

MOBILITY AND COMMUNICATION IN WIRELESS ROBOT AND SENSOR NETWORKS

By

Yuanteng Pei

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science

2011

ABSTRACT

MOBILITY AND COMMUNICATION IN WIRELESS ROBOT AND SENSOR NETWORKS

By

Yuanteng Pei

Mobility is a primary goal of many wireless communication systems. In recent years, mobile multi-hop wireless networks, such as mobile wireless sensor networks and wireless robot networks, have attracted increased attention and have been extensively studied. However, most current research does not consider the interdependence of communication and mobility and much assume an obstacle-free environment in their problem modeling and solving process.

In this dissertation, we discuss several research topics relevant to the above two issues of communication and mobility in wireless robot and sensor networks. First, we present multi-robot real-time exploration, which calls for the joint consideration of mobility and communication: it requires video and audio streams of a newly explored area be transmitted to the base station in a timely fashion as robots explore the area. Simulations show that our mobility model has achieved both improved communication quality and enhanced exploration efficiency.

Second, we further investigate the above problem with two critical and real-world network conditions: (1) heterogeneous transmission ranges and link capacities, and (2) the impact of interference. The conditions increase the model complexity but significantly influence the actual available bandwidth and the required node size in placement. We jointly consider the relay placement and routing with these two critical conditions.

Third, we introduce an online relay deployment paradigm to support remote sensing and control when mobile nodes migrate farther from the base station in a cost-effective system of mobile robots, static sensors and relays. A novel multi-robot real-time search method called *Static Relay aided Search* (STARS) is presented to allow robots to search in a *known* environment. Its solution is based on our near-optimal solution to a new variation of the multi-traveling salesman problem: precedence constrained two traveling salesman (PC2TSP).

Fourth, we propose a heterogeneous multi-robot exploration strategy with online relay deployment for an *unknown* environment called *Bandwidth aware Exploration with a Steiner Traveler* (BEST). In BEST, a relay-deployment node (RDN) tracks the FNs movement and places relays when necessary to support the video/audio streams aggregation to the base station. This problem inherits characteristics of both the Steiner minimum tree and traveling salesman problems. Extensive simulations show that BEST further enhances the exploration efficiency.

While the first four topics deal with communication and mobility issues in powerful but expensive robotic systems, the fifth topic focuses on a special type of low cost, limited capability mobile sensors called hopping sensors, whose unique method of movement makes them suitable for rugged terrains. We present (1) a distributed message forwarding model called Binary Splitting Message Forwarding (BSMF) and (2) a grid based movement model unique to these hopping sensors. Simulation shows that our scheme significantly reduces the communication overhead and achieves relatively constant total energy consumption with varying amount of obstructions.

Finally, we discuss the future work directions of this research work. We believe that a heterogeneous mobile platform to support real-time stream transmission by mobile robots, static sensor and communication devices, have great potential in various civilian and military applications, where the communication quality of service is critically important, as well as the mobility efficiency.

ACKNOWLEDGMENTS

I would like to express my thanks and deep appreciation to my advisor Professor Dr. Matt W. Mutka. This dissertation cannot be completed without his years of inspiring guidance and encouragement, his major impact on rigorous research attitude, his valuable insights in many discussions, his generous financial support, and his efforts in careful review of paper manuscripts. The Ph.D. study under Dr. Mutka was part of the most creative and treasured time in my life.

I am also thankful to Dr. Abdol-Hossein Esfahanian, Dr. Ning Xi and Dr. Li Xiao, for their insightful comments on this dissertation and efforts in serving in my Ph.D. Guidance Committee. Besides, I am grateful to the computer science and engineering department, the graduate school of Michigan State University for providing the teaching assistantship and fellowship to support the research and study.

At a personal level, I am indebted to my parents Xiaoge Pei and Weiwei Ma, for their continuous encouragement to pursue graduate study. Their attitude towards integrity, science, knowledge, and education significantly impact my life. Special thanks go to my wife Wenting Zeng, for all of her continuous support and encouragement. It was my fortune to meet her here, at the Michigan State University.

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	x
1 Introduction	1
1.1 Background	2
1.1.1 Wireless Mobile Networks	2
1.1.2 Network Quality of Service	4
1.2 Challenges and Motivations	5
1.2.1 Network QoS Provision in Mobile Networks	5
1.2.2 Relay Deployment to Maintain Connectivity	7
1.2.3 Sensor Relocation in Rugged Terrains	8
1.3 Structure of the Content	8
2 Related Work	10
2.1 Mobility Models in Wireless Mobile Networks	10
2.2 Multi-Robot Coverage and Exploration	11
2.2.1 Multi-Robot Coordination	12
2.2.2 Communication in Multi-Robot Exploration	13
2.3 Teleoperation and Remote Sensing in Robotics	15
2.4 Relay Placement in Wireless Networks	17
2.5 Static Relay Deployment for Multi-Robot Search and Exploration	18
2.6 Relocations in Wireless Mobile Networks	20
3 Connectivity and Bandwidth Aware Real-Time Exploration in Mobile Robot Networks	22
3.1 Preliminaries and Motivations	22
3.2 System Model and Problem Formulation	25
3.2.1 System Model	26
3.2.2 Problem Formulation	29
3.3 Connectivity and Bandwidth Aware eXploration (CBAX)	30
3.3.1 CBAX Overview	30
3.3.2 Frontier Robot Placement	34
3.3.3 Relay Robot Placement with Routing Path Selection	36
3.3.4 Position Assignment And Path Generation	49
3.3.5 CBAX Enhancement	52
3.4 Performance Evaluation	55
3.5 Summary	59

4	Stream Aggregation in Heterogeneous Range and Rate Mobile Robot Networks . . .	62
4.1	Preliminaries and Motivations	62
4.1.1	Key Contributions	64
4.2	System Model	65
4.2.1	Environment Model	65
4.2.2	Network Model	66
4.2.3	Mapping of Range and Rate	67
4.2.4	Discussions	69
4.3	Problem formulation	70
4.3.1	Challenges	71
4.4	Relay Placement and Routing	72
4.4.1	General Procedure	72
4.4.2	Evaluating Flow Aggregation's Impact on Relay Number	73
4.4.3	Flow Aggregation in Sending Terminals \mathcal{SN}	79
4.4.4	New Layer Relay Placement with Flow Aggregation	80
4.4.5	Upper Bound of Approximation Ratio	85
4.4.6	Extension	87
4.5	Channel Assignment	88
4.6	Performance Evaluation	89
4.6.1	Comparing to Minimum Spanning Tree based Relay Placement	89
4.6.2	Comparing to Uniform Range and Rate Relay Placement	94
4.6.3	Evaluating HBSR's Impact on Mobility Efficiency	95
4.7	Summary	99
5	STARS: Static Relays for Multi-Robot Real-time Search and Monitoring	100
5.1	Preliminaries and Motivations	100
5.1.1	Main Contribution	104
5.2	System Model	105
5.3	Problem formulation	107
5.3.1	Challenges	109
5.3.2	STARS: STAtic Relay aided Search	110
5.4	Preprocessing	113
5.4.1	Identify Sensing Positions	113
5.4.2	Identify Relay Positions	114
5.4.3	Assign the Precedence Constraint	116
5.5	Tour Generation by PC2TSP	117
5.5.1	Cluster Visiting Positions	118
5.5.2	Obtain Optimal Single Robot Tour	118
5.5.3	Prune and Balance Tours	120
5.6	Extensions	126
5.6.1	Search with a single or more than two robots	127

5.6.2	Enable constant monitoring with static sensor deployment	128
5.6.3	Consider a non-uniform search and transmit interval and relay deploying time	128
5.6.4	Leverage pre-existing relays	129
5.6.5	Leverage extra carried relays to reduce dependency	129
5.7	Performance Evaluation	130
5.8	Summary	138
6	Steiner Traveler: Relay Deployment for Remote Sensing in Heterogeneous Multi- Robot Exploration	139
6.1	Preliminaries and Motivations	139
6.1.1	Key Contributions	143
6.2	System Model	144
6.3	Problem formulation	146
6.3.1	Challenges	149
6.4	BEST: Bandwidth-aware Exploration with a Steiner Traveler	151
6.4.1	BEST with Sufficient Bandwidth	151
6.4.2	BEST with Constrained Bandwidth	153
6.5	Performance Evaluation	154
6.5.1	Simulation Setup and Parameters	155
6.5.2	Exploration Efficiency	155
6.5.3	Comparing to a Solution with Known Environments	160
6.6	Future Work	161
6.6.1	Extension with Multiple Relay-deploying Robots	161
6.6.2	Towards Traveling Path Aware Relay Deployment	161
6.7	Summary	163
7	Hopping Sensor Relocation in Rugged Terrains	164
7.1	Preliminaries	164
7.2	Motivation	166
7.3	System Model	171
7.4	Matching Process	173
7.4.1	Message Forwarding Process	173
7.4.2	Path Optimization	178
7.5	Sensor Migration	178
7.5.1	Grid Based Movement Model	178
7.5.2	Energy Computation	179
7.6	Performance Evaluation	183
7.7	Summary	184

8	Summary and Proposed Future Research Work	185
8.1	Summary	185
8.2	Proposed Future Research Work	188
	BIBLIOGRAPHY	191

LIST OF TABLES

3.1	Notations in CBAX.	28
3.2	Notations in Algorithm 1-7 in CBAX.	35
4.1	Modulation rate vs. link bandwidth (capacity) vs. transmission range in 802.11a [1].	68
4.2	Notations in HBSR.	78
5.1	Notations in STARS.	108
6.1	Notations in BEST.	150

LIST OF FIGURES

1.1	Static and mobile sensors. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.	3
1.2	Pioneer 3 mobile robot [2].	3
2.1	Data flow in a usage scenario of robotic teleoperation [3].	16
3.1	Exploration snapshot of the <i>Garden</i> environment given in Section 3.4. (Nodes are scaled larger for illustration purpose.)	24
3.2	Robot status in one iteration.	30
3.3	Example of bandwidth-constrained relay node placement and routing (Step 1 and 2 out of 4 total steps). ($\gamma=3$) Edge weight denotes $ \mathbb{FS}_{\mathcal{E}} $	42
3.4	Example of bandwidth-constrained relay node placement and routing (Step 3 and 4 out of 4 total steps). ($\gamma=3$) Edge weight denotes $ \mathbb{FS}_{\mathcal{E}} $	43
3.5	Aggregate flows to nodes with maximal number of fully covered neighbors first to conserve RNs.	44
3.6	Simulation results on exploration time.	57
3.7	Environments and exploration time.	58
3.8	Simulation results on communication quality.	60
4.1	Star and the reduction index illustration. In $\delta(S)$, the underlined “-1” will be applied when the center is an extra relay we need to place: in placing the new layer node. It will not be applied when the center is an existing node: in aggregation in \mathbb{SN}	75
4.2	Illustration of relay node placement. (Uniform sending rate $r_s=3\text{Mbps}$)	76
4.3	Illustration of relay node placement (Continued). (Uniform sending rate $r_s=3\text{Mbps}$)	77
4.4	(a)-(b): Compare relay number to bandwidth aware S-MST: Varied flow sending rate r_s of 2-3Mbps.	90
4.5	(a): Compare relay number to bandwidth aware S-MST: Varied flow sending rate r_s of 4Mbps. (b): Obstructive environments.	91
4.6	(a)-(b): Compare relay number to uniform range and rate relay placement (UR^2): varied $ \mathbb{SN} $ and distance to BS.	92

4.7	(a)-(b): Compare exploration efficiency to UR ² based exploration by total number of iterations.	93
4.8	Stream aggregation for area exploration.	96
4.9	Channel sizes in exploration; $ \mathbb{FN} $ as exploration proceeds.	98
5.1	Two robots search the environment and deploy static relays as they travel along the paths. The left robot transmits streams to the base station via the center relay deployed by the right robot.	102
5.2	Multiple choices of precedence constraints.	123
5.3	Shared relay pruning and vertex transfer.	123
5.4	Invalid cases.	124
5.5	Compare the search time with optimal under varying width regions (widths: 25-35m).	131
5.6	(a) Compare the search time with optimal under varying width regions (widths: 45m). (b) Varied obstacle density from 0% to 25% for 35*50 case.	132
5.7	(a) Search time in larger regions. (b) Number of visiting positions and relays.	133
5.8	Compare the search time with varying communication ranges and sensing ranges (50m*75m region).	134
5.9	Compare the search time with optimal under varying STI and relay deploying time.	135
6.1	A relay deploying robot supports the stream aggregation for remote control of frontier nodes in exploration.	141
6.2	Merge border relay's pathToBS to new relays.	154
6.3	Exploration time compared to CBAX with varying (a) region sizes (b) number of robots.	156
6.4	(a) Exploration time compared to CBAX with varying obstacle ratios. (b) Impact of varying bandwidth constraints on the number of deployed relays.	157
6.5	(a)-(b): Number of deployed relays and traveled path length by RDN in BEST, compared to those results computed with aid of a known environment.	158
7.1	Hopping sensors we designed deployed near a construction area [4].	166
7.2	Coordinate frame for slope plane.	168
7.3	Energy ratio for $0^\circ \leq \beta \leq 75^\circ$	170

7.4	Binary splitting message forwarding.	175
7.5	Movement path optimization.	177
7.6	PMM vs. GBMM.	180
7.7	Simulation results: broadcast vs. binary splitting matching.	181
7.8	Simulation results: number of cells in matching and number of hops in movement.	182

CHAPTER 1

Introduction

Mobility is a primary goal of many wireless communication systems. In the last decade, mobile multi-hop wireless networks, such as mobile wireless sensor networks and wireless robot networks, have drawn increased attention and have been extensively studied. However, little current research has investigated in-depth the influence of a specific controllable mobility model on the network quality of service (QoS), especially when all nodes are highly mobile. In addition, mobility and communication models in much current research work often assume an obstacle-free environment where the obstacles can have a significant impact on mobility.

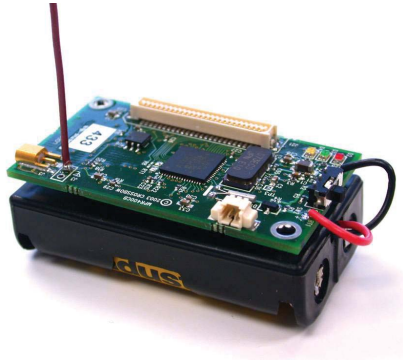
This dissertation aims to jointly consider mobility and communication in wireless mobile networks to improve the network quality of service as well as to enhance the mobility effectiveness and efficiency in an obstructive environment. In this chapter, we first introduce the background of wireless mobile networks. Then we present the challenges and motivations for quality communication and effective and efficient mobility in these networks.

1.1 Background

1.1.1 Wireless Mobile Networks

Recent decades have witnessed the widespread use of mobile and portable devices, which are likely to popularize wireless mobile networks. Such networks do not require any wired infrastructure and its nodes intercommunicate through single hop and multi-hop paths [5]. A mobile node can be a sensor, a robot or any other mobile device. A sensor is small and relatively inexpensive. It has limited processing and computing resources. Sensor nodes can sense, measure, and gather information from the environment. With some local decision process, they are able to transmit the sensor data to the users [6]. Fig.1.1(a) shows a typical sensor node: a Mica2 mote. It cannot move by itself, but will have mobility when equipped with additional wheels. Another type of mobile sensors is called hopping sensors, whose mobility design is more adaptable for rugged terrains or difficult areas where wheeled mobility cannot perform well. Fig. 1.1(b) shows hopping sensors in jumping actions. Compared to these limited capability sensor nodes, a robot is usually more powerful but has higher cost. Equipped with accessories such as robot arms, laptop computers, laser range finders, sonars, and cameras, they are able to be teleoperated and perform a range of complicated tasks such as remote monitoring, reconnaissance, search and rescue in dangerous terrains. Fig. 1.2 shows a wheel drive Pioneer 3 robot equipped with (i) a robot arm and (ii) a camera and a laser range finder.

We categorize wireless mobile networks into two categories: (i) Mobile Ad Hoc Networks (MANETs) and (ii) Mobile Coordinated Networks (MCNs). A MANET is a collection of mobile nodes forming a temporary network on the fly, without relying on any fixed infrastructure such as a base station [8]. MANETs are mostly featured by lack of central coordination, unpredictable and



(a) A Mica2 sensor mote [7].



(b) Hopping sensors in jumping action [4].

Figure 1.1. Static and mobile sensors. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.



(a) Pioneer 3 mobile robot with a camera and a laser range finder.



(b) Pioneer 3 mobile robot with a robot arm.

Figure 1.2. Pioneer 3 mobile robot [2].

varying network topology, and limited available resources. Because it is difficult to characterize mobility pattern in a specific way, only general mobility models such as random waypoint have been proposed.

Mobile Coordinated Networks (MCNs), in contrast to MANETs, are the multi-hop mobile networks that solely rely on the wireless communication but can be coordinated and controlled by a fixed base station. While MANETs have been extensively studied, mobile coordinated networks are relatively new in the network research community. However, MCNs has been the underlying network model for a number of multi-robot systems in the robotics research community. These multi-robot systems have extensive applications in surveillance, reconnaissance, search and rescue missions. As human operators at the base station often need to monitor the powerful and expensive robot team's behavior in real-time, and to coordinate the robot team's actions, MCNs emerges to be a new paradigm with wide applications in wireless mobile networks.

A crucial advantage of MCNs over MANETs is that the coordination and topology control in MCNs facilitates QoS provision, which will be discussed in the Section 6.3.1.

1.1.2 Network Quality of Service

Network quality of service (QoS) is a term that specifies the performance level of a service offered by the network to the user. The aim of QoS provision is to obtain a more deterministic network behavior, for better delivering the information carried by the network and better utilizing the network resources [8]. Most multimedia and other time or error sensitive applications have QoS requirements. The parameters to characterize the quality of service includes: (i) bandwidth: the rate at which an application's traffic must be carried by the network, (ii) latency: the delay that an appli-

cation can tolerate in delivering a data packet, (iii) jitter: the variation in latency, and (iv) loss: the percentage of data loss [9].

1.2 Challenges and Motivations

1.2.1 Network QoS Provision in Mobile Networks

In the recent years, video and audio content has been increasingly popular in today's digital networks. With proliferation of portable mobile devices such as inexpensive HD camcorders, iPhone like WiFi-enabled video-chatting phones, and iPad like tablets, efficient and reliable communication for bandwidth consuming and real-time multimedia content in multi-hop wireless network becomes increasingly interesting, especially when most nodes are highly mobile.

Most current research on QoS provision for mobile networks focuses on the MANETs. Several QoS aware routing protocols have been proposed to provide assured throughput. However, due to MANETs's dynamic and unpredictable nature, it is usually a challenging task. QoS and best effort routing can only be successfully achieved if the network is "combinatorially stable": Nodes cannot move faster than routing updates propagate [10].

There is a more fundamental issue that challenges the QoS provision in mobile network. The issue has not been addressed by most current work: the mobility pattern and its resulting topology are usually unpredictable and uncontrollable, even though they have a profound impact on the QoS. They are often the essential causes for failures to meet QoS requirements and only applying certain networking measures cannot solve the problem. For example, movement can easily result in a topology having all broken links to a destination or an over-saturated shared link as a bottleneck.

No matter what TDMA allocation method or routing metric a protocol uses, there is little chance to solve it: the problem is caused by the mobility. However, current QoS provision schemes in MANETs have no control on the mobility and topologies, which are critical to communication quality. These schemes cannot make the mobility patterns to have QoS awareness.

Given the significance of mobility control over QoS provision, our research improves QoS by designing QoS-aware mobility patterns in mobile coordinated networks. *Multi-robot real-time exploration* presents a complex and coordinated mobility model on top of MCNs, where the base station can control and coordinate a team of mobile robots in a multi-hop fashion. It is complex in that it has strong requirements for both mobility efficiency and communication quality: it requires bandwidth consuming real-time video and audio streams of a newly explored area be transmitted to the base station in a timely fashion as robots explore the area. With such communication constraints, the exploration should be completed in minimum time. These requirements calls for joint consideration of mobility and communication: we need to provide a QoS aware mobility model that is also efficient in mobility efficiency.

Furthermore, multi-hop wireless communication always faces the challenges in dealing with (1) heterogeneous transmission ranges and link capacities; ii) the impact of interference. These conditions increase the model complexity but significantly influence the actual available bandwidth and mobility patterns. Successfully tackling these conditions is critical in obtaining QoS provision in a coordinated mobile situation.

1.2.2 Relay Deployment to Maintain Connectivity

Given the nature of the applications such as reconnaissance, search and rescue missions in earthquake, radioactive, and other dangerous or hostile regions, prior relay deployment or infrastructure networks are often infeasible when the environment is unknown and relays have to be placed at the same time as the exploration. However, as robots migrate farther from the base station, smooth communication to the base station remains a key requirement. Solely relying on mobile robots to serve as relays may not be a cost-effective approach. Online relay deployment by the mobile robots (at the same time as robots migrate), is an effective alternative to support the remote sensing and control.

All robots or a partial of the robot team may have the ability to deploy relays. When all robots move to search with ability to deploy the robots, the relay deployment introduces the precedence requirement: the needed relays for stream transmission must be deployed prior to the stream transmission. Therefore, it is possible that one robot's video transmission will rely on the relay deployed by another robot, introducing the dependency between the robots. An efficient motion planning for multiple robots satisfying the precedence and dependency constraint is then challenging.

It is also possible that only one or a partial team of robots has the ability to deploy relays. The relay deploying robots must track the movement of other frontier robots and effectively deploy new relays considering the previously deployed relays and the possible bandwidth constraints on transmitting via both the new relays and/or the previous relays to the base station. An efficient motion planning for the relay deploying robots is then interesting and challenging since we need to first define the visiting positions constrained by the communication requirements.

1.2.3 Sensor Relocation in Rugged Terrains

Besides QoS provision and motion planning, coping with obstructive environment in mobile networks is another important topic in this dissertation. In a vast array of military and civilian applications, most environments are not obstacle free. Regarding the network composed of mobile robots, we have focused on the exploration problems to search in an unknown obstructive environment. Other than the powerful but expensive mobile robots, the other most common devices in wireless mobile networks are the sensors. In fact, mobile sensor networks have increased in interest to serve a broad range of applications with their ability to monitor large scale real-world phenomena. In this dissertation, we focus on hopping sensors, a promising type of sensors whose unique method of movement is designed to fit the obstructive environment, or rugged terrains.

Hopping sensors is more adaptable for rugged terrains or difficult areas that are challenging for wheeled mobility. (Figure 7.1). In addition, their diminishing cost enables these inexpensive sensors to be deployed in large numbers. The applications of hopping sensor networks working in such areas include weather sensing, environment monitoring, disaster management, and battlefield surveillance [11]. Along with the benefits of hopping sensor networks there are several challenges such as imprecise movement, limited power and occasional failure owing to problems regarding the node, link and global maintenance and communication [12].

1.3 Structure of the Content

The remainder of this dissertation is organized as follows: In Chapter 2 we review the related work. Chapter 3 introduces the connectivity and bandwidth aware real-time exploration in mobile robot networks. We discuss the stream aggregation in heterogeneous range and rate mobile robot net-

works in Chapter 4. In addition, Chapter 5 presents the STARS framework, or STAtic Relay aided Search, to enable online relay deployment for remote sensing in multi-robot real-time search and monitoring in a known environment. Chapter 6 presents the BEST framework, or the Bandwidth-aware Exploration with a Steiner Traveler, to employ a relay deployment node (RDN) to “chases” a fixed number of frontier robots movement and places relays when necessary to support stream aggregation in a heterogeneous robot team. Last, in Chapter 7, we propose a hopping sensor relocation scheme in rugged terrains. Finally, a summary and proposed future research work are given in Chapter 8.

CHAPTER 2

Related Work

In this chapter, we first survey some general mobility models in MANETs. In the mobile coordinated networks, we then present some existing work related to the controllable mobility model of multi-robot coverage and exploration. As the mobility model for exploration is pertinent to teleoperation and remote sensing in robotics, the related work in these topics is also discussed. In addition, we also reviewed the existing work on relay placement in wireless networks. Besides, the relevant work of multi-robot search with online relay and sensor deployment to support remote sensing is presented. Last, mobility issues for relocation purpose in wireless sensor networks are briefly reviewed.

2.1 Mobility Models in Wireless Mobile Networks

Numerous general mobility models have been proposed for performance evaluation in the dynamic and uncoordinated MANETs. Three common ones are (i) the random walk mobility model based on random directions and speeds, (ii) random waypoint mobility mode with pause times between

changes in destination and speed, (iii) random direction mobility model that makes mobile nodes to move to edge of the area before changing direction and speed [13]. Such models have two major drawbacks. First, the randomness in these models leads to significant topology changes as the nodes migrate, which poses a severe threat in network QoS control. Second, such mobility models do not fit all mobile applications. They are primarily designed for ad hoc networks where mobility is unpredictable.

2.2 Multi-Robot Coverage and Exploration

Area coverage is a fundamental topic for mobile robots with extensive real-world applications such as floor cleaning, tracking hostile targets, monitoring, and search and rescue missions [14]. In these problems, a robot or multiple robots are given a bounded area that may have obstacles. Each robot is equipped with a tool of a shape corresponding to the robot's sensor or actuator range. The range should cover all the area. The area coverage has been categorized as a static or a dynamic problem in [15]. The static version is to converge to a static configuration so that every point in the area is under the robots "sensing shadow" with minimum number of robots. By contrast, the dynamic coverage is solved by exploring and covering the environment with motion over time. It requires a coverage path planning algorithm to have the robots' footprint (detector range) along the coverage path yields an area same to the target region in minimum amount of time [16]. By the definition of robotic exploration given in [17, 18], the problem of exploration is the same as dynamic area coverage: the problem of mobile robots to sweep or cover the working area with their sensing area in minimum amount of time. In this dissertation, we primarily focus on the robotic exploration, or the dynamic coverage problem.

2.2.1 Multi-Robot Coordination

Using multiple robots for exploration significantly improves efficiency but coordination becomes essential to avoid visiting duplicate regions. Yamauchi [19] presented a distributed exploration algorithm and introduces the concept of frontier cells, which are the ones on the border between the known and unknown regions. In this method, robots independently move to the closest cells, rather than in a coordinated fashion. Burgard et al. show a frontier cell based coordinated exploration. The scheme trades off the utilities of unexplored areas and the cost of relocating to the areas. The coordination is achieved by reducing the utilities based on the number of robots that already headed towards that area [18]. Although the protocol considers limited communication range, it has no guarantee on connectivity. Among the above schemes, none of them consider coordination with QoS awareness.

Another important category of coordination is based on bidding or auction. Simmons et al. propose a scheme where a central agent is to evaluate the bidding from other robots for the highest gain. Besides, a market economy based exploration strategy is given in [20] with similar bidding mechanism. Nonetheless, their negotiation procedure is complex. The goal generation is as well complicated compared with the frontier based one. A spanning tree based dynamic coverage algorithm is presented in [21] and [14]. A survey of market and bidding based coordination is presented in [22]. According to the survey, this type approach faces the challenges of effectively sharing information among all team members and responding quickly to dynamic conditions of the network.

Coordination for area coverage and exploration is also studied from the control perspective [23–27]. However, none of these schemes has bandwidth awareness or the specific goal of

area exploration. Novel research for coordinating robots and maintaining connectivity of mobile networks from a distributed control viewpoint is presented in [23,24]. The authors apply algebraic graph theory to verify link deletions regarding to connectivity. Tan et al. present a distributed model for cooperative multi-robot system [25]. With Voronoi diagram and Delaunay triangulations, an autonomous deployment is achieved based on control law to maximize the area coverage. Additionally, Ogren et al. show a distributed and stable control method to perform a gradient climbing task for a mobile robot network.

2.2.2 Communication in Multi-Robot Exploration

Communication in multi-robot systems has drawn increased attention [3, 23, 24]. In [28], four types of routing protocols are compared in an experiment of one mobile robot with several stationary relay nodes. In addition, Dantu et al. [29] incorporated mobility metrics such as directional and location cue to the Optimized Link State Routing (OLSR) protocol to provide more stable routes. However, they do not have bandwidth constraints and they do not have control on the mobility. Furthermore, a network architecture is developed for large mobile robotic environment [30]. However, it is an overlay network based on existing organizational wired network and Internet with additional mobile robot nodes.

While the problem of multi-robot exploration has been widely studied in [31–36] and various exploration strategies are compared in [37], little prior work considers the bandwidth constraints and only some schemes assume limited communication range and attempt to maintain connectivity. Distributed local random graph, segmentation and hill-climbing based methods in [31–33] explore the area efficiently but provide only intermittent connectivity or assume robots are always in range.

To maintain connectivity, “nearest measure” is applied in [34] to let robots *tend* to stay close to each other but it cannot be guaranteed. A concept of “comfort zone” is proposed in [35, 38, 39], where each node monitors the distance to its neighbor and keeps it in a safe range. When the distance is beyond a safe threshold, the node will change its behavior into tracking the neighbor to retain connectivity. Such schemes lack systematical analysis of how to share relay robots for placing more frontier ones to enhance efficiency. Also, with communication latency and non-uniform velocities of robots, the “detect and chase” method may lead to a sequence of trace movement and the resulting topologies need to be proved deadlock-free and invariably-connected.

In addition, an iterative motion planning method in [40] maintains connectivity by rejecting trajectories that break tree edges in the distributedly-computed minimum spanning tree. However, its exploration model is different from ours without a base station and may not need to consider the bandwidth constraint. Besides, a centralized scheme in [36] guarantees connectivity by solely selecting fully-connected topologies with detecting and recovering from deadlock. Nevertheless, it only allows nodes to move to their immediate four-neighbor positions in each step and does not attempt to optimally place relay nodes to improve efficiency. In [41], a simple and practical method on maintaining connectivity is presented and demonstrated in a complex real environment. However, it does not aim to maintain the connectivity to a particular base station while expending the team, nor does it try to minimize the exploration efficiency.

The problem of relay node placement studied in [42, 43] is also pertinent to maintaining connectivity. Authors in [42] employ bipartite-graph to cover maximum number of mobile nodes using fixed number of relay nodes in one tier. In [43], the relay sensor placement problem is modeled as a variant of Steiner tree problem and improved approximation ratio algorithms are provided for this NP-hard problem without bandwidth consideration. A measurement study in [44] shows a

testbed of wireless camera networks. Nevertheless, nodes are static without optimized placement to reduce relay nodes.

We are not aware of previous work on using node placement in a controlled mobility model to ensure bandwidth adequacy and connectivity while considering realistic conditions such as interference and heterogeneous communication range and rate.

2.3 Teleoperation and Remote Sensing in Robotics

The idea of remotely controlling mobile robots is not new. Cen et al. [45] define the information in a real-time teleoperation system as “Supermedia”, which includes video, audio, haptic, temperature, control commands and other media. Supermedia is distinct from the traditional multimedia in that certain types of new media are notably sensitive to network delay. The same authors provide a framework for QoS management in teleoperating robotics over overlay networks. Multipath routing is employed to reduce the transmission latency. However, it is based on wired network (Internet) for a single robot [46] rather than a multi-hop wireless network.

Additionally, a networking framework for teleoperation in safety, security and rescue robotics in unstructured environments is given in [3]. Fig. 2.1 shows the data flow are transmitted among robots, operators, and the control center. The flow has a wide range of contents, priorities, and bandwidth requirement. The authors argue that teleoperation of mobile robots requires wireless communication with complex transmission of heterogeneous data. In addition, three levels of teleoperation are defined: (i) motion level of direct mapping between operator inputs (e.g. joystick) and robot’s motions, (ii) behavior level: command for short-time functions, (iii) mission level: high-level goals for a long time autonomy.

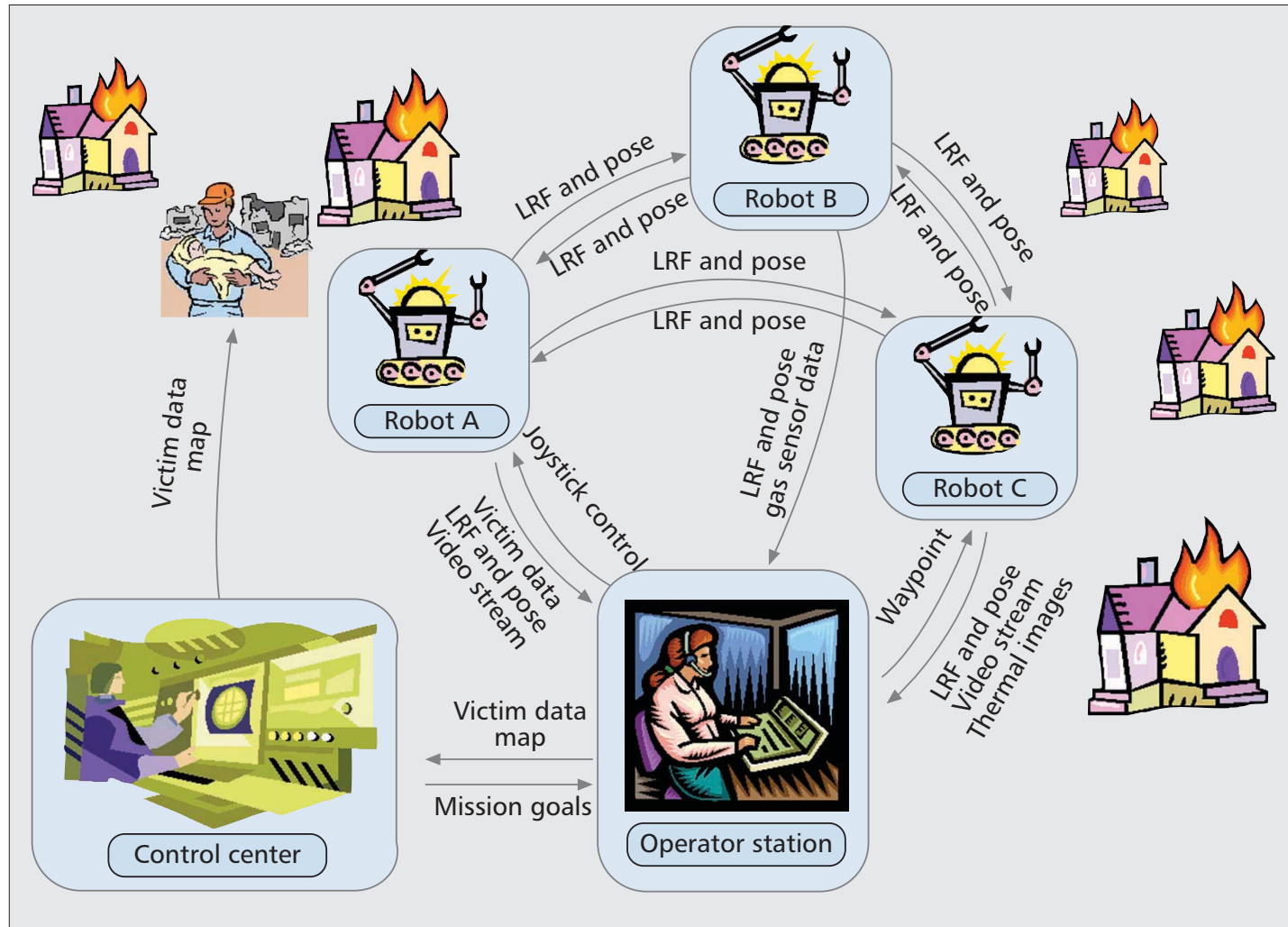


Figure 2.1. Data flow in a usage scenario of robotic teleoperation [3].

2.4 Relay Placement in Wireless Networks

Relay placement has been extensively studied [43, 47–53]. However, there is little previous work on bandwidth-aware relay placement with the heterogeneous range and rate.

Misra et al. [47] presented a rigorous relay placement framework and gave lower bounds where relay candidate positions are constrained and sensor/relay range may differ. However, their work's aim is to achieve connectivity and survivability. Authors use existing Steiner tree approximation algorithm based on minimum spanning tree (MST) as input without a heterogeneous bandwidth constraint for each link based on distance. Besides, a bandwidth aware cooperative relay placement is given in [53]. However, it does not aim to minimize the number of relays, nor considers more than two hop relaying or the heterogeneous range and rate.

Wang et al. [49] presented an interesting traffic aware relay placement scheme. However, their work has a different goal of reducing energy consumption rather than satisfying bandwidth sufficiency; the problem is modeled by Euclidean Steiner Minimum Tree problem to reduce the sum of edge length rather than the number of relays. Similar to [49], a minimum cost pipe network is computed in [54] to connect many sources to a sink. Their goal is to minimize the total pipe length cost rather than the number of relays. Besides, there is no capacity limit for each link.

Yang et al. [50] presented a relay placement to improve the network capacity in WiMax networks with heterogeneous ranges. However, the scheme considers only a single hop of relays between the base station and subscriber stations, which is different from our problem that may need multi-hop relays. Besides, relays are used differently for cooperative communication, without considering the heterogeneous bandwidth constraint based on ranges for simultaneous stream aggregation. In [51], a relay placement algorithm was presented to achieve fault tolerance with

more than one vertex-disjoint path with heterogeneous range. Similar to other work, bandwidth is not taken into account.

In [52, 55], a bandwidth aware relay placement is modeled as the Steiner tree problem with γ -inflow constraint to support multi-robot exploration. However, it assumes uniform communication range and rate. In [56], a joint single base station placement and routing approximation solution is proposed. However, it has a different goal and does not consider the relay placement problem.

Real-time video and voice stream aggregation has been investigated in multi-hop wireless sensor and camera networks [44, 57] to collect rich information about the environments. These systems have extensive applications in coal mine monitoring [58], search and rescue missions in dangerous areas, and security surveillance [52]. In [57], real-time voice streams are sent from 20 static sensor nodes to a sink in a wireless sensor network. In [44], real-time video streams are sent from 15 static camera nodes to a sink in a wireless mesh network. However, these systems do not particularly tackle the relay placement problem.

2.5 Static Relay Deployment for Multi-Robot Search and Exploration

A survey on multi-traveling salesman (mTSP) is given in [59] to describe various problem formulations, solution procedures, and real-life applications of mTSP, such as school bus routing, interview scheduling and mission planning. However, mTSP is to minimize the sum of the traveler tours rather than the maximum of them. Also, mTSP does not consider the precedence or the dependency between travelers. A comparative analysis of several linear programming formu-

lations for asymmetric (single) traveling salesman problem is given in [60]. In [61], extended formulations for the precedence constrained (single) asymmetric traveling salesman problem are presented. In [62], a multiple depot, multiple traveling salesman problem is solved by transformation into single asymmetric TSP problem. However, this problem does not have the precedence constraint. In summary, none of the above work attempts to solve the multiple traveling salesman with the precedence constraints.

Balas et al. present a class of polynomially solvable (single) traveling salesman problem with precedence constraints [63]. However, their precedence constraint does not fit our application and is notably stricter than ours. In [63], a node i must be prior to *all* $i + k$ nodes given an initial order, whereas we only require i precedes a subset of nodes. Similarly unnecessary and stricter precedence constraint is imposed in the multiple traveling salesman problem with time windows (m-TSPTW) [64]. Besides, we do not have a predefined time window for each node.

There is little previous work on using online relay deployment to support remote sensing and surveillance with mobile robots and static sensors. Mobile robot search and exploration have been extensively studied without the consideration of online relay deployment [31–36, 52, 55, 65–69]. Besides, hybrid static sensor and mobile robot systems have been studied in [70–74]. In [73], authors deploy static mote sensors, RFID (Radio-frequency identification) tags, and mobile robots to guide fire fighters in burning buildings. However, the static sensors are deployed ahead by humans. Sensors are not used as relays for real-time operation control. In [71], a single robot is used to assist the sensor network deployment and data collection. However, sensors are not used to transmit streams from the robot.

An interesting paper by Tan et al. [75] uses both mobile and static sensors to improve the target detection accuracy. The authors develop a scheduling for movement to minimize average moving

distance. However, they have a different goal compared with our problem. The movement path in [75] is markedly simpler: each mobile sensor moves on a straight line toward a known target place in the center. One of the most relevant research work is presented in [74], where robots carry sensors as payload with online deployment. However, sensors are uniformly deployed. They serve as the navigation guides rather than transmission relays. Similar ideas of using sensors to navigate robot movement are presented in [70, 72].

2.6 Relocations in Wireless Mobile Networks

Sensor relocation problems are extensively studied in [76–81]. When a sensor node depletes its energy or fails, sensor coverage holes evolve. Areas with redundant sensors can supply sensors to be relocated to these holes to maintain coverage.

Grid-based solutions for sensor relocation have been widely proposed [77, 80–82]. Wang, et al. [80] introduced two kinds of nodes, the grid head and its members, in their model and adopted a Grid-Quorum approach. Their scheme makes use of some notations. A *quorum* is mentioned in [83] as an element in a subset of a given set. In [84], a quorum system is defined as a collection of sets such that the intersection of any two sets is always non-empty. Grid cells with redundant sensors are identified as *suppliers*. They send an *advertisement message* to form a *supply quorum*. *Consumers* are grid cells in need of sensors, which send a *request message* to form a *consumer quorum*. However, with the presence of obstructions, supply and consumer quorum are not guaranteed to intersect on the grid for the matching between the supplier and consumer cells.

Similarly, [76] deals with the matching and relocation for hopping sensors. However, the issue

that potential obstructions hamper the relocation process also remains unsolved. In [85], the problem of clustering sensors under obstructive and inhospitable conditions is addressed. However, matching and relocation are not specifically tackled. In Chapter 7, we propose a simple and distributed Grid-Quorum based solution to effectively cope with obstructive environment like rugged terrains with hopping sensors.

At the same time of our work [86] and thereafter, several related research have been proposed for relocation of hopping sensors in rugged terrains [87,88]. Kim et. al [87] argues that the shortest path based scheme suffers from repeated use of sensors in specific areas, leading to occurrence of new sensor holes. They propose a new multipath relocation scheme to evenly and fairly distributing sensor to provide the requested ones to a sensing hole consumer. In addition, the definition of regularity is introduced in [88] to depict the ratio of obstructive regions in the whole area. With regularity defined, the authors proposed a reliability based scheme to relocate sensors.

CHAPTER 3

Connectivity and Bandwidth Aware

Real-Time Exploration in Mobile Robot

Networks

3.1 Preliminaries and Motivations

Exploring an environment is one of the fundamental problems in mobile robotics. Recently, multi-robot exploration has received increased attention for its notable benefit for enhanced efficiency and robustness [32, 34]. Many application domains of multi-robot exploration, such as surveillance, reconnaissance, search and rescue missions in dangerous areas require robot's bandwidth-consuming video/audio information from newly-explored area to be reliably and quickly sent back to an operator at the base station [36].

The reasons for the above transmission requirement are threefold. First, human operators often need to monitor the robot team's action and obtain the sensed information immediately. Second,

as the current robotic sensing ability is not sufficient to accurately detect complex targets, such a process should be simultaneously augmented with human recognition [38]. In fact, the use of human operators as “perceptual sensors” to process camera video is a standard practice for both UAVs and ground robotics [89]. Third, operators may even need to teleoperate robots promptly after target discovery or under exceptional circumstances. For instance, when a robot finds a victim partially covered by earthquake debris, the operator may teleoperate the robot to move closer to the target and control the robot arm to remove the obstacle and telediagnose the victim. Thus, limited robot intelligence may need to be augmented with an operator as a necessary and effective way to monitor and control a robot team. Reliable and smooth communication is essential in such a process. Hence, such Multi-Robot *Real-Time* eXploration (MRRTX), is critical.

MRRTX can be described as follows: A team of n homogenous robots are sent from the *base station* (**BS**) to explore an unknown area. Each robot communicates with limited range by forming a multi-hop network with all robots and the BS. The question is: how to design a coordinated exploration strategy for the robot team to explore the area in a minimum amount of time under the following *two constraints*? (1) The network is always connected when data is transmitted. ii) The aggregated consumed bandwidth of data flows from the frontier nodes cannot exceed the link bandwidth (capacity) when transmitting back to the BS. In fact, many types of real-time streaming data such as video/audio have a stream floor rate [46], or a minimum rate to make the stream workable. For example, a typical DVD-quality video stream requires about 5 Mbps of throughput [90]. A link should always provide adequate bandwidth for each flow to meet the floor rate requirement. Without these two constraints, disconnected nodes can evolve and the aggregated data flows may exceed the link bandwidth, leading to considerable data loss and control disturbance.

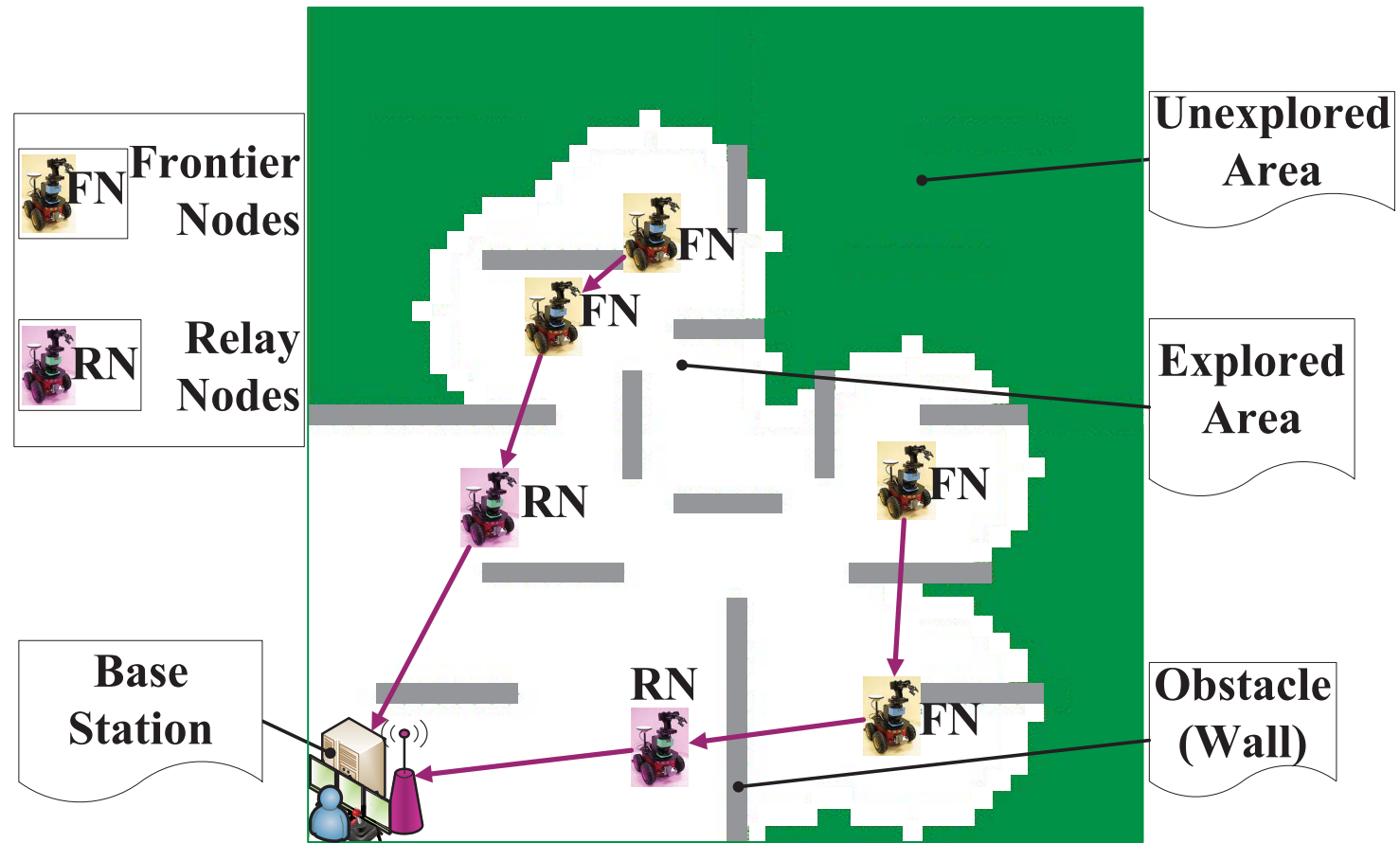


Figure 3.1. Exploration snapshot of the *Garden* environment given in Section 3.4. (Nodes are scaled larger for illustration purpose.)

To solve MRRTX, we present *Connectivity and Bandwidth Aware eXploration (CBAX)*. Differing from existing approaches, CBAX is an iterative exploration that divides the problem of the next round movement path generation and message routing into three subproblems, which are modeled and solved accordingly by variations of *algorithmic and graph-theoretic* problems, such as the set cover problem, the Steiner tree problem, and the linear bottleneck assignment problem (LBAP). When we solve the bandwidth-constrained relay node placement, we model it as a new variation, or one with the γ -inflow constraint, of the *Steiner Minimum Tree Problem with Minimum Number of Steiner Points and bounded edge length (SMT-MSP)* [43,91]. CBAX not only reduces the exploration time compared with recent work [32,36], but also guarantees the adequacy of link bandwidth and enhances the communication *quality*. In Fig. 3.1, CBAX is illustrated by an exploration snapshot obtained from our simulation program with the *Garden* environment given in Section 3.4. The figure describes that 4 frontier nodes with 2 relay nodes have explored 56.3% of the area in their fourth iteration of movement, where all robots with the BS are connected and frontier nodes send non-overflow streams to the base station by the routing paths CBAX generates.

The outline of this chapter is as follows. Related work is described in Section 2. Section 3.2 introduces the system model and the problem formation. In Section 3.3, CBAX’s general procedure is given, along with the three subproblems and optimizations on the model. Performance evaluation is in Section 3.4 while Section 3.5 summarizes the chapter.

3.2 System Model and Problem Formulation

Before presenting our solution, the system model and problem formulation are introduced as follows.

3.2.1 System Model

Robot Model: Each robot has traveling, communication, and sensing capability. It scans the environment using cameras and acoustic sensors with sensing range r_s and communicates with other nodes with communication range r_c by a 802.11 radio, where $r_c > r_s$.

Environment and Exploration Model With a Base Station: The exploration begins with the operator selecting a targeting area, all of whose borders are reachable when robots form a straight line from BS. The area is modeled by the 2D occupancy map composed of grid cells. The cell size equals that of a robot. The states of a cell are *unvisited*, *exploring*, and *explored*. An explored cell can be either *free* or *obstructed*. An obstacle does not affect communication but blocks the sensing range. An unvisited cell denotes the area that has not been reached by the sensing range of robots. Arriving at their target positions, all nodes are synchronized and start simultaneously to sense the area and transmit streams back to BS for a fixed *sensing and transmitting interval* (STI). In STI, the unvisited grid cells within the range of r_s of a node and not blocked by obstacles are changed to *exploring* status. After STI, these cells change to *explored* status since the operator has observed and detected the targets in STI. The cells that are explored but have at least an unvisited neighbor cell are called *frontier cells* and denoted as \mathbb{FC} . We mainly focus on how to efficiently explore the area and the issues after the target discovery, i.e., potential teleoperation tasks, are not discussed here.

Coordinated exploration is executed in an iterative and synchronized way controlled by the BS. At iteration t , Map^t denotes the explored map and T^t gives the *total time* elapsed. Each robot i has a position P_i^t and is dynamically classified as a *relay node* (RN) or a *frontier node* (FN). A FN explores a new area while a RN maintains connectivity for FNs with the BS. The *robot*

team size is n with n_{fn} number of FNs and n_{rn} number of RNs. We define the *information gain* of node i in iteration t , or IG_i^t , as the number of unvisited cells that can be explored with a new position P_i^t . A *position configuration* is the array of positions of the n robots. Specifically, pCG^t , the position configuration at iteration t is defined as: $pCG^t = \{P_1^t, P_2^t, \dots, P_n^t\}$. The set of all possible pCG^t is denoted as $\mathcal{P}CG^t$. Each pCG^t is associated with a *total information gain*, or tIG^t , which is the total number of non-overlapped unvisited cells that can be explored with a new position P_i^t for each node i . A *valid* position configuration is the one that satisfies the *Constraint I* defined below. Besides, \mathbf{Path}_i^t is the migrating path of node i from iteration $t-1$ to t and \mathbf{Path}^t is the set of \mathbf{Path}_i^t for each node i . Similar to [23], we define a *dynamic graph* $G(t) = \{V, E(t)\}$ where V denotes the sets of vertices indexed by the set of robots with BS and $E(t)$ is the edge set representing the time varying set of bi-directional communication links of vertices and $(i, j) \in E(t)$ iff $\text{dist}(P_i^t, P_j^t) \leq r_c$ ($\text{dist}(i, j)$ denotes the straight-line distance between i and j .)

In addition, we define a *routing configuration* by the path array from all FNs back to BS where the streams traverse. Specifically, rCG^t , the routing configuration at iteration t is defined as $rCG^t = \{\mathbf{RL}_1^t, \mathbf{RL}_2^t, \dots, \mathbf{RL}_n^t\}$, where \mathbf{RL}_i^t is the *route list*, or the sequence of nodes on the routing path from FN i to BS in iteration t . A *valid* routing configuration is the one satisfying the *Constraint II*, or Equation 3.2 defined below. Combining both position and routing configurations, we define the *configuration* at iteration t as $CG^t = \{pCG^t, rCG^t\}$. A *valid* configuration is the one that satisfies *Constraints I & II* given below and \mathcal{CG}_v^t denotes the set of all valid configurations in iteration t .

Wireless Communication Model: We analyze the wireless communication in area exploration with the following properties: (1) The unit-disk communication model is adopted and two nodes

i, j can communicate as long as $dist(i, j) \leq r_c$. (2) All links have a uniform *link capacity* $R_{capacity}$ and a FN has a uniform *flow sending rate* R_s for its video/audio streams not less than their stream floor rate in STI, where a *flow* is the combined video and audio streams from a FN to BS. (3) Streams are sent in a single selected path rather than multiple paths. (4) Interference between nodes will be analyzed in future work.

Terms	Definitions
BS	Base station.
FN, RN	Frontier node and relay node. \mathbb{FN}, \mathbb{RN} represent their set.
n, n_{fn}, n_{rn}	Number of all robots, number of FN and number of RN.
STI	Sensing and transmitting interval.
Flow	Combined video and audio streams from a FN to BS.
r_c, r_s	Communication range and sensing range.
\mathbb{FC}	Set of Frontier cells.
P_i^t	Position of node i in iteration t .
$G(t)$	Dynamic graph, $G(t) = \{V, E(t)\}$.
Map^t	Explored map in iteration t .
$Path_i^t, Path^t$	Moving path of node i and set of all robots moving path from iteration $t-1$ to t .
pCG^t	Position configuration, array of node positions in iteration t .
rCG^t	Routing configuration, path set of data flows from FN to BS.
CG^t	Configuration, $CG^t = \{pCG^t, rCG^t\}$.
\mathbb{CG}_v^t	Set of all <i>valid</i> configurations in iteration t .
IG_i^t	Information gain of node i in iteration t : Number of unvisited cells to be explored.
tIG^t	Total information gain.
T^t, mT^t	Total time elapsed after iteration t ; Migration time in iteration t .
γ in Eq. 3.2	Bandwidth ratio.
$R_{capacity}$	A uniform maximum link rate.
R_s	A uniform flow sending rate from FN.

Table 3.1. Notations in CBAX.

Constraints I & II: In the STI of each iteration t :

1. *Constraint I.* The robot team with BS's network $G(t)$ is connected.

2. *Constraint II.* The aggregated consumed bandwidth cannot exceed the link capacity on each link of the path from FNs to BS, which can also be expressed as:

$$R_{capacity}^e \geq \sum_{f \in F_e} R_{consumed}^e(f) \quad (3.1)$$

f represents a flow and F_e denotes all the flows passing through link e . With the properties in our wireless communication model, equation 3.1 is rewritten as:

$$R_{capacity}^e \geq \gamma \cdot R_s \quad (3.2)$$

An integer γ , or the *bandwidth ratio*, denotes the maximum number of flows that can pass through link e without overflowing the link, e.g., $\gamma = 3$ when $R_s=450\text{KB/s}$ with 11Mbps link; $\gamma = 5$ when $R_s=128\text{KB/s}$ with 5Mbps link.

Illustration: When iteration t begins, BS computes CG^t and sends P_i^t and $Path_i^t$ to each node i . When receiving them, i will migrate to the targeting position P_i^t by path $Path_i^t$ from P_i^{t-1} and send the acknowledgement back to the BS when it arrives. mT^t , or the *migration time* in iteration t is determined by the bottleneck, or the longest moving time among all nodes. After the BS receives acknowledgements from all nodes, it informs all FNs to enter STI and starts the stream transmission to the BS by the routes defined in rCG^t . Then the human operator can start to identify targets in STI by monitoring the video/audio sent from FNs. An iteration adjourns with the completion of STI. Fig. 3.2 also illustrates a robot's states in exploration.

3.2.2 Problem Formulation

Based on the system model, the major problem is how to explore the whole area for a robot team with minimum time *under the communication constraints I & II*. Although the problem can be

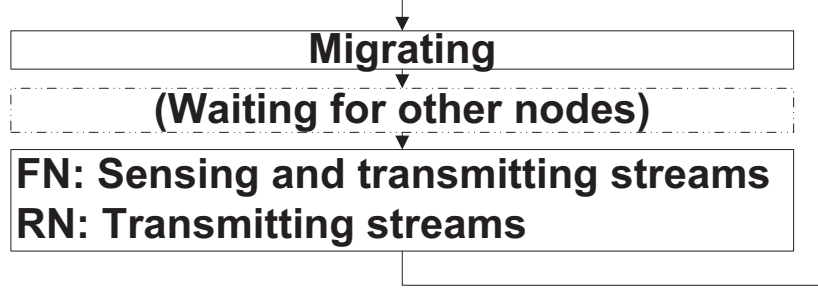


Figure 3.2. Robot status in one iteration.

difficult to solve as a whole, our solution leverages a heuristic answer to a subproblem. The subproblem becomes more tractable but remains non-trivial, since it is further divided into variants of two NP-hard problems and a polynomially-solvable problem. This subproblem is to find a *local optimal configuration* in iteration t , or CG_{opt}^t , which has maximal total number of cells explored in unit migration time. Formally, the subproblem is: given pCG^{t-1} , find CG_{opt}^t such that:

$$CG_{opt}^t = \operatorname{argmax}_{CG^t \in \mathbb{CG}_v^t} \left(\frac{tIG^t}{mT^t} \right) \quad (3.3)$$

3.3 Connectivity and Bandwidth Aware eXploration (CBAX)

3.3.1 CBAX Overview

Generally, CBAX attempts to enhance efficiency by maximizing the explored area by placing more robots in frontier areas, while reserving less robots as relay nodes. Additionally we endeavor to reduce the bottleneck longest distance that robots travel, as it also directly impacts the exploration time. The problem of finding CG_{opt}^t in Section 3.2.2 is largely divided into three subproblems:

- Frontier node placement: Where to place n_{fn} number of FNs in \mathbb{FC} to cover maximum amount of unexplored area? The distances from current positions to new ones are also con-

sidered and modeled in the utility function.

- Relay node placement with routing path selection: How to place minimum number of RNs in explored area to satisfy the *Constraint I,II* and select which paths to route flows from FNs to BS?
- Position assignment and path generation: How to assign each robot with its target position and generate movement paths to minimize the bottleneck time mT^t ?

Algorithm 1: General Procedure of CBAX

Input: $pCG^{t-1}, Map^{t-1}, T^{t-1}$, Robot team size n

Output: $CG^t, Path^t, mT^t, T^t, Map^t$

```
1  $n_{fn} \leftarrow n$  // “tmp” means “temporary” throughout this dissertation
2 while  $n_{fn} > 0$  do
3    $\mathbb{FN}^t \leftarrow placeFN(Map^{t-1}, n_{fn})$  // in Alg. 2
4    $rCG^t, \mathbb{RN}^t, n_{rn} \leftarrow placeRN\_selectRoutingPath(\mathbb{FN}^t)$  // in Alg. 3
5   if  $n_{fn} + n_{rn} \leq n$  then
6     if  $n_{fn} + n_{rn} < n$  then
7        $pCG^t \leftarrow placeRemainNode(pCG^{t-1}, pCG^t)$  // in Alg. 8
8     break
9   else
10    reset  $\mathbb{FN}^t, \mathbb{RN}^t, rCG^t, n_{rn}$ 
11     $n_{fn} \leftarrow n_{fn} - 1$ 
12 Generate migration paths with time:  $mT^t, T^t, Path^t \leftarrow genPath(pCG^{t-1}, T^{t-1})$  // in Alg. 7
13  $Map^t \leftarrow search(Map^{t-1}, CG^t)$ 
```

Alg. 1 illustrates the general procedure of CBAX for each iteration. Initially, n_{fn} is set as the team size because more FNs normally can cover more area. Given FN positions computed in line 3, RNs are placed accordingly with routing to relay FNs with BS and n_{rn} is decided. Line 5 checks whether required number of robots $n_{fn} + n_{rn}$ exceeds the fixed team size. If yes, the algorithm reduces n_{fn} and places FNs and RNs again until it finds the required number of robots

can be satisfied. If $n_{fn} + n_{rn} < n$, then the remaining nodes may be able to move to frontier areas to explore extra areas where RNs can support the additional flows. In line 12, function `genPath` is called to assure that bottleneck distance is minimized when generating movement paths for each node to reach its target. The map will be updated after nodes arrive at their new positions and enter STI.

CBAX is a centralized scheme. Although it takes time to synchronize and transmit control messages to the BS, it fits our model as global topology information is requested by the operator at the BS anyway for remote monitoring, control, and even teleoperation. More notably, it facilitates graph theoretic modeling and methodically solving the problem. Furthermore, a distributed approach usually suffers from local suboptimal results, large amount of local data exchange [40], and non-trivial waiting time for other nodes to broadcast better results [34]. Therefore, CBAX computes robot's trajectories at the powerful base station in a safe area to ensure effective control and monitoring.

3.3.2 Frontier Robot Placement

Algorithm 2: Frontier Node Placement *placeFN()*

Input: *Map* as a copy of Map^{t-1} ; *i* as a copy of n_{fn}

Output: \mathbb{FN}^t

```

1 while  $i > 0$  do
2    $q^* \leftarrow \operatorname{argmax}_{q \in \mathbb{FC}} U(q)$ 
3    $\mathbb{FN}^t.add(q^*)$ 
4    $Map \leftarrow \operatorname{updateLocalMap}(q^*, Map)$ 
5    $\mathbb{FC} \leftarrow \operatorname{updateFrontierPosSet}(Map, \mathbb{FC})$ 
6    $i \leftarrow i - 1$ 

```

Our utility function is defined as follows:

$$U(q) = IG_q^t \cdot e^{-\frac{d(q)}{\theta}}, \quad (3.4)$$

$$\text{where } d(q) = \min_{i \in pCG^{t-1}} d(q, i),$$

$$\theta = \max\{\theta_0 \cdot (1 - \delta), \theta_{th}\} \quad \left(\delta = \frac{\operatorname{size}(Map^{t-1})}{\operatorname{size}(Map_{total})}\right)$$

Where δ is the *exploration ratio*, or explored area over total area. Empirically, θ_0 is set as 20 and θ_{th} (the threshold) is set as 12. When $\theta \rightarrow \infty$, distance is not considered and FNs are greedily placed as the positions that cover most. On the other hand, when $\theta \rightarrow 0$ the distance is weighed heavily so the frontier cell closest to the current positions pCG^{t-1} is selected. Thus, θ defined in equation 3.4 enhances performance with initial emphasis on coverage while focusing on migrating time in the end to avoid unnecessary fluctuating moves. This approach resembles the candidate evaluation method in [92] but shows disparities in setting θ dynamically to adjust the distance's weight and defining distance as the minimum from one point to a set of positions.

This problem can also be viewed as a variant of the well-known NP-hard *Set Cover* problem [93] when $\theta \rightarrow \infty$. In the set cover problem, the minimum number of sets are selected so that all elements are contained, while in our problem a fixed number of sets are selected with the goal that the maximum of elements are covered when assuming a *set* is the unvisited cell set by each frontier cell and an *element* as an unvisited cell. It is not difficult to prove that our problem is *no easier than* the set cover (thus is also NP-hard) because a polynomial solution to ours, if it ever exists, will lead to a polynomial solution to the set cover. To sum up, our solution inherits a greedy approximation of set cover, along with dynamic cost modeling of migrating distance.

<i>Terms</i>	<i>Definitions</i>
δ	Exploration ratio. Explored area compared with total area.
$G_f(t)$	Hybrid flow graph, $G_f(t) = \{V, E(t) \cup \vec{E}(t)\}$.
$dist(i, j)$	Straight line distance from i to j .
$path(i, j)$	Shortest obstacle-aware migrating path from i to j .
L_{cur}, L_{new}	The current layer and the new layer.
N_i	Node i 's Neighbor node set excluding L_{new} , j is a neighbor of i when $dist(i, j) \leq r_c$.
CH, CH_{cur}, i_{ch}	Cluster head, vector of CH of current layer, and cluster head node i .
$FS_{\vec{e}}, FS_i$	Vector of carried and to-be relayed Flow Source passing through edge \vec{e} and node i .
CF_i, FCN_i	Covered Flows (CF) and Fully Covered Nodes (FCN) of node i . $ CF_i \leq \gamma$.
RL_i^t	Route List. Sequence of nodes on routing path from FN i to BS in iteration t .
N_{BS}^-	Unsaturated 1-hop Neighbors to the BS.
n_{remain}	Number of remaining nodes.

Table 3.2. Notations in Algorithm 1-7 in CBAX.

3.3.3 Relay Robot Placement with Routing Path Selection

Now that FNs are placed, we need to find (1) whether the rest $(n - n_{fn})$ nodes are adequate to relay BS and FNs and how to place them; (2) If the nodes are abundant, then how to route the streams from each FN to BS. We present two solutions with respect to situations considering different bandwidth ratio γ defined in equation 3.2.

Algorithm 3:	Relay Node Placement With Routing Selection	<i>placement</i>
<hr/> <i>eRN_RoutingPathSelection()</i> <hr/>		
Input: pCG^t		
Output: CG^t, n_{rn}		
1 if isBandwidthSufficient then		
2	$n_{rn}, pCG^t \leftarrow \text{SteinerizedMST}(\mathbb{FN}^t)$	// in Alg. 4
3	$rCG^t \leftarrow \text{BFS_RoutingPath}(pCG^t)$	
4 else		
5	$CG^t, n_{rn} \leftarrow \text{ConstrainedSteinerTree_Routing}(\mathbb{FN}^t)$	// in Alg. 5

- When γ is sufficiently large, the first problem can also be viewed as a decision version of *Steiner Minimum Tree Problem with Minimum Number of Steiner Points and bounded edge length (SMT-MSP)* where FNs and BS are the terminal points and RNs are the Steiner points. The routing problem can be easily solved afterwards.
- When γ is small and the bandwidth of aggregated flows may exceed link capacity, we model the problem as a decision version of SMT-MSP with the γ -inflow constraint and present our solution.

Bandwidth-Sufficient Relay Node Placement and Routing

The problem is modeled as the NP-hard problem of SMT-MSP, which was introduced and investigated in [43, 91]. First we build a *complete graph*¹ G_{comp} of BS and FNs. In the next step, our solution is built on the well-known approximation algorithm called *Steinerized Minimum Spanning Tree* for SMT-MSP, with an approximation ratio of 4 [91]. Interested readers may try more sophisticated schemes in [43] with ratio of 3 and 2.5. Since the ratio is only an upper bound compared with the optimal, the improvement in our application may not be that significant. When the RN is placed at an obstructed cell, *A* search* [94] is adopted to compute the obstacle-aware shortest path from one end (starting point) to the other end (goal) of the edge (line 7 in Alg. 4) and the RN is placed again on the new path. We use the straight-line distance to the goal as the *admissible*² and *consistent* heuristic function $h(n)$ that never overestimates the distance to goal. A* search guarantees the same optimal result as Dijkstra while reducing the searching space. With RNs placed and tree constructed, the shortest routing path (in terms of number of hops) from each FN to BS can be obtained by breath-first search (BFS) (line 3 in Alg. 3). We are able to use BFS because only the

¹A complete graph is a simple graph in which every pair of distinct vertices is connected by an edge.

²Most admissible $h(n)$ are also consistent. Details of the proof of consistency is in [94].

number of intermediate nodes counts while the weight of each edge is not considered.

Algorithm 4: Bandwidth-Sufficient Relay Node Placement: Steinerized Minimum Spanning

Tree *SteinerizedMST()*

Input: \mathbb{FN}^t

Output: pCG^t

- 1 Construct a complete graph G_{cmp} from $\{\mathbb{FN}^t \cup BS\}$
 - 2 Construct a MST T_{mst} from G_{cmp} , $T \leftarrow T_{mst}$
 - 3 $\mathbb{RN}^t.add(\text{non-leaf nodes in } T)$
 - 4 **foreach** edge e in T whose length $l(e) > r_c$ **do**
 - 5 Cut long edge e into $\lceil l(e)/r_c \rceil$ shorter ones of equal length by placing $\lceil l(e)/r_c \rceil - 1$ Steiner points $\{sp\}_e$ on e
 - 6 **if** $\exists i(i \in \{sp\}_e \wedge P_i = \text{obstructed})$ **then**
 - 7 Update e by: $e \leftarrow A^*Search(e)$
 - 8 Place Steiner points again on e and update $\{sp\}_e$
 - 9 $\mathbb{RN}^t.add(\{sp\}_e)$
 - 10 Replace edge e with $|\{sp\}_e|+1$ shorter edges
 - 11 $n_{rn} \leftarrow |\mathbb{RN}^t|$
 - 12 $pCG^t \leftarrow \mathbb{FN} + \mathbb{RN}$
-

Bandwidth-Constrained Relay Node Placement and Routing

When bandwidth ratio γ is small, the problem is modeled as SMT-MSP with the constraint that each node can admit no more than γ flows from FNs (FNs may also relay flows). Alternatively, if we define a node as *saturated* when it carries γ flows and as *over-saturated* when it carries more

than γ ones, the problem becomes SMT-MSP without *over-saturated* nodes when flows transmit from $n - 1$ terminal points to one terminal point (BS). The problem is *as hard as* SMT-MSP (thus is also NP-hard) since if we can find a polynomial algorithm for it and we set $\gamma \rightarrow \infty$, SMT-MSP would be polynomially solvable.

The solution's central theme is to reduce unnecessary RNs by *aggregating flows* and the following definitions relevant to flow aggregation are introduced. First, on top of the dynamic graph $G(t)$ of FNs and BS, a *hybrid flow graph* $G_f(t) = \{V, E(t) \cup \vec{E}(t)\}$ is defined where an undirected edge e denotes nodes in communication range while a directed edge \vec{e} represents not only connected vertices, but also directed flow(s) from FNs to BS. Second, the *Vector of carried Flow Source* for each \vec{e} , or $\mathbb{FS}_{\vec{e}}$, denotes the vector of sources of the flows passing through this edge. For each node i , \mathbb{FS}_i gives the *carried and to-be relayed Flows Source vector* (Sources here are the FNs which produce flows). Third, L_{cur} gives the *current layer* of nodes and L_{new} gives the *new layer*. Fourth, a *cluster head* \mathbf{CH} is defined as the node that aggregates flows in each layer and \mathbf{CH}_{cur} denotes the cluster head vector of current layer nodes. Last, but not least, $path(i, j)$ denotes the shortest obstacle-aware migrating path from i to j and \mathbb{N}_i is node i 's neighbor node set excluding L_{new} (j is a neighbor of i when $dist(i, j) \leq r_c$).

Additionally, we define \mathbb{CF}_i as the *Vector of Covered Flow* of i , in which each element is a linked list that begins with the node ID $j \in \mathbb{N}_i \cup i$ as the *head* and continues with sources from \mathbb{FS}_j (An example has been provided in Fig. 3.5. Note that \mathbb{CF}_i depicts the potential flows that may be carried where \mathbb{FS}_i describes the flows that have already been carried. A node $j \in \mathbb{N}_i \cup i$ is *fully covered* by node i when *all* of j 's flows can be aggregated by i . \mathbb{FCN}_i denotes the vector of i 's *Fully Covered Nodes* ($j \in \mathbb{FCN}_i$ iff $j \in \mathbb{N}_i \cup i \wedge \mathbb{FS}_j \subseteq \mathbb{CF}_i$). Note that $|\mathbb{CF}_i| \leq \gamma$ and $|\mathbb{CF}_i|$ only counts the source nodes. Each node must first carry its own flow if it is in L_{cur} , or its parent's

flow if it is a RN in L_{new} . Then $|\mathbb{FCN}_i| - 1$ denotes the number of extra RNs can be saved with such aggregation using the remaining capacity.

Flow aggregation: To minimize RN size, flow aggregation is executed in an order with maximal $|\mathbb{FCN}_i|$ first. The problem of finding maximal $|\mathbb{FCN}_i|$ is solved optimally by greedily covering minimal $|\mathbb{FS}_i|$ node first. For example, in Fig. 3.5, new layer RN node n can maximally cover two extra nodes, or save two RNs immediately otherwise needed by node 3 and 4. n can aggregate all flows from them: n 's $\mathbb{FCN}_n = \{3, 4\}$ with the remaining capacity $9-4=5$ after it obtains its parent node 1's four flows.

Major strategy with illustration: RN placement and routing path selection are solved jointly in a layer-based approach, which is also described in Alg. 5 and illustrated in an example in Fig. 3.3 and 3.4. First, FNs \mathbb{FN}^t are set as L_{cur} and covered nodes/flows are computed for each node $i \in L_{cur}$ based on \mathbb{FS}_i , which is initialized as itself being the only element since a node carries its own flow (line 3). With L_{cur} , the algorithm is to compute L_{new} as new RN until all current layer nodes are in range of BS, which is the terminating condition gives in line 4.

Computing L_{new} is executed in two steps: (1) Clustering nodes and aggregating flows in L_{cur} with CH selection and RN placement. (2) Aggregating flows into L_{new} with L_{new} RN filtering,

which are given in line 5-12, 13-22 respectively.

Algorithm 5: Bandwidth-Constrained Relay Node Placement With Routing Path Selection

ConstrainedSteinerTree_Routing()

Input: \mathbb{FN}^t

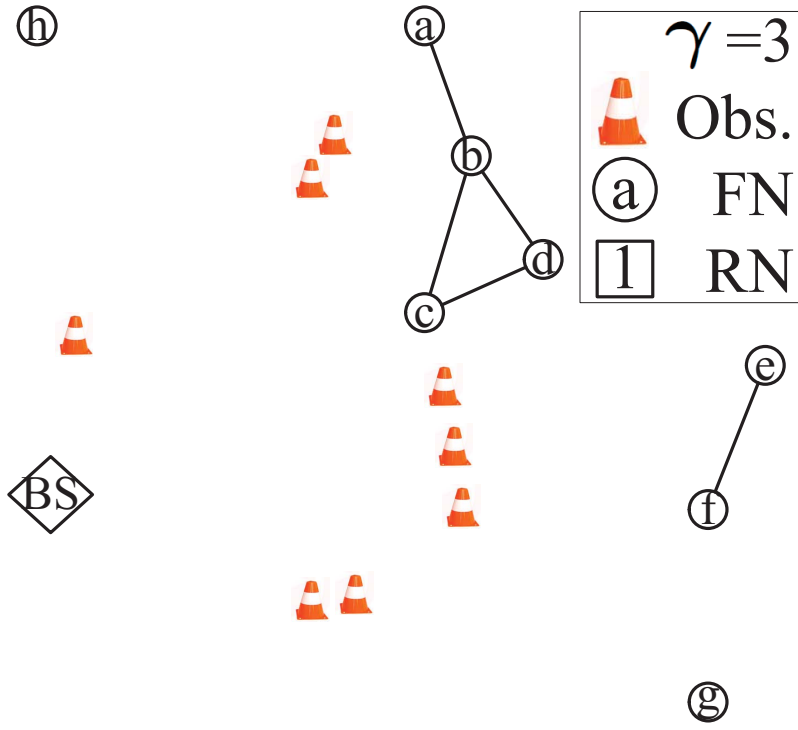
Output: CG^t, n_{rn}

```

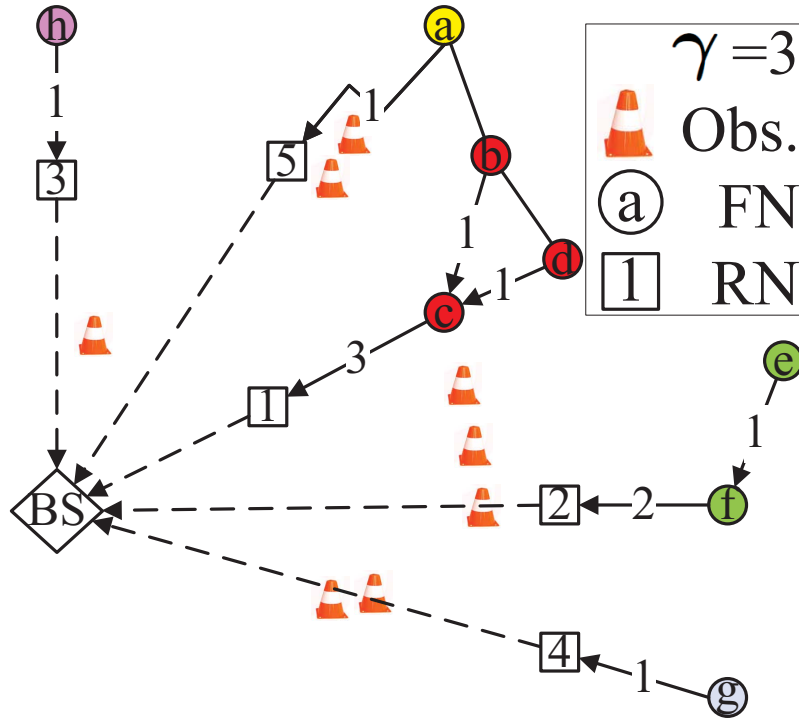
1   $L_{cur} \leftarrow \mathbb{FN}^t, L_{new} \leftarrow \emptyset$  // FNs as current layer
2  foreach  $i \in L_{cur}$  do
3     $\text{initialize}(\mathbb{FS}_i), \text{setCoveredFlow\_Node}(i, \mathbb{N}_i \cup i, \text{false})$ 
4  while  $\exists \text{ node } i (i \in L_{cur} \wedge \text{dist}(i, BS) > r_c)$  do
5     $L_{cur}^{tmp} \leftarrow L_{cur}$ 
6    while  $L_{cur}^{tmp} \neq \emptyset$  do
7       $i_{ch} \leftarrow \max |\mathbb{FCN}_i| \text{ node } i \in L_{cur}^{tmp} \text{ with min } \text{dist}(i, BS)$  // Select cluster head
8       $\mathbb{CH}_{cur}.\text{add}(i_{ch}), L_{cur}^{tmp}.\text{remove}(\mathbb{FCN}_{i_{ch}})$ 
9       $\text{updateFlow\_Route}(i_{ch})$  // in Alg. 6
10      $\text{placeRN\_along\_path}(i_{ch})$  // Generate  $L_{new}$  in Alg. 6
11     foreach  $i \in L_{cur}^{tmp}$  do
12        $\text{setCoveredFlow\_Node}(i, \mathbb{N}_i \cup i, \text{false})$  // in Alg. 6
13    $L_{new}^{tmp} \leftarrow L_{new}$ 
14   foreach  $i \in L_{new}^{tmp}$  do  $\text{setCoveredFlow\_Node}(i, \mathbb{N}_i, \text{true})$ 
15   while  $L_{new}^{tmp} \neq \emptyset$  do
16      $i^\dagger \leftarrow \max |\mathbb{FCN}_i| \text{ node } i \in L_{new}^{tmp} \text{ with max } |\mathbb{CF}_i|$ 
17     if  $|\mathbb{FS}_{i^\dagger}.\text{parent}| = 0$  then  $L_{new}.\text{remove}(i^\dagger)$  // Filtered
18      $L_{new}^{tmp}.\text{remove}(i^\dagger)$ 
19      $\text{updateFlow\_Route}(i^\dagger)$ 
20     foreach  $i \in L_{new}^{tmp}$  do  $\text{setCoveredFlow\_Node}(i, \mathbb{N}_i, \text{true})$ 
21   Add  $L_{new}$  to  $\mathbb{RN}^t$ 
22   Set  $L_{new}$  as  $L_{cur}$ ; Clear  $L_{new}$ 
23  $n_{rn} \leftarrow |\mathbb{RN}^t|, pCG^t \leftarrow \mathbb{FN}^t + \mathbb{RN}^t$ 
24 foreach  $i$  do  $rCG^t.\text{add}(\mathbb{RL}_i^t)$ 
25  $CG^t \leftarrow pCG^t \cup rCG^t$ 

```

In the first step, we repeatedly select cluster heads **CH** in L_{cur} and inserts them into \mathbb{CH}_{cur}



(a) With FNs and BS only.



(b) Placing 1st layer RNs.

Figure 3.3. Example of bandwidth-constrained relay node placement and routing (Step 1 and 2 out of 4 total steps). ($\gamma=3$) Edge weight denotes $|\mathbb{FS}_{\vec{e}}|$

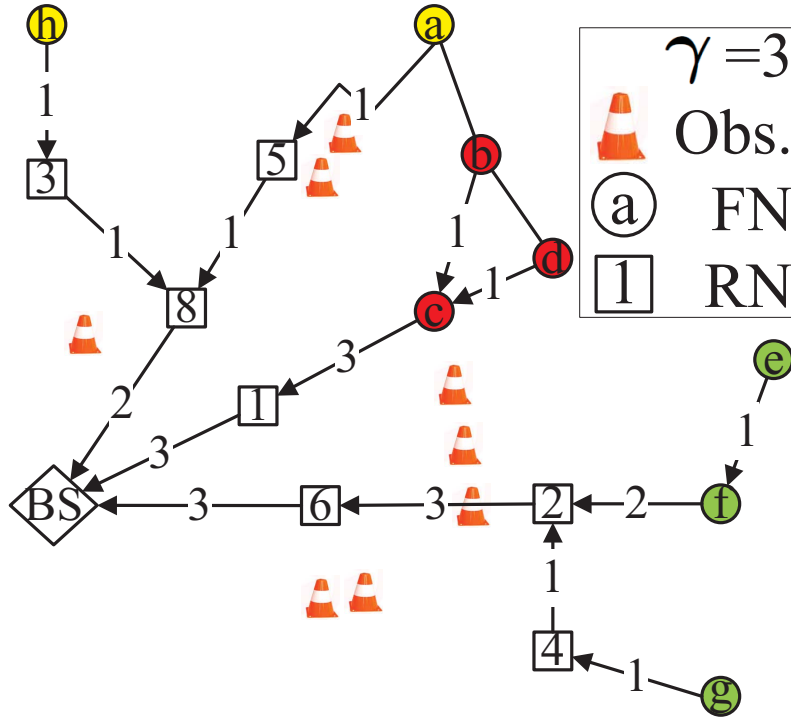
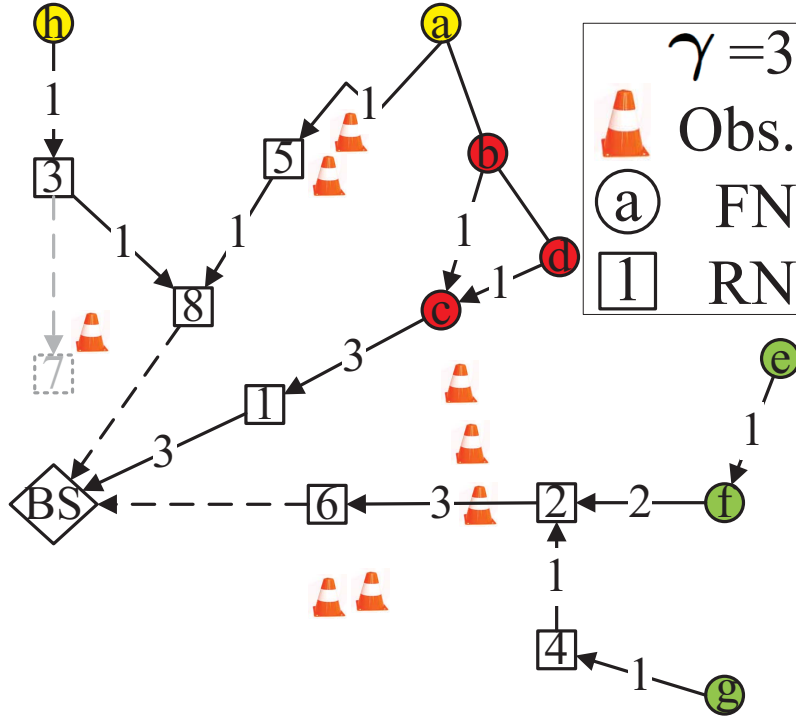


Figure 3.4. Example of bandwidth-constrained relay node placement and routing (Step 3 and 4 out of 4 total steps). ($\gamma=3$) Edge weight denotes $|\mathbb{FS}_{\vec{e}}|$

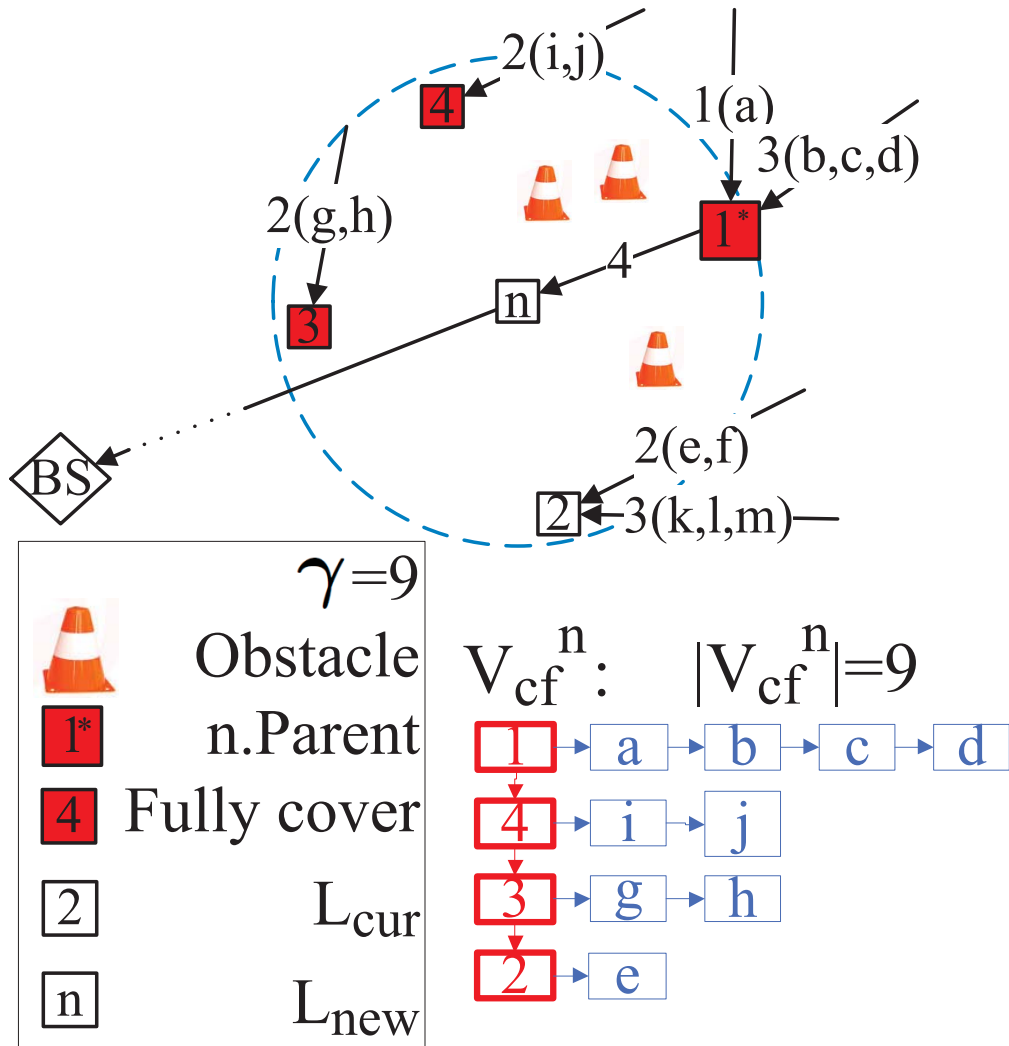


Figure 3.5. Aggregate flows to nodes with maximal number of fully covered neighbors first to conserve RNs.

to aggregate flow in the cluster in the order of “maximal $|\mathbb{FCN}_i|$ first” (Choose minimum distance to BS if $|\mathbb{FCN}_i|$ are equal since the shorter distance indicates fewer needed RNs). Moreover, we place RNs when needed to form L_{new} in the paths from \mathbb{CH}_{cur} to BS and A* search is used similarly as in Alg. 4 to avoid obstacle. Note that i_{ch} is set as the new RN’s *parent*. Function *updateFlow_Route* (i) (line 15-19 in Alg. 6) describes how a node i aggregates flows by moving elements in neighbor node j ’s \mathbb{FS}_j to \mathbb{FS}_i following i ’s covered flow \mathbb{CF}_i . Moreover, source s ’s \mathbb{RL}_s^t is updated with the new RN appended. For instance, in Fig. 3.3(a), FN c is chosen as cluster head since it is the closest to BS among the ones with maximal $|\mathbb{FCN}_i| = 3$. \mathbb{FS}_c is updated as

$\{c, b, d\}$ and c is added in \mathbb{RL}_b^t and \mathbb{RL}_d^t for sources b and d .

Algorithm 6: Supporting Functions for Algorithm 5

```

    // Obtain (pre)knowledge on covered flows (CF) and fully covered nodes (FCN); not to
    // transmit flows yet.
1  setCoveredFlow_Node (node  $i$ , node set  $S$ , bool  $isNewRN$ ):
2   $\mathbb{CF}_i.clear()$ ,  $\mathbb{FCN}_i.clear()$ 
3  if  $isNewRN = true$  then
4  |    $S.remove(i.parent)$ 
5  |    $\mathbb{CF}_i.add(i.parent \rightarrow \mathbb{FS}_{i.parent})$ ,  $\mathbb{FCN}_i.add(i.parent)$  // Obtain all of its parent's flows
6  else
7  |    $S.remove(i)$ 
8  |    $\mathbb{CF}_i.add(i \rightarrow \mathbb{FS}_i)$ ,  $\mathbb{FCN}_i.add(i)$  // Obtain all of its own flows
9  while  $|\mathbb{CF}_i| < \gamma \wedge (\exists \text{ node } k \in S \text{ whose } |\mathbb{FS}| \neq 0)$  do
10 |    $j \leftarrow \text{such node } k \text{ with min } |\mathbb{FS}|$ 
11 |    $S.remove(j)$ 
12 |   while  $|\mathbb{CF}_i| < \gamma$  do
13 |   |    $\mathbb{CF}_i.add(j \rightarrow \text{distinct source } k \in \mathbb{FS}_j)$  //  $\mathbb{CF}_i.j \rightarrow \{k\}$ 
14 |   |   if  $\mathbb{FS}_j \subseteq \mathbb{CF}_i$  then  $\mathbb{FCN}_i.add(j)$  // Fully covered node

    // To transmit flows.
15 updateFlow_Route (node  $i$ ): // Function begins
16 foreach neighbor (head in  $\mathbb{CF}$ )  $j \in \mathbb{CF}_i$ . *  $\wedge j \neq i$  do
17 |   foreach FN  $s \in \text{source nodes for flows carried by } j$  do
18 |   |   Add  $i$  as a relay for FN  $s$  in  $\mathbb{RL}_s$ 
19 |   Transmit  $j$ 's flows from  $j$  to  $i$  by updating their  $\mathbb{FS}$ 

20 placeRN_along_path ( $i$ ): // Function begins
21 if  $dist(i, BS) > r_c$  then // Place RN if needed
22 |   Place relay  $rn$  at  $dist(rn, i) = r_c$  on the obstacle aware path from  $i$  to  $BS$  generated by
    |   A* search( $i, BS$ )
23 |   Set  $rn$ 's parent as  $i$ ; Add  $rn$  to  $L_{new}$ .

```

In the second step, CBAX executes flow aggregation with the same order of “maximal $|\mathbb{FCN}_i|$

first” (Choose maximum size of covered flows if $|\mathbb{FCN}_i|$ are equal) for L_{new} and attempts to remove unnecessary nodes by filtering the new layer nodes whose parent’s flows have all been carried by others. To illustrate in Fig. 3.4(a), new layer RN_8 is first to aggregate flows with $|\mathbb{FCN}_{RN_8}| = 2$ and then RN_7 has $|\mathbb{FS}_{RN_3}| = 0$ and is consequently filtered.

A layer placement ends when L_{new}^{tmp} is empty (line 15). Then L_{new} becomes L_{cur} and a new round starts. To conclude, the flows, denoted by \mathbb{FS}_i for node i , are aggregated in CH in L_{cur} and then in L_{new} with best endeavor to reduce the number of RNs.

Update flow information: In the beginning of the algorithm and after both flow aggregation in the two steps, function *setCoveredFlow_Node* (line 1-14 in Alg. 6) is called to update the covered flows and fully covered node information. Furthermore, rest of the nodes still in L_{new}^{tmp} are updated in line 20 and these updates are necessary for the next round computation. With $\gamma \rightarrow \infty$, this flow aggregation based relay placement algorithm also fits the sufficient bandwidth situation.

Upper bound of approximation ratio: We derive the upper bound of the approximation ratio of the bandwidth-constrained relay placement algorithm.

First we define some notations. A maximum degree in a graph G is: $\Delta(G) = \max\{deg_G(v)|v \in V(G)\}$. We denote n , n_s , n_d , and n_{opt} are the respective relay size obtained by: (i) flow constrained relay placement algorithm; (ii) S-MST without the flow constraint; (iii) DT: direct tree, or a simple aggregation tree (star graph with center BS) where all flows from FNs are directly relayed to BS without aggregation; and (iv) T_{opt} : An optimal Steiner tree with flow constraint and minimum number of RN.

In addition, MST denotes the minimum spanning tree for FNs and BS. DT and MST are two very simple tree structures easily obtained when given the input of FNs and BS. We define $\kappa =$

$n_d/n_s = \sum_{e \in E(DT)} \lfloor e/r_c \rfloor / \sum_{e \in E(MST)} \lfloor e/r_c \rfloor$, where κ is the ratio of relays size for DT over that of MST. Hence, κ is related to \mathbb{FN} and can be directly obtained from input.

Lemma 1. *Every Steiner tree satisfies the flow constraint must have its maximum degree $\Delta(T) \leq 1 + \gamma$.*

Proof. For a vertex, its incoming degree is constrained by γ , the number of incoming flows for aggregation. With only one out-coming link, the total degree is at most $\Delta(T) \leq 1 + \gamma$. \square

Lemma 2. *There exists a shortest optimal Steiner tree T_{opt} whose maximum degree is less than or equal to five, i.e., $\Delta(T_{opt}) \leq 5$, in an obstacle free area or obstructive area where triangular inequality holds.*

Proof. When the obstacle distribution does not impact the triangular inequality: with A* obstacle aware paths, the shortest side is still across from the smallest angle. Then we can prove two edges meeting at a vertex in a T_{opt} form an angle of at least 60 degree, by the proof similar to that in [91] on an Euclidean plane. Besides, authors in [91] also proved that a six degree tree can be modified to a 5 degree one. Therefore, $\Delta(T_{opt}) \leq 5$. \square

Theorem 1. *The approximation ratio of bandwidth constrained relay placement algorithm is no greater than $\kappa \cdot (1 + \gamma)$. In an environment where obstacle distribution does not affect the triangular inequality, the ratio is no greater than $\kappa \cdot \min\{5, 1 + \gamma\}$.*

Proof. With an optimal tree T_{opt} , we duplicate each edge and build an Eulerian tour such that every Steiner point appears at most $\Delta(T_{opt})$ times in the tour while each terminal point appears exactly once. Removing one (or more) edge in the tour will give a spanning tree T' where Steiner points have degree at most two. Its Steiner point size $n(T')$ is no more than $\Delta(T_{opt})$ times in T_{opt} .

Because n_s is from S-MST: a *minimum* spanning tree, we have:

$$n_s \leq n(T') \leq \Delta(T_{opt}) \cdot n_{opt} \quad (3.5)$$

With $\kappa = n_d/n_s$ and the fact that flow aggregation guarantees $n \leq n_d$, we have:

$$n \leq n_d \leq \kappa \cdot \Delta(T_{opt}) \cdot n_{opt} \quad (3.6)$$

$$n/n_{opt} \leq \kappa \cdot \Delta(T_{opt}) \quad (3.7)$$

With Lemma 3 and 4, $\Delta(T_{opt}) = \min\{5, 1 + \gamma\}$ in obstacle free area or obstructive area where triangular inequality holds. Otherwise: $\Delta(T_{opt}) = 1 + \gamma$. Replacing $\Delta(T_{opt})$ concludes the proof. \square

3.3.4 Position Assignment And Path Generation

With current positions pCG^{t-1} from the execution of last iteration of n robots and the computed n next positions pCG^t , there is a problem with how to assign each robot to go to which newly computed position, so that mT^t , which is decided by the bottleneck longest distance of all pairs, is minimum. We formally model this problem as the *linear bottleneck assignment problem (LBAP)* [95]. LBAP can also be viewed as finding a perfect matching in a weighted bipartite graph that minimizes the maximum weight of all matched edges [96]. Mathematically, LBAP is:

$$\begin{aligned} mT^t &= \min_{x_{ij} \in \{0,1\}} \max_{i,j} c_{ij}x_{ij}, \\ \text{subject to } \sum_{i=1}^n x_{ij} &= 1, \quad j = 1, \dots, n, \\ \sum_{j=1}^n x_{ij} &= 1, \quad i = 1, \dots, n, \\ x_{ij} &\in \{0, 1\}, \quad i, j = 1, \dots, n \end{aligned} \quad (3.8)$$

Where $\{c_{ij}\}_{n \times n}$ is the cost matrix and c_{ij} is the cost as the length of shortest obstacle-aware path by A* from node i to position j . Moreover, $\{x_{ij}\}_{n \times n}$ is the resulting binary matrix where $x_{ij}=1$ iff the node i is assigned to position j . We adopt the algorithm in [97] with expected running time $O(n^2)$, which generates the *optimal* result. Its general procedure is as follows: (1) From the original bipartite graph, a subgraph with $2n \ln n$ minimum-cost edges is selected. (2) In this subgraph, LBAP is solved by scheme in [98]. (3) If a perfect matching is not found with a low probability then, the matching is done in the original graph. According to [96,99], it is one of the

most efficient algorithms for LBAP.

Algorithm 7: LBAP Based Position Assignment and Path Generation with Collision Avoidance *genPath()*

Input: $pCG^{t-1}, pCG^t, T^{t-1}$

Output: $mT^t, T^t, Path^t$

```

1 Construct complete bipartite graph  $G_{cb} = (pCG^{t-1} + pCG^t, E)$ 

   // update edge weight as obstacle-aware ones

2 foreach  $e \in E_{G_{cb}}$  do  $e \leftarrow A^*$  search( $e$ )

   // Call bipartite graph and perfect matching based LBAP() in [97]

3  $mT^t, Path^t \leftarrow graphBasedLBAP(G_{cb})$ 

4  $mT^t, Path^t \leftarrow avoidCollision(mT^t, Path^t)$ 

5  $T^t \leftarrow T^{t-1} + mT^t$  // Algorithm ends

6  $avoidCollision(mT^t, Path^t)$ : // Function begins

7 if  $IntersectionExists(Path^t)$  then
8   | if  $SameTimeIntersectionExists(Path^t)$  then
9   | |  $mT^t \leftarrow RescheduleCollisionNodes(mT^t)$ 

```

Since the paths obtained may not be collision free, we adopt a simple scheduling to avoid collision (line 6-9 in Alg. 7). First we check whether intersection points of the paths exist. If true, then we will check whether the intersection is reached by different nodes simultaneously. If true again, collision may occur and different nodes may wait different amounts of time, e.g. node i waits i second(s) before reaching the intersection place to avoid such collision.

3.3.5 CBAX Enhancement

Besides the major steps above in CBAX, we improve the control message transmission and use remaining nodes for placement to further enhance CBAX's performance.

Filter Based Control Message Transmission

In each iteration, a node needs to notify BS by sending an "arrived" control message when arriving at the target position. Since a node in movement indicates it will send to BS its "arrived" message in the future, the communication overhead can be reduced by not forwarding packets to BS if a moving node j receives the "arrived" message from a stopped (arrived) node i . When j also arrives at the targeting position, j can append i 's information in j 's "arrived" message. The completion of the *last* mobile node's movement indicates the completion of all nodes in one iteration. Besides, it is possible that when an arrived node sends the message, the intermediate nodes on the path back to the BS are still moving. They may be out of reach of the sender and therefore not able to relay the message. Under such occasions, the arrived node will retransmit the packet on timeout until the BS sends acknowledgement to confirm it has received the message.

Remaining Node Placement

When $n_{fn} + n_{rn} < n$ (line 6-7 in Alg. 1), we need to identify the remaining node positions to maximize the exploration efficiency. Generally, we attempt to place nodes as FNs using the unsaturated paths to carry flows to the BS. First, we identify the unsaturated 1-hop neighboring nodes of BS \mathbb{N}_{BS}^- which have extra capacity not yet used. A node i is unsaturated when $|\mathbb{FS}_i| < \gamma$. Second, to place remaining nodes as FNs, we collect cells in \mathbb{FC} which are in range of the flow source nodes of \mathbb{N}_{BS}^- to form \mathbb{FC}^* . The remaining node positions are selected as the cell with the

maximum gain in \mathbb{FC}^* . When there are no unsaturated paths, we place the remaining node using the previous iteration position in pCG^{t-1} which is closest to the nearest node in pCG^t . Such placement avoids the remaining node relocation to be the bottleneck in assignment. Alg. 8 shows

the procedure in detail.

Algorithm 8: Remaining Node Placement *placeRemainNode()*

Input: pCG^{t-1}, pCG^t

Output: pCG^t

```

1   $n_{remain} \leftarrow n - (n_{fn} + n_{rn})$ 

   // Put unsaturated and close to BS's nodes in  $pCG^t$  to  $\mathbb{N}_{BS}^-$ 

2  foreach  $i \in pCG^t \wedge dist(i, BS) \leq r_c \wedge |\mathbb{FS}_i| < \gamma$  do  $\mathbb{N}_{BS}^-.add(i)$ 

3  while  $|\mathbb{N}_{BS}^-| \neq 0 \wedge n_{remain} > 0$  do
4       $\mathbb{FC}^* \leftarrow$  Frontier cells that are in range of flow sources of  $\mathbb{N}_{BS}^-$ 
5      foreach  $q \in \mathbb{FC}^*$  do Store corresponding node pair  $(i, j)$  of BS's neighbor  $i$  and flow
        source  $j: \{i, j | i \in \mathbb{N}_{BS}^-, j \in \mathbb{FS}_i \wedge dist(q, j) \leq r_c\}$ 
6       $q^* \leftarrow \operatorname{argmax}_{q \in \mathbb{FC}^*} U(q)$ ; add  $q^*$  to  $pCG^t$ .
7       $i^*, j^* \leftarrow q^*$ 's corresponding  $i, j$ 
8       $Map \leftarrow updateLocalMap(q^*, Map)$ 
9       $\mathbb{FC} \leftarrow updateFrontierPosSet(Map, \mathbb{FC})$ 
10     Add  $j^* \cup \mathbb{RL}_{j^*}^t$  to  $q^*$ 's route list
11     for  $k \in \mathbb{RL}_{q^*}^t$  do update  $\mathbb{FS}_k$ 
12     if  $|\mathbb{FS}_{i^*}| = \gamma$  then  $\mathbb{N}_{BS}^-.remove(i^*)$ 
13      $n_{remain} \leftarrow n_{remain} - 1$ 

14 while  $n_{remain} > 0$  do // No unsaturated paths
15      $q^* \leftarrow$  node  $q \in pCG^{t-1}$  with min distance to nearest node in  $pCG^t$ 
16      $pCG^t.add(q^*)$ 

```

3.4 Performance Evaluation

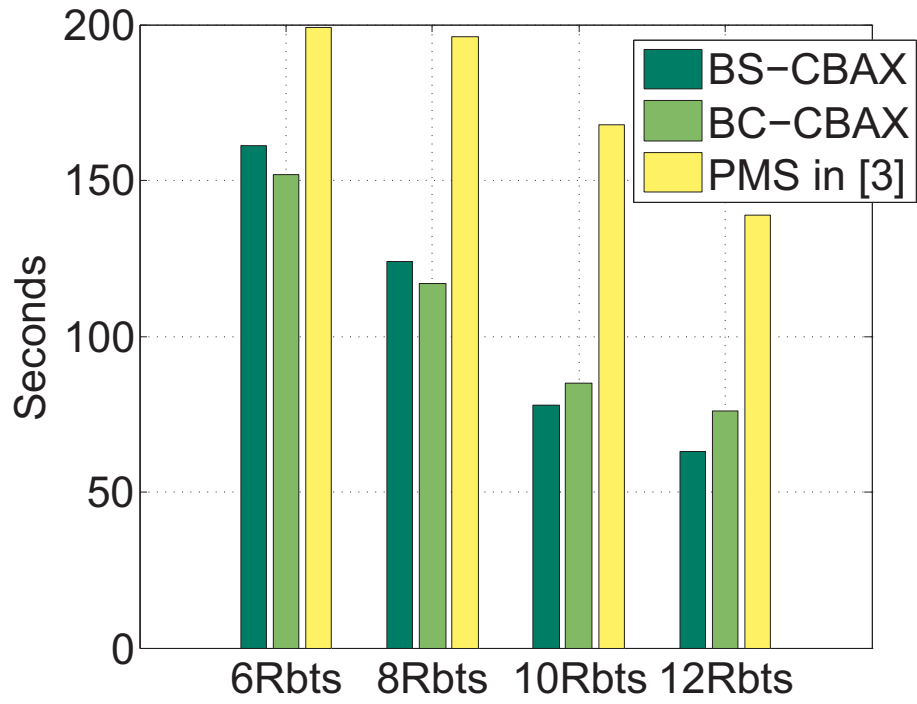
We evaluate CBAX’s performance using a multi-robot simulator in C++ and compare CBAX with two recent schemes that consider limited communication range: (1) A distributed “Sensor-based Random Graph” (**SRG**) scheme in [32], which is more efficient but does not guarantee connectivity; (2) A centralized “Possible Moves Sampling” (**PMS**) approach in [36] that assures connectivity but is less efficient. The metrics for comparison include exploration efficiency and communication quality. We use the total exploration time as the primary criteria for efficiency. If not stated otherwise, the time is measured from a clustered start at the left-bottom corner to a state with over 95% of explored area. Also, the constant STI in each iteration is not included in the measured time for the three schemes. To evaluate the communication quality, the *Non-Overflow Transmission Time ratio* (**NO_TT**), or the percentage of transmission time for not over-saturated flows divided by the total transmission time, is used. Besides, the percentage of iterations in which all robots maintain connected with BS and within the robot team, or *fully-Connected Time ratio* with the Base Station (**CT_BS**) and within the Robot Team (**CT_RT**), are also measured. Two environments, the *Open* and *Garden* in Fig. 3.7(a), are used to compare the schemes.

The parameters are set to make the comparisons fair. We obtain the results and parameters from SRG’s sample video in [100] and simulate CBAX using the same set of parameters (with proper scaling) and the same *Garden* environment. While teams with only 4,6,8 robots are tested in SRG, we simulate PMS and compare with CBAX on 6-12 robots to obtain results for more complex topologies using the *Open* environment. Specifically, the grid cell and robot size are 1m, the robot’s moving velocity is 1m/s, and the accelerating/decelerating time is ignored. Map sizes for the *Open* and *Garden* environments are 45m×45m and 48m×48m and (r_c, r_s) are (10m,7m)

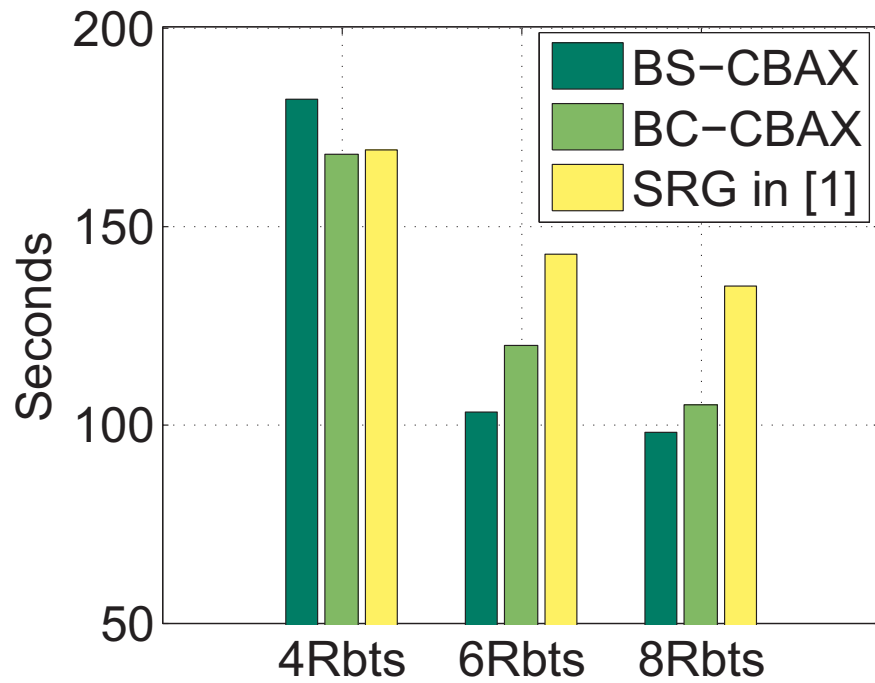
and (21m,7m) respectively. All experiments are run five times to obtain the average. For CBAX, two relay placement strategies are evaluated: **BS-CBAX**: CBAX with Bandwidth-Sufficient relay placement; **BC-CBAX**: CBAX with Bandwidth-Constrained relay placement ($\gamma=3$).

Exploration efficiency: Compared with PMS, BS-CBAX and BC-CBAX reduce explore time on average by 40.9% and 39.7% respectively as shown in Fig. 7.7. The improvement is more significant when team size increases. With 12 robots, the improvement are 54.7% and 45.3%. The major reason is that CBAX expends nodes promptly and systematically places nodes to reduce relay robots while in PMS, nodes are expended slowly especially when number of nodes increase. In addition, Fig. 7.8(b) shows that CBAX is more efficient than PMS with different percentages of explored area. Compared with SRG, CBAX remains the more efficient approach, even though SRG is a distributed and efficient scheme maintaining neither connectivity nor bandwidth adequacy. BS-CBAX and BC-CBAX decrease explore time on average by 15.9% and 13.0% respectively as shown in Fig. 7.8(a). With 8 robots, the improvement are 27.4% and 22.2%. In most cases, BC-CBAX is slightly more efficient than BS-CBAX when n is small but is less efficient when more robots are in the team. The reasons are: (1) Our flow aggregation based solution is more efficient than the general SMT based approximation algorithm. (2) The flows are less likely to be overflowed ($\gamma=3$) when node number is small. With more robots, however, the impact of the constraint is more noticeable, resulting in longer exploration time with more RNs placed.

Communication quality: Compared with PMS that only guarantees connectivity, CBAX further assures the real-time data flow of being under link capacity. In PMS, overflow transmissions occur in 21.5% of the iterations on average, in which 31.5% of data is lost. While CBAX always maintains connectivity and ensures no overflow occur, SRG with 4 robots shows that only in 20.7% of iterations all nodes are (through multi-hops) connected to BS and 48.3% of iterations all nodes are



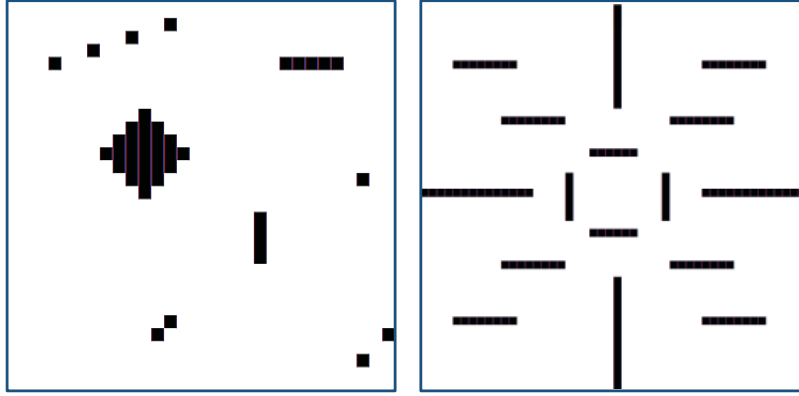
(a) Exploration time: CBAX vs. PMS in [36] with 6,8,10,12 of robots in the *open* environment.



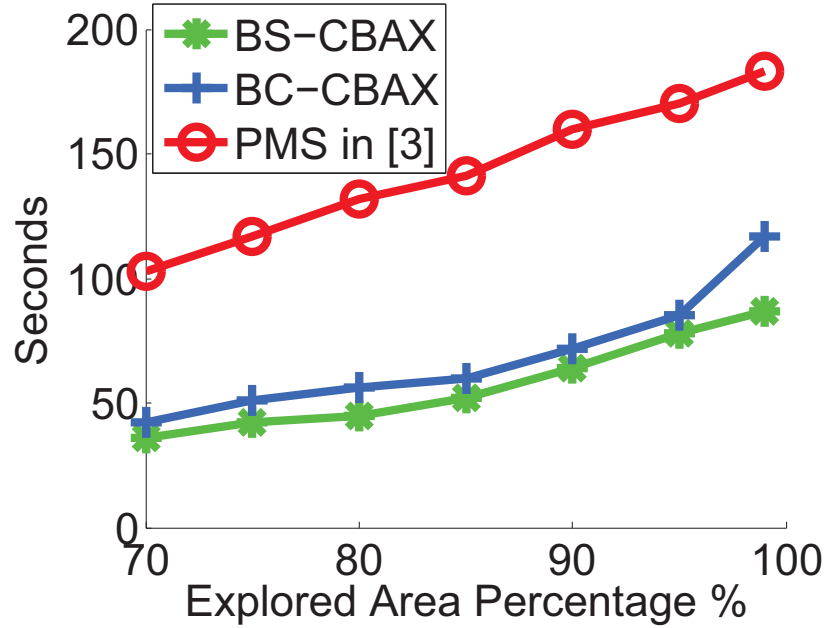
(b) Exploration time: CBAX vs. SRG in [32] with 4,6,8 robots in the *garden* environment.

Figure 3.6. Simulation results on exploration time.

connected within the robot team. Note that video available online is only with 4 robots. However, the results with 6/8 robots are similar since SRG is not aware of connectivity and bandwidth. Besides, in 79.3%³ of iterations overflow transmission occurs and 83.7% of data is lost in such overflow sessions. Fig. 3.8 also depicts the results.



(a) The *Open* and *Garden* environments where black parts denote obstacles.



(b) Exp. time: CBAX vs. PMS in [36] with varying percentages of explored area (10 Robots).

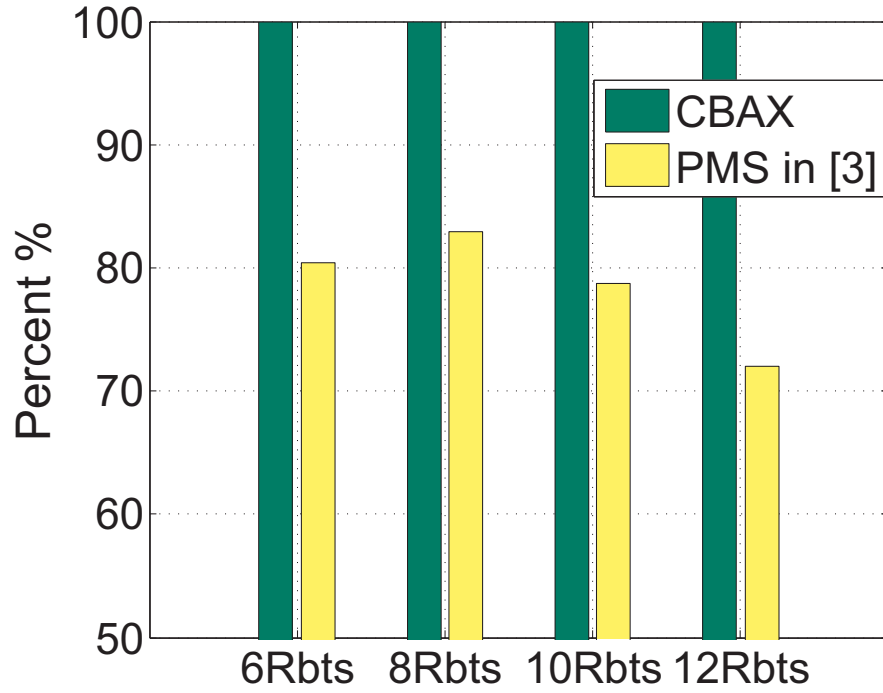
Figure 3.7. Environments and exploration time.

³We define a disconnected link has link capacity 0 bit/sec.

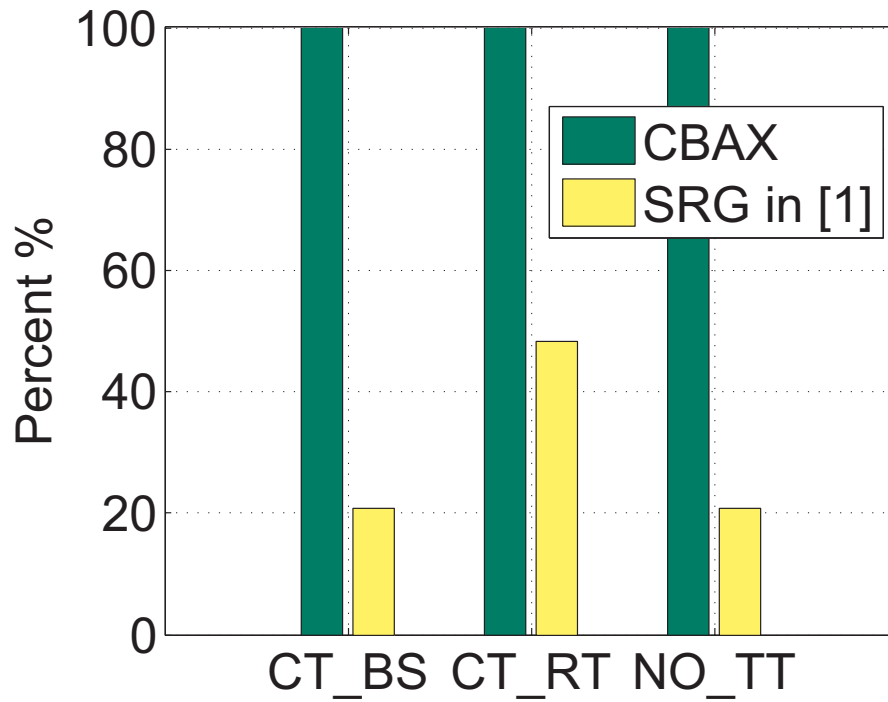
Discussion: A mobile multi-hop wireless network's quality of service (QoS) is a complex issue affected by various factors such as underlying MAC protocol, interference, channel quality, scheduling, congestion control, and routing path selection. Besides, the contention based CSMA/CA MAC protocol in 802.11 may not guarantee an absolute link rate and throughput. However, topology and node movement significantly influence the performance of the network protocol in mobile ad hoc networks [101] and with our QoS-aware mobility model, we provide the most fundamental assurance of bandwidth adequacy. To further enhance rate control, we can employ the Time Division Multiple Access (TDMA) based scheduling to guarantee the required bandwidth for each flow on a certain link.

3.5 Summary

In this chapter, we have proposed a scheme for multi-robot exploration called *Connectivity and Bandwidth Aware eXploration (CBAX)*, which is an efficient iteration based *real-time* exploration. CBAX divides the problem into frontier node placement, relay node placement with routing path selection, and matching of each robot with its target position. Moreover, we model bandwidth-constrained relay node placement into a new variant of the Steiner Minimum Tree problem and present our solution. While reducing the exploration time, CBAX maintains the network's connectivity and ensures the aggregated data flows are under the link capacity in transmission. Simulation shows that CBAX outperforms two recent exploration schemes *qualitatively* by demonstrating major improvement in terms of non-overflow transmission time and fully-connected transmission time. With enhanced communication quality, CBAX still reduces the exploration time, on average, by 40% and 15% respectively. In moderately dense scenarios, CBAX even decreases time by 50%



(a) **NO_TT**: CBAX vs. PMS in [36] with 6,8,10,12 of robots in the *open* environment.



(b) **CT_BS**, **CT_RT**, and **NO_TT**: CBAX vs. SRG in [32] with 4 robots in the *garden* environment.

Figure 3.8. Simulation results on communication quality.

and 25%. In the next chapter, we will consider the heterogeneous communication ranges and data rates along with the impact of interference.

CHAPTER 4

Stream Aggregation in Heterogeneous Range and Rate Mobile Robot Networks

4.1 Preliminaries and Motivations

In the previous chapter, when we cope with the key component of the exploration process, i.e., relay placement in each iteration, we have assumed *uniform link capacity and transmission range* (UCR). (We use *link capacity* to refer to the actual bandwidth or achievable throughput while the *modulation rate* refers to the much higher theoretical data rate that neglects numerous factors such as MAC overhead [102].) We modeled the problem by a variant of the *Steiner Minimum Tree Problem with Minimum Number of Steiner Points (SMT-MSP)* with integer inflow constraints. It assumes: (1) uniform disk models with homogenous link capacity and transmission range; (2) the impact of interference is minimal and is therefore neglected. While such assumptions are common and make the problem be easily analyzed and solved, they are inconsistent with real 802.11 network conditions.

In this chapter, however, we consider two new conditions, or **HCRI**²: (1) *the Heterogeneous link Capacities and transmission Ranges* (HCR); (2) *the Impact of Interference* (I^2). The motivation of considering HCRI² is threefold. First, as multimedia streams demand high bandwidth, it is also important to consider the tradeoff between link capacity and transmission range. There is a fundamental tradeoff between data rate and range as higher modulation rates increase the link capacity (throughput) but decrease the range at which the transmission can be successfully decoded as signal power and channel capacity decrease with distance.

Second, similar to our communication task's strong dependence on mobility, the *tradeoff* between link capacity and transmission range also reflects such dependence if we regard link capacity as the critical communication metric and transmission range variance as the result of movement. Since such tradeoff interestingly has the same property of tight connection of mobility and communication as our task, we would like to explore whether the tradeoff can be leveraged in favor of our task's goal. Third, as interference significantly impacts the relations of transmission ranges with rates and undermines the communication quality, we consider interference as well.

Third, HCRI² reflects the central characteristic of wireless communication in real-world settings although it makes the node placement more complex: The transmitter and receiver's distance and I^2 cause the link capacity to be dynamic. To our best knowledge, no prior work takes advantage of the tradeoff in distance and bandwidth to ensure bandwidth adequacy with controlled mobility.

UCR based relay placement cannot take advantage of HCR. Besides, I^2 is also detrimental to bandwidth adequacy. To solve the problem with HCRI², we propose a multi-channel dual radio network architecture with dynamic TDMA scheduling to minimize interference. A channel assignment scheme, modeled by bandwidth coloring, tackles out-of-band (OOB) emission [103]

in the same node by limiting the minimum channel distance. The details of the multi-channel dual radio network architecture and the channel assignment scheme are presented in Section 4.2 and Section 4.5 respectively.

Similar in CBAX, we also consider the relay placement and routing jointly because routing decides the path on which a stream traverses; and the bandwidth sufficiency depends on both supply (the link capacity) and demand (which streams use the link given the routing paths). We model the stream aggregation problem as the *relay placement and routing for bandwidth sufficiency* ($\mathbf{R}^2\mathbf{BS}$) with HCRI², which can be described as follows: Given a set of stream sending nodes (\mathbb{SN}) and a base station (BS), how to place a minimum number of relay nodes (\mathbb{RN}) in a set of candidate positions and choose the routing paths from each SN to BS, such that (1) all \mathbb{SN} are connected with the BS with the help of relays; (2) the aggregated consumed throughput by the streams is under the link capacity, considering heterogeneous range and rate in transmission. $\mathbf{R}^2\mathbf{BS}$ may be classified as a *single-tiered* and *constrained* relay placement according to [47]. It is single-tiered in that \mathbb{SN} also forward packets. It is constrained because there are forbidden areas where relays cannot be placed.

4.1.1 Key Contributions

- To the best of our knowledge, this work is the first of its kind to jointly consider relay placement and routing to achieve bandwidth sufficiency for multimedia real-time stream aggregation considering the heterogeneous range and rate and impact of interference.
- To solve $\mathbf{R}^2\mathbf{BS}$, we formulate the problem as a new variant of the Steiner tree problem called the heterogeneous bandwidth Steiner routing (HBSR). We prove the problem NP-hard and

present an efficient heuristic with 44% reduced relay number compared to the widely used minimum spanning tree based approximation algorithm for relay placement. We also show the upper bound of the algorithm’s approximation ratio in the Appendix.

- We also found that considering heterogeneous range and rate is beneficial in relay placement. Compared to the uniform range and rate bandwidth-aware placement algorithm, our scheme reduces the number of relays by 25%-39%.
- We propose the “expected relay count” to quantify the flow aggregation’s impact on relay reduction number. With expected relay count, we evaluate different aggregation cases and select the ones with maximum reduction, resulting in significant reduced number of relays.

The rest of the chapter proceeds as follows. Section 4.2 introduces the system model. In Section 4.3, we formulate the problem and discuss the challenges. Section 4.4 presents the main solution of relay placement and routing. Section 4.5 gives the channel assignment. Performance evaluation is in Section 4.6. Section 4.7 gives the summary of the chapter.

4.2 System Model

4.2.1 Environment Model

We apply a 2D occupancy map with grid cells to represent the environment as a rectangular region $\mathbb{R} = X \times Y$. A subset of cells are marked as “obstacles” to be excluded from the candidate positions for the relays.

4.2.2 Network Model

We consider a static wireless network, or a mobile wireless network where nodes are static when streams are in transmission. The network has 3 types of entities: (1) stream sending nodes (\mathbb{SN}), (2) stream relay nodes (\mathbb{RN}), and (3) the base station (BS). Each SN has a uniform *flow sending rate* r_s for its streams. We relax r_s to be non-uniform in the Appendix.

A link with a sender u and a receiver v is denoted by a directed edge $e(u, v) \in E$ in the graph $G(V = \{\mathbb{SN}, \mathbb{RN}, BS\}, E)$. The *edge length* is the physical distance between u and v . Given its edge length, a link e 's *bandwidth* B_e is the maximum allowed link bandwidth mapped to the distance no smaller than its edge length. e 's *link utilization ratio* U_e is e 's carried flow rate divided by its bandwidth. A *routing path*, or $\text{path}(V) = (v_0, v_1, \dots, v_n)$, gives an ordered vector of positions.

Due to the complex impact of interference on link bandwidth, a multi-channel dual-radio with local TDMA scheduling model is adopted to eliminate interference. Note that the effectiveness and validity of the model has been demonstrated by previous testbeds [1, 103–105]. In future work, we will consider the complex interference impact under CSMA/CA.

- We adopt a multi-channel dual-radio network model to eliminate interference. The channel assignment scheme is presented in Section 4.5. Our model uses the 802.11a band because of its (1) high bandwidth, (2) 24+ non-overlapping channels for assignment: original 12 channels and another 12 channels approved in 5.47 to 5.725 GHz Band [106].
- Each node is equipped with two radios with one for transmitting (sending) data while the other for receiving data. A testbed of such multi-channel and dual-radio wireless network is given in [104].
- When multiple nodes send streams to one receiving node, the channel contention is eliminated

by the time division multiple access (TDMA) based scheduling, where the bandwidth is dynamically allocated to the senders. Contention based 802.11 CSMA/CA reduces the achievable bandwidth and also makes bandwidth estimation difficult. A testbed with similar local TDMA scheduling is presented in [105].

- Similar to the 15-radio testbed in [103], the BS has adequate radios to support streams aggregated from \mathbb{SN} .
- We adopt a coarse-grained piecewise mapping function between distance and link bandwidth based on the modulation rate, because a precise mapping is often inaccurate in practice due to multi-path fading and attenuation variation.

The BS is able to collect the current \mathbb{SN} positions and the obstacle distribution in the environment, which enables our centralized solution to R^2BS . Similar to previous centralized relay placement schemes [43, 47, 49, 51–53], such an approach facilitates graph theoretic modeling. It also avoids the local data exchange and local suboptimal results in a distributed manner. Besides, it fits the multi-robot search and rescue applications of our problem in mobile networks for video aggregation: the BS will remotely control robots and collect position and obstacle information anyway [55].

4.2.3 Mapping of Range and Rate

We present a coarse-grained piecewise mapping function between distance and link bandwidth (capacity) based on the modulation rate, given that (1) the TDMA scheduling eliminates interference between multiple senders to the same receiver; (2) the multi-channel and dual-radio framework eliminates the interference between different links.

A simple measurement in the targeted placement environment can tell the minimum link bandwidth and the maximum range for each modulation rate. The link bandwidth and the range should sustain at least 85% (a safety ratio) of the total measurement time. In a more dynamic environment, this ratio is set higher to be more conservative. Due to limited space we leave the analysis on its optimal value in future work.

We adopt the measured data in Table 4.1 from a real testbed measurement conducted by Atheros researchers in [1] with Atheros 802.11a in an office environment. Because the data describes the averaged performance, we also conservatively adjusted the safe ranges as 85% of the original ones. Different settings and hardware require remeasurement, which only affects the data in Table I but not the proposed algorithm.

Hence, given a link $e(u, v)$ with sender u and receiver v , the function $\omega(B) = R$ maps link bandwidth from range while $\omega^{-1}(R) = B$ gives the inverse mapping. Additionally, \mathbb{B} and \mathbb{R} denote the *link bandwidth (capacity) vector* and *range vector* in the table respectively.

<i>Modulation Rate,</i> $r_m(\text{Mbps})$	<i>Minimum Link Bandwidth</i> <i>(Capacity), $B(\text{Mbps})$</i>	<i>Safe Transmission Range,</i> $R(\text{ft}), \alpha = 0.85$
6	5	191
12	6	148
18	10	115
24	17	75
36	18	72
48	23	24
54	25	20

Table 4.1. Modulation rate vs. link bandwidth (capacity) vs. transmission range in 802.11a [1].

4.2.4 Discussions

The above measurement should be conducted in the target environment; In this paper, the obstacles are the ground level ones mainly serving as the forbidden areas for relay placement, therefore having little impact on the mapping so that the mapping is effective in the relay placement. In future work, we would like to cope with the impact of different obstacles to this mapping with two proposed solutions. The first one is based on a recent work [107] which uses pure measurement to predict effectively the best modulation rate and achievable link capacity in their testbed. To tackle the frequency-selective fading due to multipath, the authors in [107] employed the subcarrier channel state information (CSI) to compute the total bit error rate (BER). Here, we can measure the difference of CSI with and without a typical obstacle, e.g. a wall. Because CSI can predict the link performance and the obstacle distribution is known, we can adjust the new mapping with the obstacle's impact before the relay placement. A similar idea of modeling obstacle effect to predict signal attenuation is presented in [108].

The second proposed solution is to adjust the relay positions after placement if the measured link capacity is deviated from the mapping. This solution is more appropriate when relays are mobile devices such as mobile sensors or robots. When mobile relays move towards the computed initial relay positions, they can keep sending measurement packets as they move and collect CSI to maintain a database for link capacity with positions along the paths. Relays can update the target positions to adjust for radio irregularity and other dynamics.

4.3 Problem formulation

Given the system model, we model R^2BS by a new variant of the Steiner tree problem called *heterogeneous bandwidth Steiner routing* problem (**HBSR**) with minimum Steiner points. HBSR can be described as follows. Given

- a region \mathbb{RG} with an obstacle-cell subset;
- a finite set of n terminal points including one base terminal (BS) and $n - 1$ sending terminals (SN) with a uniform flow sending rate r_s to the BS;
- mappings of allowed flow sizes (link bandwidth) inversely proportionate to edge lengths;
- bandwidth constraint: the sum of link utilization ratios of all incoming edges at each SN or RN is less than one;
- placement constraint: no relay is placed on obstacles;

to find a directed Steiner tree $T(V_T = \{SN \cup RN \cup BS\}, E_T)$ that uses Steiner points (RN) to connect all sending terminals to the BS with a valid routing path from each SN to the BS, such that the number of Steiner points in T is minimized. Mathematically, HBSR is

$$\text{Minimize } |\mathbb{RN}|, \quad (4.1)$$

$$\text{S.T. } \exists \text{ path}(V), V = (i, v_1, \dots, v_n, BS), \forall i \in \mathbb{SN}, v_i \in \mathbb{N}, \quad (4.2)$$

$$\text{dist}(v_j, v_{j+1}) \leq \max\{R_i\}, \forall v_j \in V, j \leq n, \quad (4.3)$$

$$B_e = \omega^{-1}(\min\{R_i\}), R_i \geq \text{dist}(e), \quad (4.4)$$

$$U_e = \sum_f r_s(f)/B_e, \forall \text{ flow } f \text{ on } e, \quad (4.5)$$

$$\sum_{e(u,v)} U_e \leq 1, \forall v^* \in \mathbb{N}, \forall e \in E_T \wedge v = v^*, \quad (4.6)$$

$$\text{position}(i) \neq \text{obstacle}, \forall i \in \mathbb{RN}, \quad (4.7)$$

$$\forall R_i \in \mathbb{R}, \quad \forall e \in E_T, \quad (4.8)$$

Where $\mathbb{N} = \mathbb{SN} \cup \mathbb{RN}$. Equations (2) and (3) show the a valid routing path from each SN to the BS with relays under the communication range. Equations (4)-(6) give the bandwidth constraint. Equation (7) provides constraints so that the relays cannot be placed at the obstacle cells.

4.3.1 Challenges

We first prove that the HBSR problem is *NP-hard*.

Theorem 2. *The HBSR problem is NP-hard.*

Proof. We prove the theory by contradiction. Suppose we have a polynomial solution to this problem, by relaxing the constraint by giving an infinite link capacity and a single transmission range, the problem becomes the *Steiner minimum tree Problem with minimum number of Steiner points*

(SMT-MSP) [43], which will then be polynomially-solvable by our solution. This contradicts the fact that SMT-MSP is NP-hard [43], and this completes the proof. \square

Another challenge is to solve the problem that the heterogeneous range and rate adds to the relay placement’s complexity and substantially increases the solution space for possible relay positions. We need to determine whether we should place relays on \mathcal{SN} ’s direct paths to the BS or select some other positions to let \mathcal{SN} share relays with aggregated flows. With uniform range and rate, the relay placement in [52] greedily aggregates flows at the earliest possible time, enabling multiple senders to share relays to reduce the relay number.

Such a method is no longer feasible due to the following two reasons. First, with only one radio for incoming traffic, flow aggregation can cause disconnections by reduced transmission ranges resulting from the increased link capacity requirement. Second, the flow aggregation’s effect on the number of relays becomes complex: Aggregated flows demand more bandwidth, which reduces the effective transmission range and leads to a denser relay placement that compromises its benefit. Hence, flow aggregation does not necessarily reduce the relay number and it is important to evaluate its impact.

4.4 Relay Placement and Routing

4.4.1 General Procedure

We present an efficient polynomial-time heuristic that places relays *layer by layer* such as the breadth-first search (BFS) from \mathcal{SN} to iteratively approach the BS. First, we obtain the first layer nodes from \mathcal{SN} with the flow aggregations expected to reduce the number of relays (refer to Sec-

tion 4.4.3).

Second, with the first layer nodes as the *current layer*, we call Alg. 10 for placing the *new layer* nodes to approach the BS (refer to Section 4.4.4). Similar to BFS, we only place the single-hop relays for *all* current layer nodes to be the next layer, rather than placing multiple relays for a *single* node. The new layer nodes not in range of the BS will become the next round current layer. We continue the loop of: (1) calling Alg. 10 to generate the next layer; and (2) using the new layer to replace the current layer. The loop breaks when the current layer becomes empty. In addition, routing paths are generated when the flows are aggregated or forwarded to the next layer.

4.4.2 Evaluating Flow Aggregation's Impact on Relay Number

Basic Idea: Rather than greedily aggregating flows, we now evaluate whether aggregation is expected to reduce the relay number under the bandwidth constraints. We conduct aggregation only when the aggregation is expected to reduce relay number. We first give the following three definitions.

Definition 5.1: The *bandwidth-sufficient (communication) range* to support k flows $R(k)$ is the maximum range mapping to the (minimum) link bandwidth more than $k \cdot r_s$:

$$R(k) = \omega(\min\{B_i\}), \quad \forall B_i \in \mathbb{B} \wedge B_i \geq k \cdot r_s \quad (4.9)$$

Note that Eq. 4 and Eq. 4.9 are not in a loop of computing range and rate. Eq. 4 is used to compute the expected bandwidth when an edge length is known. Eq. 4.9 is used to compute the expected edge length when its carried flow rate is known.

Definition 5.2: We propose a metric called *expected relay count* (denoted by X). $X(v, k)$ predicts the relay number from v to BS when carrying k number of flows. In an obstacle free or

sparse region, an approximate prediction is:

$$X(v, k) = \lfloor \frac{d(v, BS)}{R(k)} \rfloor \quad (4.10)$$

We can obtain a more accurate $X(v, k)$ by using A* search to find the shortest obstacle-aware path and use its length to replace $d(v, BS)$.

Definition 5.3: We define a *star* (denoted by $S_i(i, L_i)$), or a star subgraph structure, to serve as the unit for flow aggregation. A star graph $S_i(i, L_i)$ is a complete bipartite graph $K_{1, |L_i|}$, or a tree with one center vertex i and the remaining vertices to be the leaves L_i .

We define that the star *leaves* are the stream senders and the star *center* is the stream receiver or the aggregation destination in the star. Star $S_i = (i, L_i)$ has $|L_i|$ number of *leaves* sending flow to center i , the aggregation point.

Recall in the system model that TDMA scheduling eliminates the interference inside a star when leaves share a channel in transmission. We say two stars S_i and S_j are *compatible*, iff they have distinct leaves and centers. The channel assignment scheme ensures no interference between stars.

Definition 5.4: Given the star subgraphs and expected relay count, we propose the *reduction index* (denoted by $\delta(S)$) to be the expected reduced number of relays per leaf by a star aggregation. Namely, to compute $(\alpha - \beta)/|L(i)|$ in S_i :

- α : Expected number of relays used when relays are placed on individual path to the BS without aggregation: sum of the star leaves expected relay count.
- β : Expected number of relays used when relays are placed on the aggregation path: potential “denser” placement due to increased carried flow size.

An aggregation on star S is *beneficial* when it is expected to reduce the relay number ($\delta(S) > 0$).

A precise description of δ computation is given in the following sections. Besides, an illustration of the star and δ is given in Fig. 4.1.

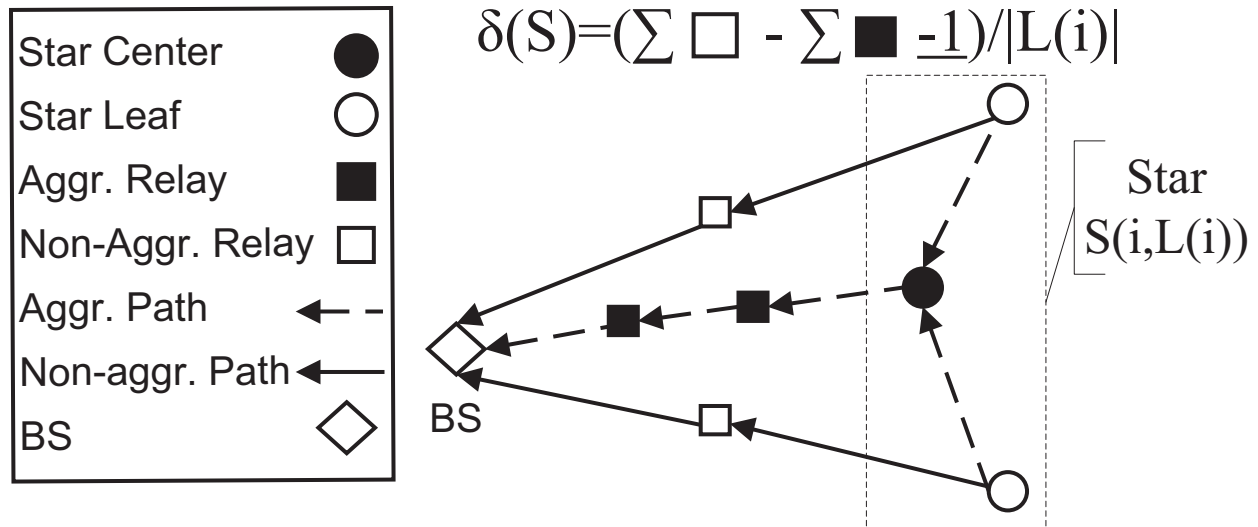


Figure 4.1. Star and the reduction index illustration. In $\delta(S)$, the underlined “-1” will be applied when the center is an extra relay we need to place: in placing the new layer node. It will not be applied when the center is an existing node: in aggregation in \mathbb{SN} .

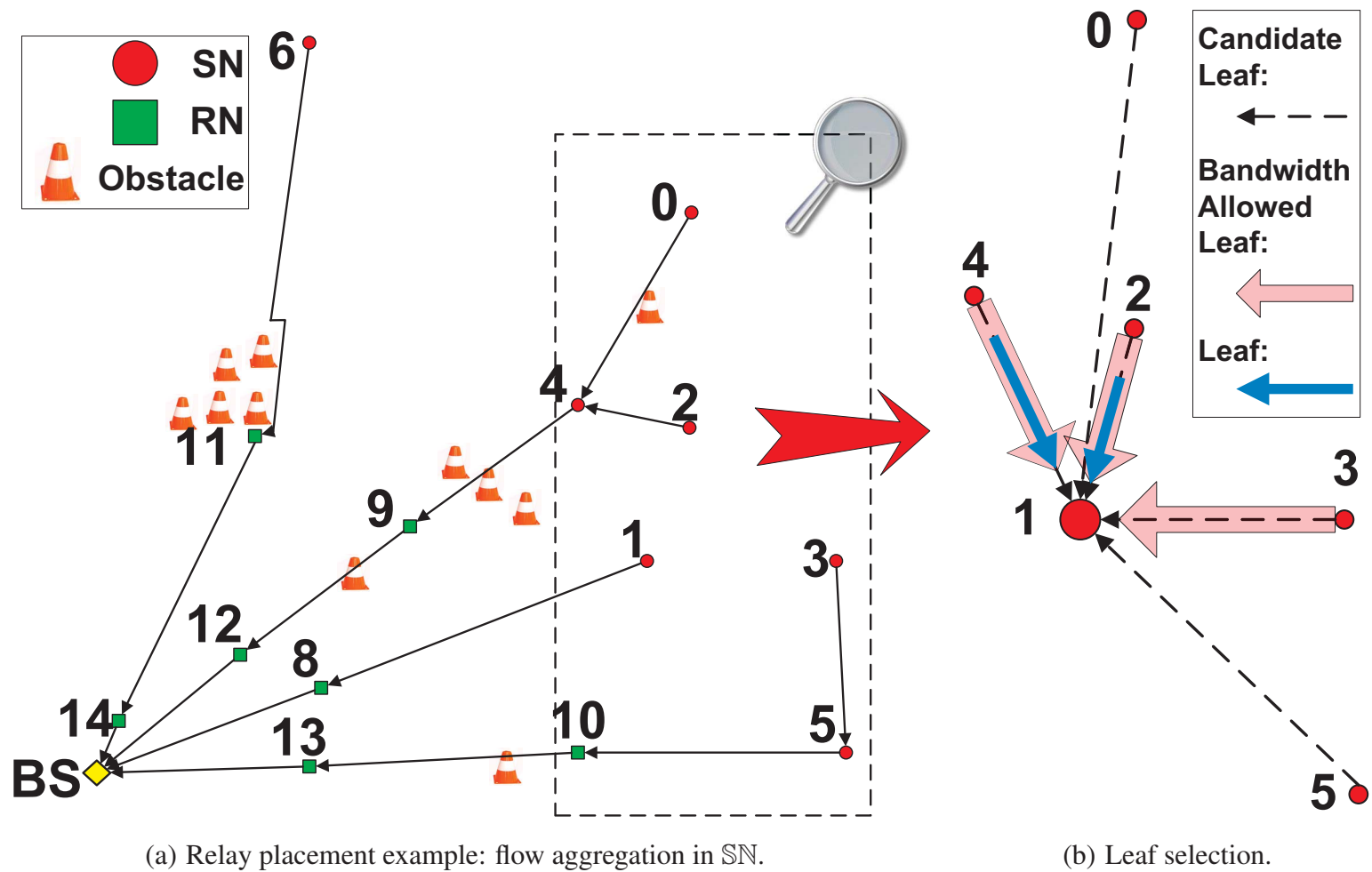


Figure 4.2. Illustration of relay node placement. (Uniform sending rate $r_s=3\text{Mbps}$)

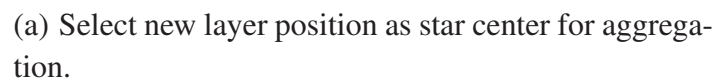


Figure 4.3. Illustration of relay node placement (Continued). (Uniform sending rate $r_s=3\text{Mbps}$)

<i>Terms</i>	<i>Definitions</i>
HBSR	The problem of heterogeneous bandwidth Steiner routing.
\mathbb{SN}, \mathbb{SN}	Sending terminals (stream sending nodes).
\mathbb{RN}, \mathbb{RN}	Steiner points (relay nodes).
r_s	A uniform single flow sending rate at each SN.
B_e	Achievable bandwidth based on distance of link e .
U_e	e 's link utilization ratio: its carried flow rate divided by its bandwidth.
$R(k)$	Bandwidth-sufficient range for k number of flows.
$\mathbb{R}, \mathbb{B}, \omega$	link bandwidth, communication range vectors and their mapping function from Table I.
$X(v, k)$	Expected relay count for v with k flows.
$S(i, L(i))$	A star with center i and leaves $L(i)$. Leaves are the stream senders and the center is the stream receiver.
$\delta(S)$	(Relay) reduction index by a star S .
$\mathbb{S}_{sn}, \mathbb{S}_c, \mathbb{S}$	The star vector in \mathbb{SN} , the candidate star vector, the final compatible star vector for channel assignment.

Table 4.2. Notations in HBSR.

4.4.3 Flow Aggregation in Sending Terminals \mathbb{SN}

Algorithm 9: Selecting leaves for a node i with maximum reduction index.

- 1 **Input:** Node i . **Output:** Leaves $L(i)$ with reduction index $\delta(S(i, L(i)))$.
 - 2 Form an i 's *candidate leaf set* by node j in current layer satisfying: (1) farther from BS than i . (2) j is in the bandwidth-sufficient range of i .
 - 3 Compute bandwidth B_e for each link whose sender is in the candidate leaf set and the receiver is i using Eq. 4.
 - 4 Sort the candidate leaf set by distance to i in ascending order.
 - 5 Add each node $j \in i$'s candidate leaf set to a *bandwidth-allowed leaf set* $BL(i)$ in order, until the sum of e_{ji} 's link utilization ratio is greater than 1: Eq. 6 is violated. Each time when a new node is added: update reduction index by computing $\delta(S(i, BL(i)))$ as in Eq. 4.11.
 - 6 i 's leaves L_i is set as the BL_i when it has the maximum reduction index.
-

Main Procedures

- For each $i \in \mathbb{SN}$ not in 1-hop bandwidth-sufficient range of the BS, we form a star with center i , select its leaves and compute the reduction index δ by calling Alg. 9 to evaluate how much “benefit” we can obtain if using i as the aggregation point to aggregate flows with different leaves.
- Given all the possible star candidates, we have a loop of selecting the star with the maximum reduction index and conduct aggregation until the reduction index of the star is negative. (With non-negative reduction index, aggregation never increases the number of relays).
- The remaining nodes not from the selected stars, along with selected star centers, serve as the

first layer node. They will be the input of Alg. 10 for the next step. All selected stars are inserted into a star vector in \mathbb{SN} , or \mathbb{S}_{sn} .

- For each selected star with center i , we also transfer the flows from leaves to i and insert i to the routing path (relay path) for each leaf.

In Alg 9, the reduction index $\delta(S(i, BL_i))$ is computed by:

$$\delta(S(i, BL_i)) = \frac{\sum_{j \in BL_i \cup i} X(j, 1) - X(i, |BL_i| + 1)}{|BL_i|} \quad (4.11)$$

Example of Leaf Selection: We illustrate reduction index computation with Fig. 4.2(b). For n_1 , its candidate leaf set has 5 nodes: $\{n_4, n_0, n_2, n_3, n_5\}$, which are farther from the BS than n_1 is. By greedily selecting nodes from the candidate set in ascending order to the BS, only three nodes are allowed before the capacity is full. Besides, we discover that δ reaches its maximum as 0.5 when only the first two leaves are used. Therefore, L_{n_1} 's final leaf set has two nodes: n_2 and n_4 .

Example of Aggregation: We illustrate the flow aggregation in \mathbb{SN} in Fig. 4.2(a). Each node computes leaves and the reduction index δ . The node with maximum δ node n_5 is chosen ($\delta_{n_5} = 2$) to be aggregated with flow from n_3 . With n_5 and n_3 eliminated, δ is recomputed for the rest nodes. Node n_4 with maximum $\delta_{n_4} = 1$ is selected as the destination for flows from n_0 and n_2 . The aggregation terminates as the rest of nodes n_1 and n_6 have maximum $\delta < 0$.

4.4.4 New Layer Relay Placement with Flow Aggregation

Overview: Given the first layer node as the current layer, the next layer nodes are placed to approach the BS. A new layer node is placed at the aggregation point to approach the BS if the flow aggregation is expected to reduce relays. Otherwise it is placed to be closer to the BS by its

bandwidth-sufficient range, on the direct path from the current layer node to the BS. The major steps are in Alg. 10. Alg. 10 calls Alg. 11 to compute the next layer candidate relay positions. They are the potential aggregation destinations given the current node pairs.

Select Compatible Stars with Maximum Sum of Reduction Index: In Alg. 10, given the candidate star vector \mathbb{S}_c computed by calling Alg. 11, we need to select a subset of stars with the maximum sum of star's reduction index to form a final star vector \mathbb{S} . A new layer relay will be placed in the center at each star in \mathbb{S} . The stars in \mathbb{S} must be *compatible* because they cannot share leaves or centers since a flow can only be aggregated to one destination under the bandwidth constraints.

The problem is modeled as the NP-hard [109] *maximum weighted set packing* problem, which is defined as follows: Given a finite set and a list of subset, find the disjoint subsets with maximum sum of weight. In our problem, the finite set is the set of all the leaves of stars in \mathbb{S}_c . Each star is a

subset including its two leaves, and the star's reduction index is the weight of the subset.

Algorithm 10: New layer relay placement with flow aggregation.

- 1 **Input:** Current layer nodes. **Output:** New layer nodes; routing paths.
 - 2 Form a *candidate star vector* \mathbb{S}_c by a hypothetical placement of center vertex as destination for each pair of nodes in current layer. Call Alg. 11 $\binom{k}{2}$ times (k is the number of nodes in current layer). Use each node pair to be Alg. 11's input.
 - 3 Select a final star vector \mathbb{S} with the maximum sum of reduction index from \mathbb{S}_c : Iterate through \mathbb{S}_c 's powerset, filter out the incompatible subsets, and obtain the *compatible* subset with maximum of the sum of reduction index. Denote the resulting star vector \mathbb{S} .
 - 4 New layer is set by the following two types of nodes: (1) Centers in \mathbb{S} ; (2) New nodes placed for non-aggregated flows for rest of the current layer nodes at bandwidth-sufficient range on the obstacle-aware paths to the BS by A* search.
 - 5 Flows and routing paths are updated similar to the procedure in Section 4.4.3.
 - 6 Links with non-aggregated flows (a star with 1 center and 1 leaf) are also inserted into \mathbb{S} .
Besides, \mathbb{S}_{sn} is appended to \mathbb{S} .
-

Algorithm 11: Choose new layer node position for aggregation.

- 1 **Input:** Leaf positions m, n . **Output:** A candidate star S with center position p and reduction index $\delta(S(p, L(p)))$, where $L(p) = \{m, n\}$.
 - 2 Obtain range vectors $\mathbb{R}_m \subseteq \mathbb{R}, \mathbb{R}_n \subseteq \mathbb{R}$ constrained by their bandwidth requirement for the carried flow sizes.
 - 3 Compute Bandwidth B_e for links with length equal to each range $R \in \mathbb{R}_m \cup \mathbb{R}_n$ by Eq. 4.
 - 4 Test each pair $\langle i, j \rangle \forall i \in \mathbb{R}_m, j \in \mathbb{R}_n$. When (1) total flows do not exceed the shared bandwidth: Eq. 6 is not violated; and (2) valid overlapping area exists for range disks m and n : The closest position p' to the BS in the overlapping area is saved as a candidate star center for this pair.
 - 5 Final center p is the closest position to the BS among all candidate centers. Form $S(p, L_p)$ with leaves $L_p = \{m, n\}$.
 - 6 Compute reduction index by Eq. 4.12. Add $S(p, L_p)$ to the candidate star vector \mathbb{S}_c if $\delta(S(p, L_p)) > 0$ (Not beneficial stars cannot be candidates: aggregation never increases the number of relays).
-

Due to the NP-hard nature of this problem, we compute the powerset to try all possible cases to obtain the optimal result when the input size is less than a threshold k : $|\mathbb{S}_c| \leq k$. (The powerset of a set is the set of all its subsets.) If $|\mathbb{S}_c| > k$, we iteratively greedily select stars with the maximum reduction index to form \mathbb{S} . In the simulation process, we set $k = 12$ and find that all our test cases have $|\mathbb{S}_c| < k$ and the results are computed efficiently.

Choose the New Layer Node Position for Aggregation: In Alg 11, the reduction index

$\delta(S(i, L_i))$ is computed by:

$$\delta(S(i, L_i)) = \frac{\sum_{j \in L_i} X(j, |f_j|) - 1 - X(i, \sum_{j \in L_i} |f_j|)}{|L_i|} \quad (4.12)$$

where f_j is the flow size carried by j . Note that the flexible position of the star center is the main difference compared to the aggregation in SN. The star center p is generally the closest position to the BS in the overlapping area of range disks of i, j . Note that it is possible all range combinations fail to provide bandwidth adequacy after aggregation. In addition, the formed star S is added to \mathbb{S}_c only when beneficial: $\delta(S) > 0$.

Example of Choosing New Layer Node Position for Aggregation: Fig. 4.3(a) illustrates Alg. 11. Given a pair of current layer nodes m, n with $f_m=8\text{Mbps}$, $f_n=2\text{Mbps}$, the range sets $\mathbb{R}_m, \mathbb{R}_n$ are known. Each pair of ranges for m, n are tested until we find the final position p . Note that p is not placed in m 's out-most disk because the maximum ranges for $R_{max}^m = 115\text{ft}$, $R_{max}^n = 191\text{ft}$ cannot be used together: They exceed the bandwidth with sum of link utilization ratio $\sum U = 8/10 + 2/5 = 1.2 > 1.0$.

Example of Aggregation: Fig. 4.3(b) illustrates placing new layer relays with flow aggregation. While n_0 and n_4 are too far apart to be aggregated, n_7 and n_8 are in range to aggregate each of its flow 3Mbps to n_9 with 6Mbps. In addition, n_1 's 6Mbps flow, including that from n_2 , is aggregated with n_3 's 3Mbps flow to n_6 .

Example of Relay Density Variance: Both Fig.4.2(a) 4.3(b) show relay density variation among paths with different flow sizes. In Fig. 4.3(b), more relays are placed on path from n_6 than that from n_9 to the BS. Likewise, relays are placed more sparsely for a single flow in path from n_6 to the BS compared with other paths in Fig. 4.2(a). To sum up, flows are aggregated when they are expected to reduce relay number in placement.

4.4.5 Upper Bound of Approximation Ratio

We derive the upper bound of the approximation ratio of our solution to HBSR. First we define some notations. R gives the transmission range corresponding to a single flow rate r_s from each SN. Furthermore, B_m represents the maximum link bandwidth $B_m = \max_{B_i \in \mathbb{B}} \{B_i\}$, i.e., $B_m = 25\text{Mbps}$ according to Table 4.1. A maximum degree in a graph G is: $\Delta(G) = \max\{\deg_G(v) | v \in V(G)\}$. A degree of a vertex v is the times v is used as an end vertex of an edge in the graph.

We denote n , n_s , n_d , and n_{opt} are the respective relay number obtained by: (1) algorithm for HBSR; (2) S-MST: the Steinerized minimum spanning tree [91] with a uniform range R *not* considering the flow constraint; (3) DT: direct tree, or the simple aggregation tree (star graph with center BS) where all flows from \mathbb{SN} are directly relayed to BS without aggregation; and (4) T_{opt} : An optimal Steiner tree with flow constraint and heterogeneous ranges that requires minimum number of relays. All four schemes except (2) satisfy our problem statement.

Besides, we use MST to denote the minimum spanning tree for \mathbb{SN} and BS. Note that DT and MST are two very simple tree structures easily obtained when given the input of \mathbb{SN} and BS. We define $\kappa = n_d/n_s = \sum_{e \in E(DT)} \lfloor e/R \rfloor / \sum_{e \in E(MST)} \lfloor e/R \rfloor$, where κ is the ratio of relays size for DT over that of MST with uniform range R . Thus, κ is relevant to \mathbb{SN} and can be directly obtained from the input of HBSR.

Theorem 3. *The approximation ratio of our relay placement algorithm is no greater than $\kappa \cdot (1 + B_m/r_s)$. In an environment where obstacle distribution does not affect the triangular inequality, the ratio is no greater than $\kappa \cdot \min\{5, 1 + B_m/r_s\}$.*

Proof. With an optimal tree T_{opt} , we duplicate each edge and build an Eulerian tour such that every Steiner point appears at most $\Delta(T_{opt})$ times in the tour while every terminal point appears

exactly once. Removing one (or more) edge in the tour will give a spanning tree T' where Steiner points have degree at most two. Its Steiner point size $n(T')$ is no more than $\Delta(T_{opt})$ times in T_{opt} . Because n_s is from S-MST: a *minimum* spanning tree with relays on uniform non-shortened edges for single flows. Clearly, we have:

$$n_s \leq n(T') \leq \Delta(T_{opt}) \cdot n_{opt} \quad (4.13)$$

With $\kappa = n_d/n_s$ and the fact that flow aggregation guarantees $n \leq n_d$, we have:

$$n \leq n_d \leq \kappa \cdot \Delta(T_{opt}) \cdot n_{opt} \quad (4.14)$$

$$n/n_{opt} \leq \kappa \cdot \Delta(T_{opt}) \quad (4.15)$$

Lemma 3. *Every Steiner tree that satisfies the flow constraint must have its maximum degree $\Delta(T) \leq 1 + B_m/r_s$.*

Proof. For a vertex, its incoming degree is the same as the number of incoming flows for aggregation, which is constrained by the maximum link bandwidth B_m . As we allow only one outgoing link, the total degree is at most $\Delta(T) \leq 1 + B_m/r_s$. \square

Lemma 4. *There exists a shortest optimal Steiner tree T_{opt} whose maximum degree is less than or equal to five, i.e., $\Delta(T_{opt}) \leq 5$, in an obstacle-free area or obstructive area where triangular inequality holds [91].*

From Lemma 3 and 4, $\Delta(T_{opt}) = \min\{5, 1 + B_m/r_s\}$ in obstacle free area or obstructive area where triangular inequality holds. Otherwise: $\Delta(T_{opt}) = 1 + B_m/r_s$. Replacing $\Delta(T_{opt})$ concludes the proof. \square

4.4.6 Extension

Theorem 4. *Our solution to HBSR is extensible to deal with non-uniform flow sending rates r_s from \mathbb{SN} .*

Proof. In the current solution, the flow aggregation in \mathbb{SN} assumes each flow sending rate is uniform, while the new layer node placement already assumes different flow sizes after possible aggregations in \mathbb{SN} . The major changes to support non-uniform sending rates r_s are as follows. (1) Change from the number of carried flows to the sum of all flow rates on a link in Eq. 4.11 and 4.12. (2) The expected relay count needs to be updated with a new bandwidth sufficient range in Eq. 4.9 mapping to the minimum link bandwidth more than $\sum r_s$ rather than $k \cdot r_s$. (3) The sorting in step 3 in Alg. 9 is now based on the ascending order of link utilization ratio rather than the distance to BS. These changes enable the reduction index computation with heterogeneous sending rates and this completes the proof. □

4.5 Channel Assignment

Algorithm 12: Channel assignment.

- 1 **Input:** Star vector \mathbb{S} . **Output:** A channel assigned for each $S \in \mathbb{S}$.
 - 2 Construct star reduced graph (SRG) $G_{srg}(V, E)$ from \mathbb{S} .
 - 3 Construct link contention graph (LCG) $G_{lcg}(V, E) \leftarrow G_{srg}(L(V), L(E))$.
 - 4 Compute channel distances for each e and update E in G_{lcg} .
 - 5 Vertices $V_{tmp} \leftarrow V$ in G_{lcg} .
 - 6 **while** V_{tmp} is not empty **do**
 - 7 $i^* \leftarrow \operatorname{argmax}_{i \in V_{tmp}} \sum_{j \in V_{tmp} \wedge \exists e_{ij}} \text{weight}_{e_{ij}}$.
 - 8 Assign least unused channel ID to S_{i^*} . Remove i^* from V_{tmp} .
-

Although relays are placed with selected flow aggregation and varying distances according to different flow sizes, bandwidth adequacy cannot be satisfied without proper channel assignment to eliminate the inter-star interference. We model the channel assignment problem as the NP-hard *bandwidth coloring problem* [110] for a minimum total number of channels. Here, a channel represents the color. Bandwidth coloring problem generalizes the classical vertex coloring problem with the edge weight constraints on color distances of adjacent vertices. Such modeling enables us to tackle the problem of out-of-band (OOB) emission (or the adjacent channel interference with power leakage when radios are placed very close (less than 1m) [103]). To overcome this problem, we double the channel distance for the two radios at the same node, which is modeled by the doubled weight. A greedy algorithm is designed to choose the vertex with the maximum sum of adjacent edge weights (Detailed procedures are in Alg. 12).

4.6 Performance Evaluation

Summary

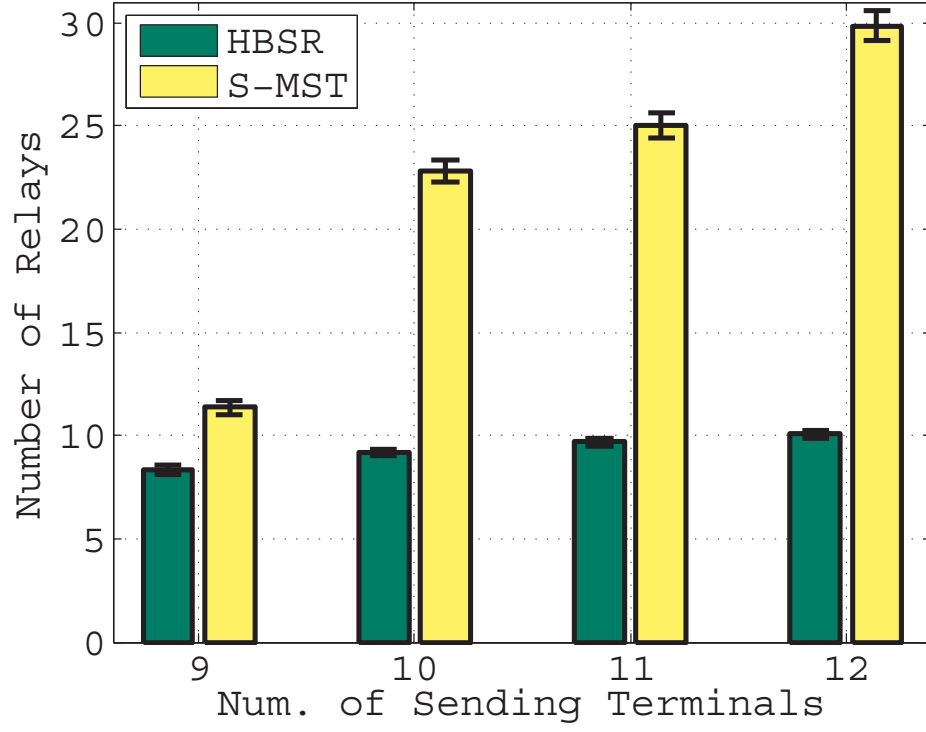
- To evaluate the quality of the proposed method (for simplicity, we call our method HBSR), we compare HBSR to the widely used minimum spanning tree based approximation for relay placement.
- To evaluate the benefit of considering heterogeneous range and rate, we compare HBSR to the uniform range and rate based relay placement.
- In addition, we also apply HBSR to a coordinated mobility model where fewer relays can enable more robots to conduct search and thus increase the mobility efficiency. The mobility model is a multi-robot area exploration strategy with video aggregation [52].

We use a C++ simulation program with the *Open* and *Garden* environments as in Fig. 4.5(b), where obstacles are denoted by black cells. Both environments are represented by 150×150 grid cells with cell edge length of 6ft.

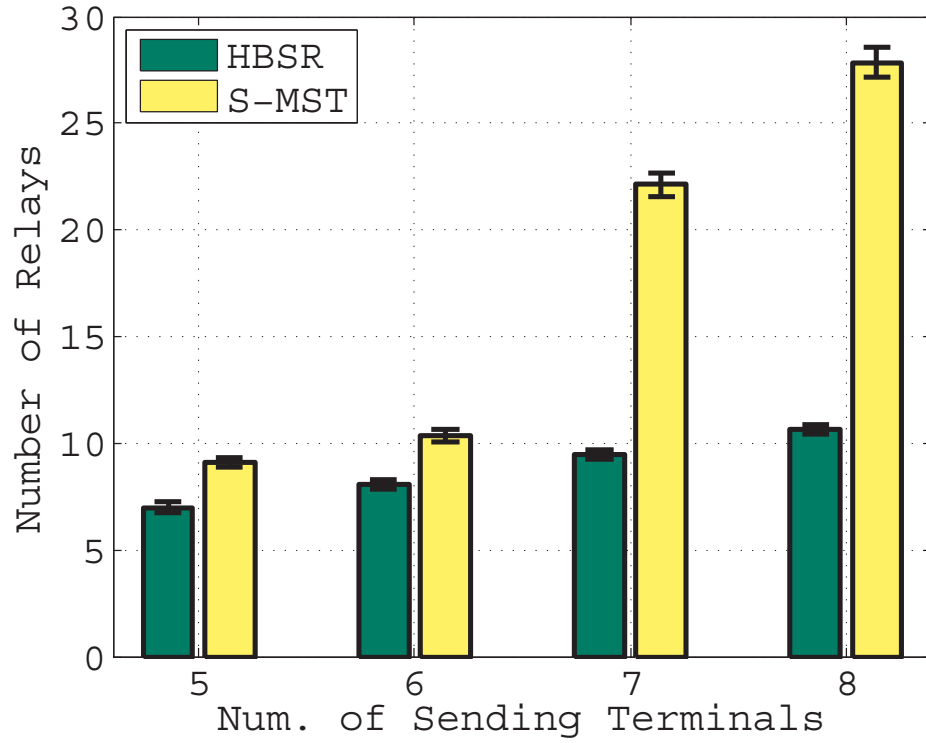
Due to the exponential nature of HBSR, and the highly complex linear programming formulation for heterogeneous ranges and rates, we did not directly compare the results with the optimal or its lower bound. Instead we provided the upper bound of the approximation ratio to the optimal. We will carry out the direct comparison to optimal in future work.

4.6.1 Comparing to Minimum Spanning Tree based Relay Placement

First, we compare the required relay number to a heterogeneous range and data placement scheme satisfying the bandwidth requirement. It is built upon a widely used approximation algorithm called the *Steinerized Minimum Spanning Tree* [91] (for simplicity we call this scheme **S-MST**)

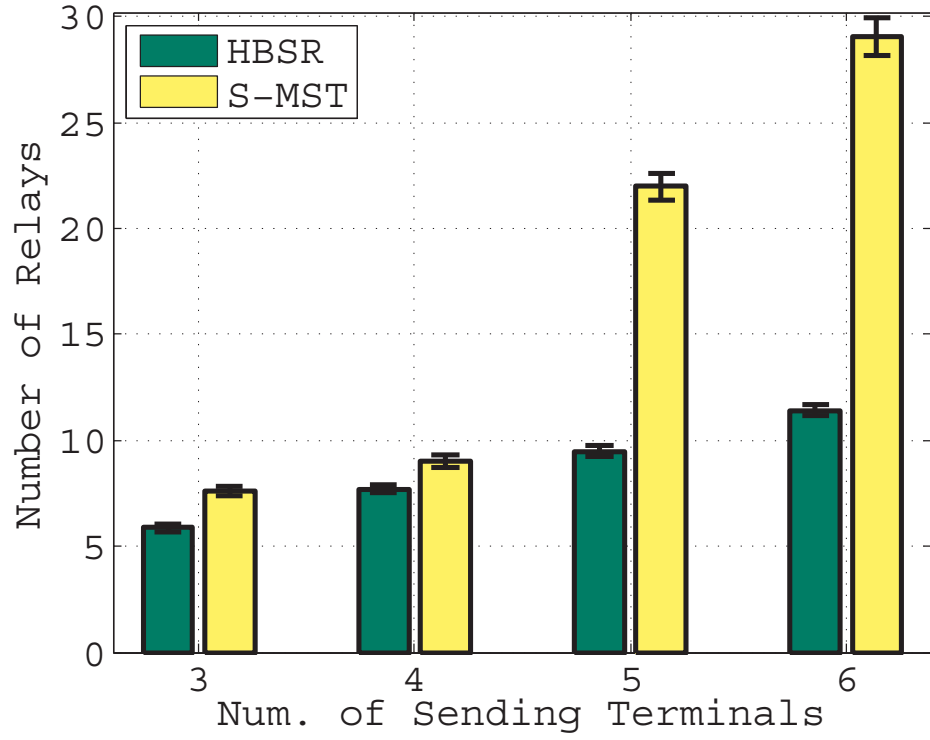


(a) S-MST fails when $|\mathcal{S}\mathcal{N}| > 12$. $r_s=2\text{Mbps}$.

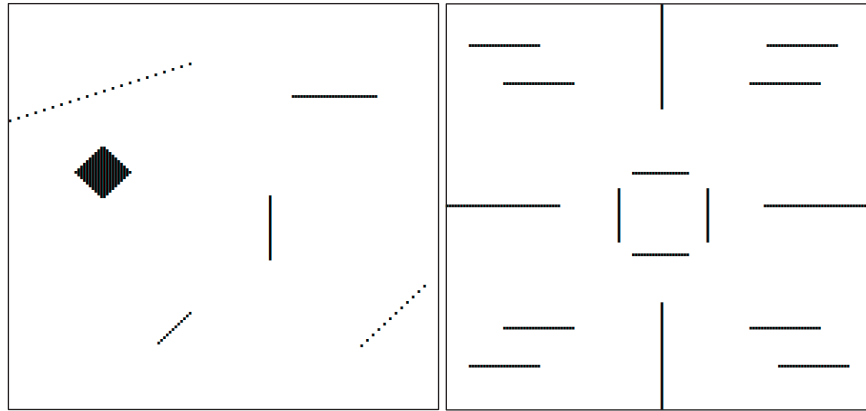


(b) S-MST fails when $|\mathcal{S}\mathcal{N}| > 8$. $r_s=3\text{Mbps}$.

Figure 4.4. (a)-(b): Compare relay number to bandwidth aware S-MST: Varied flow sending rate r_s of 2-3Mbps.

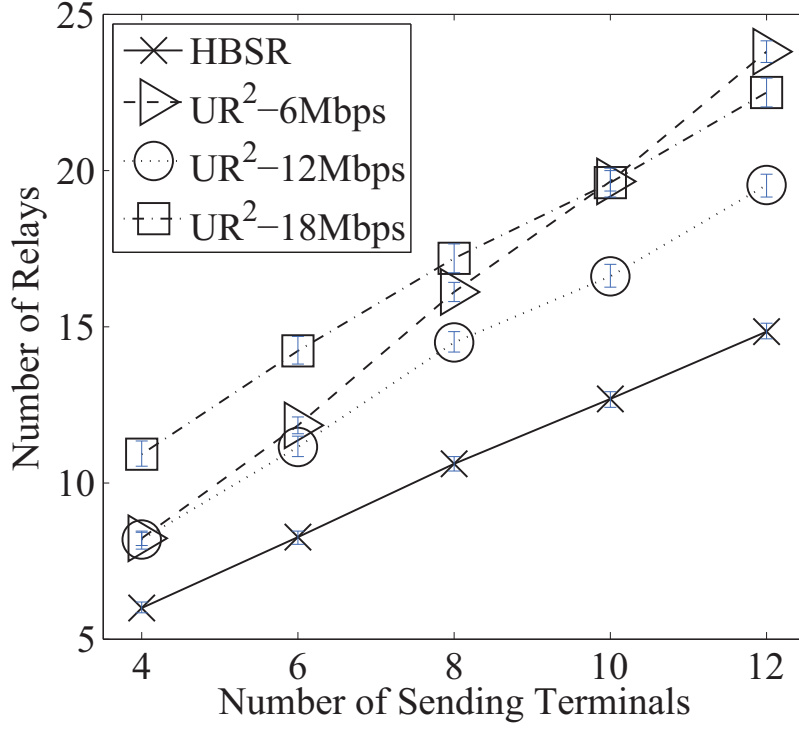


(a) S-MST fails when $|\text{SN}| > 6$. $r_s=4\text{Mbps}$.

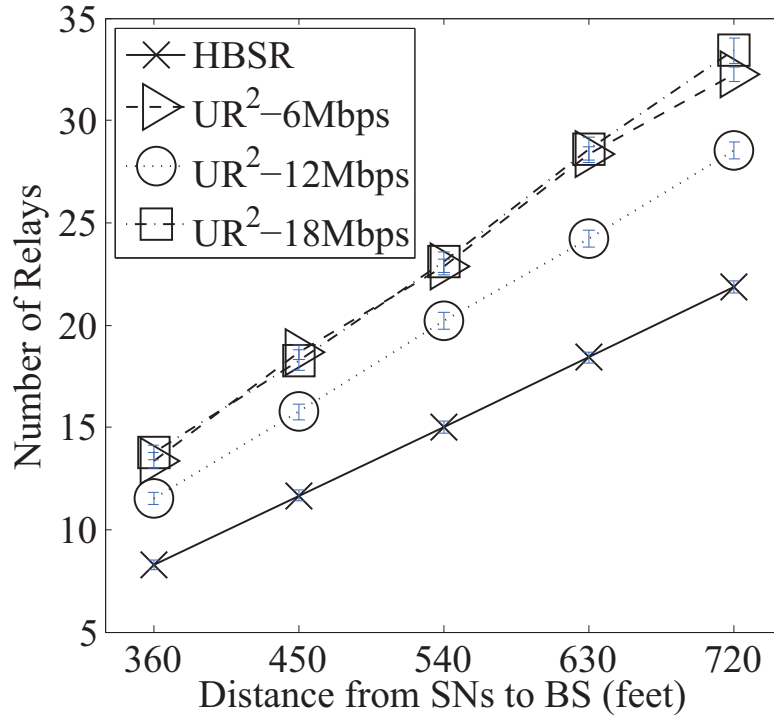


(b) *Open* and *Garden* environment.

Figure 4.5. (a): Compare relay number to bandwidth aware S-MST: Varied flow sending rate r_s of 4Mbps. (b): Obstructive environments.

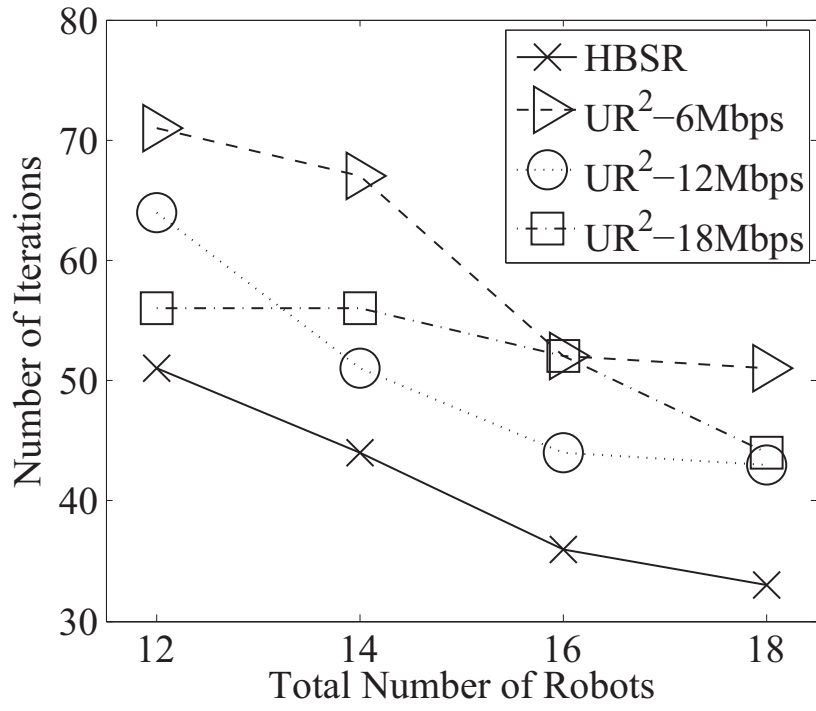


(a) Varying $|\mathcal{SN}|$: 4-12.

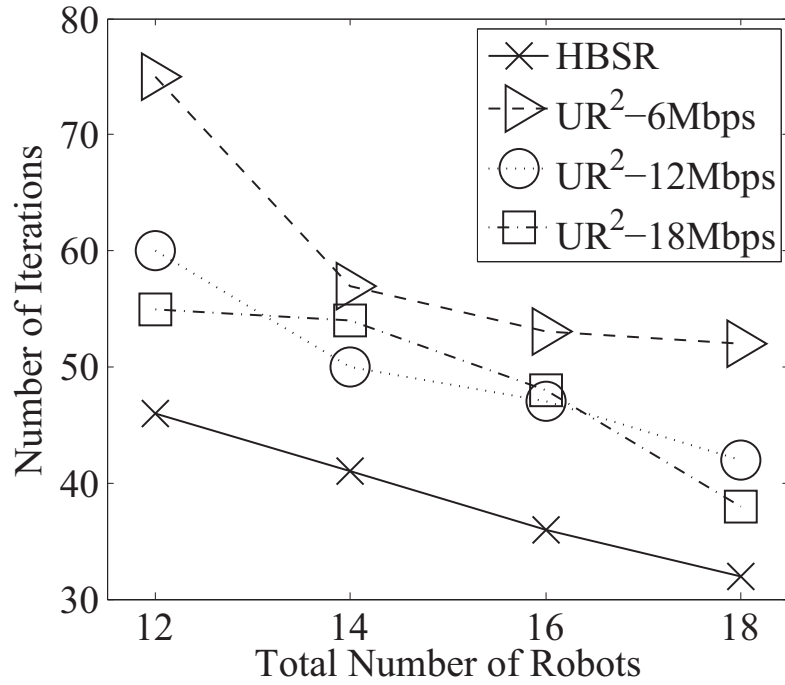


(b) Varied distance. $|\mathcal{SN}|=10$.

Figure 4.6. (a)-(b): Compare relay number to uniform range and rate relay placement (UR²): varied $|\mathcal{SN}|$ and distance to BS.



(a) Garden environment.



(b) Open environment.

Figure 4.7. (a)-(b): Compare exploration efficiency to UR² based exploration by total number of iterations.

for relay placement. We choose S-MST for comparison because there is no existing algorithm exactly match our problem statement, but many solutions to the Steiner tree problem variants are based on the minimum spanning tree [43, 47, 91, 111].

In S-MST, relay are added on the longer than R edges in MST to connect all terminals according to different flow sizes (R is the bandwidth-sufficient range). S-MST has a severe weakness of overflow that our scheme does not have: When all flows are aggregated to one SN and transmitted to BS via a single edge on the MST, flows may exceed the maximum link capacity (e.g. 25Mbps in Table 4.1). Because of overflow and our interest to observe situations where bandwidth's impact is non-trivial, we test on relay number with a fixed range of distance to the BS and use the highest four numbers of SN below overflow with varying sending rate r_s from 2-4Mbps.

All relay number tests in Section 4.6.1, 4.6.2 are run 200 times to obtain the average in the Garden environment. We found that our algorithm is efficient and no instance runs more than one second. The 95% confidence interval is shown by the error bars. We use the polar coordinate (R, θ) to represent SN positions with the BS at $(0, 0^\circ)$. R is random uniformly distributed in $[R_0 - \beta, R_0 + \beta]$. When comparing with S-MST, we set $R_0=480\text{ft}$ and $\beta=240\text{ft}$, and θ is uniformly distributed in $[0^\circ, 90^\circ]$. Fig. 4.4 and 4.5 show HBSR reduces the number of relays by 53.1%, 41.0%, and 38.6% on average. For the situations with the highest two SN numbers before overflow, the relay number reduction is more notable, which is 60.4% on average.

4.6.2 Comparing to Uniform Range and Rate Relay Placement

Second, we compare the number of relays to the *uniform range and rate* (\mathbf{UR}^2) scheme in [52] with two test sets: (1) Varying size of 4-12 SN with fixed range of distance to the BS

($R_0=480\text{ft}$ and $\beta=240\text{ft}$). (2) Fixed size of 10 SN with varying ranges of distance to the BS ($R_0 \in \{360, 480, 600, 720\}$ (ft)); r_s is set as 3Mbps hereafter. As shown in Fig. 4.6(a) 4.6(b), our scheme reduces the number of relays on average by 25.4%-39.0% and 25.5%-36.3% respectively in the two test sets compared to UR² schemes with range corresponding to 6-18Mbps modulation data rates. (UR² with rates above 18Mbps are not compared as they need notably more relays.) These results demonstrate the benefit of considering heterogeneous range and rate in relay placement because of its flexibility in adapting to different bandwidth and range requirements.

4.6.3 Evaluating HBSR's Impact on Mobility Efficiency

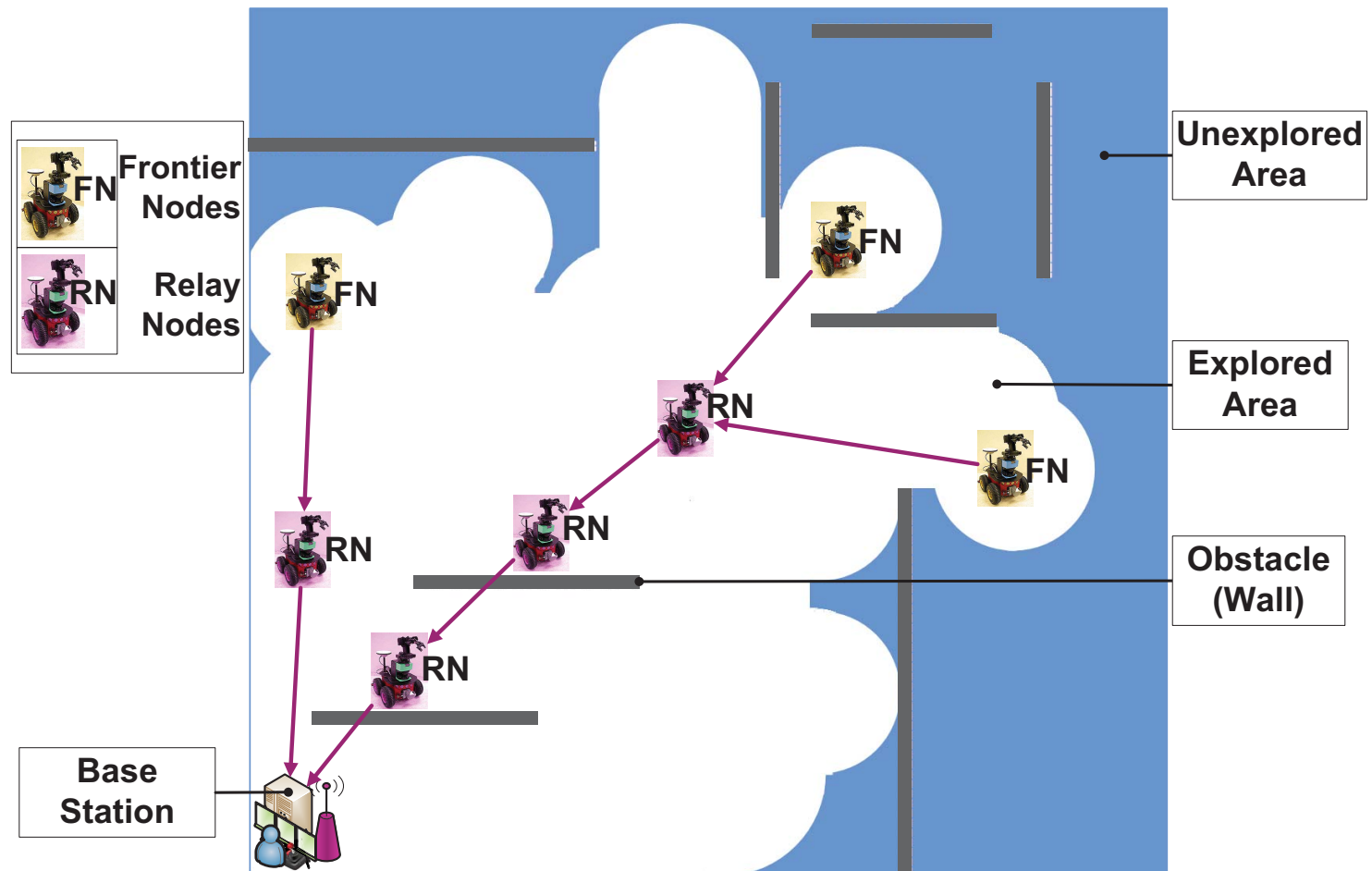


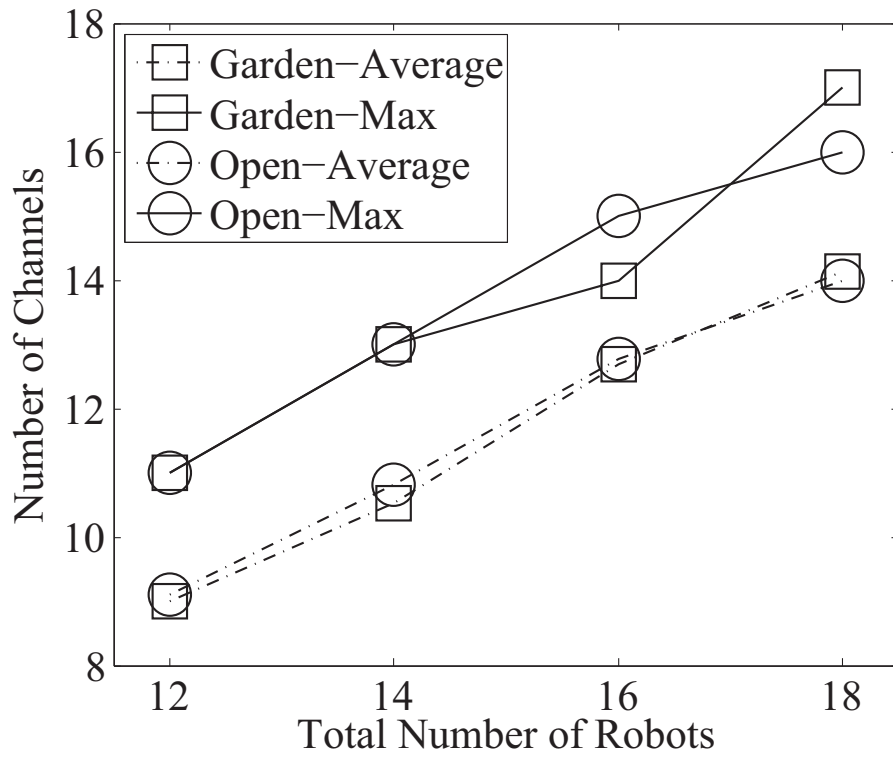
Figure 4.8. Stream aggregation for area exploration.

We also apply HBSR to a coordinated multi-robot exploration strategy with video aggregation [52]. An screenshot of the exploration with HBSR is shown in Fig. 4.8. Relay placement is a critical component in its iterative mobility model. In each round, a portion of the robots will be the relays and the remaining will be the “frontier” robots to explore the area. Less relays can enable more frontier robots to conduct search and thus increase the mobility efficiency. We compare the exploration-efficiency difference when using (1) HBSR vs. (2) UR² to be the relay placement component. The S-MST is not evaluated here due to its overflow problem.

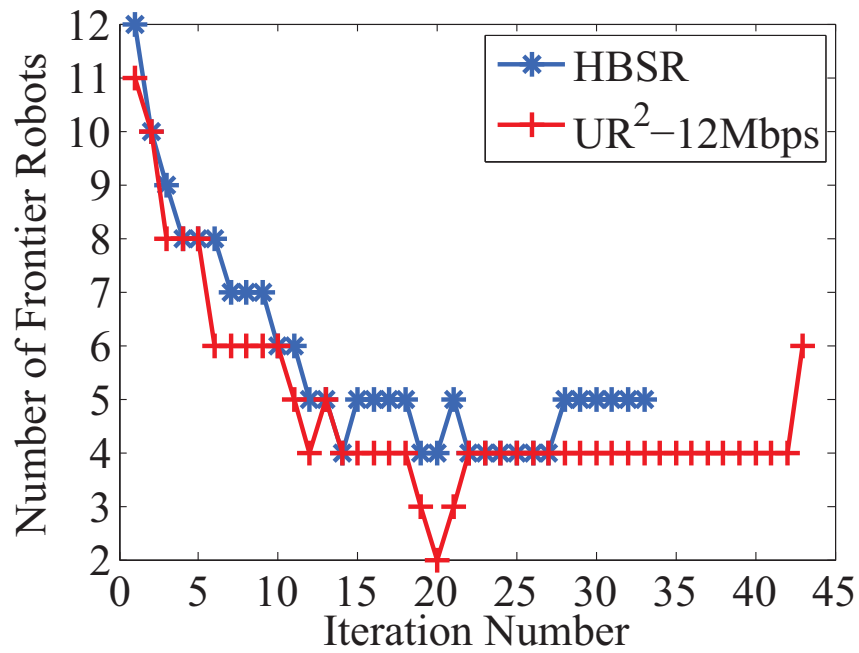
We measured the number of iterations for exploration from a clustered start to 95% of explored area. Each robot has a sensing range of 48ft. Fig. 4.7(a) 4.7(b) demonstrate that the decrease in movement iteration number is on average 18.9%-32.1% and 20.1-34.3% compared to UR² 6-18Mbps for the garden and open environments respectively given the total robot number of 12,14,16,18. Note that the improvement is also influenced by this specific exploration model in [52]. In future work, it is desirable to devise a heterogeneous mobile robot exploration model to fully exploit HBSR’s benefit.

The figures also show that a single range and rate choice has the least number of movement iterations for a certain total robot number; however, exploration with this range and rate may not have the least iterations for other total robot numbers. This fact also demonstrates that a single transmission range based placement lacks the flexibility to adapt to different node densities and the total bandwidth demands.

Furthermore, Fig. 4.9(b) shows the FN numbers in the exploration progress with 18 robots. With less relays, we successfully place more FNs than UR²-12Mbps version in 21 out of 33 iterations (63%), which effectively completes the exploration at 33_{rd} round, 10 rounds earlier than UR²-12Mbps. Besides, Fig. 4.9(a) shows the average/maximum number of channels assigned in



(a) Assigned channel sizes in exploration process.



(b) $|\mathbb{FN}|$ as exploration proceeds.

Figure 4.9. Channel sizes in exploration; $|\mathbb{FN}|$ as exploration proceeds.

exploration for the two environments. Different environments do not have a significant impact on the channel size. The average/maximum number of channels is approximately 3 or 1 less than the total node number, respectively.

4.7 Summary

In this chapter, we consider the heterogeneous link capacity and transmission range in the real-time multimedia aggregation for mobile robot networks.

We formulate the problem as a new variant of the Steiner tree problem called the heterogeneous bandwidth Steiner routing problem (HBSR). We show HBSR's NP-hardness and the upper bound of our solution's approximation ratio. Our solution to HBSR places relay nodes iteratively and aggregates flows when the aggregation is predicted to reduce the number of relays, reducing relay number by an average of 44% compared to the widely used minimum spanning tree based algorithm.

We also found that considering heterogeneous range and rate is beneficial in relay placement and significantly reduces the number of relays compared to the uniform range and rate placement. A single range and rate lack the flexibility to adapt to different stream demands of rates and ranges. Besides, HBSR improves the mobility efficiency in the multi-robot exploration task. In the next chapter, we will introduce an alternative paradigm of online relay deployment to support remote sensing and control in mobile wireless networks.

CHAPTER 5

STARS: Static Relays for Multi-Robot Real-time Search and Monitoring

5.1 Preliminaries and Motivations

In the previous chapters, we attempt to reduce the the number of relaying robots to have more frontier robots to explore the area, aiming to improve the efficiency in area exploration. However, using expensive and powerful robots to merely serve as simple relays may not be a cost-effective approach. Besides, it is extra cost for an increasing number of robots to “retreat” from frontier areas to approach the base station to serve as relays. Furthermore, given a fixed number of robots, an increasing number of relay robots will finally prevent the frontier robots from moving forward.

In this chapter, we present a novel approach of using online relay deployment (deploying relays at the same time as robots migrate) by mobile robots to support stream transmission to the base station given a known environment. The motion planning to support transmission to the base station with relay deployment heavily relies on a new variation of the traveling salesman problem.

Therefore, we first introduce the theoretical problem. Then, we show its application in the tour generation in the multi-robot search problem.

The traveling salesman problem (TSP) is a widely studied combinatorial optimization problem with extensive applications. A variation of TSP that we address here is the minimum time *precedence constrained two traveling salesman (PC2TSP)* from a given city. We first introduce the problem and then show its application.

PC2TSP can be stated as follows. Given 1) a finite set of cities $\mathbb{C} = \{1, 2, \dots, n\}$ with each city marked as a consumer, a supplier, or both, and each consumer city has its supplier city set, 2) the cost c_{ij} of traveling time between each pair of cities $i, j \in \mathbb{C}$, c_{ij} and c_{ji} may differ, 3) the precedence constraint that the arrival time plus waiting time at each consumer city is no earlier than the arrival time of all its supplier cities, to find two non-overlapping tours that allow waiting time at cities and both start from the base city 1 to visit each of the remaining cities exactly once, such that the maximum of the two tour times is minimized.

The problem is more challenging than other variants of multi-traveling salesman problem (mTSP) in that a consumer city and some of its supplier cities may be visited by different travelers. Therefore, it is possible that a traveler arriving at a consumer city i must wait for the other traveler to visit i 's supplier cities, introducing the dependency between travelers.

One interesting application of PC2TSP in mobile surveillance systems is the problem of minimum time *two-robot real-time search with relay deployment (2RSRD)*, which is described as follows: Given two robots, or mobile sensors, with ability of deploying relays as they search the area, how to compute the moving paths for robots and the proper time and positions to deploy relays along the movement path, so that i) robots can efficiently search the area with minimum amount of time; ii) video streams from robots at any sensing position can be successfully transmit-

ted back to the base station (BS). Figure 5.1 illustrates the two robots in the search process. Any stream transmission at the sensing positions is relayed by a subset of two relays in the middle.

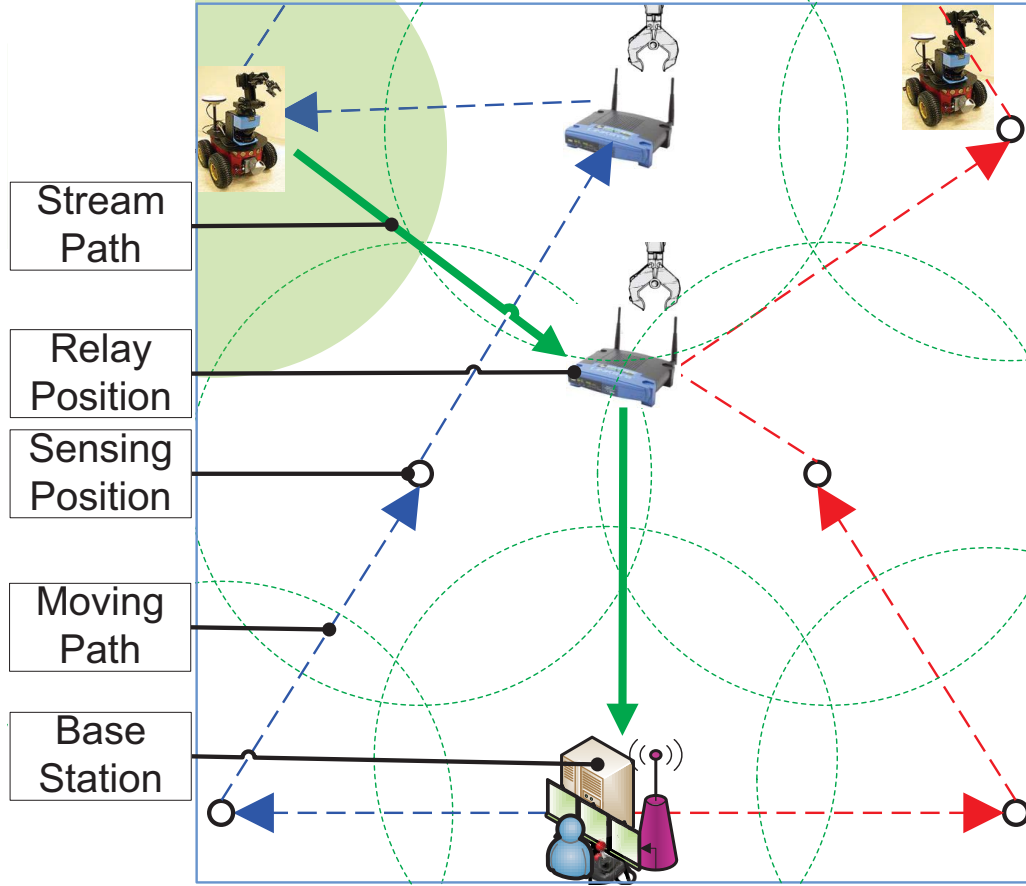


Figure 5.1. Two robots search the environment and deploy static relays as they travel along the paths. The left robot transmits streams to the base station via the center relay deployed by the right robot.

Similar to PC2TSP, 2RSRD implies a *precedence* constraint: When a robot conducts search and video transmission at sensing locations, supporting relays must be deployed *prior to* the robot's video transmission, so that streams can always be transmitted back to the BS on a valid path.

Now that 2RSRD resembles PC2TSP, we model 2RSRD in part by PC2TSP as follows. The mobile robots become the travelers. The sensing positions for robots to conduct search become the consumer cities and the positions at which relays are deployed become the supplier cities.

For each sensing position modeled as a consumer city, its corresponding supplier city set is all of its supporting relay positions that form a valid path to the BS. Therefore, as long as we have 1) identified sensing and relay positions, and 2) assigned the precedence constraint that choose the relay set for each sensing position as a valid path to BS, we can use PC2TSP to solve 2RSRD as the final step to generate tours.

By modeling in part by PC2TSP, we present *Static Relay aided Search* (STARS) to solve 2RSRD. STARS enables remote robotic sensing and control. It has extensive applications such as reconnaissance, surveillance, search and rescue missions in dangerous and hostile areas. One example is the robotic teleoperation and remote search tasks in the recent Japan earthquake debris and radioactive environments. STARS substantially reduces cost by replacing expensive robots by low cost static relays, compared to a homogeneous mobile robot system where robots have to be the relays to transmit streams. Another benefit is that STARS also enables the deployment of static sensors to monitor the “suspicious areas” where the human operator and the computer system decide that the targets may (re)appear after the robots migrate away. The details of how STARS supports constant monitoring by static sensors is described in Section 5.6.2.

With this heterogeneous system of mobile robots, static sensors and relays, STARS has the merits of both static sensors and mobile robots. Sensors are relatively inexpensive devices that are good at constantly monitoring regions, but they require large quantities in deployment to cover the area. Robots are more powerful to perform complex tasks but are considerably more expensive. They are better suited for situations requiring a single time search and exploration process. Compared with homogeneous static sensor networks, the proposed method saves unnecessary sensors because sensors only need to be placed as the suspicious area rather than the whole area. Compared with homogeneous mobile robot networks, our scheme has notably less cost but can monitor

suspicious area constantly thanks to the deployed sensors.

To the best of our knowledge, STARS is the first of its kind to leverage online relay deployment as a communication backbone to support real-time communication, remote control and sensing with mobile robots and static sensors. The problem is motivated but has many challenges. We discuss these challenges in Section 5.3.1.

5.1.1 Main Contribution

- We first present a problem called precedence constrained two traveling salesman (PC2TSP). We propose a near-optimal heuristic to PC2TSP to generate tours by clustering points, generating optimal single-traveler tours, and tour pruning and balance.
- To solve PC2TSP, we propose the tour validation and dependency checking scheme to detect deadlock and self-flipped cases that are invalid. It also checks the dependency on two tours and adds necessary waiting time.
- To solve 2RSRD, we propose STARS, which divides the problem into two subproblems: 1) preprocessing: identify visiting positions and assign the precedence constraint, which are modeled by set cover, Steiner connected dominating set, and breadth-first search; 2) tour generation, which is modeled by PC2TSP.

To maintain consistency and show a valid application of PC2TSP, we present the solution of PC2TSP in the context of STARS as tour generation. However, it can be used as a stand alone solution to PC2TSP with minor changes given in Section 5.5. The rest of the paper proceeds as follows. Section 5.2 introduces the system model. In Section 5.3, we formulate the problem and discuss the challenges. Section 5.4 and 5.5 give the preprocessing and tour generation. Then,

several extensions are discussed in Section 5.6. Performance evaluation is in Section 5.7. Section 5.8 concludes the chapter.

5.2 System Model

Environment Model: We use a 2D occupancy map to model the environment as an unsearched rectangle region $\mathbb{R} = X \times Y$ with grid cells similar to that in [55]. The cell size equals to the footprint size of the robot. A cell status includes searched, unsearched, and obstructed. An obstacle cell cannot be visited by the robots nor can a relay be deployed on. Obstacles do not affect communication but block the sensing range. We assume the obstacle distribution in the environment is known. We would like to tackle the unknown or inaccurate obstacle distribution in the future work. *Base station BS* is the control center where the human operator can remotely monitor and potentially operate the robots if necessary.

Robot Model: The number of robots in our model is limited as two because the goal is to use low cost static relays to replace high cost mobile robots. The *two robots* are denoted as a and b . The solution is extensible to more than two robots, which will be discussed in Section 4. The robots move at a constant speed where the acceleration and deceleration time is not considered. Each robot has traveling, communication, and sensing capability. It scans the environment using cameras and acoustic sensors with sensing range r_s and communicates with other nodes with communication range r_c by a 802.11 radio, where $r_c \geq r_s$. We denote the communication to sensing range ratio as $\theta = r_c/r_s$. Since only 2 robots are sending streams, we assume bandwidth is adequate in transmission. The unit disk model is used for sensing and communication. As robot's sensing devices, such as laser finder and cameras, are more powerful than those in limited

capability sensors, the sensing range is more stable. A more sophisticated model may not be necessary.

Search with Relay Deployment Model: The robots search the region by visiting a *sensing position* **SP**, where the robots stop and enter the *search and transmit interval* (**STI**) and the unsearched cells in its covered area with radius r_s will be marked as searched. The STI is for the robots to sense the area and send video streams back to the BS. The ratio of searched cells out of total non-obstacle cells is denoted as θ_s . The *completion* of search is defined by $\theta_s \geq \delta_s$, where δ_s is a *search complete threshold*, e.g. 99%. **SP** is the set of SPs whose visit complete the search. **SP** includes the BS.

The robots deploy relays by visiting a *relay position* **RP** in the search tour. A relay is a non-mobile device with 802.11 for transmitting streams from a robot with same communication range r_c . A robot has a maximum carrying capacity of n_r relays. We assume STI and relay deploying time are negligible.

SP(a) and **SP(b)** denote the vector of SPs robot a and b need to visit; **RP(a)** and **RP(b)** denote the vector of RPs a and b need to visit. $\mathbf{P}_a, \mathbf{P}_b$ denote the ordered vector of sensing and relay positions a and b traverse: $P_a = SP(a) \cup RP(a)$, $P_b = SP(b) \cup RP(b)$. $\mathbf{T}_a, \mathbf{T}_b$ denote the *tour* a and b traverse, or the vector of position and the waiting time w at each position: $T_a = \{P_a, w(P_a)\}$, $T_b = \{P_b, w(P_b)\}$. A *path* denoted by $\text{path}(V)$ gives a route with an ordered vector of positions of $V=(v_0, v_1, \dots, v_n)$. \mathbf{t}_a and \mathbf{t}_b give the tour time in seconds, or the total time needs to traverse all positions in P_a or P_b including the possible waiting time. $t(i)$ gives the start of search time for position i .

We compute the tour off line with the information of the environment before the actual search starts. We do not model robots to come back to the BS immediately after the search is complete.

Because robots should remain in the frontier area for remote sensing.

Discussions: Due to its high complexity, the problem is modeled as above and we focus on the tour generation. The model is easily extensible to consider a non-trivial STI and relay deploying time, which will be discussed in Section 4. In the future, we would like to tackle: 1) the bandwidth aware relay placement, 2) the probability based communication model, and 3) the online tour adjustment with inaccurate initial obstacle distribution. Note that the first two changes and most parameters in the model only affect the first subproblem in STARS and have no impact on the solution to the second subproblem of tour generation. For the third change, we may recompute the tour by updating the new data of obstacles as input.

5.3 Problem formulation

Given the system model, we formulate the problem and explain the basic idea.

Minimum time *two-robot real-time search with relay deployment* (**2RSRD**): Given a region \mathbb{R} , two robots a and b each with relay carrying capacity of n_r , compute tours T_a, T_b to obtain the minimum search time t_{ab} such that:

$$\text{Minimize } t_{ab} = \max\{t_a, t_b\}, \quad (5.1)$$

$$\text{Subject to } \mathbb{SP} = SP(a) \cup SP(b), \quad (5.2)$$

$$\exists path(V), V = (BS, rp_0, \dots, rp_n, i), \forall i \in \mathbb{SP}, \quad (5.3)$$

$$dist(v_j, v_{j+1}) \leq r_c, \quad \forall v_j \in V, j \leq n+1, \quad (5.4)$$

$$t(rp_j) \leq t(i), \quad \forall j \in 0, \dots, n, \quad (5.5)$$

$$|RP(a)| \leq n_r, \quad |RP(b)| \leq n_r, \quad (5.6)$$

Notations	Definitions
PC2TSP	Precedence constrained two traveling salesman: the theoretical problem.
2RSRD	Two-robot real-time search with relay deployment: the robotic search problem.
STARS	STATIC Relay-aided Search: our solution to 2RSRD, where the tour generation subproblem is modeled by PC2TSP.
\mathbb{R}	Search region.
\mathbb{P}	Precedence constraint matrix.
$\mathbb{SP}, \mathbb{RP}, \mathbb{SP}, \mathbb{RP}$	Sensing position, relay position; their set.
a, b	The two mobile robots.
BS	Base station.
P_a, P_b	The ordered vector of sensing and relay positions a and b traverse.
T_a, T_b	Tours of robot a and b.
t_a, t_b	Tour times in seconds of robot a and b.
t_{rd}	Relay deploying time at the relay positions.
$t(i)$	The start of search time for position i .
STI	Search and transmit interval when robots stops at the sensing positions.
S-CDS	Steiner connected dominating set.
n_r	Robot's relay carrying capacity.
$dist(i, j)$	Distance between i and j .
r_c, r_s, θ	Communication and sensing range, and their ratio.

Table 5.1. Notations in STARS.

Where rp_i is a relay position in the path ($rp_i \in RP(a) \cup RP(b)$). Equation (2) indicates the search is complete. Equations (3)-(5) give that for each sensing position i in a or b's tours, there is a valid path back to the BS at the time of $t(i)$. Equation (6) gives the carrying capacity constraint.

5.3.1 Challenges

We first prove 2RSRD is NP-hard and then show other challenges to directly solve the problem.

Theorem 5. *2RSRD is NP-hard.*

Proof. Given an instance of a square region, we can exactly duplicate the region and put the duplicated copy on the right of the original one. With the BS on the bottom center of duplicated region, minimum t_{ab} is achieved when both a and b traverse on its own half region the same way. This is because any traverse on the other half part will increase t_{ab} . Therefore, a polynomial time solution in 2RSRD will lead to a polynomial solution to a single *precedence constrained asymmetric traveling salesman problem (PCATSP)* on the positions. It contradicts to the fact that PCATSP is NP-hard [112], and this completes the proof. \square

There are other challenges to directly solve 2RSRD other than its NP-hardness. First, the robots may need to wait for each other if one relies on the other's not deployed relay to transmit streams. There is a dependency between both robots: One may have to wait for the other to deploy the supporting relays. The dependency and waiting time make it challenging to directly apply linear programming formulation to solve the problem.

Second, search time t_{ab} is affected by the robot's relay carrying capacity and whether to have a goal of reducing relay size. Suppose that relays are sufficient, carrying capacity is infinite, and robots place a relay at every SP. As long as robots travel on consecutive SPs, the stream

transmission requirement is always met (recall $r_c \geq r_s$). The relay requirement will have little impact on the tour generation. The problem is then simplified and less interesting to solve. On the other hand, if RPs are only placed selectively (such as at only S-CDS positions as in Section 5.4.2), the tour computation is constrained by visiting the relay positions first before visiting other sensing positions that are supported by these relays.

Third, the precedence constraint is *uncertain* compared with that in PCATSP, where position i preceding j is explicitly stated. 2RSRD requires that *at least* one valid relay path has been deployed at the searching time on any SP. However, multiple sets of relays can fulfil this requirement. To illustrate, there are 4 RPs as shown in Figure 5.2. When search starts at SP, at least one pair of RPs (1,3),(2,4),(1,4),(2,3) is required to be deployed. The uncertainty makes the problem more difficult to deal with.

Therefore, to solve 2RSRD optimally, which is to find *all possible*: 1) SPs that complete coverage, 2) RPs that cover all SPs (which is a S-CDS as we define below), 3) precedence constraints that ensure the stream relay requirement, 4) partitioning of visiting positions, 5) permutation of tours, and 6) adding waiting time for each pair of tours when dependency occurs.

5.3.2 STARS: STAtic Relay aided Search

Due to the problem's high complexity, we propose a high quality heuristic called **STARS**, or the *STAtic Relay aided Search*. It divides the problem into two subproblems: 1) preprocessing: identify visiting positions and assign the precedence constraint, 2) tour generation, which is modeled by PC2TSP.

We present a valid solution for the first subproblem. We also show how our near-optimal so-

lution to PC2TSP is applied to solve the second subproblem of tour generation. Such division separates the key tour generation problem from a specific region with different visiting position selections, precedence constraints and robot carrying capacities. In future work, the tour generation is important to evaluate different schemes of selecting visiting positions and precedence constraints, making it possible to find the optimal visiting positions and precedence constraint.

The formulation for each part of the subproblems is as follows:

Problem 1a: Identify sensing positions: Given a region \mathbb{R} , how to place a minimum number of sensing positions \mathbb{SP} that completes the search (Coverage requirement: searched cell ratio $\theta_s \geq \delta_s$):

$$\text{Minimize } |\mathbb{SP}|, \quad (5.7)$$

$$\text{Subject to } \bigcup_{i \in \mathbb{SP}} \text{cell}_{searched}(i) \geq \delta_s \cdot |\text{cell}_{nonobstacle\mathbb{R}}|$$

Problem 1b: Identify relay positions: As the robots visit all sensing positions, the area will be fully searched. However, the video streams cannot be transmitted back to the BS unless a path formed by relays always exists. \mathbb{RP} forms a communication *backbone*. Without worrying about the visiting sequence at this step, the backbone must at least “cover” all \mathbb{SP} including the BS. We assume the user would like to save relays and with such an attempt the carrying capacity constraint is satisfied (Satisfying Eq.(6)). An extension to take advantage of extra relays under the carrying capacity (difference of carrying capacity and the computed minimum number of relays) is presented in Section 5.6.5. Hence, the problem is: given \mathbb{R} and \mathbb{SP} , how to place a minimum number of relay positions \mathbb{RP} that satisfies 1) each SP is either in \mathbb{RP} or a neighbor of RP; 2) \mathbb{RP}

includes the BS, and \mathbb{RP} is connected:

$$\begin{aligned}
& \text{Minimize } |\mathbb{RP}|, \\
& \text{s.t. } i \in \mathbb{RP} \mid \text{dist}(i, j) < r_c, \quad \forall i \in \mathbb{SP}, \exists j \in \mathbb{RP}, \\
& \quad \exists \text{path}(V), V = (v_0, \dots, v_{n-1}), \quad \forall v_0, v_{n-1} \in \mathbb{SP}, \\
& \quad \text{dist}(v_k, v_{k+1}) \leq r_c, \quad \forall v_k \in V, k < |V| - 1, \\
& \quad BS \in \mathbb{RP}
\end{aligned} \tag{5.8}$$

In addition, a simple extension to leverage the pre-existing relays is presented in Section 5.6.4.

Problem 1c: Assign the precedence constraint: Given \mathbb{R} , \mathbb{SP} , and \mathbb{RP} , how to define a *precedence constraint matrix* $\mathbb{P} : \mathbf{p}_{ij}, \forall i, j \in \mathbb{SP} \cup \mathbb{RP}$:

$$p_{ij} = \begin{cases} 1 & \text{position } i \text{ precedes } j \\ 0 & \text{otherwise} \end{cases}$$

Validity: Matrix \mathbb{P} is *valid* if and only if any tours on \mathbb{SP}, \mathbb{RP} satisfying \mathbb{P} meet the constraints defined in Eq.(3)-(5).

Problem 2: Tour generation: Given $\mathbb{R}, \mathbb{SP}, \mathbb{RP}$, how to find two minimum time tours T_a, T_b that satisfy \mathbb{P} :

$$\begin{aligned}
& \text{Minimize } t_{ab} = \max\{t_a, t_b\}, \\
& \text{Subject to } P_a \cup P_b = \mathbb{SP} \cup \mathbb{RP}, \\
& \quad t(i) \leq t(j), \quad \forall p_{ij} = 1
\end{aligned} \tag{5.9}$$

Rationale: Problem 1a is formulated by minimum $|\mathbb{SP}|$ as a heuristic because less visiting positions may lead to a shorter total tour. Problem 1b is formulated to reduce the number of relays.

Problem 1c is to remove the uncertainty of precedence constraints for the tour generation. With the result of problems 1a,1b,1c as input, the problem 2 is formulated to generate the tours.

5.4 Preprocessing

5.4.1 Identify Sensing Positions

Recall the problem formulation 1a in Section 5.3.2. We formulate the problem as the well-known NP-hard *set cover* problem, which is to select a minimum number of sets so that the union of the sets contains all the elements in the universe. Here, the search region is divided into grid cells. With each cell as an element, all the non-obstacle cells become the universe. Each SP's sensing area is a set that contains some elements within sensing range, r_s . Since the problem is NP-hard, we leverage a heuristic modified from the *triangle lattice pattern* (TLP), which is proven optimal in terms of the number of sensing disks to obtain full coverage [113] if the region is free of obstacle.

Alg. 13 describes the procedure.

Algorithm 13: Identify sensing positions for full region coverage.

Input: A region \mathbb{R} .

Output: SP.

- 1 Follow the triangle lattice pattern to place SP on non-obstacle cells.
 - 2 **While** $\theta_s < \delta_s$ **do** greedily select a position i to cover a maximum number of uncovered cells. Insert i into SP.
 - 3 With an equal coverage, the position with minimum distance to the nearest SP neighbor is selected.
-

5.4.2 Identify Relay Positions

Recall the problem formulation 1b in Section 5.3.2. We model the problem as the minimum *Steiner connected dominating set* (**S-CDS**) problem.

Connected Dominating Set (CDS) problem is a classical problem in combinatorial optimization with the following definition: Given a graph $G = (V, E)$, find the smallest subset S of vertices that induce a connected subgraph and each vertex in $V - S$ is adjacent to at least one vertex in S . S-CDS is a generation of CDS where only a specified set $R \subseteq V$ of required vertices has to be dominated by a connected dominating set [114]. Given all non-obstacle cells in \mathbb{R} as V . \mathbb{RP} only needs to cover \mathbb{SP} , which is a subset of V . We require \mathbb{RP} induce a connected subgraph because streams from any vertex in \mathbb{SP} is sent to the BS.

Differing from those formulations in previous chapters, this relay placement problem is not modeled as the Steiner Minimum Tree with the Minimum Steiner Point (SMT-MSP) problem. The reason is that a terminal point can always also serve as a Steiner point to connect to other terminal points in SMT-MSP. However, in our application, the robots only pass the sensing positions once. A sensing position (terminal point) cannot serve as a Steiner point (relay) unless a real relay is

deployed.

Algorithm 14: Steiner-CDS algorithm to identify relay positions

Input: \mathbb{SP} .

Output: \mathbb{RP} .

// 1. Obtain the dominating set (DS) on \mathbb{SP} .

1 **if** $|\mathbb{SP}| < \delta_{ds}$ **then**

2 Obtain DS by the optimal DS finder.

3 **else**

4 Obtain DS by the greedy set cover with maximum nodes.

// 2. Apply Steiner tree approximation algorithm to connect DS.

5 Call *Steinerized Minimum Spanning Tree* [91] to connect DS. The newly added nodes are

RP_2

6 Return $\mathbb{RP} = DS \cup RP_2$.

The basic idea is simple. First find the minimum dominating set which may not be connected, then add vertices to connect the dominating set. The details of the algorithm are shown in Alg. 14. The algorithm is similar to the Steiner-CDS algorithm in [114]. The three major differences are: First, we use an optimal dominating set (DS) finder (finding DS is NP-hard) when the size is smaller than a threshold $\delta_{ds} = 70$ and use the greedy heuristic as in [114] when the input size is larger. Leveraging a recursive algorithm with techniques such as node ordering, branch and bound, and backtracking, the DS finder can compute up to 70 vertices to optimal in less than 1 minute on a regular notebook, improving the S-CDS quality. **DS** is the overlapping set: $DS = \mathbb{SP} \cap \mathbb{RP}$.

The second difference is that RP_2 are placed on the shortest obstacle-aware paths computed by A* search between its end vertices when necessary to avoid obstacles. A* search guarantees

the same optimal result as Dijkstra but is more efficient. Because it reduces the searching space compared to BFS using an admissible and consistent heuristic function which does not overestimate the distance to the goal [55]. Third, we always include the BS in DS because the BS belongs to the backbone: $BS \in \mathbb{RP}$.

5.4.3 Assign the Precedence Constraint

Algorithm 15: Assign parentIDs and the precedence constraint

Input: \mathbb{SP}, \mathbb{RP} .

Output: \mathbb{P} .

// 1. Assign parentIDs. The BS has an invalid parentID(-1).

1 Breadth-first search on \mathbb{RP} to obtain the $RPlevel$ in BFS. Assign a parentID in BFS for each relay according to BFS queueing order.

2 For each $i \in \mathbb{SP}_{pure}$, insert all i 's 1-hop RP neighbors into its parent candidate set.

3 Loop through $i \in \mathbb{SP}_{pure}$, **if** i has a single parent candidate j **then**

4 $i.parentID \leftarrow j$

5 **else**

6 $i.parentID \leftarrow$ minimum $RPlevel$ node j .

7 $j.children.push_back(i)$. End loop.

// 2. Assign values in the precedence matrix \mathbb{P}

8 For each $j \in \mathbb{SP}$, get ancestors of j by recursively backtracking its parentID until root. Set

$p_{ij} = 1$ when i is an ancestor of j .

Recall the problem formulation 1c in Section 5.3.2. We first define notations, then present our solution and prove its validity. We define the *pure sensing positions* $\mathbb{SP}_{pure} = \mathbb{SP} \setminus \mathbb{DS}$ (' \setminus ' means

subtraction), and *visiting positions* $\mathbb{VP} = \mathbb{SP} \cup \mathbb{RP}$.

Alg. 15 shows the procedure of assigning the precedence constraint by first assigning a *single* parentID for each node in \mathbb{VP} . Assigning parentIDs is important because we can always backtrack a node's parentID recursively to the BS, forming a valid path for communication. The procedure builds a precedence tree to find parentIDs on \mathbb{VP} . The tree has 1) the non-leaf backbone as \mathbb{RP} , 2) leaf nodes as \mathbb{SP}_{pure} , 3) BS as the root.

Theorem 6. *The precedence constraint matrix \mathbb{P} obtained by Alg. 15 is valid.*

Proof. Each node in \mathbb{VP} has a single parentID. Because $\mathbb{SP} \subseteq \mathbb{VP}$, each node in $i \in \mathbb{SP}$ has a single parentID. For each i , its ancestors form a valid path back to the root BS, being part of the backbone. Because \mathbb{P} is defined that all ancestors of i are required to be visited before visiting i , \mathbb{P} is valid, and this completes the proof. \square

5.5 Tour Generation by PC2TSP

Recall the problem formulation 2 in Section 5.3.2. We model the problem as PC2TSP. By a similar proof of Theorem 5 we can prove PC2TSP is NP-hard. Therefore, we provide a near-optimal heuristic that 1) clusters the nodes (with possible overlapping of shared relays) into two parts, 2) solves them individually by PCATSP optimally, and 3) prunes the shared relays and balances the two tours.

The tour generation procedure can be used as a stand alone solution to PC2TSP with the changes as follows. 1) Change robots, sensing positions, relay positions, and the BS to travelers, consumer cities, supplier cities, and the base city. 2) Change ones ancestors and parentID to

its supplier cities in tour validation and dependency checking. 3) Values in the cost matrix c_{ij} and whether to have the conversion from Hamiltonian path depend on the application.

5.5.1 Cluster Visiting Positions

We first cluster each nodes $i \in \mathbb{SP}$ according to its x coordinate and insert it accordingly to either P_a or P_b : assign to robot a or b by whether the position is on the left or right of the center line. A *center line* $X = x_{\mathbb{R}}/2$ is a vertical line to partition the region in the middle. Then, the supporting relays, or the ancestors of each $i \in \mathbb{SP}$ are assigned accordingly to the same cluster as i . Therefore, position sets P_a and P_b include the sensing positions and their supporting relays. Relays may be (partially) shared by P_a and P_b . The motivation of such clustering is to make a tour *self-reliant* without need of waiting for relays from the other tour.

We have tried more advanced schemes such as K-means and also attempted to balance the two sets to have similar amount of nodes such that $||P_a| - |P_b|| \leq 1$, however they do not improve the performance. Although the initial P_a and P_b may be unbalanced, our tour pruning and balance in Section 5.5.3 work effectively in balancing the tours.

5.5.2 Obtain Optimal Single Robot Tour

Given the positions P_a and P_b for robot a and b to traverse, and the precedence constraint in \mathbb{P} relevant to each part, we need to compute the precedence constrained shortest Hamiltonian path starting at BS for a and b individually. We first convert the Hamiltonian path problem to the traveling salesman problem, then model the problem as (single) PCATSP [115].

Conversion: In our system model, robots will not immediately return to the BS after the search

completes. Therefore, the shortest Hamiltonian path problem with a given starting city should be first converted to an asymmetric TSP by setting the distance on the arcs from all vertices to BS as zero [116]. It forces the evaluation of all possible paths starting with BS and disregards the way back to the BS in the tour.

Mathematically, precedence constrained asymmetric (single) traveling salesman problem (PCATSP) is:

$$\begin{aligned}
& \text{Minimize } \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}, & (5.10) \\
& \text{Subject to } \sum_{i=1, i \neq j}^n x_{ij} = 1, \quad \forall j = 1, \dots, n, \\
& \quad \sum_{j=1, j \neq i}^n x_{ij} = 1, \quad \forall i = 1, \dots, n, \\
& \quad p_{ij} \geq x_{ij}, \quad \forall i, j = 2, \dots, n, \quad i \neq j, \\
& \quad p_{ij} + p_{ji} = 1, \quad \forall i, j = 2, \dots, n, \quad i \neq j, \\
& \quad p_{ij} + p_{jk} + p_{ki} \leq 2, \quad \forall i, j, k = 2, \dots, n, \quad i \neq j \neq k, \\
& \quad p_{ij} = 1, \quad \forall j = 2, \dots, n, \quad \forall i \in SPC_j
\end{aligned}$$

Where c_{ij} is the traverse time from i to j based on the obstacle aware path length by A* search and a traverse velocity. c_{ij} forms the *cost matrix* \mathbb{C} . x_{ij} and p_{ij} are binary and $x_{ij} = 1$ indicates position i precedes j immediately while $p_{ij} = 1$ means position i precedes j not necessarily immediately in the tour. SPC_j gives the subset of positions in $\{2, \dots, n\}$ that are required to precede j .

Solving PCATSP by branch and cut method with the above integer linear programming formulation is relatively fast for a reasonable size despite its NP-hardness. Real-world instances with

200 nodes PCATSP have been solved optimally in a few minutes [117].

5.5.3 Prune and Balance Tours

Pruning shared relays

Algorithm 16: Path pruning

Input: T_a, T_b .

Output: updated T_a, T_b .

// Obtain shared relays and divide into segments if $|SR| > k$

1 Obtain shared relays SR from T_a, T_b in order. Divide it into $\lceil |SR|/k \rceil$ segments, each with k nodes (except the last).

2 **foreach** *segment* i **do**

3 Get powerset of i : $PS(i)$, $|PS(i)| \leq 2^k$

4 **foreach** $j \in PS(i)$, $j' = i \setminus j$ **do**

5 Get temp tours by $T'_a = T_a \setminus j$, $T'_b = T_b \setminus j'$. Update search time.

6 Call Alg. 17 to check dependency and obtain the true time t'_{ab} by adding wait time (if it exists) on T'_a, T'_b .

7 Save j_{min}, j'_{min} with minimum t'_{ab} and then update tours: $T_a = T_a \setminus j_{min}$,
 $T_b = T_b \setminus j'_{min}$.

Tour balance is to minimize the total search time t_{ab} , which is dominated by the longer tour.

The two paths by PCATSP are self-reliant but may have visited duplicate relay positions. Our goal is to reduce t_{ab} by reducing duplicate RPs, considering the possibility and cost of introducing

dependency with waiting time.

Algorithm 17: Tour validation and dependency checking

Input: Local copies of any tours T_a, T_b .

Output: t_{ab} , updated T_a, T_b with waiting time, isValid.

```

1  isDeadlock=false, isValid=true.

2  if isFlipped( $T_a$ ) || isFlipped( $T_b$ ) then return  $t_{ab}$ =FLT_MAX, isValid=false.

3  Iterators  $it_a, it_b$  at the beginning of  $T_a, T_b$ .  $t_a = 0, t_b = 0$ 

4  repeat
5      repeat
6           $it_a$  keeps on proceeding to the next position ++ $it_a$  in  $T_a$ . At each new position  $it_a$ ,
            first to update time by  $c_{ij}$ :  $t_a + = c\{*(it_a - 1), *it_a\}$ , and then make sure the robot
            waits for its parent:  $t_a = \max(t_a, t(\text{parentID}(*it_a)))$ , update wait time vector
             $w(P_a)$ . update the visited time at  $t(it_a) = t_a$ , mark  $it_a$  as visited.
7      until  $it_a$  is blocked when its parent is not visited.
8      Same repeat procedure as above for  $it_b$ , which keeps on proceeding until it is blocked.
9      if Both  $it_a, it_b$  are blocked then isDeadlock=checkDeadlock( $it_a, it_b$ ).

10 until  $it_a, it_b$  reach the end of  $T_a, T_b$  || isDeadlock

11 if isDeadlock then return  $t_{ab}$ =FLT_MAX, isValid=false.
```

The pruning strategy has shown in Alg. 16. With $|SR|$ number of shared relays, we try all possible ways of allocating them to robot a and b by computing the powerset. (A powerset of a set S is the set of all subsets of S .) Due to the exponential nature, we set the maximum segment length $k = 12$ and to divide SR into segments to limit the powerset size. For each segment, we compute the best allocation of new tour cost by first computing the time based on paths from c_{ij} .

Then the dependency checking is called in line 7 to check whether dependency occurs and to add waiting time, if it exists. Figure 5.3(a) illustrates the idea. Denote the two shared relays as SR 1 and 2. They are divided by each tour evenly after evaluating $PS(\{1, 2\}) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

Validate tours and check dependency: With the path positions P_a, P_b , we need to verify whether they are feasible by checking whether self-flipped and deadlock cases occur. Then, we compute the true search time considering dependency.

Definitions: A tour is *self-flipped* if and only if it has a position i whose ancestors are visited after i . A self-flipped tour is invalid because by itself it violates the precedence constraint. A position i is *blocked* when it has its parent not visited at time $t(i)$. i has a missingID as its parentID. Two tours T_a, T_b are in *deadlock* status when there exist $i \in T_a, j \in T_b$ whose missingIDs are after i, j in the other tour: missingID(i) in T_b after j , missingID(j) in T_a after i . A deadlock tour is invalid because both robots cannot proceed. Figure 5.4 illustrates the two invalid cases. Figure 5.4(a) shows the robot arrives at a SP that requires a not yet deployed relay. Figure 5.4(b) shows robot a needs the relay to be deployed by b and vice versa.

The basic idea is to traverse T_a, T_b again and one tour is blocked when the needed relay is not yet visited in the other tour. At each position i , the tour time t is updated not only from the cost matrix c_{ij} , but also considers the waiting time for i 's parent: update t as the maximum of its parent's visited time and itself. Note that when a parent is visited, all ancestors must have been visited, because an unvisited parent will immediately block its children. In addition, when self-flipped and deadlock cases occur, we mark the tours invalid and set the tour time to be a very large number, i.e., FLT_MAX, making the tour not selected in its parent function. The detailed procedure is shown in Alg. 17.

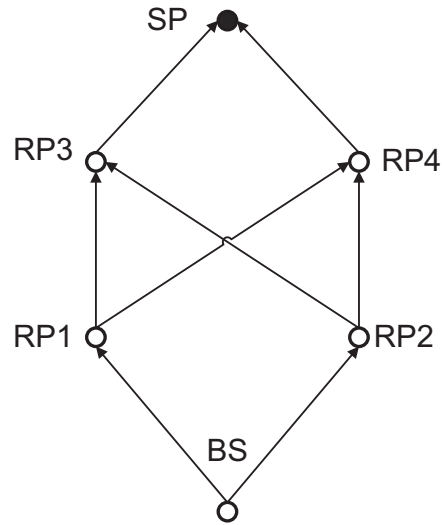
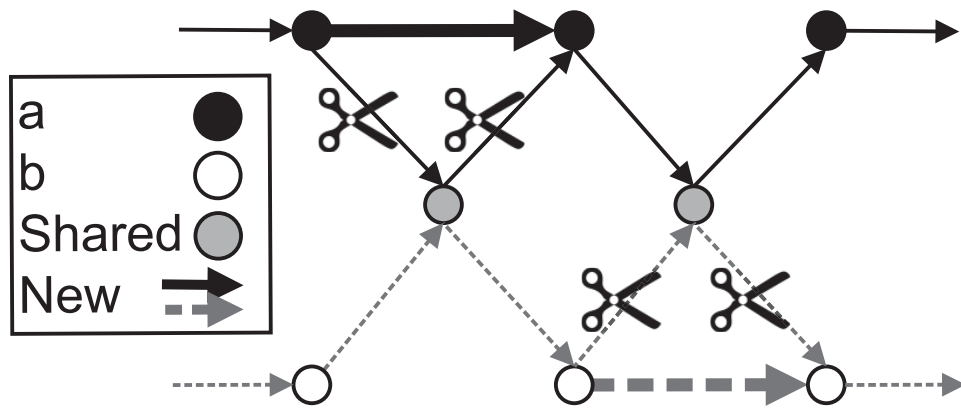
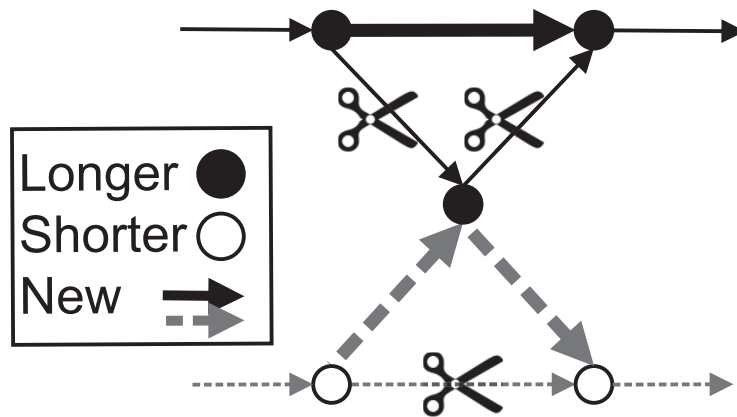


Figure 5.2. Multiple choices of precedence constraints.

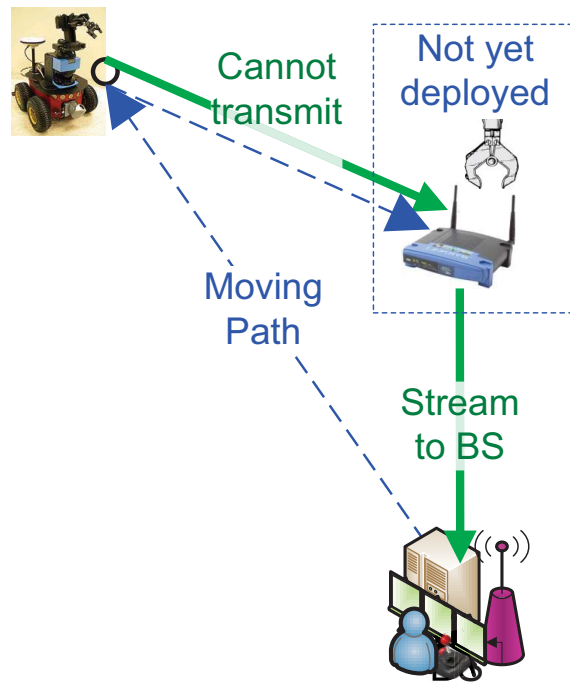


(a) Shared relay pruning.

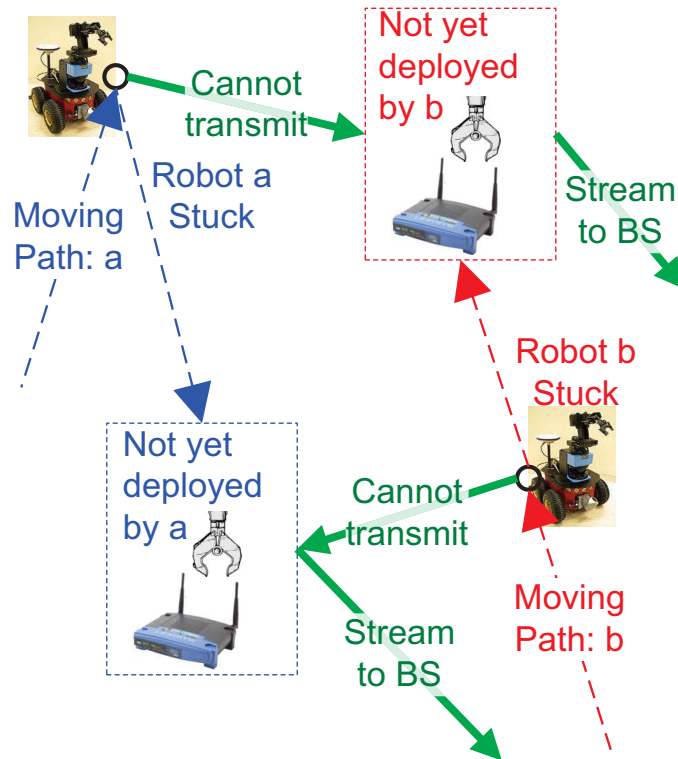


(b) Vertex transfer.

Figure 5.3. Shared relay pruning and vertex transfer.



(a) Self-flipped.



(b) Deadlock.

Figure 5.4. Invalid cases.

Balance with vertex transfer

While pruning is only effective on shared relays, vertex transfer attempts on any node and further balances tours to reduce t_{ab} . The basic idea is to try transferring a position on the longer tour and inserting it into the shorter one. We greedily choose the best pair of removal and insertion position until no further reduction is found. For each pair of newly modified tours, Alg. 17 is called to add waiting time when it is applicable. The detailed procedure is presented in Alg. 18. As illustrated in Figure 5.3(b), the black visiting position in the middle is transferred to the shorter white one,

where black positions are those on the longer tour and the white ones are those on the shorter tour.

Algorithm 18: Balance with vertex transfer

Input: T_a, T_b .

Output: updated T_a, T_b

```

1 repeat
2   Mark the longer and shorter tours of  $T_a, T_b$  as  $T_l, T_s$ .
3   foreach  $i \in T_l$  do
4     foreach  $j \in T_s, j \neq i$  do
5       Get new tours  $T'_l \leftarrow (T_l \text{ with removed } i), T'_s \leftarrow (T_s \text{ with inserted } i \text{ after position}$ 
        of  $j)$ .
6       Call Alg. 17 to validate tours and check dependency. Obtain the true time  $t'_{ls}$  by
        adding wait time (if it exists) on  $T'_l, T'_s$ .
7   Get  $\min t'_{ab\_min}$  from all  $t'_{ab}$ , record corresponding  $i'_{min}, j'_{min}$ 
8   if  $t'_{ab} < t_{ab}$  then executes transfer: Remove  $i'_{min}$  from  $T_l$  and insert  $i'_{min}$  at position
        after  $j'_{min}$  in  $T_s$ . Update  $T_a, T_b$ .
9 until there is no improvement:  $t'_{ab} \geq t_{ab}$ 

```

5.6 Extensions

The previous sections provide the basic procedures to describe how STARS solves the 2RSRD problem. In this section, we present several extensions to STARS, such as (1) dealing with more than two-robot scenarios, (2) placing static sensors to enable constant area monitoring, (3) considering a non-uniform search and transmit interval and relay deploying time, (4) leveraging pre-

existing relays, (5) leveraging extra carried relays to reduce dependency. These extensions make STARS adaptive to more general conditions and varying user requirements.

5.6.1 Search with a single or more than two robots

STARS trivially supports a single robot search since tour generation is simplified as PCATSP. In STARS, the solution to the first subproblem is irrelevant to the number of robots and is always valid. Therefore, STARS is extensible to search with more than two robots as long as PC2TSP is extensible to deal with more than two travelers.

Theorem 7. *Our solution to PC2TSP is extensible to deal with more than two travelers.*

Proof. We first illustrate the changes with $n=3$ robots of R_0, R_1, R_2 and then extend it with $n > 3$.

The sensing positions are clustered by $3 - 1 = 2$ vertical lines $X = \mathbb{R}_x/3, X = 2 \cdot \mathbb{R}_x/3$ to evenly partition the region. Pruning is first conducted the same as before to relays only shared by two robots $SR_{01}, SR_{02}, SR_{12}$. The shared relays of all 3 robots SR_{012} , if they exist, are then partitioned and evaluated by a recursive powerset computation. In tour validation, deadlock checking needs to consider the additional cases with 3-robot interdependency. Finally, vertex transfer between each robot pair $(0,1),(0,2),(1,2)$ is executed until the search time cannot be reduced any further.

Cases with $n > 3$ are similarly extended. \mathbb{SP} are clustered by $n - 1$ lines. The shared relays are evaluated with $\sum_{i=2}^n \binom{n}{i} \cdot 2^\lambda$ different cases, where λ is the number of shared relays between that set of robots. Up to n interdependencies are checked for deadlocks. Besides, vertex transfer between $\binom{n}{2}$ pairs of robots is evaluated. □

When $n > 5$, the procedure of shared relay pruning may be simplified due to high computation

cost. However, with the goal of replacing robots by relays and the strategy of partitioning areas, we believe $n \leq 5$ is sufficient in most cases.

5.6.2 Enable constant monitoring with static sensor deployment

Search paths and relay deployment positions are computed offline before search starts. During the search process, additional sensors may be deployed at $SP_{susp} \subseteq \mathbb{SP}$ to cover suspicious areas when they are identified during the search and transmission interval (STI), when human and/or computational analysis is conduct on the video streams at the base station. With the communication backbone to cover \mathbb{SP} , the sensor deployed at any sensing position has a valid path back to the BS. Such sensor deployment significantly reduces the number of sensors compared to a full sensing coverage deployment. In addition, it adds on a constant monitoring on suspicious areas, compared to the single-time area scan and pass by the homogeneous mobile robots.

5.6.3 Consider a non-uniform search and transmit interval and relay deploying time

We may define a non-uniform STI time $t_{STI}(i)$ for each $i \in \mathbb{SP}$ and a relay deploying time $t_{rd}(j)$ for each $j \in \mathbb{RP}$. The time only affects the cost matrix $c_{ij} \in \mathbb{C}$ in Eq. 5.10 for each edge ij . Besides the current value based on obstacle aware traverse path from i to j , any edge with a destination vertex of $j \in \mathbb{RP}$ will add the value of $t_{rd}(j)$, $c_{ij}+ = t_{rd}(j)$, to consider the extra time of relay deployment. To take into account of the time for search and transmission interval, any edge with a starting vertex of $i \in \mathbb{SP}$ will add the value of $t_{STI}(i)$, $c_{ij}+ = t_{STI}(i)$.

5.6.4 Leverage pre-existing relays

The applications of the multi-robot search and monitoring include rescue and reconnaissance missions in radioactive, earthquake, and battlefield environments. Therefore, pre-deployed relays and infrastructure networks are often not available. However, STARS can take advantage of the pre-existing relays gracefully with a simple modification. The basic idea is to use them to cover the nearby sensing positions. Recall the Algorithm 14 in Section 5.4.2. Given the set of pre-existing relays \mathbb{R}_{exist} , we may first use them to dominate (cover) their neighboring sensing positions. Then we call the DS finder to dominate the rest of the uncovered sensing positions. A union of \mathbb{R}_{exist} and the DS by the DS finder is the final dominating set for all sensing positions. The rest of procedure remains the same.

5.6.5 Leverage extra carried relays to reduce dependency

Recall the Problem 1b in Section 5.3.2, we assume our attempt to minimize relay number by S-CDS will satisfy the robot carrying capacity constraint. It is also possible that each robot's relay carrying capacity n_r is larger than the computed relay number. Under such a condition, we may take advantage of the remaining carried relays to reduce the search time when dependency and waiting time between robots exist. The basic idea is as follows. On robot a's travel path, check the sensing positions relying on the relays deployed by robot b (or vice versa). When the dependency occurs and the waiting time is not zero, we attempt to deploy additional relays on robot a's own path to "skip" such relays and remove the dependency. This approach is mostly effective when the relay deploying time (added new cost) is notably smaller than the saved waiting time. Alg. 19

describes the algorithm in detail.

Algorithm 19: Leverage extra relays to reduce dependency and waiting time.

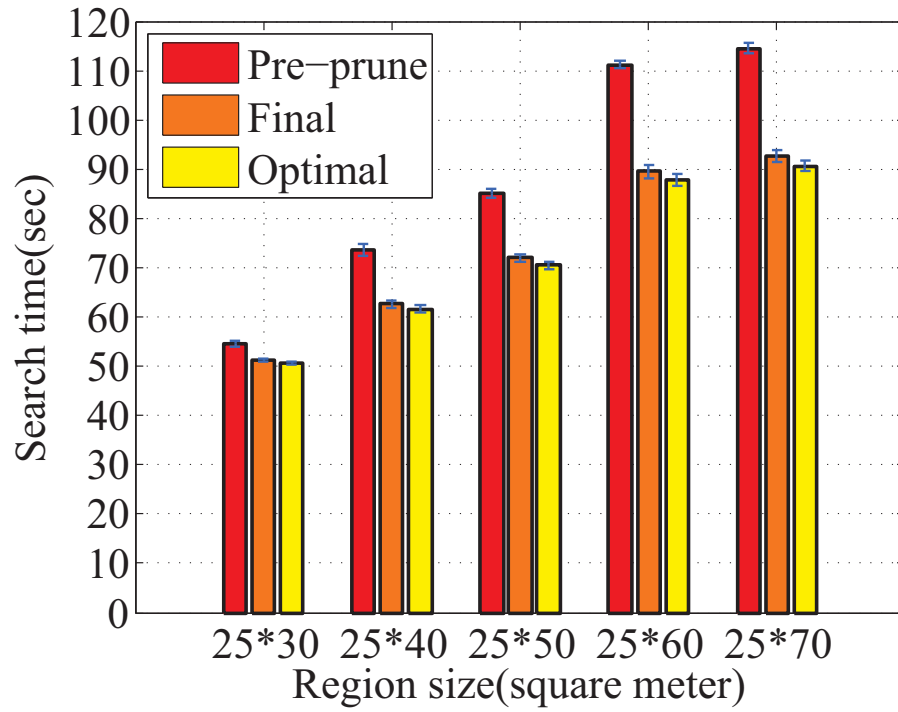
Input: Robots migration paths and k_i number of extra relays for each robot i .

Output: Enhanced traveled paths.

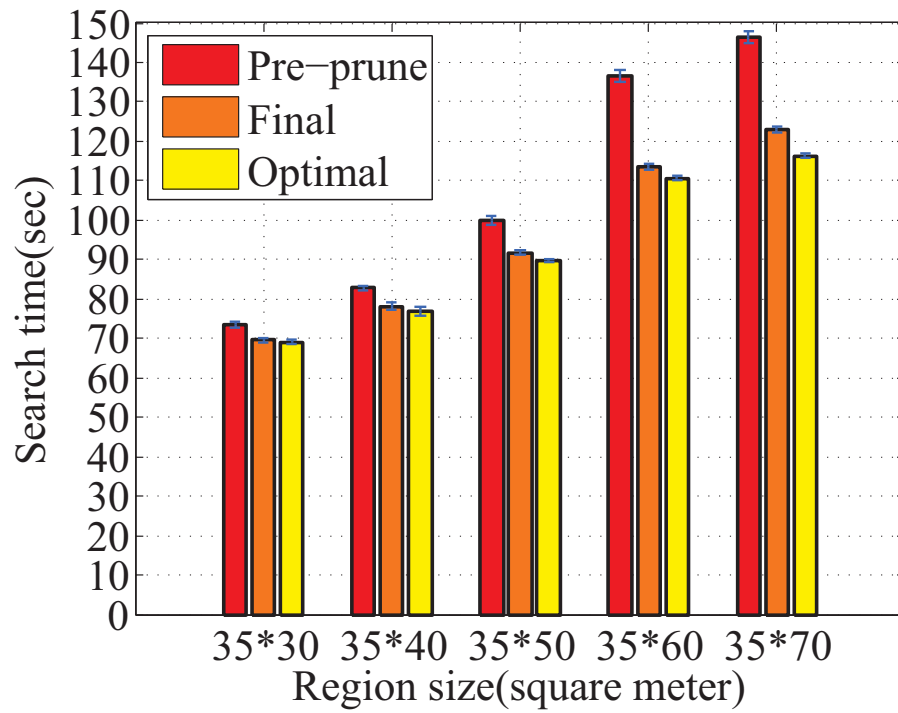
- 1 Compute the traveling paths and search time for both robots using the current algorithms.
 - 2 Obtain all the sensing positions \mathbb{SP}_{wait} relying on the relays deployed by the other robot \mathbb{R}_{other} with a non-zero waiting time.
 - 3 **foreach** $sp \in \mathbb{SP}_{wait}$ and its corresponding $r_o \in \mathbb{R}_{other}$ **do**
 - 4 *// Check whether the k extra relays are sufficient to build a path back to the BS.*
 - 4 Call breadth-first search from sp to find the minimum number of hops on robot's sensing positions to skip r_o to reach the BS.
 - 5 Evaluate the *time-saving effectiveness* by the average saved waiting time per extra relay.
 - 6 Greedily adding the extra relays in the order of maximum time-saving effectiveness until extra relays run out or no extra relays can reduce the search time.
-

5.7 Performance Evaluation

We use a C++ simulation program to evaluate STARS search efficiency. PCATSP is computed in part by IBM ILOG CPLEX Optimizer V12.2. The region is represented by the grid-map model with $X \times Y$ cells. Cell edge length is 1 meter, and the BS is located at the bottom center of the region. The robots travel at a constant speed of 1m/sec. $r_s=10m$ and $r_c=20m$. Obstacle cells are generated randomly each time with an obstacle ratio r_o as the parameter. All tests are run 200 times to obtain the average. The error bars in the figures show the 95% confidence interval.

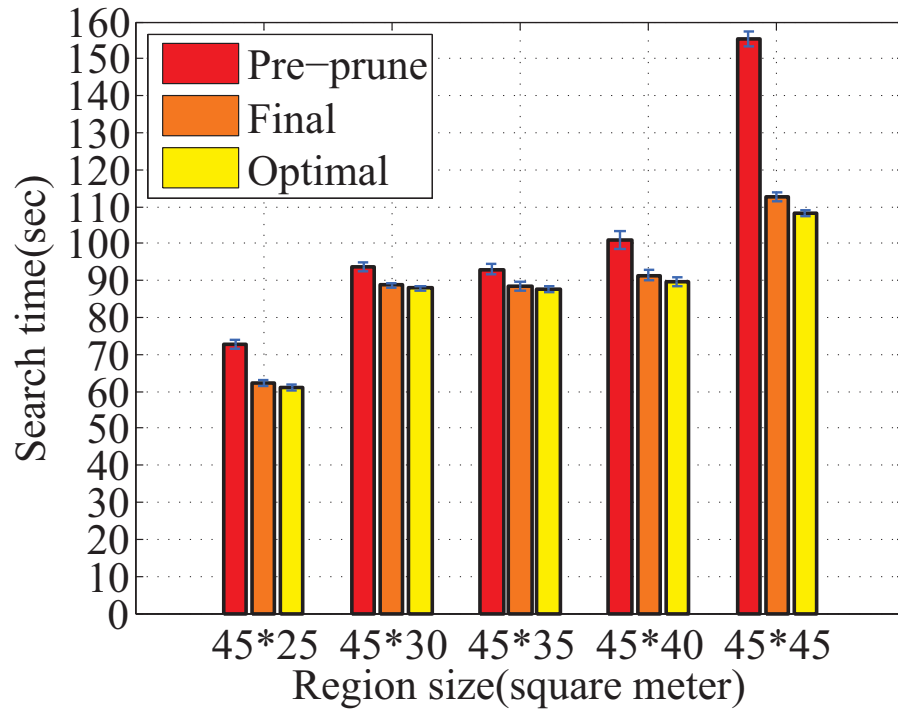


(a) Extra-narrow region.

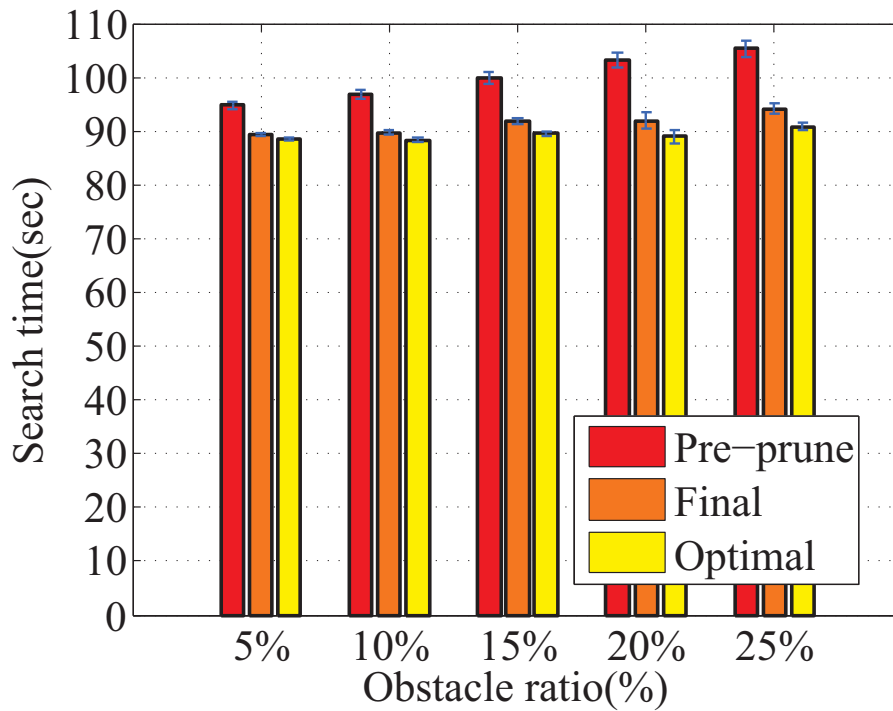


(b) Narrow region.

Figure 5.5. Compare the search time with optimal under varying width regions (widths: 25-35m).

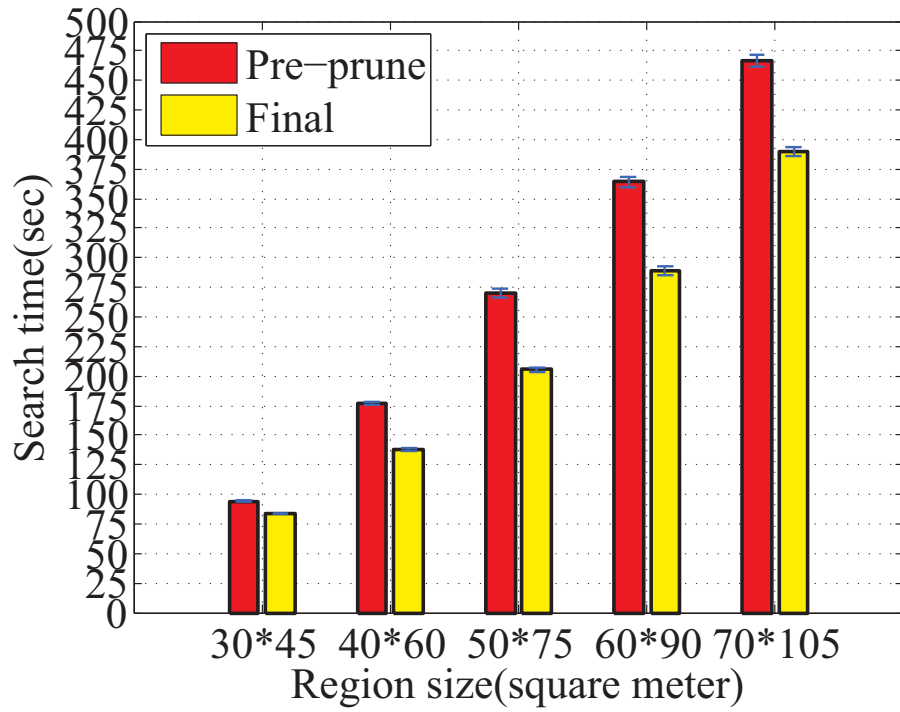


(a) Wide region.

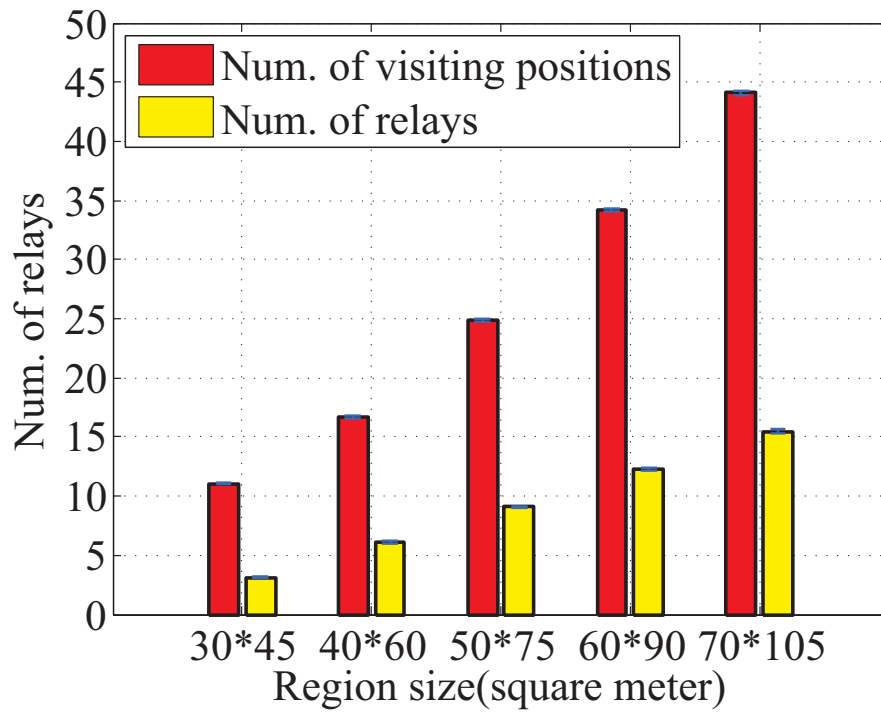


(b) Varied obstacle density.

Figure 5.6. (a) Compare the search time with optimal under varying width regions (widths: 45m). (b) Varied obstacle density from 0% to 25% for 35*50 case.

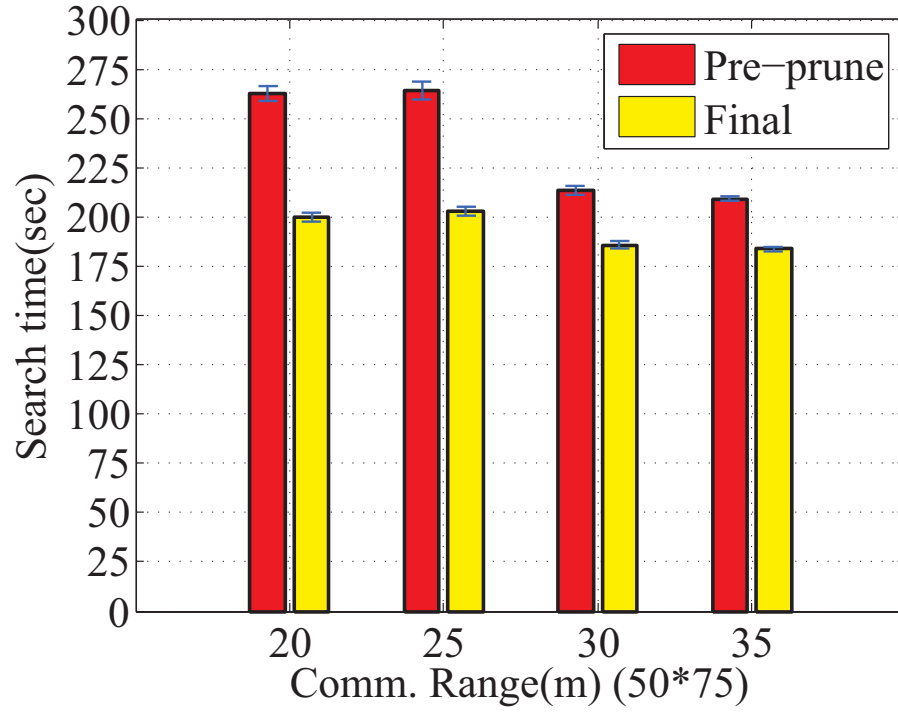


(a) Search time in larger regions.

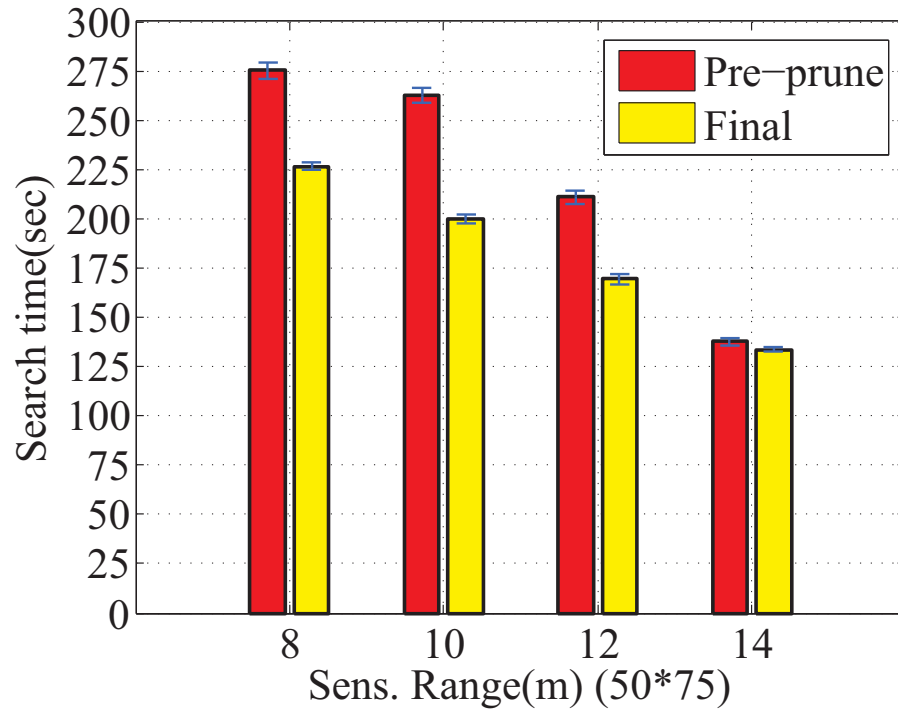


(b) Number of visiting positions and relays.

Figure 5.7. (a) Search time in larger regions. (b) Number of visiting positions and relays.

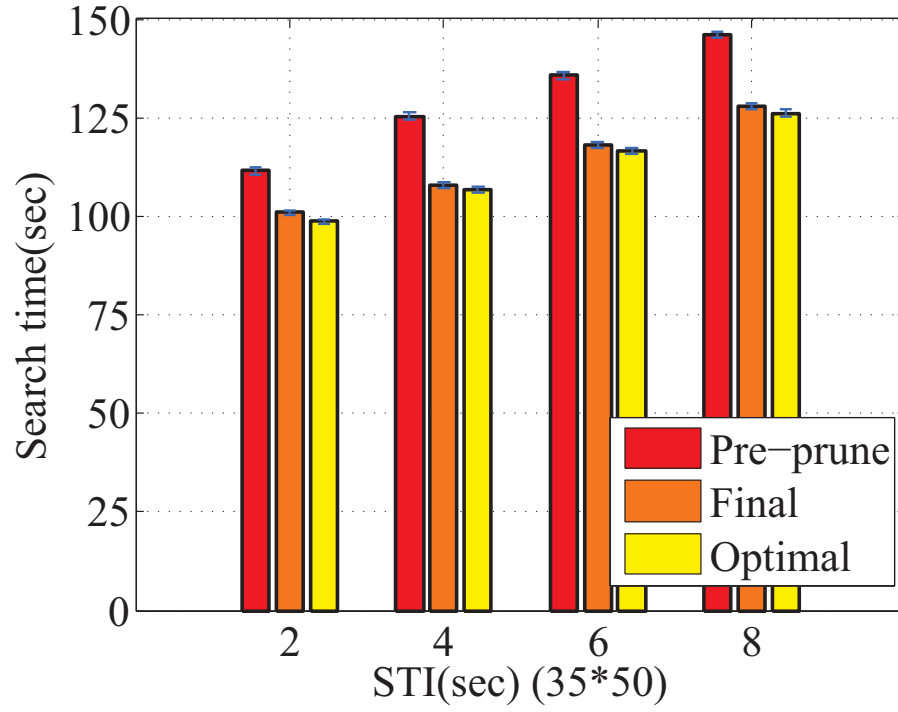


(a) Varying communication ranges. $r_s=10m$.

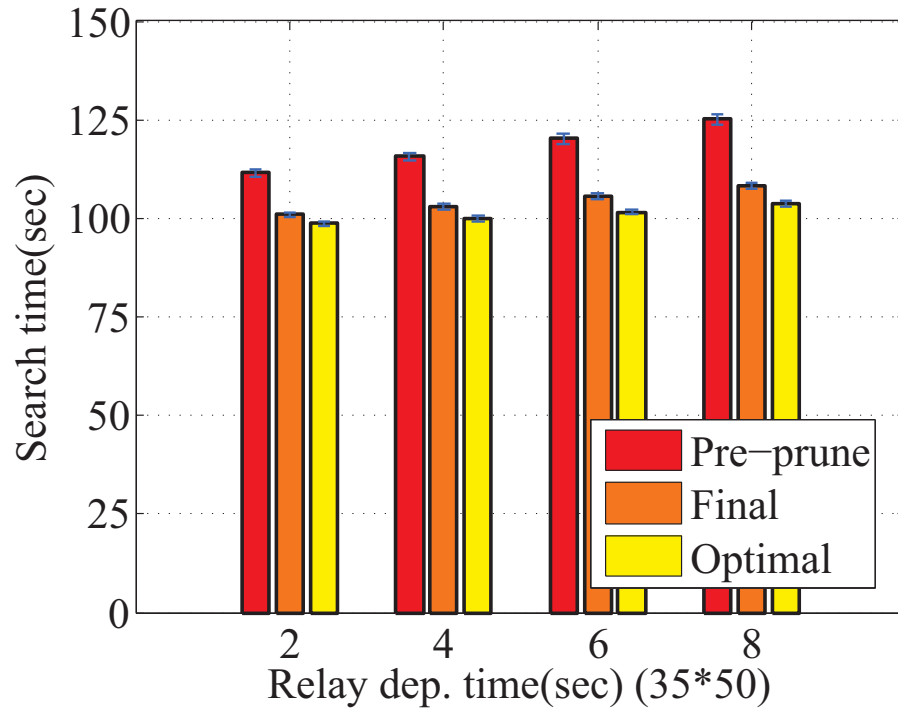


(b) Varying sensing ranges. $r_c=20m$.

Figure 5.8. Compare the search time with varying communication ranges and sensing ranges (50m*75m region).



(a) Varying STI time. $t_{rd}=2\text{sec}$.



(b) Varying relay deploying time. $t_{STI}=2\text{sec}$.

Figure 5.9. Compare the search time with optimal under varying STI and relay deploying time.

First, we compare the search time with the optimal result in different shapes of regions with widths of 25m, 35m, and 45m respectively as shown in Figures 5.5(a), 5.5(b), 5.6(a) ($r_o=0.15$). The optimal result generator tries all possible tours with the same visiting positions and precedence constraint as STARS. Due to the high computation cost of the optimal result, we limit the region size where up to 18 visiting positions need to be traversed. The average difference of our heuristic from optimal is 1.94%, 1.98%, and 1.91% for the above 3 cases. Also, tour pruning and balance effectively reduce search time by 25% on average.

Figure 5.6(b) shows the 30*50 grid cell case with varying densities of obstacles ($r_o=\{0.05, 0.1, 0.15, 0.2, 0.25\}$). As r_o increases from 5% to 25%, the optimal tour time increases from 88.6 to 90.8 seconds (2.4%). The difference of our heuristic from optimal increases slightly from 0.8% to 3.7%. We believe the near optimal results are in no small part contributed by (1) the optimal single robot tour by linear programming, (2) the effective post-pruning: reducing shared relays and vertex transfer.

We focus on narrow regions because a clear separation of two robots is then challenging. Also, the dependency between robots is likely to occur; the two robots tours are likely to be mixed together rather than clearly separated. (Recall that the robots start searching from BS at the bottom center of the region.) In wider regions, it is possible to assign each robot a sub-area to search, converting the problem into a simpler single robot search problem.

Second, we show the search time trend with larger regions of 30*45, 40*60, 50*75, 60*90, 70*105 grid cells in Figure 5.7(a) where optimal results are too slow to be obtained. Results show the two robot tours are well-balanced. The differences between tours are only 6.8% 7.4% 4.1% 3.1% 2.2% respectively, as compared to the longer tour. Well-balanced tours do not guarantee near optimal performance but most optimal tours we observed are well-balanced. Figure 5.7(b) shows

the number of visiting positions and relays in the above cases. The ratio of relays to visiting positions is 28.1%, 36.5%, 36.4%, 36.0%, and 35.1% respectively, which shows that approximately one third of the positions need to be deployed relays.

Third, we investigate on the impact of varying communication ranges and sensing ranges on the search time. Figure 5.8(a) demonstrates the search time modestly reduces 8.15% from 199.9 to 183.6 seconds with an increased communication range from 20m to 35m, in 50m*75m regions. The sensing range is set at 10m and the obstacle ratio is 15%. At the same time, the number of relays notably decreases 65.4% from 8.82 to 3.05. These results first show that an increased communication range has a direct impact on the number of relays. However, a reduced number of relays does not necessarily reduce the final search time. The reason is that the removed relays may be exactly on the robot traveling paths. Therefore, whether or not to have these relays does not affect the search time (assuming a zero relay deploying time).

Figure 5.8(b) shows the search time notably reduces 41.0% from 226.5 to 133.6 seconds with an increased sensing range from 8m to 14m, in 50m*75m regions. The communication range is set at 20m and the obstacle ratio is 15%. While an increased communication range may only remove the relays that are already on the traveling paths, reduced sensing ranges will affect all sensing positions and therefore more notably change the final tours. We did not compare the varied sensing and communication range results with optimal because the node number exceeds 18 under many cases, and the exponential computation for optimal is then formidable.

In the above evaluation, we assume a zero second search and transmit interval(STI) at the sensing positions and a zero second relay deploying time at relay positions. Now, we evaluate the impact of varying STI and relay deploying time on search time. Figure 5.9(a) shows the search time increases 21.1% from 100.9 to 127.9 seconds when STI increases from 2 to 8 seconds in the

35m*50m regions. A constant relay deploying time is set at 2 seconds. The search time difference of STARS from optimal ranges from 1.25% to 2.23% in Figure 5.9(a).

In addition, Figure 5.9(b) shows the search time increases 6.83% from 100.9 to 108.3 seconds when relay deploying time increases from 2 to 8 seconds with a constant STI of 2 seconds. The search time difference of STARS from optimal ranges from 2.23% to 4.42% in Figure 5.9(b). We found search time increases more notably with increased STI compared to relay deploying time since there are more sensing positions than relay positions. We also found the our final search time is less influenced by the increased STI or relay deploying time, compared to the pre-pruned values.

5.8 Summary

In this chapter, we first present a problem called precedence constrained two traveling salesman (PC2TSP). A near-optimal heuristic to PC2TSP is presented with 1.94% average difference from optimal, which is in part contributed by optimal single tour computation, dependency handling, and effective pruning and balance. Leveraging PC2TSP, we propose a real-time search and monitoring scheme called STARS, which builds on a heterogeneous and cost-effective platform of mobile robots, static relays and sensors. Mobility of robots enables relay deployment on the fly to form a communication backbone at carefully selected locations. The backbone facilitates the robots to transmit streams to the base station for remote sensing and control. STARS significantly reduces cost by replacing mobile robots by static relays and enables constant area monitoring by sensor deployment only at specific suspicious areas rather than a full coverage. In the next chapter, we will introduce an exploration strategy with online relay deployment when the environment is *unknown*.

CHAPTER 6

Steiner Traveler: Relay Deployment for Remote Sensing in Heterogeneous Multi-Robot Exploration

6.1 Preliminaries and Motivations

The previous chapter presents STARS, or the *Static Relay aided Search* to support remote sensing in a *known* environment when all robots may deploy relays. In this chapter, we will present a new heterogeneous exploration model based on the online relay deployment for an *unknown* environment with a single or a partial team of robots that may deploy relays. We call the approach *Bandwidth-aware Exploration with a Steiner Traveler* (**BEST**). BEST has a heterogeneous robot team with a fixed number of frontier nodes (\mathbb{FN}) to sense the area iteratively. In addition, a relay-deployment node (RDN) “chases” the (\mathbb{FN}) movement and places relays when necessary to support the video/audio streams aggregation to the base station. This exploration model is illus-

trated in Fig. 6.1. We show BEST is extensible with multiple RDNs in Section 6.6.

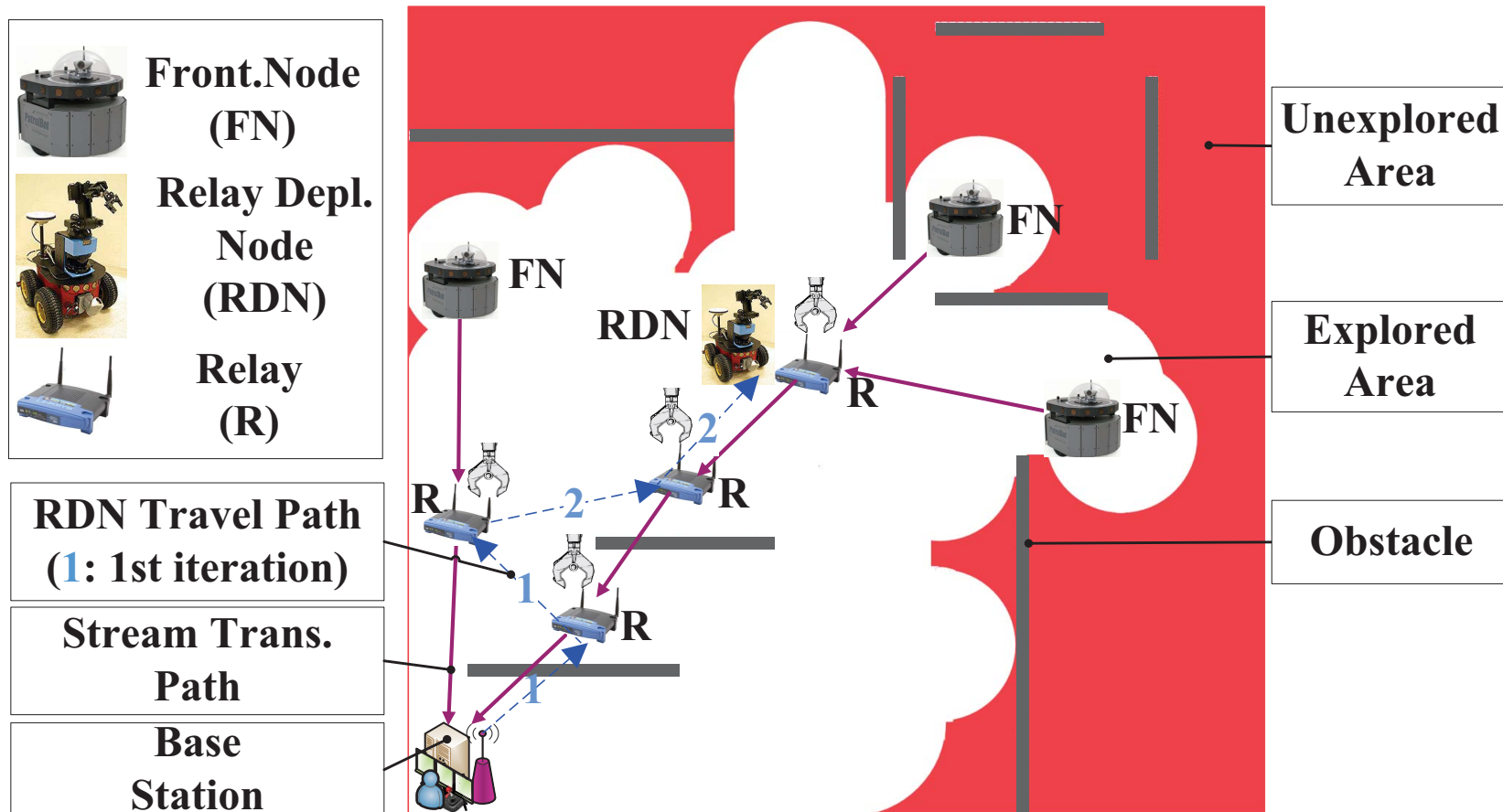


Figure 6.1. A relay deploying robot supports the stream aggregation for remote control of frontier nodes in exploration.

This approach is motivated by the following four reasons. First, as mentioned in Chapter 1, prior relay deployment is often infeasible when the environment is unknown and relays have to be placed at the same time as the exploration. Second, the FN become “worry-free” of the relay deployment with the clear separation of relay and search robot. Therefore, it is flexible for the FN to concentrate on exploration and use any existing exploration scheme.

Third, we may markedly save cost and improve efficiency compared to the homogeneous robot approach where all the robots have to be equipped with relay deployment devices or move backwards from frontier areas to merely serve as relays for the streams. Fourth, BEST does not have the drawback of hard limit on exploration range comparing to a homogeneous robot team. When a homogeneous robot team tries to explore farther away and maintain connectivity to the BS, more robots have to retreat from frontier areas to serve as relays. With a fixed number of robots, increasing number of relay robots will finally prevent the frontier robots from moving forward. In BEST, on the other hand, a constant number of FN are not restrained from the increased distance from the BS.

The key problem is to compute the minimum path for the RDN to travel and find a minimum number of positions to deploy necessary relays to support the stream aggregation in each movement iteration. This problem inherits characteristics of both the traveling salesman and *Steiner Minimum Tree with Minimum Steiner Points (SMT-MSP)* [43] problems. We model the novel problem as the minimum velocity *Flow constrained Steiner Traveler* problem (**FST**),

FST is described as follows: Given (1) a fixed number of FN to explore an unknown environment in a synchronized iterative (round by round) fashion with a traveling time and a constant velocity, (2) a relay deploying node (RDN) with a relay deploying time t_d , (3) a transmission constraint that when all FN arrive at their destinations in each round, a transmission path formed

by relays from each FN to the BS always exists, (This also implies a precedence constraint that needed relays must be deployed by RDN before FN arrive), (4) a flow constraint that the number of flows for each relay never exceeds an upper limit K , to find the traveling path for the RDN and the positions to deploy the relays such that the average traveling velocity for the RDN is minimized.

We minimize the velocity in order to reduce the energy cost. Because velocity is related to both traveling path length and the traveling time, we need to (1) reduce the number of relays for deployment time to increase the RDN travel time; (2) reduce the traveling path length. Therefore, a joint consideration on the number of Steiner points and the traveling paths is needed in this problem.

6.1.1 Key Contributions

- To the best of our knowledge, BEST is the first of its kind to (1) jointly consider traveling salesman with flow-constrained Steiner Tree to achieve bandwidth sufficiency and (2) present a heterogenous exploration scheme with relay deployment for unknown environment achieving significant efficiency improvement.
- To solve R^2BS , we formulate the problem as the minimum velocity *Flow constrained Steiner Traveler* problem (FST). We present an efficient heuristic that takes advantage of existing relays that have unsaturated paths to the BS to reduce number of relays.
- Designing exploration schemes for an *unknown* environment, we also wonder what will be the impact of a *known* global map on performance? Considering the two metrics (1) RDN traveling path length and (2) the number of deployed relays, we find the traveling path length notably benefits from a global map while the number of deployed relays only has marginal

improvement.

The rest of the chapter proceeds as follows. Section 6.2 introduces the system model. In Section 6.3, we formulate the problem and discuss the challenges. Section 6.4 presents the BEST scheme. Performance evaluation is in Section 6.5. Section 6.6 discusses the future work and Section 6.7 concludes the chapter.

6.2 System Model

Robot Model: A robot team includes $|\mathbb{FN}|$ number of \mathbb{FN} and a single RDN. In future work, we will consider the extension of multiple RDNs, which is discussed in Section 6.6.1. \mathbb{FN} have traveling, communication, and sensing capability while the RDN can travel, communicate and deploy static relays with a maximum carrying capacity and a deploying time t_d . We assume BEST satisfies this carrying capacity constraint since it attempts to use a minimum number of relays. Both \mathbb{FN} and RDN have a communication range r_c and a sensing range r_s where $r_c \geq r_s$. \mathbb{FN} move at a constant velocity v_{fn} in all iterations and RDN moves at a changing velocity v_{rdn}^t in each iteration t . The acceleration and deceleration time is not considered.

Environment Model: A 2D occupancy map is adopted to model the environment as an unknown rectangle region $\mathbb{R} = X \times Y$ with grid cells similar to that in [52]. The cell size equals to the largest robot size. A cell status include unexplored, exploring, and explored. An explored cell is marked as either free or obstructed. An obstacle cell blocks the traveling path. Such cells cannot be visited by the robots or placed relays. We assume the obstacle distribution in the environment is unknown. A *base station* is the control center where the human operator remotely monitor and potentially operate the robots when necessary.

Iterative Exploration Model: A fixed number of \mathbb{FN} explores an unknown environment in a synchronized and iterative (round by round) fashion with a traveling time T_{fn}^t in iteration t . After all \mathbb{FN} arrive at their target positions, \mathbb{FN} and also RDN will receive \mathbb{FN} 's next iteration target positions and then enter a fixed length sensing and transmitting interval (STI) T_{sti} to sense the area and mark cells as searched. RDN must finish deploying all necessary relay in round t (denoted as \mathbb{R}^t) before STI starts because the relays are essential to support the transmission. Since the RDN receives \mathbb{FN} ' target positions before last round STI, the total time for RDN to travel T_{rdn}^t in round t is:

$$T_{rdn}^t = T_{fn}^t + T_{sti} - |\mathbb{R}^t| \cdot T_d \quad (6.1)$$

We assume $T_{fn}^t + T_{sti} - |\mathbb{R}^t| \cdot T_d > 0$. We may find that T_{rdn}^t is increased with the decreased number of deployed relays. It is especially significant when the deploying time T_d is lengthy. All \mathbb{FN} and RDN start traveling from the BS. After the RDN completes the relay deployment in each iteration t , it will stop as the last relay deployment position R_{stop}^t ,

Note that BEST is compatible with any iterative FN movement strategies for area exploration or other purposes. FN movements can be controlled by a computer program or a human operator at the BS. Under either case, \mathbb{FN} can only move to sensed (known) areas because of the unknown obstacles in unknown areas. In BEST, we apply the frontier cell based strategy to maximize sensing gain as in [52].

Wireless Model: The unit-disk model where two nodes i, j can communicate as long as $dist(i, j) \leq r_c$. $dist(i, j)$ gives the distance from i to j . We will consider a probability based communication range in future work. Similar to [52], each FN has a uniform flow sending rate, $rate_{send}$ and a link capacity, $rate_{capacity}$. A bandwidth factor K defines how many flows can a

relay maximally carries before exceeding its capacity:

$$K = \lfloor \frac{\text{rate}_{\text{capacity}}}{\text{rate}_{\text{send}}} \rfloor \quad (6.2)$$

In addition, $\text{path}(V)$ gives the transmission (routing) path by a vector of nodes V , where all adjacent pair of nodes have a distance less than r_c . We assume the powerful BS have multiple radios to have sufficient bandwidth for flow aggregation. A similar 15-radio testbed is shown in [103]. The interference between nodes can be resolved by a multi-channel dual radio strategy with proper channel assignment.

6.3 Problem formulation

With the system model the formal formulation of the minimum velocity *Flow constrained Steiner Traveler* problem (**FST**) is as follows: Given

- a vector of \mathbb{FN}^t positions, a FN travel time T_{FN} , a (partially) explored region \mathbb{RG}^t with an obstacle-cell subset in each iteration $t = \{1, \dots, n_{it}\}$ with a total number of iterations n_{it} .
- a constant FN velocity v_{fn} , FN STI time T_{sti} and a RDN relay deploying time t_d ,
- a transmission constraint that when all \mathbb{FN} arrive at their destinations in each round, a transmission path formed by relays from each FN to the BS always exists, (This also implies a precedence constraint such that needed relays must be deployed by RDN before \mathbb{FN} arrive),
- a flow constraint that the number of flows for each relay never exceeds a bandwidth factor K ,
- placement constraint: no relay is placed on obstacles;

To find (in each iteration t)

- new relay positions \mathbb{R}^t (or its subset) in each iteration t , along with (a subset of) existing relays $\mathbb{R}^{1..n-1}$, to form a directed Steiner tree to connect \mathbb{FN} to the BS.
- a RDN Hamiltonian path $path_{rdn}^t$ starting from last round stop position(R_{stop}^{t-1}) to travel each position in \mathbb{R}^t exactly once,

such that the average RDN travel velocity $\overline{v_{rdn}}$ is minimized. $\overline{v_{rdn}}$ depends on both the travel path length and travel time:

$$\overline{v_{rdn}} = \frac{\sum_{t=1}^{n_{it}} path_{rdn}^t}{\sum_{t=1}^{n_{it}} T_{rdn}^t} \quad (6.3)$$

Mathematically, FST is

$$\text{Minimize } \overline{v_{rdn}}, \quad (6.4)$$

$$\text{S.T. } \exists path(V), V = (i, v_1, \dots, v_n, BS), \forall i \in \mathbb{FN}, v_i \in \mathbb{R}^{1..t}, \quad (6.5)$$

$$dist(v_j, v_{j+1}) \leq r_c, \forall v_j \in V, j \leq n, \quad (6.6)$$

$$\sum_{i \in \mathbb{FN} \cup \mathbb{R}^{1..t}} |f_{ij}| \leq K, \forall j \in \mathbb{R}^{1..t}, \quad (6.7)$$

$$\text{position}(i) \neq \text{obstacle}, \forall i \in \mathbb{R}^{1..t}, \quad (6.8)$$

$$\forall t = 1, \dots, n_{it} \quad (6.9)$$

Where (5) and (6) shows there is a valid routing path formed by existing and newly deployed relays from each FN to the BS via deployed relays under the communication range. (7) gives the bandwidth constraint. (8) gives that the relays cannot be placed at the obstacle cells.

In each iteration t , the computation of the Hamiltonian path for RDN $path_{rdn}^t$ is formulated by the asymmetric traveling salesman problem (TSP). A simple conversion to TSP is achieved by

setting the distances on the arcs from all nodes to position R_{stop}^{t-1} as zero [116]. Then, we apply the integer linear programming (ILP) formulation for asymmetric TSP in [115] as below:

$$\begin{aligned}
& \text{Minimize } \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}, \\
& \text{Subject to } \sum_{i=1, i \neq j}^n x_{ij} = 1, \quad \forall j = 1, \dots, n, \\
& \quad \sum_{j=1, j \neq i}^n x_{ij} = 1, \quad \forall i = 1, \dots, n, \\
& \quad y_{ij} \geq x_{ij}, \quad \forall i, j = 2, \dots, n, \quad i \neq j, \\
& \quad y_{ij} + y_{ji} = 1, \quad \forall i, j = 2, \dots, n, \quad i \neq j, \\
& \quad y_{ij} + y_{jk} + y_{ki} \leq 2, \quad \forall i, j, k = 2, \dots, n, \quad i \neq j \neq k, \\
& \quad y_{ij} = 0, \quad \forall i, j = 2, \dots, n
\end{aligned} \tag{6.10}$$

Where c_{ij} is the obstacle-aware path length computed by A* search from position i to j . x_{ij} and y_{ij} are binary and $x_{ij} = 1$ shows position i is before j immediately in the tour. It is reasonably fast to solve TSP by a branch and cut method with the ILP formulation. Instances with 200 nodes is computed optimally in a couple of minutes [117].

Note that we describe FST using the robotic notations to illustrate our application. A theoretical and general description of FST is made possible with following term changes: (1) \mathbb{FN} and BS become the terminal points where \mathbb{FN} are the sending terminals and BS is the base terminal. (2) Existing and current iteration relays become existing and new Steiner points respectively. (3) $R_{stop}^{t-1} \cup \mathbb{R}^t$ becomes the set of cities where R_{stop}^{t-1} is the starting city for the traveling salesman.

6.3.1 Challenges

There are two major challenges to solve FST. First, FST combines two NP-hard problems: Steiner minimum tree and traveling salesman. The traveling path generation depends on the locations and the number of relays deployed. There is also a trade-off of the number of relays and the traveling path length. To minimize the RDN velocity, both the number of relays and the traveling path need to be reduced. However, it is possible that a relay placement with a least relay number but improper positions will lead to a longer travel path. Besides, the importance of the two factors may vary with different parameter settings. The importance of a shorter path outweighs that of the number of relays with a sufficient lengthy FN traveling time T_{fn}^t and a small deploying time T_d (recall Eq. 6.1 and Eq. 6.3). On the other hand, the relay size reduction is more important than generating a shorter path when T_{fn}^t is small and T_d is large.

Second, the possible usage of existing Steiner points increases the complexity of the problem. Many related work models the minimum relay placement by SMT-MSP [43, 52, 91], Compared to SMT-MSP, a major difference is that when we place relays (new Steiner points) in each iteration t , existing relays may be used. It brings a new problem of whether and how to use them. Note that the existing Steiner points are not modeled as the terminal points because not necessarily all previous relays will be on the transmission paths, especially when the FN positions vary considerably in each iteration.

Terms	Definitions
FN, RDN, R, FN, R, BS	Frontier nodes, relay-deploying node, deployed relays, and base station. FN, R represent FN and relay set. $ \text{FN} $ gives the number of FN.
$\mathbb{R}^{1..t-1}, \mathbb{R}^t, \mathbb{R}^{1..t}$	Existing relays deployed before iteration t , relays deployed in iteration t , and their union.
$f_{i\vec{j}}$	A flow from i to j with the sending rate rate_{send} .
r_c, r_s	Communication range and sensing range.
T_{rdn}^t, T_{fn}^t	The RDN and FN travel time in iteration t .
T_d, T_{sti}	A constant relay deploying time for RDN; A sensing and transmitting interval for FN.
$path_{rdn}^t$	Traveling path by RDN in iteration t .
$v_{fn}, v_{rdn}^t, \bar{v}_{rdn}$	A uniform FN travel velocity, a RDN travel velocity in iteration t , and RDN's average velocity.
n_{it}	Total number of iterations in the iterative movement model.
K in Eq. 6.2	Bandwidth ratio: maximum number of allowed carried flows per relay.
$f_{i\vec{j}}, f_{i\vec{j}} $	The flows on edge from i to j , the number of flows.
$dist(i, j)$	The distance from i to j .
$path\{V\}$	A transmission path by a vector of nodes V .
BRN	The border RN set.

Table 6.1. Notations in BEST.

6.4 BEST: Bandwidth-aware Exploration with a Steiner Traveler

We first present BEST's simpler version when the bandwidth ratio K is sufficient large, then we show how we deal with the extra constraint of bandwidth. Before presenting the solution, we first give the definitions.

Definition 5.1: A flow aggregation point is the point to which sending terminals send flows.

Definition 5.2: A border relay $BR_i^{1..t-1}$ for a FN $i \in \mathbb{FN}$ is an existing relay with the shortest distance to i compared to other existing relays. A BR is *qualified* to be an aggregation point if there is an unsaturated path from the BR to the BS.

6.4.1 BEST with Sufficient Bandwidth

As discussed in Sec. 6.3.1, the major difference of FST from SMT-MSP is the availability of existing Steiner points. Note that effectively using $\mathbb{R}^{1..t-1}$ in the Steiner tree does not add to the “cost”, we design our strategy as follows.

Basic Idea: We cluster \mathbb{FN} to nearby qualified border relay. For each FN cluster, we compare the relay number needed to reach its two potential aggregation points, the border relay and the BS. The border relay is only used when expected to reduce relays. We compute the traveling path afterwards. The detailed algorithm is given in Alg. 20.

We attempt to minimize the deployed relay number and only deploy relay when immediately needed. As a result, there is always a valid path from an existing relay to the BS since it was previously used to send streams. Therefore, all flows sent to the border relays will reach the BS. To

consider the impact of relay placement on traveling path, we place relays as far as possible from the existing ones with the room for adjustment: when distance divided by r_c yields a fractional number.

With \mathbb{R}^t and $path_{rdn}^t$ for each iteration t , the RDN average velocity is then computed.

Algorithm 20: Bandwidth Sufficient Steiner Traveler.

Input: \mathbb{FN} , $\mathbb{R}^{1..t-1}$, R_{stop}^{t-1} in each iteration t

Output: New relays \mathbb{R}^t and RDN traveling path $path_{rdn}^t$

1 For each $i \in \mathbb{FN} \wedge dist(i, BS) > r_c$, add its closest existing RN j to the border RN set

\mathbb{BRN} . Add i to j 's FN cluster \mathbb{FN}_j .

2 **foreach** relay $j \in \mathbb{BRN}$ and its FN cluster \mathbb{FN}_j **do**

3 Call MST-MSP relay placement in [52] to compute relay positions for connecting (1)

\mathbb{FN}_j to j ; (2) \mathbb{FN}_j to BS. Denote their resulting candidate relays as (1) $\mathbb{R}_{brn-temp}$ and

(2) $\mathbb{R}_{bs-temp}$.

4 **if** $|\mathbb{R}_{brn-temp}| \leq |\mathbb{R}_{bs-temp}|$ **then**

5 Add $\mathbb{R}_{brn-temp}$ to new relays \mathbb{R}^t .

6 **else**

7 Add FN cluster \mathbb{FN}_j to FN set \mathbb{FN}_{direct} .

// For FNs not benefit from connecting to existing relays.

8 Call MST-MSP relay placement in [52] to compute relay positions for FN set \mathbb{FN}_{direct} with

BS as the aggregation point. Add resulting relays to \mathbb{R}^t .

9 Compute Hamiltonian path $path_{rdn}^t$ by the ILP formulation in Eq. 6.10 for $R_{stop}^{t-1} \cup \mathbb{R}^t$.

Mark the stopping position R_{stop}^t for next iteration use.

6.4.2 BEST with Constrained Bandwidth

Previous work [52] handles bandwidth constrained relay placement without existing RNs. It guarantees bandwidth adequacy when directly aggregating flows to the BS or the part from \mathbb{FN} to the border relays. The only unchecked part for bandwidth sufficiency is from the border relays to the BS. Hence, before selecting border relays, we need to check whether there are unsaturated paths back to the BS. The problem resembles the maximum flow problem with vertex capacities. The difference are: (1) We not necessarily need to obtain the “maximum” flow but simply need to check the existence of an unsaturated path (which is also called an argument path in maximum flow). (2) Because the relays are placed by our algorithm, we are able to keep track of a path from each relay to the BS. This is different from the EdmondsCKarp algorithm for maximum-flow to use Breadth-first search (BFS) to find the argument path each time.

The major modifications on Alg. 20 to support the constrained bandwidth are as follows.

1. Add structures to save path and flow information. We maintain a vector of *pathToBS* for each relay to save the ordered node list on the path from the relay to the BS. It is updated each time we add new relays to maintain history information.

We first enhance the flow constrained relay placement to save the path from each FN to the aggregation point. When aggregating to border relays, *pathToBS* from BR is appended to the new relays. For example, in Fig. 6.2, the BR’s *pathToBS* is appened to that on R4 and R3.

Besides, another vector of *carriedFNflow* is defined for existing relays. It shows those FN whose flow passes this relay in the current iteration. *carriedFNflow* is reset and cleared at the end of each iteration.

2. Check unsaturated path. Before selecting an existing relay to be a border relay, we check its

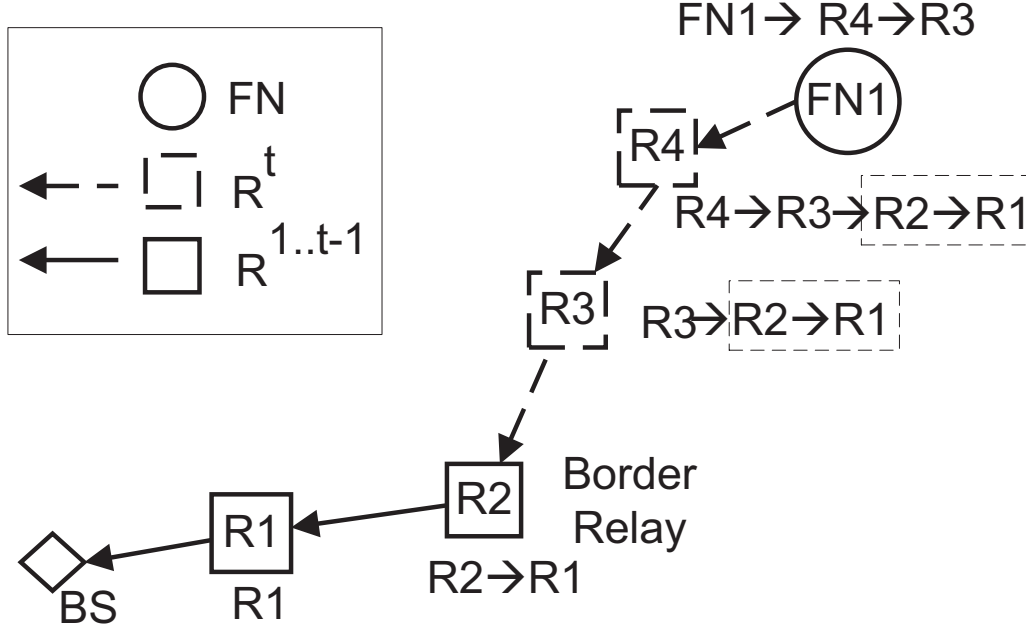


Figure 6.2. Merge border relay's pathToBS to new relays.

qualification by checking whether the path back to the BS is still unsaturated: For each node $i \in$ in its pathToBS, $\text{carriedFNflow}(i).\text{size}()$ is less than K . Unqualified border relays cannot be used as aggregation points. When we run out of qualified border relays, all remaining FN will connect to the BS without using existing relays.

3. Update carried flows. After we verify using BR as the aggregation point reduces the relay number, the carriedFNflow on the pathToBS is updated: For each node $i \in \text{pathToBS}(\text{BR})$, the cluster of FNs for this BR is pushed back to $\text{carriedFNflow}(i)$.

6.5 Performance Evaluation

Summary

- We evaluate the exploration efficiency with varying region sizes, number of robots and ob-

stacle ratios.

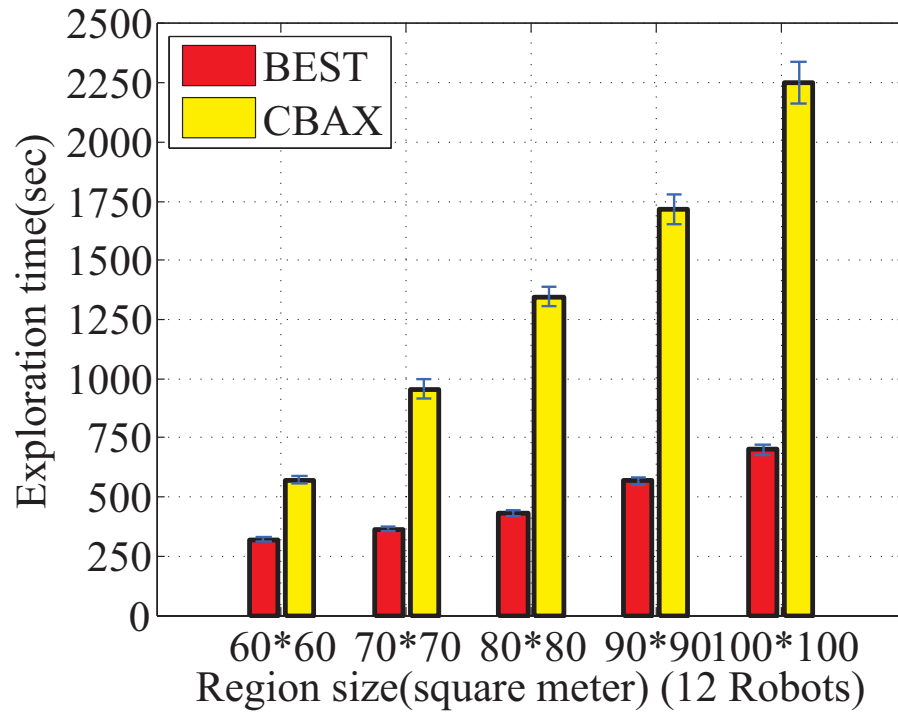
- We also evaluate the performance difference of BEST (given an unknown environment) compared to the solution where a global obstacle distribution is known and thus relay positions and traveling paths can be globally optimized.
- In addition, the impact of bandwidth constraints on number of deployed relays are investigated with respect to different K values.

6.5.1 Simulation Setup and Parameters

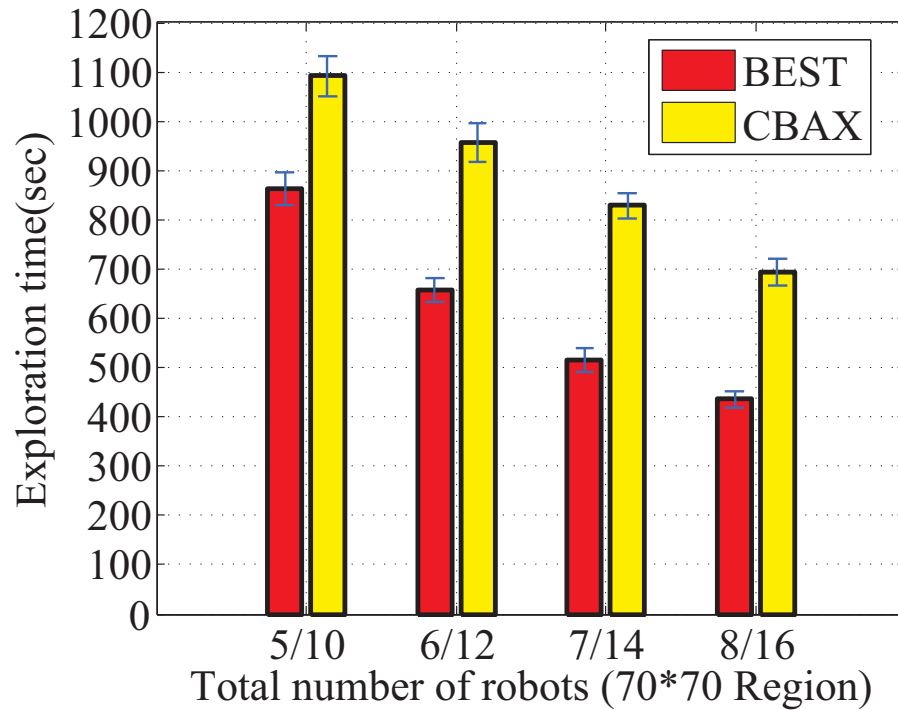
We develop a simulation program in C++ for robot exploration. The traveling salesman problem is solved in part by IBM ILOG CPLEX V12.2. The unknown exploration region is represented by the grid-map model with $X \times Y$ cells. Each cell has an edge length of 1 meter, and the BS is located at the left-bottom center of the region. Obstacles are randomly generated with the ratio (percentage of obstructed cells out of total cells) given by the input. A default bandwidth factor K is set at 3 and obstacle ratio is set at 10%, unless specified otherwise. The communication range r_c is 12m and the sensing range r_s is 6m. The relay deploying time t_d is set as 1 second and FN's T_{sti} is set as 3 seconds. FN has a constant travel velocity of 0.5m/sec. All results are the average obtained by running tests 30 times with different environments of randomly generated obstacles. The 95% confidence interval is shown by the error bars.

6.5.2 Exploration Efficiency

Among the current exploration strategies with connectivity and bandwidth awareness for an *unknown* environment, the connectivity and bandwidth aware exploration (CBAX) [52, 55] uses less

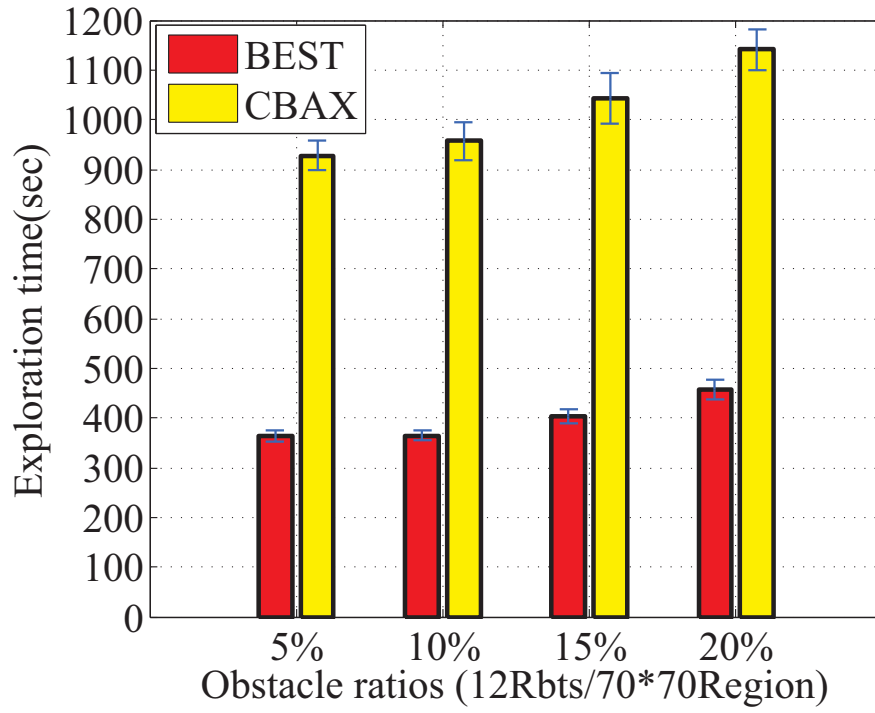


(a) Exploration time in varying region sizes.

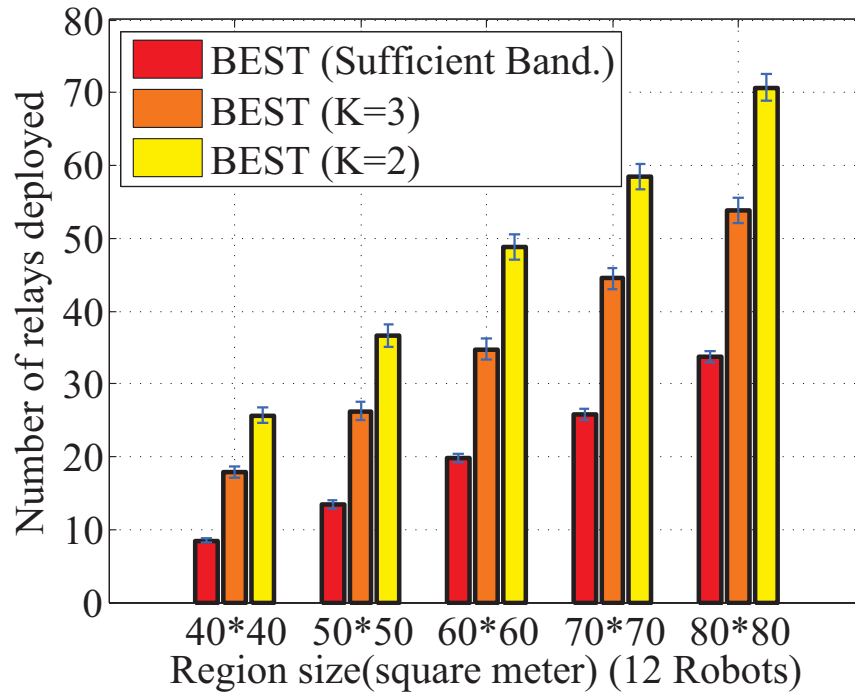


(b) Exploration time in varied number of robots. (BEST uses half the number of robots of CBAX.)

Figure 6.3. Exploration time compared to CBAX with varying (a) region sizes (b) number of robots.

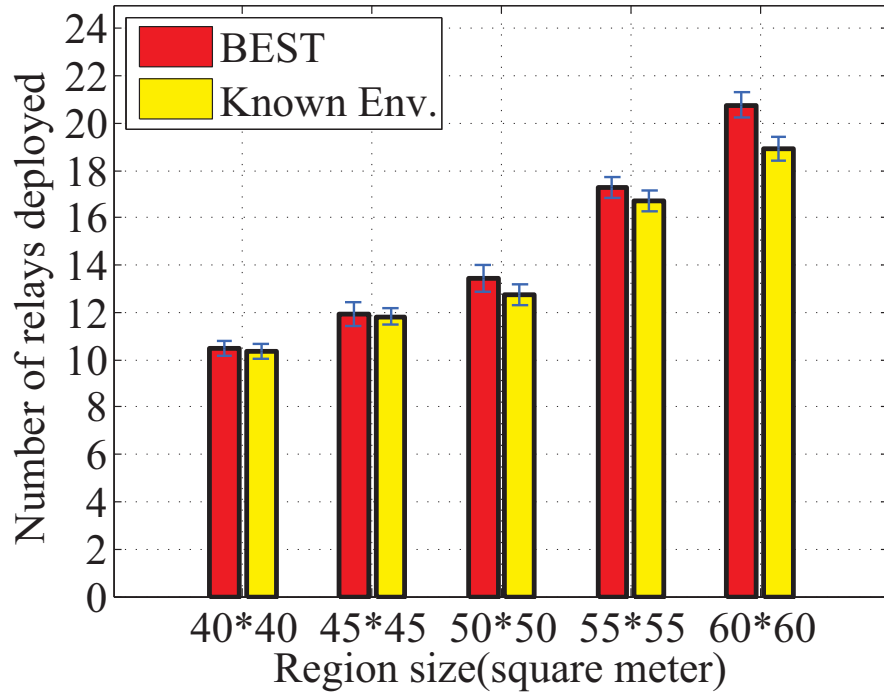


(a) Exploration time with varying obstacle ratios.

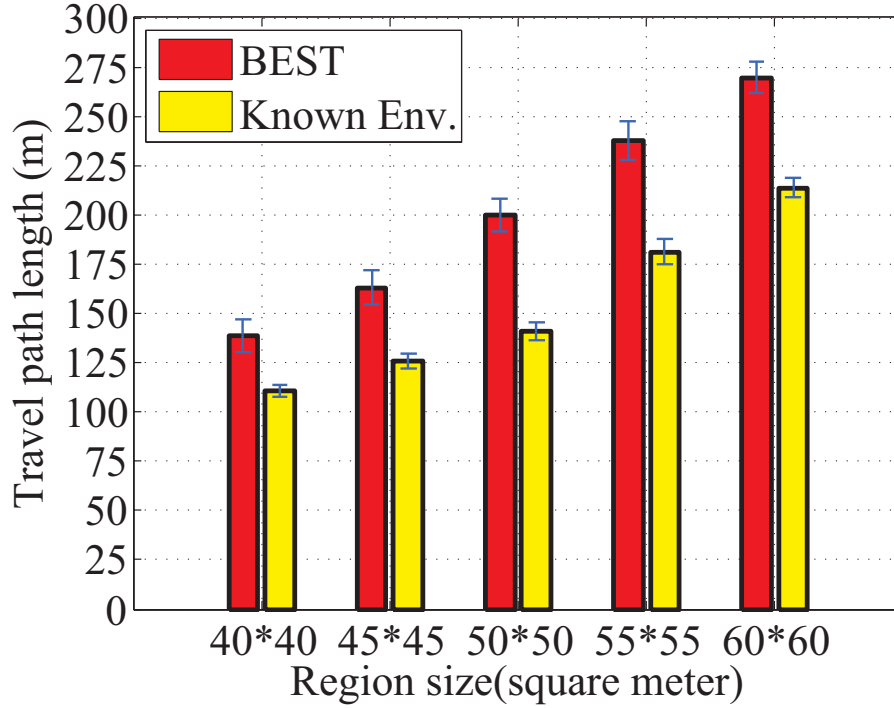


(b) Number of deployed relays with varying bandwidth constraints.

Figure 6.4. (a) Exploration time compared to CBAX with varying obstacle ratios. (b) Impact of varying bandwidth constraints on the number of deployed relays.



(a) Number of deployed relays in BEST compared to a S-CDS based solution in known environment.



(b) Traveled path length by RDN in BEST compared to solution in known environment.

Figure 6.5. (a)-(b): Number of deployed relays and traveled path length by RDN in BEST, compared to those results computed with aid of a known environment.

exploration time compared to strategies in [32, 36], therefore we choose CBAX as the comparison counterpart on exploration efficiency rather than those in [32, 36]. CBAX explores the unknown environment iteratively and dynamic selects and places a minimum subset of robots to be the relays in each iteration while the remaining robots to be frontier nodes, given a fixed total number of robots.

The exploration time is computed from the start to the time when 95% of the non-obstructed region has been searched. Fig. 6.3(a) shows the exploration time with varying region sizes from 60m*60m to 100m*100m. BEST reduces the exploration time by 62.0% on average compared to CBAX given a fixed number of twelve total robots. The exploration time reduction is slightly more significant with larger regions. The reduction increases from 44.1% in 40m*40m regions to 69.0% in 100m*100m regions. The major reason is that more robots move backwards to the BS to serve as relays in CBAX, while BEST uses constant number of robots to explore the area.

The average RDN travel velocity in the five regions is 1.07m/sec, 1.14m/sec, 1.13m/sec, 1.01m/sec, 0.95m/sec respectively, approximately 2 times the FN travel velocity. Applying multiple RDN will help to relieve the workload of a single RDN and therefore decrease the travel velocity.

Besides, Fig. 6.3(b) demonstrates BEST with half the number of robots (e.g., a four-robot team including 1RDN+3FNs in BEST compared to eight node team in CBAX) stills outperforms CBAX in exploration efficiency. In a 70m*70m region, a half number team in BEST remains 24.1% faster than CBAX on average.

In addition, we evaluate the impact of different obstacle ratios on the exploration efficiency. Fig. 6.4(a) shows BEST performances similarly to CBAX with regard to increased ratio of obstacles. An increase from 5% to 20% of obstacle ratio results in a climb of 18.6% for CBAX and a

rise of 20.4% for BEST in exploration time.

6.5.3 Comparing to a Solution with Known Environments

A known environment and obstacle distribution enables optimization of the relay placement and traveling paths under a global view. BEST is designed for exploring an *unknown* environment. Although only a partial (local) map is exposed for the BS to compute the relay positions and traveling paths in each iteration, it is desirably to know the how much we may improve if a global view were provided.

The solution to FST given a known environment is briefly described here. It is a simplified and partial version of the “STARS” solution for a known environment in [118]. Given the obstacle distribution, we compute the sensing positions, relay positions and traveling paths by modeling using set cover, Steiner connected dominating set (S-CDS), and traveling salesman problems (all are NP-hard), respectively.

Fig. 6.5(a) shows BEST uses only an average of 4% more relays compared to the S-CDS based solution (both given sufficient bandwidth conditions). It shows that inaccessibility to a global map does not hurt much for deploying relay placement. On the other hand, the extra traveled path length compared to the S-CDS solution is a more notable 30.7% on average, according to Fig. 6.5(b). An explanation is that the global map will help to choose the shortest paths globally, leading to significant saves in path length. For the relay placement, however, a local placement, which attempts to “stretch out”: sparsely placing relays to be as far as possible to existing ones, performs similar to a global S-CDS to connect the BS to any sensing position, which together covers the whole region.

Fig. 6.4(b) shows the bandwidth constraint's impact on the number of relays deployed. The number of relays rises 82.8% for $K = 3$ and another 68.9% for $K = 2$ on average respectively compared to the bandwidth sufficient case. The results demonstrate the significant impact of QoS on the number of relays: when there is no unsaturated path, new relays are deployed.

6.6 Future Work

6.6.1 Extension with Multiple Relay-deploying Robots

In the future, we will enhance BEST with multiple RDN to reduce the work load of each RDN and make the solution more scalable with increase in $|\mathbb{FN}|$ and flow sending rate. Multiple relay deploying robots are a natural consideration when a single RDN has a high workload of relay deployment and a lengthy traveling path to keep track a large number of \mathbb{FN} .

With n number of RDN, we may dynamically partition the \mathbb{FN} to different clusters and assign the RDN to its closest cluster of FNs. The partition's objective may be set to (1) achieve a work load balance for each RDN: minimize the largest difference of $\overline{v_{rdn}}$ among RDNs; (2) reduce the total energy cost: minimize the sum of all $\overline{v_{rdn}}$ among RDNs.

6.6.2 Towards Traveling Path Aware Relay Deployment

Currently BEST solves FST problem by first computing the relay positions. With the relay positions as input, the traveling path is obtained by computing the Hamiltonian path. In a more integrated approach, it is desirable that the relay placement outputs a set of candidate positions where the next step can further evaluate which candidates are better. The uncertainty of the posi-

tions for the traveler to travel resembles the *Generalized TSP* problem [119], where the position set G is partitioned into clusters and the objective is to find the shortest cycle in G which passes at least one position in each cluster.

Modeling: We may model the FST problem in part by Generalized TSP since there is flexibility and “room” for the relay placement between any two end positions. (e.g. Given distance of 5m and communication range of 2m, two relays can be placed in a set of positions as long as any two neighboring nodes has a distance of no more than 2m.) Therefore, the relay placement will output a candidate set of positions for each relay, which will be the input for Generalized TSP to further reduce the path length compared with fixed positions.

Discussions: In the future, we would like to implement BEST with traveling path-aware relay deployment to observe its impact on the performance and verify different possible impacts discussed below. It is possible that we may not benefit from the Generalized TSP based solution when the relay deploying time is large and obstacle ratio is low. The reason is that relays may be placed more densely and closer to the existing ones, in order to reduce the local travel path given the flexibility of relay positions. This measure, however, may not help from a global view.

On the other hand, it is also possible performance may be improved when the obstacle ratio is high and the relay deploying time is low. The reason is that the reduction in traveling paths may outweigh the increased number of relays. With densely random obstacles, placing relays sparsely to be “as far as possible” from existing relays may not be optimal.

6.7 Summary

In this chapter, we propose BEST, or the *Bandwidth-aware Exploration with a Steiner Traveler* for an unknown environment. BEST computes the relay deployment positions and traveling paths for the relay-deploying robot to keep track of a group of frontier robots. BEST enables bandwidth-aware real-time multimedia transmissions to support remote sensing and control of a group of robots. In addition, we show BEST is extensible with multiple RDNs in Section 6.6.

We model the problem by the minimum velocity *Flow constrained Steiner Traveler* problem (FST). FST combines two of the most important combinatorial optimization problems: the traveling salesman and the Steiner minimum tree problems. Our solution to FST places new relays to connect to existing ones with unsaturated paths rather than to the BS when using existing relays can reduce the number of relays.

BEST significantly reduces the exploration time by 62% compared to existing homogeneous robot exploration strategies. BEST also reduces the cost by achieving 24% better exploration efficiency with only half the number of robots compared to its comparison counterpart. We also find a marginal improvement in relay number but a notable traveling path length reduction, if a global obstacle distribution were known. In the next chapter, we will focus on coping with mobility and communication in rugged terrains with the other type of mobile nodes: sensors with hopping capability.

CHAPTER 7

Hopping Sensor Relocation in Rugged Terrains

7.1 Preliminaries

Besides the QoS provision and mobile control for teleoperation discussed in the previous four chapters, another major topic in this dissertation is to deal with obstructive environment. The motivation is that the environments are not obstacle free in most applications of mobile networks such as weather sensing, environment monitoring, disaster management, and battlefield surveillance. For the network whose nodes are mobile robots, we have tackled the exploration problems to search in an unknown obstructive environment. Besides the powerful robots, the other most common devices in mobile networks are the sensor nodes. In this chapter, we focus on hopping sensors, whose distinctive movement approach is designed to fit the rugged terrains.

In addition to the inadequacies of hopping sensors such as imprecise movement and limited power mentioned in Section 1.2.1, rugged terrains may also adversely affect their behaviors.

Rugged terrains are the areas that are largely inaccessible to wheeled vehicles, where hopping sensors may operate by trading-off movement accuracy. We model the rugged terrains with two characteristics. The first one is their unevenness with diverse slope patterns. Due to such diversity and arbitrary nature of the terrains we did not model them with slopes in the relocation process. However, we find that hopping sensors are more energy efficient when moving on certain slope ground by comparing their energy consumption with the wheeled ones.

The second feature is the existence of great obstructions in the field that can significantly influence the sensor relocation. To illustrate, water, huge rocks, and other obstructions are inaccessible areas to deploy sensors. It is critical to design a movement scheme that can wisely avoid or circumnavigate these obstructions to reach the expected position. Given an initial sensor deployment, the proposed solution is to match and relocate a redundant sensor to the needed position in an obstructed environment. Furthermore, it must also consider the limited number of hops per sensor and the incapability of sensors landing at a precise location. To tackle these problems, we present (i) a distributed message forwarding model called Binary Splitting Message Forwarding (BSMF) and (ii) a grid based movement model unique to these hopping sensors.

The rest of the chapter is organized as follows. In Section 7.2, the motivation for using hopping sensors is presented. A description of our system model is in Section 7.3. The matching scheme is in Section 7.4. Section 7.5 defines the Grid Based Movement Model. Section 7.6 provides simulations and performance analysis. Summary is presented in Section 7.7.



Figure 7.1. Hopping sensors we designed deployed near a construction area [4].

7.2 Motivation

The motivation of using hopping sensors over wheeled ones in rugged terrains is twofold. The first one is that hopping sensors can access many difficult areas whereas wheeled sensors cannot. Second, for certain types of slopes, which frequently exist in rugged terrains, hopping sensors are more energy efficient. In this section, we focus on the second motivation to compare the energy consumption for two cases: flat plane and slope. To make the comparison reasonable, only the energy used to drive the sensor is considered.

We first conduct an analysis on a flat plane. Under ideal condition, the hopping sensor's trajectory is a projectile motion. In [76], wind influence is modeled as a predictable and stable parameter, which is a strong assumption due to its dynamic nature. Here we assume a negligibly weak wind setting and no sliding occurs during initial state of hopping. Given the sensor's mass m , the initial velocity v , the takeoff angle α , and the distance traveled by the sensor d , the energy per hop will

be:

$$E_h = \frac{1}{2}mv^2 = \frac{mgd}{2 \sin 2\alpha} \quad (7.1)$$

For wheeled sensors, we suppose they have the same weight and need to traverse the same distance. Let the rolling friction coefficient for the flat plane be μ and the sensor run at a constant speed. Then energy for traversing such a distance is:

$$E_w = \mu mgd \quad (7.2)$$

To compare:

$$\frac{E_w}{E_h} = 2\mu \sin 2\alpha \quad (7.3)$$

The rolling friction coefficient between wheels and land is generally less than 0.1 [120], therefore $E_w/E_h < 0.2$, which means $E_h > E_w$. Therefore, for flat plane, the wheeled sensor is more energy efficient. This is because part of the energy for hopping sensors is converted to the potential energy for hopping height.

Now suppose both sensors need to traverse the same distance along a slope with an inclination angle β . For hopping sensors, with the same assumption in the flat plane, the jumping distance along the slope can be found from the intersection of the projectile trajectory and the slope. Establish a coordinate frame as shown in Fig. 7.2, where O is the starting point. Then the x coordinate for the intersection point is:

$$x = d(1 - \cot \alpha \tan \beta) \quad (7.4)$$

Thus $d' = x / \cos \beta$. The energy for hopping sensors to travel d' is the same as traveling d on the flat plane.

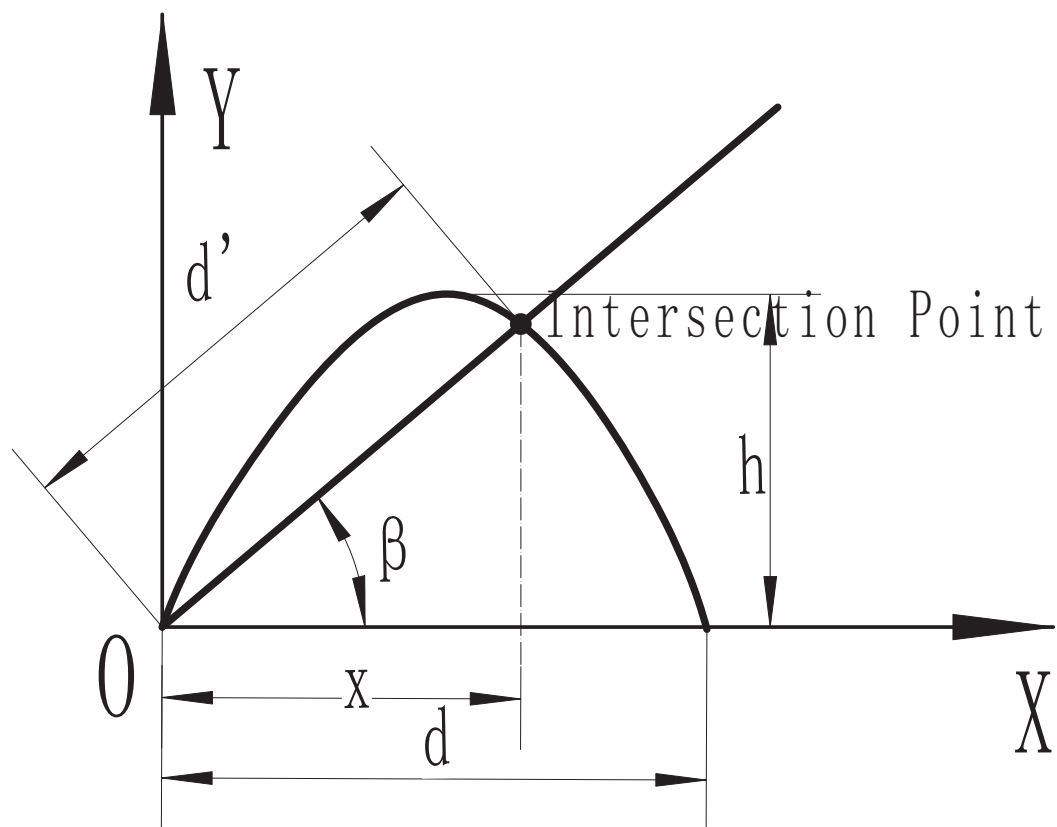


Figure 7.2. Coordinate frame for slope plane.

For wheeled sensors, using the same assumption for flat plane, the energy consumption of traversing d' along the slope is:

$$E_w = d(\mu mg + mg \tan \beta)(1 - \cot \alpha \tan \beta) \quad (7.5)$$

To compare:

$$\frac{E_w}{E_h} = 2 \sin 2\alpha(\mu + \tan \beta)(1 - \cot \alpha \tan \beta) \quad (7.6)$$

Typically, the rolling friction coefficient for car tire on concrete is between $[0.01, 0.015]$ [120]. Let $\mu = 0.0125$. Given $\alpha = 75^\circ$, the takeoff angle of the hopping sensor we have developed [4] (Figure 1.1(b)), we derive the diagram in Fig. 7.3 from equation (7.6) to show the change of above energy ratio with respect to slope angle $0^\circ \leq \beta \leq 75^\circ$.

As we can see from the figure, when $30^\circ < \beta < 72^\circ$ (the curve between two vertical lines), $E_h < E_w$. The ratio decreases after about 62° because d' becomes increasingly smaller as the inclination angle is close to the takeoff angle, resulting in less energy consumed by wheeled sensor. The energy per hop for hopping sensors, however, remains the same. This explains why the ratio decreases. Nevertheless, most wheeled sensors can only move along slope with an angle less than 45° [121]. Hence we can neglect the declining part after approximately 62° , and conclude the hopping sensor is more energy-efficient than the wheeled sensor for slope angle greater than 30° for takeoff angle 75° . In fact, with different takeoff angles, a critical angle always exists when a hopping sensor consumes less energy than a wheeled one.

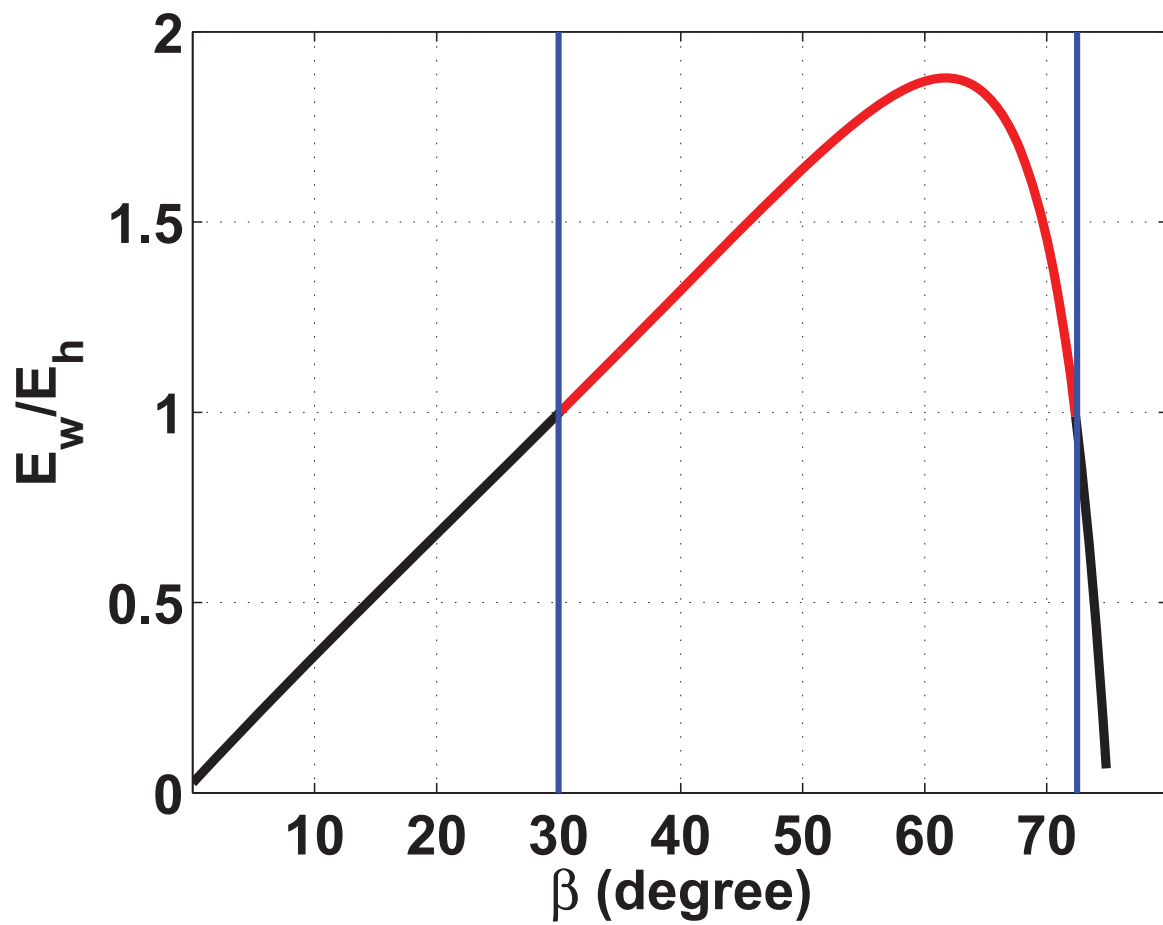


Figure 7.3. Energy ratio for $0^\circ \leq \beta \leq 75^\circ$.

7.3 System Model

Now that we have shown the hopping sensors are energy efficient on uneven ground, in this section we discuss the system model. A grid-based architecture is a natural solution in a network where nodes are relatively regularly deployed. Each grid cell is controlled by a gateway sensor which is capable of communicating with its peers in its immediate four neighbors (North, South, East, West). The gateway sensors can retrieve their absolute positions and have a larger sensing range than other sensor nodes. Sensor nodes within a grid cell register to the cell gateway and perform their required tasks.

The width (size) of a square grid cell is defined by: $W_c = R_g/k$, where R_g is the effective communication range of the gateway and k is the coverage factor. Assuming the gateway located at the center of the cell and given $k=1.41$ or $k=1.5$, 88.4% or 97.2% of 4-neighbor area can be covered respectively for effective inter-gateway communication. Due to space limitation we leave the analysis of the optimal grid cell size in future work.

When a sensor dies, a redundant sensor needs to be relocated to cover the sensing hole area. A redundant sensor can be easily identified on a given cell by the gateway. To maintain the sensor coverage when sensor holes appear, two major steps remain. In the first step, a match is needed between the supplier and consumer cell. Note that more than one supplier may exist. The consumer decides which supplier is selected. Then, a viable path from the supplier to the consumer needs to be computed for the movement. Second, the consumer triggers the relocation process by notifying the selected supplier, which in turn selects one of its redundant sensors and commands it to relocate to the neighbor cell included in the path. We mainly focus on single sensor relocation. The problem of multiple sensor relocation can be easily solved by executing the scheme repeatedly.

The principal objective of this work is to provide an optimized matching path between a consumer and a supplier cell, involving the least amount of intermediate cells possible. A matching process is fundamentally important to the actual sensor movement and must remain workable in presence of obstructions. In our model, obstructions are represented as non-functional grid cells in which a gateway is hard to be placed, or if it exists, it fails to communicate with neighboring cells. These obstructed cells, with high probability, will not allow an intersection cell to match consumer and supplier cell. To overcome this problem, centralized or decentralized algorithms can be considered.

A centralized solution usually inherits the single-point-of-failure weakness, which is less fault-tolerant in hostile environment or a rugged terrain where adverse circumstances are frequent. In addition, it is a strong assumption that a single node has adequate energy and computation capability to communicate with all the other sensor nodes and to find the optimized path for relocation. Since the existence of supplier and consumer is ad hoc and the relocation should be executed efficiently, the latency incurred by gathering supplier and consumer information becomes a drawback.

To be more fault-tolerant, less susceptible to the impact of mobility, and to reduce message overhead and latency in the process of collecting information [122], our algorithm should depend only on local information collected by each gateway to establish the matching path between the supplier and the consumer. To make it more practical, gateway communication is restricted to be with the four-neighboring gateways, and between gateway and managed sensors in the cell.

7.4 Matching Process

7.4.1 Message Forwarding Process

Previous Grid-Quorum relocation would fail if there are obstructions in the middle of the supplier or consumer quorum. We propose Binary Splitting Message Forwarding (BSMF) to cope with obstructions. The algorithm provides a matching path for a consumer and supplier cell, and it considers previously known and newly appearing obstructions.

When a gateway receives an advertisement or request message, it verifies whether it satisfies the demand or needs to forward the message. Before forwarding messages, gateways check the availability of the succeeding neighbor. Succeeding neighbors will vary depending on the message type. Advertisement messages are forwarded to neighbor grid cells in a grid row manner (East-West), while request messages are forwarded in a grid column manner (North-South). When a gateway in a grid cell receives both the advertisement and the request message, the cell becomes an intersection node and the gateway in it will be responsible to match the request to the advertisement. Without obstructions, it is ensured by geographic relations that an intersection is made [80]. If a known or a new obstruction is detected in the next forwarding grid cell, this gateway will change the course by modifying message forwarding behavior in an effort to successfully forward the message. Given an obstruction, a request message may attempt to be forwarded to the East-West grid cell neighbors, while an advertisement message may be forwarded to the North-South neighbors. However, they will return to their default (initial) directions whenever it is possible. If both of the intended grid cell destinations are obstructed or visited before, the message will be forwarded in a direction opposite to the default until one available grid cell is found. Algorithm

BSMF gives a step representation of this procedure.

Algorithm 21: Binary splitting message forwarding (BSMF).

Input: Current cell's gateway G receives an incoming message with i) default forward direction D_{fw} ; ii) message traversed path

Output: Destination neighbor cell(s) $\{NC_d\}$ of the message

```

1 Begin: Supplier and consumer cells have sent row advertisement or column request
   messages respectively
   // Set opposite and two perpendicular directions of  $D_{fw}$ 
2  $D_{opp}, D_{left}, D_{right} \leftarrow D_{fw}$ 
   //  $NC$  denotes a "neighbor cell"
3 if  $NC$  on  $D_{fw}$  is available  $\wedge$  unvisited then
4   |  $\{NC_d\} \leftarrow NC$  // On default whenever possible
5 else
6   |  $\{NC_{uv\_av}\} \leftarrow$  unvisited  $\wedge$  available  $NC$  on  $D_{left}, D_{right}$ 
7   | if  $\{NC_{uv\_av}\} \neq \emptyset$  then
8   |   |  $\{NC_d\} \leftarrow \{NC_{uv\_av}\}$ 
9   | else
10  |   | if  $NC$  on  $D_{opp}$  is available then
11  |   |   |  $\{NC_d\} \leftarrow NC$  on  $D_{opp}$ 
12  |   | else
13  |   |   |  $\{NC_d\} \leftarrow NC$  from which the message was sent
14 foreach  $i \in \{NC_d\} \wedge i$  is out of border do  $\{NC_d\} \leftarrow \{NC_d\} \setminus i$ 
15 The message is forwarded to  $\{NC_d\}$ 
16 if both advertisement and request messages are found then
17   | The advertisement of the closest supplier with the request are forwarded to  $\{NC_d\}$  on
   | the path to the consumer

```

Figure 7.4 illustrates an example of BSMF. A supplier $S(7,0)$ sends the advertisement message

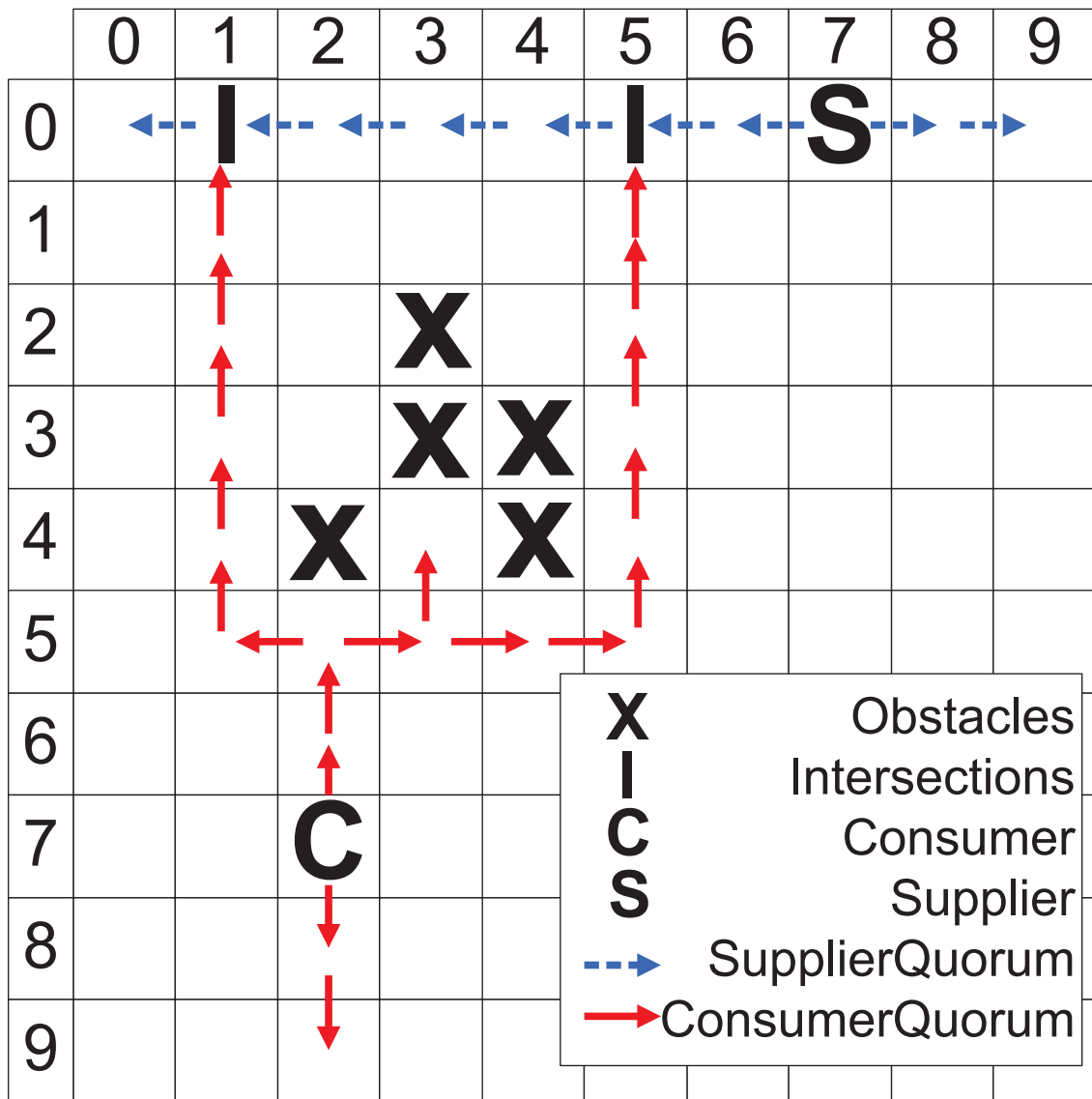


Figure 7.4. Binary splitting message forwarding.

in the first row (X,0) while the consumer C(2,7) initiates the request message in column (2,X). When the gateway at (2,5) receives the request message, it sends a message to check grid cell (2,4) availability. As (2,4) fails to reply, it indicates a new obstacle. Consequently the message is split to (2,5) West and East neighbors (1,5) and (3,5). The message continues to be forwarded in the column of (1,X), while in (3,4), the gateway finds the targeting grid cell (3,3) is an obstructed one. It attempts to ask its East-West neighbors to help forwarding but they do not respond. The gateway at (3,4) sends the message back to (3,5). The gateway at (3,5) will try to split the message but since it finds that (2,5) has been visited it forwards the message to (4,5) only. As (4,4) contains an obstacle, (4,5) will forward the message to (5,5) as well, which then is able to forward in an upward direction. Finally the intersections I(1,0) and I(5,0) are made. Both of them will forward the path information to the consumer. The consumer in (0,7) calculates the length of the two paths gathered as 15 and 13 and selects the path with length 13. It then triggers the movement process by notifying the supplier.

When multiple suppliers exist, the consumer selects the supplier by the following rules. (i) An intersection cell informs the consumer of the advertisement from the closest suppliers. An intersection cell formed by immediate neighboring supplier does not continue to forward the advertisement in the same row. (ii) A consumer cell waits for a predetermined time after receiving the first supplier quorum to allow the arrival of other existing quorums. (iii) The level of sensor redundancy breaks the tie when multiple supplier paths are of equal distance. A supplier cell with more sensors is chosen.

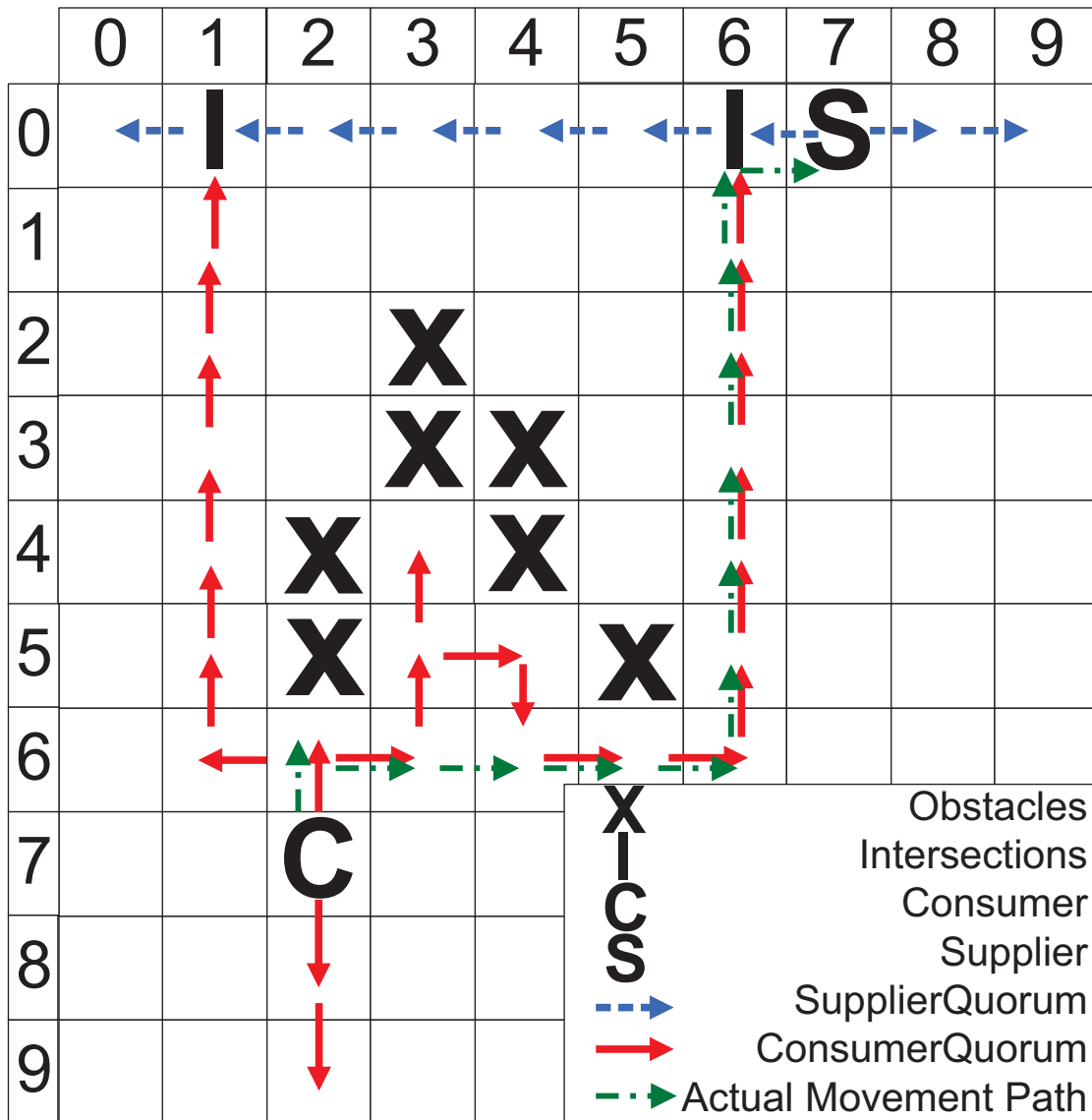


Figure 7.5. Movement path optimization.

7.4.2 Path Optimization

The matching process of the sensor supplier and consumer lays a foundation for the actual sensor movement. With the presence of obstructions, the path in the matching process may include some redundant cells that can be eliminated from the actual relocation process. Such an optimization can further reduce the movement time in the relocation process, which is defined by two additional rules below to check the cells in the matching process. Figure 7.5 also illustrates the idea.

- Rule 1 (Remove dead-end routes): The cell sending a backward message is removed. (e.g., In Fig.7.5, cell (3,4) is removed.)
- Rule 2 (Remove non-straight neighboring routes): The nodes between two adjacent neighboring path nodes are removed. (e.g., In Fig.7.5, cells (3,5), (4,5) are removed.)

7.5 Sensor Migration

7.5.1 Grid Based Movement Model

Having obtained the matching path between the consumer and supplier cells, we need to determine how to move the sensor to the target location. There are typically two migration methods:

- Direct movement: transfer the sensor from supplier to consumer directly;
- Cascaded movement: using intermediate nodes as relaying ones.

Nevertheless, both methods are based on Precise Movement Model (PMM) [80], which uses a sensor to exactly replace another one. In [76], although the authors assume that the sensors can reach the destination if they hop into the target cluster, this is also PMM since the clusters are

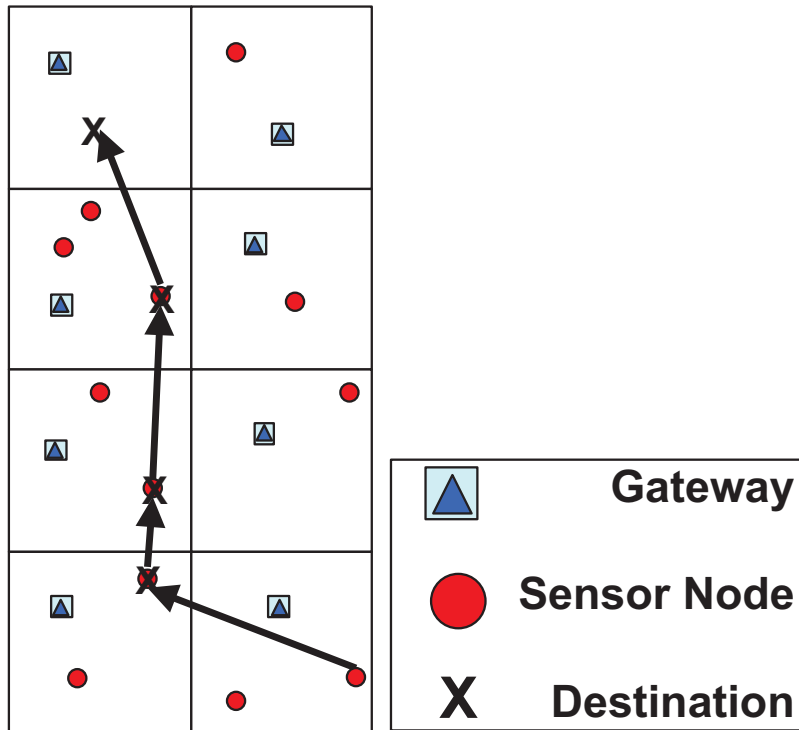
treated as points. Thus each sensor's target is also precise. Due to the inaccurate nature of hopping sensors, we propose a Grid Based Movement Model (GBMM), in which the sensor is only required to move to the target grid cell. Such a pattern fits hopping sensors better since it is not easy for them to be relocated to the exact position and the gateway can easily manage the sensors within its grid cell. Moreover, relaying sensors can move concurrently to save time.

Using GBMM, the most energy efficient way is to transfer the sensor closest to the border between its current cell and the neighboring target cell. The main idea of PMM and GBMM is shown in Fig. 7.6. In Fig. 7.6(a), the predecessor is to move to the successor's position to replace it until the last sensor arrives at the destination. Fig. 7.6(b) illustrates the GBMM. As long as a sensor enters its destination cell, its relocation is accomplished. When the path is the same, it is easy to observe that GBMM can consume less energy compared with PMM because of the decrease of the path length for each transfer.

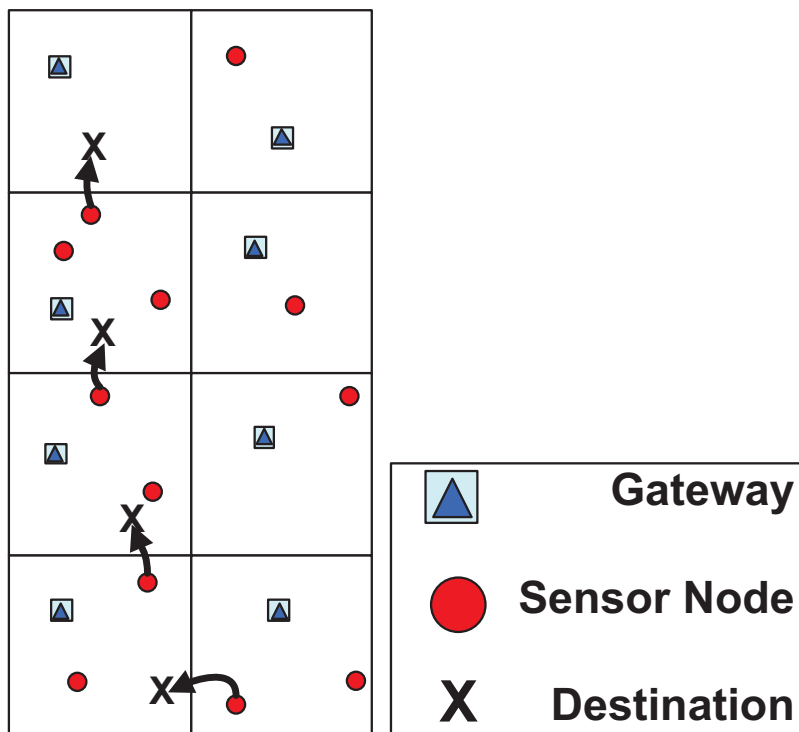
7.5.2 Energy Computation

Using the hopping model in [76] and given the average distance per hop for the sensor as r , the final landing point is subjected to a two dimensional normal distribution $\Delta \mathbf{r} \sim N(\mathbf{0}, \delta^2 \mathbf{I})$, where \mathbf{I} is the 2×2 identity matrix. Note that we assume the independence of the two dimensions because the final landing point is independent of the directions. Assuming the acceptable landing area is within 3δ (this can ensure most of the jumps land in the target area), the number of hops to traverse a distance l satisfies:

$$\lfloor \frac{l}{r + 3\delta} \rfloor \leq N \leq \lceil \frac{l}{r - 3\delta} \rceil \quad (7.7)$$



(a) Precise Movement Model.



(b) Grid Based Movement Model.

Figure 7.6. PMM vs. GBMM.

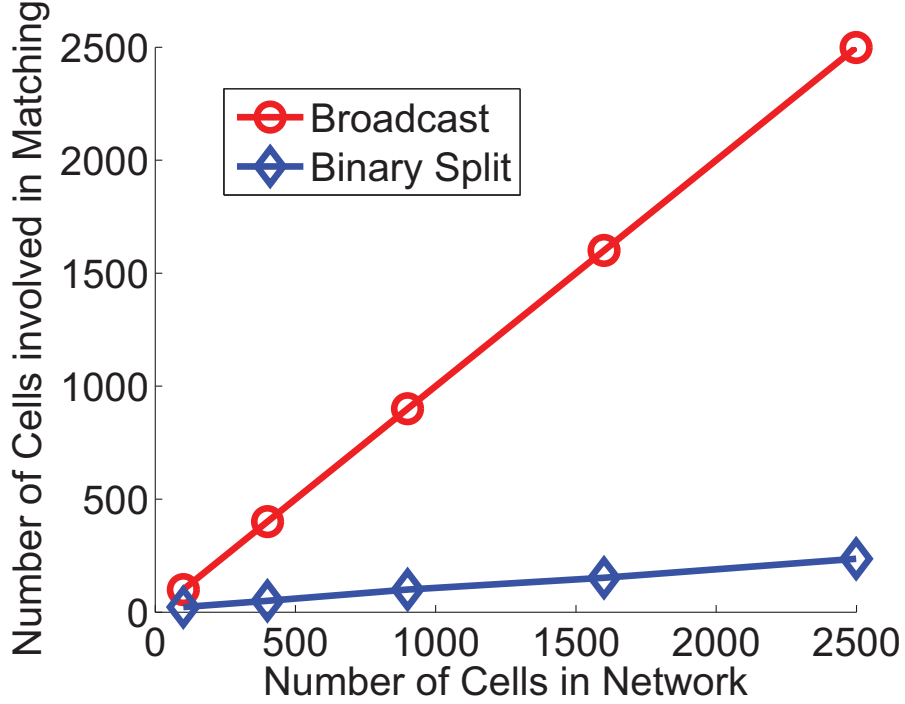


Figure 7.7. Simulation results: broadcast vs. binary splitting matching.

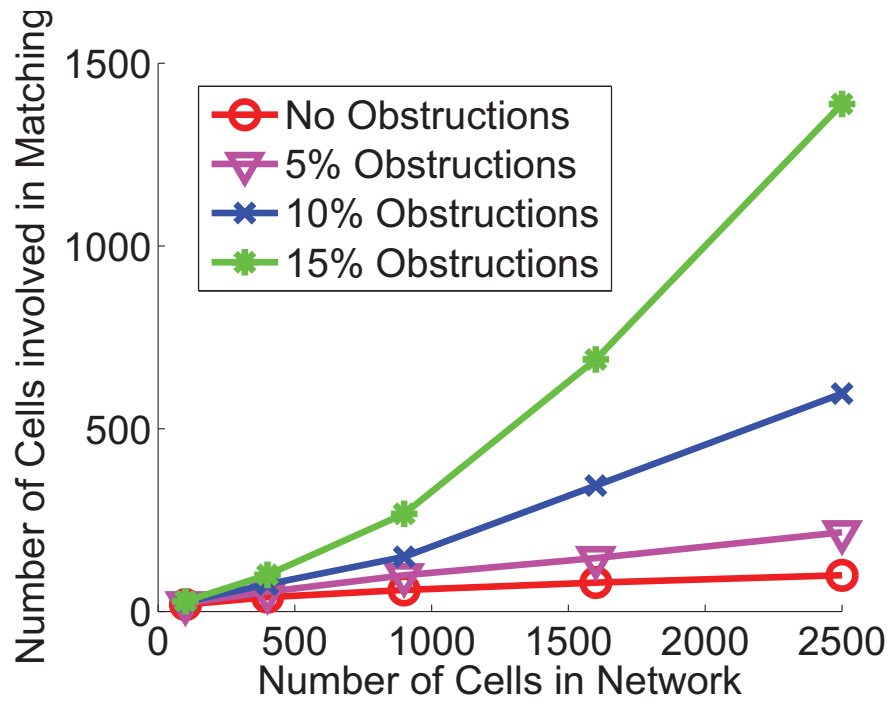
We use the floor on the left and ceil on the right to ensure that the robot can travel the distance l . Assume the number of cells along the path is $n + 1$ including the starting and destination cell, then the number of cell crossing is n . Suppose the distance from the closest sensor in each cell i to the border is l_i , ($i = 1, \dots, n$) (this can be maintained or computed by the gateway), then total hops H for GBMM is between:

$$\sum_{i=1}^n \lfloor \frac{l_i}{r + 3\delta} \rfloor \leq H \leq \sum_{i=1}^n \lceil \frac{l_i}{r - 3\delta} \rceil \quad (7.8)$$

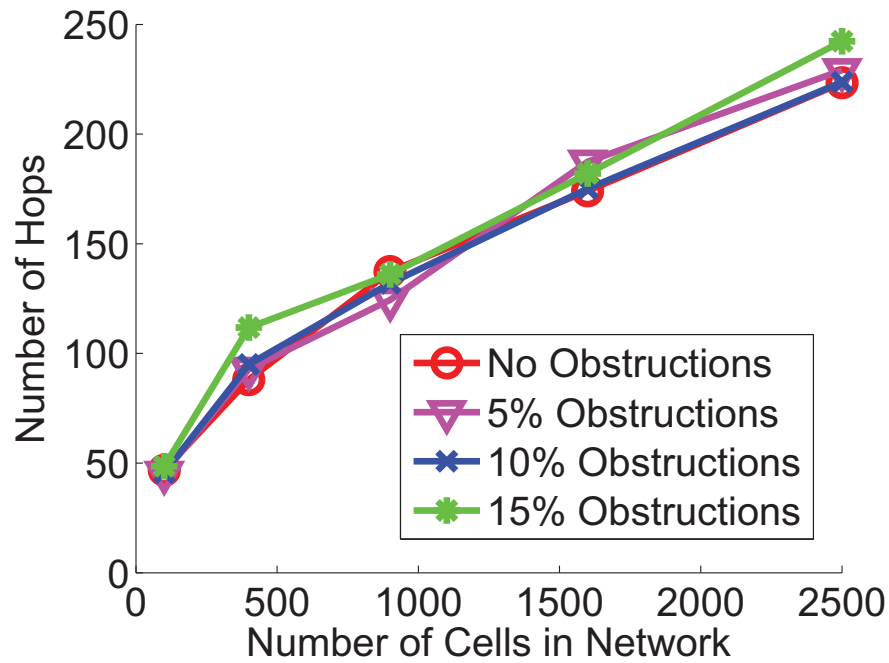
If the energy for each hop is E , then we can get the upper bound and lower bound for the energy as EH for migration. We can also estimate the maximum energy as:

$$En \lceil \frac{W_c}{r - 3\delta} \rceil \quad (7.9)$$

because l_i is always less than the width of grid cell W_c .



(a) Number of cells in matching with varying obstructions.



(b) Number of hops in movement with varying obstructions.

Figure 7.8. Simulation results: number of cells in matching and number of hops in movement.

7.6 Performance Evaluation

To validate the correctness and effectiveness of our algorithm and proposed models, we simulated the hopping sensor network with random positioned supplier and consumer cells. Scenarios were designed with grid sizes of $10 * 10$, $20 * 20$, $30 * 30$, $40 * 40$ and $50 * 50$ grid cells. Each grid cell, with 3-5 hopping sensors randomly distributed, represents an area of $60 * 60$ square meters, which allowed us to represent fields areas up to $9km^2$. Hopping sensors are assumed to have a hopping range with a landing precision radius varying from 2.1 to 3.9 meters. Each hop is modeled to consume a random time between 2 and 5 seconds. Obstructions are randomly distributed in the network and are simulated as non-functional grid cells. Simulations also compared varying number of obstructions with: 0%(no obstructions), 5%, 10%, 15% of obstruction ratio to the total number of grid cells. For each simulation, experiments are run 50 times to calculate average values.

The first two simulations are to evaluate the performance in the matching process by measuring network load (grid cells involved in packet forwarding). As shown in Figure 7.7, BSMF has a much fewer number of grid cells involved in the matching process with 5% obstruction ratio compared to the broadcast approach which guarantees a matching if it ever exists. The conventional quorum-based matching fashion is not compared as it fails with presence of obstructions. In the extreme case, BSMF involves only 9.4% cells to participate compared with broadcast. In Figure 7.8(a), the performance of BSMF for varying grid sizes and amount of obstacles is tested and verified. On average, BSMF involves 80.2% less cells compared to the broadcast approach. Fewer amounts of involved cells indicate much lower total energy consumption, considerably less packet transmission and network traffic generated in the process.

The third simulation evaluates the number of hops and time in the sensor movement process.

Its objective is to measure the total amount of hops taken by sensors to perform the relocation movement after obtaining the relocation path by BSMF. As presented in Figure 7.8(b), the varying number of obstacles has little impact on the number of hops in our model, which demonstrates that the movement spends almost a constant amount of energy, regardless of the amount of obstacles in the movement process. The nearly constant number of hops is contrasted by the trend of network traffic in the second simulation. This result can be explained as the movement is only related to the quorum with the “optimized” intersection, while the matching process is to generate all the possible routes to link the consumer and supplier.

7.7 Summary

In this chapter, we address the problem of relocating the capability-constrained hopping sensors in an obstructive environment. We propose an enhanced Quorum-Grid solution with Binary Splitting Message Forwarding (BSMF), which is decentralized and can detect both existing and newly appearing obstructions in the supplier and consumer cells matching process. Furthermore, a grid-based movement model is introduced for the hopping sensors. Simulation shows that our scheme significantly reduces the communication overhead and achieves relatively constant total energy consumption with varying amount of obstructions. In the next chapter, we summarize the schemes in previous chapters and show our proposed future research work.

CHAPTER 8

Summary and Proposed Future Research Work

8.1 Summary

As mobile devices have been increasing popular in the last decade, supporting multi-hop mobile wireless networks have attracted increased attention. In this dissertation, first, we argue that it is of vital importance to consider the impact of mobility on the network QoS in wireless mobile networks. As topology is the core element in wireless networks influencing their performance, mobility should be more thoroughly analyzed even controlled because it is the mobility pattern that changes the topology all the time. While mobility patterns for humans and many other objects are highly unpredictable and uncontrollable, the patterns in many applications for mobile sensors and robots are not uncommon to be predictable even controllable with a certain constraint. The model of mobile *coordinated* networks has not yet received adequate attention. However, it enables coordination and control on mobility patterns, and is therefore a promising new paradigm for QoS

provision in wireless mobile networks.

Second, we argue that mobility and communication are two critical and inter-dependent topics and we believe that *jointly* considering the two aspects is essential in many research fields. An interdisciplinary research from both networking and robotics perspective is necessary for solving our problem in Chapter 3. From the mobile robotics viewpoint, we leverage algorithmic and graph-theoretic tools to systematically solve the connectivity and bandwidth-aware multi-robot exploration problem. CBAX in Chapter 3 achieves both improved efficiency in exploration time and enhanced quality for communication. From the multi-hop mobile network perspective, CBAX demonstrates how a pragmatic coordinated mobility model can ensure the network connectivity and enhance its quality of service.

Third, we believe the extensive applications in mobile robot networks endow us with new opportunities of exploring the tight *interdependence* and connection of mobility and communication. Multi-robot real-time exploration exhibits such interdependence: On one hand, the aggregated video/audio streams at the BS reveals the global obstacle distribution and guides the robot placement and mobility pattern. On the other hand, the robot movement and proper placement impact the quality of the stream aggregation. As such interdependence is also reflected in the tradeoff of link capacities and transmission ranges, we leverage the tradeoff in the exploration task to significantly reduce the RN size while guarantee the connectivity and bandwidth adequacy for stream aggregation. In addition, interference is tackled by the multi-channel multi-radio architecture with dynamic TDMA scheduling. Our framework in Chapter 4 has broad applications in area exploration, stream aggregation, and teleoperation tasks with remote sensing and control.

Fourth, we propose a novel online relay deployment paradigm to support remote sensing and control in multi-robot target search, environmental monitoring, and area exploration. With a more

cost-effective heterogeneous system composed of mobile robots, sensors, and relay nodes, the on-line relay deployment gives much larger freedom for robots to travel in the area while satisfying the connectivity and bandwidth requirement for remote sensing and control from the base station.

Fifth, we find that the well known combinatorial problems, such as the traveling salesman and the Steiner minimum tree problems have wide applications when we jointly consider mobility and communication for mobile wireless networks. In this dissertation, several new variations of these classic problems have been formulated to model the coordinated mobility patterns and communication requirements, such as (1) Steiner minimum tree problem with minimum Steiner points and γ -inflow constraint, (2) heterogeneous bandwidth Steiner routing (HBSR), (3) minimum time precedence constrained two traveling salesman problem (PC2TSP) from a given city, and (4) minimum velocity flow constrained Steiner traveler problem (FST). Formulation of the problem (4) FST combines both the traveling salesman and the flow constrained Steiner tree problems, further demonstrates the need of jointly considering the (i) tour generation, node placement (mobility), and (ii) bandwidth constraints (communication). These new problems and their solutions facilitate us to improve the mobility efficiency and communication quality in mobile wireless networks.

Last, we argue that the obstructive environment such as rugged terrains is non uncommon in many applications and a special type of low cost mobile sensors effective in such environment is highly desirable. Therefore, we studied the sensor relocation considering obstructions and proposed an enhanced Quorum-Grid relocation solution with an optimized BSMF algorithm in Chapter 7. The proposed algorithm is decentralized and can detect new obstructions in the sensor supplier and consumer matching process. Furthermore, the grid-based movement model is developed and studied with energy computation, suitable for imprecise-movement hopping sensors. The relocation cost has been taken into account, by implementing a path optimization during

matching process execution. Simulation with different grid sizes, random distribution of hopping sensors, and varying amount of obstructions shows that our scheme reduces the involved number of cells in the matching process by 80.2% on average compared to the broadcast approach and achieves relatively constant total energy consumption by evaluating total number of hops in the actual movement.

8.2 Proposed Future Research Work

In the future work of this research, the proposed methods to jointly consider mobility and communication to improve the network quality of service as well as to enhance the mobility effectiveness and efficiency can be further enhanced and verified. In particular, we propose the following major directions below in the future research work.

- We plan to build a real multi-robot platform to evaluate the effectiveness and efficiency of coordinated mobility control and path planning scheme, QoS in video and audio stream aggregation, and interference elimination of multi-radio channel assignment schemes presented in Chapter 3, 4. With the popularity of 802.11n devices, we would like to evaluate the improvement in link capacity when mobile nodes are equipped with 802.11n communication chips. The testing environment will be (i) outdoor, i.e., the garden with pathways and obstacles close to the engineering building of Michigan State University, (ii) indoor, i.e., the hallway of engineering building. Collaborating with the hopping sensor development team at Robotics and Automation Laboratory in Electrical and Computer Engineering Department at Michigan State University, we would like to build a small scale (i.e., 3-10 nodes of hopping sensors) to test on the energy consumption and hopping accuracy when relocating in

both artificial and real rugged terrains.

- It is promising to further refine the multi-robot real-time exploration for remote sensing and control. It is desirable to test the real-time aggregation under 802.11n environment. Also, with the proliferation of 3G/4G networks, it is also interesting to observe a hybrid use of 802.11 and 3G/4G network for bandwidth-consuming real-time data transmission. In addition, some other issues, such as search with control from multiple base stations, handling fault sensors and relays, and coping with dynamic bandwidth requirements due to changing tasks of teleoperation, are also interesting and worthy of further investigation.
- Online relay deployment with precedence constrained multi-robot motion planning provides an alternative method to support remote sensing and control in search targets and monitor environments . In the future, it would be interesting to (1) evaluate the impact of different relay sizes, relay deployment strategies, and carrying capacities on search time, (2) find the optimal precedence constraint assignment or solving the constrained tour generation problem with uncertain precedence constraints, (3) find the optimal precedence constraint assignment or solving the problem with uncertain precedence constraints, (4) apply probability based sensing and communication ranges.
- The hopping sensors are a promising type of mobile sensor for relocating in rugged terrains. In the future, we will (1) model and analyze the unevenness and patterns of the rugged terrain; (2) enhance the BSMF algorithm to cope with newly appearing obstructions during the sensor movement process; (3) compute the optimal grid cell size for the grid structure.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] A. C. I. Chen, James C., “Measured performance of 5-GHz 802.11a wireless LAN systems, 2001, available online,” http://epsfiles.intermec.com/eps_files/eps_wp/AtherosRangeCapacityPaper.pdf.
- [2] M. Inc., “Research robots: Pioneer 3 mobile platform,” <http://mobilerobots.com/ResearchRobots/ResearchRobots/P3AT.aspx>.
- [3] A. Birk, S. Schwertfeger, and K. Pathak, “A networking framework for teleoperation in safety, security, and rescue robotics - [wireless communications in networked robotics],” *Wireless Communications, IEEE*, vol. 16, no. 1, February 2009.
- [4] J.G. Zhao, R.G. Yang, N. Xi, B.T. Gao, and X. G. Fan, “Development of a miniature self-stabilization jumping robot,” *Intelligent Robots And Systems, IROS 2009. IEEE/RSJ Int. Conf. on*, 2009.
- [5] P. Mohapatra, J. Li, and C. Gui, “Qos in mobile ad hoc networks,” *Wireless Communications, IEEE*, vol. 10, no. 3, pp. 44 – 52, 2003.
- [6] “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.
- [7] Crossbow, “Mica2 and micaz motes,” <http://www.xbow.com>.
- [8] “Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions,” *Ad Hoc Networks*, vol. 4, no. 1, pp. 83 – 124, 2006.
- [9] M. TechNet, “Quality of Service technical white paper, chapter 2 - what is network QoS,” <http://technet.microsoft.com/en-us/library/bb742481.aspx>.
- [10] L. Hanzo-II and R. Tafazolli, “A survey of qos routing solutions for mobile ad hoc networks,” *Communications Surveys Tutorials, IEEE*, vol. 9, no. 2, pp. 50 –70, 2007.
- [11] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks - Technology, Protocols and Applications*. John Wiley & Sons, 2007.
- [12] M. Ringwald and K. Römer, “Deployment of sensor networks: Problems and passive inspection,” in *5th Workshop on Intelligent Solutions in Embedded Systems. WISES 2007*, 2007.
- [13] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, pp. 483–502, 2002.

- [14] “On redundancy, efficiency, and robustness in coverage for multiple robots,” *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1102 – 1114, 2008, towards Autonomous Robotic Systems 2008: Mobile Robotics in the UK, 10th British Conference on Mobile Robotics - Towards Autonomous Robotic Systems (TAROS 2007).
- [15] M. A. Batalin and G. S. Sukhatme, “Coverage, exploration and deployment by a mobile robot and communication network,” *Telecommunication Systems*, vol. 26, pp. 181–196.
- [16] H. Choset, “Coverage for robotics a survey of recent results,” *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126.
- [17] C. Trevai, Y. Fukazawa, H. Yuasa, J. Ota, T. Arai, and H. Asama, “Cooperative exploration path planning for mobile robots by reaction-diffusion equation on graph,” in *Industrial Technology, 2002. IEEE ICIT '02. 2002 IEEE International Conference on*, vol. 2, 2002, pp. 1266 – 1271 vol.2.
- [18] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, “Coordinated multi-robot exploration,” *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, June 2005.
- [19] B. Yamauchi, “Frontier-based exploration using multiple robots,” in *Proceedings of the second international conference on Autonomous agents*, ser. AGENTS '98, 1998, pp. 47–53.
- [20] R. Zlot, A. Stentz, M. Dias, and S. Thayer, “Multi-robot exploration controlled by a market economy,” in *Robotics and Automation, ICRA 2002. IEEE International Conference on*, 2002.
- [21] N. Hazon, F. Mieli, and G. Kaminka, “Towards robust on-line multi-robot coverage,” in *Robotics and Automation. ICRA 2006. IEEE International Conference on*, May 2006, pp. 1710 –1715.
- [22] M. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257 –1270, 2006.
- [23] M. Zavlanos and G. Pappas, “Distributed connectivity control of mobile networks,” *Robotics, IEEE Transactions on*, vol. 24, no. 6, Dec. 2008.
- [24] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, “Maintaining connectivity in mobile robot networks,” in *International Symposium on Experimental Robotics, ISER 2008.*, Athens, Greece, 2008.
- [25] J. Tan, N. Xi, W. Sheng, and J. Xiao, “Modeling multiple robot systems for area coverage and cooperation,” in *Robotics and Automation. ICRA 2004. IEEE International Conference on*, 2004.
- [26] S. Poduri and G. Sukhatme, “Constrained coverage for mobile sensor networks,” in *Robotics and Automation. Proceedings. ICRA 2004. IEEE International Conference on*, 2004.

- [27] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment," *Automatic Control, IEEE Transactions on*, vol. 49, no. 8, pp. 1292 – 1302, 2004.
- [28] F. Zeiger, N. Kraemer, and K. Schilling, "Commanding mobile robots via wireless ad-hoc networks - a comparison of four ad-hoc routing protocol implementations," *Robotics and Automation. ICRA 2008. IEEE International Conference on*, May 2008.
- [29] K. Dantu and G. Sukhatme, "Connectivity vs. control: Using directional and positional cues to stabilize routing in robot networks," in *Robot Communication and Coordination, 2009. ROBOCOMM '09. Second International Conference on*, 31 2009.
- [30] D. Moraes, P. Coelho, E. Cardozo, T. Johnson, F. Atizani, and E. Guimaraes, "A network architecture for large mobile robotics environments," in *Robot Communication and Coordination, 2009. ROBOCOMM '09. Second International Conference on*, 31 2009.
- [31] K. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008.
- [32] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 2, April 2009.
- [33] R. Rocha, F. Ferreira, and J. Dias, "Multi-robot complete exploration using hill climbing and topological recovery," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008.
- [34] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robotics and Autonomous Systems*, vol. 54, no. 12, 2006.
- [35] J. Vazquez and C. Malcolm, "Distributed multirobot exploration maintaining a mobile network," in *Intelligent Systems, 2004. 2nd International IEEE Conference on*, vol. 3, June 2004.
- [36] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, 2007.
- [37] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," in *Robotics and Automation. ICRA 2008. IEEE International Conference on*.
- [38] H. Sugiyama, T. Tsujioka, and M. Murata, "Coordination of rescue robots for real-time exploration over disaster areas," in *Object Oriented Real-Time Distributed Computing, ISORC 2008. 11th IEEE International Symposium on*, May 2008.

- [39] A. Mosteo, L. Montano, and M. Lagoudakis, “Multi-robot routing under limited communication range,” *Robotics and Automation. ICRA 2008. IEEE International Conference on*, May 2008.
- [40] K. E. Bekris, K. I. Tsianos, and L. E. Kavraki, “A distributed protocol for safe real-time planning of communicating vehicles with second-order dynamics,” in *Robot Communication and Coordination. RoboComm 2007. 1st International Conference on*, 2007.
- [41] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, “Connectivity maintenance in mobile wireless networks via constrained mobility,” in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 927–935.
- [42] W. Guo and X. Huang, “Mobility model and relay management for disaster area wireless networks,” in *Wireless Algorithms, Systems, and Applications, WASA 2008. 3rd International Conference on*, 2008.
- [43] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, “Relay sensor placement in wireless sensor networks,” *Wireless Networks*, vol. 14, no. 3, 2008.
- [44] N. Li, B. Yan, and G. Chen, “Measurement study on wireless camera networks,” in *Distributed Smart Cameras. ICDSC 2008*.
- [45] Z. Cen, M. W. Mutka, D. Zhu, and N. Xi, “Supermedia transport for teleoperations over overlay networks,” in *NETWORKING 2005*, ser. Lecture Notes in Computer Science, R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. P. Black, Eds. Springer Berlin, Heidelberg, 2005, vol. 3462, pp. 1409–1412.
- [46] Z. Cen, M. Mutka, Y. Liu, A. Goradia, and N. Xi, “Qos management of supermedia enhanced teleoperation via overlay networks,” in *Intelligent Robots and Systems, 2005. IROS 2005. 2005 IEEE/RSJ International Conference on*, Aug. 2005.
- [47] S. Misra, S. D. Hong, G. Xue, and J. Tang, “Constrained relay node placement in wireless sensor networks: Formulation and approximations,” *Networking, IEEE/ACM Transactions on*, vol. 18, no. 2, 2010.
- [48] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, “Two-tiered constrained relay node placement in wireless sensor networks: Efficient approximations,” in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 IEEE Communications Society Conference on*.
- [49] F. Wang, D. Wang, and J. Liu, “Traffic-aware relay node deployment for data collection in wireless sensor networks,” in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2009 IEEE Communications Society Conference on*.

- [50] D. Yang, X. Fang, G. Xue, and J. Tang, "Relay station placement for cooperative communications in wimax networks."
- [51] X. Han, X. Cao, E. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks."
- [52] Y. Pei, M. Mutka, and N. Xi, "Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness," in *Robotics and Automation, ICRA 2010. IEEE International Conference on*.
- [53] B. Lin, P.-H. Ho, L.-L. Xie, X. Shen, and J. Tapolcai, "Optimal relay station placement in broadband wireless access networks," *Mobile Computing, IEEE Trans. on*, vol. 9, no. 2, feb. 2010.
- [54] G. Xue, T. P. Lillys, and D. E. Dougherty, "Computing the minimum cost pipe network interconnecting one sink and many sources," *SIAM J. on Optimization*, vol. 10, 1999.
- [55] Y. Pei, M. Mutka, and N. Xi, "Connectivity and bandwidth aware real-time exploration in mobile robot networks," in *Wiley Wireless Communications and Mobile Computing Journal, WCMC, (in press)*.
- [56] D. Yang, S. Misra, and G. Xue, "Joint base station placement and fault-tolerant routing in wireless sensor networks."
- [57] L. Li, G. Xin, L. Sun, and Y. Liu, "Qvs: Quality-aware voice streaming for wireless sensor networks," in *Distributed Computing Systems, 2009. ICDCS 2009. 29th IEEE International Conference on*.
- [58] M. Li and Y. Liu, "Underground structure monitoring with wireless sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*.
- [59] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega, Int. J. Management Sci.*, vol. 34, no. 3, 2006.
- [60] T. Oncan, I. K. Altinel, and G. Laporte, "A comparative analysis of several asymmetric traveling salesman problem formulations." *Computers & Operations Research*.
- [61] L. Gouveia and P. Pesneau, "On extended formulations for the precedence constrained asymmetric traveling salesman problem," *Networks*, vol. 48, no. 2, pp. 77–89, 2006. [Online]. Available: <http://dx.doi.org/10.1002/net.20122>
- [62] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a heterogeneous, multiple depot, multiple traveling salesman problem," in *American Control Conference, 2009. ACC '09.*, june 2009, pp. 1292–1297.

- [63] E. Balas and N. Simonetti, "Linear time dynamic-programming algorithms for new classes of restricted TSPs: A computational study," *INFORMS J. on Computing*, vol. 13, no. 1, 2000.
- [64] "The multiple TSP with time windows: vehicle bounds based on precedence graphs," *Operations Research Letters*, vol. 34, no. 1, 2006.
- [65] A. Haumann, K. Listmann, and V. Willert, "Discoverage: A new paradigm for multi-robot exploration," in *Robotics and Automation, ICRA 2010. IEEE International Conference on*.
- [66] P. Brass, F. Cabrera-Mora, A. Gasparri, and J. Xiao, "Multirobot tree and graph exploration," *Robotics, IEEE Transactions on*, no. 99, 2011.
- [67] J. Yuan, Y. Huang, T. Tao, and F. Sun, "A cooperative approach for multi-robot area exploration," in *Intelligent Robots and Systems, 2010. IROS 2010. IEEE/RSJ International Conference on*.
- [68] P. Mukhija, K. Krishna, and V. Krishna, "A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint," in *Intelligent Robots and Systems, 2010. IROS 2010. IEEE/RSJ International Conference on*.
- [69] A. Marjovi, J. Nunes, L. Marques, and A. de Almeida, "Multi-robot exploration and fire searching," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*.
- [70] M. Batalin and G. Sukhatme, "The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment," *Robotics, IEEE Transactions on*, vol. 23, no. 4, 2007.
- [71] Y. Wang and C.-H. Wu, "Robot-assisted sensor network deployment and data collection," in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*.
- [72] P. Corke, R. Peterson, and D. Rus, "Localization and navigation assisted by networked cooperating sensors and robots," *International Journal of Robotic Research*, 2005.
- [73] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *Pervasive Computing, IEEE*, vol. 3, no. 4, 2004.
- [74] G. Fletcher, X. Li, A. Nayak, and I. Stojmenovic, "Back-tracking based sensor deployment by a robot team," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 IEEE Communications Society Conference on*.
- [75] R. Tan, G. Xing, J. Wang, and H. C. So, "Exploiting reactive mobility for collaborative target detection in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 3, pp. 317–332, 2010.

- [76] Z. Cen and M. W. Mutka , “Relocation of hopping sensors,” in *Robotics and Automation. ICRA 2008. IEEE Int. Conf. on*, May 2008.
- [77] S. Chellappan, X. Bai, B. Ma, and D. Xuan, “Sensor networks deployment using flip-based sensors,” in *Mobile Adhoc and Sensor Systems. MASS 2005. IEEE Int. Conf. on*, 2005.
- [78] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Distributed Autonomous Robotics Systems. DARS 2002. Int. Symposium on*, 2002.
- [79] G. Wang, G. Cao, and T.L. Porta, “Movement-assisted sensor deployment,” *INFOCOM 2004. 23rd Conference of the IEEE Computer and Communications Societies*, vol. 4, 2004.
- [80] G. Wang, G. Cao , T.L. Porta, and W. Zhang , “Sensor relocation in mobile sensor networks,” *INFOCOM 2005. 24th Conference of the IEEE Computer and Communications Societies*, vol. 4, 2005.
- [81] J. Wu and S. Yang, “Smart: a scan-based movement-assisted sensor deployment method in wireless sensor networks,” *INFOCOM 2005. 24th Conference of the IEEE Computer and Communications Societies*, vol. 4, 2005.
- [82] R. Akl and U. Sawant, “Grid-based coordinated routing in wireless sensor networks,” *Consumer Communications and Networking Conference. CCNC 2007. IEEE*, 2007.
- [83] G. Cao and M. Singhal, “A delay-optimal quorum-based mutual exclusion algorithm for distributed systems,” *Transactions on Parallel and Distributed Systems. T.PDS*, vol. 12, no. 12, 2001.
- [84] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, “Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks,” *Mobile Networks and Applications*, vol. 10, no. 1-2, 2005.
- [85] M. Younis, P. Munshi, G. Gupta, and S. Elsharkawy , “On efficient clustering of wireless sensor networks,” *Dependability and Security in Sensor Networks and Systems. DSSNS 2006. IEEE*.
- [86] Y. Pei, F. Cintron, M. Mutka, J. Zhao, and N. Xi, “Hopping sensor relocation in rugged terrains,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 3856 –3861.
- [87] M. Kim and M. Mutka, “Multipath-based relocation schemes considering balanced assignment for hopping sensors,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 5095 –5100.

- [88] M. Kim, M. Mutka, and H. Choo, "On relocation of hopping sensors for rugged terrains," in *Computational Science and Its Applications (ICCSA), 2010 International Conference on*, 2010, pp. 203–210.
- [89] M. Lewis, H. Wang, P. Velagapudi, P. Scerri, and K. Sycara, "Using humans as sensors in robotic search," in *Information Fusion. FUSION 2009. 12th Int. Conference on*, 2009.
- [90] "Key performance benefits of 802.11n, available online," http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns348/ns767/white_paper_c11-513840.html.
- [91] D. Du and X. Hu, *Steiner Tree Problems In Computer Communication Networks*. World Scientific Publishing Co., 2008, pp. 177–193.
- [92] H. H. Gonzales-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *International Journal of Robotics Research*, vol. 21, no. 10-11, 2002.
- [93] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 2nd Edition*. The MIT Press, 2001, pp. 1056–1061.
- [94] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd Edition*. Prentice-Hall, 2003, pp. 97–101.
- [95] R. Burkard, M. Dell'Amico and S. Martello, *Assignment Problems*. SIAM: Society for Industrial and Applied Mathematics, 2009, pp. 172–191.
- [96] P. Krokhmal and P. P.M., "Random assignment problems," *European Journal of Operational Research*, vol. 194, no. 1, 2009.
- [97] U. Pferschy, "The random linear bottleneck assignment problem," *RAIRO-Operations Research*, vol. 30, no. 2, 1996.
- [98] H. Gabow and R. Tarjan, "Algorithms for two bottleneck assignment problems," *Journal of Algorithms*, vol. 9, no. 3, 1988.
- [99] R. Burkard and E. Cela, "Linear assignment problems and extensions," *Handbook of Combinatorial Optimization, vol. A (Suppl.)*, 1999.
- [100] "SRG-based video clips from a cluster start with 4 robots," <http://www.dis.uniroma1.it/~labrob/research/multiSRG.html>.
- [101] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Mobile Computing and networking, MobiCom 2003. 9th ACM International conference on*, 2003.

- [102] A. Wijesinha, Y. tae Song, M. Krishnan, V. Mathur, J. Ahn, and V. Shyamasundar, "Throughput measurement for UDP traffic in an IEEE 802.11g WLAN," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on*.
- [103] S. Kakumanu and R. Sivakumar, "Glia: a practical solution for effective high datarate wifi-arrays," in *Mobile Computing and networking, MobiCom 2009. ACM International conference on*.
- [104] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Mobile Computing and networking, MobiCom 2004. ACM International conference on*.
- [105] P. Huang, C. Wang, L. Xiao, and H. Chen, "RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks," in *Quality of Service (IWQoS), 2010 18th International Workshop on*.
- [106] "FCC 15.407 as of august 8, 2008 - hallikainen.com, available online," <http://sujan.hallikainen.org/FCC/FccRules/2008/15/407/>.
- [107] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *SIGCOMM 2010: Proceedings of the ACM SIGCOMM conference on Data communication*.
- [108] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "Virtual calibration for RSSI-based indoor localization with IEEE 802.15.4," in *Communications, 2009. ICC 2009. IEEE International Conference on*.
- [109] P. Crescenzi and V. Kann, "A compendium of NP optimization problems," p. 52, 1998.
- [110] E. Malaguti and P. Toth, "A survey on vertex coloring problems," in *International Transactions in Operational Research*, 2010.
- [111] F. Wang and J. Liu, "Networked wireless sensor data collection: Issues, challenges, and approaches," *Comm. Surveys Tutorials, IEEE*, 2010.
- [112] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European J. of Operational Res.*, vol. 140, no. 3.
- [113] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *MobiHoc 2006: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*.

- [114] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.
- [115] S. C. Sarin, H. D. Sherali, and A. Bhootra, “New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints,” *Operations Research Letters*, vol. 33, no. 1, pp. 62 – 70, 2005.
- [116] K. Hornik and M. Hahsler, “TSP–infrastructure for the traveling salesperson problem,” *J. of Statistical Software*, vol. 23, no. i02, 2007.
- [117] N. Ascheuer, M. Jünger, and G. Reinelt, “A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints,” *Comput. Optim. Appl.*, vol. 17, 2000.
- [118] Y. Pei and M. Mutka, “STARS: Static relays for multi-robot real-time search and monitoring,” in *Distributed Computing in Sensor Systems (DCOSS), 2011 International Conference on*, June 2011, pp. 1–8.
- [119] G. Gutin, A. Punnen, A. Barvinok, E. K. Gimadi, and A. I. Serdyukov, “The traveling salesman problem and its variations,” 2002.
- [120] T. E. Toolbox, “Friction and coefficients of friction,” http://www.engineeringtoolbox.com/rolling-friction-resistance-d_1303.html.
- [121] <http://www.mobilerobots.com/>.
- [122] N. Li, J. Hou, and L. Sha, “Design and analysis of an mst-based topology control algorithm,” *INFOCOM 2003. 22nd Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, 2003.