine their next action by consulting the map data produced locally by their sensors or provided to them by their fellow robots. In this sense, the robots combine the valuable map information from other robots to navigate effectively.

- The robots can communicate with each other to exchange information about the global map at that particular time. This also makes it possible to synchronize the robots' actions. They may also share additional information, such as their pose, the current task assigned to them, the status of its completion, and their current health status.

One important difference between the two applications is that, in the exploration process, the robots have a sensing radius higher than one cell. For the terrain coverage task, the robots can sense only the eight surrounding cells from the robot's position. It is crucial to discuss recent state-of-the-art multi-robot exploration approaches before accepting this argument as a fact. Brick&Mortar [4], RAPID [85], and BMI [225] are three different ant-based approaches in which the environment is divided into a grid of square cells. The size of the cell depends on two things: (a) the distance measured by the range sensors (Rs) and (b) the communication range (Cr) of the agents.

Communication within MRS makes timely coordination possible. As a result, finishing the OTC task on time is feasible [24]. It is a reasonable assumption that communication is constantly occurring. Under the assumption of widespread communication, we developed both centralized [79] and decentralized [45] coordination algorithms for OTC. However, global communication is a strong assumption in the real world because the robots may travel to far-off regions while performing the coverage task. Some robots may become disconnected from the rest of the team. It is especially true for online exploration [50, 119], and coverage tasks [4, 269], in which the robot team may become dispersed while carrying out the mission, resulting in a fragmented multi-robot network.

As a result, the algorithms used for OTC coordination must be able to deal with communication interruptions and use the limited and variable communication available for a limited time.

Researchers have addressed the issue of limited-range communication in recent years [60]. On the other hand, written material is abundant, so it is assumed that a long-range communication channel is always accessible. These strategies can be adapted with additional work to function effectively within the constrained communication environment. However, the performance of these approaches is hindered by the limitations placed on the communication range. This means the coverage task may take longer because of redundant coverage. As a result, robots may be unable to share important information, like a map of the environment they have already covered. As a result, it is of the utmost importance to investigate the impact of communication range limitations on selecting these strategies and evaluating how well they work in practice settings. In this chapter, five different approaches to multi-robot online terrain coverage are set. These approaches assume global communication and represent the current state of the art (SoTA).

In addition, the presumption of global communication is refuted. The effect of imposing communication range restrictions on these methods is studied by adjusting the number of robots and the communication range in three different maps with varying degrees of complexity. This is done to investigate these restrictions' influence on these methods. Because of this, it is possible to investigate these restrictions' effects. Experiments have been conducted in a laboratory setting under strict control, using simulation software and various mobile robots as part of the testing process. To the best of our knowledge, there has not been a previous attempt made to reproduce SoTA OTC approaches by utilizing a communication model that is less reliable than the one we are using in this research. However, we are open to the possibility that such an attempt may have occurred.

However, we are open to the possibility that such an attempt might have been made.

3.2 PROBLEM DEFINITION

This chapter considers the OTC task using a team of homogeneous mobile robots under communication range restrictions. The same problem is called Online Multi-Robot Coverage Path Planning (MR-CPP). The terrain to be covered is decomposed into grid cells using approximate cell decomposition described in [235]. Formally, the MR-CPP problem can be stated as follows:

Given a bounded environment E consisting of a finite set C of free cells, such that each free cell $c_i \in c$ *is in one connected component, and a set R of n homogeneous mobile robots, find trajectories for each robot* $r_i \in R$*, such that each cell* $c_i \in c$ *is traversed by at least one robot* $r_i \in R$ *in finite time while minimizing repeated coverage.*

The following assumptions are considered for solving the MR-CPP problem:

- Initially, the robots start from arbitrary locations.
- Each robot in the team can detect whether its surrounding cells are occupied with obstacles. In other words, if the robot is present in some cell $c_x \in C$, the sensing range of robots is restricted to eight surrounding cells of c_x.
- The robots can communicate with their peers within a re-specified communication range, i.e., their communication range is restricted.

3.3 ONLINE TERRAIN COVERAGE STRATEGIES

We choose decentralized algorithms because this chapter evaluates the impact of imposing communication range restrictions on various OTC strategies involving multiple robots. To do this, we compared the effects of these restrictions. The area that needs to be traversed is referred to as being bound.

1. Backtracking Spiral Approach – Cooperative Multi-Robot (BSA-CM) [1]
2. Spiraling and Selective Backtracking (SSB) [2]

3. Boustrophedon and Backtracking mechanism (BoB) [3]

4. Multiple Depth First Search (MDFS) [4]

5. Brick and Mortar (BnM) [4]

We chose these strategies because they are distinctive from one another and have received much attention in the relevant research. In the following subsections, we explain how each algorithm works in more in-depth.

3.3.1 BSA-CM

The Backtracking Spiral Algorithm, commonly called the *BSA*, applies exclusively to scenarios involving a single robot [6]. The assumption is made that the environment in which the robot operates can be partitioned into grid cells, with each cell determined by the size of the robot's footprint, to ascertain occupancy. It is assumed that the free space constitutes a single connected component. The algorithm concurrently partitions available space into multiple organized spiral trajectories. The algorithm guarantees the utilization of every available cell in the grid map.

The robot is able to successfully navigate free space due to the structured spiral trajectories, which it uses while it is also incrementally constructing the simple regions. At first glance, each cell in the grid appears to be a puzzle. Following the robot's passage through a given cell, that cell is then flagged as a ”virtual obstacle,” meaning that the robot will not be able to access it again as it travels along the spiral path. The remaining cells that contain obstacles are referred to as ”real obstacles.”

Following a trajectory in the shape of a spiral results in the formation of paths similar to concentric rings. As a result of the formations, there is an unbroken path that can be followed from the region's outskirts to the center of the spiral's termination. At first, the robots are positioned near real or simulated obstructions. It is important to position the obstacle so that it is visible from the Reference Lateral Side (RLS) of the robot. The relative direction in which obstacles should be referred throughout the spiral filling technique is indicated by RLS. RLS is predetermined and cannot be altered. The antipode of RLS is identified by the opposite lateral side, OLS. The robots carry out a reactive execution of the BSA Coverage Algorithm, which is referenced as 3.1, to generate spiral trajectories. The following is a list of some of the essential features that this algorithm possesses:

- When executing the algorithm's instructions, both virtual and physical obstacles are treated in the same manner.
- Upon the termination of the algorithm, the cells that have been marked as virtual obstacles indicate the presence of unobstructed covered cells.
- During the algorithm's execution, the robot identifies and records backtracking points along the spiral path, which indicate alternative routes.
- Upon the termination of the algorithm, the robot proceeds to select and follow the shortest path to reach the closest backtracking point. From this juncture onward, the robot resumes its spiral descent.
- The algorithm terminates when the robot has exhausted all available backtracking points in its list. This observation also suggests that there are no remaining cells that are free and uncovered.

Algorithm 3.1 BSA Coverage [6]

```
if obstacles in all directions then
    stop motion because end spiral point detected;
else if no obstacle in RLS then
    turn to RLS and start forward motion;
else if obstacle in front then
    turn to OLS;
else
    move forward;
end if
```

The process of executing the fundamental *BSA* algorithm is shown in Figure 3.1. Black lines represent the three spirals. The robotic entity, represented by the maroon

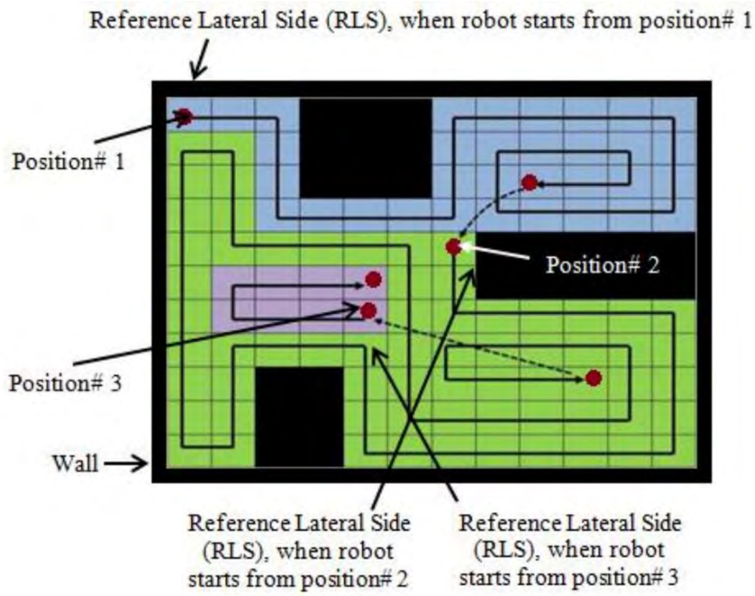


Figure 3.1: Basic BSA Algorithm in Execution [6]

circular shape, consistently initiates its trajectory from a cellular unit that is obstructed on one of its sides. When the robot is located at position 3, the cells on its RLS that are colored green have already been covered and are considered as a virtual wall. This phenomenon occurs due to the cells being regarded as a barrier. The algorithm's simplicity and its requirement for the robots to start their journey adjacent to a wall on the lateral side of the reference are factors that limit its effectiveness despite its ability to traverse the entire topography. The algorithm exhibits potential for improvement.

In this approach, the robots follow a spiral trajectory to cover the unknown environment. The uncovered cells touching the trajectory of the robots are marked as backtracking points (BPs). The robots spiral inwards until they reach the spiral center. A covered cell is virtually marked as an obstacle, and the robots are not allowed to navigate through it while spiraling. Once the robots complete their spiral path, they select a BP to spiral on an alternate path. Using auctions, the most suitable BP is selected. Each robot submits its bid for all BPs. The robot's bid is calculated as the sum of the length of the spiral path it is yet to cover plus the length of the path the robot will have to follow to reach the BP being auctioned. The robot moves to the nearest unvisited BP if it does not win an auction because some other robot may have won this BP. When there are no more BPs left to be visited, the algorithm terminates. This approach's main limitation is that the robots must start from a near-wall position, or else many cells will be marked as BP points.

Consequently, several auctions will be initiated to select alternate paths after any robot completes its assigned coverage task. Also, this algorithm is sensitive to communication breakdown. Under communication range restrictions, the algorithm's performance severely deteriorates as fewer robots send their bid for some BPs. The worst case is when no robot participates in auctions of the BPs starting point of longer paths. Such paths may be redundantly traversed if the robots never meet and exchange coverage information, thereby increasing redundant coverage.

3.3.2 SSB

SSB and *BSA-CM* share the same mechanism of spiraling inwards while labeling the cells as BPs. However, SSB selectively marks the cells with the BP marker. Additionally, once each robot has finished its coverage task, the BPs are put up for auction to choose alternative paths.

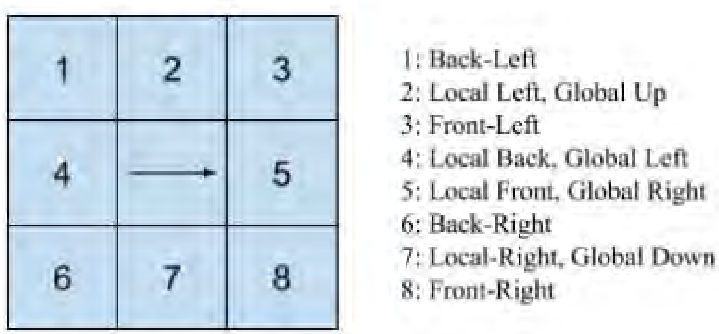


Figure 3.2: The Robot Moved Right (arrow) and Utilized Algorithm 3.2 Terminology [2]

Like BSA-CM, SSB also works by spiraling inwards while marking the cells as BPs. However, it selectively marks the cells as a BP. Further, these BPs are auctioned for selecting alternate paths after any robot accomplishes its coverage task. The robot moves to the nearest unvisited BP if it does not win any auction. When there are no more BPs

Algorithm 3.2 Marking Backtrack Points [2]

Let F represent the Free
Let O represent the Obstacle
Let BPs represent the Backtrack Points

```
if local-right is F then
    if local-front ∨ front-right ∨ back-right is O then
        mark local-right as BPs;
    end if
end if
if local left is F then
    if local-front ∨ front-left ∨ back-left is O then
        mark local-left as BPs;
    end if
end if
```

left to be visited, the algorithm terminates. The main advantage of Algorithm 3.2 is that it generates fewer BPs. The terminology used in this algorithm is presented in Figure 3.2. As a result, the robots have to participate in fewer auctions and, therefore, have lesser reliance on global communication. However, this algorithm also gets affected by communication disruptions, resulting in redundant coverage, but it is less than BSA-CM. On the other hand, this algorithm is also vulnerable to communication problems, which can lead to duplicate coverage, though this is less likely to happen than with *BSA-CM*. Please find the link below to access the demonstration of our SSB approach: [270].

3.3.3 BoB

In this approach, Boustrophedon motion is followed by the robots to cover the terrain, succeeded by backtracking. The covered cells are marked as virtual obstacles. The robot executing the algorithm initiates backtracking when it arrives in a cell surrounded by obstacles (real or virtual) from all sides. The cells are marked as BPs only when the robot initiates backtracking. Only the corner most cells of the covered and uncovered region are marked as BP. The robots select the nearest BP to traverse. When all the robots arrive at a dead end and no BPs are left to be traversed, the algorithm terminates. This algorithm has much less reliance on communication than *BSA-CM* and *SSB* because it only marks a few cells as BPs. A step-by-step Boustrophedon motion algorithm may include the following steps:

1. Initialize the robot in the starting position and set the direction of movement.
2. Move forward in the set direction until the end of the row or area is reached.
3. Detect the end of the row or area using sensors or other means.
4. Reverse the direction of movement.
5. Move forward in the new direction until the end of the next row or area is reached.
6. Repeat steps 3-5 until the entire area has been covered.
7. Adjust the movement based on sensor data or other inputs to avoid obstacles or make other necessary corrections.
8. Stop the robot or machine.

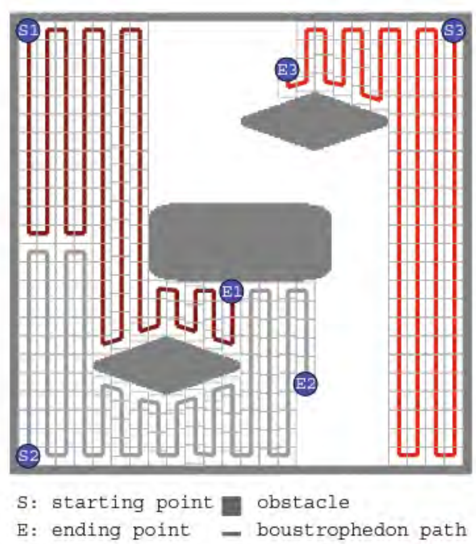


Figure 3.3: Boustrophedon Paths Created by Robots [3]

This algorithm is not capable of sufficiently dispersing the robots. As a result, network breakdown events due to robots moving far and out of each other's range are less, even when the communication range is small. Simultaneously, the algorithm does not take advantage of long-range communication (if available).

3.3.4 MDFS

The robots construct their exploration trees individually, traversing them using depth-first search Figure 3.4 while the cells are marked as having been explored. Further, the

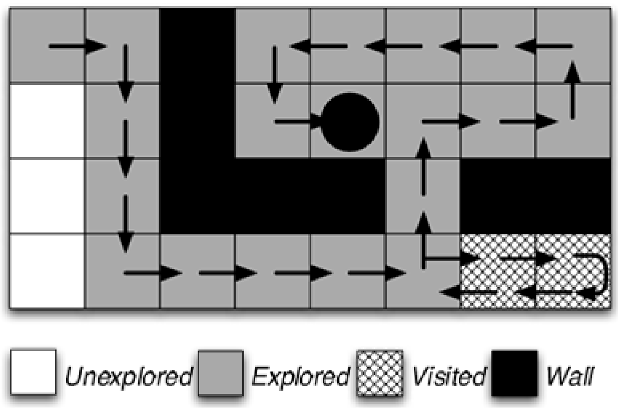


Figure 3.4: Different branches in a Depth First Search Exploration [4]

explored cells are marked as visited when the robots traverse the tree upwards. Visited cells are not traversed again. On encountering their peers' explored cells, the robots traverse that path of explored cells to investigate any new neighboring cells on that path. If a new cell is discovered, this cell is marked as a fresh tree's root node. When all the cells are surrounded by visited or wall cells, the algorithm terminates. There is no explicit communication between the robots.

Algorithm 3.3 is inspired by the behavior of ants leaving pheromones. It works with the premise that the robots can leave tags in the terrain that can be read and updated by the robots themselves or other robots. They treat the terrain as a shared memory for communication. It is non-trivial to practically tag the terrain for communication purposes in the real world as the tags themselves are prone to failure. For our purpose in this chapter, the robots can read or update the grid decomposed terrain map (local to the robot) while traversing their trees downwards and upwards. The robots explicitly communicate with other robots when they are in each other's communication range for sharing coverage information.

Algorithm 3.3 Multiple Depth First Search [4]

Let U_c represent the Unexplored_cell
Let C_c represent the Current_cell
Let P_c represent the Parent_cell
Let E_cs represent the Explored_cell
Let V_c represent the Visited_cell

```
if the C_c is U_c then
    mark it as explored;
    annotate the cell with your ID and the direction of the previous cell (P_c);
end if
if there are U_cs around then
    go to one of them randomly;
else if the C_c is marked with your ID then
    mark it as visited;
    go to the P_c;
else
    randomly goes to one of the E_cs, avoiding selecting the cell from which you are
    coming unless it is the only candidate.;
end if
```

3.3.5 BnM

In the same way, as in *MDFS*, the terminology of the wall visited, explored, and unexplored cells are utilized in BnM. Wall cells and cells that have already been visited are examples of inaccessible cells, whereas explored and unexplored cells are examples of accessible cells. The algorithm guarantees success for any robots that implement it. They continually mark their current cells as visited if the cell does not block the path between any two accessible cells, and then they move to any of the accessible cells that are nearby in random order. This process continues until all of the cells in the area have been visited. Because of this, the algorithm causes the blocks of inaccessible cells to become dense. It continues to do so until the entire topography is transformed into a single large block of inaccessible cells. The algorithm's execution is complete when it reaches this stage.

It is necessary to assume that the environment has been mapped out as a grid and that an intelligent tag or device has been pre-assigned to each grid cell in advance. The robots cannot communicate or share any information regarding their map coverage with the other robots. Instead, they read and update the state of each cell that they cover independently. A robot considers the currently occupied cell to have been visited if it does not obstruct movement between the uncovered two cells. This includes cells that are adjacent to one another. After that, it looks for a vacant nearby cell and wanders around until it finds one. The robot team is progressing toward converting the entire landscape into blocks of cells that can be explored, walled off, or made inaccessible in this manner. After each grid cell's state has been updated to reflect that the algorithm has been "visited," the algorithm will be finished. Regarding inter-robot communication and restrictions on the communication range, we have handled *BnM* comparable to how we have handled *MDFS*. Specifically, we have taken inspiration from how we have handled *MDFS*.

3.4 EXPERIMENTAL TEST-BED SETUP

In this chapter, we conducted extensive experiments both in simulation and on a physical multi-robot test-bed. A detailed description of the same is presented in the following sub-sections.

3.4.1 Simulation Setup

We used the ROS-based Gazebo simulator with a varying team size of two to eight Pioneer2 DX robots. The communication range of the robots is kept as a configurable parameter. All the robots are equipped with a 2D Lidar sensor, i.e., Sick Tim 561, which has a 270-degree field of view, a resolution of 0.33 degrees, a maximum range of 10 meters, and an update rate of 15 Hz. However, we have considered laser scan readings in the maximum range of five meters. The robots are also equipped with an IMU, i.e., ADXL 345, and wheel odometry. The robots' wheel odometry is believed to accumulate errors over time; therefore, in simulation, it is corrupted with Gaussian noise and is a configurable parameter. In our simulation, the robots generate a 2D occupancy grid map of the environment while carrying out OTC using Hector SLAM [271]. Hector SLAM does not utilize odometry and IMU and entirely relies on scan matching based on exact laser scans obtained from the sensor. However, in indoor symmetric environments with long corridors, the 2D point cloud obtained from the laser scanner is distributed uniformly, and the variation in the point cloud is not visible.

As a consequence, Hector SLAM produces an inaccurate robot pose. Furthermore, the map generated by the robots drifted and did not align with the actual map of the environment. Therefore, in this chapter, we have augmented the Hector SLAM and combined odometry and IMU sensor readings to estimate the robot's pose using the Extended Kalman Filter (EKF). We used the odometry and IMU sensor readings for symmetry breaking and aligned the point cloud obtained from the laser scanner with the robot's accurate pose. The grid map is obtained from the point cloud using the scan matching algorithm, which uses the Gauss-Newton method to minimize the error between the alignment of the laser scan and the map. It is assumed that the robots can measure their relative position and orientation in the simulated world.

Inter-robot communication is achieved using the adhoc_communication ROS node [272]. The robots executing the corresponding node establish an ad-hoc network with their peers for exchanging data. The data can be exchanged with several roscores. This ROS node allows unicast, multicast, and broadcast-based communication. The robots communicate with their peers by publishing information on a specified ROS topic subscribed by other robots. Each robot has its multicast group for communication that other robots can join using the join_mc_group service. Information is routed in the multi-robot network using the robots' hostnames. In our simulation, for imposing communication range restrictions on the robots, the ad hoc detailed unification ROS node is modified to allow communication between any two robots only after verifying that they are inside a pre-specified communication range of each other.

3.4.2 Physical Test-Bed Setup

Conducting experiments with the multi-robot system in the real world is non-trivial since we have to address three vital requirements simultaneously, i.e., (a) multi-robot localization and mapping, (b) inter-robot communication and (c) robot control, which also takes care of robust trajectory tracking and collision avoidance. We used three Firebird V

(FB-V) robots in our team of mobile robots. This FB-V is a two-wheel differential drive robot controlled by an AVR (ATMEGA2560) microcontroller. These robots have a 2.4 GHz Xbee module, allowing multi-channel (16) wireless communication. The robots are equipped with a skirt of 12 analog IR proximity sensors and position encoders on both wheels. The remaining section elaborates on how the requirements mentioned above were accomplished.

Multi-robot localization is the most challenging task when dealing with autonomous multi-robot systems. The robots must determine their positions and localize their peers to plan the next move. For our experiments, we have used a roof-mounted camera that acts as a pseudo-GPS system to localize the robots moving under its field of view while tracking fiducial markers, i.e., April Tag-2 [273], that is attached on top of each robot as shown in Figure 3.5. The data obtained from the camera is transmitted to a PC, i.e., a Dell OptiPlex 7040. This PC runs a C program based on the April Tag-2 libraries. The April Tag-2 libraries can calculate the precise 3D pose of a robot relative to the camera. The library uses sophisticated feature extraction algorithms for robustly localizing tags from a single image. A unique number is assigned to each tag, starting from zero. After camera calibration, the camera's intrinsic parameters and the physical size of the printed tags are fed to the April Tag-2 libraries. The library then determines the relative transformation between each tag and the camera. The 3D world coordinates of each tag concerning the camera, which is at coordinate (0, 0, 0), are shown in Figure 3.5.

In our system, tag number zero is stationary and used as an origin. On the other hand, the robots are assigned the rest of the tags. The origin tag is used to localize each robot in the 2D world. The 2D points on the camera's image plane are projected onto the plane containing the tags using simple 3D geometry and tag orientations. We call this the working plane. The X and Y axes of the working plane coincide with those of tag zero, and the Z-axis is perpendicular to the tag. The angle measures each robot's orientation that the Y-axis of their tag subtends to the working planes' X-axis. This orientation, combined with the robot's coordinates in the working plane, determines the robot's pose. The final output of these transformations is shown in Figure 3.5. Each robot maintains

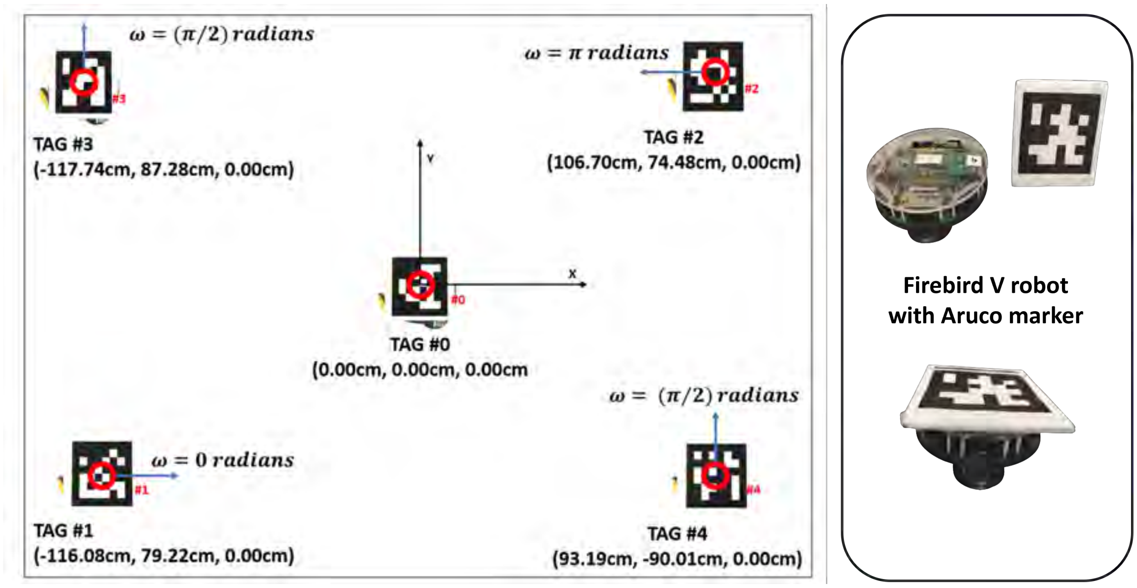


Figure 3.5: Robot's Localization using April Tags – After Transformation

a local instance of the environment map for mapping and coverage. As the robots cover the terrain, they transmit their local map and coverage information to the other robots. Whenever a robot receives information from other robots, it fuses this new information into its local map.

We have allowed the robots to communicate when inside the communication circle to impose communication range restrictions. They can communicate when their communication circles intersect, as shown in Figure 3.6, where only Robot-1 and Robot-2 can communicate with each other but not Robot-3.

For inter-robot communication, a radio communication module, i.e., Digi XBee (ZigBee/IEEE 802.15.4 compliant), is used to share the terrain coverage information along with the map discovered by the robots. This module allows mesh networking and packet rerouting. The XBee is also suitable for swarm applications, as it scales well with its 16-bit addressing capability. It requires significantly low power (a few milliwatts) and offers a range of 100 meters. The XBee chip communicates via TTL-level UART. Nex-Robotics has designed a PCB that acts as an interface between the XBee and the Firebird-V robot. For XBee, Digi provides extensive documentation and many functionalities to implement reliable communication while efficiently supporting automatic retransmissions and ac-

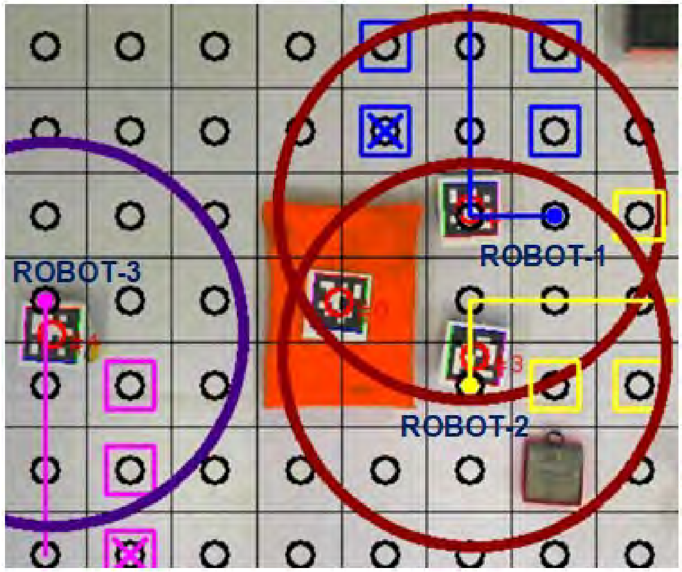


Figure 3.6: Communication Circle of a Robot

knowledgments. We used the Digi XBee ANSI C Library [274] for communication. This library is used for disparate data values that can be packed (for sending) easily into a buffer and unpacked quickly by the receiver. A robot can transfer data packets to other robots in a unicast, multicast, and broadcast manner, allowing individual robots to communicate (relaying and querying map and coverage information) with other robots locally in their communication circle.

Low-level robot control and collision avoidance for terrain coverage are implemented as a hierarchical process with two cascading loops. The outer loop is the planning loop that allocates the cells to each robot and commands a trajectory to the inner loop as an input. The inner loop commands the ground robots to follow the commanded trajectory. The collision avoidance among the robots is integrated into the planning loop. As the robot's size is strictly less than the cell decomposition size, the collision event is averted by not assigning an identical cell to any pair of robots at any time along the trajectory. Note that the assigned trajectory to an inner loop is a function of time for each robot. Therefore, following a trajectory is essential to satisfy both position and temporal constraints. Besides, the performance of trajectory tracking can also degrade in the presence of external disturbances. We implement the trajectory controller described in [275] to achieve robust performance. The problem is formulated as a two-player zero-sum differential game against external disturbances. Once the trajectory is received from the planning loop, the controller gains for tracking are computed for a finite interval of time, and the ground robots precisely track the trajectory. Once the tracking for the finite interval is complete, the planning loop sends the new trajectories for the successive interval.

3.5 RESULTS AND DISCUSSIONS

In this section, We have presented the results obtained through simulation and on a real test platform using numerous robots.

3.5.1 Results

Two different terrain maps were considered for simulation and experiment. The map M1 is a large map of $150\ m^2$ used in the simulation. For the experiments, the terrain of size $108\ ft^2$ (M2), which is grid decomposed, is considered. The two maps are shown in Figure 3.7. The performance metric used for the empirical comparison of algorithms is redundant coverage – which is the total number of times each cell was covered more than once by any robot to complete the OTC task. It is the most dominant metric because all other metrics, like completion time, cumulative distance traveled, etc., are directly dependent on redundant coverage. We have conducted 100 simulation runs for each algorithm in *Large Map (M1)* by varying the communication range with different robot team sizes. Ten experimental runs were conducted on a physical multi-robot test bed with a team size of three robots and varying communication range of 1 ft to 3 ft.

The initial position of the robots is randomly chosen. One factor evident from all the simulations and experiments, irrespective of the OTC algorithm employed, is that with the increase in communication range, the OTC algorithm becomes effective, resulting in lesser redundant coverage. Moreover, after a specific communication range, i.e., critical range, any OTC algorithm's behavior for a particular number of robots starts to emulate the behavior achieved with global communication.

In contrast to the aforementioned, it was observed through empirical research that

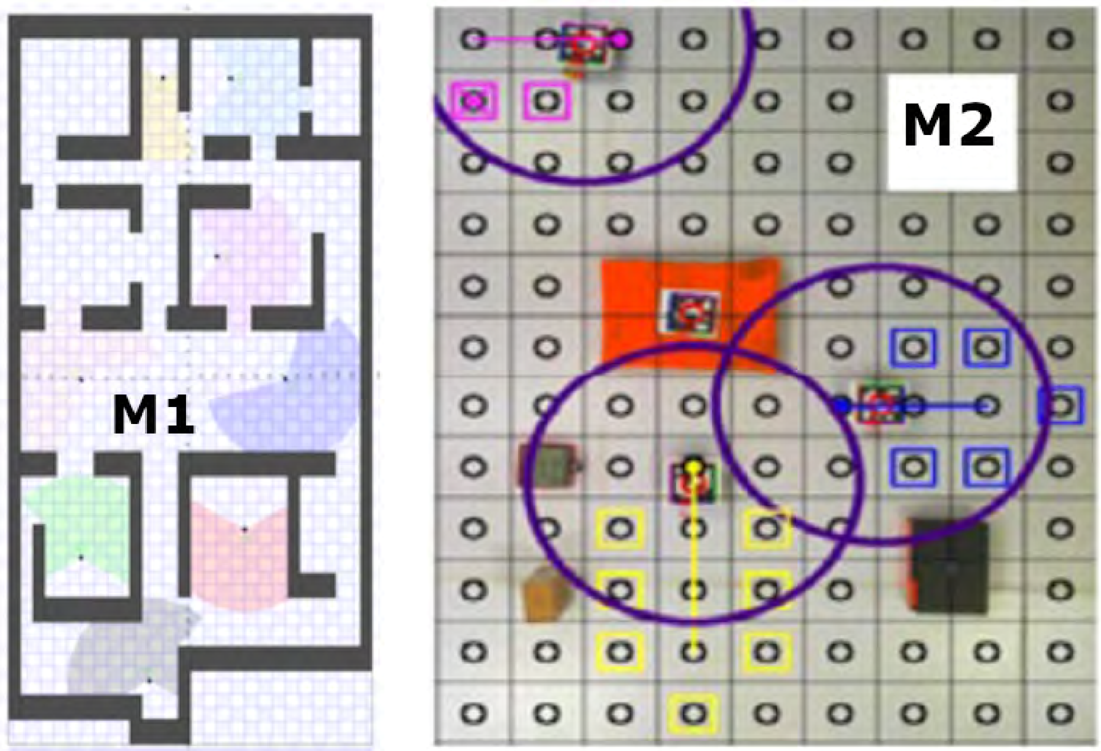


Figure 3.7: Terrain Maps Used in Simulation (M1) and Experiment (M2)

when the communication range was kept small (between 4 and 8 meters), the algorithms *MDFS* and *BnM* could not complete the coverage and terminated too soon. This contradicts the findings of the aforementioned. Both of these algorithms constantly mark the grid cells as having been visited, rendering them inaccessible to ever again being visited. Some cells may be marked as visited by one robot (robot A) but not by another. This could be due to random chance (say, robot B). As soon as robot B makes contact with robot A, robot B's map is updated, and it soon discovers that it is surrounded by cells that have already been visited. As a direct result, the two OTC algorithms finish their work prematurely. No matter how far apart they were able to communicate with one another, *BSA-CM, SSB*, and *BoB* never failed to complete their coverage. We separated the analysis of the *BnM* and *MDFS* algorithms from the analysis of the remaining ones; as a result, we discuss the two algorithms independently from this point onward in the chapter.

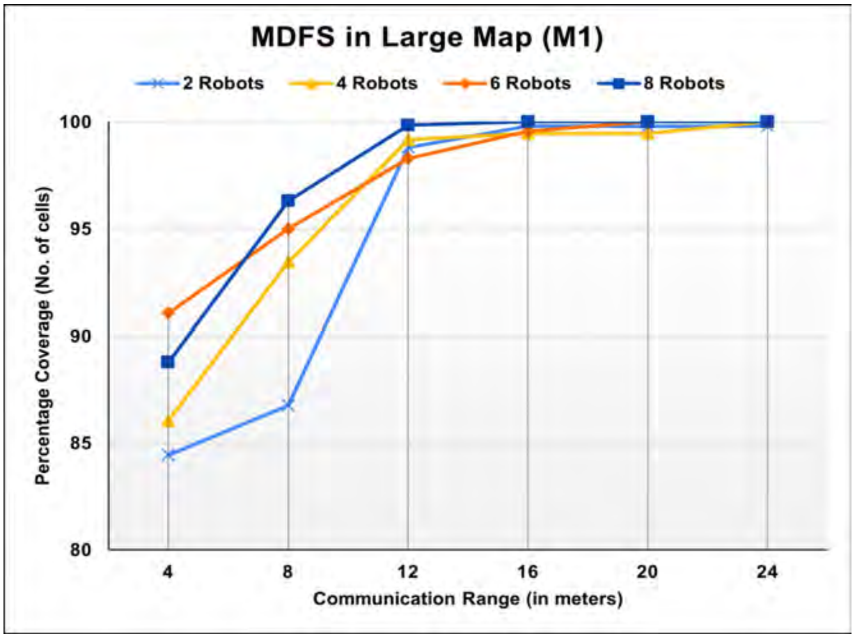


Figure 3.8: Percentage Coverage of *MDFS* in M1

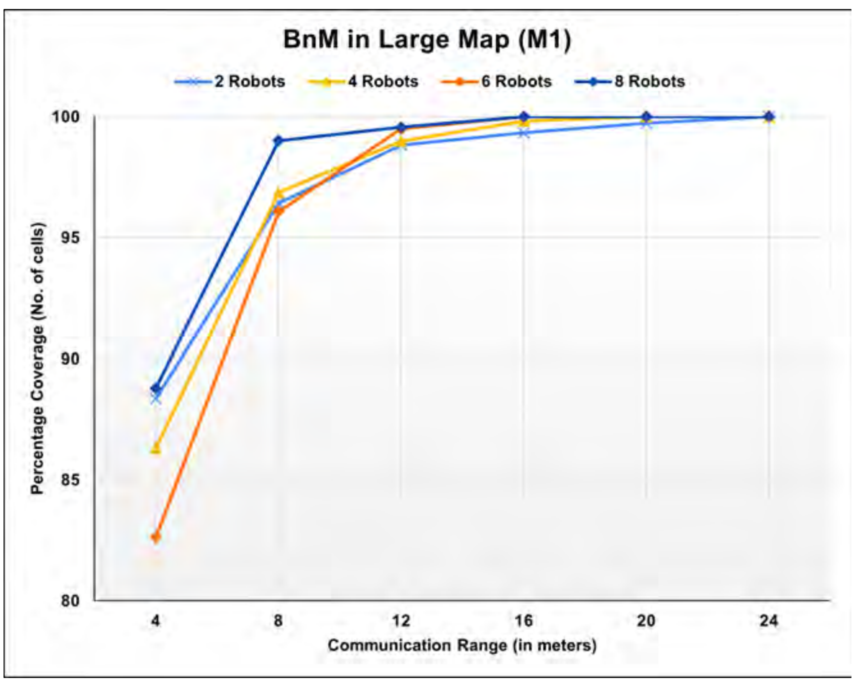


Figure 3.9: Percentage Coverage of *BnM* in M1

The comparison of percentage coverage variation with the communication range for both the algorithms is shown in Figure 3.8 and Figure 3.9. *BnM* suffers more from premature termination than *MDFS* because the robots executing *BnM* are more involved with each other and have significantly overlapping paths. As the communication range increases, the percentage of coverage increases. The critical communication range for both algorithms is 12 meters. The general behavior of redundant coverage for *BSA-CM, SSB, and BoB*, in maps M1 and M2, is that, as the communication range increases, redundant coverage decreases, see Figure 3.10 to Figure 3.13. There is a significant dip in the redundant coverage as the communication range is changed from 4 meters to 8 meters. It can be inferred that even a small increase is significant at smaller communication ranges.

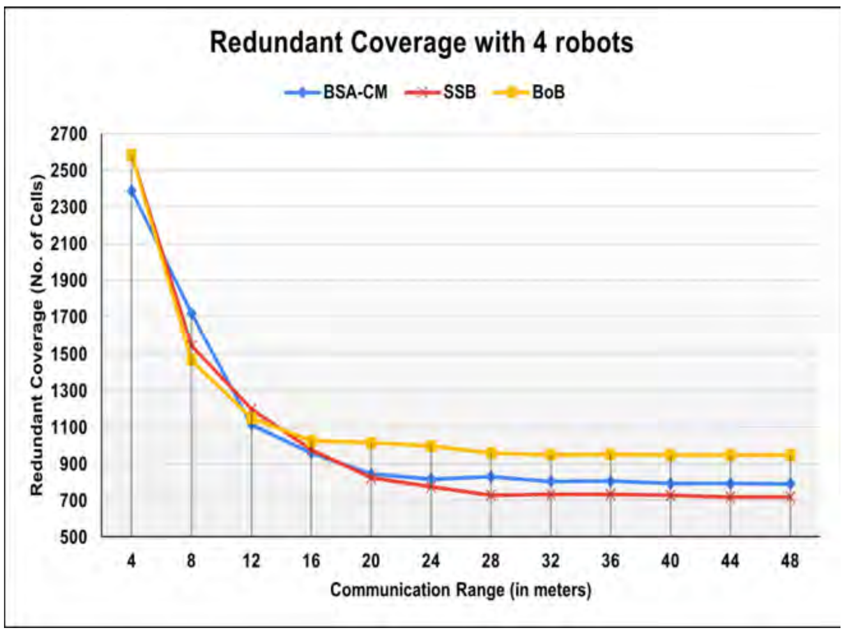


Figure 3.10: Redundant Coverage in M1 for 04 Robots

As shown in the graphs from Figure 3.10 to Figure 3.12, the performance of *BSA-CM* and *SSB* remains comparable even when the communication range and the number of robots vary. *SSB* has a slight edge in performance. It can also be observed performance of *BoB* is better than the other two when the communication range is minimal (between 4 and 8 meters). The experiment results show the same observation; see Figure 3.13. It can be seen that *BoB* performs better when the communication range is small, i.e., between 1 feet and 1.75 feet. However, the performance of the other two algorithms improves with a marginal increase in the communication range. The boustrophedon motion in *BoB* does

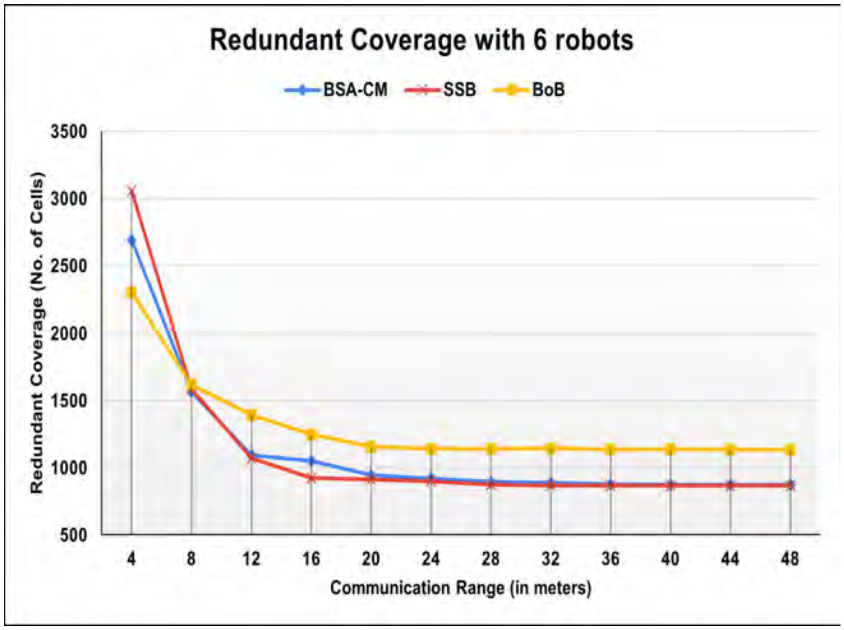


Figure 3.11: Redundant Coverage in M1 for 06 Robots

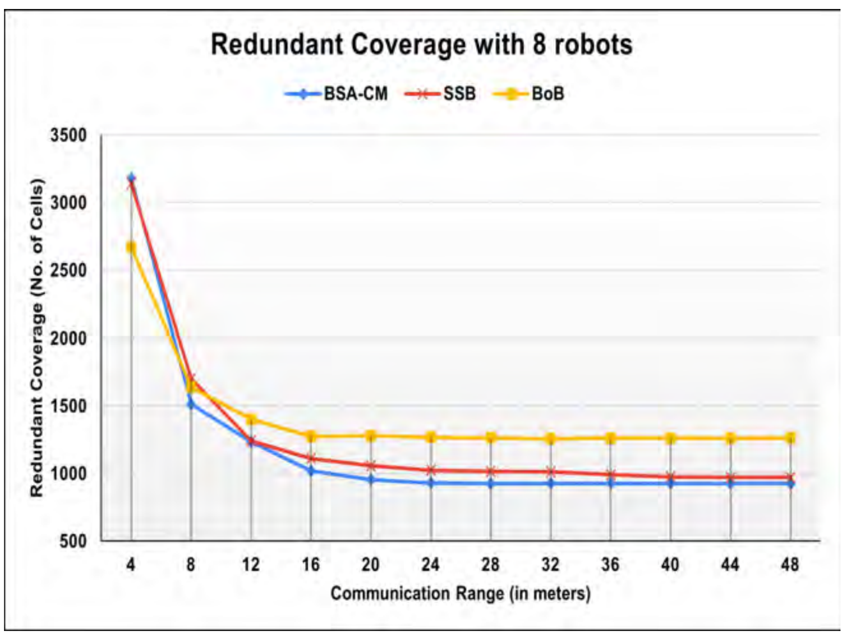


Figure 3.12: Redundant Coverage in M1 for 08 Robots

not focus on spreading the robots in space, unlike the spiral motion in *BSA-CM* and *SSB*.

It is also why the critical range for *BoB* is roughly 8 meters less than that of *BSA-CM* and SSB in the simulated map. It is also why the critical range for *BoB* is roughly 8 meters less than that of *BSA-CM* and *SSB* in the simulated map. The redundant coverage graph

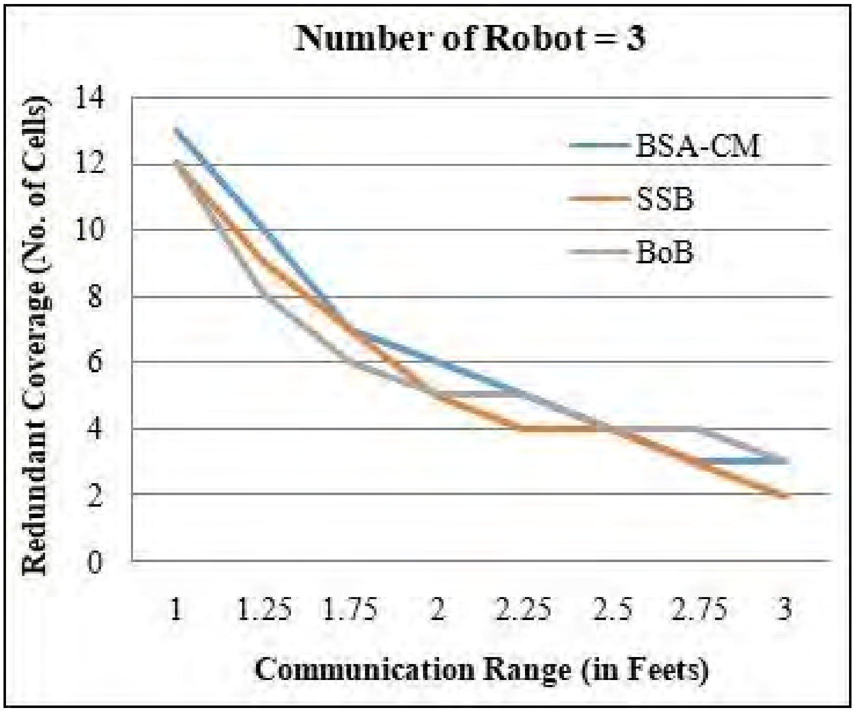


Figure 3.13: Redundant Coverage in M2 for 03 Robots

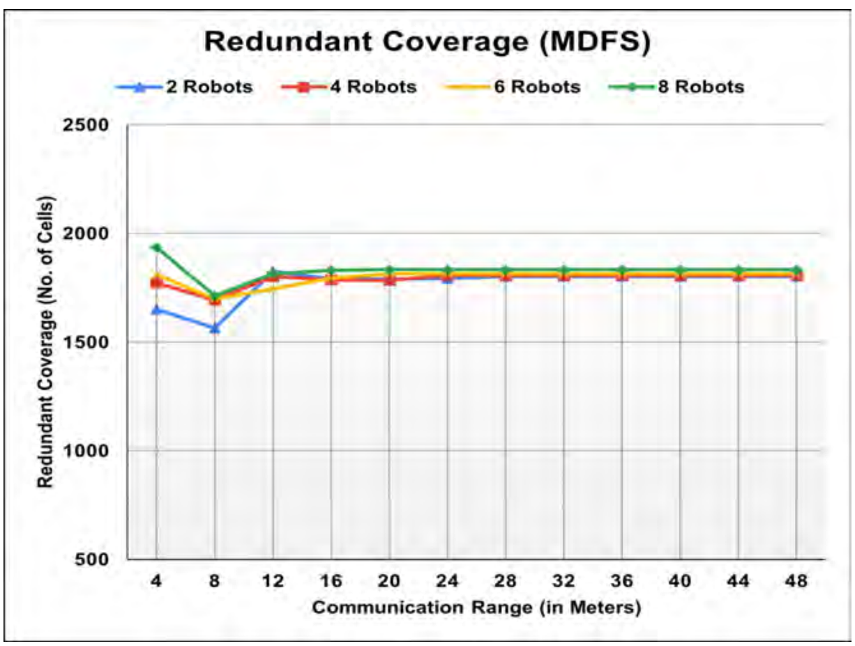


Figure 3.14: Redundant Coverage of *MDFS* in M1

for *MDFS*, shown in Figure 3.14, is more or less straight and tightly packed, apart from initial inconsistencies due to premature terminations. It is expected because, in *MDFS*,

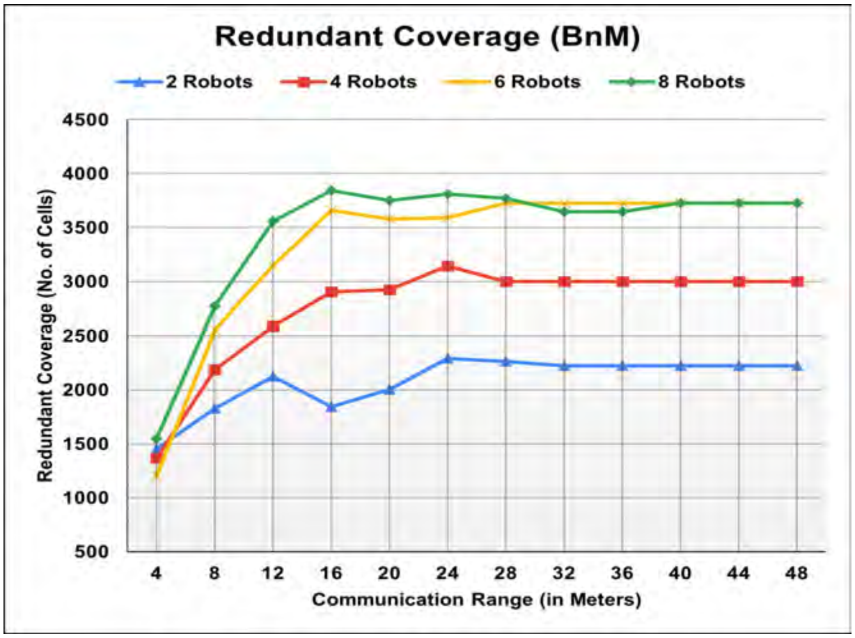


Figure 3.15: Redundant Coverage of *BnM* in M1

the robots are less involved and communicate less. The communication constraints have no significant impact on *MDFS*. The effect of premature termination is far more prominent in the redundant coverage of *BnM* than in *MDFS*. It can be easily observed from Figure 3.15 that the redundant coverage of *BnM* increases irrespective of the number of robots. The percentage coverage is lower with less communication, translating to less redundant coverage. By using terrain coverage, we can cover the area, but for Unknown areas, exploration requires the robots to build a map of the environment without visiting/ traversing the entire navigable region that cannot be covered by the terrain coverage, so it is better. Still, Terrain coverage and task completion are two different aspects of exploration that are closely related. Terrain coverage refers to the extent to which an area has been explored and mapped, while task completion refers to the successful accomplishment of specific objectives within the explored area.

Exploration involves two important aspects - terrain coverage and task completion. While terrain coverage helps in identifying the extent of the explored area and areas that need further exploration, task completion is crucial to achieve the specific objectives of the exploration and obtain the desired results. By properly coordinating and planning both these aspects, successful exploration can be ensured. To optimize the exploration process, it is recommended to move from terrain coverage to a centralized coordination algorithm for multi-robot system unknown area exploration. This approach helps in allocating tasks efficiently, improving coordination, communication, and safety, and making the exploration process more robust and adaptable. Ultimately, this approach can lead to new discoveries and insights into the natural world.

3.6 SUMMARY

Many state-of-the-art multi-robot OTC approaches assume that communication is omnipresent. In this chapter, the assumption of global communication is abandoned. Instead, five state-of-the-art multi-robot OTC approaches are re-implemented in simulation and on a physical robotic test bed. The effectiveness of these five methods is analyzed and compared using empirical data, and the findings are discussed. We have concluded that the performance of *BoB* is superior for a shorter communication range (one that is less than or equal to four meters). When the distance between the sender and receiver is greater than four meters, *SSB* and *BSA-CM* perform more effectively. When the communication range is less than 12 meters, *MDFS* and *BnM* should be avoided because they cannot complete the coverage in some situations and are inefficient (premature termination). Last but not least, one can conclude from the findings that SSB performs better than *BSA-CM*, *BnM*, *MDFS*, and *BoB*, which holds for both short and long-communication ranges.

Chapter 4

MULTI-ROBOT UNKNOWN AREA EXPLORATION USING FRONTIER TREES

4.1 INTRODUCTION

Using a team of autonomous mobile robots for unknown area exploration (online exploration) is a non-trivial problem. The objective is to obtain accurate and quality information about the environment while minimizing the team effort. Multi-robot coordination algorithms for online exploration have broad applicability in many real-world applications like mapping, search and rescue, ocean and space exploration, etc. Multi-robot systems (MRS) are capable of performing the online exploration task in a robust and fault-tolerant manner through redundancy. However, the design and selection of an appropriate coordination strategy are paramount, as it assigns specific exploration goals to the individual robots while reducing redundant exploration. Many researchers have proposed solutions for online exploration using MRS.

A market economy-based approach is suggested in [118] to govern the coordinated exploration, thus avoiding redundant exploration. In this approach, the robots negotiate the assignment of frontiers, which are potential targets for exploration using auctions. Since this approach only considers the distance of the robots to the frontiers, it failed in dispersing the robots in the environment. Therefore, the problem of redundant exploration is partially addressed in [118]. There are many extensions of the auction-based strategy suggested by [118]. For example, in [257], using auctions, the authors have addressed the problem of coalition formation for such situations when the number of exploration targets is lesser than the number of robots. In [268] and [139], the authors have addressed the problem of Simultaneous Localization and Mapping (SLAM) by using a Rao-Blackwellized particle filter (RBPF). They have proposed an online algorithm for exploration that selects the next best location that maximizes the size of the explored region and minimizes the uncertainties associated with the robot's location and the map. In [276] and [163], the authors have proposed distributed inference techniques to coordinate the robot team. Visual features were used in [276], and in [163], simulated laser scans are used. Some authors used semantic information for assigning exploration goals to the robot team [277], [278].

Some other coordination strategies are based on segmentation of the environment wherein the environment map is partitioned into segments using Voronoi graph [8], [258], [279], [280]. These segments typically represent an individual room or a portion of long corridors. Combinatorial optimization methods like the Hungarian method for optimal task assignment are used to assign these environment segments to the robots for further exploration. In [98], the authors used K-means clustering algorithm to obtain as many segments of the unknown environment as the number of robots. Although these approaches achieve reasonable performance in terms of exploration time, their applicability is limited in environments that can be partitioned into large and separate segments. Moreover, in a grid-decomposed large environment, iteratively performing clustering is computationally expensive and time-consuming. In small and cluttered environments, methods like the one proposed in [281] perform better. They have suggested an algorithm called *MinPos* (for Minimum Position), which computes the robot's rank towards its selected target (say f) based on path distance. The robot computes its rank for f, taking into account the number of other robots closer to f. The primary advantage of this approach is that it is decentralized and has low computational complexity. Furthermore, this algorithm is capable of dispersing the robots quickly in different directions, thereby reducing the overall exploration time. However, it requires a good deal of common knowledge, i.e., every robot should be aware of the position of its other teammates and a shared map with a common coordinate system. Also, in a communication-constrained environment *MinPos* will have high redundant exploration as several robots may have the same rank for the same target frontier.

The problem of intermittent communication in multi-robot exploration has been addressed by many researchers in recent years [60], [173], [249], [109]. A role-based distribution is proposed in [173]. This approach ensures that the robot network does not get fragmented, i.e., the exploration is accomplished while maintaining a continuous connection between all the team members. In [249], a comparative study of six different auctioning algorithms for multi-robot task allocation considering unreliable communication is conducted. To efficiently complete the online exploration task in a communication-restricted environment [109] proposed an approach using which the robot team while pursuing exploration, learns the network topology and updates the graph.

In [103], the problem of exploring unknown areas using a single robot with sensing, localization, and mapping capabilities was addressed. The *frontier tree* data structure proposed in [103] for single robot area exploration offers an elegant way to save frontiers to memory while generating a semantic understanding of their positions. As it grows, the tree simulates a fairly descriptive skeleton structure of the environment map with the position of each node and exploration status. This structure acts as a map approximation and a holder of the exploration state and can directly decide further exploration steps. Although the *frontier tree* approach suggested in [103] is significantly faster than state-of-the-art (SOTA) at calculating and recommending goals, it cannot be used in situations where more than one robot is available for exploration.

This chapter presents a novel multi-robot coordination approach for unknown area exploration, viz., Multi-Robot Frontier Tree Exploration (*MRFTE*). The primary contributions of this chapter are as follows:

- We have proposed a new abstraction called the *group*, which is a sub-set of robots whose explored regions overlap and thereby form a contiguous region. Each group exploits the *frontier tree* data structure for coordination. Our version of the *frontier tree* data structure is dynamic, informative, and provides the backbone of this proposal. In each iteration, in addition to storing frontiers, their positions and exploration state, it also keeps a record of on which node a particular robot is at present.

- We have proposed a technique to merge *groups* and their trees as soon as their maps overlap. Merging maps is necessary due to the distinct starting locations of the robots. By ensuring that each *group* has precisely one *frontier tree*, we create a shared state for the members in that *group*.

- We have suggested a method for designating exploration goals to the individual robots by picking from unexplored nodes on this tree.

- Finally, We have re-implemented seven (SOTA) approaches for multi-robot unknown area exploration to be compared with the proposed approach. Our approach outperforms all the seven SOTA approaches.

The rest of the chapter is structured in four sections. Section II gives a brief description of seven SOTA approaches compared to the proposed *MRFTE* approach. Section III presents a detailed description of the proposed approach. In Section IV, simulation settings and comparison results are described. Finally, the conclusion and future work is presented in Section V.

4.2 SOTA Approaches

This section briefly describes a representative subset of unknown area exploration approaches that we have considered for comparison with MRFTE.

4.2.1 Nearest Frontier Heuristic (NF)

It is a frontier-based decentralized coordination algorithm [5] for online exploration. It was one of the earliest pioneering works wherein each robot independently selected the nearest frontier for exploration. This approach greedily assigns the nearest frontier to each robot. In this approach, the target frontier cell (tf_{greedy}) is selected by using 4.1.

$$tf_{greedy} = \arg\min_{f \in F} path(f, r_i^p) \tag{4.1}$$

where F is the set of frontier cells presently visible and *path(f)* is the shortest path length between the frontier cell f and the i^{th} robot's current position, i.e., r_i^p. Dijkstra algorithm is used for path planning. Dead reckoning is conducted for continuous localization of the robots.

4.2.2 Information Gain Based Heuristic (D+IG)

In this approach, [6], the robot moves to the location, which reveals maximum-quality information. By executing *Next–Best–View (NBV)* algorithm, the robot generates a set (N_{sam}) of feasible candidate locations. Each location $q \in N_{sam}$ is assigned a score by using 4.2.

$$g(q) = A(q)exp(-\lambda L(q)) \tag{4.2}$$

Here, λ is a positive constant, *L(q)* is the length of the collision-free path obtained by executing the DFS algorithm, and *A(q)* measures the unexplored region visible from location q. Model alignment and merging are used for *SLAM*.

4.2.3 Cost Utility Based Heuristic (C+U)

This approach [7] simultaneously considers the cost of reaching a target location that has so far explored and its utility in a coordinated fashion. The cost of reaching frontier cells is computed by taking the product of the occupancy probability of the frontier cell and the distance to that cell in an iterative fashion. The utility of a target location is estimated as the probability that this location is visible from target locations assigned to other robots. Finally, a robot is assigned a target location based on the best trade-off between utility and cost. They have used a deterministic variant of the value iteration method for path planning based on dynamic programming. *SLAM* is achieved using the maximum likelihood and posterior pose estimators.

4.2.4 Voronoi Graph-Based Segmentation (VGS)

This approach [8] partitions the environment into multiple disjoint segments, and the robots are sent to individual segments for exploration. Segments could be separate rooms, corridors, or parts of larger corridors. The segment–based exploration reduces overlap among the field of view of robots' sensors, resulting in faster exploration. The Voronoi graph is created using an occupancy grid map for partitioning the environment. After that, graph nodes are segmented into disjoint sets such that each set of nodes belongs to a particular segment. The Hungarian method for task assignment [101] is used to assign these segments to the individual robots. The path planning algorithm is based on dynamic programming. Scan-matching algorithm is used for robot localization.

4.2.5 Goal Assignment Using Distance Cost (GADC)

This approach [11] formulates the exploration problem as multiple traveling salesman problems (*MTSP*). It applies a clustering-based *MTSP* solution that groups the targets into clusters and then determines the *TSP* distance cost for each cluster–robot pair. The target clustering uses the *K–means* algorithm with Geodesic distances. These clusters are sequentially assigned to the robots such that the length of the longest *TSP* tour is minimal. A distance transform-based path planner is used. This approach does not address SLAM. The robots are assumed to operate in a common coordinate system with perfect localization.

4.2.6 Multiple Rapidly Exploring Random Trees (M-RRT)

This approach [9] uses multiple Rapidly-exploring Random Trees (*RRTs*). The *RRT* generally samples the environment in a tree-like structure consisting of nodes and edges. Robots achieve exploration by traversing over the tree as it grows. They present a modified *RRT* in which a tree is used to search filtered frontier points. The approach constructs local and global frontier trees for achieving faster exploration. Multiple local trees are constructed using the frontier points near the robot.

In contrast, a single global tree is constructed using the frontier points in areas far from the robot. The filtering process clusters the points too close and stores only the centroid of each cluster, as no additional information is gained by including all the points. It also deletes the invalid points, leading to overlapping exploration in successive iterations. The master robot follows a market-based task assignment strategy. The robots send their bids to the master robot. The bid is a function of the robot's navigation cost and the information gained for a given frontier. The robot that places the highest bid is allocated the frontier. Dijkstra's algorithm is used for path planning. The ROS "gmapping" package implements Rao-Blackwellized particle filter [282] for SLAM is used.

4.2.7 Information-Driven *RRT* (ID-RRT)

The robot selects the target frontier for exploration by predicting the information gained from reaching each frontier based on local structures in the map built so far [10]. This prediction is based on how the walls around each frontier are expected to propagate in the unknown regions. The robot follows an *RRT*-based exploration path with a bias toward frontiers having higher estimated information gain. This approach for unknown area exploration is for a single robot. To compare it with *MRFTE*, we have extended [10] for multiple robots by following the rank-based strategy suggested in *MinPos* [281]. However, the rank calculation based on distance cost is replaced with the heuristic proposed in [10]. ROS "gmapping" package [282] is used for SLAM.

4.3 The Proposed Approach- MRFTE

This section presents the proposed modified approach for multi-robot coordination for unknown area exploration using *frontier trees*. Let R = $\{r_1, ..., r_n\}$ be a set of available robots, the union of whose initial sensing maps form a contiguous region.

4.3.1 Detailed Description of MRFTE

The main objective of exploration is to construct a map of the unknown environment and deliver it to a base station. For example, consider an earthquake-hitting building that looks visually intact. However, it may be unsafe for rescuers to move inside without an accurate damage assessment. The map of this building would assist the rescue team in efficiently and safely conducting the rescue mission. Therefore, multi-robot systems can be used to conduct the exploration mission in a time-efficient manner multi-robot systems can be used. With this motivation, we describe our multi-robot coordination approach, i.e., *MRFTE*. The robot team in our approach synchronizes with the *coordinator process* executing on the base station while exploring the environment.

Algorithm 4.1 Robot Behaviour

```
while true do
    wait for the coordinator's permission;
    if exploration complete then
        break
    end if
    visit assigned frontier;
    update robot map;
    post(finished exploration);
end while
```

The algorithm's pseudo-code executed by an individual robot is presented in *algorithm–4.1*. When the *coordinator process* signals the robots, they move to the frontiers assigned to them. While exploring, the robots build a map of their surroundings. Furthermore, they keep the local copy of their explored map up to date. After reaching the destination, they notify the *coordinator process*, which then assigns them a new exploration target.

Algorithm 4.2 Coordinator Process Behaviour

1: **while** true **do**
2: wait for the notification from each robot;
3: **for** each group g_i handled independently **do**
4: merge local map of each robot $r_j \in g_i$ to generate a common consistent map;
5: **end for**
6: **for** every pair of groups say $group_i$ and $group_j$ **do**
7: **if** explored regions of $group_i$ and $group_j$ overlap **then**
8: $closest\_node\_group_i$ = find node in
9: $group_i$.tree which is closest to
10: $group_j$.tree.root insert $group_j$.tree
11: under $closest\_node\_group_i$;
12: **end if**
13: **end for**
14: update maps of all robots;
15: **for** each group g_i **do**
16: *Assign_Goals(g_i)* [*refer Algorithm-4.3*];
17: **end for**
18: **if** exploration complete **then**
19: break;
20: **end if**
21: **end while**

The pseudo-code of the *coordinator process* that is executing on the base–station is presented in *algorithm–4.2*. Upon getting a signal from all the robots, the *coordinator process* merges the maps obtained from each robot into a separate unified map for each *group* (lines 3-4). The maps are merged by identifying and aligning the overlapping segments in the local maps of the robots. The rotation and translation between the two occupancy grid maps, say m_1 and m_2, are calculated by the coordinator. The objective is to maximize the overlap between the map m_1 and the transformed map $m_2^{'}$. Like [283], for map-merging, the coordinator calculates the image descriptors using Harris or Kanade-Lucas-Tomasi feature detectors. After that, the correspondences between the features of the two maps to be merged are determined. The occupancy grid maps are not sufficiently rich or nor do they have distinct features. As a result, it is not uncommon to have multiple correspondences for a given feature. Therefore, to determine consistent correspondences and transformations between the two maps, a modified RANSAC algorithm [283] has been implemented. The modified RANSAC algorithm enforces single correspondence in the other map for a given feature. Also, it ensures the rigid transform constraint, such that the features should have the same relative position in both maps. The coordinator then updates all the robots of each *group* with their respective merged maps.

1. **Merging Local Maps from Multiple Robots:** An occupancy grid map represents the local map of each robot in the coordinated robot system. A few procedures must be taken to combine these local maps into a logical global map.

   Robot A's Local Map ($\mathbf{R1}$):

   $$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

   Robot B's Local Map ($\mathbf{R2}$):

   $$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

   Figure 4.1: Robot A's Local Map ($\mathbf{R1}$) and Robot B's Local Map ($\mathbf{R2}$)

2. **Identify and Align Overlapping Segment:** The goal is to identify the overlapping region between two local maps and calculate the final overlapping segment. To

   $$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

   Figure 4.2: Overlapping Segment

   achieve this, it is necessary to compute the rotation and translation between the maps.

3. **Calculating Rotation and Translation:** Once the overlapping segments have been identified, the coordinator computes the rotation and translation between the two

maps $\mathbf{R1}$ and $\mathbf{R2}$ to align them. To accomplish this, we can determine the difference in the X and Y directions between the coordinates shared by both maps.

For R1, the center of the overlapping region (coordinates in common) is located at (2, 2). The overlap region's center for R2 is also located at the coordinates (2, 2).

To align R2 with R1, we calculate the translation required for R2's center to align with R1's:

- Translation in X direction $= 2$ (R1's center X) -2 (R2's center X) $= 0$
- Translation in Y direction $= 2$ (R1's center Y) -2 (R2's center Y) $= 0$

Since the translation in both the X and Y directions is 0, it means Robot B's local map (R2) is already aligned with Robot A's local map (R1) in the overlapping region. No translation is needed in this case.

Therefore, the output at this step is:

- Translation in X direction: 0
- Translation in Y direction: 0

4. **Image Descriptors and Feature Detectors** Image descriptors are calculated using *Harris or Kanade-Lucas-Tomasi feature detectors*. These detectors help in identifying distinctive points or features in the maps. Applying the Harris corner detection algorithm, we identify distinctive features in each map:

   Features in R1:

   (1, 1), (1, 2), (2, 2), (2, 3), (3, 2), (3, 3)

   Features in R2:

   (2, 3), (3, 3), (3, 2), (4, 3), (4, 4)

   Harris corner detection identifies points in the maps that are considered ”corners” or regions of significant intensity variation in different directions.

5. **Determine Correspondences:** Correspondences between the features of the two maps are determined. The goal is to find matching features between $\mathbf{R1}$ and the transformed version of $\mathbf{R2}$ ($\mathbf{R'2}$).

   To establish correspondences, we look for features in one map that are closest to features in the other map. The feature points (2, 3) and (3, 3) are common to both maps. So, they form a correspondence pair:

   (a) Correspondence 1: (2, 3) in R1 corresponds to (2, 3) in R2.

   (b) Correspondence 2: (3, 3) in R1 corresponds to (3, 3) in R2.

   The features (3, 2) in R1 and (3, 2) in R2 are also close to each other, so they form another correspondence pair:

   (c) Correspondence 3: (3, 2) in R1 corresponds to (3, 2) in R2.

   The other features do not have direct correspondences since they do not have close matches in the other map.

6. **Handle Multiple Correspondences:** It is possible to have multiple correspondences for a given feature due to the nature of occupancy grid maps, which may be lacking in distinct features. To address this situation, a modified RANSAC algorithm is used.

   Robot A's local map (R1) is merged with Robot B's local map (R2) by employing the modified RANSAC algorithm to handle multiple correspondences and find the optimal translation between Robot A's local map (R1) and Robot B's local map (R2). Despite potential outliers in the correspondences, the RANSAC algorithm helps to estimate the translation reliably.

   (a) Correspondences:

   - Correspondence 1: (2, 3) in R1 corresponds to (2, 3) in R2.

- Correspondence 2: (3, 3) in R1 corresponds to (3, 3) in R2.
- Correspondence 3: (3, 2) in R1 corresponds to (3, 2) in R2.

(b) Modified RANSAC Algorithm:

i. Select at random a minimal set of correspondences (two in this instance) to estimate the translation model. Suppose we select Correspondence 1 and Correspondence 2.

ii. Calculate the vector of translation that will align the selected correspondences. In this instance, we are able to compute the translation vector as follows:

Translation vector = $(2, 3)$ (Correspondence 2 in R1) $-(2, 3)$ (Correspondence 1 in R1) $= (0, 0)$

The translation vector is $(0, 0)$ since the selected correspondences are already in the same location in both maps.

iii. Count the number of inliers (correspondences that match the estimated translation well). In this case, all three correspondences are inliers, as they correspond well with the translation $(0, 0)$.

iv. Repeat steps 1 through 3 for a predetermined number of iterations while keeping track of the translation model with the highest number of inliers.

v. After the iterations, the model with the greatest number of inliers will be selected as the final translation model.

(c) Output:

In this case, after running the modified RANSAC algorithm, the output will be the translation vector that aligns Robot B's local map (R2) with Robot A's local map (R1). As the algorithm determined that no translation is needed (translation vector = $(0, 0)$), it means Robot B's local map (R2) is already aligned with Robot A's local map (R1) in the overlapping region.

- Translation vector: $(0, 0)$

This means no translation is required, and Robot A (R1) and Robot B (R2) are already in alignment in the overlapping region.

7. **Enforce Rigid Transform Constraint:** Ensure that each feature in one map has a unique counterpart in the other map, maintaining the relative positions of features after the transformation. In this step, the rigid transform constraint is enforced, which means that the relative positions of features in both maps should remain unchanged after the transformation. Since the translation vector is (0, 0) as determined in Step 5, Robot B's local map (R2) is already aligned correctly with Robot A's local map (R1) in the overlapping region, satisfying the rigid transform constraint.

   The rigid transform constraint ensures that features shared by both maps have the same relative positions in the merged global map as in their respective local maps. In this instance, since no translation is required, the relative positions of the common features are preserved, and the two maps can be seamlessly combined without distortion.

   Therefore, the output in Step is: Rigid Transform Constraint Satisfied

   This indicates that the alignment of the overlapping regions between the two local maps (R1 and R2) is accurate and that the relative positions of the common features after merging are preserved. Now, Robot A (R1) and Robot B (R2) can confidently proceed to update their local maps with the merged information from Step 8 to create a more inclusive global map.

8. **Update Robots with Merged Maps:** After determining the correspondences and transformations, the coordinator updates each group's robots with their respective merged maps. This ensures that the entire coordinated system has access to the updated global map, which represents the information gathered from all robots.

   In this step, both Robot A (R1) and Robot B (R2) update their respective local maps with the merged data to create a more comprehensive global map. Given that the translation vector is (0, 0) (as determined in Steps 5 and 7), Robot B's local map (R2) is already aligned with Robot A's local map (R1) in the overlapping region.

To update the maps, we combine the data from the two local maps in the region of overlap. Since both maps agree on the content of the overlapping region, we can update the merged global map by taking the maximum value (1) from each corresponding cell in the two maps. The merged map will look like the following:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure 4.3: Merged Map and Global Representation

Further, the *coordinator process* examines if the explored regions of any of the *groups* overlap. Such *groups* are combined (lines 6-9). When the *groups* are combined, their maps and trees are merged. We attach the absorbable tree under the nearest node in the absorbing tree to merge the trees. The positions of the robots are also recorded in the merged tree. Finally, the coordinator assigns exploration goals to the robots in each *group* by executing *Algorithm-4.3*.

As the proposed approach is based on the *frontier tree* data structure, we initialize the tree for multi-robot by executing the steps described in *algorithm–4.4*. It is a cautious method that directs every robot in the *group* to face inwards so that there is a clear, common explored area to begin. The tree's root is set to the location's centroid. In the first iteration of the algorithm's execution, the current locations of the robots are treated as frontiers. When these initial positions are fully processed, one coordinator iteration later, initialization is complete. *Algorithm–4.4* ensures that there is indeed a valid *frontier tree* available to each *group*. It first marks the robots' current positions as having been visited and then updates the tree with the current location of the robots and the current frontiers on the latest map. These are used to synchronize the tree and bring it up to date.

Algorithm 4.3 AssignGoals(*Group* g_i)

if g_i frontier_tree not initialized **then**
    InitializeFrontierTree(g_i) [refer Algorithm-4.4];
    return;
end if
in g_i.frontier_tree, for each robot $r_i \in g_i$ set r_i's previous destination as its current position and mark it as visited;
frontiers = getFrontiersFromEachGroupMap();
SynchronizeFrontierTree(g_i.frontier_tree, frontiers);
$exploration_{goal}$ = GetGoalsForExploration(g_i, g_i.frontier_tree);
get optimal robot-destination mapping using Hungarian Algorithm [101] with geodesic distance cost;
for each robot $r_i \in g_i$ **do**
    if $exploration_{goal}$ is available **then**
        r_i.assigned_frontier = $exploration_{goal}$
        post(r_i's permission to run);
    else
        post(r_i finished exploration);
    end if
end for

The *frontier tree* needs to be regularly updated with the latest information to stay updated with the most recent exploration. It is called *frontier tree* synchronization that follows from [103].

Algorithm 4.4 InitializeFrontierTree(*Group* G_i)

get centroid of robot positions;
set centroid as group.tree.root node;
for each robot $R_i \in G_i$ **do**
    set R_i.assigned_frontier as R_i.current_position;
    reset R_i's heading towards centroid;
    post(R_i's permission to run);
end for

Every frontier in the latest frontier set (say f_{new}) is inserted as a node into the tree made (unless its location is already a node). Further considerations decide the node's parent. Every frontier (say f_{old}) in the old set votes for its closest frontier, f_{new}, in the current frontier set. It must be noted that f_{new} might coincide with the f_{old}. This situation arises when the f_{old} is not chosen for exploration in the previous round, resulting in no change to its exploration state. In this case, f_{old} votes for f_{new}, but f_{new} is not inserted as a new node in the tree. If more than one f_{old} nodes vote for the same current f_{new} node, the f_{new} node is made a child of the closest assigned f_{old} node. The other assigned old nodes are considered completely explored and are "marked." If a new frontier f_{new} has only one voter f_{old} (and the new and old nodes have distinct locations), f_{new} node becomes a child of f_{old} node, and the f_{old} node becomes ineligible for selection as a goal. No action needs to be taken if they do not have distinct locations. Finally, if f_{new} node has no f_{old} voters, it is made a child of the node corresponding to the closest robot position and is assumed to have been freshly explored by that robot.

Next, *algorithm–4.3* returns at most as many exploration goals as the number of robots by invoking *GetGoalsForExploration* procedure (line–7). These goals are apportioned to the robots using the Hungarian method for task assignment [101]. The cost matrix used for the task assignment is initialized with the shortest-path distances between each robot and each goal by executing the *Jump Point Search (JPS)* Algorithm [284], [285] for path planning. Further, the coordinator notifies the robots to resume their exploration (line–12) if they have goals or posts finished on their behalf (line 14). The latter occurs when the robot has no goals assigned to it. The robot is not permitted to execute during the current round.

Any node in the tree can be in one of the three possible states - *marked*, *unmarked*, and *visited*. *Unmarked* nodes represent unexplored frontiers and are the only ones eligible for goals. *Marked* nodes are explored indirectly when a robot proceeds to explore another frontier in its vicinity, and these are not useful as exploration goals. *Visited* nodes represent those directly explored by a robot visiting them and are often on the way to further potential explorations. By default, a node is created *unmarked* and is later updated otherwise. The pseudo-code of the algorithm for obtaining goals is listed in *Algorithm-4.5*, which derives its motivation from [103].

However, several changes have been made to adapt it for multiple robots. At first, every node is considered *unreserved* (not to be confused with *unmarked*), and every robot is considered *undecided*.

Algorithm 4.5 Get Goals For Exploration(*Group* G_i, *Tree* T)

```
set T.node.reserved = false for each node ∈ T;
set G_i.robot.decided = false for each robot ∈ G_i;
for each robot R_i ∈ G_i do
    for each T.node freshly marked in synchronize do
        if T.node is in sensing range of R_i then
            flag = DetectCycle() [103];
            if flag == true then
                mark (T.node, R_i) pair as a possible cycle;
            end if
        end if
    end for
end for
for possible cycles do
    if R_i.decided == true then
        continue
    end if
    node = ExecuteTwoStepSearch() while avoiding reserved nodes [103]
    if node ≠ φ then
        node.reserved = true;
        R_i.decided = true;
    end if
end for
for each Robot R_j that is undecided do
    for each sibling_node of R_j.current_location do
        if sibling_node == unmarked and sibling_node.reserved == false then
            sibling_node.reserved = true;
            R_j.decided = true;
            skip to next robot;
        end if
    end for
end for
for each Robot R that is still undecided do
    destination = nearest unmarked, unreserved frontier, if any;
    if destination ≠ φ then
        destination.reserved = true;
    end if
end for
return all reserved nodes;
```

Along with the procedure, candidate nodes become *reserved* if selected and are removed from the selection pool; similarly, robots are set to be *decided* and removed from the decision-making process. The *GetGoalsForExploration* algorithm consists of three sub-parts - *cycle detection*, *sibling search*, and *nearest neighbour* (*NN*). *Sibling search* is a proposal made by this chapter to improve on the nearest neighbour (*NN*) should cycle detection fail to yield an assignment. The first step is to obtain possible cycles (lines 3-8). The cycles are detected by checking every freshly marked node against the position of every robot within the sensing range. If for any such *(robot, node)* pair, their rank, i.e., the tree depth, differs by more than one, it is considered a cycle.

Next, we execute the *two-step search* (lines 9–15) for each possible cycle. In the first step, a breadth-first traversal is made from the *marked* node to the root, looking for an eligible (*unmarked* and *unreserved*) goal. If this step fails, another step is executed, where the breadth-first traversal is made from the robot's (say $r's$) position to the *marked* node. If an eligible goal (say node n_e) is found at either stage, the procedure immediately reserves n_e, sets the robot r as having *decided*, and removes r and n_e from further consideration. The *two-step search* frequently fails in uncluttered and bigger maps, owing to the low number of cycles encountered. Without a *sibling search*, opting straight for the nearest neighbour (*NN*) approach forms a very deep tree, causing the tree to become a poor representation of the real map. The purpose of the *sibling search* step is to try to broaden the tree where possible to counter this. Thus, a *sibling search* (line 15-20) is attempted if the *two-step search* procedure fails to generate goals for some or all robots. Here, the sibling nodes of the robot's current position are examined for their eligibility for exploration.

If they are found eligible, then they are reserved as goals. Finally, should this also fail, the procedure defaults to *nearest neighbour* (*NN*) (lines 21–24), where the nearest eligible frontier to each remaining robot is reserved. All *reserved* nodes are then returned as goals for the robots. Remember that these measures may result in fewer goals than robots, especially in the beginning (when the number of frontiers may be less) or toward the end (when most of the map is explored and only a few frontiers are left). In such cases, some robots may not be assigned any frontier.

4.4 Complexity Analysis of MRFTE

4.4.1 Algorithm–4.1

This algorithm defines how each robot in the system behaves. It waits for permission from the coordinator (Algorithm 4.2) before performing tasks such as exploring assigned areas (frontiers), updating its map, and signalling the completion of exploration. The complexity of Algorithm 4.1 is mainly linear, O(m), for each robot's actions. This involves waiting, exploring, updating, and signalling.

4.4.2 Algorithm–4.2

Algorithm 4.2 is a complex system that coordinates multiple robot groups (m groups). It merges maps, manages group interactions, assigns goals, and monitors completion conditions. The algorithm's complexity combines linear O(m) operations with nested loops and conditional checks that depend on group sizes $|g_i|$. This yields a complexity ranging from linear O(m) to quadratic O(m^2), depending on the specific tasks like map merging and pairwise group interactions being performed. Finally, The complexity of this algorithm is O(m * AssignGoals(Group g_i)) depending on tasks like map merging and pairwise group interactions.

4.4.3 Algorithm-4.3

The overall complexity of this algorithm is O($|g_i|$)+O(K)+O(n^3). The dominant factors contributing to the complexity are typically the number of robots in the group G_i $|g_i|$, K is the no. of the group and the complexity of solving the assignment problem using the Hungarian Algorithm (n^3).

4.4.4 Algorithm-4.4

The function initializes a frontier tree for a group, which is based on the positions of robots. The complexity of this algorithm is generally determined by the number of robots in the group, denoted as $|g_i|$. Other operations in this function have either constant-time complexity or complexity linearly proportional to the number of robots in the group. The overall complexity of this algorithm is O($|g_i|$)

4.4.5 Algorithm-4.5

The ”Get Goals for Exploration” algorithm’s complexity depends on its individual steps, specifically cycle detection, two-step search, and node reservation. Overall complexity can be represented as O($|T|$ + $|G_i|$ + cycle detection complexity + two-step search complexity + node reservation complexity), where T is the number of nodes in the tree, and G_i is the number of robots in the group G_i.

4.5 EXPERIMENT AND RESULT

The proposed approach for multi-robot exploration is thoroughly evaluated through a series of experiments performed in simulation in two different environments. This section describes the simulation settings and the results thus obtained. Seven SOTA approaches have been re-implemented and extensively evaluated. The proposed approach is compared with these seven approaches, and the results thus obtained are also explained.

4.5.1 Simulation Setup

We have performed the experiments in the Player/Stage robot simulator [286]. The simulated robot is a model of Pioneer3-AT robot [287], which is a four-wheel differential drive robot equipped with Hokuyo Laser Range Finder - URG-04LX-UG01 [288] that has a field of view (FoV) of 240 degrees and a scanning range of six meters. All experiments are conducted on a computer with 32 GB of RAM and an Intel Core i7 (10^{th} generation) processor. Two different types of environments, unknown to the robots initially, are used for exploration. The first environment (Map-1) map is displayed in Figure 4.4(a), an uncluttered indoor environment of size 900×600 pixels.

The second environment is a cluttered indoor environment (Map-2) of size 900×900 pixels that are displayed in Figure 4.4(b). The third environment is a hospital indoor environment (Map-3) of size 1800×900 pixels that are displayed in Figure 4.4(c). The

(a) Map-1: Uncluttered Indoor Environment

(b) Map-2: Cluttered Indoor Environment

(c) Map-3: Hospital Indoor Environment

Figure 4.4: Various Types of Environment

resolution of both maps is set to 0.03 m per pixel. The white color is the free space to be explored, while the black color represents the walls and the obstacles. Map-1 does not have obstacles, and the robots easily navigate in open areas. In Map-1, lesser but bigger-sized frontiers appeared compared to Map-2. Also, both long and narrow corridors frequently generate fewer frontiers than the number of available robots. Map-2 and Map-3 are both challenging environments with different obstacles. While Map-2 has numerous obstacles, Map 3 is made up of many small and large rooms, a long corridor, and a complete hospital structure. As a result, Map-2 produces more smaller-sized frontiers than Map-3. These maps are specifically designed to test the performance of coordination algorithms in environments with a high density of obstacles.

4.5.2 Simulation Workflow

Here, the workflow of the Coordinator and each Robot is described using a state diagram shown in Figure 4.5. The Coordinator and the Robot execute in their thread. More than one Robot thread can be active at a time. After initialization, the Coordinator sends a start message to each robot and goes into the waiting state. Essentially, it waits for the local map (LM[i]) of each Robot-i, where ”i” indicates the i^{th} robot. At this point, the working of the two threads, i.e., the Coordinator and the Robot, is explained separately as follows:

- *Robot thread* - each Robot-i, after receiving the start message from the Coordinator, scans the environment using its sensors, extracts frontiers, and builds its local map (LM[i]). It then signals and sends LM[i] to the Coordinator and goes into the waiting state for the Coordinator to send the global map (GM) and assign the task, i.e., the target location for exploration. Each Robot-i examines if it has received a valid task from the Coordinator and moves toward the assigned target. It then signals the Coordinator of task completion and goes into the waiting state for subsequent task assignments and the updated GM. The robot terminates when the Coordinator does not assign it any task (a goal for exploration).

- *Coordinator thread* - upon receiving a signal from each Robot-i and LM[i] the waiting Coordinator, merges LM[i] and updates the global map (GM). Then, the coordinator finds frontiers in the GM. If no frontiers are found in the GM, the Coordinator sends ”null” tasks to the robots and terminates, indicating the end of exploration.

4.5.3 Results and Discussions

This section presents the results of performing experiments in both Map-1 of Figure 4.4(a) and Map-2 of Figure 4.4(b). A series of 100 simulations are conducted for each of the

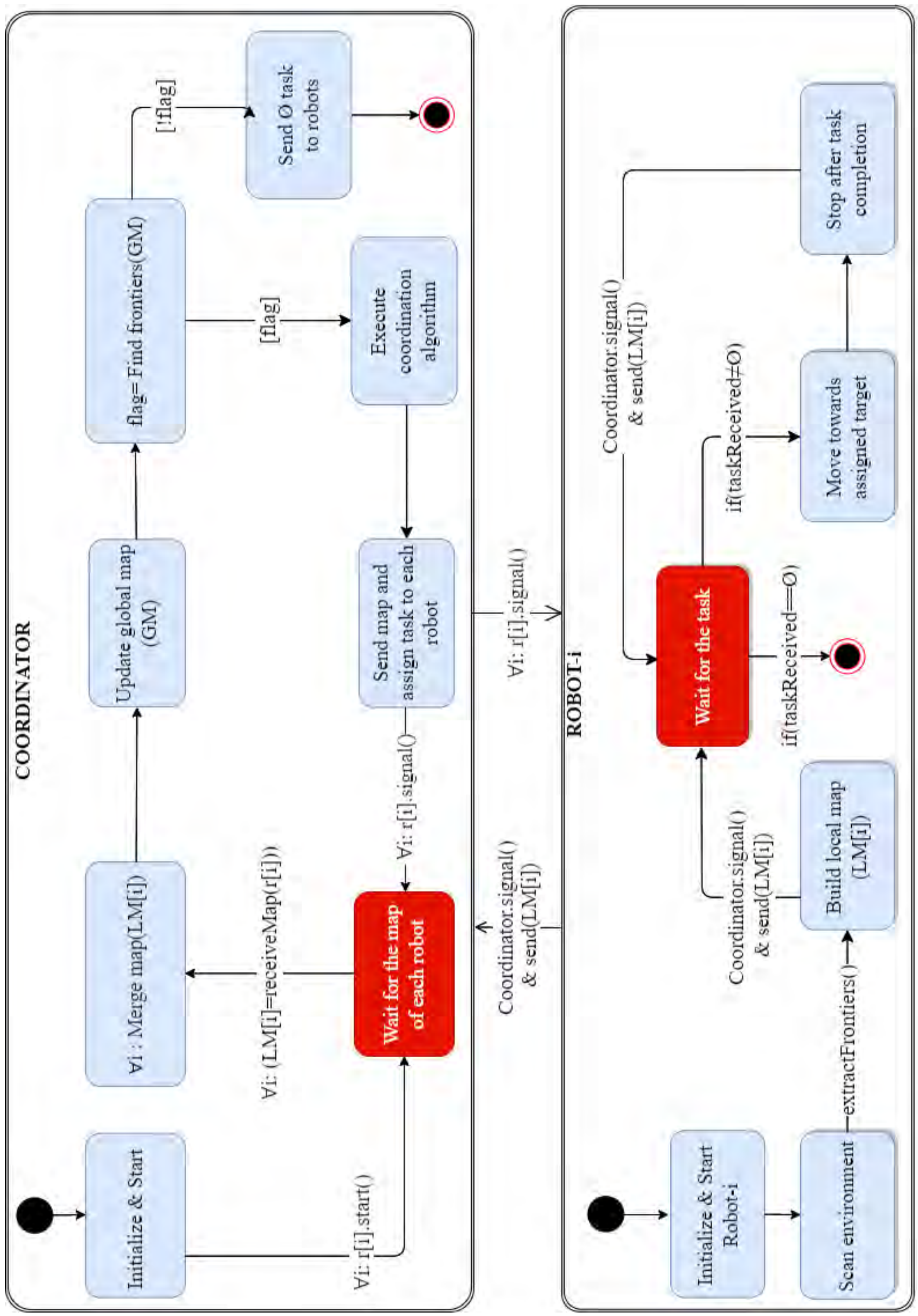


Figure 4.5: Workflow of the Coordinator and the Robot in Simulation

seven SOTA approaches we have re-implemented and the proposed *MRFTE* approach. The number of robots varies from 2 to 4, 6, and 8. In each simulation run, the starting positions of the robots are randomly chosen. However, for a fair comparison, the robots' starting positions are kept the same across different approaches. We have compared all the approaches based on two metrics.

1. *Completion Time* - the time it takes for the robot team to complete exploration. The exploration is complete when no more frontiers are left for any robot to explore.

2. *Cumulative Distance Travelled* - that is the summation of the distance traveled by each robot until the whole map is explored and no more frontiers are visible to any robot. The time constitutes sensor data acquisition and processing time; time is taken in the robot's navigation while avoiding obstructions and synchronization with the base station. Also, many a time, the robots interfere with each other. As a result, the planner wastes time in detouring. Therefore, the completion time metric is not directly proportional to the distance traveled by the robot team.

One important observation is that irrespective of the coordination algorithm and the map used, both the *Completion Time* and *Cumulative Distance Travelled* decrease due to increasing the number of robots. The coordination mechanism of the first three approaches, i.e., *NF* [5], *D+IG* [6], and *C+U* [7] is not very sophisticated. *NF* does not consider the usefulness of the frontier. The robots executing *NF* do not coordinate. As a result, they do not get dispersed, and the two robots operating in close proximity end up choosing the same frontier cell to explore. Dijkstra algorithm is used for path planning, which does not scale well on occupancy grid maps. As shown in Figure 4.6-4.11, this approach is the least effective on both metrics. *D+IG* does not consider the errors that occurred in poly-line extraction or image alignment. Such errors result in a longer motion path for the robot. Moreover, the suggested heuristic in *D+IG* could not disperse the robots operating in close proximity, and therefore, the robot team gets locally dispersed. However, some form of coordination produces better results compared to *NF*. *C+U* employs a better heuristic function and successfully disperses the robots. Still, it works with the premise that the robots are aware of other robots' relative positions, which is a strong assumption. It is evident from Figure 4.6-4.11 that with the increase in size and complexity of the map, both the exploration completion time and the distance traveled drastically increase for $C+U$. Nevertheless, this approach is superior to *NF* and *D+IG* on both metrics.

VGS [8] is an example of a segmentation based approach. This approach successfully achieves global dispersion, thereby significantly minimizing the overlapping exploration. It is evident from the graphs of Figure 4.6-4.11. However, in complex environments, the performance suffers due to the fact that the approach is too restrictive, particularly at the doorways, in the sense that the two rooms form a single segment if the local minima in between are not in the direct proximity of a junction node. It also assumes that the relative starting pose of the robots is known beforehand. It can be seen from the plots of Figure 4.6 and 4.9 that *VGS* and *MRFTE* have comparable performance on both the metrics in Map-1 of Figure 4.4(a). These results prove that *MRFTE* is capable of quickly dispersing the robot team. moreover if we refer to the graphs of Figure 4.7 and Figure 4.10, the effectiveness of *MRFTE* is more visible in Map-2 of Figure 4.4(b), wherein its performance is significantly better than *VGS* and all the other approaches. The ability of *MRFTE* to achieve global dispersion is attributed to the fact that the goal assignment algorithm (*algorithm-4.3*) ceases to assign to the robots, previously explored frontiers (say f_{past}), and the exploration goals that appear in the close proximity of f_{past}. Additionally, the mechanism of *sibling search* introduced in *MRFTE* does not allow the combined *frontier tree* of each group to grow far down. Thus, collectively, the robot team does not leave too many frontiers far behind and explores them early while achieving dispersion. These two characteristics are not evident in other SOTA approaches.

In *GADC* [11], K-means clustering is employed to produce frontier clusters that are assigned to the robots based on *TSP* distance cost. The benefit obtained from the heuristic function that is used for splitting the frontiers is evident only for a certain number of robots, i.e., four robots in the map of Figure 4.4(a), and it reduces as we increase the size of the robot team to 6 and 8 (refer to Figure 4.6 and 4.9). Moreover, its performance

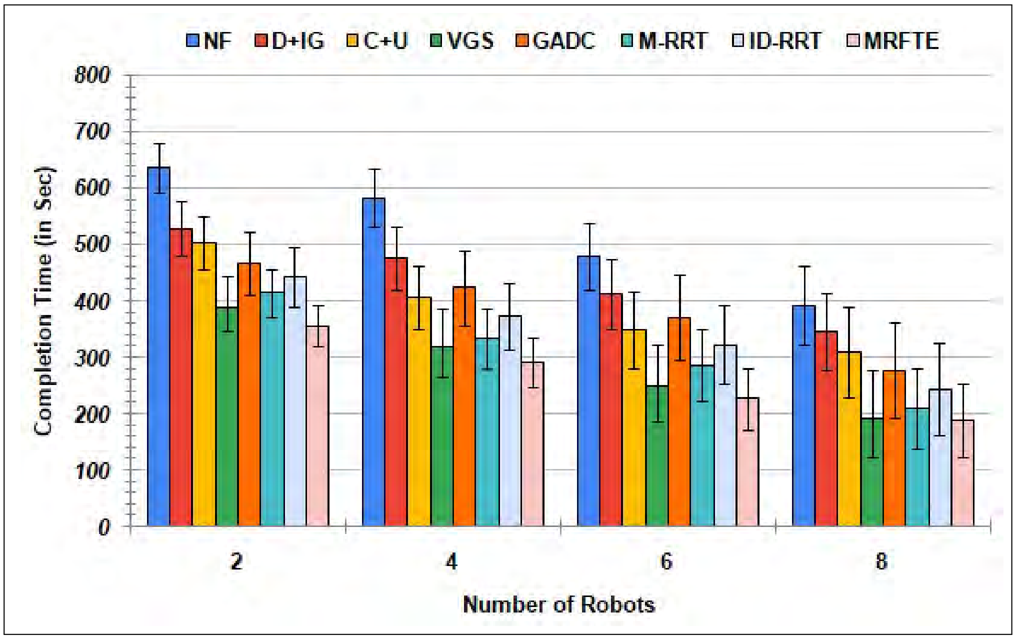


Figure 4.6: Exploration Completion Time in Map-1

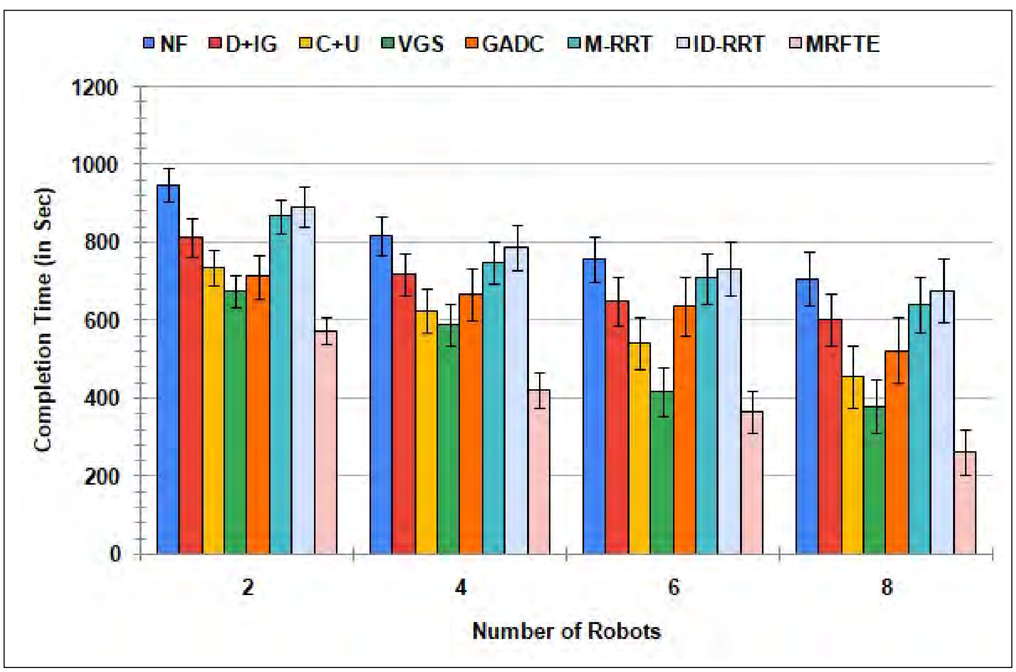


Figure 4.7: Exploration Completion Time in Map-2

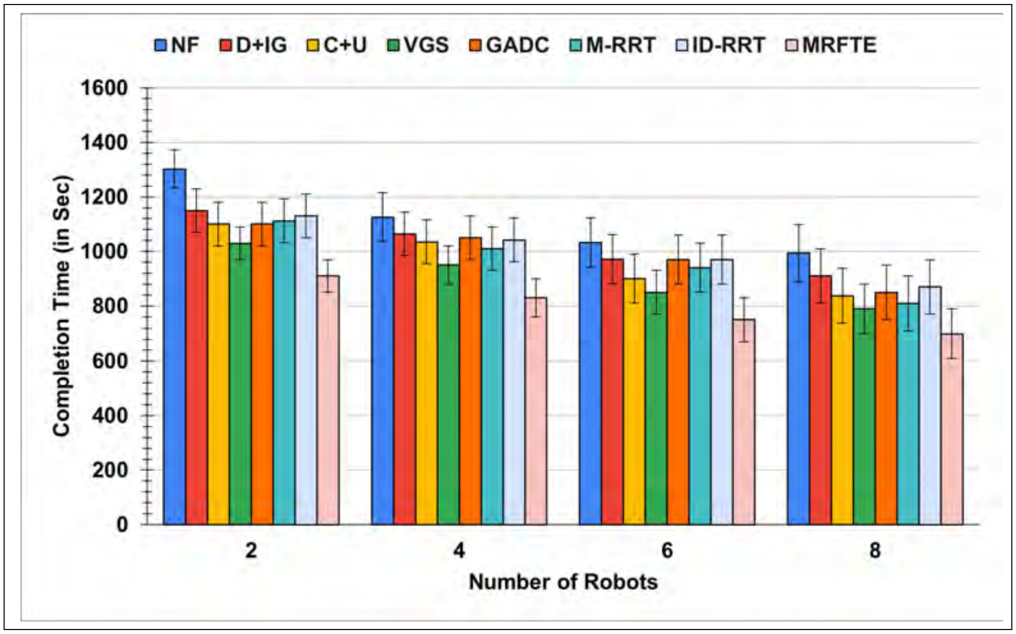


Figure 4.8: Exploration Completion Time in Map-3

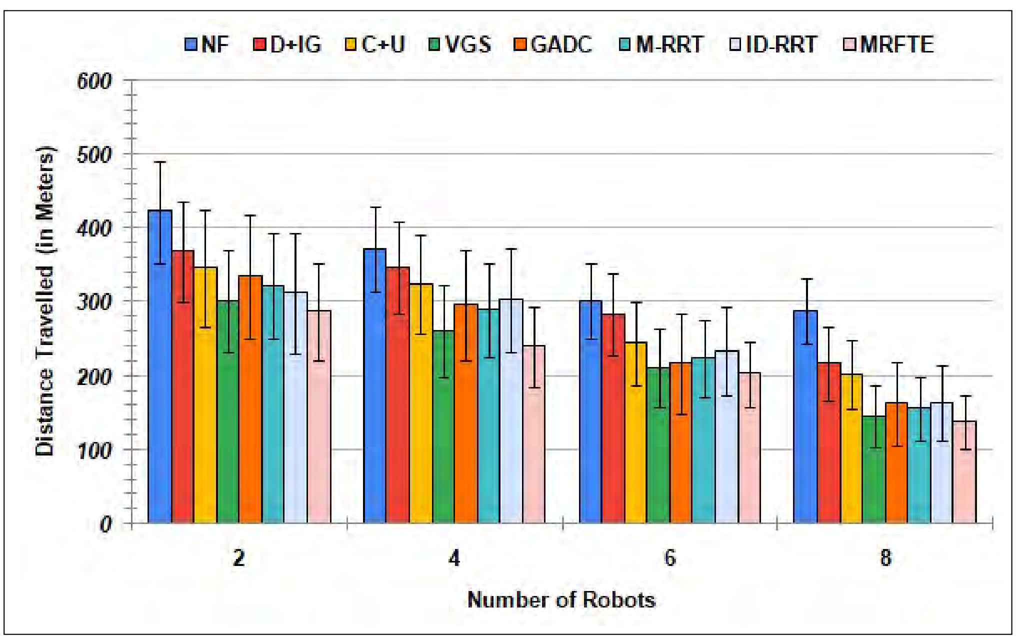


Figure 4.9: Cumulative Distance Travelled in Map-1

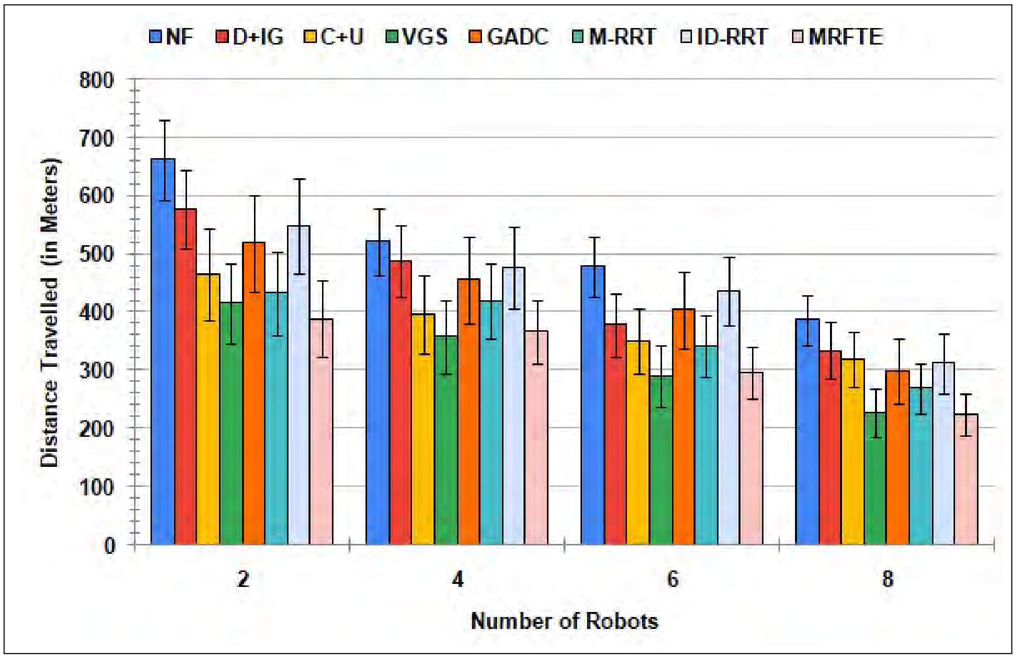


Figure 4.10: Cumulative Distance Travelled in Map-2

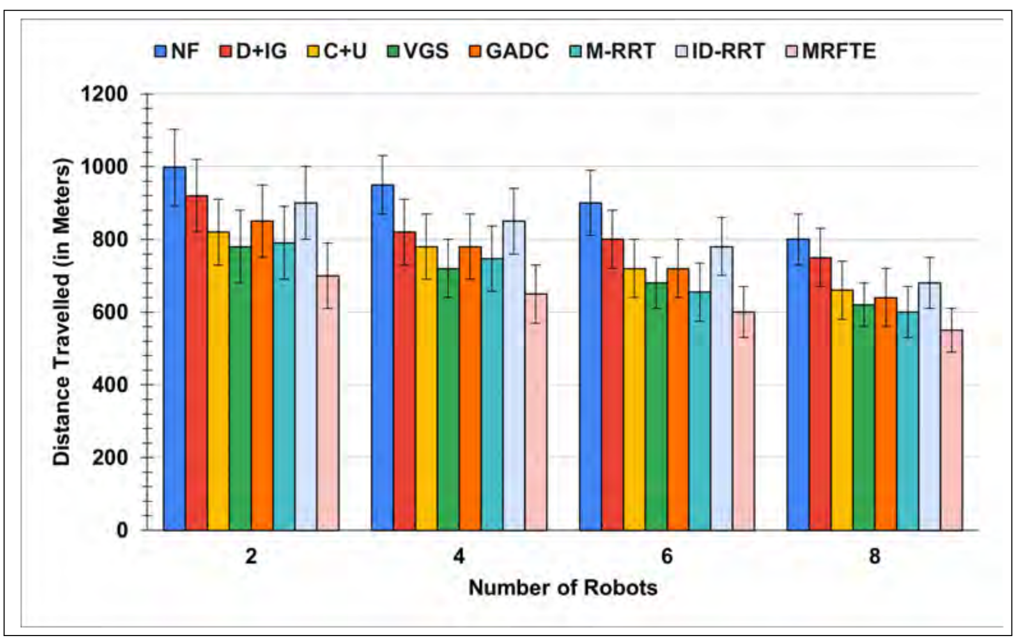


Figure 4.11: Cumulative Distance Travelled in Map-3

Figure 4.12: Firebird-VI Robot Equipped with HOKUYO URG-04LX-UG01 2D Laser Scanner for Real Robot Experiments

severely degrades in unstructured environments, i.e., in the map of Figure 4.4(b), as can be seen in the graphs of Figure 4.7 and 4.10. This approach does not address SLAM. The robots are assumed to operate in a common coordinate system with perfect localization. Therefore, it has an advantage in completion time compared to other approaches.

Sampling–based approaches such as *M-RRT* [9] and *ID-RRT* [10] improve the efficiency of single robot exploration by generating collision-free paths by appending a randomly sampled point to the tree. In an unknown, cluttered environment, it is non-trivial to sample a point occluded by obstacles and can still be connected to the tree. *M-RRT* and *ID-RRT* are computationally expensive, restricting their use in large-scale planning and real-time operations. These findings can be corroborated by the graphs of Figure 4.6 and 4.7. However, the distance traveled by the robot team for *M-RRT* is lesser than *NF*, *D+IG*, *C+U*, *GADC*, but it is higher than *MRFTE*, as can be seen in Figure 4.9 and 4.10. On the contrary, both the time and the distance traveled of *ID-RRT* is higher than *C+U*, *VGS*, *GADC*, *M-RRT*, and *MRFTE* in the Map of Figure 4.4(b) because it fails to predict the information gain in complex unstructured environments, which leads to inferior task allocation.

The quantitative results of the comparison between *MRFTE* and seven other state-of-the-art approaches, using both two and eight robots in Map-2, are described below. Please refer to Figure 5.4 and Figure 5.7 for more details.

- **Comparison with heuristic-based approaches** - In comparison with heuristic-based approaches, MRFTE demonstrated notable advantages. With two robots, MRFTE exhibited a time reduction of 41.05%, 31.70%, and 25.33% compared to NF, D+IG, and C+U, respectively. Additionally, MRFTE covered 42.42%, 34.48%, and 17.39% less distance than NF, D+IG, and C+U, respectively. Scaling up to eight robots, MRFTE outperformed significantly, consuming 62.16%, 53.33%, and 41.66% less time than NF, D+IG, and C+U, respectively. Similarly, MRFTE traveled 39.47%, 32.35%, and 28.12% less distance than NF, D+IG. These results emphasize the superiority of MRFTE over heuristic-based approaches, and consistent findings were observed even with changes in the map.

- **Comparison with segmentation-based approaches** - When employing two robots, it was observed that MRFTE exhibited a 12.5% reduction in time compared to VGS. Additionally, MRFTE covered 9.52% less distance than VGS in this scenario. Furthermore, utilizing eight robots, MRFTE demonstrated notable performance improvements, showcasing a substantial 28.81% reduction in time compared to VGS. Similarly, MRFTE traveled 4.16% less distance than VGS in this eight-robot configuration.

- **Comparison with sampling-based approaches** - When employing two robots, it was observed that MRFTE demonstrated a time reduction of 34.11% and 36.36% compared to M-RRT and ID-RRT, respectively. Similarly, MRFTE covered 13.63% and 30.90% less distance than M-RRT and ID-RRT, respectively, in this scenario. Furthermore, with the use of eight robots, MRFTE exhibited notable performance improvements, showcasing a significant 56.25% and 58.82% reduction in time compared to M-RRT and ID-RRT, respectively. Likewise, MRFTE traveled 17.85% and 28.12% less distance than M-RRT and ID-RRT.

- **Others** - When utilizing two robots, it was observed that MRFTE exhibited a 17.64% reduction in time compared to GADC. Additionally, MRFTE covered 26.92% less distance than GADC in the same configuration. Furthermore, with the use of eight robots, MRFTE demonstrated noteworthy performance improvements, showing a substantial 46.15% reduction in time compared to GADC. Similarly, MRFTE traveled 23.33% less distance than GADC in this eight-robot setup.

4.5.4 Real Robot Experiments

We conducted experiments with a team of four Firebird-VI (*FB-VI*) robots [287] see Figure 4.12, to verify the functionality of the proposed *MRFTE* algorithm. *FB-VI* is a differential drive robot equipped with HOKUYO URG-04LX-UG01 2D Laser Scanner (FoV of 240^o) for constructing the map of the unknown environment. The robot also has an onboard Intel NUC 10 PC with WiFi and 10^{th} Generation Intel Core i5-10210U Processor with 4 GB RAM and 128 GB SSD. Ubuntu 14.04 LTS operating system is installed on the onboard PC. Additionally, it has 8 MaxBotix Ultrasonic sensors with a range of up to 5 meters, a 9 DOF IMU, and position encoders for obstacle detection/avoidance and navigation, respectively. The experiments were conducted indoors, as shown in Figure 4.14. The software stack for unknown area exploration using a team of *FB-VI* robots is developed in the Robot Operating System (ROS). It comprises ROS packages for SLAM, Navigation, Control, Map Merging, Coordination, and Communication. We have implemented our approach, i.e., *MRFTE* for multi-robot coordination as a ROS package. However, for SLAM, we used Hector-SLAM [271], for Navigation and Control *FB-VI* vendor libraries were used, and for inter-robot communication, FKIE Multimaster ROS package is used that allows a set of nodes to establish and manage a multi-master network [289] as shown in Figure 4.13.

The multirobot_map_merge ROS package [290], which was originally designed to work with the Gmapping SLAM library [291], successfully merges the maps of multiple robots. But, it has a few limitations, such that it works with the premise that all the robots have the same origin and produce maps of the same size. In our experiments, Hector-

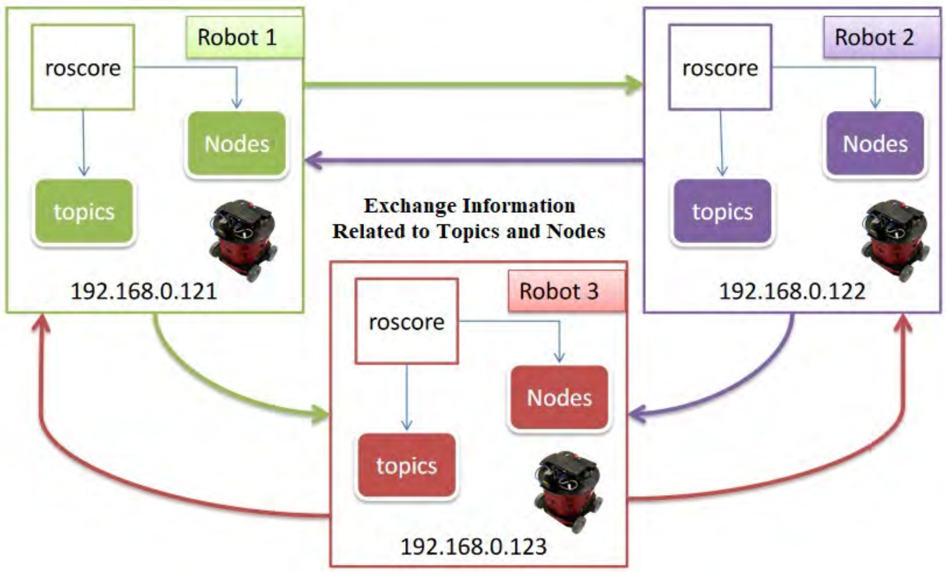


Figure 4.13: Example Network of Three *FB-VI* Robots based on FKIE Multimaster for Information Interchange

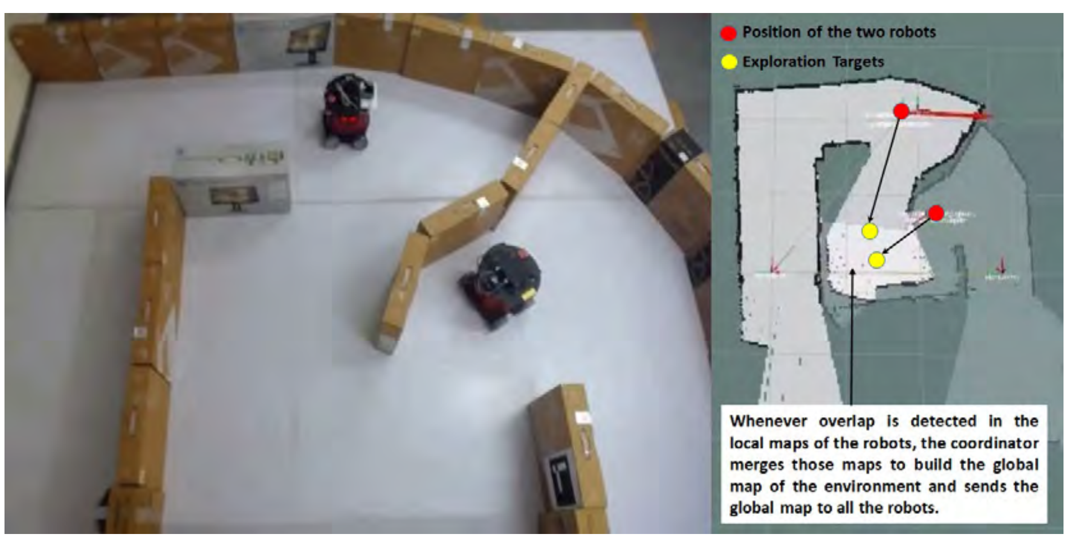


Figure 4.14: Combined Map of Two *FB-VI* Robots

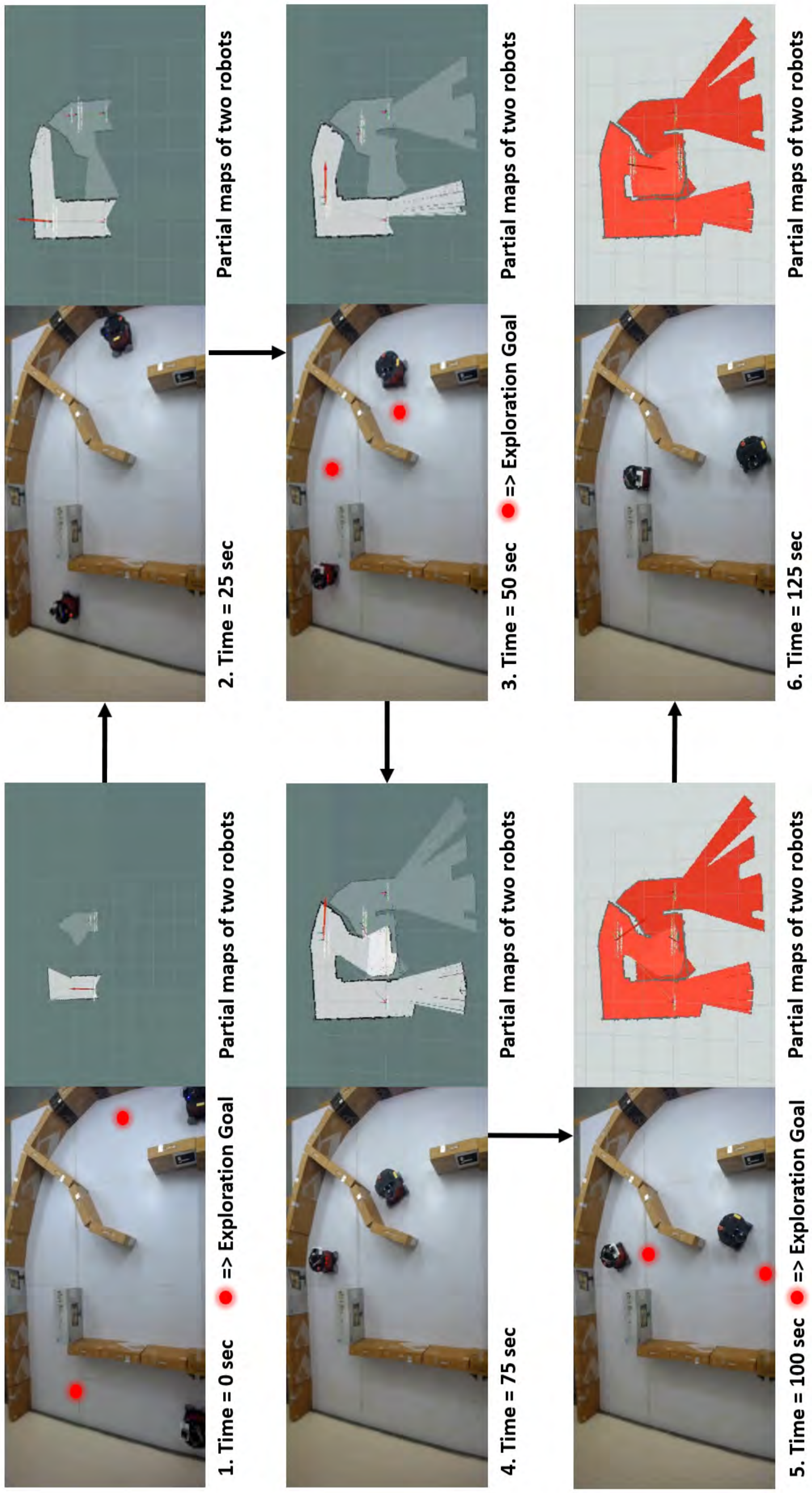


Figure 4.15: An Example Snapshot of the Joint Exploration of Unknown Environments with a Team of Two *FB-VI* Robots

SLAM [271] is employed for mapping, which produces maps of different sizes. Thus, the multirobot_map_merge ROS package cannot be used in its present form. In order to ensure that all robots have the same sized map and origin, we expanded the maps received from Hector-SLAM and resized them. After that, the map-merging procedure described in Section 3.1 is used to merge the maps of multiple robots. The map constructed by the team of two robots is shown in Figure 4.14. Snapshots of the execution of our approach at different time intervals on two *FB-VI* robots in our research lab and the location of the two robots along with their exploration targets are shown in Figure 4.15. Here is the complete exploration process video link [292]. The size of the map is 92 ft^2.

MRFTE is especially capable of quickly dispersing the robot team in an unknown environment, thereby completing the exploration task more efficiently. These observations are validated through simulation by comparing them with seven state-of-the-art approaches. The Centralized MRFTE problem faces the potential issue of a single point of failure as there is no communication model in place. To address this, it is recommended to transition towards implementing a decentralized MRFTE solution on a team of mobile robots in real-world scenarios. This will allow for addressing situations where robot(s) or communication may fail.

4.6 Summary

We have suggested a novel approach, viz., *MRFTE*, for unknown area exploration using a team of mobile robots. Groups of robots use the *frontier tree* data structure to maintain the exploration state of the frontiers, their positions, and the occupancy grid map. *MRFTE* allows the robots who belong to the same group to communicate through their shared frontier tree. When maps of two groups overlap, these teams are integrated, and their frontier trees are merged. Finally, exploration goals are assigned to the individual robots by selecting nodes from the combined frontier tree through a novel strategy. *MRFTE* is especially capable of quickly dispersing the robot team in an unknown environment, thereby completing the exploration task more efficiently. These observations are validated through simulation by comparing them with seven state-of-the-art approaches.

Chapter 5

D-MRFTE: A DECENTRALIZED RELAY-BASED APPROACH FOR MULTI-ROBOT UNKNOWN AREA EXPLORATION

5.1 INTRODUCTION

Environment exploration and mapping is a fundamental task in robotics, which is especially important in dangerous and unsafe environments. For example, robots can be used for the detection of radiation in a nuclear plant, planetary exploration, battlefield exploration for landmine detection, etc. Autonomous multi-robot systems (MRS) have been demonstrated to perform far better than single mobile robots as they can carry out these tasks in parallel by being redundant and exhibiting fault tolerant behavior [293]. Delivery of vital information, i.e., critical sensor data, video footage, pictures, etc., about the environment being explored is paramount. Many times, this data needs to be collected and transmitted wirelessly to a control center or base station for viewing and/or analysis by the humans [50, 171].

In the real world, there are many practical challenges faced by multi-robot systems, for example, dynamically changing environments, limited power sources, and intermittent loss of communication between robots. Therefore, it becomes vital for the designers to suggest coordination approaches that maintain a good balance between exploration and information exchange while coordinating the team of mobile robots. It is not a trivial task to coordinate the robot explorers that are information-hungry in the presence of communication disruptions, i.e., exploring some regions that are rich in information can render the multi-robot network defragmented and vice-versa. Some researchers have addressed this problem for the task of unknown area exploration; for example, in [193], the authors have suggested an algorithm called Bandwidth aware Exploration with a Steiner Traveler (BEST). They have employed robots that deploy relays on precomputed positions which ensure connectivity between new relays and existing relays on unsaturated (in terms of bandwidth) paths. In another work [200], the authors have suggested the use of relay-based communication in multi-robot networks. The relay robots were used for last-mile wireless communication. They gathered data and transmitted it while traveling. The authors have suggested two optimization models, viz., Single-Source-Multiple-Relays (SSMR) and Multiple Source-Single-Relay (MSSR), for handling dense and sparse deployments of relays, respectively.

In our previous chapter [17], which was published in IROS 2022, we proposed an approach viz., MRFTE for rapid exploration of an unknown bounded region using a team of mobile robots. The MRFTE algorithm used frontier tree data structure to maintain the exploration state of the frontiers, the position of those frontiers, and the occupancy grid map of the environment. The robot team creates a shared frontier tree by communicating with their peers. Although MRFTE is capable of quickly dispersing the robot team in an unknown environment, thereby completing the exploration task more efficiently compared to seven state-of-the-art algorithms, it has certain limitations that were also pointed out by the reviewers of IROS 2022. The two main limitations of MRFTE are as follows:

- It is a centralized algorithm, and therefore, it has a single point of failure.
- It works with the assumption that robots can always communicate with their peers without any disruptions, completely ignoring the possibility of intermittent connectivity, which is inevitable due to communication range restrictions.

We worked on the above-mentioned limitations, and in this chapter, we propose a completely decentralized approach named D-MRFTE for unknown area exploration using relay robots. We have validated the efficacy of the proposed approach, viz., D-MRFTE, in the Player/Stage simulator by varying the number of explorer robots and relay robots and by restricting the communication range of the robots in maps of different environmental complexity. The simulation results were compared with our previous approach, i.e., MRFTE [17], on two metrics, i.e., (a) exploration completion time and (b) cumulative distance traveled. We found that DMRFTE is equally capable of dispersing the robot team in the unknown environment and quickly completing the unknown area exploration task. We observed that there is a difference between the performance of the two approaches, such that the D-MRFTE completes the task of exploration slower than its centralized counterpart MRFTE and travels longer distances. However, these differences are not significant.

5.2 The Proposed Approach – DMRFTE

Let R = r_1, ..., r_n represent the available robots that are classified into two categories: an explorer robot (now on-wards explorers) or a relay robot (now on-wards relays). Relays do not explore frontiers but circulate among the explorers and other relays. Each robot r_i must be able to select a destination T_i, where T_i is either a frontier or a predetermined rendezvous point between an explorer and a relay or between multiple relays. Explorers head to the frontiers of the map and are occasionally obligated to rendezvous with a relay. The frequency of such rendezvous is decided by countdown timers, reset on successful communication. We assume that relays do not initiate independently but rather within the communication and exploration range of an explorer, thereby preventing them from becoming stranded. Explorers may begin without relay access.

The robots effectively form a high latency decentralized system, with distributed copies of exploration information for which eventual consistency and completeness must be ensured through meetups. Between two meetups, an explorer autonomously decides on exploration tasks, using its map and the information available to it with respect to other robots' positions and targets. Each robot regularly broadcasts its view of the system's latest state to all other robots in their range. This data exchange uses timestamps to assimilate the latest available information, resulting in a state update mechanism using version vectors [294]. Meetups act as a safety net and set a bound-on latency by ensuring data transfer at periodic intervals if they do not occur in the regular course of events.

Algorithm 5.1 Robot Behaviour

```
while true do
    Update robot state [refer Algorithm-5.2];
    Target assignment [refer Algorithm-5.6];
    if exploration complete then
        break;
    else
        Visit assigned location
        Update robot map
    end if
end while
```

The pseudo-code of the *Algorithm 5.1* is named *robot behaviour or processing loop* executed in the background and uses iteration to determine the next destination to which the robot will travel. It sleeps through the time the robot is in motion and then comes to life once it reaches its destination. It now processes the changes made to the map as a direct result of exploration, followed by the information received by the communicator during the interval. It can then use this updated information to decide on the next destination, an exploration frontier or a rendezvous point. Then, the robots visit the assigned goal. This can be different or the same for relay and explorer robots. The loop is completed when it is discovered that it is impossible to assign the next goal because all of the frontiers have been investigated. All of the necessary communication has occurred.

5.2.1 State Update

Algorithm 5.2 Update Robot State

//Mechanism Updating the robot' state

1: get frontiers on robot's map;
2: *self-synchronize* robot's frontier tree with frontiers using Algorithm 5.3;
3: get latest data from communicator using Algorithm5.11;
4: *reset timers* for robots from whom communicator received messages using Algorithm 5.5;
5: update robot maps and data;
6: verify obtained frontiers and remove obsolete (explored) frontiers;
7: *peer-synchronize* robot with frontiers using Algorithm 5.4;
8: update communicator's frontier advertisement with latest data;

The pseudo-code of the *Algorithm 5.2* is named *state update or update robot state* mechanism. This mechanism updates the robot's state based on sensor readings and control inputs. It plays a crucial role in ensuring the accuracy and reliability of the robot's actions. The data and state variables of the robot are kept up to date by this mechanism, which is used in preparation for goal assignment. It begins by acquiring frontiers on the recently explored map, bypassing the need for updates from peers. This identifies the exploration that the robot itself was engaged in and the new goals generated due to this exploration. After entering these into the tree, the communicator is contacted to continue the process. If the communicator came into contact with any relay robots while in motion, the timers associated with those robots were reset.

After this step, the robot's maps are updated with peer data, and information on other robots' status, location, goals, and frontiers is also updated. If a frontier is to be accepted, it must first be investigated, as it may have already been discovered by a different robot than the one advertising it. The remaining frontiers are acknowledged, along with the owners of each one, and any unclaimed frontiers still present on the map are also noted. In the final step of the process, the communicator is given up-to-date information on the frontiers to advertise during upcoming broadcasts. All frontiers are advertised, regardless of owner, but this ensures that outdated ones do not continue to present themselves.

5.2.1.1 Self Synchronization

Algorithm 5.3 Self-Synchronize

// Synchronize Frontier Tree with Self-Explored Frontiers
Let L_{robot} denote the robot's current location
Let T denote the Tree.

Update L_{robot} in the Frontier tree;
Call the function $tree\_sync(T)$, which follows the tree synchronization procedure laid out by Korb et al. [51];
Let F_{new} be the set of new frontiers discovered by the robot;
for each frontier $f \in F_{new}$ **do**
    Create a new node n_f in the tree, with parent L_{robot} and value f;
    if f is a potential goal owned by the robot **then**
        Mark f down as a goal.
    end if
    if f is a stale frontier **then**
        Mark f down as no longer being a goal;
    end if
end for

The proposed approach uses a frontier tree to store state and to represent exploration on the map, based on the tree suggested by Korb and Schottl [51], with some additional information. Each node also stores a frontier's owner, i.e., the explorer that first discovered the frontier, to synchronize the frontier tree with self-explored frontiers. To keep this tree, as a store of the robot's state up to date, the synchronize function or algorithm 5.3 is executed, which is based on a similar approach used by [51]. New frontiers and their owners are added as potential target nodes. Old frontiers vote for their closest representative on the new map. Provided the new frontier is distinct, if it has a single vote, it is added as a child of that voter. If it has no voters, it is added under the nearest old frontier. Old frontiers that are no longer present are marked as visited and no longer serve as targets.

5.2.1.2 Peer Synchronization

After an explorer receives the latest data, this additional information is integrated into the robot's state, which occurs in the *peer-synchronization* phase. In the peer synchronization phase, the last known locations of the peers are noted to be used as part of the

Algorithm 5.4 Peers-Synchronize

//Synchronize Robot with Frontiers Explored by Peers
Let L be the set of peers' last known locations.
Let F be the set of all frontiers.
Let M be the map containing all possible frontiers.
Let P be the set of all peers.
Let C be the communicator.

```
L ← new_location ∪ L;
for each peer p in P do
    add p's advertised frontiers to F as owned by that p;
end for
for each frontier f in M do
    if f is not owned by any peer then
        add f to F as unowned potential goals;
    end if
end for
for each frontier f in F do
    if f is valid and its owner is not already sharing it then
        add f and its owner to C's list as a shareable frontiers and owner;
    end if
end for
```

dispersion score Eq.(5.1) during frontier selection. Advertisements of frontiers by other robots are received and added to the robot's frontier tree as potential targets under peer ownership. If the map has any further frontiers that have not been claimed by any explorer, these are added as unclaimed potential targets. Adding nodes to the tree happens in the same way described in the *Algorithm 5.4*. Finally, a cumulative list of current frontiers is written up (including all frontiers and their respective owners) for the explorer to advertise.

5.2.1.3 Reset Timer

Algorithm 5.5 Reset Rendezvous Priority for a Peer using Timers

```
if either of self or peer is an explorer then
    reset meet distance timer ← 250 * number of explorers;
    reset meet node timer ← 2 * number of explorers;
else if both self and peer are relays then
    reset meet distance timer ← 0.5 * 250 * Number of explorers;
    reset meet node timer ← 1 * number of explorers;
end if
```

Meetups are scheduled using timers see in algorithm 5.5. For a meetup between a relay and an explorer, in a setup with 'k' explorers, the base timer is set to *2 × k visited nodes* or *250 × k pixel distances* (about *8×k meters*), whichever is covered first. Keeping the reset timer proportional to the number of explorers allows the approach to scale to an arbitrary number of explorers. The relay gets sufficient time to service all explorers and additional relays without being overloaded. Distance timers are preemptively reduced by the path distance that the robot will cover on its way to the destination. Meetups between relays are given higher priorities by setting a lower reset value, i.e., *0.5×250×k pixel-distance* or *1×k nodes visited*, which is half of the reset value for a meetup between a relay and explorer. This causes them to reach zero sooner than in the case of an explorer. Every relay schedule meetup with every other relay it has come in contact with, creating a tightly knit group and helping to converge quickly to consistency. The value of *8 meters* is a variable parameter intended to model the expected average distance covered by a relay moving from one rendezvous to the next. We have calculated empirically by running multiple simulations.

5.2.2 Target Assignment

The target assignment to a relay and explorer is explained in *Algorithm 5.6*. In this chapter, the terms target and goal are synonymous with each other and are used interchangeably. It returns either a meetup location or a frontier. First, it checks if a meetup is scheduled urgently, in which case it finds the destination associated with that meetup. For an explorer, this may or may not return a rendezvous, depending on the current state of the timers. However, for a relay, this check will necessarily return a meetup, except in case of termination. If the meetup is not urgent, then for an explorer, it decides on a frontier and sets that as the next target. If no frontier is found, then the map exploration is complete. The explorer's stopping flag is then set, and the robot tries to 'force' a meetup, trying to access the relays before the meet is naturally scheduled. This step will succeed unless the robot has completed all its communication obligations, as described in the termination conditions. When this, too, fails, it can be assumed that the robot's obligations have been fulfilled, and the robot shuts down. When this fails, That means Now finish-flag set and count-flag increments by one. This count flag is used to check whether all the explorers have finished their work. It is also indicated that the robot has completed its tasks and switches off.

Algorithm 5.6 Target Assignment

// Assign Goal Location Let U is representing the Unforced
Let F is representing the Forced
Let E is representing the Exploration
Let g is representing the Goal
Let m is representing the meetup
Let n.timer is representing the node timer
Let d.timer is representing the distance timer

$U_m \leftarrow$ check for an unforced meeting; [*refer Algorithm-5.7*];
$E_g \leftarrow$ get exploration destination [*refer Algorithm-5.9*];
$F_m \leftarrow$ check for a forced meeting; [*refer Algorithm-5.7*]
if $U_m = True$ **then**
    G $\leftarrow U_m$ use algorithm-5.8;
else if $E_g = True$ **then**
    G $\leftarrow E_g$ use algorithm-5.9;
else if $F_m = True$ **then**
    G $\leftarrow F_m$ use algorithm-5.8;
    set $stoppingflag$
end if
if $stopping_flag == True$ **then**
    set $finish_flag$;
    $Countflag = Countflag + 1$;
    Stop explorer exploration;
end if
set the G as the robot's assigned location;
for all nodes n **do**
    $n.timer \leftarrow n.timer - 1$;
end for
for all distances d from the robot's current location to L **do**
    $d.timer \leftarrow d.timer -$ travel distance;
end for

After deciding on a goal, the procedure communicates this to the robot and then preemptively updates all meet timers. Upon deciding on a target, the *Algorithm 5.6* communicates this to the robot and then preemptively updates all meet timers. Node countdown timers are decremented by one, indicating the completion of one target assignment. Distance timers are preemptively reduced by the path distance that the robot will cover on its way to the destination. To find the distance of a path, the Jump Point Search technique is utilized.

5.2.2.1 Check Meetup

Target assignment for a relay depends entirely on deciding the next robot to meet, while explorers also need to check for scheduled meetups that they are obligated to attend.

Algorithm 5.7 Checks the Meetup Location

// Decide on a Robot for a Rendezvous

if *any distance timers are negative* **then**
    get a robot with the most negative distance timer;
    if *self is a relay* **then**
        reset meet priority with that robot using *algorithm*-5.5;
    end if
    return robot;
else if *any node timers are negative* **then**
    get a robot with the most negative node distance timer;
    if *self is a relay* **then**
        reset meet priority with that robot using *algorithm*-5.5;
    end if
    return robot;
else if *self is regular* **and** *meetup is unforced* **then**
    return no urgent meetups;
end if
if *self is relay* **then**
    if *all known explorers have stopped exploration* and *communicated with self* **then**
        set robot's stopping flag;
    end if
end if
if *no further robots can be met with* **then**
    returns with no meetups;
else
    get a robot with the minimum distance timer;
    if *self is a relay* **then**
        reset meet priority with that robot using *algorithm*-5.5;
    end if
    return robot;
end if

The *Algorithm 5.7* checks the meetup location using the current timer values to identify which peer the robot should meet next, if any. This algorithm is also used to set relays' stopping flags when all the explorers they serve have set their own stopping flags.

As the first thing, a check is made to see if any distance timers have any meetups that have reached an urgent priority, which is indicated by a negative timer value. If this is the case, the most pressing (negative with the highest magnitude) peer is the one that is returned. If no distance timer is urgent, the procedure continues with the node (iteration) timers. If neither of the two cases brings back a meetup, an explorer can continue exploring for another iteration rather than arranging a meeting with a relay. If there is a relay, a peer meetup needs to be arranged even if there is no pressing need for one because this is the only goal selection option that can be chosen.

Similarly, if an autonomous vehicle has completed its exploration, it can also request to meet a relay ahead of schedule, also known as a non-urgent meetup. This type of scenario is called ”forcing” a meetup from the perspective of an explorer. In such situations, the peer to meet is decided based on the node timer's lowest value but the highest relative priority. It is possible that even when forced, no meetups can be scheduled. There are a few stipulations attached to the termination. First, the explorer does not locate any frontiers in the surrounding environment. Second, once the communication between the explorers has been confirmed as successful, when the robot's stopping flag has been set, followed by communication with every relay (for an explorer), or when all explorers have set their stopping flags, followed by communication with every other relay (for a relay). The final step is for the exit counter to catch up to the number of explorers. In this case, the procedure returns without offering a peer to meet with.

It is important to ensure that relays are not rendered inoperable by a single explorer's actions. To achieve this, the priority of the meetup peer is always set to its lowest possible level before making a decision regarding which meetup to use for a relay. This means that if the peer does not show up at the scheduled time, the relay will not wait at the rendezvous point but will move on to the next step. This ensures that the relay can continue providing service to the rest of the system, enabling other explorers to return to the exploration process. As a result, the relay facilitates fast data transfer between explorers.

5.2.2.2 Get Meetup Location

Upon deciding on a peer to meet, the robot heads for the location associated with that rendezvous. This location must be agreed on by both the robots meeting up. While this can be implemented by having the robots communicate and reach a consensus, the proposed approach instead gets the robots to execute *Algorithm 5.8*. To find a meetup location with a peer robot, *Algorithm 5.8* decides between two possible candidate meetup methods, i.e., (a) either each robot goes to its location at the last communication or (b) each robot goes to the first robot's target as of the last communication.

Algorithm 5.8 Get Meetup Location

```
// Obtain Rendezvous Location with a Peer Robot
if self's and peer's locations at last communication are mutually inaccessible then
    return self's location at last communication;
else if self is relay and peer is explorer then
    return peer's goal at last communication;
else if self and peer are both relays then
    return goal at last communication of robot with lower id;
else if self is an explorer then
    return own goal at last communication;
end if
```

The first method is used as a backup and is used only in cases where the maps of the robots communicating are disconnected, in which case it is not possible for both robots to move their way to a common location. Thus, both robots recreate the situation as it occurred the last time; they meet by returning to the same location. This is not used in normal scenarios, as it has the drawback of pulling back robots deep into old territory, incurring travel costs to get there and back to the hot zones where exploration is occurring. The second method is regularly used. Here, it is essential to decide which is the first robot. The first robot's target at the moment of last communication is decided as the rendezvous point that both robots will now head to. In a meetup between an explorer and a relay, the explorer is given priority as the first robot in a bid to reduce its travel cost so that it can quickly return to exploring. In this situation, giving the relay precedence would cause a lazy relay that acts as a static base station that explorers need to return to, resulting in large travel costs. In a meetup between two relays, the relay with the lesser ID number is given precedence as the first robot. ID numbers are assigned to robots uniquely at the start of exploration and are used to identify robots for message passing and within data transfers.

5.2.2.3 Frontier Selection

The explorer selects a frontier using a combination of tree-based, ownership-based, and dispersion-based techniques. *Algorithm 5.9* selects frontier using information on available potential targets, the positions of each robot, and the ownership relations between potential targets and explorers.

Algorithm 5.9 Provide an Exploration Assignment

//Finding the goal node

if *a sibling node of the current position in the tree is a potential goal* **then**
    return sibling node as goal;
else if *nodes owned by self include a potential goal* **then**
    return potential goal node owned by self with optimal dispersion score;
else if *unowned nodes include a potential goal* **then**
    return unowned potential goal node with optimal dispersion score;
else if *nodes owned by other robots include a potential goal* **then**
    return potential goal node regardless of ownership with optimal dispersion score;
else
    no destination found;
end if

As a first step, an explorer tries to explore sibling nodes close to its position in the frontier tree in an effort to explore a local area thoroughly. If such a sibling node is found, it is returned as the selected frontier. If no such node is present, then ownership and dispersion-based techniques are used. The choice of a frontier within an ownership group is decided by the dispersion score (DS) given by Eq.(5.1).

$$DM_{(frontier,robot)} = ED(frontier, robot) - Avg(ED(frontier, peer)) \tag{5.1}$$

Where ED is Euclidean distance, and the average is taken $\forall$peer $\in$ peers.

A lower dispersion score is better for a frontier. The dispersion score prioritizes frontiers closer to the explorer while preferring those that have a greater average distance from other explorers (peers). This directly causes them to fan out from any given starting position and during normal exploration. The dispersion score is used to decide on a frontier within a preference group. All frontiers in the group have their dispersion scores calculated, and the one with the lowest value is picked as the next frontier to be explored. Robots classify potential targets under three categories - self-owned, unowned, and peer-owned, in order of preference during frontier selection.

This helps each robot to demarcate an ‘area’ of its own. The demarcation of the area prevents explorers from converging in a small area. Further frontiers generated while exploring an area are also owned by the same robot, as the robot claims ownership of these frontiers in the self-synchronization phase. Robots also accept other explorers’ areas during peer-synchronization by accepting advertised data specifying frontiers and their owners. An explorer first looks for a potential target in its own area. When these are exhausted, it attempts to set up a new area by heading for an unowned frontier and claiming it as its own. Unowned frontiers are formed when one robot in transit shares its map with another before processing and claiming the frontiers it is generating on the way. To the second robot, which cannot claim this frontier either, such a node is unowned. On opting to explore an unowned frontier, a robot sets itself as the owner of the frontier and explores further from there on within a self-owned area. Finally, when a robot finds itself with no other choice, it joins another peer in the peer’s area by picking a peer-owned frontier. It then sets itself as the frontier’s new owner and tries to carve out a new domain with the new frontiers formed by exploring that frontier. This is a last resort, as it causes explorers to work in an area too small for them to use their independent exploration capacities efficiently. In the following section, the simulation framework is described.

Algorithm 5.10 is used to calculate the dispersion metric that is used to select a frontier within a preference group. First, the dispersion scores of all of the frontiers in the group are calculated, and then the frontier with the lowest value is chosen as the next frontier to be explored. A better score for a frontier is thought to have a lower dispersion metric. The dispersion metric precedes frontiers closer to the robot but prefers frontiers at

Algorithm 5.10 Dispersion_Score

Let self-num is representing the robot-ID
Let the node is representing the frontier-tree node
Let the score is representing the Dispersion Score

$score \leftarrow$ distance between $current\_pos[self\_num]$ and $node$;
$total\_dist \leftarrow 0$;
$counter \leftarrow 0$;
for $i \leftarrow 0$ to $size(current\_pos)$ **do**
    if $i = self\_num$ or $current\_pos[i] = -1$ **then**
        continue;
    end if
    $total\_dist \leftarrow total\_dist+$ distance between $current\_pos[i]$ and $node$
    $counter \leftarrow counter + 1$;
end for
if $counter = 0$ **then**
    $return\ score$;
end if
$avg\_dist \leftarrow total\_dist/counter$;
$score \leftarrow score - avg\_dist$;
$return\ score$;

a greater average distance from other robots (peers). This causes them to spread out from any starting position while they typically explore new territory. After the explorers have located their frontier, the next step is to use the jump point search path planning technique to begin moving toward the frontier. This step is the next in the process. During the time that they are traveling, they come across other explorers and begin the process of communicating with each other.

5.2.3 Communicator

Algorithm 5.11 Communicator Behaviour

while robot is not finished **do**
    sleep for 1000 milliseconds;
    Assimilate all available data;
    Broadcast a snapshot of the current state and data to all robots currently in the range;
end while

The communication loop algorithm, as shown in 5.11, is an independent and persistent component that runs continuously as the robot's communicator. Unlike the processing loop that runs after a particular task, the communication loop runs at a fixed periodic interval. The primary task of the communicator is to assimilate data from received messages and broadcast new information to other robots via their communicators. The robots access their communicators to obtain this information. Communication is restricted to nearby robots under range-based communication or visible robots for line-of-sight communication, depending on the communication model used. In the case of range-based communication, the broadcast step only contacts those robots in range and deposits a message in its message queue. A message contains information about the sender, such as:

1. Robot-ID
2. Type (relay/explorer)
3. Stopping flag (set / not set)
4. Set of robots whose existence is known
5. Timestamp of latest known position data for each robot
6. Latest known position of each robot
7. Latest known goal of each robot
8. Known frontiers
9. Owner for each known frontier
10. Maps

The robot's basic information and state are provided by numbers 1 and 2 respectively. Number 3 is used to check the completion status of all tasks. Number 4 is useful when multiple robots begin in different locations but their existence is known. Number 5 is utilized to gather data on number 6 and its timestamps. Numbers 6 and 7 are used to calculate the dispersion metric with the help of a robot's position and goal. Number 8 is the frontier advertisement and number 9 offers ownership information for it. Number 10 is used to advertise the known maps that need to be assimilated.

Initializing the communicator, it accesses the physical simulator for sending messages. Additionally, the communicator generates a "current" message that conforms to the previously described message structure and serves as a nearly comprehensive record of the communicator's state. This message is regularly updated and broadcasted to provide the most current data available to the communicator. When the robot needs to inquire with the communicator, it requests a snapshot of this message. Semaphores are used to regulate access to this structure due to its heavy cross-thread utilization. The communicator also manages a message queue, a basic buffer that stores messages until they are processed. Inputs to the queue come from the Physical simulation, which manages communication, and the robot associated with the communicator, whose data (a pseudo-message) is produced at regular intervals. After processing the data, the queue can be cleared.

The workflow of the communicator is described in Figure 5.1. The ID number of the robot is also accessible to the communicator, along with a hash set used to monitor frontiers and rendezvous information regarding other robots. This information encompasses details such as whether a given robot should be met, the positions of both the own and peer robots, goals, maps from the previous meeting, and the time of the last meeting. Each peer's information is stored separately and is not included in the shared message(current). The communicator executes a basic main loop once every second, during which it handles incoming messages and disseminates the resultant processed state message via broadcasting.

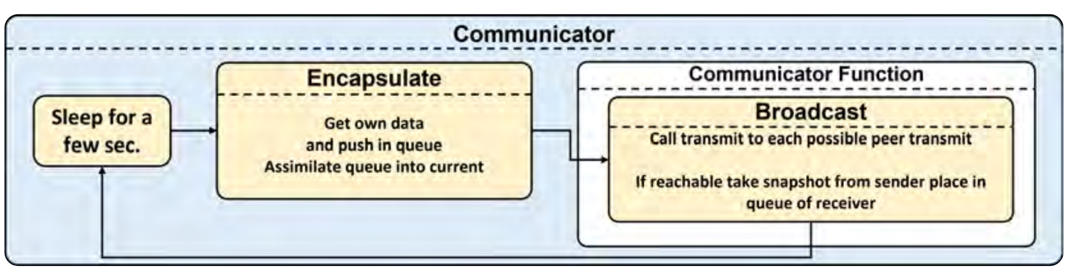


Figure 5.1: Workflow of Communicator

The communication process regularly *encapsulates the latest data* from the message queue, clears it and broadcasts a message with updated information using algorithm 5.12. As a first step, the robot's data is added as a fake message for simplicity to ensure that the robot's information is available during assimilation. The procedure also incorporates robots met for the first time, allowing for the dynamic addition of explorers and relays to the environment. To handle this, the robot will note the peer's existence and set up timers unless both self and peer are explorers.

Algorithm 5.12 Encapsulate Latest Data

Let Q be the message queue.
Let T be the set of rendezvous timers.
Let P be the set of all peers.
Let S be the set of peers whose stopping flag has been set.
Let L be the set of all peers' latest known positions.
Let G be the set of all peers' latest known goals.
Let F be the set of all frontier advertisements.
Let M be the combined map.

```
Q ← own robot's data;
Set rendezvous timers for newly met robots:
Disable rendezvous for peers whose stopping flag has been set:
for each peer p in S do
    T_p ← ∞;
end for
for each peer p do
    log p's latest position and goal in L and G, respectively;
end for
for each message m in Q do
    update L, G, and S with the latest information;
end for
for each frontier advertisement f do
    add f to F;
end for
M ← combine and update maps using L, G, and F;
Q ← ∅
```

The procedure then disables further rendezvous for peers who have set their stop flag, as per the termination procedure. It logs the robot's and the peer's current locations, goals, and maps for future use while deciding rendezvous points. The latest information from across messages is used to obtain the best guess for every robot's position and goals using the timestamps attached to the data. Maps are merged and updated, and received frontier-owner advertisements are added to the communicator's frontier-owner advertisement.

5.2.3.1 Assimilation

Assimilation, also known as Encapsulation, is a process that involves scanning all the available messages and obtaining the latest message system state to be used before processing. During this procedure, data is obtained from the robot as a pseudo-message, which is the robot's input to its communicator. The communicator then uses its access to the robot to read its status, maps, frontiers, and other relevant data. The pseudo-message is timestamped at this point, and it is treated identically to any message.

To combine the data, the procedure first updates rendezvous data. Any robot from whom a message was received in the current iteration is treated as being met. Data transfer is possible in three types: explorer-relay, relay-relay, and explorer-explorer. However, explorer-explorer meets are ignored as no further rendezvous would be planned in this case. These are strictly incidental meets. This opportunity is also used to observe the peer's stopping flag. If this is set, further meetings with that peer who is disabled. Other meeting data stored are meet time (timestamp is taken) and own and peer positions, goals, and maps. Any robots whose data is received for the first time are noted, and their positions, etc., will subsequently be tracked. The latest position data on each robot is picked from across the messages (each has a complete vector of their opinions of each robot's positions). The same is done for goals. The maps advertised by each message are accumulated and merged messages. The robot's stopping flag is observed from the pseudo-message and updated in the current.

Finally, frontiers (and owners) are processed. Every frontier received is hashed (with the product of their coordinates) and compared with a set to check if it has previously been observed. If so, the frontier is ignored; this is useful when robots remain in communication over a stretch and re-advertise the same data when obsolete frontiers continue to be advertised by a robot unaware of their exploration, etc. Otherwise, the frontier-owner pair is added to the communicator's advertisement for further broadcast. The message queue can now be safely cleared.

5.2.3.2 Snapshot

The communicator has a feature that allows it to create a snapshot of its current message. The robot uses this feature to access the data and simulate message transmission using the physical simulator. The copy of the message received by the robot is then added to the receiver's message queue. This process is also used to update the robot's meet urgency parameters. To update the robot's meet urgency parameters, its copy of previous meet timestamps is compared to the communicator's. If they are different (i.e. if the communicator received a communication that the robot is not aware of), the robot's timers for that rendezvous are reset, and the meet time is updated. If the meet validity of any robot has changed (e.g. if a new relay was discovered or a relay found an explorer with a stopping flag set), the robot's copy is also updated accordingly.

5.2.3.3 Advertisement

The communicator maintains an 'advertisement' of frontiers and their owners. The communicator has heard of this set of frontiers, which have not yet been confirmed as explored. When the communicator comes across a new frontier, either as a result of updates from its robot or by reading other frontier advertisements, these are considered for addition to the advertisement. Whenever a frontier is added to the advertisement, it is hashed (using the product of the frontier's coordinates), and the hash is added to a hash set. This prevents the advertisement of multiple copies of the frontier and helps ensure that the communicator does not re-advertise a frontier that was removed (due to exploration). Deletion of frontiers from the advisement occurs when the associated robot observes, upon processing the latest map, that the frontier no longer exists and has been explored.

5.2.4 Physical Simulator

The Physical simulator is a global, independent thread that simulates the communication medium. It does this by maintaining a connectivity matrix between robots. Given the communication model, their positions, and the map, this matrix is queried to check if any two robots can communicate. Upon initialization, the simulator has complete access to the map; this is required to evaluate whether the communication is possible.

5.2.4.1 Workflow of Physical Simulator

The simulator runs a loop every 0.9 seconds (a frequency marginally higher than the communicators' central loop frequency). Each iteration considers every pair of robots, decides whether they may communicate, and updates this as a boolean in the matrix. To do this, it feeds the actual map and the robots' positions to the communication model, which takes a decision. The matrix also prohibits communication with an existing robot. When the matrix is recalculated, a semaphore blocks a broadcast call from getting in.

5.2.4.2 Communication Models

We have incorporated two communication models: range-based and line-of-sight (LOS) communication. In *range-based communication*, we calculate the Euclidean distance between a pair of robots and compare it to a pre-defined limit to establish communication. In *LOS communication*, we first establish a limit on the communication range, and if it is not exceeded, assume a straight line between the two robots. However, communication cannot occur if there are obstructions along the line. Instead of examining every point, we make certain assumptions about the obstacles, such as assuming that each obstacle is at least one meter in any dimension. Therefore, we divide the line into one-meter segments and test each segment's endpoints to determine if they are obstacles. If none of these endpoints act as barriers, communication can occur.

5.2.4.3 Broadcast and Transmit

The simulator object allows independent communicator threads to access broadcast and transmit procedures, using their connection to the simulator to invoke these functions. In a broadcast call, a communicator requests that its current message be sent to all other communicators within range. This process involves repeatedly invoking the transmit function for each robot and potential peer. The transmit function verifies if the desired pair of endpoints can communicate by checking the matrix. If communication is possible, the function takes a *snapshot* of the sender's current message and adds it to the receiver's message queue. Semaphores are utilized to access the matrix and the receiver queue and, by the snapshot function, to access the sender's message.

5.3 Simulation Framework

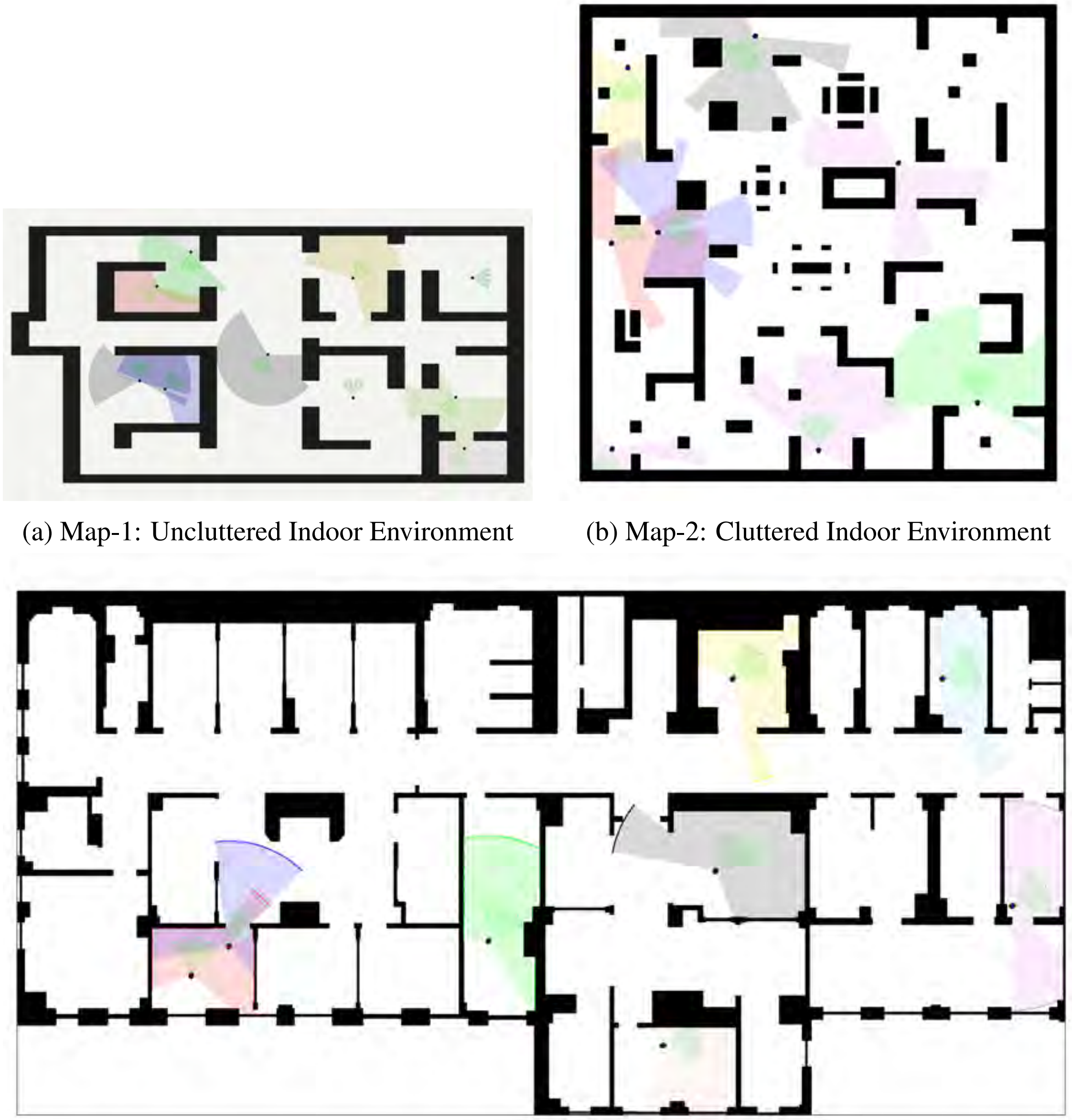
We conducted simulations using the Player/Stage simulator, varying team sizes to include 4, 6, and 8 Pioneer3-AT robots. The Pioneer3-AT is a four-wheel differential drive robot equipped with a 2D LiDAR sensor featuring a 240-degree field of view (FoV). However, during our experiments, we restricted the sensing range to six meters. Simulations were performed on three distinct maps. The first map depicted a large indoor environment measuring 900 x 600 pixels, depicted in Figure 5.2(a). The second map portrayed a more cluttered indoor environment with dimensions of 900 x 900 pixels, shown in Figure 5.2(b). The third map represented a large hospital environment measuring 1800 x 900 pixels and is shown in Figure 5.2(c). In these maps, the white region denoted free space for exploration, while black lines represented walls and obstacles. The map's resolution was set at 0.03 meters per pixel. The robot configurations included two state-of-the-art communication models [60] for inter-robot information sharing, namely:

- Disk-based communication
- Line-of-sight-based communication

We conducted separate analyses for the simulation results of D-MRFTE, focusing on each communication model. The performance of D-MRFTE was measured based on two metrics:

- **Completion Time** - It is the time the explorer robot team takes to explore the unknown environment completely. The exploration is complete when no more frontiers are left for any explorer robot.
- **Cumulative Distance Traveled** - It is the sum of the distance traveled by each explorer robot until the whole map is explored and no more frontiers are visible to any explorer robot.

It should be noted that the completion time comprises sensor data acquisition, processing time, robot navigation, and moving towards the rendezvous for information ex-



(a) Map-1: Uncluttered Indoor Environment (b) Map-2: Cluttered Indoor Environment

(c) Map-3: Hospital Indoor Environment

Figure 5.2: Various Types of Environment

change with relays. Also, robots frequently interfere with one another. Consequently, the planner loses time detouring. The completion time metric is, therefore, not proportional to the distance traveled by the robot team.

During the exploration of the environment by the explorer robots, we assess the redundancy of their coverage area. This assessment is done at each step, but we showcase only a few steps. The explorers' local map is used in this process.

5.4 Complexity Analysis of D-MRFTE

We are only considering the task assignment algorithm in this context.

5.4.1 Algorithm 5.6

It operates in O(m) time complexity, primarily driven by conditional checks and basic operations. The algorithm handles target assignment based on urgency and availability of meetups. Here, m is the number of nodes.

5.4.2 Algorithm 5.7

Operates in O(n) time complexity where n is the number of robots, as it involves scanning robot states and selecting based on timer values and types.

5.4.3 Algorithm 5.8

Operates in O(1) time complexity, focusing on determining meetup locations based on accessibility and robot types.

5.4.4 Algorithm 5.9

Operates in O(m) time complexity where m is the number of nodes, as it involves selecting optimal goals based on node ownership and dispersion scores.

Since Algorithm 5.7 complexity is driven by n (number of robots) and Algorithm 5.9 complexity by m (number of nodes), the overall complexity would be dominated by the more significant of these two factors. Therefore, the overall complexity can be expressed as O(max(n,m)), where n is the number of robots and m is the number of nodes. This notation captures the scalability of the system with respect to handling varying numbers of robots and nodes.

5.5 Results and Discussion

This section displays the results obtained from experiments conducted on three different maps: Uncluttered, Cluttered, and the Hospital indoor environment shown in Figure 5.2(a), Figure 5.2(b), and Figure 5.2(c), respectively. We performed a set of 100 simulation runs for the two communication models, i.e., the Disk-based communication and the Line-of-Sight based communication model involving:

- D-MRFTE+0R: Multiple Explorers without Relays
- D-MRFTE+1R: Multiple Explorers - Single Relay
- D-MRFTE+2R: Multiple Explorers - Two Relays

We compared the results of D-MRFTE+0R, D-MRFTE+1R, and D-MRFTE+2R with two state-of-the-art approaches., i.e., Voronoi Graph-based Segmentation (VGS) [8], and MRFTE [17]. In [17], we showed that MRFTE is superior to seven state-of-the-art algorithms for unknown area exploration while VGS [8] was second best. However, the main limitation of both MRFTE and VGS is that they are centralized and assume continuous global network connectivity. Nevertheless, MRFTE and VGS efficiently complete the unknown area exploration task by achieving global dispersion. Referring to the graphs of Figure 5.3-5.14, one common observation, irrespective of the communication model and the algorithms, is that the completion time and the cumulative distance traveled by the robot team decreases by increasing the number of robots. In the following, we present and discuss the results obtained for the specific communication model by varying the robot team size.

5.5.1 Disk-based Communication

The Disk-based Communication (DBC) model in multi-robot systems allows the robots to communicate and exchange information with their peers within a pre-defined maximum distance, irrespective of the presence of obstacles. The simulation results obtained under this model are shown below. The results for the uncluttered map can be seen in Figure

5.3 and Figure 5.6. For the cluttered map, Figure 5.4 and Figure 5.7 show the outcomes. Lastly, the Hospital map results are presented in Figure 5.5 and Figure 5.8.

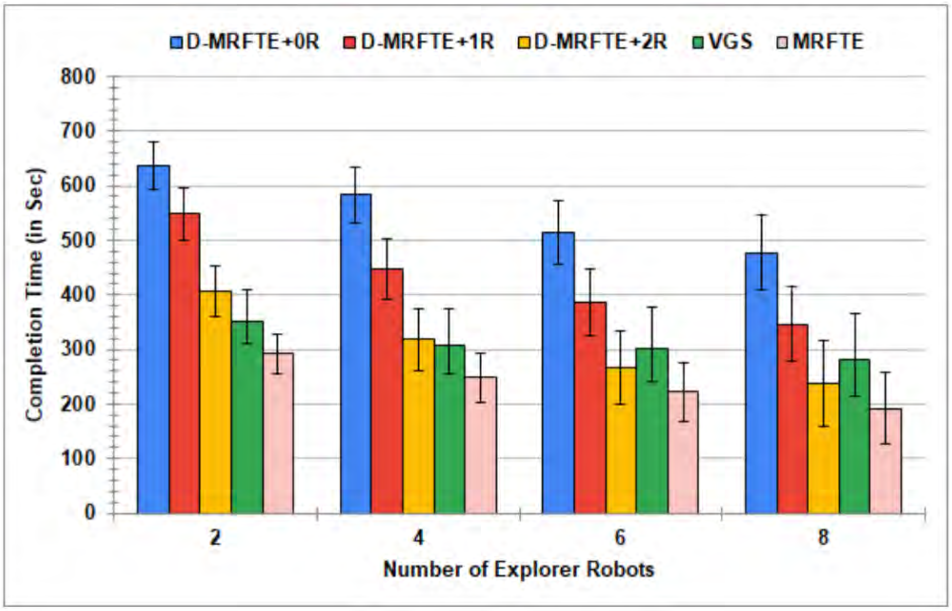


Figure 5.3: Disk Based Communication Model - Exploration Completion Time for Uncluttered Map

We observed that D-MRFTE+0R, a multi-explorer system without relay assistance for information exchange, exhibits the poorest performance compared to all other approaches on both metrics. This occurs when robot peers frequently move out of communication range, leading to intermittent connectivity and, consequently, redundant exploration. Redundant exploration results in increased exploration completion time and cumulative distance traveled. Furthermore, introducing one relay (D-MRFTE+1R) improves the situation compared to D-MRFTE+0R. However, one relay cannot entirely eliminate frequent disconnections and performs less favourably than all other approaches on both metrics. When two relays are introduced (D-MRFTE+2R), interesting results emerge. As shown in Figure 5.3, D-MRFTE+2R with two explorer robots cannot outperform VGS and MRFTE. Nevertheless, with four explorer robots, it matches VGS's performance. With six and eight explorer robots, it surpasses VGS by 13.4% and 14.2%, respectively, but falls short of MRFTE. A similar conclusion can be drawn for cumulative

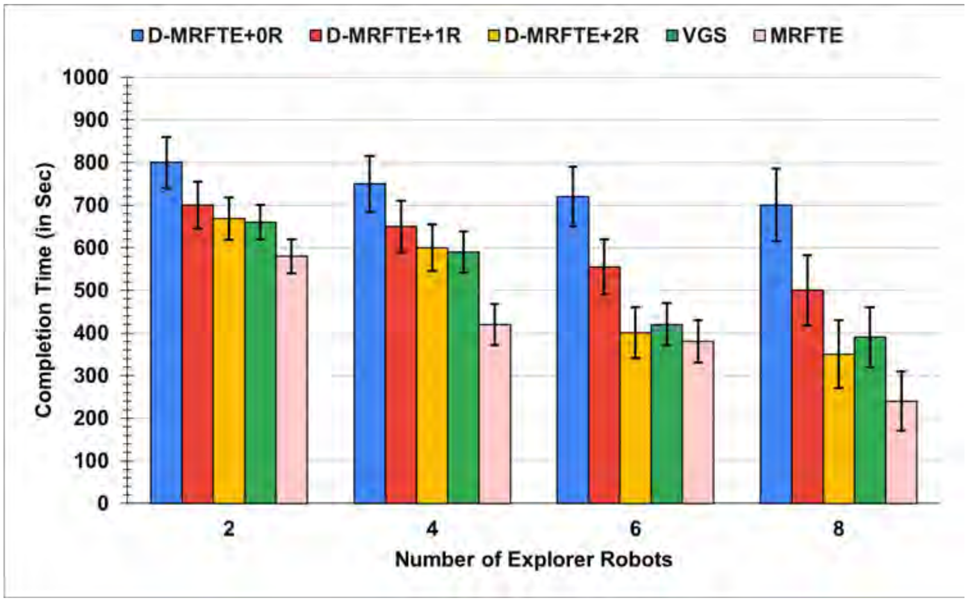


Figure 5.4: Disk Based Communication Model - Exploration Completion Time for Cluttered Map

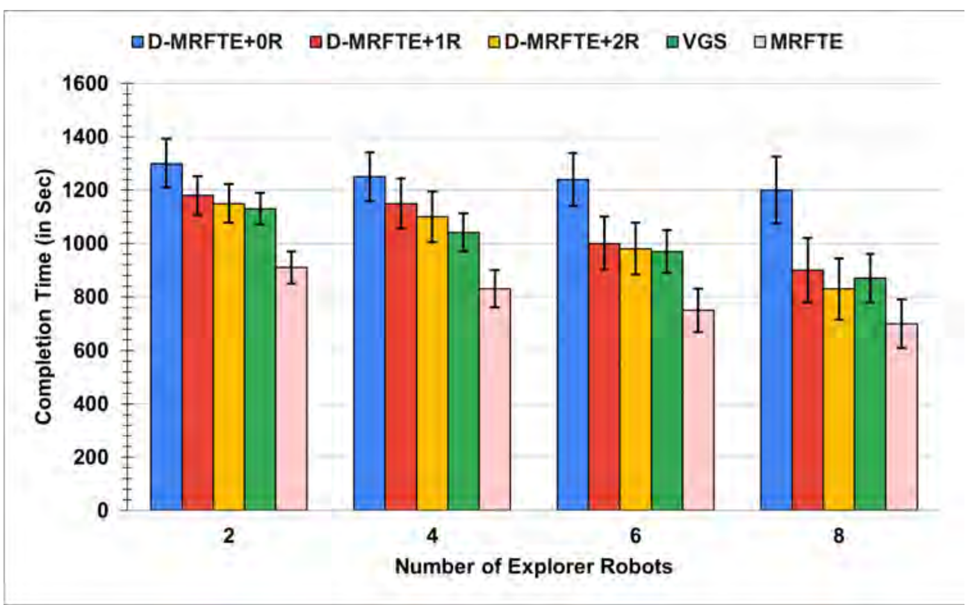


Figure 5.5: Disk Based Communication Model - Exploration Completion Time for Hospital Map

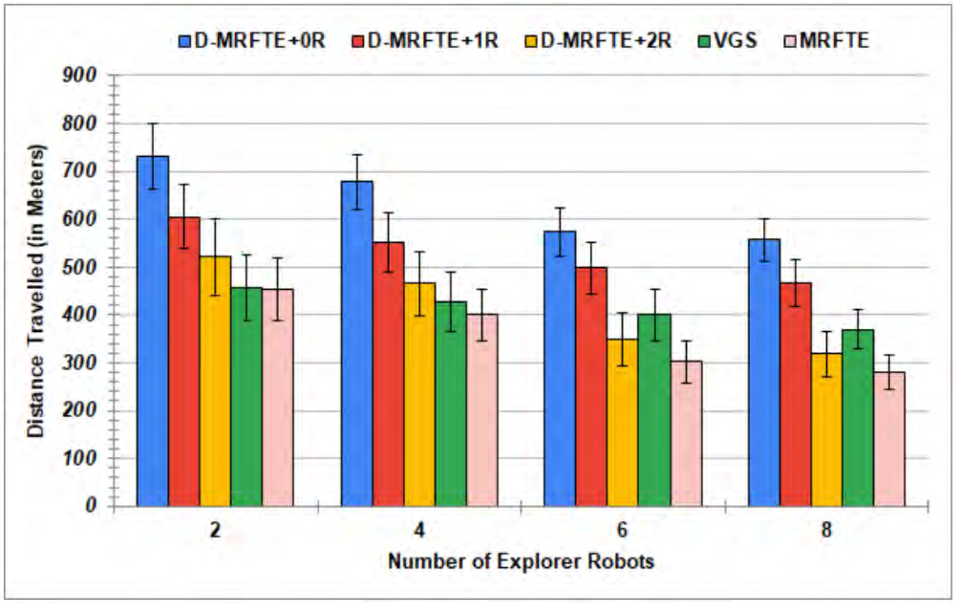


Figure 5.6: Disk Based Communication Model - Cumulative Distance Travelled for Uncluttered Map

distance traveled, as illustrated in Figure 5.6.

In cluttered environments, see in Figure 5.7, D-MRFTE+2R performs almost as well as or better than VGS and MRFTE with six and eight explorers. This improvement, besides addressing intermittent communication issues, is attributed to the increased number of explorer robots and two relay robots, leading to frequent meetings between explorer robots and relays. D-MRFTE+2R successfully mitigates performance penalties suffered by other approaches. Through experimentation, we verified that the proposed approach, D-MRFTE, is equally capable as centralized MRFTE when three relay robots and more than four explorer robots are employed. Thus, D-MRFTE achieves global dispersion. The time required to complete an exploration task depends not only on traversing the area but also on waiting time, idle time, and the time spent by the robots in linear and angular movements. For example, if four robots are assigned a task and one finishes before the others, it must wait for the others, increasing the standard deviation. However, as the number of robots increases, each robot's travel distance decreases, reducing the standard deviation.

In this study, we have provided four maps for eight explorers without relays, which are

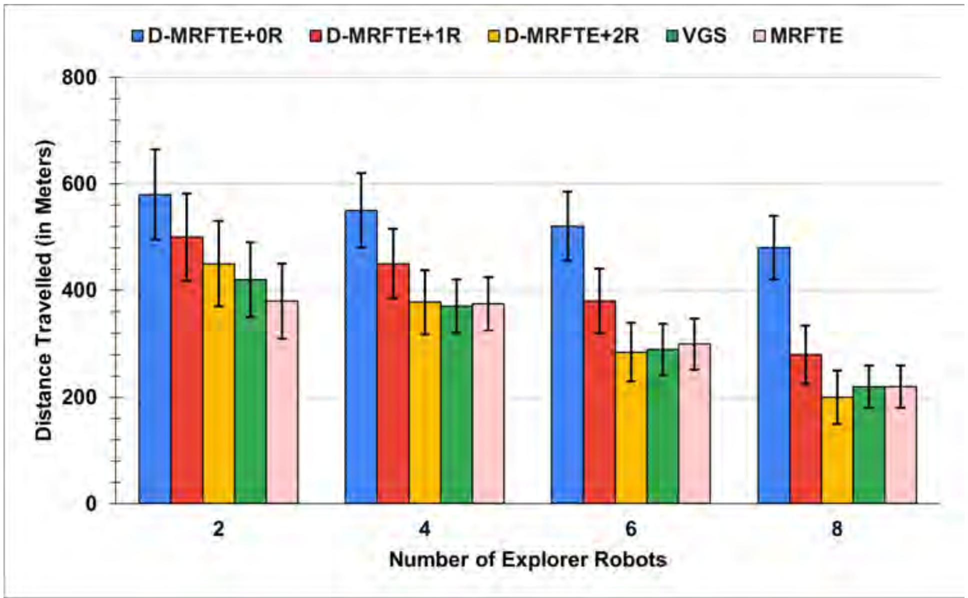


Figure 5.7: Disk Based Communication Model - Cumulative Distance Travelled for Cluttered Map

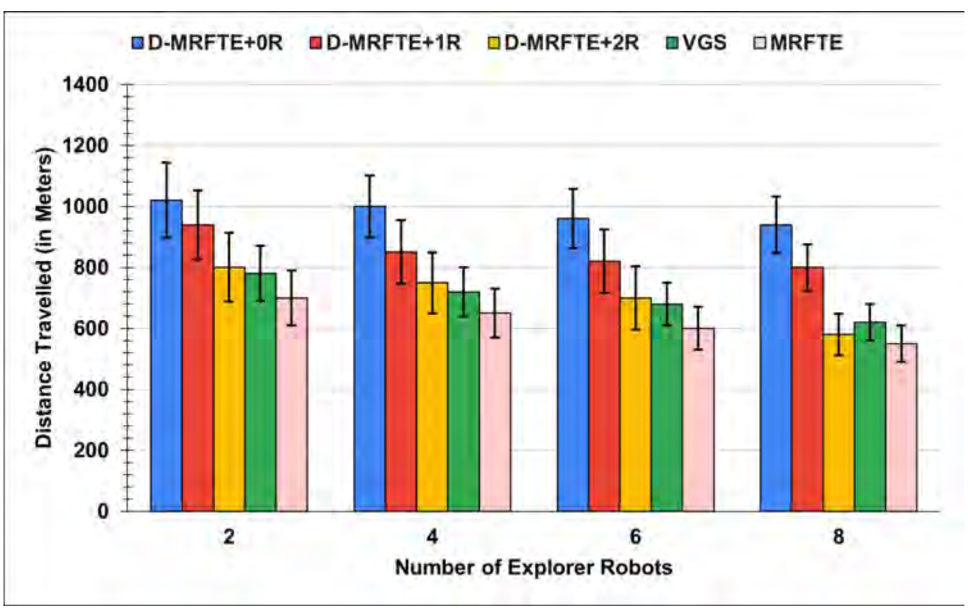
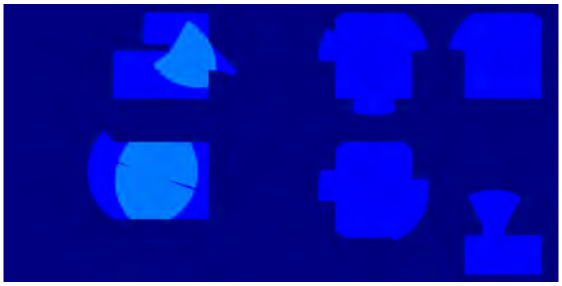
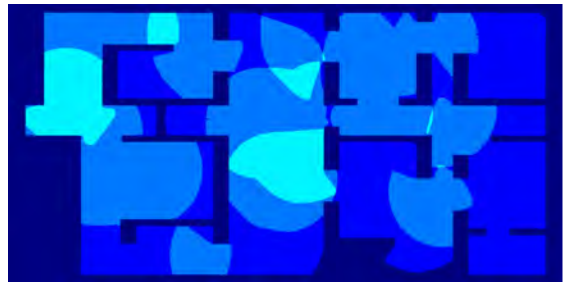
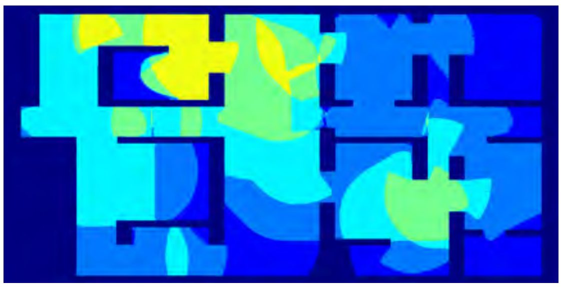


Figure 5.8: Disk Based Communication Model - Cumulative Distance Travelled for Hospital Map


(a) Step-1


(b) Step-5


(c) Step-7

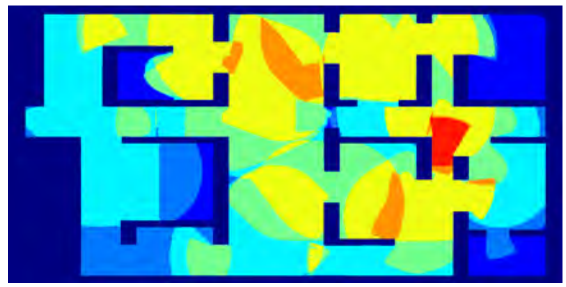
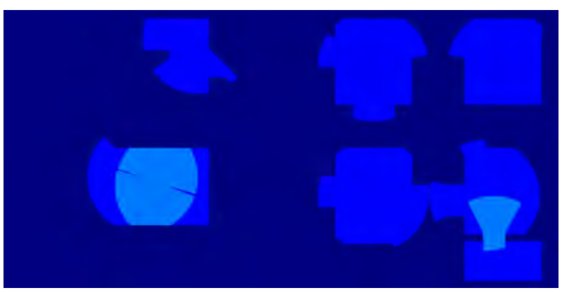
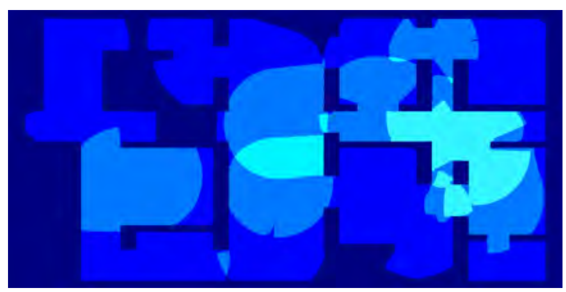
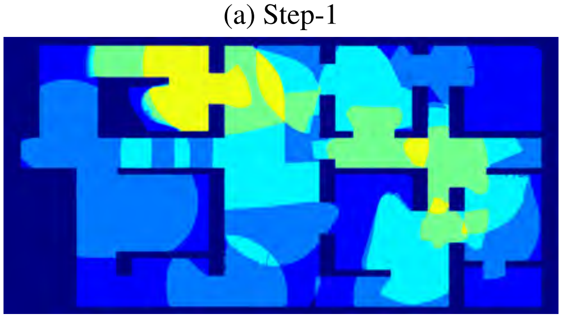

(d) Step-11

Figure 5.9: Number of Steps with Eight Explorers without Relay under Disk Based Communication Model


(a) Step-1


(b) Step-5


(c) Step-7

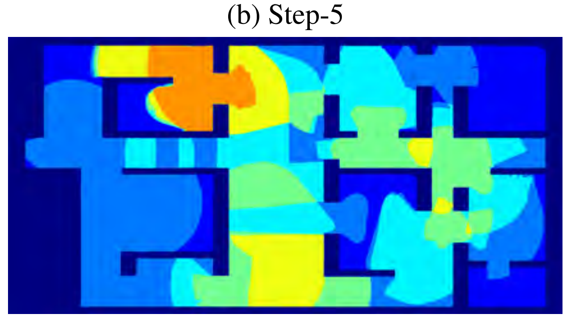

(d) Step-11

Figure 5.10: Number of Steps with Eight Explorers with One Relay under Disk Based Communication Model

shown in Figure 5.9 and Figure 5.17. Additionally, we have provided four maps for eight explorers with a single relay, which are shown in Figure 5.10 and Figure 5.18. The maps have been color-coded to indicate redundant areas. The color scale is as follows: Dark blue represents unexplored or obstacle areas; blue represents areas explored by a single robot; light blue represents areas explored by two robots; the sky color represents areas explored by three robots; light yellow represents areas explored by four robots; yellow represents areas explored by five robots; orange represents areas explored by six robots; red represents areas explored by seven robots; and dark red represents areas explored by all eight robots.

5.5.2 Line-of-Sight based Communication

Line-of-sight-based communication (LoSC) in multi-robot systems enables communication and information exchange only when the distance between two robots is within their communication range, and there are no obstacles blocking their line of sight. The simulation results for this model are presented below, with results for the uncluttered map displayed in Figure 5.11 and Figure 5.14, those for the Cluttered Map in Figure 5.12 and Figure 5.15, and outcomes for the Hospital map in Figure 5.13 and Figure 5.16.

In comparison with the Disk-Based Communication (DBC) model, the LoSC model limits robots' opportunities to communicate and exchange information, leading to redundant exploration. Consequently, algorithms operating under the LoSC model perform poorly in terms of completion time and distance traveled. This is evident in the results shown in Figure 5.11 and Figure 5.14, where DMRFTE+2R with eight explorer robots outperforms VGS by 9.09%. This contrasts with the DBC model, where DMRFTE+2R with six explorer robots sufficed. In environments with numerous obstacles, such as those depicted in Figure 5.12 and Figure 5.15, and in-hospital scenarios, as shown in Figure 5.13 and Figure 5.16, robots face communication difficulties when using LoSC due to the obstacles present. DMRFTE+2R exhibits similar performance levels when compared to both VGS and MRFTE.

Examining the exploration steps, starting from step 1 in Figure 5.17(a) or Figure

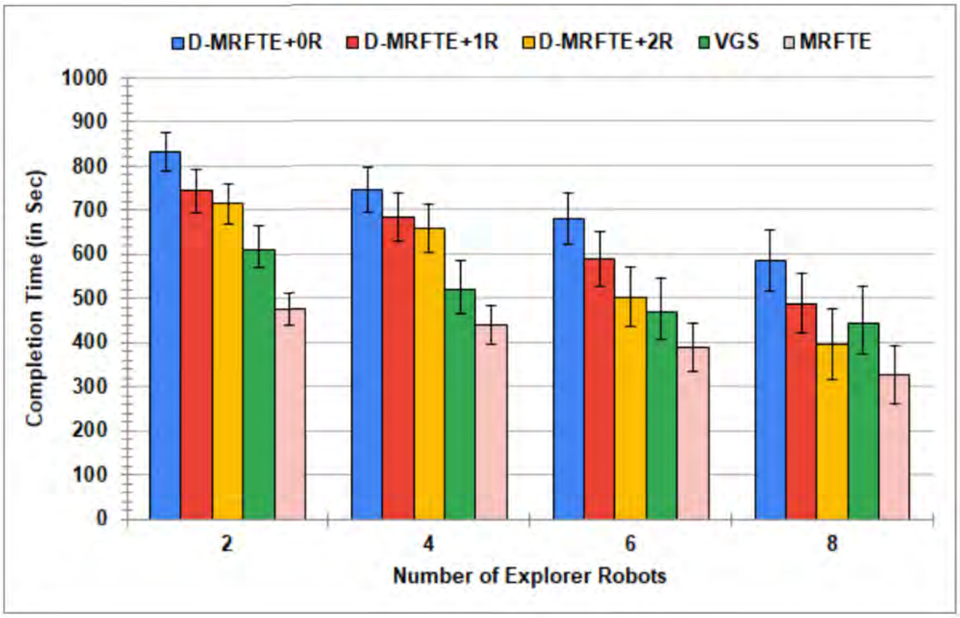


Figure 5.11: Line-of-Sight Based Communication Model - Exploration Completion Time for Uncluttered Map

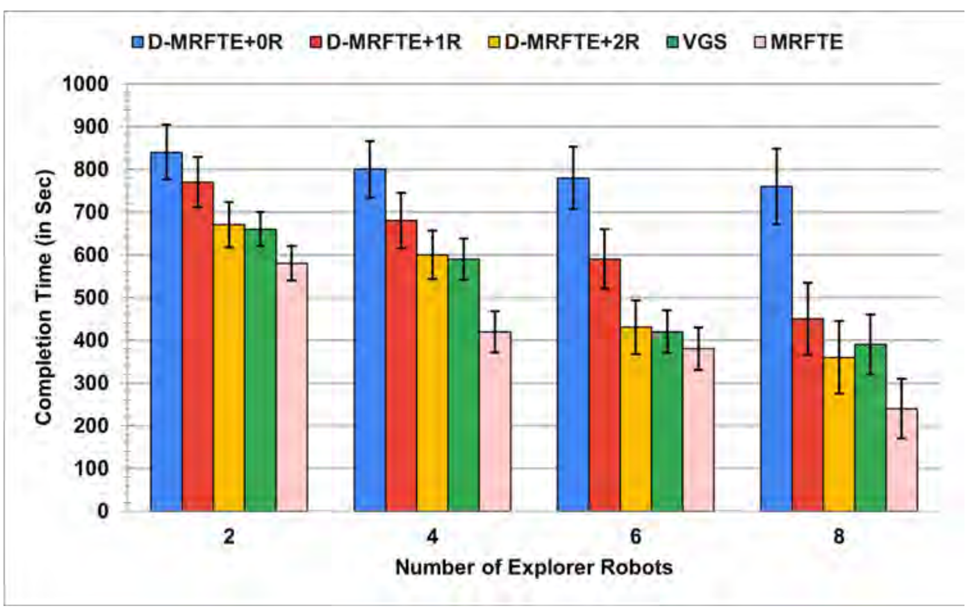


Figure 5.12: Line-of-Sight Based Communication Model - Exploration Completion Time for Cluttered Map

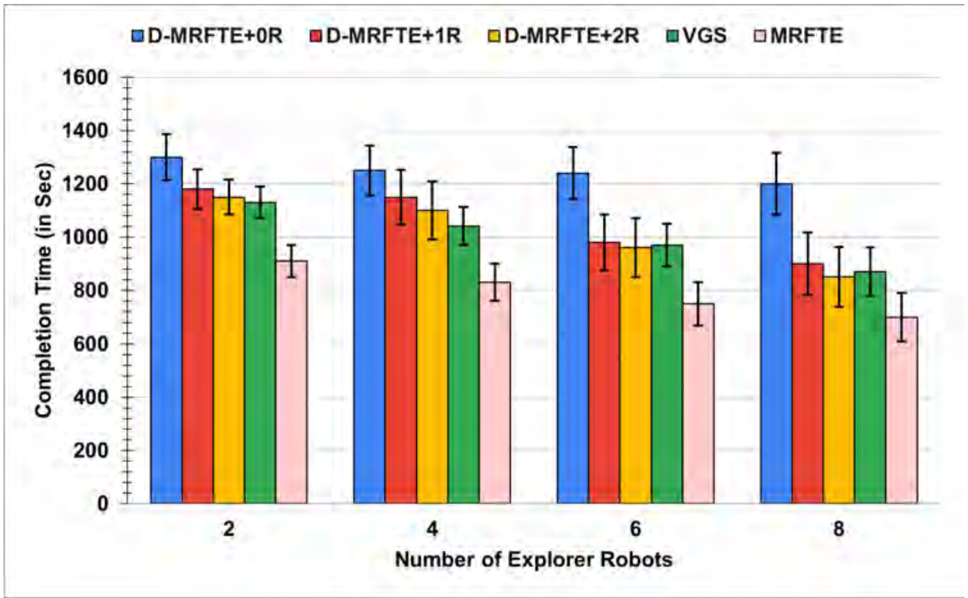


Figure 5.13: Line-of-Sight Based Communication Model - Exploration Completion Time for Hospital Map

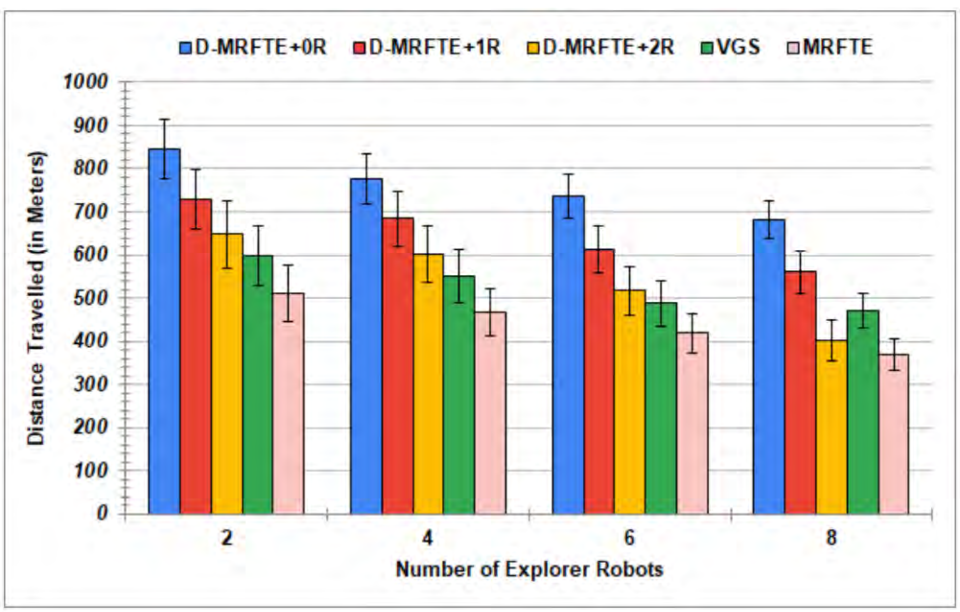


Figure 5.14: Line-of-Sight Based Communication Model - Cumulative Distance Travelled for Uncluttered Map

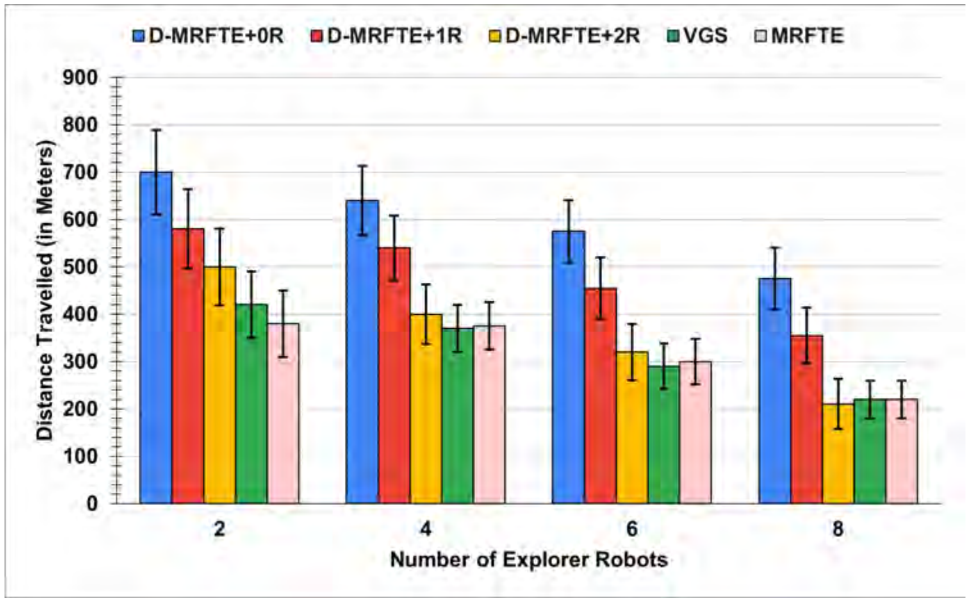


Figure 5.15: Line-of-Sight Based Communication Model - Cumulative Distance Travelled for Cluttered Map

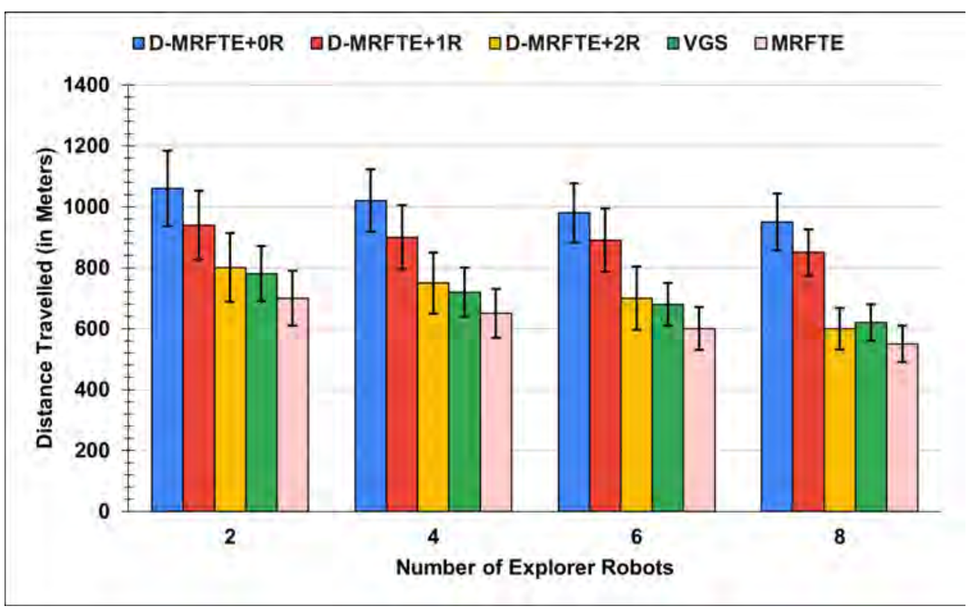
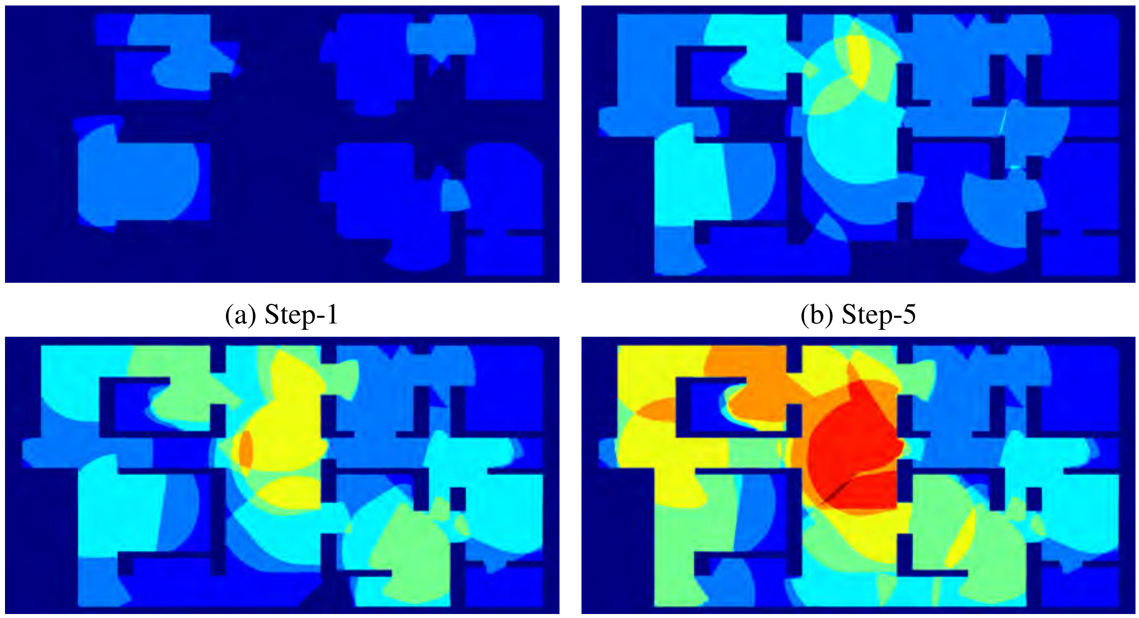


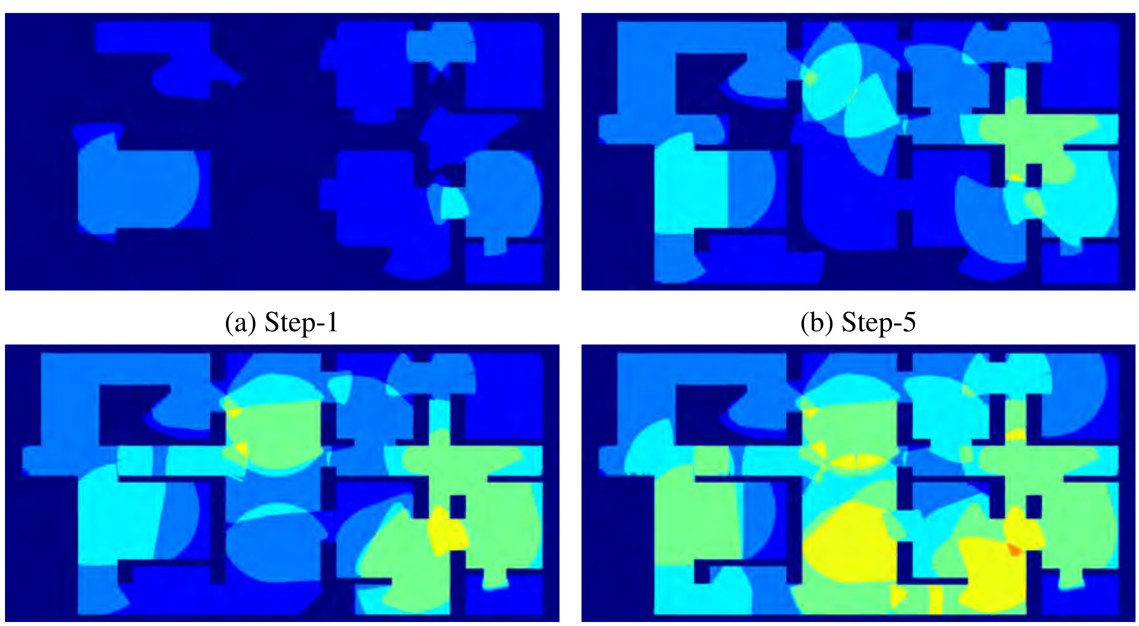
Figure 5.16: Line-of-Sight Based Communication Model - Cumulative Distance Travelled for Hospital Map



(a) Step-1 (b) Step-5

(c) Step-7 (d) Step-11

Figure 5.17: Number of Steps with Eight Explorers without Relay under Line-of-Sight Based Communication Model



(a) Step-1 (b) Step-5

(c) Step-7 (d) Step-11

Figure 5.18: Number of Steps with Eight Explorers with One Relay under Line-of-Sight Based Communication Model

5.9(a), where eight explorers begin from different or nearby locations, progressing to step 5 in Figure 5.17(b) or 5.9(b) with visible sky colors indicating increased redundancy. By step 7 in Figure 5.17(c) or Figure 5.9(c), more yellow is observed, indicating further redundancy from robots exploring previously traversed areas. In the final step-11, observed in orange-red in Figure 5.9(d) and orange, red, and dark red colors in Figure 5.17(d), it suggests that almost all robots have covered that area. Introducing a relay to assist the explorers can reduce redundancy during exploration, as shown in Figure 5.10(d) and Figure 5.18(d). It has been observed that concerning redundancy, the line-of-sight-based approach provides more redundancy than the disk-based approach, with or without a relay.

5.6 Summary

In this chapter, a novel Decentralized Relay Based Approach, viz., D-MRFTE, for unknown area exploration using a team of mobile robots is suggested. This work is an extension of our previous work MRFTE [17] approach. The novelty of the proposed approach is in the process of information dissemination in the multi-robot network under the presence of communication restrictions using relay robots. The relay robots schedule meetups at periodic intervals to ensure eventual consistency and completeness of locally distributed information in the robot network. Version vectors are used as a state update mechanism for the robots. The proposed approach gives superior performance compared with two other state-of-the-art approaches, i.e., [17], and [8] under, i.e., Disk-based and Line-of-Sight-based communication models. We conducted the simulations in the Player/Stage simulator by varying the robot team size. It has been observed that the robots perform the exploration more efficiently under a Disk-based communication model irrespective of the exploration approach.

Chapter 6

CONCLUSIONS AND FUTURE SCOPE OF WORK

6.1 CONCLUSIONS

Multi-robot systems have been observed working together in various situations to accomplish complex tasks that would be challenging or impossible for a single powerful robot to achieve alone. This concept involves dividing smaller sub-problems among individual robots while allowing them to interact and exchange information. Simple robots can be easily constructed and programmed to collaborate toward a common objective, making multi-robot systems a cost-effective solution compared to developing a single expensive robot with numerous capabilities. The group architecture can be centralized, wherein the base station/central controller plans for all the robots. However, they have a single point of failure and can introduce synchronization delays, as the robots might have to wait for other robots to finish their tasks before moving on to the next task. To address these issues, decentralized architectures are preferred. They have no single point of failure, and robots can move independently, eliminating the issue of waiting time. Coordination plays a pivotal role in centralized and decentralized multi-robot systems, enabling multiple robots to execute numerous tasks efficiently. Moreover, our research has unveiled the indispensability of effective inter-robot communication when employing a decentralized or distributed approach to tackle the challenge of exploring terrain with multiple robot systems. This thesis presents an empirical investigation into online terrain coverage (OTC) by multi-robot teams while operating within communication range constraints in Chapter-3. Online terrain coverage is yet another critical task that can be solved using multi-robot systems. The primary requirement of the task is to cover an unknown terrain completely. The state-of-the-art methods assume that communication is omnipresent. This assumption is unrealistic in the real world, so we have dropped it. Further, five state-of-the-art multi-robot OTC approaches were re-implemented in simulation and on a physical robot test bed. We began with conducting a comparative investigation of the following five state-of-the-art approaches for online terrain coverage, viz.,

1. Backtracking Spiral Approach – Cooperative Multi-Robot (BSA-CM) [1]
2. Spiraling and Selective Backtracking (SSB) [2]
3. Boustrophedon and Backtracking mechanism (BoB) [3]
4. Multiple Depth First Search (MDFS) [4]
5. Brick and Mortar (BnM) [4]

The performance of the five approaches mentioned above is empirically compared for the amount of redundant coverage carried out by each one of the algorithms. However, the MDFS and BnM approaches show premature termination for the communication ranges less than 12 meters because of their inability to complete the coverage. Therefore, both of them are excluded from the comparison. For smaller communication ranges of robots, i.e., less than or equal to four meters, the redundant coverage for BSA_CM and SSB approaches exceed the BoB by 4% and 11%, respectively. Whereas, for larger communication ranges, i.e., more than four meters, the SSB and BSA_CM perform better than the BoB. In particular, for a team of four robots, the SSB and the BSA-CM perform better by

14.8% and 9.60%, respectively, than the BoB in terms of redundant coverage. Similarly, for a team of six robots, the performance of SSB and BSA_CM further increased to 20.8% and 18.93% than the BoB.

In Chapter-4, we proposed a centralized coordination algorithm named Multi-Robot Unknown Area Exploration using Frontier Trees (MRFTE). We re-implemented seven other state-of-the-art methods to compare them with MRFTE. MRFTE exhibits superior performance compared with the existing state-of-the-art approaches. Although MRFTE is a highly efficient algorithm, it is centralized and works with the premise that the robot peers can always communicate with the base station. Previously, the Frontier tree data structure was used in single robot exploration to memorize frontiers, their positions, exploration state, and the map. This tree could be queried to decide on further exploration steps. In our approach, we take the concept further for multi-robot exploration. Vital characteristics of the proposed method are enumerated below:

1. Groups of robots use the *frontier tree* data structure to maintain the exploration state of the frontiers, their positions, and the occupancy grid map.

2. *MRFTE* allows the robots who belong to the same group to communicate through their shared frontier tree. When maps of two groups overlap, these teams are integrated, and their frontier trees are merged. Finally, exploration goals are assigned to the individual robots by selecting nodes from the combined frontier tree through a novel strategy.

3. We have compared our work with seven state-of-the-art approaches that are listed below:

   (a) Nearest Frontier Heuristic (NF) [5]

   (b) Information Gain Based Heuristic (D+IG) [6]

   (c) Cost Utility Based Heuristic (C+U) [7]

   (d) Voronoi Graph-Based Segmentation (VGS) [8]

(e) Goal Assignment Using Distance Cost (GADC) [11]

(f) Multiple Rapidly Exploring Random Trees (M-RRT) [9]

(g) Information Driven *RRT* (ID-RRT) [10]

4. We have calculated Exploration Completion Time (ECT) in seconds and Cumulative Distance Travelled (CDT) in meters for all the approaches, including MRFTE in both uncluttered and cluttered environments

Table 6.1: Comparison of Exploration Completion Time (ECT) in seconds and Cumulative Distance Travelled (CDT) in meters for Different Approaches with Different Numbers of Robots in Uncluttered Environment

S.No.	**Approaches**	**2 Robots**		**4 Robots**		**6 Robots**		**8 Robots**	
		ECT	**CDT**	**ECT**	**CDT**	**ECT**	**CDT**	**ECT**	**CDT**
1	**NF**	642	426	585	365	483	303	393	283
2	**D+IG**	523	363	485	345	422	282	364	224
3	**C+U**	503	343	415	325	345	245	322	202
4	****VGS**	381	301	321	261	261	221	193	153
5	**GADC**	462	332	422	292	365	235	284	174
6	**M-RRT**	424	324	341	281	283	243	224	164
7	**ID-RRT**	443	313	382	302	324	254	242	167
8	***MRFTE**	364	284	293	223	221	201	181	131

Table 6.2: Comparison of Exploration Completion Time (ECT) in seconds and Cumulative Distance Travelled (CDT) in meters for Different Approaches with Different Numbers of Robots in Cluttered Environment

S.No.	**Approaches**	**2 Robots**		**4 Robots**		**6 Robots**		**8 Robots**	
		ECT	**CDT**	**ECT**	**CDT**	**ECT**	**CDT**	**ECT**	**CDT**
1	**NF**	954	664	825	524	761	481	742	382
2	**D+IG**	821	581	723	483	644	384	604	344
3	**C+U**	753	463	622	391	521	361	484	324
4	****VGS**	645	425	595	365	422	282	385	245
5	**GADC**	681	521	652	452	622	402	524	304
6	**M-RRT**	855	445	742	422	723	343	643	283
7	**ID-RRT**	884	554	791	481	741	441	684	324
8	***MRFTE**	561	381	424	364	364	293	284	234

In table 6.1 and table 6.2, the symbol *MRFTE represents the best approach, while **VGS represents the second-best approach. From these results, we can conclude that MRFTE yielded better results than other existing state-of-the-art approaches. These improved results are attributed to the fact that MRFTE is especially capable of quickly dispersing the robot team in an unknown environment, completing the exploration task more efficiently.

To address this limitation of MRFTE in Chapter-5, we developed a Decentralized Relay-Based Approach for Multi-Robot Unknown Area Exploration termed DMRFTE, which is at par with MRFTE. This approach is an extension of our previous work, the MRFTE approach [17]. The novelty of this new approach lies in how information is shared in the multi-robot network, especially when the communication range is restricted. To overcome this, relay robots are used to schedule meetups between the robots at regular intervals to ensure that all the information in the robot network is consistent and complete. The robots use version vectors to update their states.

1. In this study, we used two different models to restrict the communication range [121, 148, 158, 163, 164, 191].
   (a) Disk-based communication (DBC) model
   (b) Line-of-Sight-based communication (LoSC) model
2. In addition, We used three distinct robot team compositions:
   (a) D-MRFTE+0R: Multiple explorers without relays
   (b) D-MRFTE+1R: Multiple explorers with one relay
   (c) D-MRFTE+2R: Multiple explorers with multiple relays
3. Finally, we have compared DMRFTE with the following state-of-the-art approaches.
   (a) Voronoi Graph-Based Segmentation (VGS) [8]
   (b) MRFTE [17]
4. We have calculated the Exploration Completion Time (ECT) and Cumulative Distance Travelled (CDT) for all the approaches in a large environment by employing

DBC and LoSC models. Also, we have generated the heat map while the exploration progresses to display the redundant area explored by multiple robots without relays and with one relay. Significant results obtained in the simulation are described below:

- **Disk-based communication model -**
  Regarding ECT, We found that D-MRFTE+0R is the worst-performing method, but D-MRFTE+1R showed some improvements compared to D-MRFTE+0R. Furthermore, D-MRFTE+2R performed worse than VGS and MRFTE even after adding two relays, especially when there were two and four explorer robots. Despite this, when there were six and eight explorer robots, D-MRFTE +2R performed better than VGS by 13.4% and 14.2%, respectively. But, it is inferior to MRFTE. We can conclude that adding more relays with explorer robots in the environment can improve performance. When there were six and eight explorer robots, D-MRFTE+2R outperformed VGS by 12.5% and 11.1%, respectively, in the CDT. However, it was still inferior to MRFTE.

- **Line-of-sight-based communication model -**
  In a multi-robot system utilizing the Line-of-Sight Communication (LoS) model, two robots can only exchange information within each other's communication range and have an unobstructed line of sight. This limitation often results in limited opportunities for communication among the robots, potentially leading to redundant exploration of the environment. Consequently, algorithms operating eunder the LoSC model may not be as efficient regarding completion time and distance covered. We have made some observations regarding ECT and CDT. Our findings indicate that when using the DM-RFTE+2R approach with eight explorer robots, it performs better than the VGS model by 9.09% for ECT and 6.6% for CDT. It is important to note that this performance difference is distinct from the DBC model, where using DMRFTE+2R with only six explorer robots was sufficient.

We found that incorporating relays alongside explorers for area exploration in a communication range-restricted environment can significantly reduce redundancy and accomplish the exploration task more efficiently than solely relying on multiple explorers without a relay.

6.2 SCOPE FOR FUTURE WORK

In the future, we envision expanding upon this thesis to address the following limitations inherent in the existing research conducted in this thesis:

- In terms of communication, our current system does not incorporate any communication models for addressing packet drops/losses during the communication process. Key factors such as fading (interference caused by objects like walls in the environment), path loss (communication degradation due to long distances between robots), and bandwidth saturation are yet to be addressed.

- In this thesis, we have studied and suggested coordination algorithms for a team of homogeneous autonomous robots. In our future work, we intend to design and implement decentralized coordination approaches for a fleet of heterogeneous mobile robots.

- Also, we will focus on addressing real-world scenarios involving robot malfunctions (Byzantine failures) and communication failures.

REFERENCES

[1] E. Gerlein and E. González, "Multirobot cooperative model applied to coverage of unknown regions," *Multi-Robot Systems, Trends and Development*, pp. 109–130, 2011.

[2] A. Gautam, A. Richhariya, V. S. Shekhawat, and S. Mohan, "Experimental evaluation of multi-robot online terrain coverage approach," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 1183–1189.

[3] H. H. Viet, V.-H. Dang, S. Choi, and T. C. Chung, "Bob: an online coverage approach for multi-robot systems," *Applied Intelligence*, vol. 42, no. 2, pp. 157–173, 2015.

[4] E. Ferranti, N. Trigoni, and M. Levene, "Brick& mortar: an on-line multi-agent exploration algorithm," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 761–767.

[5] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the Second International Conference on Autonomous Agents*. New York, NY, USA: Association for Computing Machinery, 1998, p. 47–53.

[6] H. H. González-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[7] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[8] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1160–1165.

[9] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1396–1402.

[10] J. M. Pimentel, M. S. Alvim, M. F. M. Campos, and D. G. Macharet, "Information-driven rapidly-exploring random tree for efficient environment exploration," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 2, pp. 313–331, Aug 2018.

[11] J. Faigl, M. Kulich, and L. Přeučil, "Goal assignment using distance cost in multi-robot exploration," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3741–3746.

[12] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, pp. 375–397, 1996.

[13] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2015–2028, 2004.

[14] B. P. Gerkey and M. J. Matarić, "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.

[15] C. Eschke, M. K. Heinrich, M. Wahby, and H. Haman, "Self-organized adaptive paths in multi-robot manufacturing: reconfigurable and pattern-independent fibre deployment," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4086–4091.

[16] H. El-Hussieny, S. F. Assal, and M. Abdellatif, “Robotic exploration: new heuristic backtracking algorithm, performance evaluation and complexity metric,” *International Journal of Advanced Robotic Systems*, vol. 12, no. 4, p. 33, 2015.

[17] A. Soni, C. Dasannacharya, A. Gautam, V. S. Shekhawat, and S. Mohan, “Multi-robot unknown area exploration using frontier trees,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9934–9941.

[18] R. Neouchi, H. Zhang, and R. Elio, “Distance based communication in the surveillance task in a multi-robot system,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2529–2535.

[19] J. De Hoog, S. Cameron, and A. Visser, “Dynamic team hierarchies in communication-limited multi-robot exploration,” in *2010 IEEE Safety Security and Rescue Robotics*. IEEE, 2010, pp. 1–7.

[20] P. Chand and D. A. Carnegie, “Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots,” *Robotics and autonomous Systems*, vol. 61, no. 6, pp. 565–579, 2013.

[21] M. Suganthi, M. Mahanty, H. Madan, K. Harisha, S. Menon *et al.*, “Autonomous robot system development for health indication and monitoring of crops using ai-ml-based system,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2023, pp. 413–416.

[22] S. Grayson, “Search & rescue using multi-robot systems,” *School of Computer Science and Informatics, University College Dublin*, pp. 1–14, 2014.

[23] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, “Collaborative multi-robot

systems for search and rescue: Coordination and perception," *arXiv preprint arXiv:2008.12610*, 2020.

[24] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in *2012 IEEE 7th international conference on industrial and information systems (ICIIS)*. IEEE, 2012, pp. 1–5.

[25] A. Quattrini Li, P. K. Penumarthi, J. Banfi, N. Basilico, J. M. O'Kane, I. Rekleitis, S. Nelakuditi, and F. Amigoni, "Multi-robot online sensing strategies for the construction of communication maps," *Autonomous Robots*, vol. 44, pp. 299–319, 2020.

[26] F. Gul, I. Mir, W. Rahiman, and T. U. Islam, "Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm," *IEEE Access*, vol. 9, pp. 22 774–22 787, 2021.

[27] Z. Zhou, J. Liu, and J. Yu, "A survey of underwater multi-robot systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 1–18, 2021.

[28] D. Payton, R. Estkowski, and M. Howard, "Compound behaviors in pheromone robotics," *Robotics and Autonomous Systems*, vol. 44, no. 3-4, pp. 229–240, 2003.

[29] J. De Hoog, S. Cameron, and A. Visser, "Role-based autonomous multi-robot exploration," in *2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*. IEEE, 2009, pp. 482–487.

[30] K. Masaba and A. Q. Li, "Gvgexp: Communication-constrained multi-robot exploration system based on generalized voronoi graphs," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2021, pp. 146–154.

[31] A. Bolu and Ö. Korçak, "Adaptive task planning for multi-robot smart warehouse," *IEEE Access*, vol. 9, pp. 27 346–27 358, 2021.

[32] R. D'Andrea, "Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 638–639, 2012.

[33] Y. Huang, Y. Zhang, and H. Xiao, "Multi-robot system task allocation mechanism for smart factory," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. IEEE, 2019, pp. 587–591.

[34] N. Teslya, A. Smirnov, A. Ionov, and A. Kudrov, "Multi-robot coalition formation for precision agriculture scenario based on gazebo simulator," in *Proceedings of 15th International Conference on Electromechanics and Robotics" Zavalishin's Readings" ER (ZR) 2020, Ufa, Russia, 15–18 April 2020*. Springer, 2021, pp. 329–341.

[35] A. Dutta, S. Roy, O. P. Kreidl, and L. Bölöni, "Multi-robot information gathering for precision agriculture: Current state, scope, and challenges," *IEEE Access*, vol. 9, pp. 161 416–161 430, 2021.

[36] A. V. Nazarova and M. Zhai, "The application of multi-agent robotic systems for earthquake rescue," *Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms*, pp. 133–146, 2020.

[37] K.-C. Ma, Z. Ma, L. Liu, and G. S. Sukhatme, "Multi-robot informative and adaptive planning for persistent environmental monitoring," in *Distributed Autonomous Robotic Systems: The 13th International Symposium*. Springer, 2018, pp. 285–298.

[38] C. M. Gifford, R. Webb, J. Bley, D. Leung, M. Calnon, J. Makarewicz, B. Banz, and A. Agah, "A novel low-cost, limited-resource approach to autonomous multi-robot exploration and mapping," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 186–202, 2010.

[39] Z. Li, A. V. Barenji, J. Jiang, R. Y. Zhong, and G. Xu, "A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand," *Journal of Intelligent Manufacturing*, vol. 31, pp. 469–480, 2020.

[40] S. Premvuti and S. Yuta, "Consideration on the cooperation of multiple autonomous mobile robots," in *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. IEEE, 1990, pp. 59–63.

[41] D. Barnes and J. Gray, "Behaviour synthesis for co-operant mobile robot control," in *International Conference on Control 1991. Control '91*, 1991, pp. 1135–1140 vol.2.

[42] M. J. Mataric, "Behaviour-based control: Examples from navigation, learning, and group behaviour," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 323–336, 1997.

[43] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.

[44] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," *Current Robotics Reports*, vol. 3, no. 4, pp. 213–225, 2022.

[45] A. Gautam, B. Jha, G. Kumar, J. K. Murthy, S. A. Ram, and S. Mohan, "Fast: Synchronous frontier allocation for scalable online multi-robot terrain coverage," *Journal of Intelligent & Robotic Systems*, vol. 87, pp. 545–564, 2017.

[46] V. Spirin and S. Cameron, "Rendezvous through obstacles in multi-agent exploration," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*. IEEE, 2014, pp. 1–6.

[47] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, pp. 117–136, 2001.

[48] E. A. Jensen and M. L. Gini, "Rolling dispersion for robot teams." in *IJCAI*, 2013, pp. 2473–2479.

[49] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 373–382.

[50] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.

[51] R. Korb and A. Schöttl, "Exploring unstructured environment with frontier trees," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 3, pp. 617–628, 2018.

[52] I. F. Chaile-Alfaro and L. Ribas-Xirgo, "Running agent-based-models simulations synchronized with reality to control transport systems," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 57, no. 2, pp. 452–465, 2016.

[53] R. T. Vaughan, K. Stoy, G. S. Sukhatme, and M. J. Mataric, "Lost: Localization-space trails for robot teams," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 796–812, 2002.

[54] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems: Part i: Transportation between customers and depot," *Journal für Betriebswirtschaft*, vol. 58, pp. 21–51, 2008.

[55] A. Campo and M. Dorigo, "Efficient multi-foraging in swarm robotics," in *European Conference on Artificial Life*. Springer, 2007, pp. 696–705.

[56] S. Krannich and E. Maehle, "Analysis of spatially limited local communication for multi-robot foraging," in *FIRA RoboWorld Congress*. Springer, 2009, pp. 322–331.

[57] M. Andries, "Object and human tracking, and robot control through a load sensing floor," Ph.D. dissertation, Université de Lorraine, 2015.

[58] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME transactions on mechatronics*, vol. 14, no. 2, pp. 163–175, 2009.

[59] L. E. Parker, "Multi-robot team design for real-world applications," in *Distributed Autonomous Robotic Systems 2*. Springer, 1996, pp. 91–102.

[60] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2017.

[61] P. E. Rybski, S. A. Stoeter, M. D. Erickson, M. Gini, D. F. Hougen, and N. Papanikolopoulos, "A team of robotic agents for surveillance," in *Proceedings of the fourth international conference on autonomous agents*, 2000, pp. 9–16.

[62] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for searching environments in rescue operations," *Autonomous Robots*, vol. 31, no. 4, pp. 401–417, 2011.

[63] M. Juliá, O. Reinoso, A. Gil, M. Ballesta, and L. Payá, "A hybrid solution to the multi-robot integrated exploration problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 473–486, 2010.

[64] S. Ismail and L. Sun, "Decentralized hungarian-based approach for fast and scalable task allocation," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 23–28.

[65] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[66] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, pp. 113–126, 2001.

[67] D. W. Gage, "Command control for many-robot systems," Naval Command Control and Ocean Surveillance Center Rdt And E Div San Diego CA, Tech. Rep., 1992.

[68] Y. Kantaros and M. M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks," *Automatica*, vol. 63, pp. 209–220, 2016.

[69] L. Ludwig and M. Gini, "Robotic swarm dispersion using wireless intensity signals," in *Distributed autonomous robotic systems 7*. Springer, 2006, pp. 135–144.

[70] M. A. Batalin and G. S. Sukhatme, "The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 661–675, 2007.

[71] I. Rekleitis, A. P. New, and H. Choset, "Distributed coverage of unknown/unstructured environments by mobile sensor networks," in *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*. Springer, 2005, pp. 145–155.

[72] J. A. Rogge and D. Aeyels, "A strategy for exploration with a multi-robot system," in *Informatics in Control, Automation and Robotics*. Springer, 2009, pp. 195–206.

[73] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity management in mobile robot teams," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008, pp. 1525–1530.

[74] P. Mukhija, K. M. Krishna, and V. Krishna, "A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4806–4811.

[75] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[76] J. G. Rogers, C. Nieto-Granda, and H. I. Christensen, "Coordination strategies for multi-robot exploration and mapping," in *Experimental Robotics*. Springer, 2013, pp. 231–243.

[77] N. Hazon, F. Mieli, and G. A. Kaminka, "Towards robust on-line multi-robot coverage," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1710–1715.

[78] D. Latimer, S. Srinivasa, V. Lee-Shue, S. Sonne, H. Choset, and A. Hurst, "Towards sensor based coverage with robot teams," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 961–967.

[79] A. Gautam, J. K. Murthy, G. Kumar, S. A. Ram, B. Jha, and S. Mohan, "Cluster, allocate, cover: An efficient approach for multi-robot coverage," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 197–203.

[80] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, 2016.

[81] R. Doriya, S. Mishra, and S. Gupta, "A brief survey and analysis of multi-robot communication and coordination," in *International Conference on Computing, Communication & Automation*. IEEE, 2015, pp. 1014–1021.

[82] P. Fazli, A. Davoodi, and A. K. Mackworth, "Multi-robot repeated area coverage," *Autonomous robots*, vol. 34, no. 4, pp. 251–276, 2013.

[83] E. A. Jensen, E. Nunes, and M. Gini, "Communication-restricted exploration for robot teams," in *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[84] D. Mitchell, M. Corah, N. Chakraborty, K. Sycara, and N. Michael, "Multi-robot long-term persistent coverage with fuel constrained robots," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1093–1099.

[85] E. Ferranti, N. Trigoni, and M. Levene, "Rapid exploration of unknown areas through dynamic deployment of mobile and stationary sensor nodes," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 2, pp. 210–243, 2009.

[86] M. Mamei and F. Zambonelli, "Pervasive pheromone-based interaction with rfid tags," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 2, no. 2, pp. 4–es, 2007.

[87] T. Mizuguchi and K. Sugawara, "Proportion regulation in task allocation systems," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 89, no. 10, pp. 2745–2751, 2006.

[88] B. L. Wellman, S. Dawson, J. de Hoog, and M. Anderson, "Using rendezvous to overcome communication limitations in multirobot exploration," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2011, pp. 2401–2406.

[89] Y. Pei, M. W. Mutka, and N. Xi, "Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 5460–5465.

[90] A. D. Kshemkalyani and F. Ali, "Efficient dispersion of mobile robots on graphs," in *Proceedings of the 20th International Conference on Distributed Computing and Networking*, 2019, pp. 218–227.

[91] R. Pandey, A. K. Singh, and K. M. Krishna, "Multi-robot exploration with communication requirement to a moving base station," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2012, pp. 823–828.

[92] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 476–481.

[93] F. Amigoni and A. Gallo, "A multi-objective exploration strategy for mobile robots," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3850–3855.

[94] M. Kulich, J. Faigl, and L. Přeučil, "On distance utility in the exploration task," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4455–4460.

[95] S. Argamon-Engelson, S. Kraus, and S. Sina, "Utility-based on-line exploration for repeated navigation in an embedded graph," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 267–284, 1998.

[96] S. Kasperski, J. Wollenberg, and M. Eich, "Evaluation of cooperative exploration strategies using full system simulation," in *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE, 2013, pp. 1–6.

[97] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the efficiency of frontier-based exploration strategies," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, 2010, pp. 1–8.

[98] D. Puig, M. Garcia, and L. Wu, "A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 635–653, 2011.

[99] K. Hungerford, P. Dasgupta, and K. Guruprasad, "Distributed, complete, multi-robot coverage of initially unknown environments using repartitioning," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 2014, pp. 1453–1454.

[100] N. Asgharbeygi and A. Maleki, "Geodesic k-means clustering," in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[101] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[102] B. Pik Lik Lau, B. J. Y. Ong, L. K. Y. Loh, R. Liu, C. Yuen, G. S. Soh, U. Tan *et al.*, "Multi-agv's temporal memory-based rrt exploration in unknown environment," *arXiv e-prints*, pp. arXiv–2207, 2022.

[103] R. Korb and A. Schöttl, "Exploring unstructured environment with frontier trees," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 3, pp. 617–628, 2018.

[104] Z. Zhang, J. Yu, J. Tang, Y. Xu, and Y. Wang, "Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 794–10 801, 2022.

[105] L. Clark, J. A. Edlund, M. S. Net, T. S. Vaquero, and A.-a. Agha-Mohammadi, "Propem-l: Radio propagation environment modeling and learning for communication-aware multi-robot exploration," *arXiv preprint arXiv:2205.01267*, 2022.

[106] L. Clark, J. Galante, B. Krishnamachari, and K. Psounis, "A queue-stabilizing framework for networked multi-robot exploration," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2091–2098, 2021.

[107] Y. Gao, Y. Wang, X. Zhong, T. Yang, M. Wang, Z. Xu, Y. Wang, Y. Lin, C. Xu, and F. Gao, "Meeting-merging-mission: A multi-robot coordinate framework for large-scale communication-limited exploration," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 700–13 707.

[108] G. A. Cardona, D. Yanguas-Rojas, M. F. Arevalo-Castiblanco, and E. Mojica-Nava, "Ant-based multi-robot exploration in non-convex space without global-connectivity constraints," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2065–2070.

[109] F. Amigoni, J. Banfi, N. Basilico, I. Rekleitis, and A. Quattrini Li, "Online update of communication maps for exploring multirobot systems under connectivity constraints," in *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 513–526.

[110] F. Kobayashi, S. Sakai, and F. Kojima, "Determination of exploration target based on belief measure in multi-robot exploration," in *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694)*, vol. 3. IEEE, 2003, pp. 1545–1550.

[111] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 4. IEEE, 2003, pp. 3232–3238.

[112] A. Solanas and M. A. Garcia, "Coordinated multi-robot exploration through unsupervised clustering of unknown space," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1. IEEE, 2004, pp. 717–721.

[113] C. Icking, T. Kamphans, R. Klein, and E. Langetepe, “Exploring simple grid polygons,” in *International Computing and Combinatorics Conference*. Springer, 2005, pp. 524–533.

[114] L. Wu, M. Á. García García, D. Puig Valls, and A. Solé Ribalta, “Voronoi-based space partitioning for coordinated multi-robot exploration,” *Journal of Physical Agents*, 2007.

[115] A. Renzaglia and A. Martinelli, “Potential field based approach for coordinate exploration with a multi-robot team,” in *2010 IEEE Safety Security and Rescue Robotics*. IEEE, 2010, pp. 1–6.

[116] J. De Hoog, S. Cameron, A. Visser *et al.*, “Autonomous multi-robot exploration in communication-limited environments,” in *Proceedings of the Conference on Towards Autonomous Robotic Systems*. Citeseer, 2010, pp. 68–75.

[117] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, “Coordination for multi-robot exploration and mapping,” in *Aaai/Iaai*, 2000, pp. 852–858.

[118] R. Zlot, A. Stentz, M. Dias, and S. Thayer, “Multi-robot exploration controlled by a market economy,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, pp. 3016–3023 vol.3.

[119] W. Sheng, Q. Yang, J. Tan, and N. Xi, “Distributed multi-robot coordination in area exploration,” *Robotics and autonomous systems*, vol. 54, no. 12, pp. 945–955, 2006.

[120] A. Marjovi, J. G. Nunes, L. Marques, and A. De Almeida, “Multi-robot exploration and fire searching,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1929–1934.

[121] P. Mukhija, R. Sawhney, and K. M. Krishna, "Multi robotic exploration with communication requirement to a fixed base station," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 2010, pp. 1515–1516.

[122] A. D. Haumann, K. D. Listmann, and V. Willert, "Discoverage: A new paradigm for multi-robot exploration," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 929–934.

[123] L. Wu, D. Puig, and M. A. Garcia, "Balanced multi-robot exploration through a global optimization strategy," *Journal of Physical Agents*, 2010.

[124] M. Keidar and G. A. Kaminka, "Robot exploration with fast frontier detection: Theory and experiments," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 113–120.

[125] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib, "Distributed value functions for multi-robot exploration," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1544–1550.

[126] A. Bautin, O. Simonin, and F. Charpillet, "Minpos: A novel frontier allocation algorithm for multi-robot exploration," in *International conference on intelligent robotics and applications*. Springer, 2012, pp. 496–508.

[127] J. Faigl and M. Kulich, "On determination of goal candidates in frontier-based multi-robot exploration," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 210–215.

[128] J. Faigl, P. Vanìk, and M. Kulich, "Self-organizing map for determination of goal candidates in mobile robot exploration." in *ESANN*, 2014.

[129] A. Nuechter *et al.*, "Multi-robot cooperation for efficient exploration," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 55, no. 3, pp. 276–286, 2014.

[130] T. Kuyucu, I. Tanev, and K. Shimohara, "Superadditive effect of multi-robot coordination in the exploration of unknown environments via stigmergy," *Neurocomputing*, vol. 148, pp. 83–90, 2015.

[131] L. Klodt and V. Willert, "Equitable workload partitioning for multi-robot exploration through pairwise optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2809–2816.

[132] R. G. Colares and L. Chaimowicz, "The next frontier: Combining information gain and distance cost for decentralized multi-robot exploration," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 268–274.

[133] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716–723, 2016.

[134] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1957–1962.

[135] G. Fang, G. Dissanayake, and H. Lau, "A behaviour-based optimisation strategy for multi-robot exploration," in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, vol. 2. IEEE, 2004, pp. 875–879.

[136] W. Sheng, Q. Yang, S. Ci, and N. Xi, "Multi-robot area exploration with limited-range communications," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 2. IEEE, 2004, pp. 1414–1419.

[137] S. Sariel and T. R. Balch, "Efficient bids on task allocation for multi-robot exploration." in *FLAIRS Conference*, 2006, pp. 116–121.

[138] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," in *Proceedings of the AAAI-05 Workshop on Integrating Planning into Scheduling*. sn, 2005, pp. 27–33.

[139] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters." in *Robotics: Science and systems*, vol. 2, 2005, pp. 65–72.

[140] X. Ma, F. Meng, Y. Li, W. Chen, and Y. Xi, "Multi-agent-based auctions for multi-robot exploration," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 2. IEEE, 2006, pp. 9262–9266.

[141] A. K. Poemomo and H. S. Ying, "New cost function for multi-robot exploration," in *2006 9th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2006, pp. 1–6.

[142] X. Ma, Q. Zhang, and Y. Li, "Genetic algorithm-based multi-robot cooperative exploration," in *2007 IEEE International Conference on Control and Automation*. IEEE, 2007, pp. 1018–1023.

[143] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "A randomized strategy for cooperative robot exploration," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 768–774.

[144] R. Vincent, D. Fox, J. Ko, K. Konolige, B. Limketkai, B. Morisset, C. Ortiz, D. Schulz, and B. Stewart, "Distributed multirobot exploration, mapping, and task allocation," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2, pp. 229–255, 2008.

[145] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2818–2823.

[146] I. Kallel, A. Chatty, and A. M. Alimi, "Self-organizing multirobot exploration through counter-ant algorithm," in *International Workshop on Self-Organizing Systems*. Springer, 2008, pp. 133–144.

[147] J. Kim, F. Zhang, and M. Egerstedt, "A provably complete exploration strategy by constructing voronoi diagrams," *Autonomous Robots*, vol. 29, no. 3, pp. 367–380, 2010.

[148] S. Frank, K. Listmann, D. Haumann, and V. Willert, "Performance analysis for multi-robot exploration strategies," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2010, pp. 399–410.

[149] J. De Hoog, S. Cameron, and A. Visser, "Selection of rendezvous points for multi- robot exploration in dynamic environments," *International Conference on Autonomous Agents and Multi-Agent Systems*, 2010.

[150] A. Bautin, O. Simonin, and F. Charpillet, "Towards a communication free coordination for multi-robot exploration," in *6th National conference on control architectures of robots*, 2011, pp. 8–p.

[151] P. Brass, F. Cabrera-Mora, A. Gasparri, and J. Xiao, "Multirobot tree and graph exploration," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 707–717, 2011.

[152] Z. Yan, N. Jouandeau, and A. A. Cherif, "Multi-robot decentralized exploration using a trade-based approach," in *8th International Conference on Informatics in Control, Automation and Robotics*, 2011.

[153] M. Meghjani and G. Dudek, "Multi-robot exploration and rendezvous on graphs," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5270–5276.

[154] J. S. Cepeda, L. Chaimowicz, R. Soto, J. L. Gordillo, E. A. Alanís-Reyes, and L. C. Carrillo-Arce, “A behavior-based strategy for single and multi-robot autonomous exploration,” *Sensors*, vol. 12, no. 9, pp. 12 772–12 797, 2012.

[155] S. Bhattacharya, N. Michael, and V. Kumar, “Distributed coverage and exploration in unknown non-convex environments,” in *Distributed autonomous robotic systems*. Springer, 2013, pp. 61–75.

[156] R. G. Colares and L. Chaimowicz, “A novel distance cost approach for multi-robot integrated exploration,” in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*. IEEE, 2015, pp. 192–197.

[157] B.-F. Florea, O. Grigore, and M. Datcu, “Pheromone averaging exploration algorithm,” in *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, 2015, pp. 617–622.

[158] M. Al Khawaldah and A. Nüchter, “Enhanced frontier-based exploration for indoor environment with multiple robots,” *Advanced Robotics*, vol. 29, no. 10, pp. 657–669, 2015.

[159] E. Kaufman, T. Lee, and Z. Ai, “Autonomous exploration by expected information gain from probabilistic occupancy grid mapping,” in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*. IEEE, 2016, pp. 246–251.

[160] S. C. Nagavarapu, L. Vachhani, and A. Sinha, “Multi-robot graph exploration and map building with collision avoidance: A decentralized approach,” *Journal of Intelligent & Robotic Systems*, vol. 83, no. 3, pp. 503–523, 2016.

[161] L. Bravo, U. Ruiz, R. Murrieta-Cid, G. Aguilar, and E. Chavez, “A distributed exploration algorithm for unknown environments with multiple obstacles by mul-

tiple robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4460–4466.

[162] A. T. Palacios, A. Sánchez L, and J. M. E. Bedolla Cordero, "The random exploration graph for optimal exploration of unknown environments," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, p. 1729881416687110, 2017.

[163] A. J. Smith and G. A. Hollinger, "Distributed inference-based multi-robot exploration," *Autonomous Robots*, vol. 42, no. 8, pp. 1651–1668, 2018.

[164] J. J. Lopez-Perez, U. H. Hernandez-Belmonte, J.-P. Ramirez-Paredes, M. A. Contreras-Cruz, and V. Ayala-Ramirez, "Distributed multirobot exploration based on scene partitioning and frontier selection," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[165] A. Hentout, A. Maoudj, N. Kaid-Youcef, D. Hebib, and B. Bouzouia, "Distributed multi-agent bidding-based approach for the collaborative mapping of unknown indoor environments by a homogeneous mobile robot team," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 84–99, 2020.

[166] A. AlDahak, L. Seneviratne, and J. Dias, "Frontier-based exploration for unknown environments using incremental triangulation," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2013, pp. 1–6.

[167] V. Spirin, J. d. Hoog, A. Visser, and S. Cameron, "Mresim, a multi-robot exploration simulator for the rescue simulation league," in *Robot Soccer World Cup*. Springer, 2014, pp. 106–117.

[168] T. Kovács, A. Pásztor, and Z. Istenes, "A multi-robot exploration algorithm based on a static bluetooth communication chain," *Robotics and Autonomous Systems*, vol. 59, no. 7-8, pp. 530–542, 2011.

[169] G. Tuna, K. Gulez, and V. C. Gungor, "The effects of exploration strategies and communication models on the performance of cooperative exploration," *Ad Hoc Networks*, vol. 11, no. 7, pp. 1931–1941, 2013.

[170] R. K. Rathnam and A. Birk, "A distributed algorithm for cooperative 3d exploration under communication constraints," *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 4, pp. 223–232, 2013.

[171] J. Banfi, A. Quattrini Li, I. Rekleitis, F. Amigoni, and N. Basilico, "Strategies for coordinated multirobot exploration with recurrent connectivity constraints," *Autonomous Robots*, vol. 42, no. 4, pp. 875–894, 2018.

[172] J. Banfi, A. Q. Li, N. Basilico, and F. Amigoni, "Communication-constrained multirobot exploration: Short taxonomy and comparative results," in *Proceedings of the IROS workshop on on-line decision-making in multi-robot coordination (DEMUR2015)*, 2015, pp. 1–8.

[173] T. Nestmeyer, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous robots*, vol. 41, no. 4, pp. 989–1011, 2017.

[174] F. Benavides, P. Monzón, C. P. C. Chanel, and E. Grampín, "Multi-robot cooperative systems for exploration: Advances in dealing with constrained communication environments," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. IEEE, 2016, pp. 181–186.

[175] E. A. Jensen, L. Lowmanstone, and M. Gini, "Communication-restricted exploration for search teams," in *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 17–30.

[176] F. Benavides, C. Ponzoni Carvalho Chanel, P. Monzón, and E. Grampín, "An auto-adaptive multi-objective strategy for multi-robot exploration of constrained-communication environments," *Applied Sciences*, vol. 9, no. 3, p. 573, 2019.

[177] D. Puig, M. A. García, and L. Wu, "A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 635–653, 2011.

[178] T. Wattanavekin, T. Ogata, T. Hara, and J. Ota, "Mobile robot exploration by using environmental boundary information," *International Scholarly Research Notices*, vol. 2013, 2013.

[179] F. F. Carvalho, R. C. Cavalcante, M. A. Vieira, L. Chaimowicz, and M. F. Campos, "A multi-robot exploration approach based on distributed graph coloring," in *2013 Latin American Robotics Symposium and Competition*. IEEE, 2013, pp. 142–147.

[180] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error," in *International joint conference on artificial intelligence*, vol. 15. LAWRENCE ERLBAUM ASSOCIATES LTD, 1997, pp. 1340–1345.

[181] M. Ahmadi and P. Stone, "A multi-robot system for continuous area sweeping tasks," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1724–1729.

[182] P. K. Penumarthi, A. Q. Li, J. Banfi, N. Basilico, F. Amigoni, J. O'Kane, I. Rekleitis, and S. Nelakuditi, "Multirobot exploration for building communication maps with prior from communication models," in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2017, pp. 90–96.

[183] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, pp. 9–9, 2008.

[184] D. F. Hougen, S. Benjaafar, J. C. Bonney, J. R. Budenske, M. Dvorak, M. Gini, H. French, D. G. Krantz, P. Y. Li, F. Malver *et al.*, "A miniature robotic system for

reconnaissance and surveillance," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 501–507.

[185] P. Senarathne and D. Wang, "Frontier based exploration with task cancellation," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*. IEEE, 2014, pp. 1–6.

[186] T. Andre and C. Bettstetter, "Collaboration in multi-robot exploration: to meet or not to meet?" *Journal of intelligent & robotic systems*, vol. 82, no. 2, pp. 325–337, 2016.

[187] E. A. Jensen and M. Gini, "Effects of communication restriction on online multi-robot exploration in bounded environments," in *Distributed Autonomous Robotic Systems: The 14th International Symposium*. Springer, 2019, pp. 469–483.

[188] S. Kavirayani, D. S. Uddandapu, A. Papasani, and T. V. Krishna, "Robot for delivery of medicines to patients using artificial intelligence in health care," in *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*. IEEE, 2020, pp. 1–4.

[189] A. Franchi, G. Oriolo, L. Freda, and M. Vendittelli, "A decentralized strategy for cooperative robot exploration," in *First International Conference on Robot Communication and Coordination (ROBOCOMM 2007)*. ICST, 2007, pp. 1–8.

[190] A. Benkrid and N. Achour, "A novel approach for coordinated multi-robot exploration," in *2017 6th International Conference on Systems and Control (ICSC)*. IEEE, 2017, pp. 509–513.

[191] C. Stachniss, Ó. Martínez Mozos, and W. Burgard, "Efficient exploration of unknown indoor environments using a team of mobile robots," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2, pp. 205–227, 2008.

[192] C. C. Ooi and C. Schindelhauer, "Minimal energy path planning for wireless robots," *Mobile Networks and Applications*, vol. 14, pp. 309–321, 2009.

[193] Y. Pei and M. W. Mutka, "Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1551–1556.

[194] H. J. Chang, C. G. Lee, Y.-H. Lu, and Y. C. Hu, "Energy-time-efficient adaptive dispatching algorithms for ant-like robot systems," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 3294–3299.

[195] 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*, pp. 31–51, 2015.

[196] M. Rudolph, S. Wilson, and M. Egerstedt, "Range limited coverage control using air-ground multi-robot teams," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3525–3530.

[197] N. Mathews, G. Valentini, A. L. Christensen, R. O'Grady, A. Brutschy, and M. Dorigo, "Spatially targeted communication in decentralized multirobot systems," *Autonomous Robots*, vol. 38, pp. 439–457, 2015.

[198] S. Mayya, D. S. D'antonio, D. Saldaña, and V. Kumar, "Resilient task allocation in heterogeneous multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1327–1334, 2021.

[199] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 28, 2013.

[200] B. Panigrahi, S. Behera, S. Shailendra, H. K. Rath, and A. Pal, “Delay-sensitive wireless relaying in multi-robot indoor networks,” in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, 2019, pp. 577–582.

[201] J. Xavier A, K. P. Nadar *et al.*, “Multirobot system to explore unknown environment with connection maintenance,” *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[202] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[203] H.-L. Choi, L. Brunet, and J. P. How, “Consensus-based decentralized auctions for robust task allocation,” *IEEE transactions on robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[204] I. Jawhar, N. Mohamed, J. Wu, and J. Al-Jaroodi, “Networking of multi-robot systems: Architectures and requirements,” *Journal of Sensor and Actuator Networks*, vol. 7, no. 4, p. 52, 2018.

[205] E. Poberezkin, H. Roozbahani, M. Alizadeh, and H. Handroos, “Development of a robust wi-fi/4g-based ros communication platform for an assembly and repair mobile robot with reliable behavior under unstable network or connection failure,” *Artificial Life and Robotics*, vol. 27, no. 4, pp. 786–795, 2022.

[206] R. Singh, R. Samkaria, A. Gehlot, and S. Choudhary, “Design and development of iot enabled multi robot system for search and rescue mission.” *Int. J. Web Appl.*, vol. 10, no. 2, pp. 51–63, 2018.

[207] J. H. Ahnn, “The robot control using the wireless communication and the serial communication,” *Master Degree, Cornell University*, 2007.

[208] K. Engin and V. Isler, "Minimizing movement to establish the connectivity of randomly deployed robots," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 28, 2018, pp. 451–458.

[209] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, 2006.

[210] Y. Pei, M. W. Mutka, and N. Xi, "Connectivity and bandwidth-aware real-time exploration in mobile robot networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 9, pp. 847–863, 2013.

[211] A. Dutta, A. Ghosh, and O. P. Kreidl, "Multi-robot informative path planning with continuous connectivity constraints," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3245–3251.

[212] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[213] A. Zhu and S. X. Yang, "A survey on intelligent interaction and cooperative control of multi-robot systems," *IEEE ICCA 2010*, pp. 1812–1817, 2010.

[214] K. Kobayashi, T. Higuchi, and S. Ueno, "Connectivity maintenance for robotic swarms by distributed role switching algorithm," *Artificial Life and Robotics*, vol. 27, no. 4, pp. 876–884, 2022.

[215] E. Dey, M. Walczak, M. S. Anwar, and N. Roy, "A reliable and low latency synchronizing middleware for co-simulation of a heterogeneous multi-robot systems," *arXiv preprint arXiv:2211.05359*, 2022.

[216] K. Khateri, M. Pourgholi, M. Montazeri, and L. Sabattini, "A comparison between decentralized local and global methods for connectivity maintenance of multi-

robot networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 633–640, 2019.

[217] ——, "Decentralized local-global connectivity maintenance for networked robotic teams," *European Journal of Control*, vol. 51, pp. 110–121, 2020.

[218] M. Boldrer, P. Bevilacqua, L. Palopoli, and D. Fontanelli, "Graph connectivity control of a mobile robot network with mixed dynamic multi-tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1934–1941, 2021.

[219] A. Gusrialdi, "Connectivity-preserving distributed algorithms for removing links in directed networks," *Network Science*, vol. 10, no. 3, pp. 215–233, 2022.

[220] T. Hiasa and T. Murayama, "Development of experimental multi-robot system for network connectivity controls," *Journal of Robotics and Mechatronics*, vol. 34, no. 5, pp. 1185–1191, 2022.

[221] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 3462–3468.

[222] A. Mannucci, L. Pallottino, and F. Pecora, "Provably safe multi-robot coordination with unreliable communication," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3232–3239, 2019.

[223] T. Kazama, K. Sugawara, and T. Watanabe, "Collecting behavior of interacting robots with virtual pheromone," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 347–356.

[224] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[225] M. Andries and F. Charpillet, "Multi-robot exploration of unknown environments with identification of exploration completion and post-exploration rendezvous using ant algorithms," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5571–5578.

[226] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Autonomous Robots*, vol. 11, pp. 319–324, 2001.

[227] I. Navarro and F. Matía, "An introduction to swarm robotics," *International Scholarly Research Notices*, vol. 2013, 2013.

[228] E. A. Jensen, "Multi-robot exploration with communication restrictions," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[229] M. Lauri, E. Heinänen, and S. Frintrop, "Multi-robot active information gathering with periodic communication," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 851–856.

[230] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo, "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 943–967, 2007.

[231] L. E. Parker, "Springer Handbook of Robotics," *Springer Handbook of Robotics*, no. June, 2008.

[232] T. Balch and R. C. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous robots*, vol. 1, pp. 27–52, 1994.

[233] K. S. Sikand, L. Zartman, S. Rabiee, and J. Biswas, "Robofleet: Open source communication and management for fleets of autonomous robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 406–412.

[234] S. H. Arul, A. S. Bedi, and D. Manocha, "Dmca: Dense multi-agent navigation using attention and communication." *CoRR*, 2022.

[235] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.

[236] J.-C. Latombe and J.-C. Latombe, "Exact cell decomposition," *Robot motion planning*, pp. 200–247, 1991.

[237] Z. Yan, L. Fabresse, J. Laval, and N. Bouraqadi, "Team size optimization for multi-robot exploration," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2014, pp. 438–449.

[238] C. Tovey and S. Koenig, "Improved analysis of greedy mapping," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 4. IEEE, 2003, pp. 3251–3257.

[239] R. Fleischer and G. Trippen, "Exploring an unknown graph efficiently," in *European Symposium on Algorithms*. Springer, 2005, pp. 11–22.

[240] M. Keidar, E. Sadeh-Or, and G. A. Kaminka, "Fast frontier detection for robot exploration," in *Advanced Agent Technology: AAMAS 2011 Workshops, AMPLE, AOSE, ARMS, DOCM 3 AS, ITMAS, Taipei, Taiwan, May 2-6, 2011. Revised Selected Papers 10*. Springer, 2012, pp. 281–294.

[241] U. Jain, R. Tiwari, and W. W. Godfrey, "Comparative study of frontier based exploration methods," in *2017 Conference on Information and Communication Technology (CICT)*. IEEE, 2017, pp. 1–5.

[242] W. Sheng, Q. Yang, S. Zhu, and Q. Wang, "Efficient map synchronization in ad hoc mobile robot networks for environment exploration," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2297–2302.

[243] E. Olcay, J. Bodeit, and B. Lohmann, "Sensor-based exploration of an unknown area with multiple mobile agents," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9621–9627, 2020.

[244] T. Patten, R. Fitch, and S. Sukkarieh, "Large-scale near-optimal decentralised information gathering with multiple mobile robots," in *Proceedings of Australasian Conference on Robotics and Automation*, no. 0.1, 2013, pp. 0–15.

[245] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2124–2130.

[246] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu, "A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints," *IEEE transactions on cybernetics*, vol. 44, no. 3, pp. 305–314, 2013.

[247] N. Sudha, S. Nandi, and K. Sridharan, "A parallel algorithm to construct voronoi diagram and its vlsi architecture," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 3. IEEE, 1999, pp. 1683–1688.

[248] K. Senthilkumar, "A study of behavior-based and role-based autonomous multi-robot exploration and coverage systems," *International Journal of Latest Research In Engineering and Computing (IJLREC)*, vol. 1, pp. 12–20, 2013.

[249] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Autonomous Robots*, vol. 44, no. 3, pp. 547–584, 2020.

[250] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation,"

in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1. IEEE, 2004, pp. 698–705.

[251] M. J. Matarić and G. S. Sukhatme, "Task-allocation and coordination of multiple robots for planetary exploration," in *Proceedings of the International Conference on Advanced Robotics*, 2001, pp. 61–70.

[252] M. B. Dias and A. Stentz, "A free market architecture for distributed control of a multirobot system," in *International Conference intelligent Autonomous Systems*. Carnegie Mellon University, 2000.

[253] D. C. Parkes and L. H. Ungar, "Iterative combinatorial auctions: Theory and practice," *Aaai/iaai*, vol. 7481, p. 53, 2000.

[254] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 1212–1217.

[255] Z. Yao, X. Ye, and X. Dai, "Bidding coordination algorithm with cfc and an emotion switch," *Robotica*, vol. 36, no. 4, pp. 607–623, 2018.

[256] S. Parti, "A new frontier based approach for multi-robot coverage in unknown environments," Ph.D. dissertation, University of Windsor (Canada), 2016.

[257] J. Hawley and Z. Butler, *Hierarchical Distributed Task Allocation for Multi-robot Exploration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 445–458.

[258] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 4373–4379.

[259] A. Soni, "Performance analysis of task allocation for mobile robot exploration under energy constraints," in *2021 13th International Conference on Information & Communication Technology and System (ICTS)*. IEEE, 2021, pp. 282–287.

[260] M. Al Khawaldah and S. Livation, "Line-of-sight exploration of unknown environment by a pair of mobile robots," in *2018 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE, 2018, pp. 1–5.

[261] F. De Rango and N. Palmieri, "A swarm-based robot team coordination protocol for mine detection and unknown space discovery," in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2012, pp. 703–708.

[262] R. Arezoumand, S. Mashohor, and M. H. Marhaban, "Finding objects with segmentation strategy based multi robot exploration in unknown environment," *Procedia-Social and Behavioral Sciences*, vol. 97, pp. 580–586, 2013.

[263] B. Yamauchi, "Decentralized coordination for multirobot exploration," *Robotics and Autonomous Systems*, vol. 29, no. 2-3, pp. 111–118, 1999.

[264] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[265] A. Datta, R. Tallamraju, and K. Karlapalem, "Multiple drones driven hexagonally partitioned area exploration: Simulation and evaluation," *arXiv preprint arXiv:1906.00401*, 2019.

[266] P. Maini and P. Sujit, "Distributed task servicing using multiple robots with human-in-the-loop under limited communication range," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 301–306.

[267] X. Chen, P. Zhang, G. Du, and F. Li, "A distributed method for dynamic multi-robot task allocation problems with critical time constraints," *Robotics and Autonomous Systems*, vol. 118, pp. 31–46, 2019.

[268] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[269] M. Hofmeister and M. Kronfeld, "Multi-robot coverage considering line-of-sight conditions," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 121–126, 2010.

[270] A. sir, "SSB," 2018. [Online]. Available: https://github.com/ankitsn609/SSB_Approach/blob/main/SSB.mp4/

[271] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 155–160.

[272] T. Andre, D. Neuhold, and C. Bettstetter, "Coordinated multi-robot exploration: Out of the box packages for ros," in *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2014, pp. 1457–1462.

[273] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

[274] D. X. A. C. Library, "GitHub," 2022. [Online]. Available: https://github.com/digidotcom/xbee_ansic_library/

[275] B. Jha, V. Turetsky, and T. Shima, "Robust path tracking by a dubins ground vehicle," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2614–2621, 2018.

[276] J. H. Strom, "Online mapping and perception algorithms for multi-robot teams operating in urban environments," MICHIGAN UNIV ANN ARBOR, Tech. Rep., 2015.

[277] D. Calisi and D. Nardi, "Performance evaluation of pure-motion tasks for mobile robots with respect to world models," *Autonomous Robots*, vol. 27, pp. 465–, 11 2009.

[278] F. Li, Y. Ding, M. Zhou, K. Hao, and L. Chen, "An affection-based dynamic leader selection model for formation control in multirobot systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1217–1228, 2017.

[279] A. Gautam, S. Arjun Ram, V. S. Shekhawat, and S. Mohan, "Balanced partitioning of workspace for efficient multi-robot coordination," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2017, pp. 104–109.

[280] A. Gautam, V. S. Shekhawat, and S. Mohan, "A graph partitioning approach for fast exploration with multi-robot coordination," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 459–465.

[281] A. Bautin, O. Simonin, and F. Charpillet, "Minpos : A novel frontier allocation algorithm for multi-robot exploration," in *Intelligent Robotics and Applications*, C.-Y. Su, S. Rakheja, and H. Liu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 496–508.

[282] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[283] J. L. Blanco, J. González-Jiménez, and J.-A. Fernández-Madrigal, "A robust, multi-hypothesis approach to matching occupancy grid maps," *Robotica*, vol. 31, 08 2013.

[284] D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, ser. AAAI'11. AAAI Press, 2011, p. 1114–1119.

[285] R. Kumar, "Mrsl jump point search planning library v1.1," 2019. [Online]. Available: https://github.com/KumarRobotics/jps3d

[286] B. Gerkey, "Player-stage: 2d multiple-robot simulator," 2014. [Online]. Available: http://playerstage.sourceforge.net/

[287] N. Robotics, "Fire bird vi robotic research platform," 2022. [Online]. Available: http://www.nex-robotics.com/products/fire-bird-vi-robots/fire-bird-vi-robotic-research-platform.html

[288] B. G. Brian P. Gerkey, Jeremy Leibs, "Urg-04lx-ug01," 2022. [Online]. Available: http://wiki.ros.org/hokuyo_node

[289] A. Tiderko, "multimaster - fkie - ros wiki," 2015. [Online]. Available: http://wiki.ros.org/multimaster_fkie

[290] J. Horner, "multirobot-map-merge - ros wiki," 2017. [Online]. Available: http://wiki.ros.org/multirobot_map_merge

[291] B. Gerkey, "gmapping - ROS Wiki," 2019. [Online]. Available: http://wiki.ros.org/gmapping

[292] C. map of Two FB-VI Robots, "MRFTE Exploration process," 2023. [Online]. Available: https://github.com/ankitsn609/MRS_with_all_approaches/blob/main/Final_indoor.mp4

[293] H. Park and S. A. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE transactions on robotics*, vol. 33, no. 3, pp. 565–582, 2017.

[294] U. Joshi, "Version vector - maintain a list of counters, one per cluster node, to detect concurrent updates," https://martinfowler.com/articles/patterns-of-distributed-systems/version-vector.html, 2021, [Accessed 12-08-2023].

LIST OF RESEARCH PUBLICATIONS

1. A. Soni, C. Dasannacharya, A. Gautam, V. S. Shekhawat and S. Mohan, ”D-MRFTE: A Decentralized Relay-Based Approach for Multi-Robot Unknown Area Exploration,” **2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)**, Koh Samui, Thailand, 2023, pp. 1-7, doi: 10.1109/ROBIO58561.2023.10354972.

   [Scopus Indexed, ERA/Core Rank = B, SJR h5-index = 9]

2. A. Soni, C. Dasannacharya, A. Gautam, V. S. Shekhawat and S. Mohan, ”Multi-Robot Unknown Area Exploration Using Frontier Trees,” ***2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)***, Kyoto, Japan, 2022, pp. 9934-9941, doi: 10.1109/IROS47612.2022.9981914.

   [Scopus Indexed, ERA/Core Rank = A, SJR h5-index = 128]

3. A. Gautam, A. Soni, V. S. Shekhawat, and S. Mohan, ”Multi-Robot Online Terrain Coverage under Communication Range Restrictions – An Empirical Study,” ***2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)***, Lyon, France, 2021, pp. 1862-1869, doi: 10.1109/CASE49439.2021.9551390.

   [Scopus Indexed, ERA/Core Rank = B, SJR h5-index = 30]

BIOGRAPHY OF THE CANDIDATE

Mr. Ankit Soni completed his undergraduate degree with honors in Electronics and Communication Engineering from Marudhar Engineering College, Bikaner, Rajasthan (2010-2014). He then pursued his Master's degree in Software Engineering from IIITT(NIT Campus), Trichy, Tamil Nadu (2015-2017) and graduated with distinction. In August 2018, he joined Birla Institute of Technology and Science (BITS) Pilani, India, as a full-time Ph.D. scholar in the Department of Computer Science and Information Systems, where he is being supervised by Dr. Avinash Gataum. His research interests include a range of topics such as Robotics, Embedded Systems, Artificial Intelligence, Software Engineering, Image Processing, and Electronics and Communication Engineering. He has worked as a teaching assistant for courses such as Computer Programming, Operating Systems, and VLSI at BITS Pilani, Pilani Campus.

BIOGRAPHY OF THE SUPERVISOR

Dr. Avinash Gautam is working as an Assistant Professor in the Department of Computer Science and Information Systems Engineering at BITS Pilani, Pilani Campus, since 2017. Dr. Gautam obtained his Ph.D. (specialization in Multi-Robot Systems) and M.E. in Software Systems degrees from BITS Pilani in 2017 and 2009, respectively. Dr. Gautam's research focuses on advanced deep learning models, specifically Transformers. He has published papers in esteemed conferences such as JJCB 2023, ICCV 2023, IROS 2022, ICFHR 2022, CASE 2021, SMC 2019, Mobiquitous 2019, among others. Additionally, his has published papers in reputable journals, including the Journal of Intelligent Robotic Systems, Intelligent Service Robotics, and Neural Computing and Applications. Furthermore, Dr. Gautam has contributed to the academic community by serving as a reviewer for notable publications, including the Journal of Intelligent and Robotic Systems, Intelligent Service Robotics, Transactions of the Canadian Society for Mechanical Engineering, IEEE Transactions on Intelligent Vehicles, and IEEE Transactions on Cybernetics.

BIOGRAPHY OF THE CO-SUPERVISOR

Dr. Virendra Singh Shekhawat is currently working as an Associate Professor in the Department of Computer Science & Information Systems at BITS Pilani, Pilani Campus, India. He also holds the position of Faculty In-charge for the Software Development and Educational Technology unit. He did his Ph.D. in Computer Science and engineering from BITS Pilani in 2014. He also holds an M.Tech. in Computer Science and Engineering and an M.Sc. in Physics from BIT Mesra, Ranchi, and MDS University Ajmer, respectively. His research interests are Software Defined Networking, Multi-robot Systems, and Distributed Systems. He has published research papers in Journals and international and national conferences of repute. He is a professional member of the IEEE and ACM.