LIFETIME MAXIMIZATION FOR CONNECTED TARGET COVERAGE IN WIRELESS SENSOR NETWORKS

ZHAO QUN

(M.S., TsingHua University)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

Acknowledgements

I would like to thank to my supervisor, Dr. Mohan Gurusamy, for his guidance, support, and encouragement throughout my study. His deep insights and advices beyond academic and research were and will be well appreciated.

I also thank to NUS CNDS lab folks, Wang Wei, Wang Bang, Luo tie, Yeow Weiliang, Qin Zheng, Li hailong, Ai Xin, Hu Zhengqing and Jia jingxi, etc. for their kind assistance and valuable discussions on algorithms, programming, and paper writing. They make my staying in the lab and Singapore enjoyable and memorable.

Finally, I thank to my parents for their love and support.

Contents

Summary

Recent advances in micro-electro-mechanical systems, digital electronics, and wireless communications have led to the emergence of *wireless sensor networks* (WSNs), which are comprised of a large number of sensors each with sensing, data processing and communication capabilities. As sensors are unattended low-cost devices, *network lifetime* is one of the most important and challenging issues in WSNs which defines how long the deployed WSN can function well. Maintaining *coverage* and *connectivity* are two fundamental requirements in a WSN. In this thesis, we consider the *connected target coverage (CTC) problem* with the objective of maximizing the network lifetime by scheduling sensors into multiple sets, each of which can maintain both target coverage and connectivity.

We first model the CTC problem as a *maximum cover tree (MCT) problem* and prove that the MCT problem is NP-Complete. We determine an upper bound and a lower bound on the network lifetime for the MCT problem and then develop a $(1 + w)H(\hat{M})$ approximation algorithm to solve it, where w is an arbitrarily small number, $H(\hat{M}) = \sum_{1 \leq i \leq \hat{M}} \frac{1}{i} \leq (\ln \hat{M} + 1)$ and $\hat{M}$ is the maximum number of targets in the sensing area of any sensor. We further prove that $[1 - O(1)] \ln(M)$ is a threshold below which the MCT problem cannot be approximated efficiently, unless NP has slightly super-polynomial time algorithms, i.e. $NP \subset TIME(n^{O(loglogn)})$, where M is

the number of targets. As the protocol cost of the approximation algorithm may be high in practice, we develop a faster heuristic algorithm based on the approximation algorithm called Communication Weighted Greedy Cover (CWGC) algorithm and present a distributed implementation of the heuristic algorithm. We study the performance of the approximation algorithm and CWGC algorithm by comparing them with the lifetime upper bound and other basic algorithms.

Next, we consider the CTC problem when the data generation rate of a sensor is proportional to the number of targets it observes and with K coverage requirement wherein each target is observed by at least K sensors. Such K-coverage requirement improves the accuracy and reliability of the observations. We formulate the problem as the Lifetime Maximization Observation Schedule (LMOS) problem and study the problem with two observation scenarios depending on whether a sensor can select a subset of targets in its sensing area to observe or not. For the first scenario, we develop a polynomial-time algorithm which can achieve the optimal solution. For the second scenario, we show that the problem is NP-complete. We develop approximation algorithms for both scenarios. Based on the approximation algorithms, we develop a low-cost heuristic algorithm which can be implemented in a distributed fashion for both scenarios.

Finally, we present a general framework of approximation algorithm for the CTC problem. We show that the CTC problem can be approximated by solving the problem of selecting a set of active sensors that minimizes the weighted communication cost while maintaining connectivity and coverage.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 An Overview of Wireless Sensor Networks

Recent advances in micro-electro-mechanical systems, digital electronics, and wireless communications have led to the emergence of wireless sensor networks (WSNs) [1, 2]. Wireless sensor networks are proposed for a wide range of applications including battlefield surveillance, environmental monitoring, biological detection, smart spaces and industrial diagnostics [3, 4, 5, 6]. In wireless sensor networks, there are a large number of low-cost, low-power, multi-functional sensing devices called sensor nodes. Each sensor node is equipped with sensing, data processing and communication capabilities. The sensor nodes form a connected network and work collectively to accomplish the assigned tasks such as surveillance, environment monitoring and data gathering.

Since sensors are low-cost devices, a large amount of sensors could be densely deployed [7] inside or surrounding the interested phenomenon to provide the measurements with satisfactory accuracy. The dense deployment of sensors makes it difficult and unnecessary to have deterministic deployment of sensors. Thus the sensor nodes could be randomly deployed in the hostile or hazardous environment. Once the sensors are randomly deployed, sensors have to be self-organized to build the

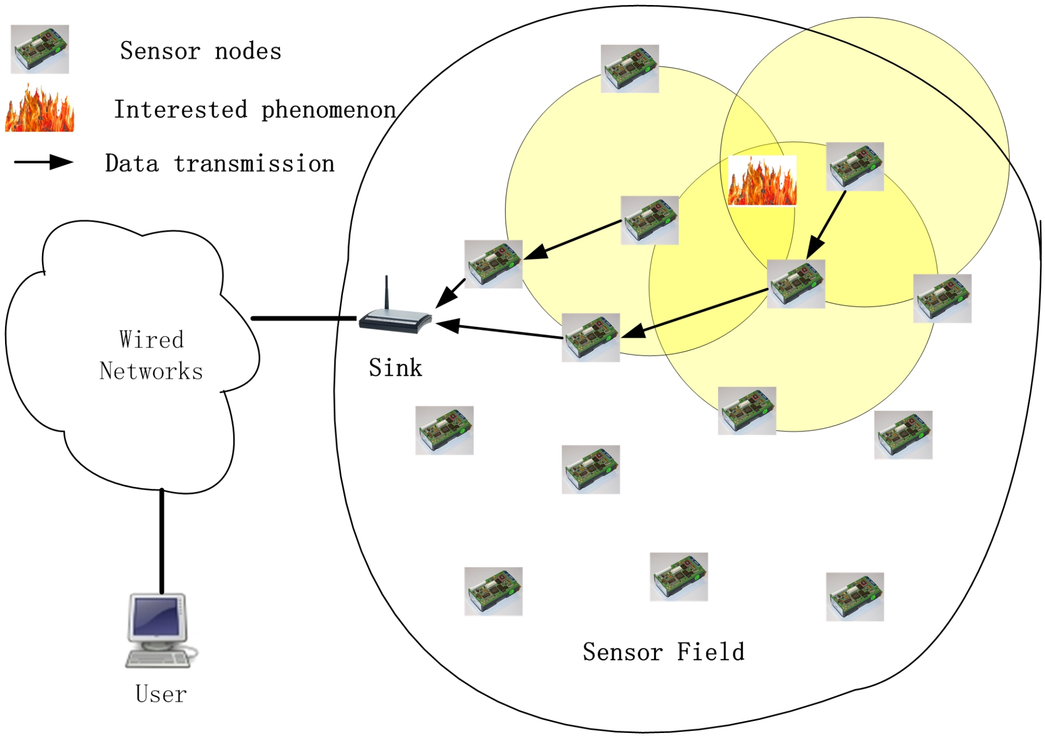

Figure 1.1: A typical sensor network architecture

network topology and route the collected information.

The dense and random deployment of sensor nodes also makes it almost impractical to recharge such a large amount of devices in a possibly hostile or rather large area. Thus sensor nodes are usually assumed unattended devices. Further, each low-cost sensor node has only limited resources such as power, computational ability, bandwidth and memory. Once a sensor node consumes all its battery energy, it will "die" – disappear in the network. The network may cease to work when the remaining sensor nodes are not sufficient to accomplish the assigned tasks. Energy efficiency is a crucial issue in sustaining sensor network functionalities and extending system lifetime.

In a typical sensor network architecture as shown in Fig. 1.1, a phenomenon of interest such as the fire is sensed by sensors around it. One or more central controllers called *sink* nodes collect and further process the data generated by the sensors. The sink node may communicate with the *users* via traditional wired or wireless network infrastructures. The sensor nodes report the sensed data and communicate to the sink node via single or multi-hop communications. As the sink node may not be unattended, it is usually regarded as a node in the network with infinite (i.e. sufficiently large) resources such as battery energy and processing capability.

1.1.1 Comparison with traditional Ad hoc networks

Wireless sensor networks are a new family of wireless ad hoc networks. Although many algorithms and protocols have been proposed for wireless ad hoc networks, they are not well suited to the unique features and application requirements of sensor networks. The key differences between wireless sensor networks and ad hoc networks are:

1. The number and the density of nodes in a sensor network are likely to be much larger than that of most ad hoc networks.

2. Sensor queries in sensor networks are often data-centric. The queries indicate the required data but not the addresses of sources that provide the data. Any sensor node that can provide the required data can be the source.

3. The limited battery energy of unattended sensor nodes makes sustaining sensor network functionalities to be one of the most important issues in WSNs.

4. As sensor nodes are densely deployed and data is being extracted from the

environment, the data from neighboring nodes is highly redundant [8]. By reducing the data redundancy, both the network traffic can be reduced and the energy efficiency can be improved.

5. Sensor nodes are prone to failures. Sensor nodes may fail due to lack of power, physical damage, or radio interference. The topology of sensor networks may be highly dynamic due to sensor node failures or environmental changes.
6. Sensor networks have a different communication paradigm compared to traditional ad hoc networks. As the sink node is the destination of most sensing data, the dominating communication paradigm in sensor networks is many-to-one communications instead of the point to point communications in ad hoc networks.
7. Sensors are cheap and simple devices, and therefore the use of complex algorithms and expensive facilities is not desirable.

The above features of sensor networks pose new challenges and require new solution approaches. The sensor network algorithms and protocols should be scalable, robust, self-organized and energy efficient.

1.2 Network lifetime of wireless sensor networks

Network lifetime is one of the most important and challenging issues in WSNs which defines how long the deployed WSN can function well. Sensors are unattended nodes with limited battery energy. In the absence of proper planning, the network may quickly cease to work due to the network departure or the absence of observation sensors deployed close to the interested phenomenon. Since a sensor network is usually expected to last several months without recharging [9, 10], prolonging network lifetime is one of the most important issues in wireless sensor networks.

A sensor node is generally composed of four components: sensing unit, data processing unit, data communication unit and power unit [1]. The power unit supplies power to the other three units. Any activity of the other three units – sensing, data processing, data transmitting and data receiving – will consume battery energy. Experiments show that wireless communication (data transmitting and receiving) contributes a major part to energy consumption rather than sensing and data processing [11]. Therefore, reducing the energy consumption of wireless radios is the key to energy conservation and prolonging network lifetime.

Radios in sensors consume energy not only when sensors are transmitting or receiving, but also when listening or idle. In idle state the radio still needs to be powered to detect the presence of incoming data packets. It is observed that the energy consumption in the idle state cannot be ignored compared with that in the state of transmitting or receiving. Sensors consume almost the same amount of energy when it is idle or receiving. For example, the power usage for WINS Rockwell seismic sensor for transmit:receive:idle:sleep operational modes is 0.38-0.7 W:0.36 W: 0.34 W:0.03 W while the sensing power is 0.02 W [12]. Therefore, the radios should be turned off to save the energy consumption when the sensors are not necessary for the assigned tasks. We call the sensors with radios turned on to be in "active" state and the sensors with radios turned off to be in "sleep" state. The network lifetime can be greatly increased by *scheduling sensor activities* wherein only a subset of sensors are let to be active and all the other sensors are let to sleep. The improved lifetime is achieved due to the reduced idle listening, collisions of media access control (MAC) and traffic load.

There are multiple definitions for the network lifetime based on different assumptions. In [13, 14, 15], etc. the network lifetime is defined as the period from the time when the network was set up to the time when the first sensor node dies due to energy dissipation. However, sensor nodes are normally highly redundant in the network to accomplish the assigned tasks. The network may still function well after the first sensor node dies. A more realistic definition of the network lifetime is the period from the time when the network was set up to the time when the WSN cannot satisfy the requirement of assigned tasks [16, 17]. For most sensor network applications such as surveillance or data gathering, coverage and connectivity are two fundamental requirements. Therefore, in this thesis, we define the network lifetime as the duration until the coverage or connectivity of the sensor network breaks.

1.3 Coverage in Wireless Sensor Networks

Coverage is a fundamental issue in a WSN, which determines how well a phenomenon of interest (area or target) is monitored or tracked by sensors [18, 19]. Each sensor node is able to sense the phenomenon in a finite *sensing area*. Any point in the sensing area of a sensor is said to be covered by the sensor. The sensing area of a sensor is normally assumed to be a disk with the sensor located at the center. The radius of the disk is called the *sensing range* of the sensor. There are broadly three types of coverage classified based on what is to be covered, namely area coverage, discrete points coverage and barrier coverage [18].

The area coverage requires that each point in the interested area is covered by at least one active sensor node. The requirement can be extended to K-coverage where each point in the area should be covered by at least K active sensors. The K-coverage requirement improves the accuracy and reliability of the observations [20], and is necessary for many applications such as localization and target classification [21].

Area coverage guarantees that each point in the interested area is continuously monitored, however, this may be more than what is necessary for applications. We may be more interested in some crucial positions (targets) than the whole area in which sensors are deployed, e.g. the street crossing in a city or the gates in a building. Instead of covering the whole area as in the area coverage problem, the target coverage problem requires to cover only a finite set of discrete points (targets) in the interested area. Clearly, providing area coverage is a sufficient condition for providing target coverage, but may waste the precious battery energy. On the other hand, providing target coverage can approximate area coverage by increasing the number of targets [22], and the target coverage will be equal to area coverage when there is at least one target in each face divided by the area boundary and boundaries of deployed sensors' sensing areas [23]. Here a face is defined as the region surrounded by the boundaries but without any boundary crossing it. The target coverage problem is useful for the kinds of applications such as surveillance or environment data collection where fixed points or locations are required to be monitored.

If the discrete targets of interest are geographically separated with known locations and the number of targets is small, deterministically deploying a cluster of sensors close to each target with a long radio range node in each cluster to communicate with the sink can be a good solution. However, a more general case needs to be considered where the targets may spread in an area and a sensor can have multiple targets in its sensing area. This can happen in applications where a cluster of sensors is casually or randomly deployed around a cluster of geographically-nearby targets. Further, in applications such as battle field surveillance the exact locations of targets may not be known in advance and the deterministic deployment is prohibitive. The sensors have to be randomly deployed into the susceptible area, where they recognize the targets, observe them and send the observation data back to the sink via multi-hop communications.

Both the area coverage and target coverage use a binary model for the sensing capacity of sensors, that is, the interested phenomenon would be equally sensed by a sensor at any point in its sensing area and would not be sensed outside the area. However, in barrier coverage [24, 25], the sensing capability of a sensor is presented as the probability that a sensor detects the phenomenon, and is assumed to be related to some other factors such as the distance between the sensor and interested phenomenon. The barrier coverage concerns with determining the probability that an undetected penetration passes through the barrier (area where sensors deployed). The maximal breach path (MBP) and the maximal support path (MSP) are defined as the path with the highest or lowest probability, respectively, that an undetected penetration passes through the barrier [24, 25].

1.4 Connectivity in Wireless Sensor Networks

Connectivity is an important issue in WSNs which concerns with delivering the sensed data from the source sensor to the destination (sink node) via radio transmissions. As sensors are low-cost devices with constrained resources, each sensor node has only limited communication range compared with the size of the monitored area. Multi-hop communications are necessary when a sensor cannot reach the sink node directly. Two sensors are called *neighbors* if they are within each other's communication range. The sensor nodes and the communication links between each pair of neighbors build the network topology, which is required to be connected by the connectivity requirement.

The network lifetime can be extended and the communication energy consumption can be saved by controlling the network topology. Two techniques are often used to control the network topology while guaranteeing network connectivity. The first one tries to adjust the transmission power of each sensor node which results in adjusting the network connectivity [26, 27, 28, 29, 30, 31, 32, 33, 34], while the other one tries to schedule the activity of sensors - turning nodes' radio on or off - to control the network topology and decrease the total energy consumption [35, 36, 37, 38, 39].

Due to the space fading of wireless signals, the transmission power used at the sender will exponentially increases as the transmission range increases. To avoid wasting the precious energy, the transceiver of a sensor could be power controlled such that different transmission power levels are used to achieve different communication ranges. A sensor may forward the data packages to different neighbors using different transceiver power lever according to the distance from itself to the neighbor. By this way, an one-hop transmission from the sender to the receiver may consume much more energy than a multi-hop transmission through relays located between the sender and the receiver [40]. By carefully selecting the relay nodes, the total data transmission energy consumption in the network can be greatly saved and many redundant links in the network can be deleted from the network topology.

On the other hand, sensors are redundantly deployed. Only a subset of sensors may be sufficient to build the network communication backbone. Other sensors not on the backbone can go into a sleep state to conserve the energy consumption of idle listening and overhearing. Therefore, many techniques are developed to carefully choose the subset of sensors providing network connectivity which can also conserve the energy consumption or maximize the network lifetime.

1.5 Scheduling sensor activities while maintaining coverage and connectivity

Scheduling sensor activities is a promising approach to save the energy consumption and prolong the system lifetime, which selects a necessary subset of sensors to be active satisfying the application requirements. The problem of scheduling sensor activities can be categorized based on different application requirements i.e. coverage and connectivity requirements. The problem of scheduling sensor activities while maintaining area coverage has been studied in [23, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. The problem of scheduling sensor activities while maintaining target coverage has been studied in [16, 51, 52, 53, 54]. The problem of scheduling sensor activities while maintaining connectivity has been studied in [35, 36, 37, 38, 39]. The problem of scheduling sensor activities while maintaining both coverage and connectivity has been studied in [55, 56, 57, 58, 59, 60, 61, 62]. It has been shown in [60, 61] that the network connectivity can be guaranteed if the complete area coverage is achieved and the communication range is at least twice the sensing range. However, for the target coverage problem, this claim does not hold as shown by an example in [62].

Although all the above techniques on scheduling sensor activities aim to save the energy consumption and prolong the system lifetime, the specific optimization objective that each technique considers may be different. A straightforward objective for the scheduling problem is to select a minimum set of sensors to be active, i.e. the number of active sensors selected is minimized ([46, 55, 56], etc.). However, the minimum set of active sensors may not be the most energy efficient one, e.g. the total data transmission energy consumption can be reduced by properly adding relay sensors between the transmitter and the receiver. In [48, 49], etc. the authors try to select a set of active sensors such that the total energy consumption is minimized. Further, as sensors are redundantly deployed, different sets of sensors can be activated within different durations before the network lifetime ends. Finding the minimum or most energy efficient set of active sensors is not sufficient to maximize the network lifetime. In [41, 51], etc. the design objective is to find a maximum number of disjoint sets of active sensors. Each set of active sensors is able to operate for a fixed duration of time, and thus the network lifetime can be prolonged by finding more sets of active sensors. In [16] the authors illustrate that network lifetime can be further improved without the constraint that the chosen active sensor sets are disjoint, i.e. a sensor may appear in different sets. In [16, 23], etc. the design objective is to maximize the network operation duration before the application requirements cannot be met due to the death of sensors.

1.6 Contribution and organization of the thesis

In this thesis, we address the problem of scheduling sensor activities while maintaining target coverage and network connectivity.

Chapter 2 reviews related work on scheduling sensor activities and lifetime maximization in wireless sensor networks.

In chapter 3, we introduce the Connected Target coverage (CTC) problem. The sensor field consists of a set of discrete targets with fixed locations, a number of randomly deployed sensors and a sink node. We assume that sensors are equipped with power controlled transceivers and non-rechargeable batteries with limited energy. The application requirements are to cover **all** the targets **all** the time and to send **all** the sensed data to the sink by a subset of the deployed sensors. In other words, the connected target coverage problem requires that all the targets are covered by a subset of sensors (coverage requirement) and all the targets are connected to the sink node through a subset of sensors by single-hop or multi-hop paths (connectivity requirement). If any of the above requirements cannot be satisfied, we say that the deployed WSN reaches its lifetime. Sensing, transmission and reception consume battery energy and the lifetime of such energy-constrained WSN is limited. Our objective is to maximize the network lifetime of such a WSN. We model the CTC problem as a Maximum Cover Tree (MCT) problem and prove that the MCT problem is NP-Complete. We develop a linear programming formulation to derive the upper bound and lower bound on the network lifetime for the MCT problem.

In chapter 4, based on the upper bound and lower bound derived in chapter 3, we develop a $H(\hat{M})(1+w)$ approximation algorithm to solve the MCT problem, where w is an arbitrarily small number, $H(\hat{M}) = \sum_{1 \leq i \leq \hat{M}} 1/i$ and $\hat{M}$ denotes the maximum number of targets in the sensing area of any sensor. Our approach is to divide the deployed sensors into a number of sensor sets each of which can cover all the targets and can send all the sensed data to the sink. These sensor sets need not be disjoint, and are activated successively one by one: Each time only one set is active. Only sensors in an active set are used to sense targets and to relay data to the sink, and all the other sensors go into an energy-saving sleep state. The energy consumption of each sensor is directly related to the amount of data sensed and relayed by the sensor. We further prove that $[1 - O(1)]ln(M)$ is a threshold below which the MCT problem cannot be approximated efficiently, unless $NP \subset TIME(n^{O(loglogn)})$, where M is the number of targets. As a practical implementation we develop a much faster heuristic algorithm called Communication Weighted Greedy Cover (CWGC). The CWGC algorithm uses a greedy method to select the set of source nodes (called source set) that cover the targets and it couples the communication cost and the selection of source sets. We carry out extensive simulations to demonstrate the effectiveness of the proposed approximation algorithm and heuristic algorithm by comparing their results with the upper bound on the lifetime. Further, we demonstrate the superiority of our algorithms by comparing them with other basic algorithms which consider the coverage and connectivity problems independently.

In chapter 5, we consider the CTC problem when the data generation rate of a sensor is proportional to the number of targets it observes and with K coverage requirement wherein each target is observed by at least K sensors. Such K-coverage requirement improves the accuracy and reliability of the observations. We model the CTC problem in this case as a Lifetime Maximization Observation Schedule (LMOS) problem and discuss the problem with two different observation scenarios depending on whether a sensor can select a subset of targets in its sensing area to observe or not. We prove that the LMOS problem for the first scenario (LMOS-1) is a P problem and develop a polynomial-time algorithm for it which can achieve the optimal solution based on Linear Programming and Integer Theorem [63]. We show that the LMOS problem for the second scenario (LMOS-2) is NP complete. We derive an upper bound and a lower bound of the LMOS-2 problem based on the optimal solution of LMOS-1 problem.

In chapter 6, approximation algorithms for both LMOS-1 and LMOS-2 problems are developed which provide insights into the LMOS problem and can be used to evaluate the performance of other algorithms. As a practical implementation we develop a faster flexible heuristic algorithm called Communication Weighted Observation Scheduling (CWOS) for both problems which can be implemented in a distributed fashion. We carry out extensive simulations to demonstrate the effectiveness of the proposed heuristic algorithm by comparing its performance with that of the optimal solution for the LMOS-1 problem and the approximation algorithm of the LMOS-2 problem.

In Chapter 7 we present a general framework of approximation algorithm for the CTC problem. This algorithm is applicable to various possible instances of the CTC problem described by different application scenarios, say for example, with different observation scenarios and communication schemes. We show that the lifetime maximization problem for connected target coverage can be approximated by solving the problem of selecting a set of active sensors that minimizes the weighted communication cost while maintaining connectivity and coverage.

Chapter 8 summarizes the work in this thesis and presents some future directions.

The list of research papers based on this thesis work is given in "List of publications".

Chapter 2

Related Work

2.1 Network coverage

The coverage concept is a measure of the quality of service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications [24]. The coverage requirements include (complete or partial) area coverage and complete target coverage. Barrier coverage is another type of coverage problem but the objective is to minimize the probability of undetected intrusion through the barrier [24, 25]. Considering the coverage concept, different problems can be formulated, based on the subject to be covered – area or discrete targets, and on the objective of the problem – maximizing network lifetime or minimizing the number of active sensors. The coverage algorithms proposed in the literature are centralized, or distributed and localized. In distributed algorithms, the decision process is decentralized. Distributed and localized algorithms refer to a distributed decision process at each node that makes use of only neighborhood information (within a constant number of hops). Because the network has a dynamic topology and needs to accommodate a large number of sensors, the algorithms and protocols designed should be distributed and localized in order to accommodate a scalable architecture better.

2.1.1 Area coverage

The problem of scheduling sensor activities for complete area coverage is addressed in [60, 61, 23, 47, 55, 48, 49, 50]. Maintaining partial (but high) area coverage is discussed in [64, 65, 45, 66].

In [41, 47] the authors consider a large population of sensors, deployed randomly for area monitoring. The goal is to achieve an energy-efficient design that maintains area coverage. Because the number of sensors deployed is larger than the optimum required to perform the monitoring task, the solution proposed is to divide the sensor nodes into disjoint sets so that every set can individually perform the area monitoring tasks. These sets are then activated successively. When the current sensor set is active, all other nodes are in a low-energy sleep mode. The goal of this approach is to determine a maximum number of disjoint sets because this has a direct impact on the network lifetime i.e., no sensor appears in two covers. The solutions proposed are centralized in nature.

In [41] the area is modeled as a collection of fields in which every field has the property that any enclosed point is covered by the same set of sensors. The most constrained, least constraining algorithm [41] is developed to successively compute the disjoint covers. The algorithm prefers to the sensors that cover the critical element (field covered by a minimal number of sensors) and gives priority to the sensors covering a high number of uncovered fields or sparsely covered fields. In [47] the disjoint sets are modeled as disjoint dominating sets. The maximum disjoint dominating sets computation is NP complete, and an algorithm based on graph coloring is proposed to compute the maximum number of disjoint dominating sets. Simulation results show that the number of sets obtained in [47] is 1.5 to 2 times more than those in [41].

The above algorithms focus on finding maximum number of disjoint sets. In [23], sensors are divided into non-disjoint sets for the area coverage problem using a packing Linear Programming technique. An approximation algorithm is proposed based on the Garg-Konemann algorithm.

The above solutions are all centralized algorithms. In [48] a distributed and localized algorithm is proposed to solve the area coverage problem, called Node Scheduling Scheme Based On Eligibly rule (SBO). In the SBO rule, the operation is divided into rounds such that at each round, the sensors decide their own state, i.e., whether to sleep or be active. At each round, the active sensors are active to cover the given area where all the other sensors are in the sleep mode. This operation repeatedly runs for next round. The main question that needs to be addressed here is that what rule the sensors should follow to determine their state. The authors proposed a Coverage-based Off-duty Eligibility rule (CBO) to address this question. In the CBO rule, a sensor decides to turn it off when its sensing area is covered by its neighbors, called sponsors. To avoid blind point, which may happen when two or more neighboring sensors expect each other's sponsoring, a Back-off based scheme is also introduced in [48]. This scheme lets each sensor delay the decision process with a random period of time. To obtain neighboring information, each sensor broadcasts a position advertisement message containing node ID and node location at the beginning of each round.There is no proof on the performance ratio of the proposed algorithm.

2.1.2 Target coverage

From the definitions of target coverage and area coverage, one can easily see that there must exist a relationship between the area coverage problem and the target coverage problem. The area coverage problem can be transformed to the target coverage problem [41] by placing a target in each face surrounded by the area boundary and boundaries of deployed sensors' sensing areas. In [23] it is proved that the number of faces of the graph is at most $n(n-1)+2$ given n sensors each with convex sensing area. If the positions of sensors are given, we could find all the faces in $O(n^3)$ time and thus reduce the area coverage problem into a target coverage problem.

In [51], the discrete target coverage problem is modeled as a disjoint set covers (DSC) problem which is proven to be NP-Complete. The DSC problem is transformed into a maximum-flow problem, which is then formulated as a mixed integer programming as a basis for a heuristic solution. The simulation results in [51] show that the heuristic outperforms the SP heuristic [41] in terms of the increased number of produced disjoint sensor covers. The above work is extended in [16] that the network lifetime can be further improved without the constraint that the chosen set covers are disjoint, that is, a sensor may appear in different covers. For example, as in Figure 2.1, the network consists of 3 sensors $\{s_1, s_2, s_3\}$ that cover 3 targets $\{p_1, p_2, p_3\}$. Target p_1 is covered by sensors s_1 and s_2. Target p_2 is covered by sensors s_2 and s_3. Target p_3 is covered by sensors s_3 and s_1. Each sensor can operate for a unit of time and each target is required to be covered by at least one sensor. If the sensors are organized into disjoint sets each of which covers all the three targets,

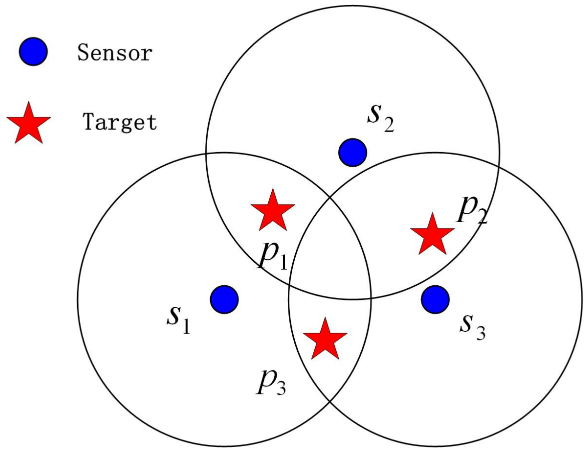

Figure 2.1: An example network for illustration of disjoint and non-disjoint sets

only one set among the three sets $\{s_1, s_2\}$, $\{s_2, s_3\}$ or $\{s_1, s_3\}$ can be selected as the set of active sensors. The set can operates for 1 unit of time and thus the network lifetime is 1 unit of time. If the sensors are organized into non-disjoint sets each of which covers all the targets, the optimal network lifetime is 1.5 unit of time by sequentially selecting sets $\{s_1, s_2\}$, $\{s_2, s_3\}$ and $\{s_1, s_3\}$ as the set of active sensors and letting each set operate for 0.5 unit of time. In [16], the problem of maintaining target coverage is modeled as a Maximum Set Covers (MSC) problem and is shown to be NP-Complete. Two heuristics are designed to efficiently compute the sets based on linear programming and a greedy approach.

Sensors are assumed to have the same sensing range in the above works [41] [51, 16]. In [52], the target coverage problem is addressed based on another assumption that sensors have adjustable sensing ranges. It is formulated as an Adjustable Range Set Covers (AR-SC) problem with the objective to find a maximum number of set covers for the ranges associated with each sensor. Three heuristics are designed for the problem. One is based on the integer programming and two others are based on a greedy approach with centralized and localized versions.

Different from the assumption used in [41, 51, 16, 52, 62] that each active sensor simultaneously observes all the targets in its sensing area, in [53] the authors assume that each sensor can freely select the target to observe and it observes only one target at each time. With this assumption, an optimal solution is proposed to find the target observation schedule that achieves maximum network lifetime. The results are extended to the situation when each target is required to be covered by at least K sensors (K coverage) in [54].

All the above works do not consider the connectivity issue. Further, the impact of communication energy consumed for sending the sensed data and relayed data on the sensor activity scheduling has not been given due consideration in the above works, because, they either ignore the energy consumption for data transmission or assume that each active sensor consumes the same amount of energy per unit time. However, in a practical scenario, the energy consumed by the active nodes can vary significantly depending on the amount of sensed and relayed data. When the objective is to maximize the network lifetime, the impact of the existence of transmission bottleneck caused by multiple flows traversing through the same relay node should not be neglected.

2.2 Maintaining network connectivity

Maintaining network connectivity is concerned with deciding which set of nodes should be turned on/off and when, for the purpose of constructing energy saving topology to prolong the network lifetime. In [36], geographical adaptive fidelity (GAF) algorithm is proposed to conserve energy consumption by identifying nodes that are equivalent from a routing perspective and then turning off unnecessary nodes. In GAF nodes use location information to divide the field into fixed square grids. The size of each grid stays constant, regardless of node density. Nodes within a grid switch between sleeping and listening mode, with the guarantee that one node in each grid stays up so that a dynamic routing backbone is maintained to forward packets. In [35], a power saving topology maintenance algorithm called Span is proposed for multi-hop wireless networks which adaptively elects coordinators from the nodes to form a routing backbone and turn off other nodes radio receivers most of the time to conserve power. In [38], STEM (Sparse Topology and Energy management) approach is proposed, which exploits the path setup latency dimension rather than the node density dimension to control a power saving topology of active nodes. They switch nodes between two states – transfer state and monitoring state. Data are only forwarded in the transfer state. In the monitoring state, nodes keep their radio off and will switch into transfer state to be an initiator node on the event detected. The extended study on combining STEM and GAF shows the potential of further power saving by exploiting both path setup latency dimension and node density dimension.

All the above works do not consider the coverage issue. Further, the objective of these works is either minimizing the energy consumption in the network or selecting a minimum subset of sensors to be active.

2.3 Coverage and connectivity

2.3.1 Maintaining both connectivity and area coverage

It has been proved that 1-coverage implies 1-connectivity when the ratio between the radio transmission range and the sensing range is at least two [61]. Based on the observation, a distributed mechanism, Optimal Geographic Density Control (OGDC), is proposed in [61] to maximize the number of sleeping sensors while ensuring that the working sensors provide complete 1-coverage and 1-connectivity. OGDC tries to minimize the overlapping area between the working sensors. A sensor is turned on only if it minimizes the overlapping area with the existing working sensors and if it covers an intersection point of two working sensors. A sensor can verify whether it satisfies these conditions using its own location and the locations of the working sensors. OGDC can maintain both 1-coverage and 1-connectivity when the radio transmission range is at least twice the sensing range.

An integrated coverage and connectivity configuration protocol called CCP proposed in [60, 67] aims to minimize the number of active nodes, while maintaining both K-coverage and K-connectivity. It is proved that K-coverage also implies K-connectivity when the transmission range is at least twice the sensing range. To ensure K-coverage, a node only needs to check whether the intersection points inside its sensing area are K-covered. Since CCP cannot guarantee network connectivity when the radio transmission range is less than twice the sensing range, CCP and Span [35] are combined to provide network connectivity when the communication range is less than 2 times of the sensing range.

2.3.2 Maintaining both connectivity and target coverage

In [60] and [61], it is shown that the network connectivity can be guaranteed if the complete area coverage is achieved and the communication range is at least twice the sensing range. However, this claim may not hold in the discrete points coverage problem as indicated by an example in [62]. In [62], the connected set cover problem with adjustable sensing ranges (called ASR-CSC problem) is considered and a heuristic is proposed for it. The connectivity in [62] refers to network connectivity and hence requires that all sensors in the network are connected to each other. The heuristic proposed for the ASR-CSC problem is to construct a virtual backbone (a *connected dominating set*, CDS) for network-wide connectivity and then select working sensors and their sensing ranges for the target coverage. A distributed and localized rule is applied to construct the CDS [68] and the selection of working sensors is based on a greedy approach that adds a sensor to the cover according to its contribution to the target coverage. However, the energy consumption model in [62] does not consider the transmission and reception power but only the sensing power. Further, in most cases, network-wide connectivity may not be necessary for target coverage and only the sensors along the routes carrying the sensed data are required to be active.

2.4 Maximizing network lifetime

Maximizing network lifetime is an important issue in wireless sensor networks. Designing efficient routing algorithms or communication mechanisms to prolong the network lifetime has been studied in [14, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86]. In [69], the routing problem is formulated as a linear programming problem, where the objective is to maximize the network lifetime, which is equivalent to the time until the network partition occurs due to battery outage, and a minimum cost path routing algorithm is proposed to prolong the network lifetime. In [14], upper bounds on the lifetime of a sensor network are provided by taking into account all the possible collaborative data gathering strategies over the possible network routes. The Maximum Lifetime Data Aggregation (MLDA) and Maximum Lifetime Data Routing (MLDR) problem [70] are also studied and solved as the LP problem. In [71], the maximum data collection problem is formulated as a LP problem and an approximation algorithm is developed. In [72], the problem of the lifetime maximization in a wireless sensor network under the constraint of the target end-to-end transmission success probability is investigated, by adopting a cross-layer strategy that considers both physical layer (i.e., power control) and network layer (i.e., routing protocol) jointly. However, all the above lifetime maximization techniques are based on a communication network graph, they do not address the issue of network coverage and scheduling sensors activities.

Chapter 3

Maximum cover tree (MCT) problem

In this chapter, we consider the problem of scheduling sensor activities to maximize network lifetime while maintaining network connectivity and target coverage. We call the problem as Connected target coverage (CTC) problem. We study the connectivity issue and consider the communication energy consumption in the network. The impact of the communication energy consumed for sending the sensed data and relayed data on the sensor activity scheduling has not been given due consideration in the former works as they ignore the energy consumption for data transmission and assume that each active sensor consumes the same amount of energy per unit time. However, in a practical scenario, the energy consumed by the active nodes can vary significantly depending on the amount of sensed and relayed data. When the objective is to maximize the network lifetime, the impact of the existence of transmission bottleneck caused by multiple flows traversing through the same relay node should not be neglected. Regarding the set of active sensors in each time point as an cover tree, we formulate the CTC problem into the maximum cover tree (MCT) problem. We prove that MCT problem is NP-complete by reducing it from the 3-SAT problem. We propose an upper bound and a lower bound for the MCT problem by solving a Linear Programming (LP) formulation.

3.1 Connected target coverage (CTC) problem

We consider the following application scenario. In a sensor field, a number of targets with fixed locations are required to be continuously monitored (covered) in the field by a (large) number of randomly scattered sensors. Each sensor is assumed to cover a fixed area and any target located in the area could be monitored by the sensor. The data that are sensed and transmitted by the sensors are collected and processed by a sink node. If a sensor is selected to be active for performing the monitoring task, it generates data messages (e.g., quantized measurements) at a certain rate. Such a sensor is called a *source* sensor. Sensed data messages are transmitted to the sink via radio communication. Multiple-hop communication may be needed from a source to the sink. A sensor node which does not perform monitoring task but needs to be activated to relay data is called a *relay* node. A sensor is called an *active* node if it is selected either as a source or as a relay or both. A sensor that is not active goes into an energy saving *sleep* state. In this thesis, *scheduling sensor activity* refers to determining the state of the deployed sensors to be either active (as source or relay or both) or sleep as well as their state durations.

We assume that the following assumptions hold initially when the network is set up:

- (assumption 1) all the sensors deployed in the WSN can reach the sink via single-hop or multi-hop communication;
- (assumption 2) each target is covered by at least one sensor;

The *network lifetime* is defined as the time period from the time when the network was set up until 1) one or more targets cannot be covered, or 2) a route cannot be found to send the sensed data to the sink. Now we define the *connected target coverage* problem (CTC).

Definition 1: *Connected Target Coverage Problem*: Given M targets with known locations and an energy constrained WSN with N sensors, it is required to schedule sensor activity so as to maximize the network lifetime subject to the conditions: 1) each target is covered by at least one source and 2) from each source to the sink, there must exist a route traversing through only the active sensors.

We illustrate the CTC problem in Fig. 3.1. There are 13 sensors, 5 targets and 1 sink in the sensor field. The sensors that can cover one or more targets are indicated by their circles – solid circles for active source sensors and others for sleep or relay sensors. Arrowed lines are used to denote the routes used to relay data from sources to the sink. Two possible solutions are illustrated in Fig. 3.1(a) and Fig. 3.1(b). In both solutions illustrated in Fig. 3.1, all the targets are covered by active sensors and each active sensor can reach the sink. This figure illustrates that only a subset of the deployed sensors is sufficient to carry out the functionalities of the WSN and different subsets can be used in different intervals.

3.2 Problem formulation

In this section, we model the CTC problem as a Maximum Cover Tree (MCT) problem, prove that it is NP-Complete and provide an upper bound on the network lifetime

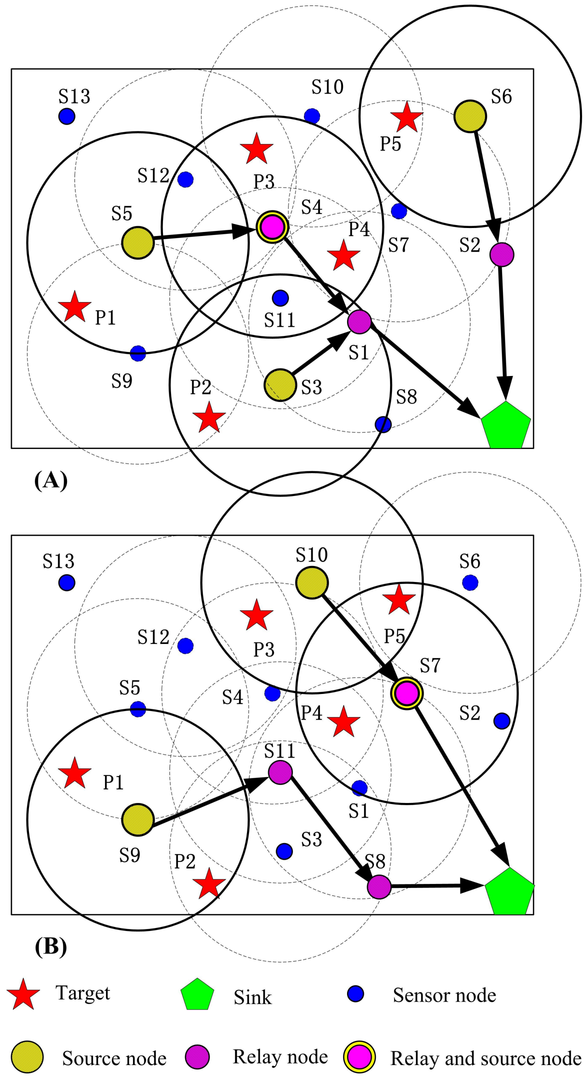

Figure 3.1: Illustration of the CTC problem. (a) solution 1; (b) solution 2 for the MCT problem.

Let $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$ ($|\mathcal{S}| = N$) and $\mathcal{P} = \{p_1, p_2, \cdots, p_M\}$ ($|\mathcal{P}| = M$) denote the set of deployed sensors and the set of targets, respectively. We use $\mathcal{R}$ to denote the sink. If sensor s_i and s_j are neighbors, we say there exists a *communication link* between them. If target p_m is in the sensing area of sensor s_i, we say there exists an *observation link* between s_i and p_m. The set of sensors, targets and sink can be modeled as a (undirected) connected graph $G = (V, \mathbb{E})$, where $V = \mathcal{S} \bigcup \mathcal{R} \bigcup \mathcal{P}$

and $\mathbb{E}$ is the set of communication links and observation links. Let $S_s(\tau)$ and $S_r(\tau)$ denote the set of selected sensors as sources and relays in an *operational time interval* (defined later) τ, respectively. The set of active sensors in τ is therefore given by $S_a(\tau) = S_s(\tau) \bigcup S_r(\tau)$, where $S_s(\tau), S_r(\tau), S_a(\tau) \subseteq \mathcal{S}$.

We assume that the active/sleep state of each sensor will not change within an operational time interval, i.e., an active sensor is always in active state and a sleeping sensor is always in energy saving state within the time interval. Further, we assume that all sources have the same data generation rate, i.e., all sources use the same sampling frequency, quantization, modulation and coding scheme. Therefore, a fixed amount of bits, denoted by $B(\tau)$, is generated by each source in a time interval τ. Since each sensor has limited energy, if the time interval is too long, it may so happen that a selected active sensor cannot carry out its functionality for generating/relaying data within such an interval. Therefore, we define the *operational time interval* (OTI) as the time duration within which each sensor remains in the same state and each selected active sensor can accomplish its data generating/relaying task using its residual energy.

For simplicity, we consider the following energy consumption model which mainly takes into account the energy consumption for sensing and relaying data. Signalling overheads are not included in the energy consumption model. Let e_s and e_r denote the energy consumed for sensing and receiving a bit, respectively. The energy consumed for transmission depends on the distance between the transmitter and receiver. Let e^t_{ij} denote the energy consumed by sender s_i for transmitting a bit to receiver s_j:

$e^t_{ij} = e_t + b \cdot d^\alpha_{ij}$, where e_t and b are constants, d_{ij} is the Euclidean distance between node s_i and s_j and α is the path loss factor. For simplicity, we omit the sender and receiver id and use e_{trans} to represent e^t_{ij}. A source needs to consume $e_s \times B(\tau)$ energy for its sensing task and at least $e_{trans} \times B(\tau)$ energy for sending out its sensed data in an OTI τ. The energy consumed by a relay is dependent on the number of bits it will transmit or receive in an OTI, where the latter is further dependent on how we construct a connected tree from the sources to the sink. We let $E_0(s)$ to denote the initial energy of a sensor s at the time of the network setup.

Let $\mathcal{T}(\tau) = (S_s(\tau) \bigcup S_r(\tau), \mathbb{E}'(\tau))$ denote the constructed tree in an OTI τ, where $S_s(\tau) \bigcup S_r(\tau)$ is the set of active sensors and $\mathbb{E}'(\tau)$ is the set of edges used to connect the selected active sensors and the sink. The tree $\mathcal{T}(\tau)$ has the following properties:

- (tree property 1) The root of the tree is the sink;
- (tree property 2) Each leaf of the tree is a source sensor;
- (tree property 3) Each target can directly connect to at least one source in the tree.

Such a tree is called as *cover tree* since it covers all the targets and, by definition, a tree is connected. Note that a sensor can act as a source or relay or both. In a cover tree, we call a sensor s_i a *descendant* of another sensor s_j if sensor s_i needs s_j to relay its data to the sink; and s_j is called the *ancestor* of s_i. Let $D(s, \mathcal{T})$ denote the number of sources among the descendants of sensor s in a given cover tree $\mathcal{T}$. If a sensor s' is a leaf, i.e., s' has no descendent, then $D(s', \mathcal{T}) = 0$. Obviously, $D(\mathcal{R}, \mathcal{T}) = |S_s|$. Since all the sensed data should be relayed to the sink, a sensor s in the tree needs

$(e_{trans} + e_s) \times B(\tau) \times D(s, \mathcal{T})$ energy to relay the data from its descendants in an OTI.

From the above discussion, for a given set of sources $S_s(\tau)$, set of relays $S_r(\tau)$ and cover tree $\mathcal{T}(\tau)$ in an OTI, the energy consumption model for a sensor s in the sensor field is given by

$$E(s, \mathcal{T}(\tau)) = \begin{cases} e_s B(\tau) + e_{trans} B(\tau), \quad s \in S_s(\tau) \text{ and } s \notin S_r(\tau); \\ (e_{trans} + e_r) B(\tau) D(s, \mathcal{T}(\tau)), \quad s \in S_r(\tau) \text{ and } s \notin S_s(\tau); \\ (e_s + e_{trans}) B(\tau) + (e_{trans} + e_r) B(\tau) D(s, \mathcal{T}(\tau)), \quad s \in S_s(\tau) \bigcap S_r(\tau); \\ 0, \quad s \notin S_s(\tau) \text{ and } s \notin S_r(\tau). \end{cases} \tag{3.1}$$

Definition 2: Maximum Cover Tree (MCT) Problem: Given a graph $G = (V, \mathbb{E})$ and the initial energy $E_0(s)$ of each sensor s, where $V = \mathcal{S} \bigcup \mathcal{P} \bigcup \mathcal{R}$ and $\mathbb{E}$ is composed of communication links and observation links, find a family of cover trees $\mathcal{T}(\tau_1)$, $\mathcal{T}(\tau_2)$, ..., $\mathcal{T}(\tau_x)$ and their OTIs τ_1, τ_2, ..., τ_x such that the network lifetime, denoted as $T(\mathcal{S}, \mathcal{T}, \mathcal{R})$, is maximized; Mathematically, the MCT problem is defined as:

$$\text{Maximize } \; T(\mathcal{S}, \mathcal{T}, \mathcal{R}) \equiv \sum_{i=1}^{x} \tau_i, \tag{3.2}$$

$$\text{subject to } \; \sum_{i=1}^{x} E(s, \mathcal{T}(\tau_i)) \leq E_0(s), \forall s \in \mathcal{S}. \tag{3.3}$$

In the MCT problem definition, the number of OTIs is denoted by x. Given a finite initial energy and a minimum time duration τ, the value of x is finite but unknown. Further, note that the duration of any two OTIs may be different. Also, a sensor can appear in different trees, i.e., the sets of sensors in different trees need not be disjoint.

3.2.1 Proof of NP-Completeness

First we define the decision version of the MCT problem and then prove that it is NP-Complete. The decision version of the MCT problem is to determine whether there exists a family of cover trees $\mathcal{T}(\tau_1)$, $\mathcal{T}(\tau_2)$, ..., $\mathcal{T}(\tau_x)$ and their OTIs τ_1, τ_2, ..., τ_x such that for a given initial energy of each deployed sensor, the value of $t_1+\cdots+t_x$ is larger than equal to a given value T.

Theorem 1 *The MCT problem is NP-Complete.*

Proof:

Given T and an arbitrary family of cover trees $\mathcal{T}(\tau_1)$, $\mathcal{T}(\tau_2)$, ..., $\mathcal{T}(\tau_x)$ with their OTIs τ_1, τ_2, ..., τ_x, we can verify in polynomial time whether 1) $\sum_{1\leq k\leq x}\tau_k \geq T$, 2) all the targets are covered in each cover tree and 3) the energy consumption of each node s_i over all the trees does not exceed $E_0(s_i)$. Therefore $MCT \in NP$.

To prove that the MCT problem is NP-Hard, we reduce the 3SAT problem to the MCT problem in polynomial-time. Let $U = \{u_1, u_2, \cdots, u_n\}$ be the set of variables and $C = \{c_1, c_2, \cdots, c_m\}$ be the set of clauses in an arbitrary instance of 3SAT.

First, we add a sink node $\mathcal{R}$ into the network. For each variable $u_i \in U$, there is a component (illustrated in Fig. 3.2-(b)), composed of one target p_i, three sensor nodes $\mathcal{S}_i = \{u_i, \overline{u}_i, r_i\}$, three communication links $\mathbb{E}_i^c = \{(u_i, r_i), (\overline{u}_i, r_i), (r_i, \mathcal{R})\}$ and two observation links $\mathbb{E}_i^o = \{(u_i, p_i), (\overline{u}_i, p_i)\}$. For each clause $c_j \in C$ with the three literals $x_j, y_j, z_j \in U \cup \overline{U}$, there is a component (illustrated in Fig. 3.2-(a)) composed of a target c_j, three sensor nodes x_j, y_j and z_j, and three observation links $\mathbb{E}_j^o = \{(x_j, c_j), (y_j, c_j), (z_j, c_j)\}$.

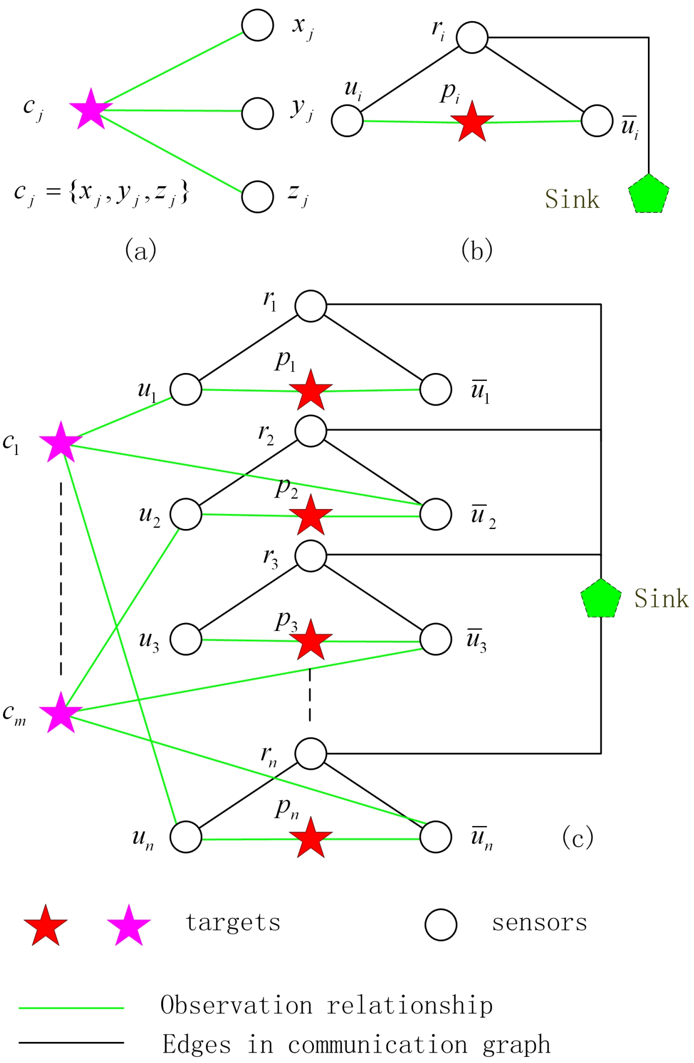

Figure 3.2: Reduction of 3SAT to MCT problem

The construction of our instance of the MCT problem is completed by setting $T = 1$, with the set of sensor nodes $\mathcal{S} = \bigcup_i \mathcal{S}_i$, targets $\mathcal{P} = (\bigcup_i p_i) \cup (\bigcup_j c_j)$, communication links $\mathbb{E}^c = \bigcup_i \mathbb{E}^c_i$ and observation links $\mathbb{E}^o = (\bigcup_i \mathbb{E}^o_i) \cup (\bigcup_j \mathbb{E}^o_j)$ (illustrated in Fig. 3.2-(c)). As an example, in Fig. 3.2-(c), the clause c_1 is assumed to be $c_1 = \{u_1 \vee \bar{u}_2 \vee u_n\}$. In that case, $x_1 = u_1$, $y_1 = \bar{u}_2$ and $z_1 = u_n$. For each sensor node, the initial energy E_0 is 1 unit, $e_r = e_s$ and $(e_r + e_{trans})B(1) = 1$. It is easy to see that the construction can be accomplished in polynomial time.

First, we show that a solution for the 3SAT problem can be transformed to the solution for the MCT problem in polynomial time. Suppose that $t : U \rightarrow \{True, False\}$ is a satisfying truth assignment for C. If $t(u_i) = True$, we assign sensor u_i to be the source sensor; otherwise assign $\overline{u}_i$ to be the source sensor. Node r_i is the relay node to connect u_i or $\overline{u}_i$ to the sink node. It is easy to verify that all the targets are covered, the lifetime is 1 and this can be done in polynomial time.

Now, we show that a solution for the MCT problem can be transformed to a solution for the 3SAT problem. Suppose that cover trees $\mathcal{T}(\tau_1)$, $\mathcal{T}(\tau_2)$, ..., $\mathcal{T}(\tau_x)$ and their OTIs τ_1, τ_2, ..., τ_x is a feasible solution for the MCT problem and $\sum_{1 \leq k \leq x} \tau_k \geq 1$. As target p_i can be covered by sensor u_i and $\overline{u}_i$ only, at least one of them should be active as the source node at any time, and thus r_i must be active as the relay node all the time. Further, u_i and $\overline{u}_i$ cannot be simultaneously chosen as the source nodes, otherwise r_i will consume more than 1 unit of energy if the lifetime is 1. Thus in any cover tree $\mathcal{T}(\tau_x)$, one and only one sensor among sensor u_i and $\overline{u}_i$ (but not both) should be chosen as the source node. Further, if target c_j is covered by a source node $x_j \in U \cup \overline{U}$, the corresponding clause c_j must be true if we set $x_j = 1$. Therefore, assigning the corresponding literals of the source nodes in any cover tree $\mathcal{T}(\tau_k)$ $(1 \leq k \leq x)$ to be true gives a satisfying true assignment for C. Therefore, the MCT problem is NP-Hard.

Since the MCT problem belongs to class NP and is NP-Hard, it is NP-Complete.

∎

3.3 Lifetime upper bound and lower bound

In this section, we develop a linear programming problem formulation, the solution of which can be used as a performance bound for the MCT problem. In OTI τ_k ($1 \leq k \leq x$), for each target p_m, let $\mathcal{S}_m(\tau_k) \subseteq \mathcal{S}_s(\tau_k)$ denote the set of source sensors whose sensing areas cover target p_m. For a given solution of the MCT problem, in OTI τ_k, for each target p_m, we can arbitrarily select exactly one source $s_i \in \mathcal{S}_m(\tau_k)$ and call it as the *solo-observer* of target p_m.

Lemma 1 *There exists an optimal solution of the MCT problem, wherein, during each OTI, no matter how the solo-observers are selected, the set of source nodes is equal to the set of solo-observer nodes.*

Proof: Consider an optimal solution of the MCT problem. From the definition of solo-observer node, each solo-observer node must be the source node. If a source node is not the solo-observer of any target, we can simply remove it from the source set without breaking the target coverage. The removal of such sensors from the source set cannot decrease the lifetime and at the same time it cannot increase the lifetime, because the original solution is optimal. Thus, we get an optimal solution which contains only the solo-observer nodes as source nodes. ∎

Let $B(\tau) = \tau \cdot f_s$, where f_s is the bit rate generated by a source sensor. Let τ_{im} denote the total length of time that sensor s_i is selected as the solo-observer of target p_m, $\max_m(\tau_{im})$ be the maximum total length of time that sensor s_i functions as the solo-observer of a target, F_{ij} be the total amount of data that are transmitted through the link (s_i, s_j) and T be the network lifetime. We introduce the following linear programming (LP) problem and prove that the optimal solution of it is the lifetime upper bound for the MCT problem:

Maximize:

$$T \tag{3.4}$$

Subject to:

$$\sum_i \tau_{im} = T; \qquad \forall p_m \in \mathcal{P} \tag{3.5}$$

$$\sum_j F_{ij} \geq \sum_j F_{ji} + \max_m(\tau_{im}) \cdot f_s; \quad \forall s_i \in \mathcal{S} \tag{3.6}$$

$$\sum_j F_{ij} e^t_{ij} + \sum_j F_{ji} e_r + \max_m(\tau_{im}) \cdot f_s e_s \leq E_0(s_i), \forall s_i \in \mathcal{S} \tag{3.7}$$

Theorem 2 *The optimal solution of the above LP problem is an upper bound of the optimal solution of the MCT problem.*

Proof: If we can prove that there exists an optimal solution of the MCT problem which is also a feasible solution of the LP problem, then the maximum lifetime obtained by solving the LP problem must be an upper bound of the optimal solution of the MCT problem. From Lemma 1, there exists an optimal solution with only the solo-observers as source nodes. Let this optimal solution of the MCT problem be $\{\mathcal{T}(\tau_1), \cdots, \mathcal{T}(\tau_x)\}$ and $\{\tau_1, \cdots, \tau_x\}$.

We have $\tau_{im} = \sum_{1 \leq k \leq x} X^k_{im} \tau_k$, where $X^k_{im} = 1$ if sensor s_i is selected as the solo-observer of target p_m in OTI τ_k; otherwise 0. As in each OTI, exactly one sensor node could be the solo-observer node of each target, $\sum_i \tau_{im} = \sum_k \tau_k \sum_i X^k_{im} = \sum_k \tau_k = T$, and hence Eq. 3.5 must be satisfied. Assuming τ^s_i is the total length of time that node s_i functions as the solo-observer node, we have

$$\max_m(\tau_{im}) \le \tau_i^s = \sum_k \tau_k \times (\bigcup_m X_{im}^k) \le \sum_m \tau_{im} \tag{3.8}$$

From Lemma 1 and the flow conservation, for each node $s_i \in \mathcal{S}$, we have

$$\sum_j F_{ij} = \sum_j F_{ji} + \tau_i^s \cdot f_s \tag{3.9}$$

Using Eq. 3.8 in Eq. 3.9 will satisfy Eq. 3.6. Further, as each node cannot consume more energy than its initial energy, for each node $s_i \in \mathcal{S}$, we have

$$\sum_j F_{ij} e_{ij}^t + \sum_j F_{ji} e_r + \tau_i^s \cdot f_s e_s \le E_0(s_i) \tag{3.10}$$

Using Eq. 3.8 in Eq. 3.10 will satisfy Eq. 3.7.

Thus, the optimal solution of the MCT problem must also be a feasible solution of the LP problem and the theorem is proved. ∎

Corollary 1 *When the maximum number of targets in the sensing area of any sensor is 1, the optimal solution of the LP problem is the optimal solution of the MCT problem.*

Proof: From Eq. 3.8, $\tau_i^s = \max_m(\tau_{im})$, thus the optimal solution of the LP problem is also a feasible solution of the MCT problem. From Theorem 2, the optimal solution of the LP problem is the upper bound of the optimal solution of the MCT problem. Hence proved. ∎

Corollary 2 *When the maximum number of targets in the sensing area of any sensor is $\hat{M}$, the upper bound achieved by solving the LP problem, which is denoted by T_{LP},* *is at most $\hat{M}$ times the optimal solution of the MCT problem. Hence, $T_{LP}/\hat{M}$ is a lower bound for the MCT problem.*

Proof: Given the optimal solution of the LP problem T_{LP} and τ_{im}^{LP}, we can construct a feasible solution of the MCT problem with lifetime $T_{LP}/\hat{M}$. First for each node $s_i \in \mathcal{S}$, we set the duration that it operates as solo-observer for target p_m to be $\tau_{im} = \frac{1}{\hat{M}}\tau_{im}^{LP}$. From the constraints given in Eq. 3.5, we could schedule the sensors as observers such that all the targets could be continuously observed for a period of time $\frac{1}{\hat{M}}T_{LP}$. For example, if target p_m is in the sensing area of $s_1, s_2, \cdots, s_j$, we could randomly select them as the solo-observer for p_m one by one, with each s_i operating for a period of time τ_{im}. If a sensor node is selected as the solo-observer, no matter for a single or multiple targets, it is an active source node. From Eq. 3.8, the total duration that a node s_i operates as the source node (also as the solo-observer node) is $\tau_i^s \le \sum_m \tau_{im} \le \hat{M}\max_m(\tau_{im}) \le \max_m(\tau_{im}^{LP})$. From Eq. 3.6 and 3.7, there exists a solution to route the observed data to the sink such that no node consumes energy more than its initial energy. Therefore, it is a feasible solution for the MCT problem and the achieved network lifetime is at least $T_{LP}/\hat{M}$. ∎

Corollary 2 gives a lower bound for the optimal solution of the MCT problem. Also it provides a basis to design a $\hat{M}$-approximation algorithm for the MCT problem, that is, first solving the LP problem and then selecting the source nodes and their operation durations according to the solution. However, the design of such an algorithm is beyond the scope of this thesis. In this thesis, we focus on an approximation algorithm with a better approximation ratio of $H(\hat{M})(1 + w)$ which is presented in the next chapter (Chapter 4).

3.4 Summary

In this chapter, we introduced the connected target coverage problem and formulated it as the maximum cover tree problem. We proved that MCT problem is NP-complete and provided an upper bound and a lower bound for it by solving the LP problem.

Chapter 4

Approximation and heuristic algorithm for the MCT problem

In this chapter, we develop a $(1+w)H(\hat{M})$ approximation algorithm to solve the MCT problem introduced in the last chapter, where w is an arbitrarily small number, $H(\hat{M}) = \sum_{1 \le i \le \hat{M}} \frac{1}{i}$ and $\hat{M}$ is the maximum number of targets in the sensing area of any sensor. The approach of the approximation algorithm developed by us is to first formulate it as a LP problem with exponential number of variables, and then use a prime-dual approach to develop the approximation algorithm. The idea of solving LP problems approximately originated with the work of [87]. Several works have been carried out in the literature improving and extending these results. Our approximation algorithm follows the algorithm and analysis developed for the flow problems in [88, 89]. Our algorithm is different in that the constraints of our problem are on the nodes. In addition, the concept of weighted greedy algorithm for the minimum weighted set cover problem is used in our approximation algorithm design. We further prove that $(1-O(1))ln(M)$ is a threshold below which the MCT problem cannot be approximated efficiently, unless NP has slightly super-polynomial time algorithms, i.e. $NP \subset TIME(n^{O(loglogn)})$ [90]. As the protocol cost of the approximation algorithm may be high in practice, we develop a faster heuristic algorithm based on the approximation algorithm called Communication Weighted Greedy Cover (CWGC) algorithm and present a distributed implementation of the heuristic algorithm. We study the performance of the approximation algorithm and CWGC algorithm by comparing them with the lifetime upper bound and other basic algorithms that consider the coverage and connectivity problems independently. Simulation results show that the approximation algorithm and CWGC algorithm perform much better than others in terms of the network lifetime and the performance improvement can be up to 45% than the best-known basic algorithm. The lifetime obtained by our algorithms is close to the upper bound. Compared with the approximation algorithm, the CWGC algorithm can achieve a similar performance in terms of the network lifetime with a lower protocol cost.

4.1 Approximation algorithm

4.1.1 LP formulation

The number of variables in the LP problem developed in section 3.3 is polynomial to the MCT problem scale (number of nodes and number of targets). However, it provides only the performance bound but not the solution of the MCT problem. In this section, we formulate the MCT problem as a LP problem. Although the number of variables in this LP may be exponential to the MCT problem scale, this formulation is helpful for developing polynomial approximation algorithm. Given an instance of the MCT problem, let us enumerate all the possible set of sources

$U = \{U_1, U_2, \cdots, U_Q\}$ in the feasible solution of the instance, such that each set $U_q \in U$ contains observing sensors that can cover all the targets. Thus, the source set of any cover tree in the feasible solution of the MCT problem will be in U. Let τ_q denote the duration that U_q is selected as the source set of the cover tree, where q denotes the index of source set in U; Let F_i^s denote the total amount of data that are generated by node s_i when it is selected as the source node; Let N_i denote the set of neighbors of sensor s_i; Let X_{iq} be 1 if node s_i belongs to the source set U_q, otherwise 0. The MCT problem can be formulated as follows:

$$\text{Maximize:} \sum_{1 \le q \le Q} \tau_q \tag{4.1}$$

$$\sum_{1 \le q \le Q} X_{iq} \cdot f_s \tau_q - F_i^s = 0; \quad \forall s_i \in \mathcal{S} \tag{4.2}$$

$$-\sum_{j \in N_i} F_{ij} + \sum_{j \in N_i} F_{ji} + F_i^s - F_{i\mathcal{R}} = 0; \quad \forall s_i \in \mathcal{S} \tag{4.3}$$

$$\sum_{j \in N_i} F_{ij} e_{ij}^t + \sum_{j \in N_i} F_{ji} e_r + F_i^s e_s + F_{i\mathcal{R}} e_{i\mathcal{R}}^t \le E_0(s_i); \forall s_i \in \mathcal{S} \tag{4.4}$$

Equations 4.2 and 4.3 are the flow conservation constraints. Equation 4.4 is the energy consumption constraint.

4.1.2 The dual problem and its interpretation

The dual problem of the above LP problem is as follows:

$$\text{Minimize:} \sum_{1 \le i \le N} c_i E_0(s_i) \tag{4.5}$$

$$-a_i + a_j + e_{ij}^t c_i + e_r c_j \ge 0; \quad \forall i \ne j, s_j \in N_i \tag{4.6}$$

$$e_{i\mathcal{R}}^t c_i - a_i \ge 0; \quad \forall s_i \in N_{\mathcal{R}} \tag{4.7}$$

$$-b_i + a_i + e_s c_i \geq 0; \quad \forall 1 \leq i \leq N \tag{4.8}$$

$$\sum_{1 \leq i \leq N} X_{iq} f_s \cdot b_i \geq 1; \quad \forall 1 \leq q \leq Q \tag{4.9}$$

where $a_i, b_i, c_i \geq 0$ $(1 \leq i \leq N)$ are variables in the dual problem. The dual problem can be interpreted as a problem of assigning weights to the links in the network. Next we define the link weight, node weight and path weight, and then rewrite the dual problem.

Let $\vec{C}$ be a vector whose i^{th} element is c_i. We define the objective function of the dual problem as

$$D(\vec{C}) = \sum_i c_i E_0(s_i) \tag{4.10}$$

In addition, we define the link weight $w_{ij}(\vec{C})$ for each link $(s_i, s_j) \in \mathbb{E}$ and node weight $w_i(\vec{C})$ for each node s_i in the original MCT problem:

$$w_{ij}(\vec{C}) = \begin{cases} e_{ij}^t c_i + e_r c_j, \text{if } j \neq \mathcal{R}, (s_i, s_j) \in \mathbb{E}; \\ e_{i\mathcal{R}}^t c_i, \text{if } j = \mathcal{R}, (s_i, \mathcal{R}) \in \mathbb{E}; \end{cases} \tag{4.11}$$

$$w_i(\vec{C}) = e_s c_i, \forall s_i \in \mathcal{S} \tag{4.12}$$

The dual problem can now be re-written as follows:

$$\text{Minimize:} D(\vec{C}) \tag{4.13}$$

$$w_{ij}(\vec{C}) \geq \begin{cases} a_i - a_j, \quad \text{if } j \neq \mathcal{R}, \text{ link } (s_i, s_j) \in \mathbb{E}; \\ a_i, \quad \text{if } j = \mathcal{R}, \text{ link } (s_i, \mathcal{R}) \in \mathbb{E}; \end{cases} \tag{4.14}$$

$$w_i(\vec{C}) \geq b_i - a_i, \quad \forall s_i \in \mathcal{S} \tag{4.15}$$

$$\sum_{1 \leq i \leq N} X_{iq} f_s \cdot b_i \geq 1; \quad \forall 1 \leq q \leq Q \tag{4.16}$$

Consider an arbitrary path P from node s_i to the sink $\mathcal{R}$. Let P be $\{s_i, n_1, n_2, \cdots, n_l, \mathcal{R}\}$, we define the path weight of P as follows:

$$\begin{aligned} w_P(\vec{C}) &= w_{s_i}(\vec{C}) + w_{s_i,n_1}(\vec{C}) \\ &+ \sum_{1\le z<l} w_{n_z,n_{z+1}}(\vec{C}) + w_{n_l,\mathcal{R}}(\vec{C}) \end{aligned} \tag{4.17}$$

Using Eq. 4.14 and 4.15, we have

$$w_P(\vec{C}) \ge b_i - a_i + a_i - a_{n_1} + \cdots + a_{n_l} = b_i \tag{4.18}$$

Let $w^i_{SPT}(\vec{C})$ denote the path weight of the shortest path (path with the minimum path weight) from node s_i to the sink. We define

$$\alpha(\vec{C}) \equiv \min_{1\le q\le Q} \left\{ \sum_i X_{iq} f_s \cdot w^i_{SPT}(\vec{C}) \right\} \tag{4.19}$$

$$\ge \min_{1\le q\le Q} \left\{ \sum_i X_{iq} f_s \cdot b_i \right\} \tag{4.20}$$

$$\ge 1 \tag{4.21}$$

The dual problem is then equivalent to assigning values to $\vec{C}$ such that $D(\vec{C})/\alpha(\vec{C})$ is minimized subject to the constraint that $\alpha(\vec{C}) \ge 1$. Let

$$\beta = \min \left\{ \frac{D(\vec{C})}{\alpha(\vec{C})} \right\} \tag{4.22}$$

4.1.3 Algorithm description

Let $\mathcal{P}_i$ be the set of targets that are in the sensing area of sensor s_i. The above interpretation of the dual LP problem leads to our approximation algorithm, which is described as follows:

1. Initialization

   (a) Properly scale the problem so that $\beta \geq 1$;

   (b) $t = 0$; $T = 0$; Set $\delta = (\frac{N}{1-\epsilon H(\hat{M})})^{-1/\epsilon}$; For each node s_i set $c_i = \delta / E_0(s_i)$; Let $\lambda = \log_{1+\epsilon}^{1/\delta}$; Set $\tau_p = 1/\lambda$;

2. Set $\tau^t = 0$, $k = 0$, loop until $\tau^t = \tau_p$;

   (a) $k = k + 1$; Build the shortest path tree rooted at the sink with the link weight function $w_{ij}(\vec{C})$ and the path weight function $w_P(\vec{C})$; Set $S_k^t = \phi$ as the observing sensor set and $\mathcal{P}_i^{tk} = \mathcal{P}_i$ as the set of uncovered targets in each node s_i's sensing area.

   (b) Until all the targets are covered by S_k^t do the following:

      i. select sensor $s_i \notin S_k^t$ which has the minimum value of $\frac{w_{SPT}^i(\vec{C})}{|\mathcal{P}_i^{tk}|}$ and add it into S_k^t.

      ii. for each sensor $s_j \notin S_k^t$, $\mathcal{P}_j^{tk} = \mathcal{P}_j^{tk} - \mathcal{P}_i^{tk} \cap \mathcal{P}_j^{tk}$.

   (c) The shortest path from each sensor in S_k^t to the sink forms the cover tree $\mathcal{T}_k^t$, and the operation duration τ_k^t of $\mathcal{T}_k^t$ ends when $\tau_k^t = \tau_p - \tau^t$ or any node s_i in $\mathcal{T}_k^t$ consumes $E_0(s_i)/\lambda$ unit of energy;

   (d) $\tau^t = \tau^t + \tau_k^t$; Let e_i^{tk} denote the amount of energy that node s_i has consumed in duration τ_k^t, $c_i(t,k) = c_i(t,k-1) \times (1 + \epsilon \cdot \frac{\lambda e_i^{tk}}{E_0(s_i)})$.

3. $t = t + 1$; $c_i(t, 0) = c_i(t-1, k)$; if $D(\vec{C}) < 1$, $T = T + \tau_p$.

4. repeat step 2 and step 3 until $D(\vec{C}) \geq 1$; double τ_p for every 2λ iterations.

The output of the algorithm is T, $\mathcal{T}_k^t$ and τ_k^t, which are the network lifetime, cover trees and their operation durations, respectively. At the beginning, we scale the problem such that $\beta \geq 1$. This scaling is useful for our analysis for computing approximation ratio as explained later. We set the initial value for δ, λ and c_i. The algorithm then proceeds in loops. Let us call the outer loop of steps 2 and 3 as an iteration, and call the inner loop of steps 2a, 2b, 2c and 2d as a phase in the iteration. The duration of each iteration is τ_p. Each iteration may be composed of multiple phases. In each phase, we try to build a cover tree, such that the total path weight from all the source nodes to the sink is minimized. The duration of the phase (OTI of the cover tree) ends when the duration of the iteration ends or any node s_i consumes $E_0(s_i)/\lambda$ units of energy in the phase. Here we use the concept of weighted greedy algorithm for minimum weighted set cover problem to select the source set. Considering $\mathcal{P}_i$ as the subset and $w^i_{SPT}(\vec{C})$ as the subset weight, we greedily select the sensor that has the minimum value of $\frac{w^i_{SPT}(\vec{C})}{|\mathcal{P}_i|}$ as the source node until all the targets are covered. The shortest path from each selected source node to the sink builds the cover tree. The value of c_i will be updated according to the energy consumption of node s_i in the phase.

We will first present the method to scale the problem (in step 1a), determine the approximation ratio and discuss the value of λ, δ in the next section (section 4.1.4). Then we explain why τ_p is doubled (in step 4) and analyze the complexity in section 4.1.5.

4.1.4 Analysis

We first explain the scaling method used in our algorithm so that $\beta \geq 1$. Let T_{LP} denote the lifetime upper bound achieved by solving the LP problem presented in section 3.3, from theorem 2 and corollary 2, we have $T_{LP}/\hat{M} \leq \beta \leq T_{LP}$. Thus scaling the initial energy of each node by $T_{LP}/\hat{M}$ or increasing f_s by a factor $T_{LP}/\hat{M}$ can guarantee that $\hat{M} \geq \beta \geq 1$. With $\beta \geq 1$, we carry out the analysis as given below.

Using Eq. 4.11 and Eq. 4.12 in the definition of path weight $w_P(\vec{C})$ (Eq. 4.17), we have

$$w_P(\vec{C}) = (e_s + e^t_{i,n_1})c_i + (e_r + e^t_{n_l,\mathcal{R}})c_{n_l} + \sum_{1 \leq z < l} (e_r + e^t_{n_z,n_{z+1}})c_{n_z} \tag{4.23}$$

Let K_t denote the number of phases in iteration t, $\vec{C}(t,k)$ denote the vector of c_i after the k^{th} phase of iteration t. Assuming that the q^{th} source set in U is selected as the source set in the k^{th} phase of iteration t, the value of $D(\vec{C})$ at the end of this phase is

$$D(\vec{C}(t,k)) = \sum_i c_i(t,k)E_0(s_i) \tag{4.24}$$

$$= \sum_i (c_i(t,k-1))E_0(s_i)(1 + \epsilon\frac{\lambda e^{tk}_i}{E_0(s_i)}) \tag{4.25}$$

$$= D(\vec{C}(t,k-1)) + \epsilon \sum_i \lambda e^{tk}_i c_i(t,k-1) \tag{4.26}$$

$$= D(\vec{C}(t,k-1)) + \epsilon f_s \tau^t_k \lambda \sum_i X_{iq} w^i_{SPT}(\vec{C}(t,k-1)) \tag{4.27}$$

where $\sum_i X_{iq} w^i_{SPT}(\vec{C}(t,k-1))$ is the total path weight from all the source nodes in U_q to the sink. Let $w_{min}(t,k-1)$ denote the minimum value of the total path weight for all the source sets in U. As greedy weighted algorithm is a $H(k)$ approximation algorithm [91] where $H(k) = \sum_{1 \le i \le k} \frac{1}{i}$ and k is the maximum subset size, we have

$$\sum_i X_{iq} w_{SPT}^i(\vec{C}(t, k-1)) \le H(\hat{M}) w_{min}(t, k-1) \tag{4.28}$$

where $\hat{M}$ is the maximum number of targets in the sensing area of any sensor. As the algorithm proceeds, the link weights are monotonically non-decreasing. Therefore,

$$w_{min}(t, k-1) \le w_{min}(t, k) \tag{4.29}$$

Further, using the definition of $\alpha(\vec{C})$ in Eq.4.19,

$$\alpha(\vec{C}) = f_s w_{min}(\vec{C}) \tag{4.30}$$

For any iteration $t \ge 1$, using Eq.4.27, Eq.4.28 and Eq. 4.29,

$$\begin{aligned} D(\vec{C}(t,0)) &= D(\vec{C}(t-1,0)) + \epsilon f_s \lambda \sum_{1 \le k \le K_{t-1}} \tau_k^{t-1} \\ &\quad \cdot \sum_i X_{iq} w_{SPT}^i(\vec{C}(t-1, k-1)) && (4.31) \\ &\le D(\vec{C}(t-1,0)) + \epsilon f_s \lambda \sum_{1 \le k \le K_{t-1}} \tau_k^{t-1} \\ &\quad \cdot H(\hat{M}) w_{min}(t-1, k-1) && (4.32) \\ &\le D(\vec{C}(t-1,0)) + \epsilon f_s \lambda \sum_{1 \le k \le K_{t-1}} \tau_k^{t-1} H(\hat{M}) w_{min}(t, 0) && (4.33) \end{aligned}$$

If τ_p is never doubled, $\sum_{1 \le k \le K_{t-1}} \tau_k^{t-1} = \tau_p = 1/\lambda$. We assume that τ_p is never doubled now and will explain latter why the approximation ratio still holds when this assumption is removed in section 4.1.5. From Eq. 4.30,

$$D(\vec{C}(t,0)) \le D(\vec{C}(t-1,0)) + \epsilon H(\hat{M}) \alpha(\vec{C}(t,0)) \tag{4.34}$$

Since $\beta = \min \left\{ D(\vec{C}) / \alpha(\vec{C}) \right\} \le \frac{D(\vec{C}(t,0))}{\alpha(\vec{C}(t,0))}$ we have

$$D(\vec{C}(t,0)) \le \frac{D(\vec{C}(t-1,0))}{1 - \epsilon H(\hat{M})/\beta} \tag{4.35}$$

Since $D(\vec{C}(0,0)) = N\delta$, for any iteration $t \geq 1$,

$$D(\vec{C}(t,0)) \leq \frac{N\delta}{(1-\epsilon H(\hat{M})/\beta)^t} \tag{4.36}$$

$$= \frac{N\delta}{1-\epsilon H(\hat{M})/\beta}(1+\frac{\epsilon H(\hat{M})}{\beta-\epsilon H(\hat{M})})^{(t-1)} \tag{4.37}$$

$$\leq \frac{N\delta}{1-\epsilon H(\hat{M})/\beta} e^{\frac{\epsilon H(\hat{M})(t-1)}{\beta-\epsilon H(\hat{M})}} \tag{4.38}$$

$$\leq \frac{N\delta}{1-\epsilon H(\hat{M})} e^{\frac{\epsilon H(\hat{M})(t-1)}{\beta(1-\epsilon H(\hat{M}))}} \tag{4.39}$$

The last inequality uses the assumption that $\beta \geq 1$.

Let N_t denote the iteration that the algorithm ends, i.e. $D(\vec{C}(N_t,0)) \geq 1$.

$$1 \leq D(\vec{C}(N_t,0)) \leq \frac{N\delta}{1-\epsilon H(\hat{M})} e^{\frac{\epsilon H(\hat{M})(N_t-1)}{\beta(1-\epsilon H(\hat{M}))}} \tag{4.40}$$

Therefore, we have

$$\frac{\beta}{N_t-1} \leq \frac{\epsilon H(\hat{M})}{(1-\epsilon H(\hat{M}))\ln(\frac{1-\epsilon H(\hat{M})}{N\delta})} \tag{4.41}$$

In each iteration, the network lifetime will be increased by a duration of $1/\lambda$. Since the algorithm ends when $D(\vec{C}(N_t,0)) \geq 1$, the network lifetime is $T = (N_t-1)/\lambda$.

Lemma 2 *The solution of our approximation algorithm is a feasible solution for the MCT problem, $(N_t-1)/\lambda$ is strictly less than the optimal solution of the MCT problem.*

Proof: The flow conservation constraints are not violated in our algorithm, thus if the energy consumption constraints are not violated, the solution of our algorithm $(N_t-1)/\lambda$ is a feasible solution.

Consider an arbitrary node $s_i \in \mathcal{S}$. In any phase k of any iteration t, the energy consumption of s_i will be less than equal to $E_0(s_i)/\lambda$. Thus for every $E_0(s_i)/\lambda$ units of energy consumed by node s_i, the value of c_i will be increased by at least a factor $(1+\epsilon)$. In other words, if c_i is increased by a factor $(1+\epsilon)$, the energy consumption of node s_i will be at most $E_0(s_i)/\lambda$. Initially, c_i equals to $\delta/E_0(s_i)$. In the iteration before the algorithm ends, as $D(\vec{C}(N_t-1,0)) = \sum_i c_i(N_t-1,0)E_0(s_i) < 1$, we have $c_i(N_t-1,0) < 1/E_0(s_i)$ for any node s_i. Therefore, the total amount of energy consumption of node s_i is strictly less than $\log_{1+\epsilon} \frac{1/E_0(s_i)}{\delta/E_0(s_i)} \times E_0(s_i)/\lambda = E_0(s_i)$. Hence proved. ∎

Theorem 3 *Our algorithm is a $H(\hat{M})(1+w)$ approximation for the MCT problem.*

Proof: Let γ denote the approximation ratio. Using Eq. 4.41 and from Lemma 2, we have

$$\begin{aligned} \gamma < \frac{\beta}{(N_t-1)/\lambda} &\le \frac{\epsilon H(\hat{M}) \log_{1+\epsilon} 1/\delta}{(1-\epsilon H(\hat{M})) \ln(\frac{1-\epsilon H(\hat{M})}{N\delta})} \\ &= \frac{\epsilon H(\hat{M})}{(1-\epsilon H(\hat{M})) \ln(1+\epsilon)} \frac{\ln(1/\delta)}{\ln(\frac{1-\epsilon H(\hat{M})}{N\delta})} \end{aligned} \tag{4.42}$$

As $\delta = (\frac{N}{1-\epsilon H(\hat{M})})^{-1/\epsilon}$,

$$\gamma \le \frac{\epsilon H(\hat{M})}{(1-\epsilon H(\hat{M}))(1-\epsilon)\ln(1+\epsilon)} \tag{4.43}$$

$$\le \frac{\epsilon H(\hat{M})}{(1-\epsilon H(\hat{M}))(1-\epsilon)(\epsilon-\epsilon^2/2)} \tag{4.44}$$

$$\le H(\hat{M})(1-\epsilon H(\hat{M}))^{-1}(1-\epsilon)^{-2} \tag{4.45}$$

$$= H(\hat{M})(1+w) \tag{4.46}$$

Hence proved. ∎

4.1.5 Complexity Analysis

From lemma 2,

$$1 \leq \gamma < \frac{\beta}{(N_t - 1)/\lambda} \quad \Rightarrow \quad N_t < 1 + \beta\lambda \tag{4.47}$$

Therefore, the total number of iterations N_t until the approximation algorithm terminates is strictly less than $1 + \beta\lambda$. In fact, if our algorithm doesn't terminate after $2\lceil\lambda\rceil$, we know $\beta \geq 2$ and we can double the duration of iterations τ_p. Note that this is equivalent to re-scaling the problem. β will be half of its previous value but still larger than 1, and therefore the approximation ratio still holds. As we repeat this procedure until the algorithm terminates, the approximation algorithm will terminate in $2\log_2 \hat{M}\lceil\lambda\rceil$ iterations ($\beta \leq \hat{M}$ after scaling).

We note that in each phase of an iteration, except for the last phase, there exists at least one node s_i that consumes energy $E_0(s_i)/\lambda$, whose c_i is increased by a factor $1 + \epsilon$. Since for any node s_i, the initial value of c_i is $\delta/E_0(s_i)$ and the final value is less than $1/E_0(s_i)$ (for $D(\vec{C}) < 1$), the number of phases exceeds the number of iterations by at most $N \log_{1+\epsilon} \frac{1}{\delta} = N\lambda$ (otherwise there exists at least one sensor s_i whose c_i exceeds $1/E_0(s_i)$). In each phase we build a shortest path tree and greedily select the source nodes until all the targets are covered, which requires $O(N^2)$ and $O(N \min(M, N))$ time, respectively. Therefore, the time complexity of our algorithm is $(2\log_2(\hat{M}) + N)\lceil\frac{1}{\epsilon}\log_{1+\epsilon}(\frac{N}{1-H(\hat{M})\epsilon})\rceil O(N^2 + N\min(M, N))$.

4.2 Inapproximality of the MCT problem

Theorem 4 *There does not exist a polynomial time algorithm which can approximate the MCT problem within* $(1 - \epsilon)ln(M)$*, for any constant* $\epsilon > 0$*, unless* $NP \subset TIME(n^{O(loglogn)})$.

Proof: We prove the theorem by reducing each instance of the minimum set cover (MSC) problem to the MCT problem. Given a collection C of subsets of a finite set S, the minimum set cover problem ties to find a set cover for S, i.e., a subset $C' \subseteq C$ such that every element in S belongs to at least one member of C', with the minimum cardinality of the set cover, i.e., $|C'|$. Given an instance of the minimum set cover problem S, C, assuming that $|S| = M$ and $|C| = N$, we construct the corresponding MCT instance as shown in Fig. 4.1 and illustrate the construction below.

For each set $c_i \in C$, we construct a sensor s_i; for each element $s_m \in S$, we construct a target p_m. If $s_m \in c_i$, we construct an observation link (p_m, s_i). A sensor node $\bar{s}$ is added with communication links connecting to all the sensors $s_1, \cdots, s_N$. A sink node R is added with a communication link $(\bar{s}, R)$. For each sensor node, the initial energy E_0 is 1 unit, $e_r = e_s$ and $(e_r + e_{trans})B(1) = 1$. It is easy to see that the construction can be accomplished in polynomial time. Also it is easy to examine that for each cover tree in the solution of the MCT instance, the corresponding sets of the sources in the cover tree build a set cover of the MSC instance. Further, for each set cover of the MSC instance, a cover tree for the MCT instance can be built by connecting the corresponding sources of the sets in the set cover to $\bar{s}$, which connects $\mathcal{R}$.

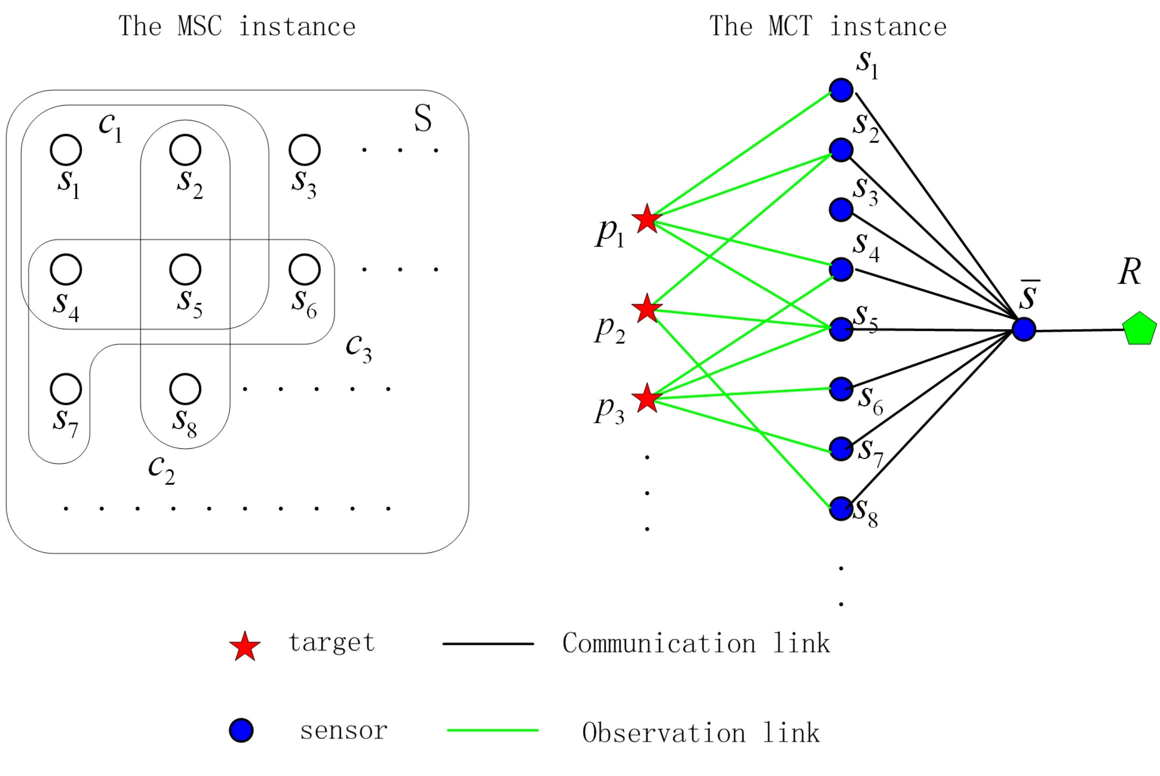

Figure 4.1: Construction of the MCT instance for a given MSC instance

As $\bar{s}$ is the single node connected with the sink, $\bar{s}$ should be active as the relay node in all the cover trees. Given any cover tree $\mathcal{T}$ with L source nodes, $\bar{s}$ will consume the energy L times of each source node. Clearly, the network lifetime is maximized when the cover tree with the minimum number of source nodes operates for the whole network lifetime. Let the minimum number of sources in a cover tree be L_{min}, the optimal solution for the MSC instance will be L_{min}, and the optimal network lifetime for the MCT instance will be $1/L_{min}$.

Suppose that there exists a polynomial-time algorithm which can approximate the MCT problem within r. Then for each instance of the MSC problem, we can construct the corresponding MCT instance and find a solution with lifetime larger than $1/rL_{min}$. There must exist a cover tree in the solution with no more than rL_{min}

source nodes. And thus, we find a set cover for the MSC instance with cardinality no more than r times of the minimum cardinality. As the reduction is in polynomial time, we also find a polynomial time algorithm which can approximate the MSC problem within r.

In [90] it has been shown that if there exists some $\epsilon > 0$ such that a polynomial time algorithm can approximate MSC problem within $(1-\epsilon)ln(|C'|)$, then $NP \subset TIME(n^{O(loglogn)})$. Hence the theorem is proved. ∎

4.3 Communication Weighted Greedy Cover algorithm

4.3.1 Motivation

The approximation algorithm provides useful theoretical insights into the MCT problem. On the other hand, the number of cover trees generated could be large, because, to achieve satisfactory results, we need to set ϵ to be small, which results in a small δ and large λ. As generating a new cover tree will incur protocol cost, e.g. exchanging the states of a node among neighbors and broadcasting the operational duration of the cover tree, the protocol cost of the approximation algorithm will be high. Therefore, we develop a faster low-cost heuristic algorithm for the MCT problem.

The similarities and differences between the heuristic algorithm and the approximation algorithm are listed below:

1. In the approximation algorithm, there are two nested loops. There may be multiple inner loops (phases) in each outer loop (iteration). The duration of each iteration is fixed while the duration of each phase can vary according to

the tree covers being generated. In each phase, no node can consume energy more than $E_0(s_i)/\lambda$. In the heuristic algorithm, we modify the nested loops into a single loop, and set the duration of the loop as a fixed value unless any node dies. This modification can greatly decrease the number of cover trees to be built and thus reduce the protocol cost. In addition, broadcasting the operational duration of each newly built cover tree is not necessary now as the duration is fixed and the duration can be broadcasted once at the beginning.

2. We modify the link weight function of the approximation algorithm. Instead of using c_i to calculate the link weight and increasing c_i after each new cover tree is built, we directly use each node's current energy to calculate the link weight. Further, as the energy consumption of data transmission is normally larger than that of data reception, we calculate the link weight considering the state of the sender only, and thus each node can update the link weight originating from itself without tracking the state of its neighbors.

3. Different from the approximation algorithm which terminates when $D(C) \geq 1$, the heuristic algorithm terminates when no new cover tree can be built, that is, either some targets cannot be covered or the selected source cannot connect to the sink.

4.3.2 Heuristic algorithm description

The heuristic algorithm uses a greedy method to select the source set to cover the targets and it couples the communication cost and source set selection. Hence it is called *Communication Weighted Greedy Cover* (CWGC). The inputs of the algorithm are $\mathcal{S}$, $\mathcal{P}$, $\mathcal{R}$ and $E_0(s_i)$ of each sensor s_i. The output of the algorithm is a sequence of cover trees $\mathcal{T}_1, \cdots, \mathcal{T}_x$ and their OTIs $\tau_1, \cdots, \tau_x$. The position of the cover tree $\mathcal{T}_x$ in the sequence is denoted as x.

The pseudo-code for the algorithm is shown in Table 4.1. In the table, the following notations are used:

$\mathcal{S}_l$	set of live sensors;
$\mathcal{S}_s$	set of live sensors that can cover targets;
P_s	set of targets that can be covered by s;
w_s	path weight of sensor s in MWCT;
$W(s)$	profit of sensor $s \in \mathcal{S}_s$;
$R(s, \mathcal{T})$	route from the sensor s to the sink $\mathcal{R}$ in tree $\mathcal{T}$, $R(s, \mathcal{T}) \equiv < s, s_1, \cdots, \mathcal{R} >$
$\overline{R}(s, \mathcal{T})$	set of sensors in route $R(s, \mathcal{T})$ excluding sensor s and sink $\mathcal{R}$;

The algorithm is initialized before constructing cover trees (lines 1-5). The algorithm repeatedly builds cover trees and stops until no new cover tree can be built (i.e., the network lifetime is reached). Each cover tree operates for a fixed time duration τ, unless some sensors in the cover tree will die before the end of the time duration due to the lack of energy. In that case, the operational time duration of the cover tree is determined by the sensor which has the least operational time until death, i.e.,

Table 4.1: Pseudo-codes for the CWGC algorithm

$\mathcal{S}_l = \mathcal{S}$; $\mathcal{S}_s = \emptyset$; $x = 1$;
for each $s \in \mathcal{S}_l$,
  $E_r(s) = E_0(s)$;
  if $P_s \neq \emptyset$, $\mathcal{S}_s = \mathcal{S}_s \bigcup \{s\}$; **endif**
endfor
while $\bigcup_{s \in \mathcal{S}_s} P_s = \mathcal{P}$ **and** $\mathcal{S}_l \neq \emptyset$,
phase 1:
  for each link (s_i, s_j), $w_{ij} = e^t_{ij} \times E_0(s_i)/E_r(s_i)$; **endfor**
  Build a MWCT $\mathcal{T}_m$ connecting each sensor $s \in \mathcal{S}_l$ to the sink
phase 2:
  $S'_s = \emptyset$; $P' = \emptyset$; $\mathcal{T}_x = \emptyset$; $\tau_x = \tau$;
  while $P' \neq \mathcal{P}$,
    Find a sensor $s^* \in \mathcal{S}_s - S'_s$ with the maximum profit $W(s^*)$
    $S'_s = S'_s \bigcup \{s^*\}$; $P' = P' \bigcup P_{s^*}$;
    for each $s \in \overline{R}(s^*, \mathcal{T}_x)$,
      $w_s = w_s + (e_{trans} + e_r)B(\tau) \times w_s/E_r(s)$;
    endfor
  endwhile
phase 3:
  for each $s \in S'_s$, $\mathcal{T}_x = \mathcal{T}_x \biguplus R(s, \mathcal{T}_c)$; **endfor**
  for each $s \in \mathcal{T}_x$, $\tau_x = \min(\tau_x, \frac{E_r(s)}{E(s, \mathcal{T}_x(\tau_x))}\tau_x)$; **endfor**
  for each $s \in \mathcal{T}_x$, $E_r(s) = E_r(s) - E(s, \mathcal{T}_x(\tau_x))$; **endfor**
  Remove dead and isolated nodes; $x = x + 1$
endwhile

$\min_{s \in \mathcal{T}_k}(\frac{E_r(s)}{E(s,\mathcal{T}_k(\tau))}\tau)$. Thus, the active time of a cover tree $\mathcal{T}_k$ is given by

$$\tau_k = \min(\tau, \min_{s \in \mathcal{T}_k}(\frac{E_r(s)}{E(s, \mathcal{T}_k(\tau))}\tau)) \tag{4.48}$$

where $E_r(s)$ is the residual energy of sensor s at the beginning of operating cover tree $\mathcal{T}_k$.

If a sensor has no residual energy, we call it a *dead* sensor. If a sensor has residual energy but cannot find a route from itself to the sink without traversing a dead sensor, we call it an *isolated* sensor. Before each iteration of building a new cover tree, the dead sensors and isolated sensors in the network will be removed. The graph used to build the new cover tree contains only the live sensors.

In each iteration, the algorithm works in three phases to construct a cover tree. In the first phase (phase 1), an energy-aware communication tree is constructed connecting all the live sensor nodes to the sink. The algorithm then greedily selects source sensors that can cover all the targets (phase 2), considering both the number of uncovered targets in the sensing area and the possible communication cost from sensors to the sink. Finally, in phase 3, the new cover tree is constructed based on the communication tree built in phase 1 and the source sensor set selected in phase 2.

In phase 1, we first assign a weight w_{ij} to each link between live sensors s_i and s_j (line 8) which reflects both the communication energy consumption on the link and the residual energy level of the sender. We use

$$w_{ij} = e^t_{ij} \times E_0(s_i)/E_r(s_i) \tag{4.49}$$

A *minimum weight communication tree* (MWCT) is then constructed connecting all sensors such that the sum of the link weights from each node to the sink is minimized (line 9). Known techniques to find the shortest path tree (e.g. Dijkstra's Algorithm) could be used to construct the tree.

In phase 2, we use a greedy method to choose the sources until all the targets are covered. The greedy method first assigns each sensor s a *profit* value $W(s)$, and then repeatedly chooses the sensor with the highest $W(s)$ into the source set (lines 13-17).

The profit function which is used in our algorithm is given by

$$W(s) \equiv \frac{|P_s - P_s \bigcap P'|}{w_s} \tag{4.50}$$

where $|P_s - P_s \bigcap P'|$ is the number of uncovered targets in the sensing area of node s and w_s is the path weight of node s (the sum of link weights in the route $R(s, \mathcal{T})$). After a new source is selected, the path weights of the upstream nodes in MWCT are updated (line 16).

In phase 3, the cover tree is extracted as a sub-tree of the MWCT based on the selected sources (line 20). After the cover tree is built, we use the energy consumption model (given in Eq. 3.1) for each sensor in the cover tree. The operation duration for the cover tree is then calculated using Eq. 4.48 (line 21). Finally, The residual energy for sensors in the cover tree is updated according to their functionalities, and the dead and isolated sensors are removed (line 23).

Let T_{LP} denote the lifetime upper bound achieved by solving the LP problem presented in section 3.3. Each cover tree operates for a duration τ otherwise at least one sensor will die. Therefore, the number of cover trees to be built in the heuristic algorithm is upper bounded by $N + \frac{T_{LP}}{\tau}$.

4.3.3 Distributed implementation

In practice, if the sensors can identify the targets in their sensing area, e.g. through target localization, the CWGC algorithm can be easily implemented in an on-line distributed manner. Before each OTI, all the sensor nodes will wake up to become active. Each node computes the weight of links originated from itself to its neighbors

(Eq. 4.49) and builds MWCT using any known distributed shortest path tree algorithm, e.g. distributed Dijkstra algorithm. Each node then knows the minimum path weight from itself to the sink node and computes its profit value (Eq. 4.50).

For each node s_i, we define set $S_n(i)$ which contains sensors that cover at least one target which is also covered by node s_i. If node s_i has the largest profit value among all the nodes in $S_n(i)$, node s_i should be chosen as the source node before any node in $S_n(i)$. Therefore, each node s_i broadcasts its profit value to all the nodes in $S_n(i)$. If all the profit values received by s_i are less than its own profit value, s_i will broadcast a message declaring itself as the source node to all the nodes in $S_n(i)$. It will then send a message to the sink node such that all its ancestors are notified to be the relay nodes and update their path weights. If the profit value of s_i is not the largest, it examines the number of uncovered targets in its sensing area and recalculates its profit value. This procedure is repeated with non-source sensors that still have uncovered targets in the sensing area.

A sensor node estimates its energy consumption using Eq. 3.1 if it is selected to be the source node or receives a message notifying it as the relay node. If any sensor forecasts that it will consume more energy than its residual energy, it will calculate the new OTI duration τ and broadcast it to all the other sensors. A sensor will wait for a period of time (sufficient to complete the building of cover tree) after all the targets in its sensing area are covered to determine the length of the next OTI duration. Then it goes into sleep until the end of the OTI duration.

4.4 Performance Study

In this section we evaluate the performance of the approximation algorithm and the proposed CWGC algorithm. The initial energy of each sensor is set to be $20J$; the value of various parameters are chosen to be $e_t = 50nJ/bit$, $b = 100pJ/bit/m^4$, $\alpha = 4$, $e_r = 150nJ/bit$ and $e_s = 150nJ/bit$ [69]; and data is generated by each source node at the rate of $10Kbps$. In the simulation, we assume that each sensor covers a disk centered at itself with a fixed *sensing range* as the disk radius. All sensors are assumed to have the same sensing range R_s and the same maximum communication range R_c.

The approximation algorithm presented in section 4.1 terminates when $D(C) \geq 1$. However, when the algorithm terminates, all the nodes are still alive and the network can still operate as shown in lemma 2. Therefore, we extend the approximation algorithm such that the algorithm terminates when some node consumes all its battery energy. Clearly, the lifetime achieved by the extended algorithm is larger than equal to that achieved by the original algorithm, and thus it can achieve at least the same approximation ratio. We call the original approximation algorithm as App_MCT and the extended one as App_MCT_Ext.

To demonstrate the effectiveness of our algorithms (App_MCT, App_MCT_Ext and CWGC), for each topology generated, we compute the lifetime upper bound achieved by solving the LP problem developed in section 3.3 using CPLEX. Further, to demonstrate the superiority of our algorithms, we compare their performance with three other basic algorithms: RANDOM, MSC_SPT and MSC_EAWARE. The *RANDOM* algorithm randomly chooses a set of sensors which can cover all the targets.

The set is selected such that no sensor in the set can be removed without breaking the coverage of the targets. Each source sensor will transmit its sensed data to the sink using the shortest path (path with minimum communication energy consumption). The *MSC_SPT* algorithm also uses the shortest path to transmit the sensed data to the sink, but uses a greedy method to select the source set. It repeatedly selects the node that covers the most uncovered targets as the source node until all the targets are covered. The *MSC_EAWARE* algorithm chooses the set of nodes in the same way as *MSC_SPT* algorithm. However, instead of using the shortest path to transmit data, it uses an energy-aware communication tree which is built in a similar way as our CWGC algorithm. The difference between this algorithm and the CWGC algorithm is that in the CWGC algorithm the selection of the source nodes is coupled with the communication tree construction. Finally, as the connectivity issue is not considered in most existing works on discrete target problem, we compare the performance of our algorithms with the *greedy_MSC* algorithm proposed in [16] with a suitable modification to account for the connectivity. This modification is done by using a shortest path tree to transmit the sensed data to the sink node. We refer the modified algorithm as Greedy_MSC_SPT. The *greedy_MSC* algorithm greedily selects a "critical" target and then selects the sensor with the greatest contribution to the "critical" target until all the targets are covered. The "critical" target is chosen as the target in the sensing area of the least number of sensors, and the contribution function is chosen as the number of uncovered targets in the sensing area of a sensor.

We use the following simulation scenario. Sensors are randomly deployed in a $100m \times 100m$ area. The sink node is placed in the middle of the area [at point

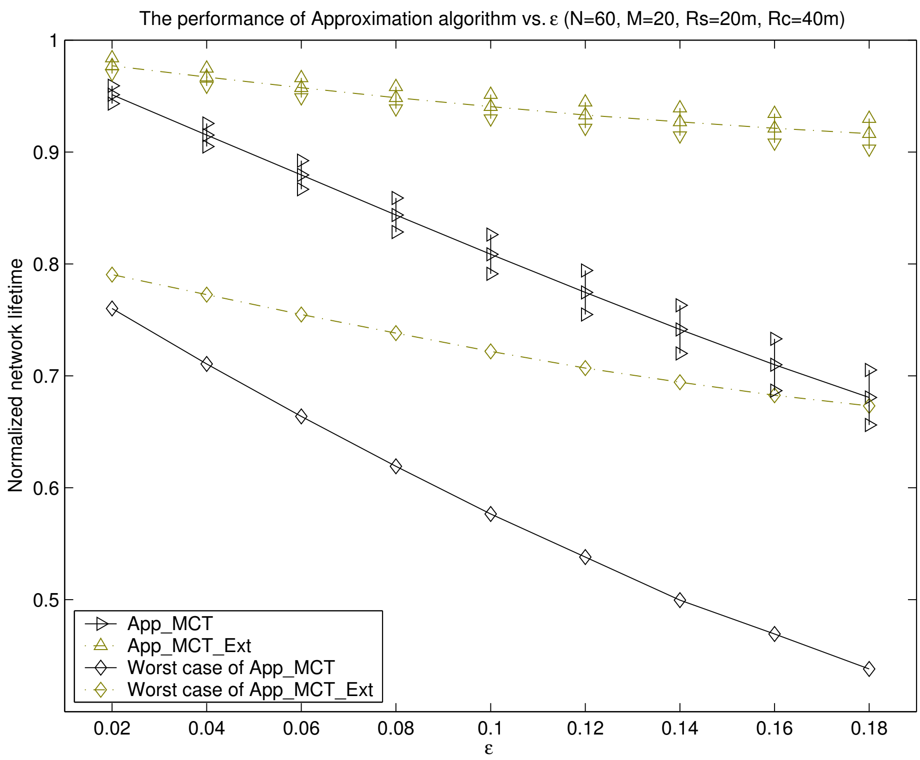

Figure 4.2: Normalized lifetime vs. ϵ ($N = 60$, $M = 20$)

$(50m, 50m)$]. Each value plotted on the curves is obtained from the results of one hundred random topologies.

4.4.1 Impact of algorithm parameters

First we study the performance of App_MCT and App_MCT_Ext by varying the value of ϵ. Sixty sensor nodes ($N = 60$) and 20 targets ($M = 20$) are randomly scattered in the area. The sensing range and the communication range is set as $R_s = 20m$ and $R_c = 40m$. For each topology the network lifetime achieved by the algorithms is normalized by the lifetime upper bound achieved by solving the LP problem. Figure 4.2 shows the average and worst case normalized lifetime achieved by App_MCT and App_MCT_Ext with different values of ϵ. For the average normalized lifetime,

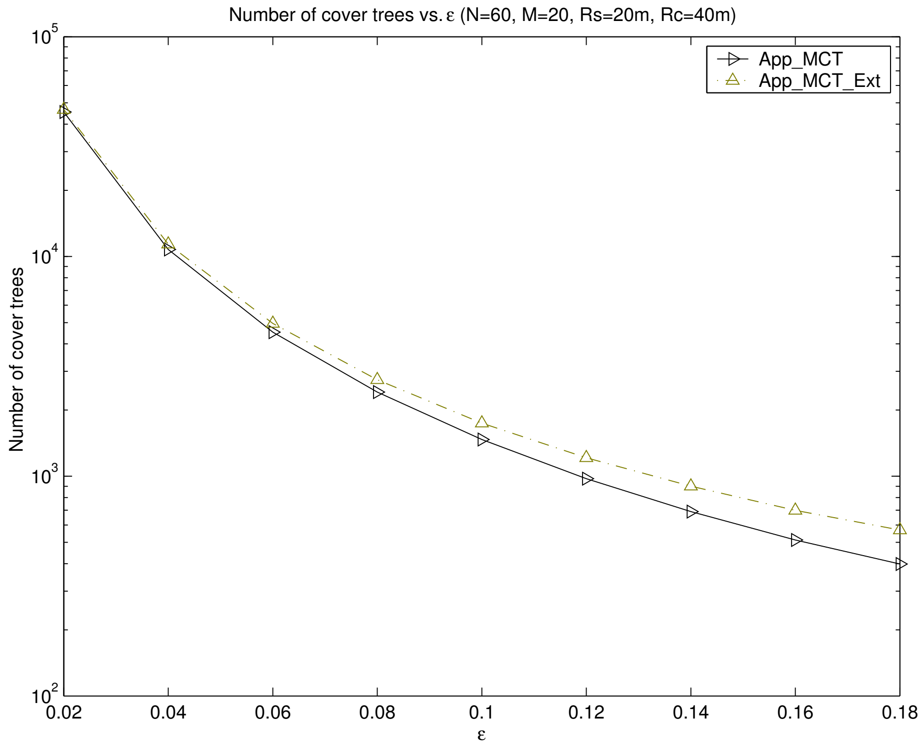

Figure 4.3: Number of cover trees vs. ϵ ($N = 60$, $M = 20$)

the 95% confidence intervals are shown in the figure. As expected, both the average and worst case normalized lifetime achieved by both algorithms decreases when ϵ increases. When ϵ is small, both algorithms can achieve near-optimal result. However, the performance of App_MCT declines much faster than that of App_MCT_Ext when ϵ increases. Figure 4.3 shows the number of cover trees generated by the two algorithms with different values of ϵ. It can be seen that the number of cover trees is very large when ϵ is small (above 40000 when $\epsilon = 0.02$).

Next we study the impact of operation duration τ on the performance of the CWGC algorithm. The same scenario as chosen for the approximation algorithms is used in the simulation. The lifetime achieved by the CWGC algorithm is normalized by the lifetime upper bound. The value of τ is taken as $\tau = T_{LP}/kM$,

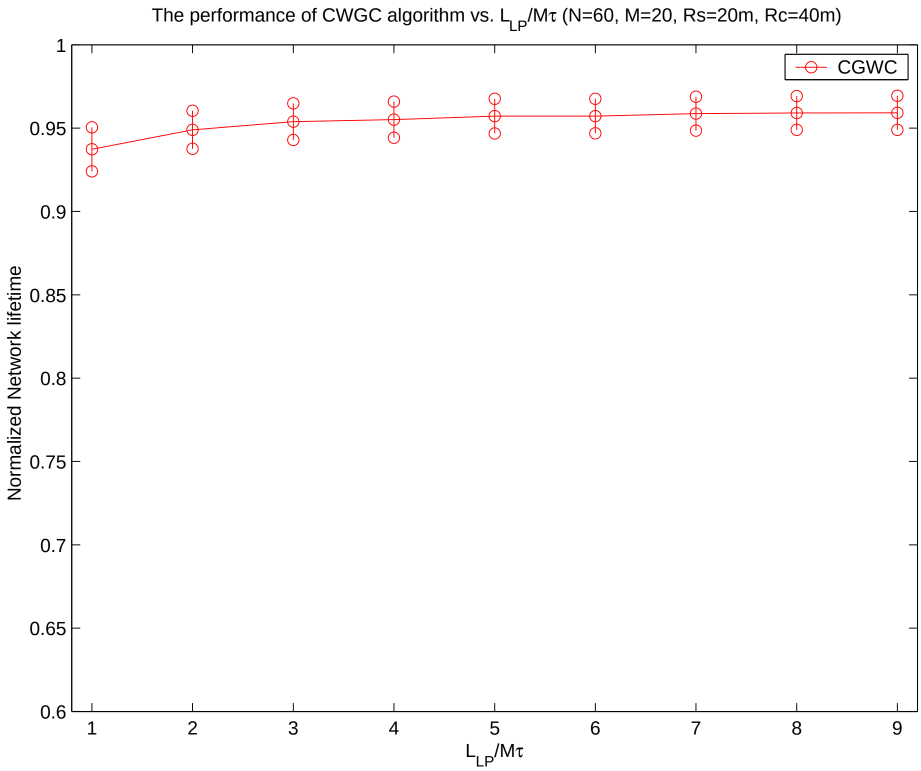

Figure 4.4: Normalized lifetime vs. $k = T_{LP}/M\tau$ ($N = 60$, $M = 20$)

where k is the parameter that we will vary. Figure 4.4 shows the average normalized lifetime achieved by the CWGC algorithm with different values of $T_{LP}/M\tau$ (From corollary 2, T_{LP}/M is the lower bound of the optimal solution). The 95% confidence intervals are shown in the figure. It can be seen that the CWGC algorithm can achieve near-optimal lifetime similar to the App_MCT_Ext algorithm. When τ is large, the performance of the CWGC algorithm declines very slowly. Fig. 4.5 shows the number of cover trees generated by the CWGC algorithm with different values of $T_{LP}/M\tau$. It increases linearly with the increase of $T_{LP}/M\tau$.

Comparing Fig. 4.2 and 4.3 with Fig. 4.4 and 4.5, we can observe that the CWGC algorithm can achieve the network lifetime close to that achieved by App_MCT and

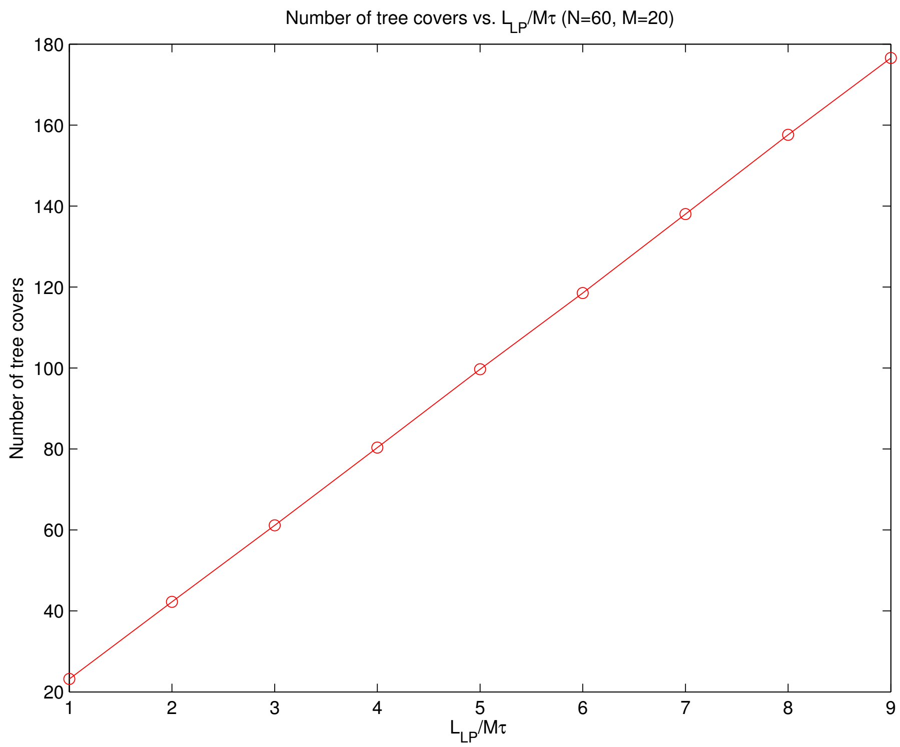

Figure 4.5: Number of cover trees vs. $k = T_{LP}/M\tau$ ($N = 60$, $M = 20$)

APP_MCT_Ext, while generating significantly small number of cover trees. When $\tau = T_{LP}/2M$, the CWGC algorithm can achieve about 95% of the network lifetime upper bound with only about 40 cover trees. The App_MCT and App_MCT_Ext can achieve about 95% of the network lifetime upper bound for $\epsilon = 0.02$ and $\epsilon = 0.08$, but with about 45000 and 2800 cover trees, respectively.

In the following simulations, we fix the value of $\epsilon = 0.1$ and $\tau = T_{Lp}/2M$ which are a reasonable balance between algorithm performance and computation complexity. As App_MCT_Ext always performs better than App_MCT algorithm, we will only show the performance of App_MCT_Ext.

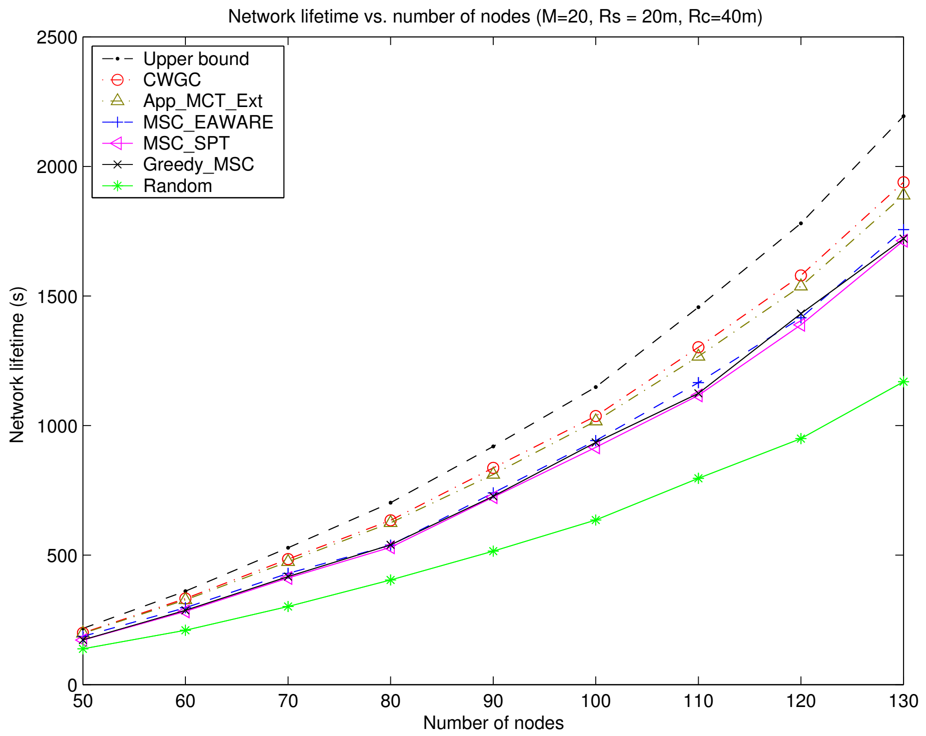

Figure 4.6: Network lifetime vs. number of nodes ($M = 20$)

4.4.2 Impact of network parameters

In Fig. 4.6 and 4.7, we study the impact of network density on the network lifetime and performance of the CWGC and App_MCT_Ext algorithms. We compare the performance of the CWGC and App_MCT_Ext algorithms with other four algorithms: MSC_EAWARE, MSC_SPT, Greedy_MSC_SPT and Random algorithm. The number of targets is fixed at 20. The sensing range and the communication range is set as $R_s = 20m$ and $R_c = 40m$. Figure 4.6 plots the network lifetime achieved by CWGC and App_MCT_Ext algorithms in comparison with other algorithms and the lifetime upper bound when the number of nodes increases from 50 to 140. As the number of nodes increases, more nodes can be scheduled to sense the targets and relay the messages, leading to the increased network lifetime.

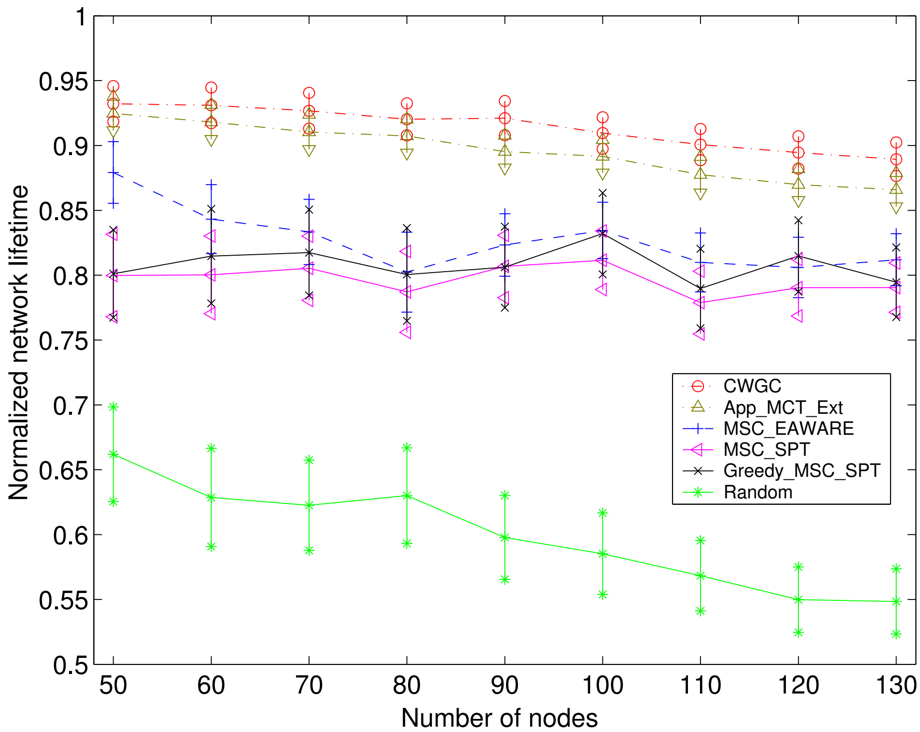

Figure 4.7: Normalized network lifetime vs. number of nodes ($M = 20$)

To demonstrate the superiosity of our algroithm more clearly, we normalize the network lifetime achieved by CWGC, App_MCT_Ext and other algorithms by the lifetime upper bound achieved by solving LP problem. Figure 4.7 plots the normalized lifetime achieved by the CWGC and App_MCT_Ext algorithms in comparison with other algorithms when the number of nodes increases from 50 to 130. It can be seen that the lifetime achieved by the CWGC algorithm is very close to the upper bound. When the number of nodes is 50, the CWGC algorithm can achieve about 95% of the lifetime upper bound. As the number of nodes increases, the performance slowly decreases – when the number of nodes is 130, about 90% of the lifetime upper bound is achieved. It can also be observed that the CWGC algorithm performs considerably better than App_MCT_Ext and significantly considerably better than the other three

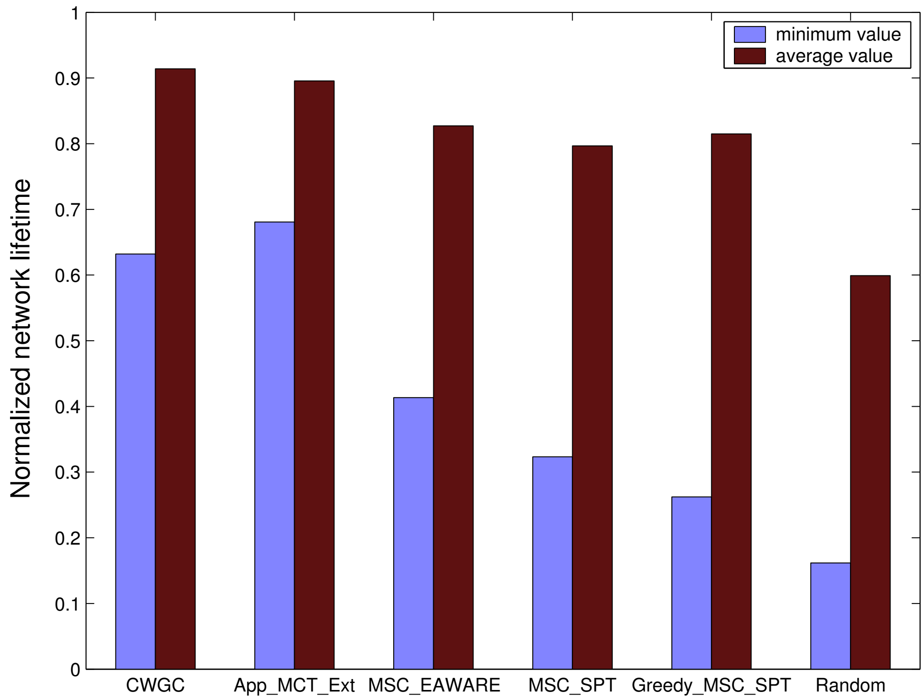

Figure 4.8: Minimum and average normalized network lifetime ($M = 20$)

algorithms. The decreasing performance trend (as the number of nodes increases) is observed for all the algorithms, the reason can be attributed to the looser upper bound when the number of nodes increases. Finally, we observe that the length of the confidence interval of the normalized lifetime achieved by the CWGC algorithm is similar to App_MCT_Ext (0.99) and is much shorter than the other three algorithms (about 0.4 of that of Random algorithm and about 0.5 of that of MSC_EAWARE, MSC_SPT and Greedy_MSC_SPT), which implies that the CWGC algorithm can achieve a stable performance.

The worst-case and average-case normalized lifetimes of the six algorithms are compared in Fig. 4.8. We consider the same random networks which are used to generate the results plotted in Fig. 4.7. A total of 900 topologies, with the number

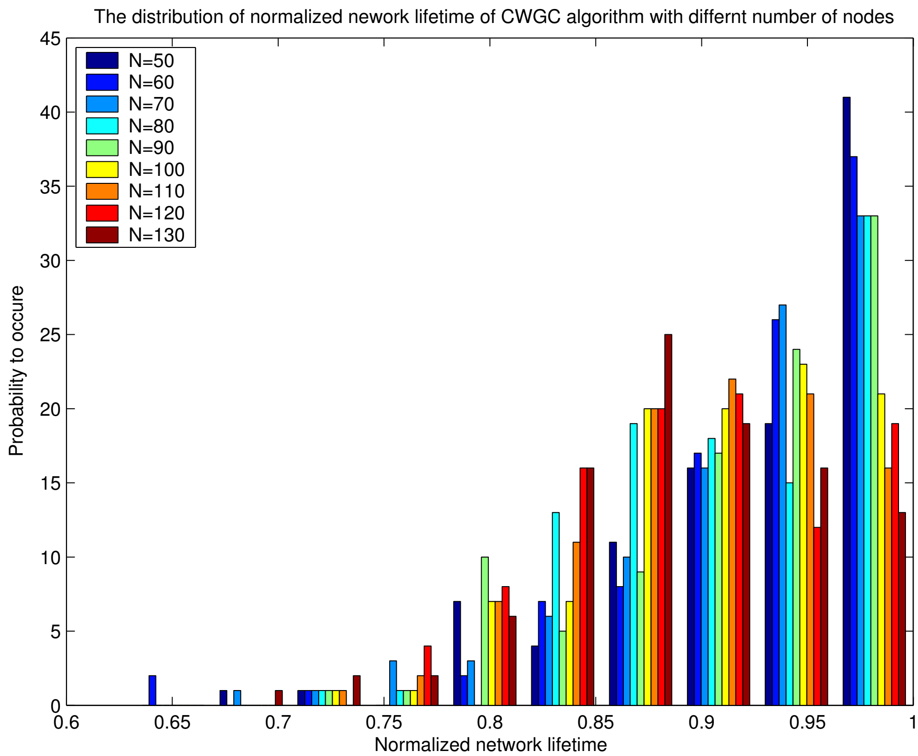

Figure 4.9: Distribution of normalized network lifetime of CWGC algorithm ($M = 20$)

of nodes $N = \{50, 60, \cdots, 130\}$ with one hundred randomly generated graphs for each value of N are used. The average performance of the CWGC algorithm is the best (0.91), but the worst case performance of it (0.63) is a little less than that of App_MCT_Ext (0.68). The CWGC algorithm performs 53% and 290% better than the RANDOM algorithm in the average and worst case, respectively. Although MSC_EAWARE, MSC_SPT and Greedy_MSC_SPT algorithms perform better than the Random algorithm, they are not as good as the CWGC algorithm, which performs about 10% better than them in the average case, and about 53%, 95% and 140% better than them in the worst case, respectively. Figure 4.9 gives the distribution of the normalized lifetime achieved by CWGC algorithm, with N increases from 50 to 130. It can be seen that In most cases (97% when $N = 50$ and 92% when $N = 130$)

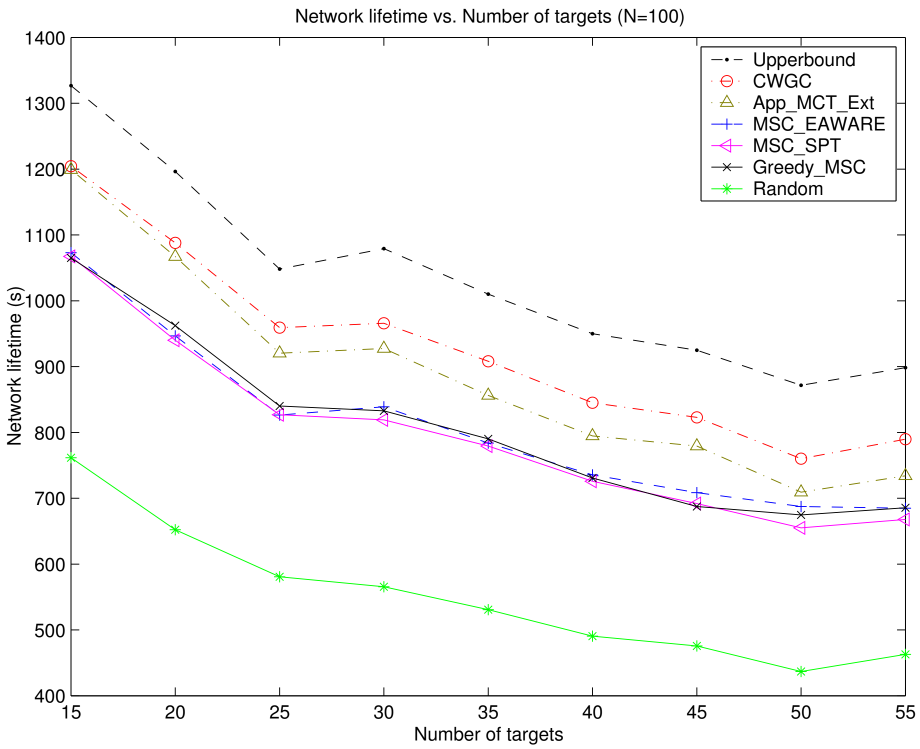

Figure 4.10: Network lifetime vs. number of targets ($N = 100$)

CWGC algorithm can achieve more than 80% of the lifetime upper bound (which is the average performance of MSC_EWARE and MSC_SPT algorithm). As we observed earlier in Fig. 4.6, when the number of nodes increases, the probability of achieving lower normalized lifetime also increases.

In Fig. 4.10 and 4.11, we study the impact of varying the number of targets on networkl lifetime and the performance of the algorithms. The number of targets is increased from 15 to 55 and the number of sensors is fixed at 100. The sensing range and the communication range are set as $R_s = 20m$ and $R_c = 40m$, respectively. It can be observed that the network lifetime decreases as the number of targets increases. This is because more nodes need to be activated to maintain the coverage of all the targets. When the number of targets increases from 15 to 55 (N=100), the absolute

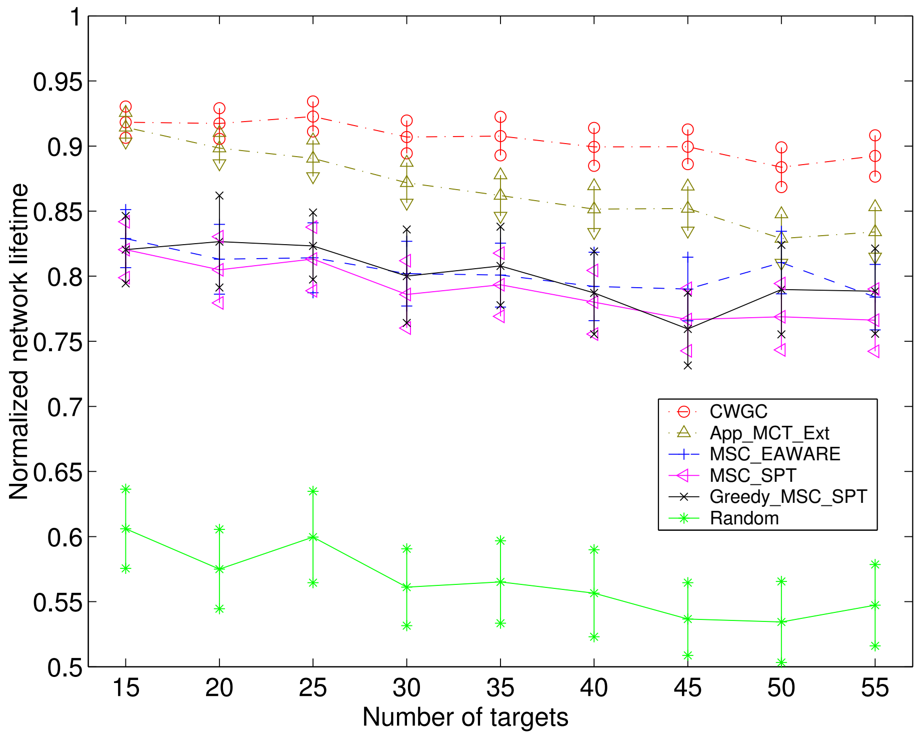

Figure 4.11: Normalized network lifetime vs. number of targets ($N = 100$)

value of average network lifetime upper bound decreases from 1326 to 898. The algorithms also show a similar declining trend for the lifetime. The CWGC algorithm performs better than all the other algorithms in all cases. It is also observed in Fig. 4.11 that the performance of the algorithms decreases as the number of targets increases, which is due to the reason that the upper bound becomes looser when the number of targets increases. For App_MCT_Ext an increase of the number of targets will in turn increase $\hat{M}$ – the maximum number of targets in the sensing range of any sensor, leading to smaller approximation ratio of App_MCT_Ext (see Theorem 3). Therefore, the performance of App_MCT_Ext declines faster than other algorithms.

4.4.3 Potential protocol cost

The construction of new cover trees incurs protocol cost. To construct a cover tree, any algorithm (MWCT or SPT based) needs to construct a communication tree and select the source nodes to cover all the targets. Therefore, the number of cover trees generated by the algorithms could be a good indicator for the potential protocol cost. We note that the construction of new cover trees in MSC_SPT algorithm is triggered by topology changes, i.e. when a node dies due to energy depletion, while the construction of new cover trees in CWGC and MSC_EAWARE algorithms is time-based and related to the chosen value of τ. We set $N = 100$ and $M = 20$. For $\tau = T_{LP}/2M$, the mean number of cover trees generated by CWGC, MSC_EAWARE and MSC_SPT algorithms is 57.2, 51.4 and 26.3, respectively. We observe that MSC_SPT algorithm generates considerably smaller number of cover trees than CWGC and MSC_EAWARE algorithms. When τ is increased to T_{LP}/M, the normalized network lifetime of CWGC algorithm slightly decreases from 0.92 to 0.91, while the number of cover trees drastically decreases from 57.2 to 39.2. If we set τ to be infinity, the construction of a new cover tree is also triggered by topology changes, and the normalized network lifetime of CWGC algorithm decreases to 0.90 with 27.8 cover trees constructed. So, by suitably selecting the value of τ, CWGC algorithm can achieve a protocol cost close to that of MSC_SPT while achieving significantly better performance in terms of lifetime.

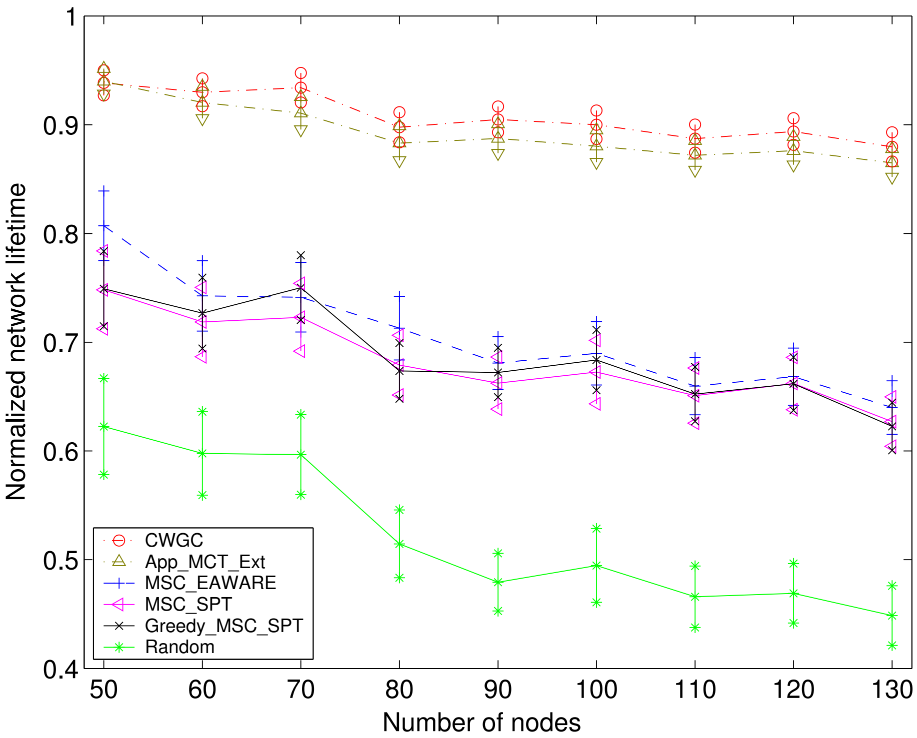

Figure 4.12: Normalized network lifetime vs. number of nodes for non-identical data generation rates ($M = 20$)

4.4.4 Impact of non-identical data generation rates

We study the performance of our App_MCT_Ext and CWGC algorithms for non-identical data generation rates and compare it with that of other algorithms. We suitably extend our formulation and analysis presented earlier, to account for non-identical data generation rates. For example, in section 4.1.3 on algorithm description, in step 2(b)i, the expression $\frac{w^i_{SPT}(\vec{C})}{|\mathcal{P}^{tk}_i|}$ can be modified as $\frac{w^i_{SPT}(\vec{C})f_s}{|\mathcal{P}^{tk}_i|}$, where f_s is the data generation rate of source s. Similarly, the right side of Eq. 4.50 can be modified as $\frac{|P_s - P_s \bigcap P'|}{w_s f_s}$. We note that the claims on bounds and analysis are not affected. In our simulation, the data generation rate of the source nodes is uniformly distributed in the range from 6 to 14Kbps (with the mean of 10Kbps). From Fig. 4.12 we observe that our algorithms perform significantly better than other algorithms and the performance improvement can be up to 45% than Greedy_MSC_SPT algorithm when $N = 130$. Further, the performance gain achieved by our algorithms is higher when compared to the case when the data generation rate of the sources is the same with 10Kbps (shown in Fig. 4.7). The reason for the increased performance gain is that, when the data generation rates are different, our algorithms tend to prefer source nodes with lower rates.

In summary, the approximation algorithm and CWGC algorithm can achieve near-optimal performance and they perform much better than MSC_EWARE, MSC_SPT, Greedy_MSC_SPT and Random algorithms. Compared with the approximation algorithm App_MCT_Ext, the CWGC algorithm can achieve better performance with much lower complexity and protocol cost.

4.5 Summary

In this chapter we developed an efficient approximation algorithm and a faster greedy heuristic algorithm for the MCT problem. We also presented a distributed implementation of the heuristic algorithm. Simulation results show that the lifetime achieved by our approximation algorithm and heuristic algorithm is very close to the upper bound and their performance is much better than that of other possible heuristics.

Chapter 5

Lifetime Maximization observation Schedule (LMOS) problem

In this chapter, we consider the CTC problem when the data generation rate of a sensor is proportional to the number of targets it observes, and with K coverage requirement wherein each target is observed by at least K sensors. Such K-coverage requirement improves the accuracy and reliability of the observations. In the CTC problem we discussed in chapter 3 and chapter 4, we assume that all the targets located in the sensing area of a sensor node would be observed by the sensor, and the amount of observation data generated by the sensor is independent of the targets observed by the sensor. However, in some applications the data generated by the source sensor is related to the targets observed by it, i.e. more the targets observed, more the observation data generated. Further, some kinds of sensor nodes may have the ability and freedom to select only a subset of targets in its sensing area to observe. For example, a fixed camera or video sensor observes all the targets in its sensing area simultaneously, while a camera sensor with adjustable observation angle can focus on only a subset of targets in its sensing area. The observation redundancy would be further reduced by carefully selecting the set of targets to be observed by each source sensor. In this chapter, we formulate the connected target coverage problem with observation-related data generation rate and K coverage requirement as a Lifetime Maximization Observation Schedule (LMOS) problem. We discuss the problem with two different observation scenarios depending on whether a sensor can select a subset of targets in its sensing area to observe or not. We prove that the LMOS problem for the first scenario (LMOS-1) is a P problem and develop a polynomial-time algorithm which can achieve the optimal solution based on Linear Programming and Integer Theorem. We show that the LMOS problem for the second scenario (LMOS-2) is NP complete. We derive an upper bound and a lower bound of the LMOS-2 problem based on the optimal solution of LMOS-1 problem.

5.1 System Model and Problem Description

We consider the CTC problem with the similar application scenario as in the Chapter 3. However, we assume that the rate at which data messages are generated by a source is related to the set of targets observed by the source. Further, we extend the coverage requirement to be K coverage wherein each target is required to be simultaneously observed by at least K sensor nodes at any time. The definition of the source nodes, relay nodes, active nodes and sleep nodes are the same as in Chapter 3. As defined earlier, the *network lifetime* is the time duration starting from when the network was set up until the sink can no longer receive the required observation reports of all the targets.

A *target rate* r_m is defined for each target p_m as the observation data reporting rate. The data rate outgoing from a source is the sum of the data rates associated with the targets (called target rate) it is observing, which is called as *source rate*. Depending on the sensor devices and applications, we consider the following two different observation scenarios (OS) regarding the observation of targets by a source sensor:

OS-1 An observing sensor is able to control the observation of the targets in its sensing area and select a subset of targets to observe. We set an integer L as the maximum number of targets a sensor can simultaneously observe and call this constraint as the *observation constraint*. When $L = 1$, each sensor can only observe one target at a time. When L is larger than the number of targets in the sensing area of a sensor node, the sensor can select any subset of targets in its sensing area to observe.

OS-2 An observing sensor should simultaneously observe all the targets in its sensing area.

We assume without loss of generosity that the whole network lifetime is slotted into a series of time slots. Within each time slot the state (observation, relay or sleeping) of each sensor node and the set of targets that each source sensor observes do not change. An *observation assignment* determines the state of each sensor and the set of targets each source observes in a time slot. The overall duration of the time slots within which the same observation assignment exists is called the *operation duration* of the observation assignment. For both observation scenarios, an observation assignment should decide the set of source nodes and relay nodes together with the path from each source to the sink. For OS-2, as the sensors and targets are all static, the set of targets observed by each source as well as the source rate is fixed. However, for OS-1, an observation assignment should additionally decide the subset of targets that each selected source sensor observes and in turn decide the source rate.

Given graph $G = \{\mathcal{S} \cup \mathcal{P} \cup \mathcal{R}, \mathbb{E}\}$ building the network topology, we define an observation assignment ϕ in OS-1 as a set of paths starting from the target set to the sink node. All the sensors on the paths are "active" nodes. For each path, the starting observation link (p_m, s_i) represents that sensor s_i is assigned as a source to observe target p_m, and the observation data of p_m will be transmitted by s_i to the sink through the path. On the other hand we define an observation assignment ϕ in OS-2 as a set of paths from the source set to the sink node. All the sensors on the paths are "active" nodes. For each path, the starting node is the source node which transmits the observation data of all the targets in its sensing area to the sink through the path.

For OS-1, an observation assignment ϕ is called feasible if and only if

1. the K coverage requirement is satisfied,
2. no observation link is selected more than once,
3. no sensor is selected as the source sensor on more than L paths.

For OS-2 an observation assignment is called feasible if and only if the K coverage requirement is satisfied.

We define an *observation schedule* as a sequence of observation assignments with their operation durations. An observation schedule is called feasible if and only if all the observation assignments in the schedule are feasible and each node consumes energy less than its initial energy after the execution of the schedule.

We consider the same energy consumption model as in Chapter 3 which takes into account the energy consumption for sensing and relaying data. Let e_s and e_r denote the energy consumed for sensing and receiving a bit, respectively. Let e^t_{ij} denote the energy consumed by sender s_i for transmitting a bit to receiver s_j: $e^t_{ij} = e_t + b \cdot d^{\alpha}_{ij}$, where e_t and b are constants, d_{ij} is the Euclidean distance between node s_i and s_j and α is the path loss factor.

The Lifetime Maximization Observation Schedule (LMOS) problem is defined below:

Definition 1 (LMOS problem) *Given a graph $G = \{\mathcal{S} \cup \mathcal{P} \cup \mathcal{R}, \mathbb{E}\}$, target rate set $\{r_m\}$, coverage requirement K, the initial energy $E_0(s_i)$ for each sensor s_i, find a feasible sensor observation schedule, which has the maximum total execution time T (i.e. maximum lifetime).*

We use LMOS-1 to refer to the LMOS problem for *OS-1* and LMOS-2 to refer to the LMOS problem for *OS-2*.

5.2 The solution for LMOS-1 problem

In this section we develop a polynomial-time algorithm to find the optimal solution for LMOS-1 problem, and thus prove that LMOS-1 problem is a P problem. The developed algorithm first solves a LP problem and then decomposes the solution of the LP problem to find the optimal observation schedule.

5.2.1 Derivation of upper bound of LMOS-1 problem – LP formulation

Let τ_{im} denote the overall time duration that sensor s_i is assigned to observe target p_m, i.e. the period during which observation link (p_m, s_i) is selected in all the observation assignments. Let F_{ij} denote the total amount of data traversing through link (s_i, s_j). Let T denote the network lifetime. The following LP formulation gives an upper bound on the maximum network lifetime of LMOS-1 problem:

Objective:

$$\text{Maximize: } T \tag{5.1}$$

Subject to:

$$\tau_{im} \leq T; \qquad \forall (s_i, p_m); \tag{5.2}$$

$$\sum_m \tau_{im} \leq LT; \qquad \forall s_i; \tag{5.3}$$

$$\sum_i \tau_{im} = KT; \qquad \forall p_m; \tag{5.4}$$

$$\sum_m \tau_{im} r_m + \sum_{j \neq i} F_{ji} = \sum_{j \neq i} F_{ij}; \quad \forall s_i; \tag{5.5}$$

$$\sum_m \tau_{im} r_m e_s + \sum_{j \neq i} (F_{ij} e^t_{ij} + F_{ji} e_r) \leq E_0(s_i); \forall s_i. \tag{5.6}$$

In the above formulation, Eq. 5.2 specifies that no observation link can be selected more than once in an observation assignment, Eq. 5.3 indicates that each sensor can simultaneously observe up to L targets, Eq. 5.4 specifies the K coverage requirement, Eq. 5.5 is the flow conservation constraint and Eq. 5.6 is the energy consumption constraint.

Since the constraints in the LP problem are only necessary conditions (i.e. an observation link can still be selected more than once in an assignment even when

Eq. 5.2 satisfies), the solution of the above LP problem provides a lifetime upper bound for the LMOS-1 problem. Further, the optimal observation schedule is not given in the LP solution. To completely solve the LMOS-1 problem, in the next sections, we develop a polynomial-time algorithm which finds a feasible observation schedule achieving the same network lifetime as given by the LP solution, and thus prove that the solution of LP problem gives exactly the optimal lifetime of LMOS-1 problem.

5.2.2 Algorithm Description

In this section we describe our algorithm for building the optimal observation schedule. The input of the algorithm is the optimal solution obtained by solving the LP problem $\{T, \{\tau_{im} : \forall(s_i, p_m)\}, \{F_{ij} : \forall(s_i, s_j)\}\}$. The output is the desired optimal observation schedule $\{\phi(x), \tau(x) : 1 \leq x \leq X\}$, where each $\phi(x)$ is a feasible observation assignment in the schedule and $\tau(x)$ is the corresponding operation duration of the assignment. The optimal observation schedule is built by iteratively decomposing the LP solution, and thus the algorithm is called as the *decomposition algorithm*.

For each feasible observation assignment $\phi(x)$, given its operation duration $\tau(x)$, we can calculate the amount of time that each observation link (p_m, s_i) is occupied in the duration – $\tau_{im}(x)$, as well as the amount of flow traversing through each communication link (s_i, s_j)– $F_{ij}(x)$ as follows:

$$\tau_{im}(x) = \begin{cases} \tau(x); & (p_m, s_i) \in \phi(x); \\ 0; & (p_m, s_i) \notin \phi(x); \end{cases} \tag{5.7}$$

Table 5.1: Pseudo-code for the decomposition algorithm

Input:	T,$\{\tau_{im}\}$, $\{F_{ij}\}$.
Output:	$\{\phi(1), \phi(2), \cdots, \phi(X)\}$, $\{\tau(1), \tau(2), \cdots, \tau(X)\}$.
(01)	$x = 1$;
(02)	**while** $(T > 0)$
(03)	Build flow network G^*;
(04)	Find $\mathbb{E}_o^{'}$, Build flow network $G^{'}$;
(05)	Find integer valued maximum flow $\bar{f}$ on network $G^{'}$;
(06)	Let $\mathbb{E}_o^f$ denote the observation links on which $\bar{f}$ is 1; $\mathbb{E}_o^\phi = \mathbb{E}_o^{'} \cup \mathbb{E}_o^f$;
(07)	For $\forall s_i$ connected with $\mathbb{E}_o^\phi$, find the path from sensor s_i to the sink via only communication links in $\mathbb{E}_c^*$;
(08)	Build observation assignment $\phi(x)$ by appending $\forall(p_m, s_i) \in \mathbb{E}_o^\phi$ with the path from s_i to the sink;
(09)	Calculate τ_{min}^o, τ_{min}^c and $\tau_{max}^{'}$; Set $\tau(x) = \min(\tau_{min}^o, \tau_{min}^p, \tau_{min}^c, T - \tau_{max}^{'})$
(10)	$\forall(p_m, s_i) \in \mathbb{E}_o^\phi$, $\tau_{im} = \tau_{im} - \tau(x)$;
(11)	$\forall(s_i, s_j) \in \mathbb{E}_c^\phi$, $F_{ij} = F_{ij} - f_{ij}^\phi \tau(x)$;
(12)	$T = T - \tau(x)$; $x = x + 1$;
(13)	**Endwhile**

$$F_{ij}(x) = \sum_{m:(s_i,s_j)\in R_m(x)} r_m \tau(x); \tag{5.8}$$

where $R_m(x)$ is the route in $\phi(x)$ from target p_m to the sink.

If all the assignments in a schedule are feasible observation assignments, together with $\sum_x \tau(x) = T$, $\sum_x \tau_{im}(x) = \tau_{im}$ and $\sum_x F_{ij}(x) = F_{ij}$ (where T, τ_{im} and F_{ij} are elements in the solution set), we can conclude that the schedule is the desired optimal schedule. Let us call $\{\tau(x), \{\tau_{im}(x) : \forall(s_i, p_m)\}, \{F_{ij}(x) : \forall(s_i, s_j)\}\}$ as the x^{th} assignment solution.

The pseudo code of the algorithm is described in Table 5.1. The algorithm runs in iterations. In each iteration x, a feasible assignment $\phi(x)$ with proper operation

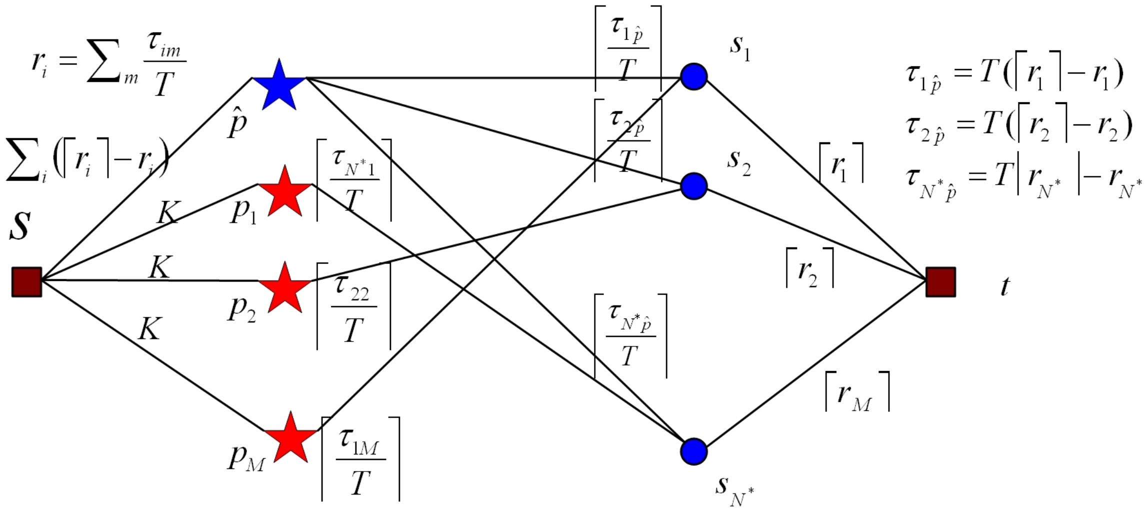

Figure 5.1: Flow network $G^* = \{V^*, \mathbb{E}^*\}$

duration $\tau(x)$ is found based on the solution set, and then the solution set $\{T, \tau_{im}, F_{ij}\}$ is updated by decreasing the assignment solution $\{\tau(x), \tau_{im}(x), F_{ij}(x)\}$. Next we introduce how to find the assignment and its operation duration in each iteration, and prove its correctness.

In the x^{th} iteration, given the current solution set $\{T, \tau_{im}, F_{ij}\}$, we first build a flow network $G^* = \{V^*, \mathbb{E}^*\}$ as shown in Fig. 5.1 (line 3). The graph consists of all the observation links (p_m, s_i) with non zero τ_{im} together with the targets and sensors connected by those links. Let $\mathbb{E}_o^*$ denote these observation links, $\mathcal{S}^*$ denotes these sensors and $\mathcal{P}^*$ denote these targets. Further, a *pseudo* target $\hat{p}$ is added in G^* with *pseudo* observation links connecting to all the sensors in $\mathcal{S}^*$. We define $\tau_{i\hat{p}} = T(\lceil r_i \rceil - r_i)$ as the *pseudo* observation duration that each sensor s_i observes $\hat{p}$, where $r_i = \sum_m \tau_{im}/T$. A source node s is added with links connecting all the targets in $\mathcal{P} \cup \hat{p}$; a destination node t is added with links connecting all the sensors in $\mathcal{S}^*$.

Unit capacity is assigned to each link in $\mathbb{E}_o^*$. The capacity of the link from s to each target p_m is set to be K. For each link from sensor s_i to the destination t, the capacity is set to be $\lceil r_i \rceil$. For each link from the pseudo target $\hat{p}$ to the sensor s_i, the capacity is set to be 0 if $r_i = \lceil r_i \rceil$, otherwise 1. The capacity of link $(s, \hat{p})$ is set to be $\sum_i (\lceil r_i \rceil - r_i)$. Now we complete the building of flow network G^*.

Let $\mathbb{E}_o^{'}$ denote the set of the observation links (p_m, s_i) on which $\tau_{im} = T$. Clearly $\mathbb{E}_o^{'} \subseteq \mathbb{E}_o^*$. For each observation link $(p_m, s_i) \in \mathbb{E}_o^{'}$, we delete the link (p_m, s_i) from the G^*, and in turn decrease the capacity of links (s, p_m) and (s_i, t) by 1, respectively. After deleting all the links with zero capacity we construct a new flow network $G^{'}$ (line 4). Using Ford-Fulkerson method [63], we find a maximum flow $\bar{f}$ on $G^{'}$ (line 5). Let $\mathbb{E}_o^f$ and $\mathbb{E}_p^f$ denote the set of observation links and pseudo observation links on which flow $\bar{f}$ is positive, respectively. We construct the set of observation links in the observation assignment as $\mathbb{E}_o^\phi = \mathbb{E}_o^{'} \cup \mathbb{E}_o^f$ (line 6).

For each observation link $(p_m, s_i) \in \mathbb{E}_o^\phi$, we find a path from s_i to the sink via only communication links with $F_{ij} > 0$ (e.g. using breadth first search) (line 7). Appending each observation link $(p_m, s_i) \in \mathbb{E}_o^\phi$ with the path, we build an observation assignment $\phi(x)$ (line 8).

After building the observation assignment $\phi(x)$, we can determine the operation duration $\tau(x)$. The value of $\tau(x)$ is chosen as the maximum value while satisfying the following four conditions: (a) $F_{ij}(x) \leq F_{ij}$ for any communication link $(s_i, s_j) \in \phi$; (b) $\tau_{im}(x) \leq \tau_{im}$ for any observation link $(p_m, s_i) \in \mathbb{E}_o^\phi$; (c) $T - \tau(x) \geq \max(\tau_{im} : (p_m, s_i) \notin \mathbb{E}_o^\phi)$; (d) $\tau(x) \leq \tau_{i\hat{p}}$ for any pseudo observation link $(\hat{p}, s_i) \in \mathbb{E}_p^f$. (Recall that $\mathbb{E}_p^f$ denote the set of pseudo observation links on which the maximum flow $\bar{f}$ is positive). Let τ^c_{min} denote the minimum value of $F_{ij}/\sum_{m:(s_i,s_j)\in R_m(x)} r_m$ on each selected communication links (s_i, s_j); τ^o_{min} denote the minimum value of τ_{im} for observation links in $\mathbb{E}^\phi_o$; τ^p_{min} denote the minimum value of $\tau_{i\hat{p}}$ for pseudo observation links in $\mathbb{E}^f_p$; and $\tau^{'}_{max}$ denote the maximum value of τ_{im} for observation links in $\mathbb{E}^*_o - \mathbb{E}^\phi_o$. The operation duration of assignment ϕ is set (line 9) as

$$\tau(x) = \min(\tau^o_{min}, \tau^p_{min}, \tau^c_{min}, T - \tau^{'}_{max}) \tag{5.9}$$

Then we calculate the x^{th} assignment solution $\{\tau(x), \tau_{im}(x), F_{ij}(x)\}$ and update the solution set $\{T, \tau_{im}, F_{ij}\}$ by decreasing it by the assignment solution. If the value of T becomes zero in the updated solution set, the algorithm stops.

5.2.3 Correctness of the algorithm

Theorem 5 *The solution set* $\{T, \tau_{im}, F_{ij}\}$ *in each iteration satisfies the following equations:*

$$\tau_{im} \le T; \quad \forall(s_i, p_m); \tag{5.10}$$

$$\sum_m \tau_{im} \le LT; \qquad \forall s_i; \tag{5.11}$$

$$\sum_i \tau_{im} = KT; \qquad \forall p_m; \tag{5.12}$$

$$\sum_m \tau_{im} r_m + \sum_{j\neq i} F_{ji} = \sum_{j \neq i} F_{ij}; \qquad \forall s_i; \tag{5.13}$$

$$\tau_{im} \ge 0; \quad T \ge 0; \quad F_{ij} \ge 0; \quad \forall s_i, s_j, p_m \tag{5.14}$$

Proof: We prove the theorem by induction. In the first iteration, as the solution set is the solution of the LP problem, all the equations are satisfied.

Suppose that the equations are satisfied in the x^{th} iteration. Let $\{T^x, \tau^x_{im}, F^x_{ij}\}$ denote the solution set in the iteration and $\{T^{x+1}, \tau^{x+1}_{im}, F^{x+1}_{ij}\}$ denote the solution set in the next iteration. From the description of the algorithm, we have

$$T^{x+1} = T^x - \tau(x); \tag{5.15}$$

$$\tau^{x+1}_{im} = \begin{cases} \tau^x_{im} - \tau(x); & (p_m, s_i) \in \mathbb{E}^{\phi}_o \\ \tau^x_{im}; & (p_m, s_i) \notin \mathbb{E}^{\phi}_o \end{cases} \tag{5.16}$$

$$F^{x+1}_{ij} = F^x_{ij} - F_{ij}(x); \tag{5.17}$$

From Eq. 5.15 and Eq. 5.16, for observation links in $\mathbb{E}^{\phi}_o$, we have $\tau^{x+1}_{im} \leq T^{x+1}$. As $\tau(x) \leq T^x - \tau^{'}_{max}$, for observation links not in $\mathbb{E}^{\phi}_o$, we also have $\tau^{x+1}_{im} \leq T^{x+1}$. Thus Eq. 5.10 is satisfied in the $x + 1^{th}$ iteration.

As $\tau(x) \leq \tau^o_{min}$ and $\tau(x) \leq \tau^c_{min}$, we have $\tau^{x+1}_{im} \geq 0$ and $F^{x+1}_{ij} \geq 0$. As Eq. 5.10 is satisfied in the $x + 1^{th}$ iteration, $T^{x+1} \geq \tau^{x+1}_{im} \geq 0$. Eq. 5.14 is satisfied in the $x + 1^{th}$ iteration.

Consider the flow network G^* as constructed in section 5.2.2. The capacity on all the links except link $(s, \hat{p})$ are integer values (K, 1 or $\lceil r_i \rceil$). As Eq. 5.12 is satisfied in the x^{th} iteration, the capacity of link $(s, \hat{p})$ is $\sum_i (\lceil r_i \rceil - r_i) = \sum_i \lceil r_i \rceil - \sum_i \sum_m \tau^x_{im}/T^x = \sum_i \lceil r_i \rceil - KM$, which is also an integer value. Thus the capacity of all the links in G^* are integer values. In turn, the capacity of all the links in $G^{'}$ are also integer values as we build $G^{'}$ from G^*. Let n denote the number of links in $\mathbb{E}^{'}_o$ (links with $\tau^x_{im} = T^x$ which are deleted from G^*); n_i denote the number of links in $\mathbb{E}^{'}_o$ connecting with s_i; n_m denote the number of links in $\mathbb{E}^{'}_o$ connecting with p_m. For each sensor s_i, as $r_i = \sum_m \tau^x_{im}/T^x$, $n_i \leq \lceil r_i \rceil$, the capacity of link (s_i, t) on $G^{'}$ is given by $\lceil r_i \rceil - n_i \geq 0$. For each target p_m, from Eq. 5.12, we infer $n_m \leq K$, and the capacity of link (s, p_m) is $K - n_m \geq 0$. Therefore, the capacity of each link in $G^{'}$ is a non-negative integer.

It is easy to prove that the capacity of $G^{'}$ is $\sum_i \lceil r_i \rceil - n$, the set of links $\{(s, p_m) : \forall p_m\} \cup (s, \hat{p})$ and $\{(s_i, t) : \forall s_i\}$ are both minimum cuts of $G^{'}$. Therefore, the value of the maximum flow $\bar{f}$ is also $\sum_i \lceil r_i \rceil - n$. In addition, as all the links in $G^{'}$ have non-negative integer capacity, from Integer Theorem (pp. 666 in [63]), the value of $\bar{f}$ on all the link are also integer values. Since unit capacity is assigned to all the observation links and pseudo observation links, $\bar{f}$ can only take a value of 0 or 1 on these links. As the set of links $\{(s, p_m) : \forall p_m\} \cup (s, \hat{p})$ is the minimum cuts of $G^{'}$, the value of flow $\bar{f}$ on each link (s, p_m) is the capacity on the link. From flow conservation, we have $\sum_i \bar{f}_{mi} = K - n_m$. Since $\bar{f}$ only take 0 or 1 values on observation links, $\sum_i \bar{f}_{mi}$ is the number of links in $\mathbb{E}_o^f$ connecting with p_m. As $\mathbb{E}_o^\phi = \mathbb{E}_o^{'} \cup \mathbb{E}_o^f$, from Eq. 5.16, we have

$$\begin{aligned} \textstyle\sum_i \tau_{im}^{x+1} &= \textstyle\sum_i \tau_{im}^x - (\sum_i \bar{f}_{mi} + n_m)\tau(x) \\ &= K(T^x - \tau(x)) = KT^{x+1} \end{aligned} \tag{5.18}$$

and thus Eq. 5.12 is satisfied in $x + 1^{th}$ iteration.

As the set of links $\{(s_i, t) : \forall s_i\}$ is also the minimum cut of $G^{'}$, the value of flow $\bar{f}$ on each link (s_i, t) is the capacity on the link. From flow conservation, we have $\sum_m \bar{f}_{mi} + \bar{f}_{\hat{p}i} = \lceil r_i \rceil - n_i$. From $\tau(x) \leq \tau_{min}^p$, we have $\tau(x) \leq \tau_{i\hat{p}} = (\lceil r_i \rceil - r_i)T^x$.

Consequently, we have

$$
\begin{aligned}
\sum_m \tau_{im}^{x+1} &= \sum_m \tau_{im}^{x} - (\sum_m \bar{f}_{mi} + n_i)\tau(x) \\
&= r_i T^x - (\lceil r_i \rceil - \bar{f}_{\hat{p}i})\tau(x) \\
&= \lceil r_i \rceil (T^x - \tau(x)) - (\lceil r_i \rceil - r_i) T^x + \tau(x)\bar{f}_{\hat{p}i} \\
&\leq \lceil r_i \rceil T^{x+1} \leq L T^{x+1}
\end{aligned}
$$

and hence Eq. 5.11 is satisfied in $x+1^{th}$ iteration.

From Eq. 5.7 and Eq. 5.8, if Eq. 5.13 is satisfied in x^{th} iteration, it is still satisfied in $x+1^{th}$ iteration.

Thus the theorem is proved. ∎

Corollary 3 *In each iteration, for each observation link $(p_m, s_i) \in \mathbb{E}_o^{\phi}$, there exists a path from s_i to the sink via only communication links with $F_{ij} > 0$.*

Proof: As Eq. 5.13 is satisfied in each iteration (flow conservation), for each sensor $s_i \in \mathcal{S}^*$ (the sensor with $\tau_{im} > 0$), there must exist a route from s_i to the sink via only links with $F_{ij} > 0$. As the set of sensors connected by $\mathbb{E}_o^{\phi}$ is a subset of $\mathcal{S}^*$, the corollary is proved. ∎

Corollary 4 *The observation assignment built in each iteration is feasible observation assignment.*

Proof: In the proof of Theorem 5, we have $\sum_i \bar{f}_{mi} = K - n_m$ in each iteration. As $\bar{f}$ can only take 0 or 1 values on observation links, there are $K - n_m$ sensors selected to observe target p_m in $\mathbb{E}_o^f$. Since there are n_m sensors selected to observes target p_m in $\mathbb{E}_o^{'}$, $\mathbb{E}_o^{'} \cap \mathbb{E}_o^f = \phi$ and $\mathbb{E}_o^{\phi} = \mathbb{E}_o^{'} \cup \mathbb{E}_o^f$, there are totally K sensors selected to observe each target p_m in the observation assignment built in each iteration.

Also, we have $\sum_m \bar{f}_{mi} + \bar{f}_{\hat{p}i} = \lceil r_i \rceil - n_i$ in each iteration. Each sensor s_i can observe at most $\lceil r_i \rceil - n_i$ targets in $\mathbb{E}_o^f$ and n_i targets in $\mathbb{E}_o^{'}$. As Eq. 5.11 is satisfied in each iteration, we have $\lceil r_i \rceil \leq L$. Each sensor s_i can observe at most L targets in the observation assignment built in each iteration.

Finally, as in corollary 3, we can find a route from each selected source node to the sink, e.g. via breath-first or depth-first search. Therefore, the observation assignment built in each iteration is feasible. Hence it is proved. ∎

Lemma 3 *In each iteration x, if $T > 0$, we have $\tau(x) > 0$.*

Proof: The operation duration $\tau(x)$ takes one of the following four values: $\tau^o_{min}, \tau^p_{min}, \tau^c_{min}, T - \tau^{'}_{max}$. As G^* contains only observation links with non-zero τ_{im}, we have $\tau^o_{min} > 0$. As all the observation links with $\tau^x_{im} = T^x$ are selected into the observation assignment, we have $\tau^{'}_{max} < T$. As all the links with zero pseudo capacity has been deleted in $G^{'}$, we have $\tau_{i\hat{p}} > 0$. As the route from each source node to the sink is built by links with $F^x_{ij} > 0$, we have $\tau^c_{min} > 0$. Hence it is proved. ∎

Theorem 6 *The decomposition algorithm terminates in at most $N(M+1) + N^2 + 1$ iterations with $\sum_x \tau(x) = T$ and has polynomial-time worst case time complexity.*

Proof: In any iteration x, the operation duration $\tau(x)$ takes one of the following four values: $\tau^o_{min}, \tau^p_{min}, \tau^c_{min}, T - \tau^{'}_{max}$. If $\tau(x) = \tau^c_{min}$, at least one positive element $F^x_{ij} > 0$ in the solution set is updated to be $F^{x+1}_{ij} = 0$. If $\tau(x) = \tau^o_{min}$, at least one positive element $\tau^x_{im} > 0$ in the solution set is updated to be $\tau^{x+1}_{im} = 0$.

If $\tau(x) = T - \tau^{'}_{max}$, at least one element $\tau^x_{im} < T^x$ in the solution set is updated to be $\tau^{x+1}_{im} = T^{x+1}$. If we have $\tau^x_{im} = T^x$ in the x^{th} iteration, as link (p_m, s_i) will be selected in the observation assignment, we have $\tau_{im}^{x+1} = T^{x+1}$ in the $x+1^{th}$ iteration, and thus we have $\tau_{im}^{z} = T^{z}$ in all the iterations with $z > x$.

If $\tau(x) = \tau_{min}^{p}$, let s_i be the sensor that has $\tau_{i\hat{p}} = \tau_{min}^{p}$. From the definition of τ_{min}^{p} we have $\bar{f}_{\hat{p}i} = 1$. As $\sum_m \bar{f}_{mi} + \bar{f}_{\hat{p}i} = \lceil r_i \rceil - n_i$, from Eq. 5.16 and Eq. 5.15, we have

$$\begin{aligned}\textstyle\sum_m \tau_{im}^{x+1} &= \textstyle\sum_m \tau_{im}^{x} - (\sum_m \bar{f}_{mi} + n_i)\tau(x) \\ &= r_i T^x + \tau(x) - \lceil r_i \rceil T^x + \lceil r_i \rceil (T^x - \tau(x)) \\ &= \tau(x) - \tau_{i\hat{p}} + \lceil r_i \rceil T^{x+1} = \lceil r_i \rceil T^{x+1}\end{aligned}$$

And thus at least one sensor s_i with $\tau_{i\bar{p}} > 0$ in the x^{th} iteration has $\tau_{i\bar{p}} = 0$ in the $x+1^{th}$ iteration. If a sensor s_i has $\tau_{i\bar{p}} = 0$ in the x^{th} iteration, as $\bar{f}_{\hat{p}i} = 0$, we have $r_i = \lceil r_i \rceil$ and $\sum_m \bar{f}_{mi} = \lceil r_i \rceil - n_i$, and thus we have $\sum_m \tau_{im}^{x+1} = \lceil r_i \rceil T^{x+1}$. Sensor s_i has $\tau_{i\bar{p}} = 0$ in all the following iterations $z > x$.

As there are at most N^2 communication links, at most NM observation links and at most N pseudo observation links, the algorithm will stop in at most $N(M+1) + N^2 + 1$ iterations. As the maximum flow on graph G' is at most $KM+N$, the time complexity of Ford-Fulkerson algorithm is $O(KM^2N+MN^2)$ [63]. The time complexity of bread-first search to find path from each source to the sink node is $O(N^3)$. Therefore, the time complexity of the decomposition algorithm is $O(N^2(KM^2 + MN + N^2)(M+N))$. From corollary 4 and lemma 3, the algorithm will continuously generate feasible observation assignments and non-zero operation duration until $T = 0$. As after each iteration T is updated by decreasing $\tau(x)$, we have $\sum_x \tau(x) = T$. ∎

Theorem 7 *The observation schedule constructed by the decomposition algorithm is feasible and the solution of the LP problem gives the optimal lifetime of LMOS-1 problem.*

Proof: The algorithm stops when $T = 0$. From Eq. 5.10 and Eq. 5.13, we have $\tau_{im} = 0$ and $F_{ij} = 0$ when $T = 0$, and thus we have $\sum_x \tau_{im}^x = \tau_{im}$ and $\sum_x F_{ij}^x = F_{ij}$. From Eq. 5.6 we can conclude that the energy consumption of each node s_i after executing the constructed schedule is less equal to $E_0(s_i)$. Since each observation assignment in the schedule is feasible (corollary 4), the constructed observation schedule is feasible.

As the LP solution gives an upper bound of the optimal lifetime of LMOS problem, and we can find an feasible observation schedule (solution of LMOS problem) that achieves this upper bound, the solution of the LP problem gives the optimal lifetime of LMOS-1 problem. ∎

5.2.4 Numerical example

In this section we present a numerical example to illustrate the LMOS-1 problem and solution. There are 7 nodes and 3 targets randomly scattered in an 100×100 area. The sink node is placed at the top left corner of the area. The coverage requirement K is set as 3 and the observation constraint L is set as 2. The value of various parameters are chosen to be $e_t = 50nJ/bit$, $b = 100pJ/bit/m^4$, $\alpha = 4$, $e_r = 150nJ/bit$ and $e_s = 150nJ/bit$ [69]. We assume that the target rate of each target is the same – $1kbps$, and each sensor has the same initial energy $20J$. The sensing range is set as 60m and the communication range is set as 100m.

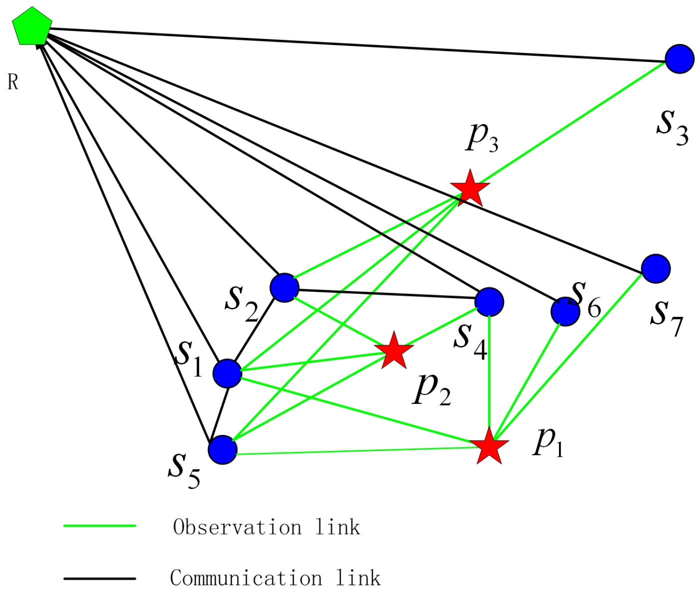

Figure 5.2: Network topology with non-zero links in the LP solution

In the first step, we formulate and solve the LP problem as described in section 5.2.1. The maximized lifetime obtained is $T = 11$. Figure 5.2 shows the network topology with those observation links (p_m, s_i) and communication links (s_i, s_j) that have non-zero values of τ_{im} or F_{ij} in the solution of LP problem. The values of $\{\tau_{im}\}$ in the LP solution set are listed in Table 5.2, for example, $\tau_{11} = 9.8$ and $\tau_{12} = 1.2$. We normalize each value of F_{ij} in the LP solution by the target rate (1Kbps) and label the normalized values on the corresponding links in Fig. 5.3, e.g. $F_{12} = 3.1$ and $F_{4R} = 6.6$. Then we execute the decomposition algorithm. The calculated pseudo observation durations are also shown in Table 5.2, e.g. $\tau_{3\bar{p}} = 7.3$.

First we build the flow network G^* as in Fig. 5.4-a. The dashed lines denote the observation links with $\tau_{im} = T$, e.g. observation link (p_1, s_4) is dashed line as $\tau_{41} = T = 11$. The values labeled on the links from the sender s to each target or

Table 5.2: Values of $\{\tau_{im}\}$ in the LP solution and $\{\tau_{i\bar{p}}\}$

$Targets$	Nodes						
	s_1	s_2	s_3	s_4	s_5	s_6	s_7
p_1	9.8	0	0	11	3.7	5.1	3.4
p_2	1.2	11	0	9.8	11	0	0
p_3	11	11	3.7	0	7.3	0	0
$\bar{p}$	0	0	7.3	1.2	0	5.9	7.6

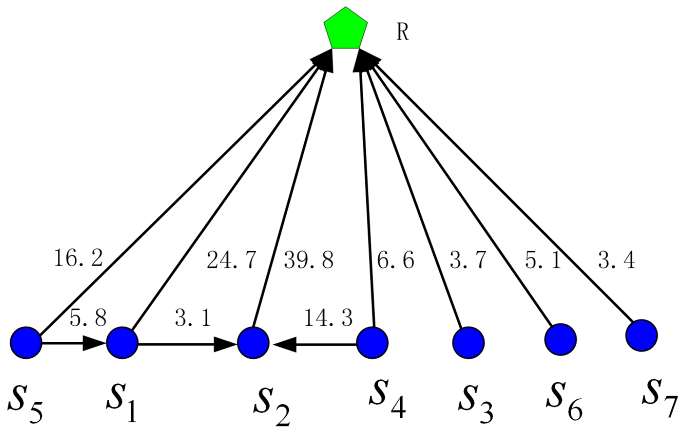

Figure 5.3: Normalized $\{F_{ij}\}$ in the LP solution

from each sensor to the destination t are the link capacities on these links, i.e. the capacity from s to $\bar{p}$ is 1. Next we find the maximum flow on graph G'. The links that the maximum flow passes through together with the observation links that have $\tau_{im} = T$ are shown in Fig. 5.4-b. It can be easily examined that the observation links shown in Fig. 5.4-b cover each target 3 times and cover each sensor less than 2 times. In Fig. 5.4-c from each sensor to the sink a path is found via only communication links with non-zero F_{ij}. Fig. 5.4-d shows the constructed observation assignment by appending the selected observation links with the path from each source to the sink.

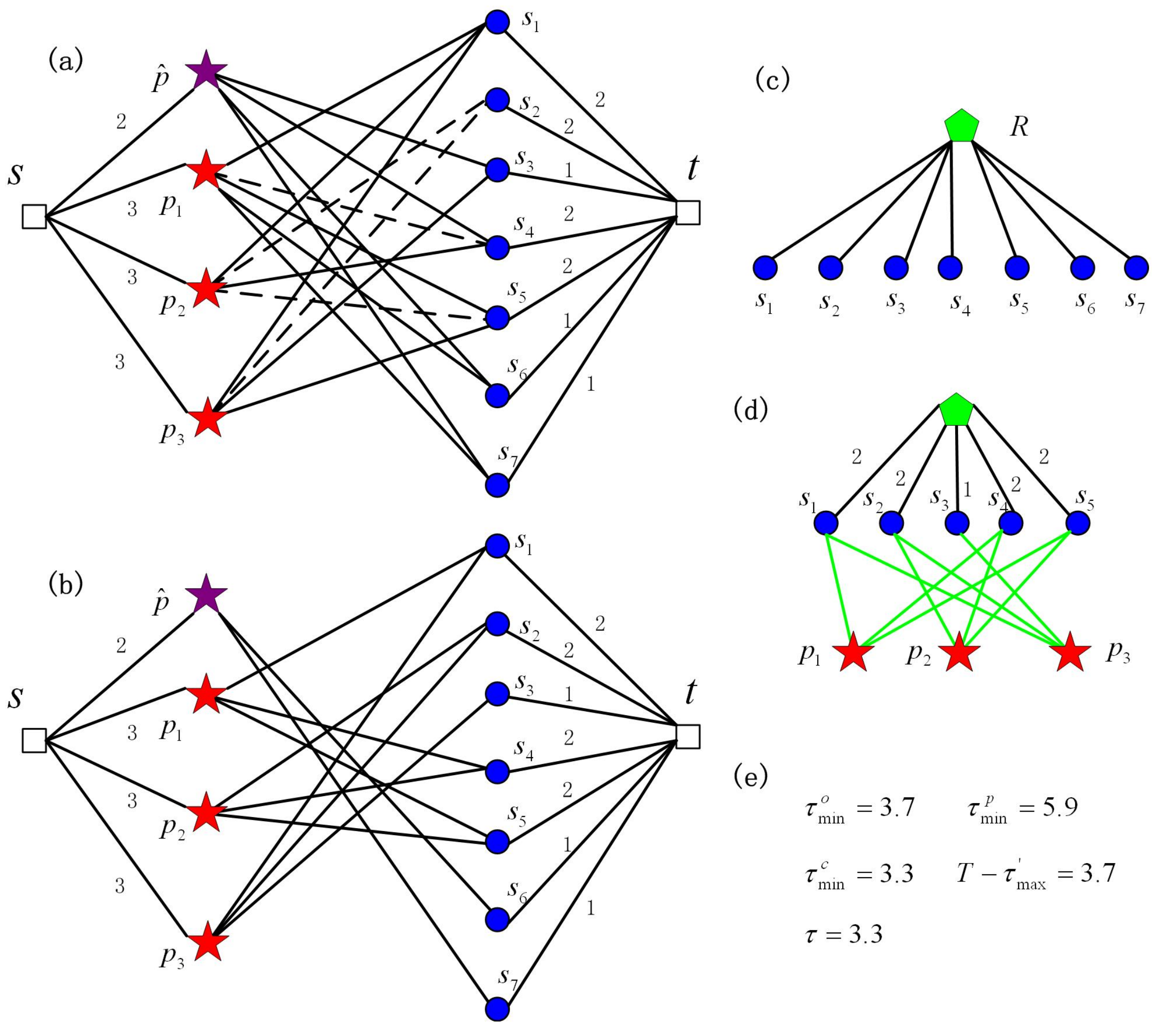

Figure 5.4: Illustration of the decomposition algorithm

The values labeled on each communication link is the normalized (by target rate) flow rate on the link when the observation assignment is executed. Finally, we calculate the observation duration of the assignment τ as in Fig. 5.4-e: $\tau^o_{min} = \tau_{51} = 3.7$; $\tau^p_{min} = \tau_{6\bar{p}} = 5.9$; $\tau^c_{min} = F_{4R}/2 = 3.3$; $T - \tau'_{max} = 11 - \tau_{53} = 3.7$. Therefore, we have $\tau = 3.3$. After that, the solution set is updated and we have $T = T - \tau = 7.7$. The updated $\{\tau_{im}\}$ and $\{F_{ij}\}$ is shown in Table. 5.3 and Fig. 5.5.

Table 5.3: Values of $\{\tau_{im}\}$ and $\{\tau_{i\bar{p}}\}$ after the first update

Targets	Nodes						
	s_1	s_2	s_3	s_4	s_5	s_6	s_7
p_1	6.5	0	0	7.7	0.4	5.1	3.4
p_2	1.2	7.7	0	6.5	7.7	0	0
p_3	7.7	7.7	0.4	0	7.3	0	0
$\bar{p}$	0	0	7.3	1.2	0	2.6	4.3

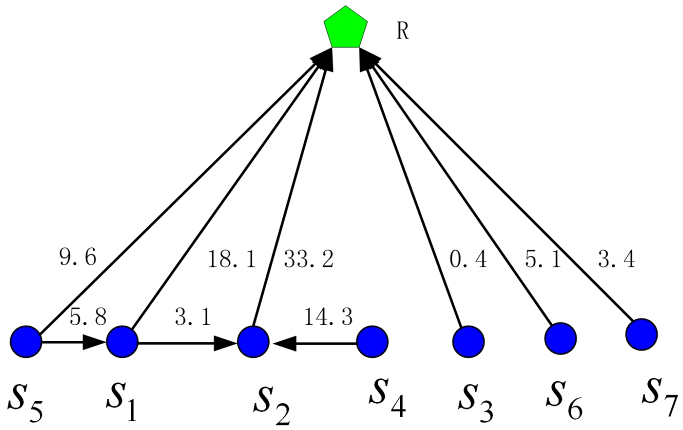

Figure 5.5: Normalized $\{F_{ij}\}$ after the first update

The above procedures are repeated until the total observation duration equals to the network lifetime. The observation schedule obtained is shown in Fig. 5.6.

5.2.5 Performance Study

In this section we study the impact of various network parameters on the network lifetime for LMOS-1 problem, including the number of nodes – N, the number of targets – M, the coverage constraint – K and the observation constraints L. We consider stationary networks with sensor nodes and targets uniformly located in a square of $100m \times 100m$ area. The sink node is placed in the middle of the area.

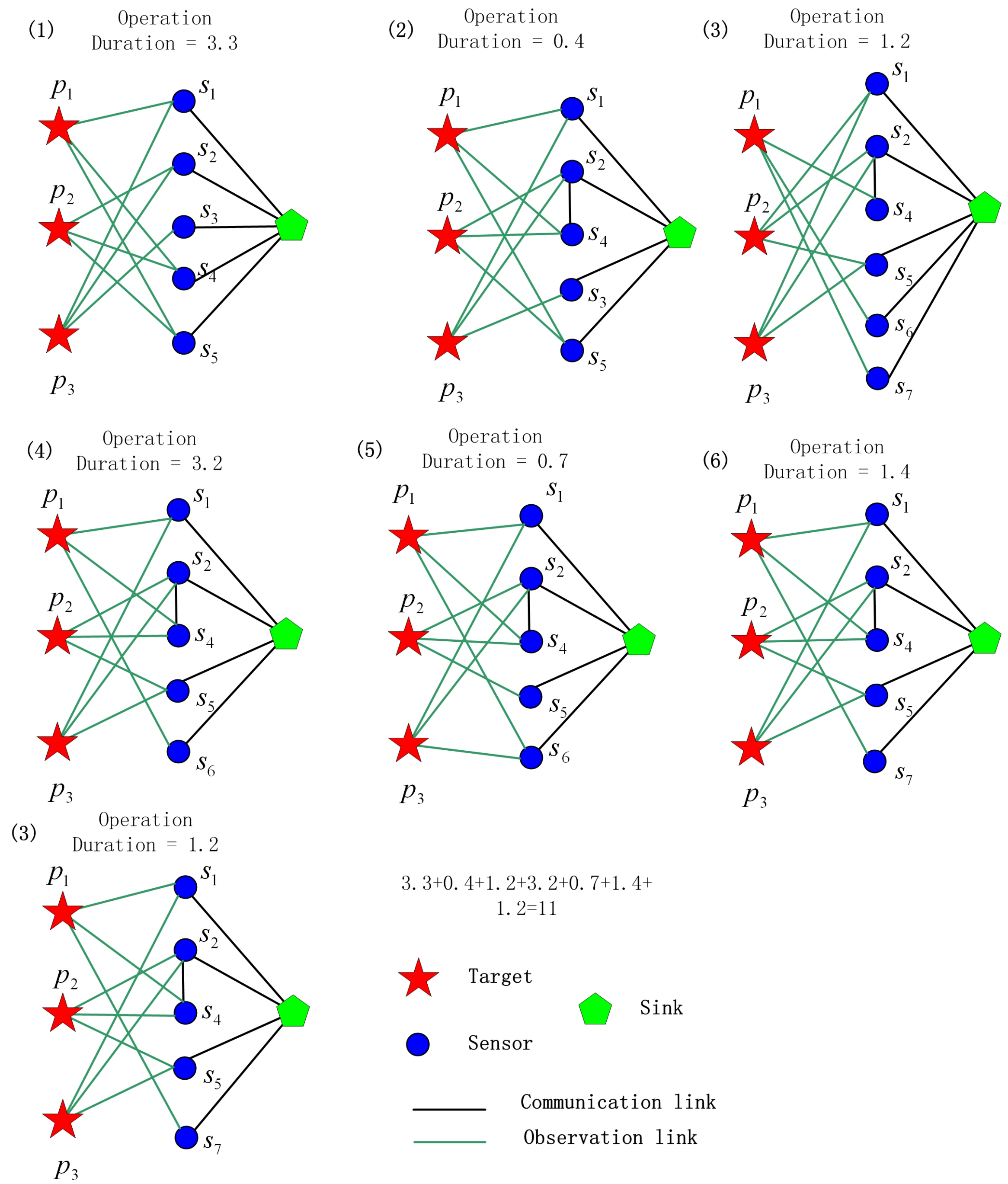

Figure 5.6: The optimal observation schedule

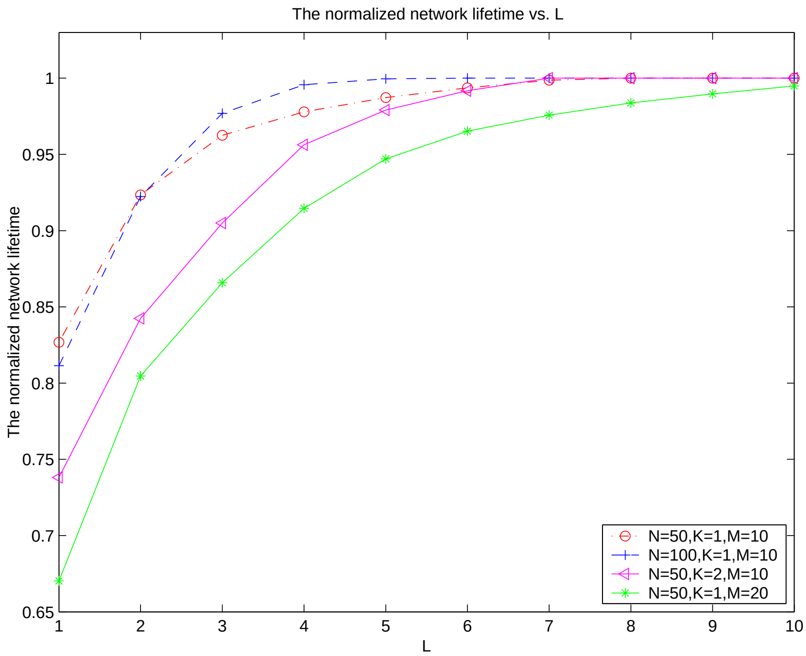

Figure 5.7: Normalized network lifetime vs. L

The sensing range and communication range of each node $R_s = 40m$, $R_c = 40m$. The value of various parameters are chosen to be $e_t = 50nJ/bit$, $b = 100pJ/bit/m^4$, $\alpha = 4$, $e_r = 150nJ/bit$ and $e_s = 150nJ/bit$ [69]. We assume that each target produces data at the same rate $10kbps$, and each sensor has the same initial energy $20J$. Each value plotted in the figure or shown in the table is the average result of 100 randomly generated topologies.

In Fig. 5.7 we study the impact of L - the number of targets a sensor can simultaneously observe - on the network lifetime with different values of K, N and M. The value of each point is normalized by the network lifetime achieved corresponding to $L = \infty$. It can be observed that the network lifetime is improved as L becomes larger. When the number of targets is larger or the coverage requirement is higher,

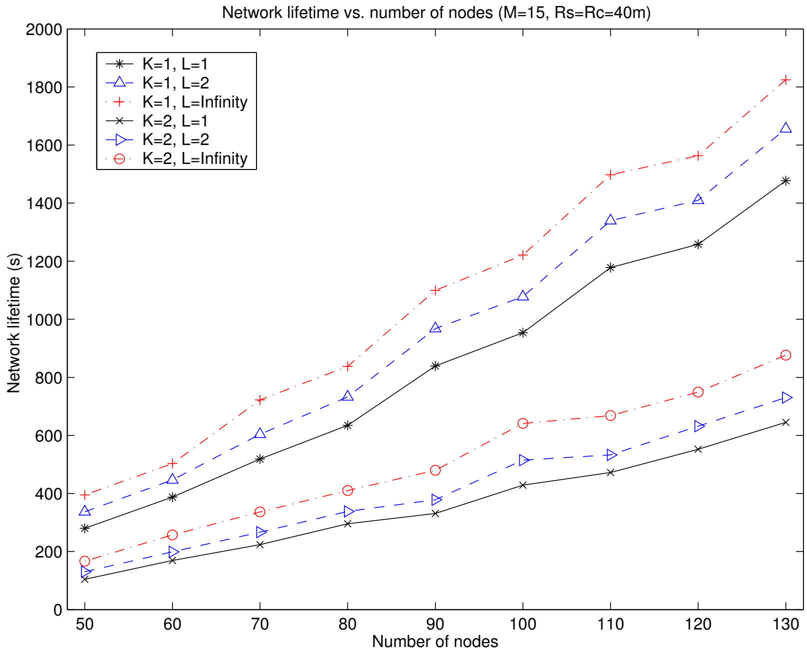

Figure 5.8: Network lifetime vs. number of nodes ($M = 15$)

the improvement of lifetime with the increasing L becomes more significant.

In Fig. 5.8 we study the impact of the number of sensors N on the network lifetime for different values of K and L. The number of targets in the network is fixed at 15. It can be observed that the network lifetime increases nearly linearly as the number of nodes increases. This is because, with more number of nodes, possibly large number of observation assignments can be built. In Fig. 5.9 we study the impact of the number of targets M on the network lifetime for different values of K and L. The number of sensors in the network is fixed at 100. The decreasing trend of network lifetime can be observed as the number of targets increases. This is because, more the number of targets, the larger the data generated, and the larger the energy consumed.

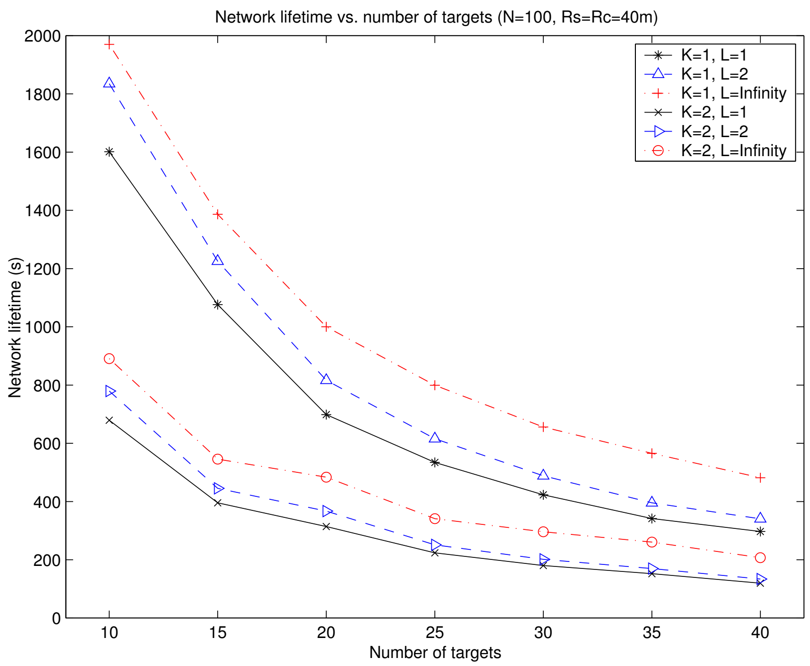

Figure 5.9: Network lifetime vs. number of targets ($N = 100$)

5.3 NP-Completeness of LMOS-2 problem

We can show LMOS-2∈NP by verifying in polynomial time whether a non-deterministically selected observation schedule is feasible. Further, LMOS-2 problem can be proven to be a NP-Complete problem as the Maximum Set Covers (MSC) problem [16], which has been proven to be NP-complete, is a special case of LMOS-2 problem when 1) each node communicates directly with the sink node, 2) $K = 1$ and 3) $(e_s + e^t_{i\mathcal{R}}) \sum_{p_m \in \mathcal{P}_i} r_m$ is the same for each sensor s_i, where $\mathcal{P}_i$ is the set of targets that are in the sensing area of sensor s_i.

5.3.1 Upper bound and lower bound of LMOS-2 problem

Given an instance of LMOS-2 problem, we can build an instance of LMOS-1 problem with the same initial requirements and $L = \infty$. Let T_1 denote the optimal lifetime of the LMOS-1 instance and let T_2 denote the optimal lifetime of the LMOS-2 instance. let $\mathcal{M} = \max_i \max_{m \in \mathcal{P}_i} \frac{\sum_{m \in \mathcal{P}_i} r_m}{r_m}$. We make the following claim.

Claim 1 $T_1 \geq T_2 \geq \frac{1}{\mathcal{M}} T_1$.

Proof: Given any feasible solution of the LMOS-2 instance with lifetime T_2, we can build a feasible solution of the corresponding LMOS-1 instance with lifetime T_2 by building the same observation assignments and setting the same operation durations as in the solution of LMOS-2 and assigning each source node to observe all the targets in it sensing area since $L = \infty$. Thus we have $T_1 \geq T_2$. Given any feasible solution of the LMOS-1 instance with lifetime T_1, we can build a feasible solution of the LMOS-2 instance by assigning the same set of source nodes as in the solution of LMOS-1 instance. The data generated by each node in LMOS-2 solution is at most $\mathcal{M}$ times of that in LMOS-1 solution, and thus we have $T_2 \geq \frac{1}{\mathcal{M}} T_1$. ∎

5.4 Summary

In this chapter, we considered the CTC problem when the data generation rate of a sensor is proportional to the number of targets observed by it and with K coverage requirement wherein each target is observed by at least K sensors. We modeled the CTC problem in this case as a Lifetime Maximization Observation Schedule (LMOS) problem and discussed the problem with two different observation scenarios depending on whether a sensor can select a subset of targets in its sensing area to observe or not. We proved that the LMOS problem for the first scenario (LMOS-1) is a P problem and developed a polynomial-time algorithm which can achieve the optimal solution based on Linear Programming and Integer Theorem. We showed that the LMOS problem for the second scenario (LMOS-2) is NP complete. We derived an upper bound and a lower bound of the LMOS-2 problem based on the optimal solution of the LMOS-1 problem.

Chapter 6

Approximation and Heuristic algorithms for the LMOS problem

In this chapter approximation algorithms for both LMOS-1 and LMOS-2 problems are developed. They provide insights into the LMOS problem and can be used to evaluate and compare the performance of other algorithms. As a practical implementation we develop a faster flexible heuristic algorithm called Communication Weighted Observation Scheduling (CWOS) for both problems which can be implemented in a distributed fashion. We carry out extensive simulations to demonstrate the effectiveness of the proposed heuristic algorithm by comparing its performance with that of the optimal solution for the LMOS-1 problem and the approximation algorithm of the LMOS-2 problem.

6.1 Approximation algorithm for the LMOS problem

6.1.1 LP packing formulation and dual problem

Given an instance of the LMOS-1 problem, let us enumerate all the possible sets of observation links $U = \{U_1, U_2, \cdots, U_Q\}$ in the feasible observation assignment of the instance, such that each set $U_q \in U$ contains a set of observation links satisfying

K-coverage requirement and L observation constraint. Thus, the set of observation links of any feasible observation assignment of the LMOS-1 problem instance would be in U. Let τ_q denote the duration that U_q is selected as the set of observation links in the observation assignment, where q denotes the index of set of observation links in U; Let F_i^s denote the total amount of data that are generated by node s_i when it is selected as the source node; Let N_i denote the set of neighbors of sensor s_i; Let P_i denote the set of targets in the sensing area of sensor s_i; Let X_{im}^q be 1 if the observation link (p_m, s_i) belongs to the set U_q, otherwise 0. The LMOS-1 problem can be formulated as follows:

$$\text{Maximize: } \sum_{1 \leq q \leq Q} \tau_q \tag{6.1}$$

$$\sum_{1 \leq q \leq Q} \sum_{1 \leq M} X_{im}^q r_m \cdot \tau_q - F_i^s = 0; \quad \forall s_i \in \mathcal{S} \tag{6.2}$$

$$-\sum_{j \in N_i} F_{ij} + \sum_{j \in N_i} F_{ji} + F_i^s - F_{i\mathcal{R}} = 0; \quad \forall s_i \in \mathcal{S} \tag{6.3}$$

$$\sum_{j \in N_i} F_{ij} e_{ij}^t + \sum_{j \in N_i} F_{ji} e_r + F_i^s e_s + F_{i\mathcal{R}} e_{i\mathcal{R}}^t \leq E_0(s_i); \forall s_i \in \mathcal{S} \tag{6.4}$$

Equations 6.2 and 6.3 are the flow conservation constraints. Equation 6.4 is the energy consumption constraint. Eq. 6.3 and Eq. 6.4 are the same as Eq. 4.3 and Eq. 4.4 in Chapter 4.

Given an instance of the LMOS-2 problem, let $U = \{U_1, U_2, \cdots, U_Q\}$ enumerates all possible sets of sources in the feasible observation assignments. Let X_{iq} be 1 if node s_i belongs to the set U_q, otherwise 0. The LMOS-2 problem can be formulated as a LP problem the same as the LP formulation of LMOS-1 problem but replacing

Eq. 6.2 by

$$\sum_{1\leq q\leq Q} X_{iq}\cdot\sum_{m\in\mathcal{P}_i} r_m\tau_q - F_i^s = 0;\quad \forall s_i\in\mathcal{S} \tag{6.5}$$

6.1.2 The dual problem and its interpretation

The dual problem of the LP formulation of LMOS-1 problem is as follows:

$$\text{Minimize: }\sum_{1\leq i\leq N} c_i E_0(s_i) \tag{6.6}$$

$$-a_i + a_j + e_{ij}^t c_i + e_r c_j \geq 0;\quad \forall i\neq j, s_j\in N_i \tag{6.7}$$

$$e_{i\mathcal{R}}^t c_i - a_i \geq 0;\quad \forall s_i\in N_{\mathcal{R}} \tag{6.8}$$

$$-b_i + a_i + e_s c_i \geq 0;\quad \forall 1\leq i\leq N \tag{6.9}$$

$$\sum_{1\leq i\leq N}\sum_{1\leq m\leq M} X_{im}^q r_m\cdot b_i \geq 1;\quad \forall 1\leq q\leq Q \tag{6.10}$$

where $a_i, b_i, c_i \geq 0$ $(1\leq i\leq N)$ are variables in the dual problem. Eq. 6.7, Eq. 6.8 and Eq. 6.9 are the same as Eq. 4.6, Eq.4.7 and Eq. 4.8 in Chapter 4. The dual problem of the LP formulation of LMOS-2 problem is the same as the above problem but replacing Eq. 6.10 by

$$\sum_{1\leq i\leq N} X_{iq}\sum_{m\in\mathcal{P}_i} r_m\cdot b_i \geq 1;\quad \forall 1\leq q\leq Q \tag{6.11}$$

The dual problem can be interpreted as a problem of assigning weights to the links in the network. Let $\vec{C}$ be the vector such that its i^{th} element is c_i. We define the objective function of the dual problem $D(\vec{C})$, the link weight $w_{ij}(\vec{C})$ for each link $(s_i, s_j)\in\mathbb{E}$, the node weight $w_i(\vec{C})$ for each node s_i and the path weight $P = \{s_i, n_1, n_2, \cdots, n_l, \mathcal{R}\}$

which are the same as Eq. 4.10, Eq. 4.11, Eq. 4.12, and Eq. 4.17 in Chapter 4. The dual problem of the LMOS-1 problem can now be re-written as:

$$\text{Minimize:}D(\vec{C}) \tag{6.12}$$

$$w_{ij}(\vec{C}) \geq \begin{cases} a_i - a_j, & \text{if } j \neq \mathcal{R}, \text{ link } (s_i, s_j) \in \mathbb{E}; \\ a_i, & \text{if } j = \mathcal{R}, \text{ link } (s_i, \mathcal{R}) \in \mathbb{E}; \end{cases} \tag{6.13}$$

$$w_i(\vec{C}) \geq b_i - a_i, \quad \forall s_i \in \mathcal{S} \tag{6.14}$$

$$\sum_{1 \leq i \leq N} \sum_{1 \leq m \leq M} X_{im}^q r_m \cdot b_i \geq 1; \quad \forall 1 \leq q \leq Q \tag{6.15}$$

The dual problem of the LMOS-2 problem is the same as that of LMOS-1 but replacing Eq. 6.15 by:

$$\sum_{1 \leq i \leq N} X_{iq} \sum_{m \in \mathcal{P}_i} r_m \cdot b_i \geq 1; \quad \forall 1 \leq q \leq Q \tag{6.16}$$

Using Eq. 6.13 and 6.14, we have

$$w_P(\vec{C}) \geq b_i - a_i + a_i - a_{n_1} + \cdots + a_{n_l} = b_i \tag{6.17}$$

Let $w_{SPT}^i(\vec{C})$ denote the path weight of the shortest path (path with the minimum path weight) from node s_i to the sink. For LMOS-1, we define

$$\alpha(\vec{C}) = \min_{1 \leq q \leq Q} \left\{ \sum_i \sum_m X_{im}^q r_m \cdot w_{SPT}^i(\vec{C}) \right\} \geq 1 \tag{6.18}$$

For LMOS-2, we define

$$\alpha(\vec{C}) = \min_{1 \leq q \leq Q} \left\{ \sum_i X_{iq} \sum_{m \in \mathcal{P}_i} r_m \cdot w_{SPT}^i(\vec{C}) \right\} \geq 1 \tag{6.19}$$

The dual problem is then equivalent to assigning values to $\vec{C}$ such that $D(\vec{C})/\alpha(\vec{C})$ is minimized subject to the constraint that $\alpha(\vec{C}) \geq 1$. Let

$$\beta = \min \left\{ \frac{D(\vec{C})}{\alpha(\vec{C})} \right\} \tag{6.20}$$

6.1.3 Algorithm description

The above interpretation of the dual LP problem leads to our approximation algorithm, which is given below:

1. Initialization

   (a) Properly scale the problem so that $\beta \geq 1$;

   (b) $t = 0$; $T = 0$; Set δ; For each node s_i set $c_i = \delta / E_0(s_i)$; Let $\lambda = \log_{1+\epsilon}^{1/\delta}$; Set $\tau_p = 1/\lambda$;

2. Set $\tau^t = 0$, $k = 0$, loop until $\tau^t = \tau_p$;

   (a) $k = k + 1$; Build the shortest path tree rooted at the sink with the link weight function $w_{ij}(\vec{C})$ and the path weight function $w_P(\vec{C})$;

   (b) If LMOS-1:

      i. Call *FindOSlink* subroutine to find a set of observation links Ψ_k^t such that K coverage requirement is satisfied for each target, L observation constraint is satisfied for each sensor, and the value of $\sum_{(p_m, s_i) \in \Psi_k^t} r_m w_{SPT}^i(\vec{C})$ is minimized; If return FALSE, exit;

      ii. The observation links in Ψ_k^t and the shortest path from each sensor in S_k^t to the sink forms the observation assignment ϕ_k^t

   (c) If LMOS-2:

      i. $S_k^t = \phi$; for $\forall s_i \in \mathcal{S}$, $\mathcal{P}_i^{tk} = \mathcal{P}_i$; Until all the targets are covered K times by S_k^t, do:

A. select $s_i \notin S_k^t$ with the minimum $\frac{\sum_{m \in \mathcal{P}_i} r_m w_{SPT}^i(\vec{C})}{|\mathcal{P}_i^{tk}|}$; Add s_i into S_k^t;

B. for each sensor $s_j \notin S_k^t$, $\mathcal{P}_j^{tk} = \mathcal{P}_j^{tk} - \mathcal{P}_i^{tk} \cap \mathcal{P}_j^{tk}$.

ii. The shortest path from each sensor in S_k^t to the sink constructs the observation assignment ϕ_k^t;

(d) The operation duration τ_k^t of ϕ_k^t ends when $\tau_k^t = \tau_p - \tau^t$ or any node s_i in ϕ_k^t consumes $E_0(s_i)/\lambda$ unit of energy;

(e) $\tau^t = \tau^t + \tau_k^t$; Let e_i^{tk} denote the amount of energy that node s_i has consumed in duration τ_k^t, $c_i(t,k) = c_i(t,k-1) \times (1 + \epsilon \cdot \frac{\lambda e_i^{tk}}{E_0(s_i)})$.

3. $t = t+1$; $c_i(t,0) = c_i(t-1,k)$; if $D(\vec{C}) < 1$, $T = T + \tau_p$.

4. Double τ_p every 2λ iterations; repeat step 2 and step 3 until $D(\vec{C}) \geq 1$.

The FindOSlink subroutine (step 2(b)i) finds the set of observation links in each constructed observation assignment of LMOS-1 problem, and is described as follows:

1. Build network flow graph G^o with all the observation links, the sensors and targets connected with the observation links, a source node s connected with all the targets and a destination node t connected with all the sensors; Assign link capacity K and link cost 0 to the link from s to each target; Assign link capacity L and link cost 0 to the link from each sensor to the destination t; Assign link capacity 1 and link cost $r_m w_{SPT}^i(\vec{C})$ to each observation link (p_m, s_i);

2. For each link $(u,v) \in G^o$, set $f_{uv} = f_{vu} = 0$;

3. **While** there exists path from s to t in the residual network G_f^o;

(a) Find the minimum cost path p from s to t in G_f^o;

(b) For each edge (u, v) in p, $f_{uv} = f_{uv} + 1$, $f_{vu} = -f_{uv}$;

(c) If $\sum_{(s,v)\in G^o} f_{sv} = KM$, return the set of observation links (s_i, p_m) with $f_{im} = 1$; else return FALSE

The output of the algorithm is T, ϕ_k^t and τ_k^t, which are the network lifetime, observation assignments and their operation durations, respectively. Similar to the approximation algorithm presented in chapter 4, at the beginning, we scale the problem such that $\beta \geq 1$ (step 1a). We then set the initial value for δ, λ and c_i. For LMOS-1 problem, δ is set as $(\frac{N}{1-\epsilon})^{-1/\epsilon}$; for LMOS-2 problem, δ is set as $\delta = (\frac{N}{1-\epsilon H(\hat{M})})^{-1/\epsilon}$, where $\hat{M}$ is the maximum number of targets in the sensing area of a sensor. The algorithm then proceeds in loops. As in chapter 4, we call the outer loop of steps as an iteration, and call the inner loop of steps as a phase in the iteration. The duration of each iteration is τ_p. Each iteration may be composed of multiple phases.

Let us define the communication weight of a link as the multiplication of the link weight and the flow on the link. In each phase, we try to find a feasible observation assignment, such that the total communication weight from all the source nodes to the sink is minimized (step 2).

For LMOS-1 problem, we find the desired observation assignment by solving a minimum cost flow problem. In step 2a we build a shortest path tree rooted at the sink node and each node get the shortest path from itself to the sink. We find the set of observation links in the desired observation assignment in step 2(b)i using the FindOSlink subroutine. After the flow network is built in step 1 of the subroutine, the minimum cost maximum flow in the constructed network gives us the observation links in the minimum communication weight observation assignment. The subroutine solves the minimum cost maximum flow problem by repeatedly admitting a minimum cost flow in the residual network (step 3 in subroutine). The residual network G_f^o of G^o consists of links that can admit more flows (pp. 651 in [63]). The cost of link in the residual network is the same as the link in the original network but is negative if the direction of the link is adverse. The subroutine returns FALSE only when there does not exist a feasible observation assignment and the approximation algorithm will exit.

For LMOS-2 problem, we use the concept of greedy algorithm for weighted set multi-cover problem to select the set of sources. The weighted set multi-cover problem is an extension to the weighted set cover problem by requiring that each element is covered by multiple times. Considering $\mathcal{P}_i$ as the subset and $w_{SPT}^i(\vec{C})$ as the subset weight in weighted set multi-cover problem, denoting $\mathcal{P}_i^{tk}$ as the set of targets that have not been K-covered in $\mathcal{P}_i$, we greedily select the sensor that has the minimum value of $\frac{\sum_{m \in \mathcal{P}_i} r_m w_{SPT}^i(\vec{C})}{|\mathcal{P}_i^{tk}|}$ as the source node until all the targets are covered K times. The shortest path from each selected source node to the sink builds the observation assignment.

The duration of the phase (operation duration of the observation assignment) ends when the duration of the iteration ends or any node s_i consumes $E_0(s_i)/\lambda$ units of energy in the phase. Once the observation links are found, appending the shortest paths from each selected source node to the sink builds the desired observation assignment. The value of c_i will be updated according to the energy consumption of node s_i in the phase.

We present the method to scale the problem (in step 1a), analyze the approximation ratio and discuss the value of λ, δ in the next section (section 6.1.4). Then we analyze the complexity in section 6.1.5.

6.1.4 Analysis

We first explain the scaling method used in our algorithm so that $\beta \geq 1$. For LMOS-1 problem, let T_{LP} denote the optimal network lifetime achieved by solving the LP problem presented in Chapter 5. Thus scaling the initial energy of each node by $T_{LP}/2$ can guarantee that $2 \geq \beta \geq 1$. For LMOS-2 problem, from claim 1, we have $T_{LP}/\mathcal{M} \leq \beta \leq T_{LP}$. Thus scaling the initial energy of each node by $T_{LP}/\mathcal{M}$ can guarantee that $\mathcal{M} \geq \beta \geq 1$.

Using the definition of path weight $w_P(\vec{C})$, we have

$$w_P(\vec{C}) = (e_s + e^t_{i,n_1})c_i + (e_r + e^t_{n_l,\mathcal{R}})c_{n_l} + \sum_{1 \leq z < l} (e_r + e^t_{n_z,n_{z+1}})c_{n_z} \tag{6.21}$$

Let K_t denote the number of phases in iteration t, $\vec{C}(t,k)$ denote the vector of c_i after the k^{th} phase of iteration t. Assuming that the q^{th} set in U is selected as the set of observation links or sources in the k^{th} phase of iteration t, the value of $D(\vec{C})$ at the end of this phase is

$$D(\vec{C}(t,k)) = \sum_i c_i(t,k)E_0(s_i) \tag{6.22}$$

$$= D(\vec{C}(t,k-1)) + \epsilon \sum_i \lambda e^{tk}_i c_i(t,k-1) \tag{6.23}$$

For LMOS-1 problem, we have

$$D(\vec{C}(t,k)) = D(\vec{C}(t,k-1)) + \epsilon\tau_k^t\lambda\sum_i\sum_m X_{im}^q r_m w_{SPT}^i(\vec{C}(t,k-1)) \tag{6.24}$$

For LMOS-2 problem, we have

$$D(\vec{C}(t,k)) = D(\vec{C}(t,k-1)) + \epsilon\tau_k^t\lambda\sum_i X_{iq}\sum_{m\in\mathcal{P}_i} r_m w_{SPT}^i(\vec{C}(t,k-1)) \tag{6.25}$$

For LMOS-1 problem, as the FindOSlink subroutine find the set of observation links of a feasible observation assignment with minimum $\sum_{(p_m,s_i)\in\psi_k^t} r_m w_{SPT}^i(\vec{C})$, we have

$$\sum_i\sum_m X_{im}^q w_{SPT}^i(\vec{C}(t,k-1)) = \alpha(\vec{C}(t,k-1)) \tag{6.26}$$

For LMOS-2 problem, as greedy algorithm is a $H(k)$ approximation algorithm for weighted set multi-cover problem [92] where $H(k) = \sum_{1\le i\le k}\frac{1}{i}$ and k is the maximum subset size, we have

$$\sum_i X_{iq}\sum_{m\in\mathcal{P}_i} r_m w_{SPT}^i(\vec{C}(t,k-1)) \le H(\hat{M})\alpha(\vec{C}(t,k-1)) \tag{6.27}$$

As the algorithm proceeds, the link weights are monotonically non-decreasing. Therefore, for both problems,

$$\alpha(\vec{C}(t,k-1)) \le \alpha(\vec{C}(t,k)) \tag{6.28}$$

If τ_p is never doubled, $\sum_{1\le k\le K_{t-1}}\tau_k^{t-1} = \tau_p = 1/\lambda$. We assume that τ_p is never doubled now and will explain latter why the approximation ratio still holds when this assumption is removed in section 6.1.5. For any iteration $t \ge 1$, for LMOS-1 problem, we have

$$\begin{aligned} D(\vec{C}(t,0)) &= D(\vec{C}(t-1,0)) + \epsilon\lambda\sum_{1\le k\le K_{t-1}}\tau_k^{t-1}\alpha(\vec{C}(t-1,k-1)) && (6.29)\\ &\le D(\vec{C}(t-1,0)) + \epsilon\lambda\sum_{1\le k\le K_{t-1}}\tau_k^{t-1}\alpha(\vec{C}(t,0)) && (6.30)\\ &\le D(\vec{C}(t-1,0)) + \epsilon\alpha(\vec{C}(t,0)) && (6.31)\end{aligned}$$

For any iteration $t \geq 1$, for LMOS-2 problem, we have

$$D(\vec{C}(t,0)) \leq D(\vec{C}(t-1,0)) + \epsilon\lambda \sum_{1 \leq k \leq K_{t-1}} \tau_k^{t-1} H(\hat{M})\alpha(\vec{C}(t-1,k-1)) \quad (6.32)$$

$$\leq D(\vec{C}(t-1,0)) + \epsilon\lambda \sum_{1 \leq k \leq K_{t-1}} \tau_k^{t-1} H(\hat{M})\alpha(\vec{C}(t,0)) \quad (6.33)$$

$$\leq D(\vec{C}(t-1,0)) + \epsilon H(\hat{M})\alpha(\vec{C}(t,0)) \quad (6.34)$$

Since $\beta = \min\left\{D(\vec{C})/\alpha(\vec{C})\right\} \leq \frac{D(\vec{C}(t,0))}{\alpha(\vec{C}(t,0))}$ and $D(\vec{C}(0,0)) = N\delta$, for LMOS-1 problem, we have

$$D(\vec{C}(t,0)) \leq \frac{D(\vec{C}(t-1,0))}{1-\epsilon/\beta} \leq \frac{N\delta}{(1-\epsilon/\beta)^t} \quad (6.35)$$

$$= \frac{N\delta}{1-\epsilon/\beta}(1+\frac{\epsilon}{\beta-\epsilon})^{(t-1)} \quad (6.36)$$

$$\leq \frac{N\delta}{1-\epsilon/\beta}e^{\frac{\epsilon(t-1)}{\beta-\epsilon}} \leq \frac{N\delta}{1-\epsilon}e^{\frac{\epsilon(t-1)}{\beta(1-\epsilon)}} \quad (6.37)$$

The last inequality uses the assumption that $\beta \geq 1$. Let N_t denote the iteration number when the algorithm ends, i.e. We have

$$\frac{\beta}{N_t - 1} \leq \frac{\epsilon}{(1-\epsilon)\ln(\frac{1-\epsilon}{N\delta})} \quad (6.38)$$

Similarly, for LMOS-2 problem, we have

$$\frac{\beta}{N_t - 1} \leq \frac{\epsilon H(\hat{M})}{(1-\epsilon H(\hat{M}))\ln(\frac{1-\epsilon H(\hat{M})}{N\delta})} \quad (6.39)$$

In each iteration, the network lifetime will be increased by a duration of $1/\lambda$. Since the algorithm ends when $D(\vec{C}(N_t,0)) \geq 1$, the network lifetime is $T = (N_t - 1)/\lambda$.

Lemma 4 *The solution of our approximation algorithm is a feasible solution for both the LMOS-1 and LMOS-2 problem,* $(N_t - 1)/\lambda$ *is strictly less than the optimal solution of the LMOS problem.*

Proof: Similar to the proof of lemma 2 presented in Chapter 4. ∎

Theorem 8 *Our algorithm is a* $(1+w)$ *approximation for the LMOS-1 problem and a* $(1+w)H(\hat{M})$ *approximation for the LMOS-2 problem.*

Proof: Let γ denote the approximation ratio. Using Eq. 6.39 and from Lemma 4, for LMOS-1 problem, we have

$$\gamma < \frac{\beta}{(N_t-1)/\lambda} \leq \frac{\epsilon \log_{1+\epsilon} 1/\delta}{(1-\epsilon)\ln(\frac{1-\epsilon}{N\delta})} = \frac{\epsilon}{(1-\epsilon)\ln(1+\epsilon)} \frac{\ln(1/\delta)}{\ln(\frac{1-\epsilon}{N\delta})} \tag{6.40}$$

As $\delta = (\frac{N}{1-\epsilon})^{-1/\epsilon}$,

$$\gamma \leq \frac{\epsilon}{(1-\epsilon)^{-2}\ln(1+\epsilon)} \leq \frac{\epsilon}{(1-\epsilon)^{-2}(\epsilon - \epsilon^2/2)} \tag{6.41}$$

$$\leq (1-\epsilon)^{-3} = (1+w) \tag{6.42}$$

Similarly for LMOS-2 problem, we have

$$\gamma < \frac{\beta}{(N_t-1)/\lambda} \leq H(\hat{M})(1-\epsilon H(\hat{M}))^{-1}(1-\epsilon)^{-2} \tag{6.43}$$

$$= H(\hat{M})(1+w) \tag{6.44}$$

Hence proved. ∎

6.1.5 Complexity Analysis

From lemma 4,

$$1 \leq \gamma < \frac{\beta}{(N_t-1)/\lambda} \quad \Rightarrow \quad N_t < 1 + \beta\lambda \tag{6.45}$$

Therefore, the total number of iterations N_t until the approximation algorithm terminates is strictly less than $1+\beta\lambda$. For LMOS-1 problem, after we scale the problem by $T_{LP}/2$ we have $1 \leq \beta \leq 2$. Therefore, the approximation algorithm will terminate in $2\lceil\lambda\rceil$ iterations. For LMOS-2 problem, after we scale the problem by $T_{LP}/\mathcal{M}$ we have $1 \leq \beta \leq \mathcal{M}$. If the algorithm doesn't terminate after $2\lceil\lambda\rceil$, we know $\beta \geq 2$ and double the duration of iterations τ_p. The approximation ratio will not be affected as this is equivalent to re-scalling the problem. As we repeat this procedure until the algorithm terminates, the approximation algorithm for LMOS-2 problem will terminate in $2\lceil\lambda\rceil \log_2 \mathcal{M}$ iterations ($\beta \leq \mathcal{M}$ after scaling).

We note that in each phase of an iteration, except for the last phase, there exists at least one node s_i that consumes energy $E_0(s_i)/\lambda$, whose c_i is increased by a factor $1+\epsilon$. Since for any node s_i, the initial value of c_i is $\delta/E_0(s_i)$ and the final value is less than $1/E_0(s_i)$ (for $D(\vec{C}) < 1$), the number of phases exceeds the number of iterations by at most $N \log_{1+\epsilon} \frac{1}{\delta} = N\lambda$ (otherwise there exists at least one sensor s_i whose c_i exceeds $1/E_0(s_i)$). For LMOS-1 problem, in each phase we build a shortest path tree and apply the FindOSLink routine to select the set of obervation links, which requires $O(N^2)$ and $O(MN(M+N))$ time, respectively. Therefore, the time complexity of approximaton algorithm for LMOS-1 problem is $N\lceil\frac{1}{\epsilon}\log_{1+\epsilon}(\frac{N}{1-\epsilon})\rceil O(NM(M+N))$. For LMOS-2 problem, in each phase we build a shortest path tree and greedily select the source nodes until all the targets are covered, which requires $O(N^2)$ and $O(N\min(M,N))$ time, respectively. Therefore, the time complexity of approximation algorithm for LMOS-2 problem is $(2\log_2(\mathcal{M}) + N)\lceil\frac{1}{\epsilon}\log_{1+\epsilon}(\frac{N}{1-H(\hat{M})\epsilon})\rceil O(N^2 + N\min(M,N))$.

6.2 Communication Weighted Observation Scheduling algorithm

6.2.1 Motivation

The optimal solution of LMOS-1 problem developed in section 5.2 is an off-line centralized algorithm. Any initial input variations due to sensor nodes failures, new nodes deployment or sensor movements, will make the computed schedule unusable and force the algorithm to be re-executed. As solving the LP problem is computationally complex and sensors could be unreliable low price devices, the implementation cost of the optimal solution in the unreliable sensor network may be high. The approximation algorithm developed in section 6.1 provides useful theoretical insights into the LMOS problem and is more flexible for input variations as it generates observation assignments one by one based on the current network state. On the other hand, the number of observation assignments generated could be large, because, to achieve satisfactory results, we need to set ϵ to be small, which results in a small δ and large λ. As generating a new observation assignment will incur protocol cost, e.g. exchanging node state among neighbors and broadcasting the operational duration of the observation assignment, the protocol cost of the approximation algorithm is likely to be high. Therefore, it becomes necessary to develop a flexible low-cost heuristic protocol for the LMOS problem for both scenarios, which can be implemented in the real applications.

6.2.2 Algorithm Description

The heuristic algorithm generates the observation assignments one by one based on the current network state. It uses a greedy method to construct the observation assignments to cover the targets and it couples the communication cost and sources selection. Hence it is called Communication Weighted Observation Scheduling (CWOS). The inputs of the algorithm include graph $G = \{V, \mathbb{E}\}$ and initial energy $E_0(s_i)$ of each sensor s_i. The output of the algorithm is a sequence of observation assignments $\phi(1), \phi(2), \cdots, \phi(X)$ and their operation durations $\tau^{\phi}(1), \tau^{\phi}(2), \cdots, \tau^{\phi}(X)$.

The pseudo-code of the algorithm is shown in Table 6.1. The algorithm repeatedly constructs feasible observation assignments and stops until no new observation assignment can be built (i.e., the network lifetime is reached). Each observation assignment operates for a fixed time duration τ_p, unless any sensor selected by the observation assignment dies due to the lack of energy. Let E_i^r denote the residual energy of s_i and e_i^{ϕ} denote the energy consumption rate of node s_i in a given the observation assignment ϕ. Thus, the operation duration of a feasible observation assignment ϕ is given by $\tau^{\phi} = \min_(\tau_p, \min_{s_i \in \phi}(E_i^r / e_i^{\phi}))$.

In each iteration, there are two steps to build the observation assignment. In the first step, an energy-aware communication tree is constructed connecting all the live sensor nodes to the sink. In the second step, the set of observation links or source sensors which cover the targets are selected, and the paths from the sources to the sink construct the observation assignment.

In the first step, a weight w_{ij} is assigned to each link between live sensors s_i and

s_j which reflects both the communication energy consumption on the link and the residual energy level of the sender. Here we use $w_{ij} = e_{ij}^t \times E_0(s_i)/E_i^r$. A minimum weight communication tree (MWCT) $\mathcal{T}(x)$ is then constructed connecting all sensors such that the sum of the link weights from each node to the sink is minimized. For each live sensor s_i on $\mathcal{T}(x)$, it remembers the sum of link weights from itself to the sink through $\mathcal{T}(x)$, which is denoted by W_i.

In the second step, a modified greedy set cover algorithm is developed to select the set of observation links or sources that covers each target at least K times. The algorithm greedily selects the sources among the live sensors which have not been selected and has at least one target not K-covered in its sensing area. For different observation scenarios, the procedure is a little different:

- For LMOS-1 problem, the sensor with the minimum path weight to the sink (W_i) is selected as the source. The new source randomly chooses target which is not yet K covered to observe until L targets or all the targets not yet K covered in its sensing range are observed by it. Then the next new source is selected.
- For LMOS-2 problem, for each sensor s_i, a cost function is defined as $C_i = W_i \sum_{p_m \in \mathcal{P}_i} r_m / |\mathcal{P}_i^*|$, where $\mathcal{P}_i$ is the set of targets in the sensing range of s_i and $\mathcal{P}_i^*$ is the set of targets without K covered in the sensing area of s_i. The sensor with the minimum cost function is selected as the new source node and the cost function of other sensors are updated accordingly.

The procedure of selection ends until no more source nodes can be selected. In practice, this algorithm can be implemented in a distributed way. For each node s_i, we define set $S_n(i)$ which contains non-source sensors that cover at least one target without K-covered which is also in the sensing area of node s_i. If s_i has the minimum value of weight/cost in $S_n(i)$, s_i must be selected before other sensors in $S_n(i)$. Therefore, let each sensor s_i broadcasts its weight/cost value to all the nodes in $S_n(i)$, if a sensor s_i finds its weight/cost value to be the minimum one in $S_n(i)$, it claims itself as the source node, selects targets to observe (for LMOS-1 problem) and broadcasts message to all other nodes in $S_n(i)$ to update their information. The procedure continues until s_i is selected as the source or all the targets in the sensing area of s_i are K covered.

Finally, each source node sends a message through $\mathcal{T}(x)$ to the sink notifying the bypassing relay nodes. Each active node determines the amount of data to be sent and estimates the amount of energy consumption in the next operation duration. If any sensor forecasts that it will consume more energy than its residual energy, it will calculate the operation duration and broadcast it to all the other sensors. A sensor will wait for a period of time (sufficient to complete the building of observation assignment) after the source selection procedure and determines the length of the next operation duration. Then it goes into sleep state until the end of the operation duration.

For LMOS-1 problem, let T_{LP} denote the optimal lifetime archived by solving LP problem in section 5.2.1; for LMOS-2 problem, let T_{LP} denote the lifetime upper bound in claim 1. Each observation assignment operates for a duration τ_p otherwise at least one sensor will die. Therefore, the number of observation assignments to be built in the heuristic algorithm is upper bounded by $N + T_{LP}/\tau_p$. In each iteration, the complexity of constructing MWCT is $O(N|\mathbb{E}|)$ using Bellman-Ford algorithm, the complexity of source selection procedure is $O(KMN)$, and thus the complexity of CWOS algorithm is $O((KMN + N|\mathbb{E}|)(N + T_{LP}/\tau_p))$.

Note that the modified greedy set cover algorithm (step 2) for LMOS-1 problem does not necessarily generate a feasible set of observation links when there still exists feasible observation assignments. To study whether the performance of CWOS algorithm can be further improved by using other algorithms to select the set of observation links, we develop another more complex algorithm to select the set of observation links for LMOS-1 problem. Consider the graph constructed by the observation links together with the sensors and targets connected with the observation links. Set the link capacity and link cost of each observation link (p_m, s_i) to be 1 and $r_m W_i$. By adding a source node s connecting all the targets with link capacity K and link cost 0, and a destination node t connecting all the sensors with link capacity L and link cost 0, the problem of finding a feasible set of observation links can be converted to a minimum cost flow problem on the graph. A modified Edmonds-Karp (E-K) algorithm (pp. 660 in [63]) can be developed to solve the minimum cost flow problem and hence find a feasible set of observation links with the minimum total path weight. The E-K algorithm has been shown to find a feasible set of observation links if exits. It repeatedly finds the minimum cost path from s to t on the residual network and admitting flow on the path. Let us call the modified CWOS algorithm which selects set of observation links using the modified E-K algorithm as CWOS-EK.

We will compare the performance of the two algorithms and show that CWOS can achieve almost the same performance as CWOS-EK in section 6.3.

6.3 Performance Study

In this section we present the numerical results and evaluate the performance of the approximation algorithm and heuristic algorithm we have proposed. CPLEX is used to solve the LP problem formulated for LMOS-1. The performance of the heuristic algorithm is evaluated in terms of network lifetime T and the number of observation assignments.

We consider the stationary networks with sensor nodes and targets uniformly located in a square of $100m \times 100m$ area. The sink node is placed in the middle of the area. The communication range of each node $R_c = 40m$. The value of various parameters are chosen to be $e_t = 50nJ/bit$, $b = 100pJ/bit/m^4$, $\alpha = 4$, $e_r = 150nJ/bit$ and $e_s = 150nJ/bit$ [69]. We assume that each target produces data at the same rate $10kbps$, and each sensor has the same initial energy $20J$. Each value plotted on the figure or shown in the table is the average result of 100 randomly generated topologies.

6.3.1 LMOS-1

In Fig. 6.1 we compare the network lifetime achieved by CWOS algorithm with the optimal lifetime when the network density increases for different values of K and L. The sensing range $R_s = 40m$. The number of targets in the network is fixed at 20.

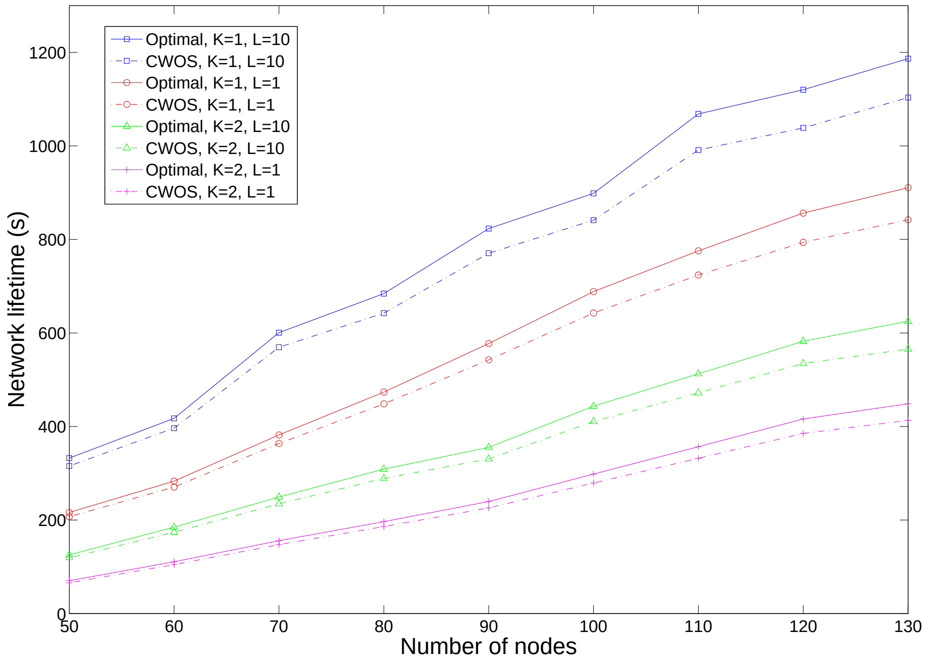

Figure 6.1: The network lifetime of optimal solution and CWOS algorithm vs. number of nodes

The number of nodes is increased from 50 to 130, thus the density is varied. As the network lifetime may vary greatly for different topology, the value of τ_p is set to be $L_{LP}/2M$, where L_{LP} denotes the network lifetime in the optimal solution. It can be observed that the network lifetime increases nearly linearly as the number of nodes increases. The CWOS algorithm can achieve at least 90% of the optimal network lifetime in all the cases.

In Table 6.2 we compare the network lifetime achieved by CWOS algorithm with CWOS-EK algorithm for different values of N and L. The sensing range R_s is set to be $40m$, the number of targets is set as 20 and K is set as 1. It can be observed that CWOS algorithm achieves almost the same lifetime as CWOS-EK.

6.3.2 LMOS-2

As the connectivity issue is not considered in most existing works on discrete target problem, we compare the performance of our algorithms with the *greedy_MSC* algorithm proposed in [16] with suitable modification to account for the connectivity. This modification is done by using an energy-aware communication tree which is built in a similar way as our CWOS algorithm to transmit the sensed data to the sink node. We refer the modified algorithm as GrMSC_EW. The *greedy_MSC* algorithm greedily selects a "critical" target and then selects the sensor with the greatest contribution to the "critical" target until all the targets are covered. The "critical" target is chosen as the target in the sensing area of the least number of sensors, and the contribution function is chosen as the number of uncovered targets in the sensing area of a sensor.

In Table 6.3, the network lifetime (lifetime) achieved and the number of observation assignments (Num. Ass) generated by CWOS algorithm, the approximation algorithm and GrMSC_EW algorithm are compared when the network parameters K, N and M are taken different values. The sensing range $R_s = 20m$. For CWOS and GrMSC_EW algorithm, the value of τ_p is set to be $L_{LP}/2M$, which implies that the number of observation assignments are bounded by $2M + N$. For approximation algorithm, the value of ϵ is set to be 0.1. It can be observed that in all the cases CWOS algorithm can achieve similar (within 5% lower) network lifetime as the approximation algorithm while generating much less observation assignments. The CWOS algorithm can achieve significantly (as 24%) higher network lifetime than GrMSC_EW algorithm. As expected, for all the algorithms the network lifetime will increase as the number of nodes N increases, the number of targets M decreases or the coverage requirement K decreases. The number of observation assignments generated by the CWOS and GrMSC_EW algorithms increases as the number of nodes or the number of targets increases. This is due to the increased upper bound of number of observation assignments $2M + N$.

The construction of new observation assignments incurs protocol cost. To construct an observation assignment, any algorithm (CWOS, approximation or GrMSC_EW) needs to construct a communication tree and then select the source nodes to cover all the targets. Therefore, the number of observation assignments generated by the algorithms could be a good indicator for the potential protocol cost. As we have discussed above, CWOS algorithm can achieve similar lifetime with a much lower protocol cost compared with the approximation algorithm. We note that the construction of new observation assignments in CWOS and GrMSC_EW algorithms is time-based and related to the chosen value of τ_p. If both the CWOS and GrMSC_EW algorithms choose the same value of τ_p in the simulation, as CWOS algorithm can achieve higher network lifetime, CWOS will generate more observation assignments. We set $K = 1$, $N = 100$, $M = 20$ and $R_s = 20m$. For $\tau_p = L_{LP}/2M$, the mean number of observation assignments generated by CWOS and GrMSC_EW algorithms is 55.6 and 42.7, respectively. The lifetime generated by the two algorithms is 408.23 and 331.09, respectively. However, when τ_p is increased to L_{LP}/M, the network lifetime of CWOS algorithm slightly decreases to 405.42 which is still much higher than the lifetime of GrMSC_EW algorithm when $\tau_p = L_{LP}/2M$, while the number of observation assignments of CWOS algorithm drastically decreases to 43.9 which is similar to that of GrMSC_EW algorithm when $\tau_p = L_{LP}/2M$. Therefore, by suitably selecting the value of τ_p, CWOS algorithm can incur a protocol cost close to that of GrMSC_EW while achieving significantly better performance in terms of lifetime.

6.4 Summary

In this chapter we developed approximation algorithms and heuristic algorithms for the LMOS problem. We demenstrated the performance of algorithms through extensive simulations.

Table 6.1: Pseudo-codes for the heuristic algorithm

Input:	$G = \{V, \mathbb{E}\}$
Output:	$\{\phi(1), \phi(2), \cdots, \phi(X)\}$, $\{\tau^{\phi}(1), \tau^{\phi}(2), \cdots, \tau^{\phi}(X)\}$.
(01)	$T = 0$, $x = 1$;
(02)	**while** (1)
	step 1:
(03)	Each node s_i assign weight $w_{ij} = e^t_{ij} \times E_0(s_i)/E^r_i$ to any link $(s_i, s_j) \in \mathbb{E}$ originating from s_i;
(04)	Build the minimum weight tree $\mathcal{T}(x)$ rooted at sink $\mathcal{R}$ with link weight w_{ij};
	step 2:
(05)	Call modified greedy set-cover algorithm to find set of observation links or sources; If return FALSE, **break**;
(06)	Appending the route from each source to the sink construct the observation assignment $\phi(x)$;
(07)	Each node s_i estimates its energy consumption rate e^{ϕ}_i;
(08)	$\tau^{\phi}(x) = min(\tau_p, min_i(E^r_i/e^{\phi}_i))$;
(09)	$x = x + 1$, $T = T + \tau^{\phi}(x)$; Update the network, delete dead or isolated nodes;
(10)	**Endwhile**
	Modified Greedy set-cover algorithm
(01)	$\mathcal{P}^* = \mathcal{P}$; let $\mathcal{S}^*$ denote the set of unselected sensors cover at least one target in $\mathcal{P}^*$;
(02)	**While** $\mathcal{S}^* \neq \phi$ and $\mathcal{P}^* \neq \phi$
(03)	If LMOS-1,
(04)	Select $s_i \in \mathcal{S}^*$ with minimum W_i as the source;
(05)	If $\|\mathcal{P}^* \cap \mathcal{P}_i\| > L$, randomly select L targets from $\mathcal{P}^* \cap \mathcal{P}_i$ to be observed by s_i;
(06)	Else all the targets in $\mathcal{P}^* \cap \mathcal{P}_i$ are observed by s_i;
(07)	If LMOS-2,
(08)	Select $s_i \in \mathcal{S}^*$ with minimum C_i as the source node;
(09)	Delete targets covered at least K times from $\mathcal{P}^*$;
(10)	Update $\mathcal{S}^*$, update c_i for each sensor $s_i \in \mathcal{S}^*$;
(11)	If the sources cannot cover all the targets K times, **return** FALSE;
(12)	Else **return** the set of obervation links or sources;

Table 6.2: Comparison of CWOS with CWOS-EK algorithm for LMOS-1 problem

(N, L)	lifetime(s)		
	Optimal	CWOS-EK	CWOS
$(60, 1)$	253.717	242.352	236.749
$(60, 10)$	387.576	369.199	364.739
$(100, 1)$	675.518	629.674	627.944
$(100, 10)$	955.277	887.049	885.589

Table 6.3: Comparison of CWOS with approximation and GrMSC_EW algorithm for LMOS-2 problem

(K, N, M)	CWOS		Approximation		GrMSC_EW	
	Lifetime	Num. Ass	Lifetime(s)	Num. Ass	Lifetime(s)	Num. Ass
$(1, 60, 15)$	152.70	26.40	156.30	2368.8	129.25	21.90
$(1, 100, 15)$	492.47	42	516.17	5069.2	426.33	36.2
$(1, 130, 15)$	898.9	71.6	938.58	8138.4	723.74	58.5
$(1, 100, 30)$	238.0	64.0	239.36	4238.3	196.0	52.4
$(2, 100, 15)$	235.51	42.3	247.06	4983.4	194.6	33.0

Chapter 7

A general framework of approximation algorithm for the Connected Target Coverage problem

In chapter 4 and chapter 6 we developed approximation algorithms for the MCT and LMOS problem, respectively. These algorithms use a prime-dual approach to approximate the problems. The approach is to repeatedly select a set of active sensors that can satisfy both the coverage and connectivity requirements. To select a suitable set of active sensors, a weight is assigned for each node and is updated according to the energy consumption of the node. We try to select the set of active sensors such that the total weighted energy consumption in the network is minimized while the target coverage and connectivity requirements are satisfied. In this chapter, we present a general framework of approximation algorithm for the CTC problem. This algorithm is applicable to various possible instances of the CTC problem described by different application scenarios, say for example, with different observation scenarios and communication schemes. We show that the lifetime maximization problem for connected target coverage can be approximated by solving the problem of selecting a set of active sensors that minimizes the weighted communication cost while maintaining connectivity and coverage.

7.1 Possible instances of the CTC problem

The CTC problem considers the problem of scheduling sensor activities while maintaining connected target coverage to maximize the network lifetime. However, different applications in WSNs may construct different instances of CTC problems as they may have different application scenarios including

- different target coverage and connectivity requirements, for example, an application may require that different targets are covered by different number of sensors, sensors observing the same target are separated by at least some pre-defined distance, or source nodes connect to the sink node through node-disjoint paths, etc.;
- different data generation and energy consumption models, for example, the amount of data generated by a source node may be defined to be related to the positions of targets observed by it (when some kind of data processing rule is applied), or the transmission power is the same for each node (when power control is not applied), etc.;
- or different observation and communication scenarios, for example, a sensor has ability to select the targets to observe but all the targets observed by the same source sensor should be within a predefined distance (e.g. smaller than the sensing range), or data aggregation is used in the network, etc.;

Apparently, the algorithm designed for one instance of the CTC problem may not be applicable for other instances of the CTC problem. Therefore, it becomes important to develop a general framework of approximation algorithm for CTC problems, which can cover several possible instances of it.

7.2 Preliminaries

Without loss of generality, we assume that the network lifetime is slotted into a series of time slots. Within each time slot the state of a sensor does not change. However, for different application scenarios, the description of the network state in a time slot may be different. For example, for the MCT problem, a cover tree provides the description of the network state in a time slot, which describes the set of source nodes and relay nodes as well as the path from each source node to the sink; whereas for the LMOS problem, an observation assignment provides the description of the network state in a time slot, which additionally describes the sensor target observation pairs in the case of the first observation scenario. In an attempt to generalize the modeling of the application scenarios, two descriptions are used together to describe the network state, which are called as the "observation" description and "energy" description. The observation description of the network state in a time slot is application-related and thus may be different for different application scenarios, such as the cover tree for the MCT problem and observation assignment for the LMOS problem. The energy description of the network state in a time slot is application-independent, which is defined as the set of energy consumption rates of sensors in the time slot. Given the specific application scenario, we assume an *energy consumption mapping* is known for mapping a specific observation description to the energy description, i.e. given the specific application scenario and the network description in a time slot, we can calculate the energy consumption rate of each sensor in the time slot. The specific mapping from the observation description to the energy description may vary for different application scenarios.

For a given application scenario, we call the observation description in a time slot as an *assignment* and the duration of the time slot as the operation duration of the assignment. If an assignment can satisfy the coverage and connectivity requirements of the CTC problem, we call it a feasible assignment. Let us call the energy description in a time slot as the *energy mapping* of the assignment in the time slot. Although different assignments may have the same energy mapping, given an optimal solution of the CTC problem, multiple time slots with different feasible assignments but having the same energy mapping can be combined without affecting the network lifetime.

We re-define the Connected Target Coverage for Lifetime Maximization (CT-CLM) problem as below:

Definition 2 *Given a network topology and an application scenario, find a series of feasible assignments with operation durations, such that the energy consumption of each node is less than its initial energy and the sum of the operation durations is maximized.*

We also define the Connected Target Coverage for Minimizing weighted Energy consumption problem (CTCME) problem as below:

Definition 3 *Given a network topology, an application scenario and a set of weights* w_i *for each sensor* s_i*, find a feasible assignment with the energy mapping* $\{e_1, e_2, \cdots, e_N\}$*, such that the total weighted energy consumption rate* $\sum_{1 \leq i \leq N} e_i w_i$ *is minimized.*

Theorem 9 *If there exists an* γ *approximation algorithm for the CTCME problem, then there also exists an* $(1+w)\gamma$ *approximation algorithm for the CTCLM problem, where* w *is an arbitrarily small number.*

We prove the theorem by constructing an $(1+w)\gamma$ approximation algorithm for the CTCLM problem based on the γ approximation algorithm for CTCME problem. Given an instance of the CTCLM problem, let us enumerate the energy mapping of all the feasible assignments $U = \{U_1, U_2, \cdots, U_Q\}$. In each energy mapping U_q the energy consumption rate of sensor s_i is e_i^q. Let τ_q denote the duration in which the energy mapping of the assignment is U_q. The CTCLM problem can be formulated as below:

$$\text{Maximize:} \quad \sum_q \tau_q \tag{7.1}$$

$$\sum_q \tau_q e_i^q \leq E_0(s_i) \qquad \forall s_i \tag{7.2}$$

The Dual problem of the above Linear programming problem is:

$$\text{Minimize:} \quad \sum_i c_i E_0(s_i) \tag{7.3}$$

$$\sum_i c_i e_i^q \geq 1 \qquad \forall U_q \in U \tag{7.4}$$

where $c_1, c_2, \cdots, c_N$ are the variables in the dual problem.

Let $\vec{C}$ be the vector such that its i^{th} element is c_i. We define

$$D(\vec{C}) = \sum_i c_i E_0(s_i) \tag{7.5}$$

$$\alpha(\vec{C}) = \min_{U_q \in U} \sum_i e_i^q c_i \tag{7.6}$$

The dual problem is then equivalent to assigning values to $\vec{C}$ such that $\beta = D(\vec{C})/\alpha(\vec{C})$ is minimized subject to the constraint that $\alpha(\vec{C}) \geq 1$.

7.3 Pseudo code of the algorithm

The approximation algorithm is given below. The explanation is similar to that presented in chapter 4 and chapter 6.

1. Initialization

   (a) Properly scale the problem so that $\beta \geq 1$;

   (b) $t = 0$, $T = 0$; Set $\delta = (\frac{N}{1-\gamma\epsilon})^{-1/\epsilon}$; For each node s_i set $c_i = \delta/E_0(s_i)$; Let $\lambda = \log_{1+\epsilon}^{1/\delta}$; Set $\tau_p = 1/\lambda$;

2. Set $\tau^t = 0$, loop until $\tau^t = \tau_p$;

   (a) $k = k + 1$; Find a feasible assignment such that $\sum_i c_i(t,k)e_i \leq \gamma \min_{U_q \in U} \sum_i c_i(t,k)e_i^q$ with the γ approximation algorithm for the CTCME problem;

   (b) $\tau^t = \tau^t + \tau_k^t$; Let e_i^{tk} denote the energy consumption of node s_i in duration τ_k^t, $c_i(t,k) = c_i(t,k-1) \times (1 + \epsilon \cdot \frac{\lambda e_i^{tk}}{E_0(s_i)})$.

3. $t = t + 1$; $c_i(t,0) = c_i(t-1,k)$; if $D(\vec{C}) < 1$, $T = T + \tau_p$.

4. repeat step 2 and step 3 until $D(\vec{C}) \geq 1$; double τ_p every 2λ iterations.

7.4 Analysis

The analysis follows the analysis presented in chapter 4 and chapter 6. As

$$c_i(t,k) = c_i(t,k-1) \times (1 + \epsilon \cdot \frac{\lambda e_i^{tk}}{E_0(s_i)}) \tag{7.7}$$

we have

$$D(\vec{C}(t,k)) = \sum_i c_i(t,k) E_0(s_i) \tag{7.8}$$

$$= \sum_i c_i(t,k-1) E_0(s_i)(1 + \epsilon \cdot \frac{\lambda e_i^{tk}}{E_0(s_i)}) \tag{7.9}$$

$$= D(\vec{C}(t,k-1) + \epsilon \cdot \lambda \sum_i e_i^{tk} c_i(t,k-1) \tag{7.10}$$

As c_i is monotonically non-decreasing, we have $\alpha(\vec{C}(t+1,0)) \geq \alpha(\vec{C}(t,k)$ for any $1 \leq k \leq K_t$, where K_t denote the number of phases in an iteration. Further, as $\sum_i c_i(t,k) e_i^{tk} \leq \gamma \min_{U_q \in U} \sum_i c_i(t,k) e_i^q \tau_k^t \leq \gamma \alpha(\vec{C}(t,k)) \tau_k^t$ satisfies for any t and k, we have

$$D(\vec{C}(t+1,0) = D(\vec{C}(t,0)) + \epsilon \cdot \lambda \sum_i \sum_{0 \leq k \leq K_t - 1} e_i^{tk} c_i(t,k) \tag{7.11}$$

$$\leq D(\vec{C}(t,0)) + \epsilon \cdot \lambda\gamma \sum_k \tau_k^t \alpha(\vec{C}(t,k)) \tag{7.12}$$

$$\leq D(\vec{C}(t,0)) + \epsilon \cdot \lambda\gamma\tau_p \alpha(\vec{C}(t+1,0)) \tag{7.13}$$

$$\leq D(\vec{C}(t,0)) + \epsilon \cdot \gamma \alpha(\vec{C}(t+1,0)) \tag{7.14}$$

The following analysis is similar to the analysis of approximation algorithm in Chapter 4 by replacing $H(\hat{M})$ by γ. Let N_t denote the number of iterations before the algorithm ends. We have

$$\frac{\beta}{N_t - 1} \leq \frac{\epsilon\gamma}{(1-\epsilon\gamma)\ln(\frac{1-\epsilon\gamma}{N\delta})} \tag{7.15}$$

Lemma 5 *The solution of our approximation algorithm is a feasible solution for the CTCLM problem, i.e., when the algorithm ends, the energy consumption of each node is less than its initial energy.*

Proof: similar to the proof of lemma 2 presented in Chapter 4. ∎

Theorem 10 *Our algorithm is a* $\gamma(1 + w)$ *approximation for the CTCLM problem.*

Proof: similar to the proof of Theorem 3 presented in Chapter 4. ∎

The complexity of the algorithm is analyzed as below. If all the sensors in the network are active, the duration until the first sensor in the network consumes all its energy gives a lower bound on the network lifetime. Let us denote it as T_l. Normalizing the initial energy of each node by T_l guarantees that $\beta \geq 1$. From Eq. 7.15, the total number of iterations until the approximation algorithm terminates is strictly less than $1 + \beta\lambda$. In fact, if our algorithm doesn't terminate after $2\lceil\lambda\rceil$, we know $\beta \geq 2$ and can double the duration of iterations τ_p. Note that this is equivalent to re-scaling the problem. β will be half of its previous value but still larger than 1, and thus the approximation ratio still holds. As we repeat this procedure until the algorithm terminates, the approximation algorithm will terminate in $2\log_2\beta\lceil\lambda\rceil$ iterations.

We note that in each phase of an iteration, except for the last phase, there exists at least one node s_i that consumes energy $E_0(s_i)/\lambda$, whose c_i is increased by a factor $1+\epsilon$.

Since for any node s_i, the initial value of c_i is $\delta/E_0(s_i)$ and the final value is less than $1/E_0(s_i)$ (for $D(\vec{C}) < 1$), the number of phases exceeds the number of iterations by at most $N \log_{1+\epsilon} \frac{1}{\delta} = N\lambda$ (otherwise there exists at least one sensor s_i whose c_i exceeds $1/E_0(s_i)$). In each phase a shortest path tree is built and the approximation algorithm for CTCME problem is executed. The building of shortest path tree requires time $O(N^2)$. Let Θ_{ME} denote the time complexity of the approximation algorithm for CTCME problem. Therefore, the time complexity of our algorithm is $(2\log_2(\beta) + N)\lceil \frac{1}{\epsilon} \log_{1+\epsilon}(\frac{N}{1-\gamma\epsilon}) \rceil (N^2 + \Theta_{ME})$.

7.5 Summary

In this chapter we developed a general framework of approximation algorithm for the CTC problem. We demonstrated that the network lifetime maximization problem for connected target coverage can be approximated by solving the problem of selecting a set of active sensors that minimize the weighted communication cost while maintaining connectivity and target coverage.

Chapter 8

Conclusions and Future Work

In this thesis, we addressed the problem of scheduling sensor activities while maintaining target coverage and network connectivity. First we introduced the Connected Target coverage (CTC) problem and modeled it as a Maximum Cover Tree (MCT) problem. We proved that the MCT problem is NP-Complete and develop a linear programming formulation to derive an upper bound and a lower bound on the network lifetime for the MCT problem.

We developed an $H(\hat{M})(1+w)$ approximation algorithm to solve the MCT problem based on the upper bound and lower bound, where w is an arbitrarily small number, $H(\hat{M}) = \sum_{1 \leq i \leq \hat{M}} 1/i$ and $\hat{M}$ denotes the maximum number of targets in the sensing area of any sensor. As a practical implementation we developed a faster heuristic algorithm called Communication Weighted Greedy Cover (CWGC). We further proved that $(1 - O(1))ln(M)$ is a threshold below which the MCT problem cannot be approximated efficiently, unless NP has slightly super-polynomial time algorithms, i.e. $NP \subset TIME(n^{O(loglogn)})$ [90], where M is the number of targets. We demonstrated the effectiveness of the proposed approximation algorithm and heuristic algorithm by carrying out extensive simulations and comparing their results with the upper bound on the lifetime and other basic algorithms which consider the coverage and connectivity problems independently. Simulation results show that our approximation algorithm and CWGC algorithm perform much better than others in terms of the network lifetime and the performance improvement can be up to 45% than the best-known basic algorithm. The lifetime obtained by our algorithms is close to the upper bound. Compared with the approximation algorithm, the CWGC algorithm can achieve a similar performance in terms of network lifetime with a lower protocol cost.

Next we considered the CTC problem with K coverage requirement wherein each target is observed by at least K sensors. We formulated the problem as the Lifetime Maximization Observation Schedule (LMOS) problem and studied the problem with two observation scenarios depending on whether a sensor can distinguish the targets in its sensing area, or not. For the first scenario, we developed a polynomial-time algorithm which can achieve optimal solution based on Linear Programming and Integer Theorem. For the second scenario, we showed that the LMOS problem is NP-complete. We developed approximation algorithms for both scenarios. Based on the approximation algorithms, we developed a low-cost heuristic algorithm which can be implemented in a distributed fashion for both scenarios. We demonstrated the effectiveness of the heuristic algorithm through extensive simulations.

Finally, we presented a general framework of approximation algorithm for the CTC problem, which is applicable to various possible instances of the CTC problem described by different application scenarios, say for example, with different observation scenarios and communication schemes. We show that the lifetime maximization problem for connected target coverage can be approximated by solving the problem of selecting a set of active sensors that minimizes the weighted communication cost while maintaining connectivity and coverage.

In this thesis the CTC problem was solved as an optimization problem which jointly considers both the target coverage and connectivity. The CTC problem may also be solved by breaking it into two stages, wherein the target coverage and connectivity problems are independently solved. Intuitively joint optimization achieves better performance compared with breaking the problem into stages. This is substantiated by our performance study in Chapter 4. The three basic algorithms designed for comparison (Random, MSC_SPT and MSC_EAWARE) solve the MCT problem by separately solving the target coverage and connectivity problems. Our algorithms which solve the coverage and connectivity problems by joint optimization outperformed them.

We now present some possible directions for future investigation. In this thesis we developed approximation algorithms for the CTC problem in wireless sensor networks. However, the protocol cost of the approximation algorithms may be high. Developing low-cost faster approximation scheme for the CTC problem is an important problem to be studied. Another interesting problem is to study and develop efficient algorithms for the case where sensors or targets are mobile. Further study could also consider other performance metrics such as reliability and quality of observation.

List of Publications

1. Zhao Qun and Mohan Gurusamy, "Lifetime Maximization for Connected Target Coverage in Wireless Sensor Networks", to appear in, *IEEE/ACM Transactions on Networking.*

2. Zhao Qun and Mohan Gurusamy, "Connected K-target-coverage in Wireless Sensor Networks with different sensing scenarios", to appear in, *Computer Networks journal.*

3. Zhao Qun and Mohan Gurusamy, "Optimal Observation Scheduling for connected target coverage problem in Wireless Sensor Networks", in *Proc. of IEEE International Conference on Communications (ICC)*, 2007.

4. Zhao Qun and Mohan Gurusamy, "Maximizing Network Lifetime for Connected Target Coverage in Wireless Sensor Networks", in *Proc. of 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2006.

5. Zhao Qun and Mohan Gurusamy, "Lifetime Maximization using Observation Time Scheduling in Multi-hop Sensor Networks", in *Proc. of 2nd IEEE/CreateNet International Workshop on Broadband Advanced Sensor Networks (BroadNets)*, 2005.

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 39, no. 4, pp. 393–422, 2002.

[2] A. I.F., S. Weilian, S. Y. and C. E., "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 – 114, 2002.

[3] E. D., G. R., H. J. and K. S., "Next century challenges: scalable coordination in sensor networks," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 1999, pp. 263–270.

[4] P. G.J. and K. W.J., "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[5] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," in *Proc. of ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001, pp. 3–5.

[6] M. Alan, P. Joseph, S. Robert, C. David and A. John, "Wireless sensor networks for habitat monitoring," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002, pp. 88–97.

[7] S. E., C. S., I. N., M. R., S. A., W. A. and C. A., "Physical layer driven

protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2001, pp. 272–286.

[8] G. Yong, W. Kui and L. Fulu, "Analysis on the redundancy of wireless sensor networks," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003, pp. 108 – 114.

[9] K. S., L. T. H. and B. J., "On k-coverage in a mostly sleeping sensor network," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2004, pp. 144–158.

[10] W. K., G. Y., L. F. and X. Y., "Lightweight deployment-aware scheduling for wireless sensor networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks*, vol. 10, no. 6, pp. 837–852, 2005.

[11] Y. Yong, Y. Zongkai, H. Zhihai and H. Jianhua, "Energy-efficient scheduling of packet transmissions over wireless networks provisioning in wireless sensor network," *Computer Communications*, vol. 29, no. 2, pp. 162–172, 2006.

[12] R. V., S. C., P. S. and M. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, pp. 40–50, 2002.

[13] C. J. and T. L., "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proc. of 37th Annual Allerton Conference on Communication, Control, and Computing*, 1999.

[14] M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks

via optimal role assignments," in *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002, vol. 3, pp. 1587 – 1596.

[15] G. A. and K. P.R., "Maximizing the functional lifetime of sensor networks," in *Proc. of IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2005, pp. 5– 12.

[16] M. Cardei, M. T. Thai, Y. Li and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005, pp. 1976–1984.

[17] S. Tao, K. Marwan and V. Sarma, "Power balanced coverage-time optimization for clustered wireless sensor networks," in *Proc. of ACM International Symposium on Mobile ad hoc Networking and Computing (MobiHoc)*, 2005, pp. 111–120.

[18] M. Cardei and J. Wu, *Coverage Problems in Wireless Ad Hoc Sensor Networks (Mohammad Ilyas and Imad Mahgoub eds.), Handbook of Sensor Networks*, chapter 19, CRC Press, 2004.

[19] C.-F. Huang and Y.-C. Tseng, "A survey of solutions to the coverage problems in wireless sensor networks," *Journal of Internet Technology*, vol. 6, no. 1, pp. 1–8, 2005.

[20] H. D. L. and L. J., *Handbook of Multisensor Data Fusion*, CRC Press, 2001.

[21] A. A., D. P., B. S. and etc., "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no.

5, pp. 605–634, 2004.

[22] Y. Shuhui, D. Fei, C. Mihaela and W. Jie, "On multiple point coverage in wireless sensor networks," in *Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2005.

[23] P. Berman, G. Calinescu, C. Shah and A. Zelikovsky, "Power efficient monitoring management in sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2004, vol. 4, pp. 2329– 2334.

[24] M. S., K. F., P. M. and S. M., "Coverage problems in wireless ad-hoc sensor networks," in *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001, pp. 1380–1387.

[25] M. S., K. F., Q. G. and P. M., "Exposure in wireless ad hoc sensor networks," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2001, pp. 139–150.

[26] L. Ning and H. J. C., "Topology control in heterogeneous wireless networks: Problems and solutions," in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2004, vol. 1, p. 243.

[27] M. Tommaso, P. Dario and A. I. F., "Optimal local topology knowledge for energy efficient geographical routing in sensor networks," in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2004, vol. 3, pp. 1705 – 1716.

[28] G. Javier and C. A. T., "A case for variable-range transmission power control in wireless multihop networks," in *Proc. of the Annual Joint Conference of the*

IEEE Computer and Communications Societies (INFOCOM), 2004, vol. 2, pp. 1425 – 1436.

[29] Z. Qun and M. Gurusamy, "Topology knowledge range control for lifetime maximization in sensor networks with data aggregation," in *Proc. of ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks, ACM/IEEE MSWiM*, 2005.

[30] L. L. (Erran) and S. Prasun, "Throughput and energy efficiency in topology-controlled multi-hop wireless sensor networks," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003.

[31] C. Xiuzhen, N. Bhagirath, S. Rahul and C. M. Xiaoyan, "Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 248–256, 2003.

[32] M. Rajit and S. Anna, "Power optimal routing in wireless networks," in *Proc. of IEEE International Conference on Communications (ICC)*, 2003, vol. 4, pp. 2979 – 2984.

[33] L. Jilei and L. Baochun, "Distributed topology control in wireless sensor networks with asymmetric links," in *Proc. of IEEE Global Telecommunications Conference (Globecom)*, 2003, vol. 3, pp. 1257 – 1262.

[34] K. Martin, K. Holger, W. Adam, Z. L. Charlie and R. Jan, "Distributed algorithms for transmission power control in wireless sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2003, vol. 1, pp. 558 – 563.

[35] C. Benjie, J. Kyle, B. Hari and M. Robert, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[36] X. Ya, H. John and E. Deborah, "Geography-informed energy conservation for ad hoc routing," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2001, pp. 70–84.

[37] A. Cerpa and D. Estrin, "Ascent: adaptive self-configuring sensor networks topologies," in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[38] S. C., T. V. and S. M.B., "Stem: Topology management for energy efficient sensor networks," in *Proc. of IEEE Aerospace Conference*, 2002, vol. 3, pp. 109–1108.

[39] G. J. van, P. Dragan and B. Alvise, "Adaptive sleep discipline for energy conservation and robustness in dense sensor networks," in *Proc. of IEEE International Conference on Communications (ICC)*, 2004, vol. 6, pp. 3657 – 3662.

[40] R. Volkan and M. T. H., "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, 1999.

[41] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. of IEEE International Conference on Communications (ICC)*, 2001, vol. 2, pp. 472–476.

[42] C. Qing, A. Tarek, H. Tian and S. John, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proc. of ACM International*

Symposium on Information Processing in Sensor Networks (IPSN), 2005.

[43] W. Tzu-Ting and S. Kuo-Feng, "Determining active sensor nodes for complete coverage without location information," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 1, no. 1, pp. 38–46, 2005.

[44] H. Chi-Fu, L. Li-Chu, T. yu Chee and C. Wen-Tsuen, "Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks," in *Proc. of International Symposium on Circuits and Systems (ISCAS)*, 2005.

[45] A. Zoe, G. Ashish and P. Serge, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proc. of ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, 2004, pp. 424–432.

[46] X. Guoliang, L. Chenyang, P. Robert and O. J. A., "Co-grid: An efficient coverage maintenance protocol for distributed sensor networks," in *Proc. of ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, 2004.

[47] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li and D.-Z. Du, "Wireless sensor networks with energy efficient organization," *Journal of Interconnection Networks*, vol. 3, no. 3-4, pp. 213–229, 2002.

[48] D. Tian and N. D. Georganas, "A coverage preserving node scheduling scheme for large wireless sensor networks," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[49] H. Gupta, Z. Zhou, S. R. Das and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," *IEEE/ACM*

Transactions on Networking, vol. 14, no. 1, pp. 55–67, 2006.

[50] Z. Rong, H. Guanghui and L. Xue, "Location-free coverage maintenance in wireless sensor networks," Tech. Rep. Tech. No. UH-CS-05-15, Department of Computer Science, University of Houston, 2005.

[51] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, May 2005.

[52] M. Cardei, J. Wu, M. Lu and M. O. Pervaiz, "Maximum network lifetime in wireless sensor networks with adjustable sensing ranges," in *Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005.

[53] L. Hai, W. Pengjun, Y. Chih-Wei, J. Xiaohua, M. Sam and N. Pissinou, "Maximal lifetime scheduling in sensor surveillance networks," in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005, vol. 4, pp. 2482 – 2491.

[54] L. Hai, W. Pengjun and X. Jia, "Maximal lifetime scheduling for k to 1 sensor-target surveillance networks," *Computer Networks*, vol. 50, no. 15, pp. 2839–2854, 2006.

[55] Z. Zhou, S. Das and H. Gupta, "Connected k-coverage problem in sensor networks," in *Proc. of IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2004, pp. 373 – 378.

[56] C. Jean, G. Antoine and S.-R. David, "Preserving area coverage in wireless

sensor networks by using surface coverage relay dominating sets," in *Proc. of IEEE Symposium on Computers and Communications (ISCC)*, 2005, pp. 347–352.

[57] Z. Zhou, S. Das and H. Gupta, "Fault tolerant connected sensor cover with variable sensing and transmission ranges," in *Proc. of IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2005, pp. 594– 604.

[58] Z. Yi and C. Krishnendu, "A distributed coverage and connectivity centric technique for selecting active nodes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 978–991, 2005.

[59] I. Metin, M.-I. Malik and Y. Bulent, "Power optimal connectivity and coverage in wireless sensor networks," Tech. Rep., Department of Computer Science, Rensselaer Polytechnic Institute, July 2003.

[60] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. of ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2003, pp. 28–39.

[61] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," Tech. Rep., Department of Computer Science, University of Illinois at Urbana-Champaign, 2003.

[62] M. Lu, J. Wu, M. Cardei and M. Li, "Energy-efficient connected coverage of discrete targets in wireless sensor netorks," in *Proc. of International Conference*

on Computer Networks and Mobile Computing (ICCNMC), 2005.

[63] C. T. H., L. C. E., R. R. L. and S. Clifford, *Introduction to Algorithms (Second Edition)*, The MIT Press and McGraw-Hill, 2001.

[64] L. Yuzhen and L. Weifa, "Approximate coverage in wireless sensor networks," in *Proc. of IEEE Conference on Local Computer Networks (LCN)*, 2005, pp. 68–75.

[65] W. Limin and K. S. S., "pcover: Partial coverage for long-lived surveillance sensor networks," Tech. Rep. MSC-CSE-0530, Department of Computer Science and Engineering, Michigan State University, 2005.

[66] Z. Honghai and H. J. C., "On the upper bound of -lifetime for large sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 2, pp. 272–300, 2005.

[67] X. Guoliang, W. Xiaorui, Z. Yuanfang, L. Chenyang, P. Robert and G. Christopher, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 1, pp. 36–72, 2005.

[68] F. Dai and J. Wu, "Distributed dominant pruning in ad hoc networks," in *Proc. of IEEE International Conference on Communications (ICC)*, 2003, pp. 353–357.

[69] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor netwokrs," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[70] K. Kalpakis, K. Dasgupta and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," Tech. Rep., Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 2002.

[71] N. Sadagopan and B. Krishnamachari, "Maximizing data extraction in energy-limited sensor networks," in *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2004, vol. 3, pp. 1717–1727.

[72] K. Hojoong, K. T. Hyun, C. Sunghyun and L. B. Gi, "Lifetime maximization under reliability constraint via cross-layer strategy in wireless sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2005, vol. 3, pp. 1891 – 1896.

[73] H. Y.T., S. Yi, P. Jianping and M. S.F., "Maximizing the lifetime of wireless sensor networks through optimal single-session flow routing," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1255 – 1266, 2006.

[74] S. Rik, Z. Xianjin and G. Jie, "Double rulings for information brokerage in sensor networks," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2006.

[75] B. Chiranjeeb, A. Divyakant, Subhash and Suri, "Power aware routing for sensor databases," in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[76] L. Juan, Z. Feng and P. Dragan, "Information-directed routing in ad hoc sensor

networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 851–861, 2005.

[77] T. Mohammed, T. K. E. and N. Mohammad, "Energy saving dynamic source routing for ad hoc wireless networks," in *Proc. of International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2005, pp. 305–310.

[78] J. Neha, B. Ratnabali, N. Nagesh and A. D. P., "Energy aware routing for spatio-temporal queries in sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2005, vol. 3, pp. 1860 – 1866.

[79] L. Qilian and R. Qingchun, "Energy and mobility aware geographical multipath routing for wireless sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2005, vol. 3, pp. 1867 – 1871.

[80] D. Arjan, P. Vamsi and B. Leonard, "Delay-energy aware routing protocol for sensor and actor networks," in *Proc. of IEEE International Conference on Parallel and Distributed Systems*, 2005, vol. 1, pp. 292 – 298.

[81] Y. T. Hou, Y. Shi and H. D. Sherali, "On lexicographic max-min node lifetime for wireless sensor networks," in *Proc. of IEEE International Conference on Communications (ICC)*, 2004, vol. 7, pp. 3790 – 3796.

[82] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2000.

[83] M. Gerla, T. J. Kwon and G. Pei, "On demand routing in large ad hoc wireless networks with passive clustering," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2000.

[84] Y. Yu, R. Govindan and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," Tech. Rep. UCLA/CSD-TR-01-0023, Department of Computer Science, University of California, Los Angeles, 2001.

[85] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2002, vol. 1, pp. 350 – 355.

[86] M. Zorzi and R. R. Rao, "Energy and latency performance of geographic random forwarding for ad hoc and sensor networks," in *Proc. of IEEE Wireless Communications and Networking (WCNC)*, 2003, vol. 3, pp. 1930 – 1935.

[87] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *Journal of the Association for Computing Machinery*, vol. 37, no. 2, pp. 318–334, 1990.

[88] N. Garg and J. Konemann, "Faster and simpler algorithm for multicommodity flow and other fractional packing problems," in *Proc. of IEEE Symposium on Foundations of Computer Science*, 1998, pp. 300–309.

[89] G. Karakostas, "Faster approximation schemes for fractional multicommodity flow problems," in *Proc. of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, 2002, pp. 166–173.

[90] F. Uriel, “A threshold of ln n for approximating set cover,” *Journal of ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[91] V. Vhvatal, “A greedy heuristic for the set-covering problem,” *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[92] G. Dobson, “Worst-case analysis of greedy heuristics for integer programming with non-negative data,” *Mathematics of Operations Research*, vol. 7, no. 4, pp. 515 – 531, 1982.