Georgia State University
ScholarWorks @ Georgia State University

Computer Science Dissertations

Department of Computer Science

Summer 8-1-2012

Connected Dominating Set Based Topology Control in Wireless Sensor Networks

Jing S. He
Georgia State University

CONNECTED DOMINATING SET BASED TOPOLOGY CONTROL IN WIRELESS SENSOR NETWORKS

by

JING (SELENA) HE

Under the Direction of Dr. Yi Pan and Dr. Yingshu Li

ABSTRACT

Wireless Sensor Networks (WSNs) are now widely used for monitoring and controlling of systems where human intervention is not desirable or possible. Connected Dominating Sets (CDSs) based topology control in WSNs is one kind of hierarchical method to ensure sufficient coverage while reducing redundant connections in a relatively crowded network. Moreover, Minimum-sized Connected Dominating Set (MCDS) has become a well-known approach for constructing a Virtual Backbone (VB) to alleviate the broadcasting storm for efficient routing in WSNs extensively. However, no work considers the load-balance factor of CDSs in WSNs. In this dissertation, we first propose a new concept — the Load-Balanced CDS (LBCDS) and a new problem — the Load-Balanced Allocate Dominatee (LBAD) problem. Consequently, we propose a two-phase method to solve LBCDS and LBAD one by one and a one-phase Genetic Algorithm (GA) to solve the problems simultaneously.

Secondly, since there is no performance ratio analysis in previously mentioned work, three problems are investigated and analyzed later. To be specific, the MinMax Degree Maximal Independent Set (MDMIS) problem, the Load-Balanced Virtual Backbone (LB-VB) problem, and the MinMax Valid-Degree non Backbone node Allocation (MVBA) problem. Approximation algorithms and comprehensive theoretical analysis of the approximation factors are presented in the dissertation.

On the other hand, in the current related literature, networks are deterministic where two nodes are assumed either connected or disconnected. In most real applications, however, there are many intermittently connected wireless links called lossy links, which only provide probabilistic connectivity. For WSNs with lossy links, we propose a Stochastic Network Model (SNM). Under this model, we measure the quality of CDSs using CDS reliability. In this dissertation, we construct an MCDS while its reliability is above a preset application-specified threshold, called Reliable MCDS (RMCDS). We propose a novel Genetic Algorithm (GA) with immigrant schemes called RMCDS-GA to solve the RMCDS problem. Finally, we apply the constructed LBCDS to a practical application under the realistic SNM model, namely data aggregation. To be specific, a new problem, Load-Balanced Data Aggregation Tree (LBDAT), is introduced finally. Our simulation results show that the proposed algorithms outperform the existing state-of-the-art approaches significantly.

INDEX WORDS: Connected dominating set, Load balance, Energy efficient, Reliability, Topology control, Stochastic wireless sensor networks

CONNECTED DOMINATING SET BASED TOPOLOGY CONTROL IN WIRELESS SENSOR NETWORKS

by

JING (SELENA) HE

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2012

CONNECTED DOMINATING SET BASED TOPOLOGY CONTROL IN WIRELESS SENSOR NETWORKS

by

JING (SELENA) HE

Committee Chair:	Dr. Yi Pan
	Dr. Yingshu Li
Committee:	Dr. Raheem Beyah
	Dr. Guantao Chen

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
August 2012

DEDICATION

This dissertation is dedicated to my parents.

ACKNOWLEDGEMENTS

I would like to show my gratitude to all whose direct and indirect support helped me complete my dissertation.

First, I would like to thank my advisors, Dr. Yi Pan and Dr. Yingshu Li for their supervision, advise, and close guidance throughout my studies at Georgia State University. They provided me unflinching encouragement and continuously support in various way. Without their assistance, I will never be able to reach this point.

Next, I would like to thank my dissertation committee members: Dr. Rehaam Beyah and Dr. Guantao Chen for helping to improve my dissertation with their thoughtful advice and suggestions.

Third, I would like to thank my group members: Shouling Ji, Mingyuan Yan, Yueming Duan, Guoliang Liu, Dr. Zhipeng Cai, and Dr. Chunyu Ai, for their friendship, willingness to help, and insightful discussions.

I am also grateful to my friends in Georgia who made my life colorful when I studied here.

My parents deserve special mention for their love, encouragement, and support throughout my life. The dissertation is impossible without their support.

Last but not least, it is a pleasure to thank everybody who made the dissertation possible, as well as express my apology that I could not mention personally one by one.

All my research works are supported by National Science Foundation (NSF) grants.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- WSN - Wireless Sensor Network
- QoS - Quality of Services
- GPS - Global Positioning System
- DNM - Deterministic Network Model
- UDG - Unit Disk Graph
- VB - Virtual Backbone
- DS - Dominating Set
- MIS - Maximal Independent Set
- CDS - Connected Dominating Set
- MCDS - Minimum Connected Dominating Set
- MOC-CDS - Minimum rOuting Cost CDS
- ABPL - Average Backbone Path Length
- LBCDS - Load-Balanced CDS
- LBAD - Load-Balanced Allocate Dominatees
- GA - Genetic Algorithm
- SNM - Stochastic Network Model
- RMCDS - Reliable MCDS
- EAP - Expected Allocation Probability

- i.i.d. - independent identically distributed
- VD - Valid Degree
- IPI - Inheritance Population Initialization
- RS - Rank Selection
- TSR - Transmission Success Ratio
- TQ - Transmission Quality
- LQI - Link Quality Index
- RWS - Roulette Wheel Selection
- GARI - GA with Random Immigrants
- GAEI - GA with Elitism-based Immigrants
- GAHI - GA with Hybrid Immigrants
- LBVB - Load-Balanced Virtual Backbone
- MDMIS - MinMax Degree MIS
- INP - Integer Nonlinear Programming
- LP - Linear Programming
- ILP - Integer Linear Programming
- DAT - Data Aggregation Tree
- PNA - Parent Node Assignment
- LBDAT - Load-Balanced DAT
- LBPNA - Load-Balanced PNA

- LBMIS - Load-Balanced MIS
- CMIS - Connected MIS
- PEDAP - Power-Efficient Data Gathering and Aggregation Protocol
- PEDAP-PA - Power Efficient Data Gathering and Aggregation Protocol-Power Aware
- MDST - Minimum Degree Spanning Tree
- L-PEDAP - Localized Power-Efficient Data Aggregation Protocol
- MLAS - Minimum-Latency Aggregation Schedule
- MLDA - Maximum Lifetime Data Gathering with Aggregation

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

1.1.1 Basic idea

Wireless sensor networks (WSNs), consisting of small nodes with sensing, computation, and wireless communications capabilities, are now widely used in many applications, including environment and habitat monitoring, traffic control, and etc. Although sensor networks share many common aspects with generic ad hoc networks, several important constraints in WSNs introduce a number of research challenges. First, due to the relatively large number of sensor nodes, it is impossible to build a global addressing scheme for the deployment of a large number of sensor nodes as the overhead of ID maintenance is high. Thus, traditional IP-based protocols may not be applied to WSNs. Second, sensor nodes are tightly constrained in terms of energy, processing, and storage capacities. Thus, they require careful resource management. Thirdly, The requirements regarding dependability and Quality of Services (QoS) are quite different. In ad hoc networks, each individual node should be fairly reliable, while in WSNs, an individual node is next to irrelevant. The QoS issues in an ad hoc network are dictated by traditional applications, while for WSNs, entirely new QoS concepts are required, which also take energy explicitly into account. Fourth, Redundant deployment will make data-centric protocols attractive in WSNs. Finally, although position awareness of sensor nodes is important, it is not feasible to use Global Positioning System (GPS) hardware for this purpose. GPS can only be used outdoors and without the presence of any obstruction. Moreover, GPS receivers are expensive and unsuitable for the construction of small cheap sensor nodes. In summary, there are commonalities, but the fact that WSNs have to support very different applications, that they have to interact with the physical environment, and that they have to carefully adjudicate various trade-offs justifies considering WSNs as a system concept distinct from ad hoc networks.

1.1.2 Deterministic Wireless Sensor Networks and Stochastic Wireless Sensor Networks

WSNs are usually modeled using the Deterministic Network Model (DNM) in recent literature. Under this model, there is a transmission radius of each node. According to this radius, any specific pair of nodes are always connected to be neighbors if their physical distance is less than this radius, while the rest of the pairs are always disconnected. The Unit Disk Graph (UDG) model is a special case of the DNM model if all nodes have the same transmission radius. When all nodes are connected to each other, via a single-hop or multi-hop path, the WSN is said to have full connectivity. In most real applications, however, the DNM model cannot fully characterize the behavior of wireless links. This is mainly due to the transitional region phenomenon which has been revealed by many empirical studies [1–4]. Beyond the “always connected” region, there is a transitional region where a pair of nodes are probabilistically connected. Such pairs of nodes are not fully connected but reachable via the so called lossy links [4]. As reported in [4], there are often much more lossy links than fully connected links in a WSN. Additionally, in a specific setup [5], more than 90% of the network links are lossy links. Therefore, their impact can hardly be neglected.

The employment of lossy links in WSNs is not straightforward, since when the lossy links are employed, the WSN may have no guarantee of full network connectivity. When data transmissions are conducted over such topologies, it may degrade the node-to-node delivery ratio. Usually a WSN has large node density and high data redundancy, thus this certain degraded performance may be acceptable for many WSN applications. Therefore, as long as an expected percentage of the nodes can be reached, that is the node-to-node delivery ratio satisfies some preset requirement, lossy links are tolerable in a WSN. In other words, full network connectivity is not always a necessity. Some applications can trade full network connectivity for a higher energy-efficiency and larger network capacity [5].

1.2 Topology Control in Wireless Sensor Networks

1.2.1 Motivation

One perhaps typical characteristic of wireless sensor networks is the possibility of deploying many nodes in a small area, for example, to ensure sufficient coverage of an area or to have redundancy present in the network to protect against node failures. These are clear advantages of a dense network deployment, however there are also disadvantages. In a relatively crowded network, many typical wireless networking problems are aggravated by the large number of neighbors: many nodes interfere with each other, there are a lot of possible routes, nodes might needlessly use large transmission power to talk to distant nodes directly (also limiting the reuse of wireless bandwidth), and routing protocols might have to recompute routes even if only small node movements have happened.

Some of these problems can be overcome by topology-control techniques. Instead of using the possible connectivity of a network to its maximum possible extent, a deliberate choice is made to restrict the topology of the network. The topology of a network is determined by the subset of active nodes and the set of active links along which direct communication can occur. Formally speaking, a topology-control algorithm takes a graph $G = (V, E)$ representing the network, where V is the set of all nodes in the network and there is an edge $(v_1, v_2) \in E \subseteq V^2$ if and only if nodes v_1 and v_2 can directly communicate with each other. Hence all active node forms an induced graph $T = (V_T, E_T)$ such that $V_T \subseteq V$ and $E_T \subseteq E$.

1.2.2 Options for topology control

To compute an induced graph T out of a graph G representing the original network G, a topology control algorithm has a few options:

- The set of active nodes can be reduced ($V_T \subset V$), for example, by periodically switching off nodes with low energy reserves and activating other nodes instead, exploiting redundant deployment in doing so.

- The set of active links/the set of neighbors for a node can be controlled. Instead of using all links in the network, some links can be disregarded and communication is restricted to crucial links. When a flat network topology (all nodes are considered equal) is desired, the set of neighbors of a node can be reduced by simply not communicating with some neighbors. There are several possible approaches to chose neighbors, but one that is obviously promising for a WSN is to limit the reach of a node's transmissions - typically by power control, but also by using adaptive modulations (using faster modulations is only possible over shorter distances) - and using the improved energy efficiency when communicating only with nearby neighbors. In essence, power control attempts to optimize the trade-off between the higher likelihood of finding a (useful) receiver at higher power values on the one hand and the increased chance of collisions/interference/reduced spatial reuse on the other hand.

- Active links/neighbors can also be rearranged in a hierarchical network topology where some nodes assume special roles. One example, illustrated in Figure 1.1, is to select some nodes as a Virtual Backbone (VB) for the network and to only use the links within this backbone and direct links from other nodes to the backbone. To do so, the backbone has to form a **Dominating Set**(DS): a subset $D \subset V$ such that all nodes in V are either in D itself or are one-hop neighbors of some node $d \in D$ ($\forall\ v \in V : v \in D\ \vee\ \exists\ d \in D : (v, d) \in E$). Then, only the links between nodes of the dominating set or between other nodes and a member of the active set are maintained. For a backbone to be useful, it should be connected. A related, but slightly different, idea is to partition the network into clusters, illustrated in Figure 1.2. Clusters are subsets of nodes that together include all nodes of the original graph such that, for each cluster, certain conditions hold (details vary). The most typical problem formulation is to find clusters with cluster heads, which is a representative of a cluster such that each node is only one hop away from its cluster head. When the (average) number of nodes in a cluster should be minimized, this is equivalent to finding a maximum (dominating) independent set (a subset $C \subset V$ such that $\forall\ v \in V - C : \exists\ c \in C : (v, c) \in E$ and no two nodes in C are joined by an edge in E, $\forall\ c_1, c_2 \in C : (c_1, c_2) \notin E$). In such a clustered network, only links within a cluster are maintained (typically only those involving the cluster head) as also selected links between clusters to ensure connectivity of the whole network. Both problems are intrinsically hard and various approximations and relaxations have been studied.

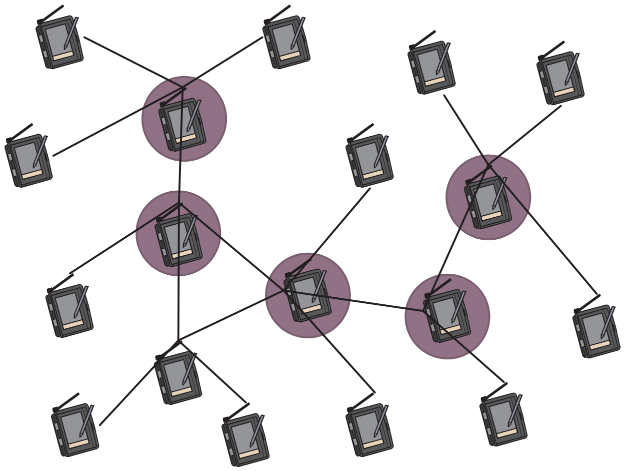


Figure 1.1. Restricting the topology by using Dominating Sets

In conclusion, there are three main options for topology control: flat networks with a special attention to power control on the one hand, hierarchical networks with backbones or clusters on the other hand.

1.2.3 Measurements of topology control algorithms

There are a few basic metrics to judge the efficacy and quality of a topology-control algorithm [6]:

- **Connectivity** Topology control should not disconnect a connected graph G. In other words, if there is a multihop path in G between two nodes u and v, there should also be some such path in T (clearly, it does not have to be the same path).
- **Stretch factors** Removing links from a graph will likely increase the length of a path between any two nodes u and v. The hop stretch factor is defined as the worst increase

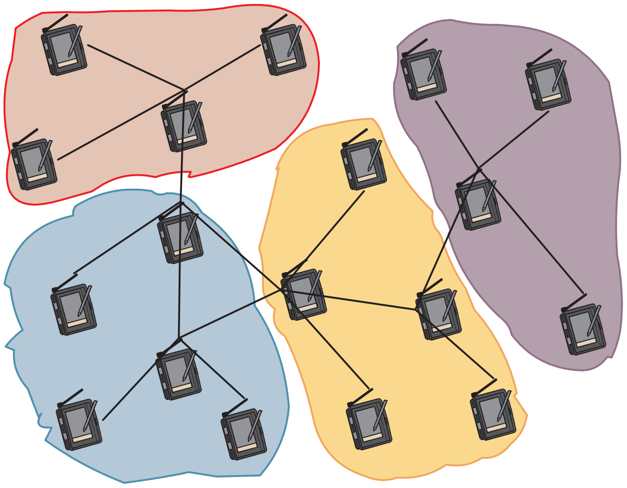


Figure 1.2. Using clusters to partition a graph

in path length for any pair of nodes u and v between the original graph G and the topology-controlled path T . Formally,

$$hop\ stretch\ factor = \max_{u,v \in V} \frac{|(u,v)_T|}{|(u,v)_G|} \tag{1.1}$$

where $(u,v)_G$ is the shortest path in graph G and $|(u,v)|$ is its length.

Similarly, the energy stretch factor can be defined:

$$hop\ stretch\ factor = \max_{u,v \in V} \frac{E_T(u,v)}{E_G(u,v)} \tag{1.2}$$

where $E_G(u,v)$ is the energy consumed along the most energy-efficient path in graph G. Clearly, topology-control algorithms with small stretch factors are desirable. It particular, stretch factors in $O(1)$ can be advantageous.

- **Graph metrics** The intuitive examples above already indicated the importance of a small number of edges in T and a low maximum degree (number of neighbors) for each node.

- **Throughput** The reduced network topology should be able to sustain a comparable amount of traffic as the original network (this can be important even in wireless sensor networks with low average traffic, in particular, in case of event showers). One metric to capture this aspect is throughput competitiveness (the largest $\varphi \leq 1$ such that, given a set of flows from node s_i to node d_i with rate r_i that are routable in G, the set with rates φr_i can be routed in T), see reference [6] for details.

- **Robustness to mobility** When neighborhood relationships change in the original graph other nodes might have to change their topology information (for example, to reactivate links). Clearly, a robust topology should only require a small amount of such adaptations and avoid having the effects of a reorganization of a local node movement ripple through the entire network.

- **Algorithm overhead** It almost goes without saying that the overhead imposed by the algorithm itself should be small (low number of additional messages, low computational overhead). Also, distributed implementation is practically a condition.

In the present context of WSNs, connectivity and stretch factors are perhaps the most important characteristics of a topology-control algorithm, apart from the indispensable distributed nature and low overhead. Connectivity treats as optimization goal.

CHAPTER 2

RELATED WORK

The idea of using a CDS as a virtual backbone was first proposed by Ephremides et al. in 1987 [7]. Since then, many algorithms that construct CDSs have been reported and can be classified into the following four categories based on the network information they used:

- Centralized algorithms;
- Substraction-based distributed algorithms;
- Distributed algorithms using single leader;
- Distributed algorithms using multiple leaders.

We use n to denote the number of sensors in a WSN, Δ to denote the maximum degree of nodes in the graph representing a WSN, and *opt* to denote the size of any optimal MCDS.

2.1 Centralized Algorithms for CDSs

Guha et al. [8] first proposed two centralized greedy algorithms with performance ratios of $2(H(\Delta) + 1)$ and $H(\Delta) + 2$ respectively, where H is a harmonic function. The greedy function is based on the number of white neighbors of each node. At each step, the one with the largest such number becomes a dominator.

Due to the instability of network topology in WSNs, it is necessary to update topology information periodically. Therefore, many distributed algorithms are proposed. These distributed algorithms can be classified into two categories: substraction-based and addition-based algorithms. The substraction-based algorithms begin with the set of all the nodes in a network, then remove some nodes according to pre-defined rules to obtain a CDS. The addition-based CDS algorithms start from a subset of nodes (usually disconnected), then include additional nodes to form a CDS. Depending on the type of the initial subset, the addition-based CDS algorithms can be further divided into single-leader and multiple-leader algorithms.

2.2 Substraction-based Distributed Algorithms for CDSs

Wu and Li first proposed a completely distributed algorithm in [9] to obtain a CDS. The CDS construction procedure consists of two stages. In the first stages, each node collects its neighboring information by exchanging message with the one-hop neighbors. If a node finds that there is a direct link between any pair of its one-hop neighbors, it removes itself from the CDS. In the second stage, additional heuristic rules are applied to further reduce the size of the CDS. Wu's algorithm [9] uses Rule 1 and Rule 2, where a node is removed from the CDS, if all its neighbors are covered by its one or two direct neighbors. Later, Dai's [10] work generalizes this as Rule k, in which coverage is defined by an arbitrary number of connected neighbors. Dai's algorithm is reduced to Wu's algorithm when k is 1 or 2.

2.3 Addition-based Distributed Algorithms for CDSs

Single-leader distributed algorithms for CDSs use one initiator to initialize the distributed algorithms. Usually, a base station could be the initiator for constructing CDSs in WSNs. In these distributed algorithms, a spanning tree rooted at the initiator is first constructed, and then maximal independent sets (MISs) are identified layer by layer, finally a set of connectors to connect the MISs is ascertained to form a CDS. Wan et al. [11] presented an ID-based distributed algorithm to construct a CDS using a single initiator. For UDGs, Wan et al.'s [11] approach guarantees that the approximation factor on the size of a CDS is at most $8opt + 1$. The algorithm has $O(n)$ time complexity and $O(n \log n)$ message complexity. Subsequently, the approximation factor on the size of a CDS was improved in another work reported by Cardei et al. [12], in which the authors used the degree-based heuristic and degree-aware optimization to identify Steiner nodes as the connectors in the CDS construction. The approximation factor on the size of a CDS is improved to $8opt$, while this distributed algorithm has $O(n)$ message complexity, and $O(\Delta n)$ time complexity. Later, Li et al. [13] reported a better approximation factor of $5.8 + \ln 4$ by constructing a Steiner tree when connecting all the nodes in the MISs.

Distributed algorithms with multiple leaders do not require an initiator to construct a CDS. Alzoubi et al.'s technique [14] first constructs an MIS using a distributed approach without a leader or tree construction and then interconnects these MIS nodes to get a CDS. Li et al. proposed a distributed algorithm r-CDS in [15], whose performance ratio is 172. r-CDS is a completely distributed one-phase algorithm where each node only needs to know the connectivity information within its 2-hop-away neighborhood.An MIS is constructed based on each node's r value which is defined as the number of this node's 2-hop-away neighbors minus the degree of this node. The nodes with smaller r values are preferred to serve as MIS nodes.Adjih et al. [**?**] presented an approach for constructing an MCDS based on multi-point relays (MPR), but there is no approximation analysis of the algorithm yet. Recently, in [**?**], another distributed algorithm was proposed whose performance ratio is 147. This algorithm contains three steps. Step 1 constructs a forest in which each tree is rooted at a node with the minimum ID among its 1-hop away neighbors. Step 2 collects the neighborhood information, which is used in Step 3 to connect neighboring trees.

2.4 Other Algorithms

Because CDSs can benefit a lot to WSNs, a variety of other factors are considered when constructing CDSs. There are more than one CDS can be found for each WSN. To conserve energy, all CDSs are constructed and each CDS serves as the virtual backbone duty cycled in [16]. For the sake of fault tolerance, k-connect m-dominating sets [17] are constructed, where k-connectivity means between any pair of backbone nodes there exist at least k independent paths, and m-dominating represents that every dominatee has at least m adjacent dominator neighbors. To minimize delivery delay, a special CDS problem — Minimum rOuting Cost CDS (MOC-CDS) [18] is proposed, where each pair of nodes in MOC-CDS has the shortest path. The work [19] considers size, diameter, and Average Backbone Path Length (ABPL) of a CDS in order to construct a CDS with better quality.

2.5 Related Literatures under the SNM model

Traditional routing schemes only considered fully connected links as a path of nodes in a WSN, and then send data through that sequence of node. Compare with the fully connected links, lossy links only provide probabilistic connectivity. However, there exists more lossy links in a WSN. Therefore, opportunistic routing schemes (e.g., ExOR [20] and More [21]) proposed to take lossy links as advantage. The Opportunistic Routing scheme called ExOR [20] proposed a new unicast routing technique for multi-hop wireless networks. ExOR forwards each packet through a sequence of nodes, who can successfully receive the transmission and are close to the destination. ExOR explore package overhearing along lossy links. When a lossy link succeeds, some transmissions can be saved. Later, Chachulski et al. combined random network coding with opportunistic routing to support both unicast and multicast routing in More [21]. The successful of opportunistic routing indicates that lossy links provide the potential throughput increase.

Recently, lots of works [5, 22–25] study the impact of lossy links to the topology control. Ma et al., in [22, 23] worked on achieving energy-efficiency by turning off redundant nodes and links, while still satisfying the given QoS requirements. And Liu et al., investigated how to control the minimal transmission range for each node while the global network reachability satisfies some constraints in [5, 24, 25].

2.6 Remarks

All the above mentioned existing works consider to construct an MCDS, a k-connect m-dominating CDS, a minimum routing cost CDS or a bounded-diameter CDS. Unfortunately, they do not consider the load-balance factor when constructing a CDS. In contrast, in this dissertation, we first show an example to illustrate that a traditional MCDS cannot prolong network lifetime by reducing the communication cost. Instead, it actually increase the workload imbalance among dominators, which leads to the reduction of network lifetime.

Based on this observation, we then study to build an LBCDS and load-balancedly allocate dominatees to dominators. We use two distinct ways to solve the problems. One way is two-phase. We first build an LBCDS. After constructing an LBCDS, we investigate how to load-balancedly allocate dominatees to dominators. The probability based centralized and distributed algorithms are proposed to obtain an optimal allocation scheme. The upper bound and lower bound of the performance ratios of the proposed algorithms are analyzed in Chapter 3. The other way is only one phase. An effective GA named LBCDS-GA is proposed to solve the problem in Chapter 4. Comprehensive theoretical analysis are given in Chapter 6.

On the other hand, all the above mentioned existing works either consider to construct an MCDS under the DNM model or design a routing protocol, investigate the topology control under the SNM model. To the best knowledge of us, however, none of them attempt to construct an MCDS under the SNM model, which is more realistic. This is the major motivation of this research work. GAs are a family of computational models inspired by evolution, which have been applied in a quite broad range of NP-Hard optimization problem [26–29]. Therefore, a GA based method, namely RMCDS-GA, is proposed in Chapter 5 to construct a reliable MCDS under the SNM model. In RMCDS-GA, each possible CDS in a WSN is represented to be a chromosome (feasible/potential solution), and the fitness function is to evaluate the trade-off between the CDS reliability and the size of the CDS represented by each chromosome.

Finally, I apply the constructed LBCDS to a practical application under the realistic SNM model, namely data gathering. Data Gathering is a fundamental task in WSNs. For applications where each sensor continuously monitors the environment and periodically reports to the sink, a tree-based topology is often used to gather and aggregate sensing data from sensor nodes. Thus, data gathering trees with aggregation are also referred to as Data Aggregation Trees (DATs). To be specific, a new problem, Load-Balanced Data Aggregation Tree (LBDAT), is introduced in Chapter 7.

CHAPTER 3

CONSTRUCTING A LOAD-BALANCED VIRTUAL BACKBONE IN WIRELESS SENSOR NETWORKS

3.1 Motivation

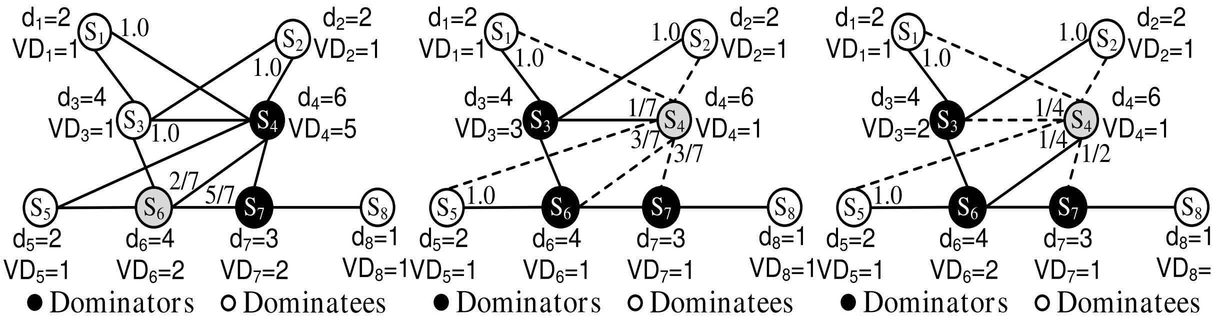
Wireless Sensor Networks (WSNs) are deployed for monitoring and controlling of systems where human intervention is not desirable or feasible. One typical characteristic of WSNs is the possibility of deploying many nodes in a area to ensure sufficient coverage of an area or/and to have redundancy against node failures. However, in a relatively crowded network, many problems are aggravated: 1). Many nodes interfere with each other, 2). There are a lot of possible routes, 3). Nodes might needlessly use large transmission power to talk to distant nodes directly, and 4) Routing protocols might have to recompute routes even if only a small number of nodes changed their locations. These problems can be overcomed by selecting some nodes as a *Virtual Backbone* (VB) for a network, in which only the links within this backbone and direct links from other nodes to the backbone nodes are mainly used in the WSN. Usually, we use a **Dominating Set** (DS) to serve as a backbone for a WSN, which is a subset of nodes in the network where every node is either in the subset or a neighbor of at least one node in the subset. For a backbone to be useful, it should be connected, namely, **Connected Dominating Set** (CDS). The nodes in a CDS are called *dominators*, otherwise, *dominatees*. In a WSN with a CDS as its VB, dominatees only forward their data to their connected dominators. Moreover, the CDS with the smallest size (the number of nodes in the CDS) is called a **Minimum-sized Connected Dominating Set** (MCDS).

With the help of a CDS, routing is easier and can adapt quickly to topology change. Only dominators need to maintain the routing information. Therefore, the search space for the route is reduced only within the CDS. Furthermore, if there is no topology change in the subgraph induced by the CDS, there is no need to update the routing information. Moreover, in addition to routing [30], a CDS has many other applications in WSNs, such as broadcasting [31], topology control [32–39], coverage [40–42], data collection [43], and data aggregation [44]. Clearly, the benefits of a CDS can be magnified by making its size smaller. Therefore, it is desirable to build an MCDS to reduce the number of nodes and links involved in communication. As a matter of fact, constructing a CDS, especially an MCDS for WSNs is one way to extend network lifetime.

Ever since the idea of employing a CDS for WSNs was introduced in [7], a huge amount of effort has been made to find CDSs with variety of features for different applications, especially the MCDS. In the seminal work [8], Guha and Kuller first modeled the problem of computing the smallest CDS as the MCDS problem in a general graph, which is a well-know NP-Hard problem [45]. After that, to make a CDS more resilient in mobile WSNs, the fault-tolerance of a VB is considered. In [17], k-connected and m-dominated sets are introduced as a generalized abstraction of a fault-tolerance VB. In [18], the authors proposed a Minimum rOuting Cost Connected Dominating Set (MOC-CDS), which aims to find a minimum CDS while assuring that any routing path through this CDS is the shortest in WSNs. Additionally, the authors investigate the problem of constructing a quality CDS in terms of size, diameter, and Average Backbone Path Length (ABPL) in [19].

Unfortunately, to the best of our knowledge, all the related works did not consider the **load-balance** factor when they construct a CDS. If the workload on each dominator in a CDS are not balanced, some heavy-duty dominators deplete their energy quickly. Then, the whole network might be disconnected. Hence, intuitively, we not only have to consider to construct an MCDS, but also need to consider to construct a load-balanced CDS (LBCDS). An illustration of an LBCDS is depicted in Fig. 3.1, in which dominators are marked as black nodes, while white nodes represent dominatees. In Fig. 3.1(b) and Fig. 3.1(c), solid lines represent that the dominatees are allocated to the connected dominators, while the dashed lines represent the communication links in the original graph shown in Fig. 3.1(a). According to the traditional MCDS construction algorithms, a CDS $\{s_4, s_7\}$ with size 2 is obtained for the network shown in Fig. 3.1(a). However, There are two severe drawbacks of the CDS shown in Fig. 3.1(a). For convenience, the set of neighboring dominatees of the dominator s_i is denoted by $ND(s_i)$. First, $ND(s_4) = \{s_1, s_2, s_3, s_5, s_6\}$, which represents that dominator s_4 connects to 5 different dominatees, and $ND(s_7) = \{s_6, s_8\}$. If every dominatee has the same amount of data to be transferred through the connected dominator at a fixed data rate, dominator s_4 must deplete its energy much faster than dominator s_7, since dominator s_4 has to forward the data collected from 5 connected dominatees. Second, dominatee s_6 connects to both dominators. If s_6 chooses dominator s_4 as its data forwarder, obviously, only one dominatee s_8 can forward its data to dominator s_7. In this situation, the workload imbalance in the CDS is further amplified. Consequently, the entire network lifetime is shortened. We show a counter-example in Fig. 3.1(b), where the constructed CDS is $\{s_3, s_6, s_7\}$. According to the topology shown in Fig. 3.1(b), we can get the dominatee sets of each dominator: $ND(s_3) = \{s_1, s_2, s_4\}$, $ND(s_6) = \{s_4, s_5\}$, and $ND(s_7) = \{s_4, s_8\}$. Compared with the MCDS constructed in Fig. 3.1(a), the numbers of dominatees of all the dominators in Fig. 3.1(b) are very similar. For convenience, we use $A(s_i) = \{s_j | s_j \text{ is a dominatee and } s_j \text{ forward its data to } s_i\}$ to represent the dominatees allocated to a dominator s_i. Thus, we can have two different dominatee allocation schemes shown in Fig. 3.1(b) and Fig. 3.1(c) respectively. One is: $A(s_3) = \{s_1, s_2, s_4\}$, $A(s_6) = \{s_5\}$, and $A(s_7) = \{s_8\}$. The other one is: $A(s_3) = \{s_1, s_2\}$, $A(s_6) = \{s_4, s_5\}$, and $A(s_7) = \{s_8\}$. Apparently, the workload on each dominator is almost evenly distributed in the CDS constructed in Fig. 3.1(c). Intuitively, the construction algorithm and dominatee allocation scheme shown in Fig. 3.1(c) can extend network lifetime notably.

To solve the workload imbalance problem of an MCDS, in this chapter, we first investigate how to construct a load-balanced CDS. When we build the LBCDS, we consider the degree of each dominator as the indicator of potential future workload. Taking the degree of each dominator in consideration, we use the *p-norm* to measure how balanced the LBCDS can make. The details are introduced in Section 3.3. After constructing an LBCDS, we explore how to Load-Balancedly Allocate dominatees to Dominators (LBAD). We propose



(a) (b) (c)

Figure 3.1. Illustration of regular CDS and LBCDS.

a novel probability-based algorithm to solve this problem. The detailed design, algorithm description and theoretical analysis are presented in Section 3.4.

3.2 LBVB Problem Statement

In this section, we introduce the network model and define the LBCDS problem and the LBAD problem formally.

3.2.1 Network Model

We assume a WSN is deployed in a square with area size $A = cn$, where c is a constant and the WSN is consisting of n sensors, denoted by $s_1, s_2, \ldots, s_n$ respectively. All sensors are independent identically distributed (i.i.d.) over the whole network. We also assume all nodes have the same transmission range. We modeled the WSN as a connected undirected general graph $G = (V, E)$, in which V represents node set and E represents the link set. $\forall\ u, v \in V$, there exists an edge (u, v) in G if and only if u and v are in each other's transmission range. In this chapter, we assume edges are undirected (bidirectional), which means two linked nodes are able to transmit and receive information from each other.

The *load-balance* factor is our major concern in this work. Thus, finding an appropriate measurement to evaluate load-balance is the key to solve the LBCDS and LBAD problems. We use *p-norm* to measure load-balance in this chapter. The definition of *p-norm* is given as follows:

Definition 3.2.1. *p-norm.* Thep-norm of an $n \times 1$ vector $\mathbb{X} = (x_1, x_2, \cdots, x_n)$ is:

$$|\mathbb{X}|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}} \tag{3.1}$$

The authors in [46] stated that p-norm shows interesting properties for different values of p. If p is close to 1, the information routes resemble the geometric shortest paths from the sources to the sinks. For $p = 2$, the information flow shows an analogy to electrostatics field, which can be used to measure the load-balance among x_i. More importantly, the smaller the *p-norm* value, the more load-balanced the interested feature vector $\mathbb{X}$.

In this chapter, we use node degree (*Definition 3.2.2*) and the number of dominatees connected to a dominator (*Definition 3.2.3*) of the interested node set as the information vector $\mathbb{X}$, since the degree of each node and the number of the dominatees connected to a dominator is a potential indicator of traffic load.

We use the WSNs shown in Fig. 3.1 to illustrate how to use *p-norm* to measure the load-balance of CDSs. Two different CDSs for the same network are identified in Fig. 3.1. The degree of the node s_i is denoted by d_i in Fig. 3.1. $|d_i - \bar{d}|$ are used as the information vector $\mathbb{X}$, where $\bar{d}$ is the mean degree of each graph in Fig. 3.1. Therefore, the *p-norm* value of the CDS shown in Fig. 3.1(a) is $\sqrt{9}$. Similarly, in Fig. 3.1(b), the *p-norm* value is $\sqrt{2}$. Clearly, $\sqrt{2} < \sqrt{9}$, which implies that the CDS in Fig. 3.1(b) is more load-balanced than the CDS in Fig. 3.1(a).

After we construct an LBCDS, the next step is to allocate dominatees to each dominator in the LBCDS. The *p-norm* can again be used to measure the load-balance of different allocation schemes, in which, the number of dominatees connected to a dominator of the interested node set is used as the information vector $\mathbb{X}$. An illustration example is shown in Section 3.4.

3.2.2 LBCDS Problem Definition

Now we give the formal definition of the problems we investigate in this chapter.

Definition 3.2.2. *Load-balanced CDS (LBCDS).* For a WSN represented by graph $G = (V, E)$, the LBCDS problem is to find a node set $D \subseteq V$, $D = \{s_1, s_2, \cdots, s_M\}$ such that

1. $G[D] = (D, E')$, where $E' = \{e| \ e = (u, v), u \in D, v \in D, (u, v) \in E)\}$, is connected.

2. $\forall \ u \in V$ and $u \notin D$, $\exists \ v \in D$, such that $(u, v) \in E$.

3. $min|D|_p = (\sum_{i=1}^{M} |d_i - \bar{d}|^2)^{\frac{1}{2}}$.

Definition 3.2.3. *Load-balanced Allocation of Dominatees (LBAD).* For a WSN represented by graph $G = (V, E)$, and a CDS $D = \{s_1, s_2, \cdots, s_M\}$. The number of the dominatees connecting to each dominator s_i $(1 \leq i \leq M)$ is denoted by $|A(s_i)|$, and the expected allocated dominatees of each dominator is denoted by $|\bar{A}|$. The LBAD problem is to find M disjoint sets on V, i.e. $A(s_1), A(s_2), \cdots, A(s_M)$, such that

1. Each set $A(s_i)$ $(1 \leq i \leq M)$ contains exactly one dominator s_i.

2. $\bigcup_{i=1}^{M} A(s_i) = V, A(s_i) \bigcap A(s_j) = \emptyset \ (1 \leq i \neq j \leq M)$.

3. $\forall \ u \in A(s_i)$ $(1 \leq i \leq M)$ and $u \neq s_i$, such that $(u, s_i) \in E$

4. $min|D|_p = (\sum_{i=1}^{M} ||A(s_i)| - |\bar{A}||^2)^{\frac{1}{2}}$.

3.3 Load-Balanced CDS

In essential, we design a greedy algorithm to solve the LBCDS problem. The algorithm starts from an empty Dominator Set (DS). Each time, it adds the node into the DS set that has the smallest $|d_i - \bar{d}|$ value (where $1 \leq i \leq n$). If there exists a tie on $|d_i - \bar{d}|$ value, we use greater d_i value to break the tie, since the nodes with higher degree can make the algorithm converge faster. The algorithm terminates when the nodes in the DS set form a CDS.

The pseudocode of the greedy algorithm is shown in Algorithm 1.

LBCDS-Approximate algorithm as shown in Algorithm 1 is a centralized one-phase greedy algorithm. Initially, all the nodes are white. All black nodes form an LBCDS finally. We use the following terms in the algorithm,

d_i: The degree of node s_i.

$\bar{d}$: The mean degree.

cur_degree: The degree number used in each round.

isFound: A flag to indicate whether there exists a node s_i satisfying $|d_i - \bar{d}| = cur\_degree$ in each round.

Algorithm 1 : LBCDS-Approximate

```
Require: A WSN represented by graph $G = (V, E)$; Node degree $d_i$; Mean degree of the
    graph $\bar{d}$.
 1: Initialize $cur\_degree = \bar{d}$
 2: Initialize $isFound = false$
 3: if All black nodes form a CDS then
 4:     return All black nodes
 5: end if
 6: Sort the $n$ sensors based on $|d_i - \bar{d}|$ values. If there exists a tie, use greater $d_i$ value to
    break the tie, where $1 \leq i \leq n$.
 7: for $i = 1$ to $n$ do
 8:     if $|d_i - \bar{d}| = cur\_degree$ and node $s_i$ is not marked as black then
 9:         Mark node $s_i$ black {dominator node}
10:         $isFound = true$
11:     end if
12: end for
13: if $isFound = false$ then
14:     $cur\_degree = cur\_degree + 1$
15: end if
16: Call Algorithm 1 {recursive call}
```

Initially, *cur_degree* is set to $\bar{d}$. From line 6 to 11, the algorithm searches the nodes with the degree *cur_degree*; mark them as black shown at Line 8; and set the flag *isFound* to *true* at Line 9. If *isFound* is *false* after searching all the n nodes, which means no satisfied sensor can be found in this round, then the algorithm gives to the next round by updating *cur_degree* at Line 13. Repeat the above procedures until all the black nodes form a CDS (from Line 3 to 5).

3.3.1 Example Illustration

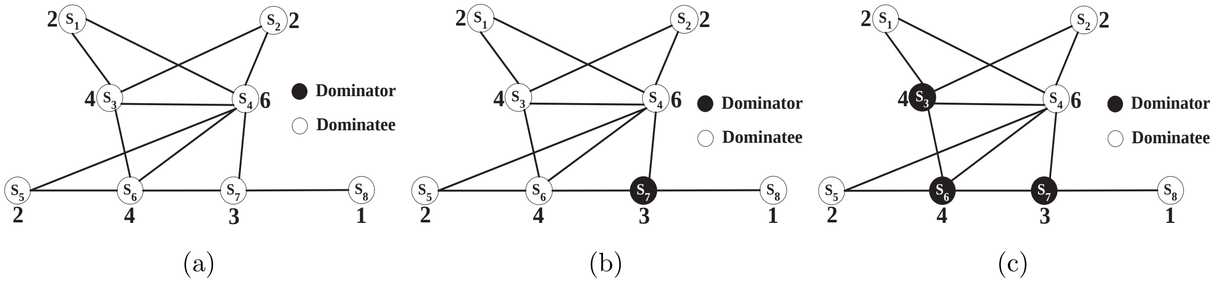


Figure 3.2. Construction of an LBCDS.

We use the WSN shown in Fig. 3.1(a) to illustrate how to build a LBCDS. Based on each node's degree, we can calculate $\bar{d} = 3$. According to the aforementioned LBCDS construction algorithm, in the first round, all the nodes with degree 3 are added into the DS set. Thus, node s_7 is added into the DS set, since $d_7 = \bar{d} = 3$. In the next round, nodes s_3 and s_6 with degree 4 are added into the DS set. Nodes s_1 and s_2 are not added into the DS set, because they have smaller degree values than nodes s_3 and s_6. So far, there are three nodes in the DS set, which forms a CDS, therefore the algorithm terminates. Finally, we get an LBCDS which is $\{s_3, s_6, s_7\}$.

3.3.2 Remarkes

According to Definition 3.2.2, *p-norm* is the measurement of the load-balance for CDSs. The smaller the *p-norm* value, the more load-balanced the CDS is. Moreover, $|d_i - \bar{d}|$ is the information vector $\mathbb{X}$ in Equation 3.1. The LBCDS construction algorithm greedily searches the dominators with the smallest $|d_i - \bar{d}|$ values. Based on this greedy criterion, the algorithm can output a CDS with a small *p-norm* value.

3.4 Load-Balanced Allocation of Dominatees

Constructing an LBCDS is the foundation to solve the LBAD problem. In this section, we introduce how to use an existing LBCDS for load-balanced allocation of dominatees.

3.4.1 Terminologies

In a traditional/naive way, such as the work in [44], each dominatee sets its data forwarder to be the connected dominator with the smallest ID. Thus, the load-balance factor in not take into account. In some environment, the dominator with the smallest ID, which is chosen by majority dominatees, probably has heavy workload than the other dominators with a smaller number of dominatees. Therefore, the node degree cannot imply the potential workload precisely. In a WSN with a CDS as the VB, only the dominator and dominatee links contribute to the workload. Based on this observation, we define the following:

Definition 3.4.1. Valid Degree (VD). For each dominatee s_i, VD_i is the number of its connected dominators. For each dominator s_j, VD_j is the number of its allocated dominatees.

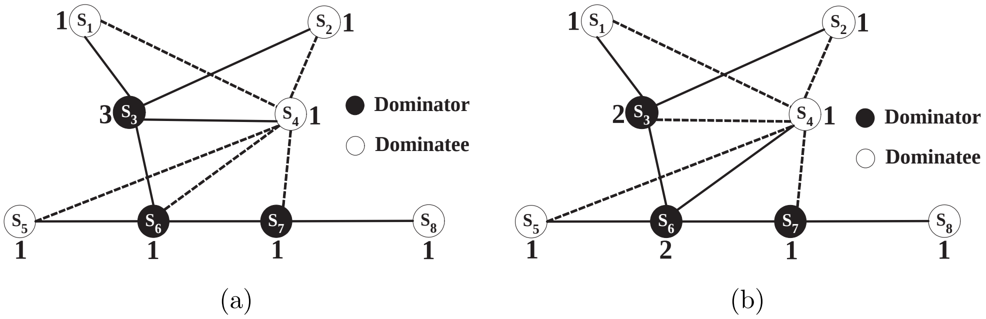


Figure 3.3. Allocation examples.

Fig. 3.1(b) and Fig. 3.1(c) illustrate an imbalanced and a balanced allocations of dominatees. Using $|VD_i - \bar{d}|$ as the information vector $\mathbb{X}$, we still can use *p-norm* to measure the load-balance factor of the dominatee allocation scheme. Therefore, the *p-norm* value of the allocation scheme shown in Fig. 3.1(b) is $\sqrt{8}$. Similarly, in Fig. 3.1(c), the *p-norm* value is $\sqrt{6}$. Clearly, $\sqrt{6} < \sqrt{8}$, which implies the allocation scheme shown in Fig. 3.1(c) is more load-balanced than the scheme shown in Fig. 3.1(b).

Due to the instability of network topology, it is not practical to always allocate one dominatee to one dominator. In order to adapt to network topology change, a terminology *Expected Allocation Probability (EAP)* is proposed as follows:

Definition 3.4.2. Expected Allocation Probability (EAP). For each dominatee and dominator pair, there is an *EAP*, which represents the expected probability that the dominatee is allocated to the dominator.

The *EAP* value associated on each dominatee and dominator pair directly determines the load-balance factor of each allocation scheme. We conclude the properties of the *EAP* values as follows:

1) For each dominatee s_i, $\sum\limits_{j=1}^{|NE(s_i)|} EAP_{ij} = 1$.

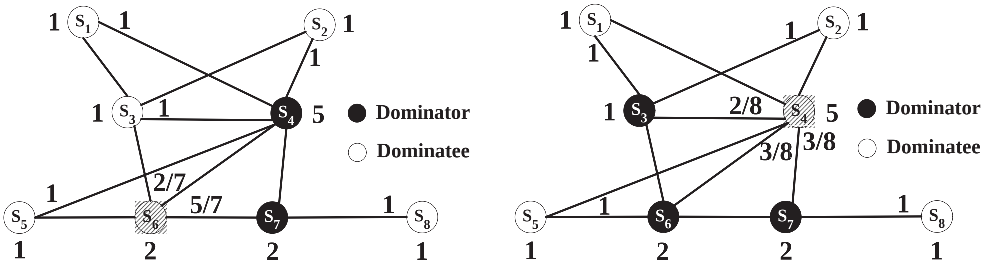
where $NE(s_i)$ is the set of neighboring dominators of s_i, $|NE(s_i)|$ is the number of the nodes in set $NE(s_i)$;

2) In order to produce the most load-balanced allocation scheme, which is obtained when the expected number of allocated dominatees of all the dominators are the same. It can be formulated as follows:

$$EAP_{i1} \times VD_1 = \cdots = EAP_{i|NE(s_i)|} \times VD_{|NE(s_i)|} \tag{3.2}$$

An example about how to calculate *EAP* values is shown in Fig. 3.1. The gray nodes i.e. s_6 in Fig. 3.1(a) and s_4 in Fig. 3.1(b) are dominatees connected to more than one dominator. The numbers shown on the links are the *EAP* values of each dominatee and dominator pair. If a dominatee only connects to one dominator, the *EAP* value associated with the pair is equal to 1. Otherwise, Equation 3.2 can be used to calculate the *EAP* values of all connected dominators links.

As mentioned in Definition 3.4.1, the degree of each dominator is not a good indicator of workload. Hence, after allocating dominatees through the *EAP* scheme, the informa-



(a) (b)

Figure 3.4. Load-Balanced Dominatee Allocation with Expected Allocation Probability (EAP).

tion vector $\mathbb{X}$ in Equation 3.1 is the summation of the EAP values of dominator j minus the expected number of allocated dominatees of each dominator, which is formulated by $\sum_{i=1}^{ND(s_j)} |EAP_{ij} - \bar{p}|$, where $\bar{p} = \frac{n-M}{M}$ representing the expected number of allocated dominatees of each dominator. In Fig. 3.1(a), $\bar{p} = 3$. Therefore, the *p-norm* value of the allocation scheme shown in Fig. 3.1(a) is $\sqrt{3.306}$. Similarly, in Fig. 3.1(c), $\bar{p} = \frac{5}{3}$ and the *p-norm* value is $\sqrt{0.51}$. Apparently, $\sqrt{0.51}$ is much smaller than $\sqrt{3.306}$, which means the allocation scheme in Fig. 3.1(c) is much more load-balanced than the scheme in Fig. 3.1(a). There are two reasons to have a very small *p-norm* value in Fig. 3.1(c). First, an LBCDS is used. Second and more important, we adopt the probability-based dominatee allocation scheme. The allocation criterion is that making the expected number of allocated dominatees of each dominator the same. The criterion implies, on average, the expected number of allocated dominatees of all the dominators are the same. If every dominatee has the same amount of data to be transferred through the allocated dominator at a fixed data rate, then the probability-based allocation scheme can achieve the maximized load-balance on the expected workload among dominators.

3.4.2 Algorithm Description

The allocation system starts from finding an LBCDS using the aforementioned LBCDS construction Algorithm. Then the EAP value is calculated for each dominatee and dominator pair. EAP only indicates the probability the dominatee will be assigned to the dominator for each dominator and dominatee pair. Thus the final step is allocate the dominatees to the dominators. We use the stochastic allocation, which is a dominatee is randomly assigned to an adjacent dominator based on the EAP value.

Fig. 3.1(c) shows an example about how to perform the stochastic dominatee allocation. In Fig. 3.1(c), only dominatee s_4 connects to more than one dominator and its associated EAP values are: $EAP_{43} = \frac{1}{4}$; $EAP_{46} = \frac{1}{4}$; and $EAP_{47} = \frac{1}{2}$. Dominatee s_4 generates a random number $\delta = 0.358$. If $\delta \in [0, 0.25]$, s_4 chooses dominator s_3 , else s_4 chooses dominator s_6 if $\delta \in (0.25, 0.5]$, otherwise s_4 chooses dominator s_7 if $\delta \in (0.5, 1]$. Since $\delta = 0.358$, dominatee s_4 is assigned to dominator s_6.

Each time a dominatee which is connected to more than one dominator wants to send data, it must redo the last step to pick a proper dominator based on the EAP probability and then forward its data. One example illustrates how to choose a random dominator based on EAP probability is shown in subsection 3.4.2.

In Section 3.3, the detailed description of how to construct a LBCDS is introduced. The third step is a trivial process. In the rest of this section, we design two algorithms to implement the second step, namely how to calculate the EAP value for each dominator and dominatee pair. We introduce the centralized algorithm first as follow:

Centralized Algorithm We propose a constrained non-linear programming scheme to solve the LBAD problem. The essence of allocating dominatees is to achieve maximum load-balance among dominators. We use the *p-norm* value to measure the load-balance factor. Consequently, the objective of the optimization problem is to minimize the *p-norm* value of the dominatee allocation scheme. In addition, the constraint is to guarantee Property 1 of EAP values. To conclude, the optimization problem is formulated as follows:

$$\begin{aligned} &Minimize: && |EAP|_p = \sum_{j=1}^{M} (\sum_{i=1}^{|ND(s_j)|} |EAP_{ij} - \bar{p}|)^2 \\ &Subject\ to: && for\ dominatee\ s_i, \sum_{j=1}^{|ND(s_i)|} EAP_{ij} = 1 \\ &Where: && 0 \leq EAP_{ij} \leq 1 \end{aligned} \tag{3.3}$$

The centralized algorithm is shown in Algorithm 2:

Algorithm 2 : LBAD-Centralized

```
Require: A WSN represented by a graph $G = (V, E)$; an LBCDS: $G[D] = (D, E')$
Solve the constrained non-linear programming formulated in Equation 3.3. Let $EAP_{ij}$ be the optimal solution of the non-linear programming.
for each dominatee $s_i$ do
    Generate a number $\delta$ between 0 and 1
    if $\delta \in [\sum_{j=0}^{k-1} EAP_{ij}, \sum_{j=0}^{k} EAP_{ij}]$, where $0 < k \leq NE(s_i)$ then
        mark the link between dominatee $s_i$ and dominator $s_k$ black
    end if
end for
return All black links
```

The centralized algorithm can guarantee to find the optimal solution. However, solving the non-linear programming is too time and energy consuming. If the precision is the major concern, we can solve the non-linear programming formulas at the base station. Nevertheless, if the energy and time are the primary concern, a distributed algorithm to find a near-optimal solution is preferred. We therefore propose the distributed algorithm as follows:

Distributed Algorithm The objective of the LBAD problem is to find a load-balanced dominatee allocation scheme. The most load-balanced allocation scheme is that the expected number of allocated dominatees of all the dominators are the same, which is formulated in Equation 3.2. Additionally, we guranntee Property 1 of EAP values. By listing all the equations, we can solve them to get EAP_{ij} of each connected dominatee s_i and dominator s_j, which is formulated as follows:

$$\begin{gathered} EAP_{i1} : EAP_{i2} : \cdots : EAP_{i|ND(s_i)|} = \\ VD_2 \times VD_3 \times \cdots \times VD_{|ND(s_i)|} : \\ \cdots : \prod_{j=1, i \neq j}^{|ND(s_i)|} VD_j : \ldots \\ : VD_1 \times VD_2 \times \cdots \times VD_{|ND(s_i)|-1} \end{gathered} \tag{3.4}$$

Therefore, the distributed LBAD problem can be transformed to calculate the EAP

value of each dominatee locally. The dominatee stochastic allocation step is the same as the centralized algorithm.

The distributed algorithm is a localized two-phase algorithm where each node only needs to know the connectivity information within its 1-hop-away neighborhood. All the nodes get the VD values by broadcasting messages to all its neighbor nodes, and then store the values locally. Each dominatee calculates the EAP values using Equation 3.4.

The pseudocode is given in Algorithm 3. We call it *LBAD-Distributed* algorithm. We use the following terms in Algorithm 3,

VD_k: The VD value of each node s_k.

$ND(s_k)$: The set of neighboring dominatees of dominator s_k.

$|ND(s_k)|$: The number of the nodes in set $ND(s_k)$.

$NE(s_k)$: The set of the neighboring dominators of dominatee s_k.

$|NE(s_k)|$: The number of the nodes in set $NE(s_k)$.

EAP_{ij}: The EAP value of each connected dominatee s_i and dominator s_j pair.

Each node s_i maintains the following data structures:

1) s_i's ID, initialized to 0.

2) The dominator/dominatee flag f. 1 means dominator; 0 means dominatee. It is initialized to 0.

3) $|ND(s_i)|$, if s_i is a dominator; $|NE(s_i)|$, if s_i is a dominatee, initialized to 0.

4) Neighboring dominator/dominatee lists. A list contains: a dominator/dominatee's ID, its VD value, and EAP_{ij}, initialized to $\emptyset$.

Initially, each node initializes its data structures and broadcasts a *hello* message containing its ID, VD, and f to its 1-hop neighbors to exchange neighbors' information. All the nodes run the following:

• For any dominator s_i, upon receiving a *hello* message from node s_j: if s_j is a dominator, ignore the message. If s_j is a dominatee, update $|ND(s_i)|$ and dominatee s_j's ID and VD value in the neighboring dominatee list of the dominator s_i.

• For any dominatee s_i, upon receiving a *hello* message from node s_j: if s_j is a dominatee, ignore the message. If s_j is a dominator, update $|NE(s_i)|$ and dominator s_j's ID and VD value in the neighboring dominator list of the dominatee s_i. Calculate and store EAP_{ij} based on the VD values stored in the neighboring dominator list using Equation 3.4.

Algorithm 3 : LBAD-Distributed

```
***Initialization Phase:***
For each dominatee $s_i$, get the number of neighbor dominators (denoted by $|NE(s_i)|$) and store locally.
For each dominator $s_j$, get the number of neighbor dominatees (denoted by $|ND(s_j)|$) and store locally.
***Allocation Phase:***
For each dominatee $s_i$, calculate its neighboring dominators' $EAP_{ij}$ by the following formula:
$EAP_{i1} : EAP_{i2} : \cdots : EAP_{i|ND(s_i)|} = VD_2 \times VD_3 \times \cdots \times VD_{|ND(s_i)|} : \cdots : VD_1 \times VD_2 \times \cdots \times VD_{|ND(s_i)|-1} = \prod_{j=1,i\neq j}^{|ND(s_i)|} VD_j$
```

The distributed algorithm is a 2-phase algorithm. The first phase is the initialization phase, where all the nodes get its neighborhood information and update its own data structure locally. In practical, it is hard to decide when the initialization phase completes. Hence we set a timer. If the timer expires, the second phase, allocation phase, starts to work. In the allocation phase, every dominatee calculates the EAP values of its connected dominators using Equation 3.4. We only use 1-hop-away neighborhood information to calculate the EAP values locally. Therefore, it is an easy and efficient algorithm. Nevertheless, only using the 1-hop-away neighborhood information to calculate the EAP values may lead us to find a local optimal solution instead of a global optimal solution.

3.4.3 Analysis

Based on the assumptions mentioned in section 3.2.1, n sensors are i.i.d. in a square with area size $A = cn$. The communication range of each sensor is 1. Thus, we denote the unit circle associated with each sensor s_i by c_i. According to the network model, the following lemma can be proved:

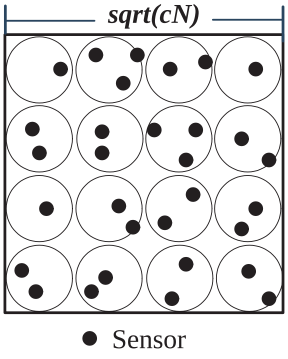


Figure 3.5. Network Partition

Lemma 1. For any unit circle c_i, let the random variable Z_i denote the number of the sensors within it. Then, the probability that c_i contains more than $\ln n$ sensors is no greater than $\frac{\exp((\exp(\gamma)-1)\times(\frac{\pi}{c}))}{\exp(\gamma\times\ln n)}$, *i.e.* $\Pr[Z_i > \ln n] \leq \frac{\exp((\exp(\gamma)-1)\times(\frac{\pi}{c}))}{\exp(\gamma\times\ln n)}$, for any $\gamma > 0$.

Proof: Since all the sensors are i.i.d., the number of the sensors in c_i satisfies the binomial distributions with parameters $(n, \frac{\pi}{A})$ [47]. Applying the Chernoff bound and for any $\gamma > 0$, we have

$$\Pr[Z_i > \ln n] \leq \frac{\mathrm{E}[\exp(\gamma Z_i)]}{\exp(\gamma \ln n)} = \frac{[1+(\exp(\gamma)-1)\frac{\pi}{A}]^n}{\exp(\gamma \ln n)}$$
$$\leq \frac{\exp((\exp(\gamma)-1)\frac{\pi}{A}\times n)}{\exp(\gamma \ln n)} \quad (by\ 1+x \leq e^x)$$
$$= \frac{\exp((\exp(\gamma)-1)\times(\frac{\pi}{c}))}{\exp(\gamma\times\ln n)} \quad (by\ A = cn) \quad \blacksquare$$

From Lemma 1, the probability that an unit circle contains more than $\ln n$ sensors is zero when $n \to \infty$. Hence, we can use $\ln n$ as the upper bound of the number of the sensors in an unit circle in our analysis. Then, we can get the following theorem which states the upper and lower bounds of the *p-norm* of the distributed Algorithm.

Theorem 1. The upper bound of the *p-norm* value in the distributed Algorithm is $M(\ln n - 1 - \frac{2\pi}{3\times c} - \frac{\sqrt{3}}{2\times c} - \frac{n-M}{M})^2$; The lower bound of the *p-norm* value in the distributed Algorithm is $M(\frac{\pi}{c} - \frac{n-M}{M})^2$.

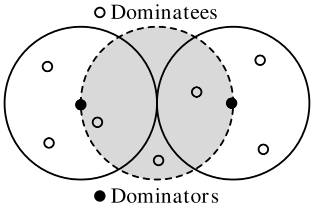


Figure 3.6. Connectivity guarantee.

Proof: According to Definition 3.2.3, the *p-norm* value of the allocate dominatees with EAP is formulated as: $|EAP|_p = \sum_{j=1}^{M}(|\sum_{i=1}^{|ND(s_j)|} EAP_{ij} - \bar{p}|)^2$. The *p-norm* value depends on how many dominatees are adjacent to each dominator, namely $ND(s_j)$ in the formula. So the upper bound and lower bound of the number of neighboring dominatees are the key challenges to analyze the performance ratio. The upper bound of the number of the sensors in a unit circle occurs when there is only one dominator in a unit circle and all the other diminatees connect to the dominator, then we can get the upper bound of the *p-norm* value. However, we are considering a CDS. In order to maintain the connectivity, at least two dominators must be within each other in the transmission range, namely in one unit circle. Fig. 3.6 illustrates the situation. There are some overlapped area shown by the gray area in the figure. The gray area is a sector with 120 degree and its size is: $\frac{2\pi}{3} - \frac{\sqrt{3}}{2}$. Because all the sensors are i.i.d., the expected number of the sensors is: $\frac{\frac{2\pi}{3}-\frac{\sqrt{3}}{2}}{c\times n} \times n = \frac{2\pi}{3\times c} - \frac{\sqrt{3}}{2\times c}$. Therefore, the upper bound of the *p-norm* value is:

$$\begin{aligned}|EAP|_p &\leq \sum_{j=1}^{M}(\ln n - 1 - \tfrac{2\pi}{3\times c} - \tfrac{\sqrt{3}}{2\times c} - \tfrac{n-M}{M})^2 \\ &= (\ln n - 1 - \tfrac{2\pi}{3\times c} - \tfrac{\sqrt{3}}{2\times c} - \tfrac{n-M}{M})^2\end{aligned}$$

The lower bound of the number of the sensors in each unit circle can be estimated by $\frac{\pi}{c\times n} \times n = \frac{\pi}{c}$. Hence the lower bound of the *p-norm* value is:

$$|EAP|_p \geq \sum_{j=1}^{M}(\tfrac{\pi}{c} - \tfrac{n-M}{M})^2 = M(\tfrac{\pi}{c} - \tfrac{n-M}{M})^2 \qquad \blacksquare$$

3.5 Simulation

In this section, we evaluate our proposed algorithms by comparing our work with the work in [44], in which each dominatee chooses the neighboring dominator of the smallest ID as its parent. Four different schemes are implemented:

- LBCDSs with LBAD, noted by LB-A.
- LBCDSs with the smallest ID dominator selection scheme, noted by LB-ID.
- MIS-based CDSs with LBAD, noted by MIS-A.
- MIS-based CDSs with the smallest ID dominator selection scheme, which is the work in [44], noted by MIS-ID.

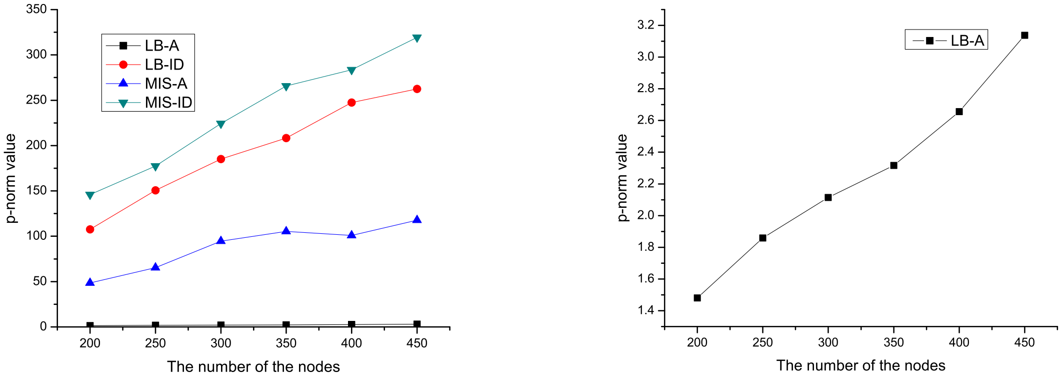
We compare them in terms of the *p-norm* value, network lifetime, which is defined as the time duration till the first dominator's energy is depleted.

3.5.1 Simulation Environment

We build our own simulator where all nodes have the same transmission range (10m). n nodes are randomly deployed in a fixed area of $100m \times 100m$. n is incremented from 200 to 450 by 100. For a certain n, 100 instances are generated. The results are averaged among 100 instances. Moreover, we use the CDS-based broadcasting as the communication mode.

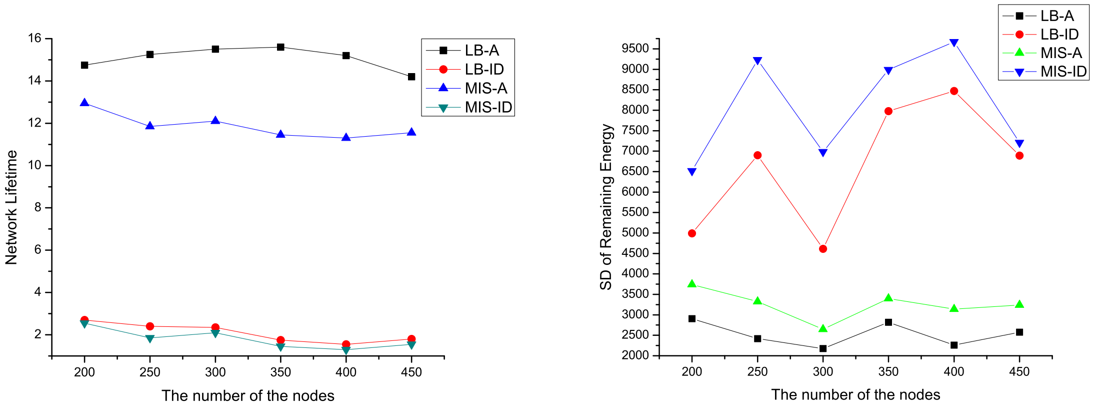
3.5.2 Simulation Results

Fig. 3.7 shows the *p-norm* values of the four schemes. With the increase of the number of the sensor nodes, the *p-norm* values increase correspondingly. This is because when the number of the nodes increases, we need more nodes to build an LBCDS. According to Definition 3.2.1, more nodes imply more sum subitems, so the *p-norm* values increase. As mentioned in Section 3.2.1, the smaller the *p-norm* value is, the more load-balanced the scheme is. From Fig. 3.7, we know that the MIS-ID scheme has the largest *p-norm* values while the LB-A scheme has the smallest *p-norm* values. This is because the MIS-ID scheme



(a) (b)

Figure 3.7. *p-Norm* value.



(a) (b)

Figure 3.8. Simulation Results: (a) Network Lifetime; (b) SD of Remaining Energy.

did not consider the load-balance factor when building a CDS and allocating dominatees to dominators. For clearly to see the *p-norm* values of the LB-A scheme, we redraw the curve using smaller scale in Fig. 3.7(b) for LB-A. Additionally, Fig. 3.7 demonstrates that the LBAD algorithm fits for any type of CDSs. The MIS-A scheme still has smaller *p-norm* values than the other two schemes using smallest ID allocation scheme, namely LB-ID and MIS-ID.

Fig. 3.8(a) shows the network lifetime of the four schemes. The simulated energy consumption model is that every node has the same initial 100 units energy. Receiving a packet consumes 1 unit energy, while transmitting a packet consumes 2 units energy. From Fig. 3.8(a), we can see the load-balanced schemes (LB-A and MIS-A) prolong network lifetime by 80% compared to non-balanced schemes (LB-ID and MIS-ID). With the number of the node increases, there is no obvious increase or decrease trend of network lifetime , since the locality of the network topology mainly decides the network lifetime. A network topology is generated randomly, so we cannot control the locality of the network. From Fig. 3.8(a), we also find the network lifetime of imbalanced schemes (LB-ID and MIS-ID) are close to 1, 2 or 3. This is because some critical smaller ID dominators are connected to many dominatees. They deplete energy very quickly, then the whole network is disconnected.

Fig. 3.8(b) shows the standard derivation of the remaining energy of the four schemes. The X-axis represents the number of the nodes. The Y-axis represents the standard derivation of the average remaining energy of all the nodes. We use the standard derivation here to observe whether the remaining energy is balanced or not. From Fig. 3.8(b), we know the balanced schemes (LB-A and MIS-A) have more balanced remaining energy than imbalanced schemes (LB-ID and MIS-ID). This is because we consider the load-balance factor when building a CDS and allocating dominatees to dominators.

The simulation results can be summarized as follows:

- The LB-A scheme always has the best performance according to the *p-norm* value, network lifetime and the standard derivation of remaining energy. The results demonstrate building a load-balanced CDS and then load-balancedly allocating dominatees can increase network lifetime significantly.

- The load-balanced dominatee allocation algorithm can be applied to not only load-balanced CDSs but also imbalanced CDSs to achieve good performances. The LB-A and MIS-A schemes have better performances over the LB-ID and MIS-ID schemes among all measures, namely the *p-norm* value, network lifetime and the standard derivation of remaining energy.
- The balanced schemes (LB-A and MIS-A) have better scalability than the imbalanced schemes (LB-ID and MIS-ID).

3.6 Summary

In this chapter, we propose a new LBCDS concept, which is a CDS with the minimum *p-norm* value in order to assure that the workload among each dominator is balanced. We also propose an LBAD problem. It aims to load-balancedly allocate each dominatee to a dominator. We use EAP value to represent the expected probability of the allocation between each dominatee and dominator pair. An optimal centralized algorithm and an efficient distributed algorithm for the LBAD problem are proposed in the chapter. The lower bound and upper bound of the approximation ratio is proved in the chapter. The extensive simulation results demonstrate that compared to Wan's work [44], using an LBCDS as a virtual backbone and EAP values to load-balancedly allocate dominatees can prolong network lifetime significantly.

CHAPTER 4

GENETIC-ALGORITHM-BASED CONSTRUCTION OF LOAD-BALANCED CDSS IN WIRELESS SENSOR NETWORKS

4.1 Motivation

As mentioned in Section 3.1, all the related works did not consider the **load-balance** factor when they construct a CDS. If the workloads on each dominator in a CDS are not balanced, some heavy-duty dominators deplete their energy quickly. Then, the whole network might be disconnected. Hence, intuitively, we not only have to consider to construct an MCDS, but also need to consider to construct a load-balanced CDS (LBCDS). An illustration of an LBCDS is depicted in Fig. 4.1, in which dominators are marked as black nodes, while white nodes represent dominatees; solid lines represent that the dominatees are allocated to the neighboring dominators, while the dashed lines represent the communication links in the original graph. For convenience, the set of neighboring dominatees of a dominator $v_i \in \mathbb{B}$ is denoted by $\mathbb{U}(v_i)$. The set of dominatees allocated to a dominator v_i is denoted by $\mathbb{A}(v_i) = \{v_j | v_j \in \mathbb{W}, v_j \text{ forwards its data only to } v_i\}$. According to the traditional MCDS construction algorithms, a CDS $\{v_4, v_7\}$ with size 2 is obtained for the network shown in Fig. 4.1(a). However, There are two severe drawbacks of the CDS shown in Fig. 4.1(a). First, $\mathbb{U}(v_4) = \{v_1, v_2, v_3, v_5, v_6\}$, which represents that dominator v_4 connects to 5 different dominatees, and $\mathbb{U}(v_7) = \{v_6, v_8\}$. If every dominatee has the same amount of data to be transferred through the neighboring dominator at a fixed data rate, dominator v_4 must deplete its energy much faster than dominator v_7, since dominator v_4 has to forward the data collected from 5 neighboring dominatees. Second, dominatee v_6 connects to both dominators. If v_6 is allocated to dominator v_4, shown in Fig. 4.1(a), obviously, only one dominatee v_8 forwards its data to dominator v_7. In this situation, the workload imbalance in the CDS is further amplified. Consequently, the entire network lifetime is shortened. We show a counter-example in Fig. 4.1(b), where the constructed CDS is $\{v_3, v_6, v_7\}$. According to the topology shown in Fig. 4.1(b), we can get the dominatee sets of each dominator: $\mathbb{U}(v_3) = \{v_1, v_2, v_4\}$, $\mathbb{U}(v_6) = \{v_4, v_5\}$, and $\mathbb{U}(v_7) = \{v_4, v_8\}$. Compared with the MCDS constructed in Fig. 4.1(a), the numbers of neighboring dominatees of all the dominators in Fig. 4.1(b) are very similar. On the other hand, we have two different dominatee allocation schemes shown in Fig. 4.1(b) and Fig. 4.1(c) respectively. One is: $\mathbb{A}(v_3) = \{v_1, v_2, v_4\}$, $\mathbb{A}(v_6) = \{v_5\}$, and $\mathbb{A}(v_7) = \{v_8\}$. The other one is: $\mathbb{A}(v_3) = \{v_1, v_2\}$, $\mathbb{A}(v_6) = \{v_4, v_5\}$, and $\mathbb{A}(v_7) = \{v_8\}$. Apparently, the workload on each dominator is almost evenly distributed in the CDS constructed in Fig. 4.1(c). Intuitively, the construction algorithm and dominatee allocation scheme shown in Fig. 4.1(c) can extend network lifetime notably. Obviously, constructing an LBCDS and then load-balancedly allocate dominatees to dominators are equally important when considering the load-balance factor to construct a CDS. Neither of these two aspects can be ignored.

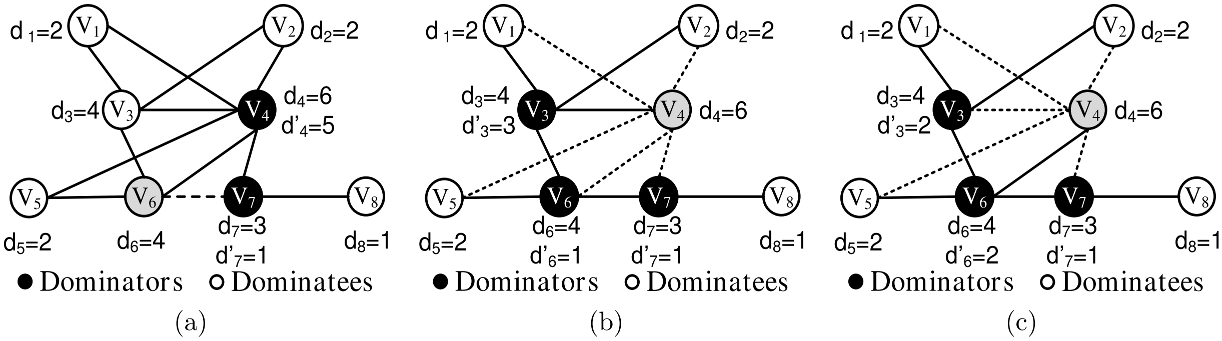


Figure 4.1. Illustration of a regular CDS and an LBCDS.

To solve the workload imbalance problem of an MCDS, in this chapter, we investigate how to construct an LBCDS and how to load-balancedly allocate dominatees to dominators simultaneously. To address this problem, we explore the Genetic Algorithm (GA) optimization approach. GAs are numerical search tools which operate according to the procedures that resemble the principles of nature selection and genetics [61]. Because of their flexibility and widespread applicability, GAs have been successfully used in a wide variety of problems in several areas of WSNs [28], and [29].

4.2 LBCDS Problem Definition

In this section, we give an overview of the LBCDS problem. We first present the assumptions and introduce the network model. Then, we give the problem definition. Finally, we point out the key issues and main challenges we are facing when solving the problem.

4.2.1 Network Model

We assume a static WSN and all the nodes in the WSN have the same transmission range. Hence, we model a WSN as an undirected graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is the set of n sensor nodes, denoted by $v_1, v_2, \ldots, v_n$; $\mathbb{E}$ represents the link set, $\forall\ u, v \in \mathbb{V}$, there exists an link (u, v) in $\mathbb{E}$ if and only if u and v are in each other's transmission range. In this chapter, we assume edges are undirected (bidirectional), which means two linked nodes are able to transmit and receive data from each other.

4.2.2 Terminologies

The *load-balance* factor is our major concern in this work. Thus, finding an appropriate measurement to evaluate load-balance is the key to solve the LBCDS problem. We use *p-norm* (Definition 3.2.1) to measure load-balance.

In this chapter, we use node degree, denoted by d_i, as the feature vector Θ in Equation 3.1 to measure the load-balance of a CDS, since the degree of each node is a potential indicator of traffic load. Thus, the definition of *CDS p-norm* is given as follows:

Definition 4.2.1. *CDS p-norm ($|\mathbb{B}|_p$).* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$, and a CDS $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$. The *CDS p-norm* of an $m \times 1$ vector $\mathbb{D} = (d_1, d_2, \cdots, d_m)$ is:

$$|\mathbb{B}|_p = (\sum_{i=1}^{m} |d_i - \bar{d}|^p)^{\frac{1}{p}} \tag{4.1}$$

where m is the number of dominators in the set $\mathbb{B}$, d_i represents the node degree of each dominator in the set $\mathbb{B}$, and $\bar{d}$ is the mean degree of $\mathbb{G}$.

We use the WSN shown in Fig. 4.1 to illustrate how to calculate the *CDS p-norm*. For simplicity, we use $p = 2$ in this chapter. Without specific explanation, p and 2 are interchangeable in this chapter. Two different CDSs for the same network are identified in Fig. 4.1. The degree of node v_i is denoted by d_i in Fig. 4.1. From the topology shown in Fig. 4.1, we can get $\bar{d} = 3$. Therefore, the *CDS p-norm* of the CDS shown in Fig. 4.1(a) is $\sqrt{9}$. Similarly, in Fig. 4.1(b), the *CDS p-norm* value is $\sqrt{2}$. Clearly, $\sqrt{2} < \sqrt{9}$, which implies that the CDS in Fig. 4.1(b) is more load-balanced than the CDS in Fig. 4.1(a).

When constructing an LBCDS, it is considerably important to allocate dominatees to each dominator. In a traditional/naive way, such as the work in [44], each dominatee is allocated to the neighboring dominator with the smallest ID. Obviously, the load-balance factor is not taken into account. In some environment, the dominator with the smallest ID, which is chosen by majority dominatees, tends to have heavier workload than the other dominators. Therefore, neither node ID nor node degree can reflect workload precisely. In a WSN with a CDS as the VB, only the dominator and dominatee links contribute to workload. Based on this observation, we define the following concepts:

Definition 4.2.2. *Dominatee Allocation Scheme ($\mathscr{A}$).* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$ and a CDS $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$, we need to find m disjoint sets on $\mathbb{V}$, *i.e*, $\mathbb{A}(v_1), \mathbb{A}(v_2), \cdots, \mathbb{A}(v_m)$, such that:

1. Each set $\mathbb{A}(v_i)$ $(1 \leq i \leq m)$ contains exactly one dominator v_i.

2. $\bigcup_{i=1}^{m} \mathbb{A}(v_i) = \mathbb{V}$, and $\mathbb{A}(v_i) \bigcap \mathbb{A}(v_j) = \emptyset$ $(1 \leq i \neq j \leq m)$.

3. $\forall v_u \in \mathbb{A}(v_i)$ $(1 \leq i \leq m)$ and $v_u \neq v_i$, such that $(v_u, v_i) \in \mathbb{E}$.

A *Dominatee Allocation Scheme* is:

$$\mathscr{A} = \{\mathbb{A}(v_i) \mid \forall v_i \in \mathbb{B}, 1 \leq i \leq m\} \tag{4.2}$$

Definition 4.2.3. *Valid Degree (d').* The $Valid\ Degree$ of dominator v_i is the number of its allocated dominatees, *i.e.*, $\forall v_i \in \mathbb{B}, d'_i = |\mathbb{A}(v_i)|$, where $|\mathbb{A}(v_i)|$ represents the number of dominatees in the set $\mathbb{A}(v_i)$.

In this chapter, we use the *Allocation Scheme p-norm* to measure the load-balance of different dominatee allocation schemes, in which, the *Valid Degree* of each dominator is used as the information vector Θ. The definition of the *Allocation Scheme p-norm* is given as follows:

Definition 4.2.4. *Allocation Scheme p-norm ($|\mathscr{A}|_p$).* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$, a CDS $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$, and a dominatee allocation scheme $\mathscr{A}$, the *Allocation Scheme p-norm* is:

$$|\mathscr{A}|_p = (\sum_{i=1}^{m} |d'_i - \mathcal{E}|^p)^{\frac{1}{p}} \tag{4.3}$$

where d'_i represents the valid degree of each dominator in the set $\mathbb{B}$, and $\mathcal{E} = \frac{n-m}{m}$ is the expected allocated dominatees on each dominator.

Fig. 4.1(b) and Fig. 4.1(c) illustrate an imbalanced and a balanced dominatee allocation scheme respectively. The *valid Degree* of dominator v_i is denoted by d'_i in Fig. 4.1. From the topology shown in Fig. 4.1 (b) and (c), we can get $\mathcal{E} = \frac{5}{3}$. Therefore, the *Allocation Scheme p-norm* of the dominatee allocation scheme shown in Fig. 4.1(b) is $\sqrt{2.67}$. Similarly, in Fig. 4.1(c), the *Allocation Scheme p-norm* is $\sqrt{0.67}$. Clearly, $\sqrt{0.67} < \sqrt{2.67}$, which implies the dominatee allocation scheme shown in Fig. 4.1(c) is more load-balanced than the scheme shown in Fig. 4.1(b). The result further confirms the observation we mentioned in Section 4.1.

4.2.3 Definition of LBCDS

Definition 4.2.5. *Load-balanced CDS (LBCDS) Problem.* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$, the *LBCDS* problem is to find a minimum-sized node set $\mathbb{B} \subseteq \mathbb{V}$ and a dominatee allocation scheme $\mathscr{A}$, such that:

1. $\mathbb{G}[\mathbb{B}] = (\mathbb{B}, \mathbb{E}')$*, where* $\mathbb{E}' = \{e|\ e = (u, v), u \in \mathbb{B}, v \in \mathbb{B}, (u, v) \in \mathbb{E})\}$*, is connected.*

2. $\forall u \in \mathbb{V}$ *and* $u \notin \mathbb{B}$*,* $\exists v \in \mathbb{B}$*, such that* $(u, v) \in \mathbb{E}$*.*

3. $min\{|\mathbb{B}|_p, |\mathscr{A}|_p\}$.

We claim that the LBCDS problem is NP-Hard, since it still belongs to the MCDS problem. Based on Definition 4.2.5, the key issue of the LBCDS problem is to seek a tradeoff between the minimum-sized CDS, the load-balance of a constructed CDS, and a dominatee allocation scheme. GAs are population-based search algorithms, which simulate biological evolution processes and have successfully solved a wide range of NP-Hard optimization problems [28, 29]. Additionally, GAs have shown themselves to be very good at discovering good solutions with a reasonable amount of time and computation effort. In the following, a novel GA algorithm, named LBCDS-GA, is proposed to solve the LBCDS problem.

4.3 LBCDS-GA Algorithm

In the following sections, we first provide some basics of the GA optimization approach, and then present the detailed design of the RMCDS-GA algorithm for the RMCDS problem.

4.3.1 Genetic Algorithm (GA) Overview

GAs are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. In nature, over many generations, natural populations evolve according to the principles of natural selection and *survival of the fittest.* By mimicking this process, GAs work with a population of *chromosomes*, each representing a possible solution to a given problem. Each chromosome is assigned a *fitness score* according to how good a solution to the problem it is. The highly fittest chromosomes are given opportunities to *reproduce*, by *crossover* with other chromosomes in the population. This produces new chromosomes as *offsprings*, which share some features taken from each *parent.* The least fittest chromosomes of the population are less likely to be selected for reproduction, and so they *die out.* A whole new population of possible solutions is thus produced by selecting the best chromosomes from the current *generation*, and mating them to produce a new set of chromosomes. This new generation contains a higher proportion of the characteristics possessed by the good chromosomes of the previous generation. In this way, over many generations, good characteristics are spread throughout the population. If the GA has been designed well, the population will converge to an optimal solution to the problem. In the following part of this section, we will design and explain LBCDS-GA step by step.

4.3.2 Representation of Chromosomes

A chromosome is a possible solution of the LBCDS problem. Hence, when designing the encoding scheme of chromosomes, we need to identify dominators and dominatees in a chromosome and a dominatee allocation scheme in a chromosome as well. For convenience, the set of neighboring dominators of each dominatee $v_s \in \mathbb{W}$ is denoted by $\mathbb{H}(v_s) = \{v_r | v_r \in \mathbb{B}, (v_r, v_s) \in \mathbb{E}\}$. In the proposed LBCDS-GA, each node is mapped to a gene in the chromosome. A gene value g_i indicates whether the sensor represented by this gene is a dominator or not. If the sensor is a dominatee, the corresponding gene value represents the allocated dominator. Hence, a generation of chromosomes with gene values is denoted as: $\mathbb{C}^g = \{C_j^g \mid 1 \leq j \leq k, C_j^g = (g_1, g_2, \cdots, g_i, \cdots, g_n)\}$, where k is the number of the chromosomes in each generation of population, and for $1 \leq i \leq n$,

$$g_i = \begin{cases} 1, v_i \in \mathbb{B}. \\ \forall v_t \in \mathbb{H}(v_i), v_i \in \mathbb{W}. \end{cases}$$

Additionally, beyond the aforementioned gene value, there is a meta-gene value G_i to store $\mathbb{H}(v_s)$, $\forall v_s \in \mathbb{W}$. Thus, a generation of chromosomes with meta-gene values is denoted as: $\mathbb{C}^G = \{C_j^G \mid 1 \leq j \leq k, C_j^G = (G_1, G_2, \cdots, G_i, \cdots, G_n)\}$, and for $1 \leq i \leq n$,

$$G_i = \begin{cases} 1, v_i \in \mathbb{B}. \\ \mathbb{H}(v_i), v_i \in \mathbb{W}. \end{cases}$$

Through the above description we know, as long as choosing a specific node from each node set $\mathbb{H}(v_i), \forall v_i \in \mathbb{W}$, we can easily generate C_j^g from C_j^G. Additionally, all the nodes with $g_i/G_i = 1$ form a CDS $\mathbb{B} = \{v_i \mid g_i/G_i = 1, 1 \leq i \leq n\}$. An example WSN is shown in Fig.4.1(c) to illustrate the encoding scheme. There are 8 nodes and the CDS is $\mathbb{B} = \{v_3, v_6, v_7\}$. Moreover, according to the topology shown in Fig. 4.1, $\forall v_i \in \mathbb{W}$, we can get $\mathbb{H}(v_i)$ easily. Thus, the 8 nodes can be encoded using 8 meta-genes in a chromosome, *e.g.*, $C^G = (\{v_3\}, \{v_3\}, 1, \{v_3, v_6, v_7\}, \{v_6\}, 1, 1, \{v_7\})$ shown in Fig. 4.2. Based on the dominatee allocation scheme shown in Fig. 4.1(c), *i.e.*, dominatee v_4 is allocated to dominator v_6, the chromosome with 8 genes is $C^g = (\{v_3\}, \{v_3\}, 1, \{v_6\}, \{v_6\}, 1, 1, \{v_7\})$. In conclusion, C_j^G stores all neighboring dominators of each dominatee, while the corresponding C_j^g records one CDS and one specific dominatee allocation scheme.

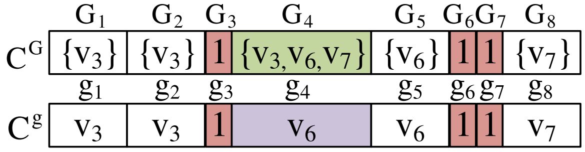


Figure 4.2. A chromosome with meta-genes and genes.

4.3.3 Population Initialization

GAs differ from most optimization techniques because of their global searching effectuated by one population of solutions rather than from one single solution. Hence, a GA search starts with the creation of the first generation, *i.e.*, a population with k chromosomes denoted by P_1. This step is called population initialization. A general method to initialize the population is to explore the genetic diversity. That is, for each chromosome, all dominators are randomly generated. However, the dominators must form a CDS. Therefore we start to create the first chromosome C_1 by running an existing MCDS method, *e.g.*, the latest MCDS construction algorithm [44], and then generate the population with k chromosomes by modifying C_1. We call the procedure, generating the whole population by modifying one specific chromosome, Inheritance Population Initialization (IPI). The IPI algorithm is summarized as follows:

If the number of the generated chromosomes in P_1 is less than k, run the following steps till k non-duplicated chromosomes are generated.

1) Let $t = 1$.

2) In the CDS $\mathbb{B}_t$ represented by the chromosome C_t, start from node v_u with the smallest ID (ID used here is only to create a sequence for generating new chromosomes. Any other features who can rank the nodes also can be applied here.) in $\mathbb{B}_t$, and add one neighboring dominatee by the order of its ID into the CDS each time. *i.e.*, $\mathbb{B}_s = \mathbb{B}_t \bigcup \{v_i \mid \forall v_i \in \mathbb{U}(v_u), v_i$ has the smallest ID among the nodes in the set $\mathbb{U}(v_u)\}$. Encoding $\mathbb{B}_s$ as a chromosome, and add it into P_1. Repeat the procedure, till all nodes in the set $\mathbb{U}(v_u)$ are added into $\mathbb{B}$. For example, in Fig.4.1(a), the CDS $\mathbb{B}_1 = \{v_4, v_7\}$ of the shown WSN is given. Thus, the encoded chromosome with meta-genes is $C_1^G = (\{v_4\}, \{v_4\}, \{v_4\}, 1, \{v_4\}, \{v_4, v_7\}, 1, \{v_7\})$. The node with the smallest ID is v_4 in $\mathbb{B}_1$. Therefore, the chromosomes from C_2^G to C_6^G are generated by adding one node from the set $\mathbb{U}(v_4) = \{v_1, v_2, v_3, v_5, v_6\}$ each time.

3) Move to the node with the second smallest ID in CDS $\mathbb{B}_t$, doing the same procedure as described in step 2), till every node in $\mathbb{B}_t$ are checked. As shown in Fig. 4.1(a), $\mathbb{U}(v_7) = \{v_6, v_8\}$. By eliminating the duplicates, the chromosome C_7^G is created by adding v_8.

4) If all the dominators in the current $\mathbb{B}_t$ are checked, move to the next CDS by setting $t = t + 1$, and repeat steps from 2) to 4).

Since each dominatee has two choices: to change to a dominator or to remain as a dominatee, consequently, there are $2^{n-|\mathbb{B}|}$ possible ways to create new chromosomes. Usually, k is much smaller than $2^{n-|\mathbb{B}|}$. Hence, the first population P_1 can be easily generated.

There are several merits that need to be pointed out here when using the IPI algorithm to generate P_1. First, we can guarantee that each chromosome in P_1 is a feasible solution (*i.e.*, a CDS) of the LBCDS problem. Second, the critical nodes (cut nodes) are chosen to be dominators. When reproducing new offsprings, the critical nodes are still dominators in the new chromosomes in the successive generations, which can help for guaranteeing the connectivity of a CDS.

4.3.4 Fitness Function

Given a solution, its quality should be accurately evaluated by the fitness score, which is determined by the fitness function. In our algorithm, we aim to find a minimum-sized CDS $\mathbb{B}$ with minimum $|\mathbb{B}|_p$ and $|\mathscr{A}|_p$ values. Therefore, the fitness function of a chromosome C_i^g is defined as:

$$\begin{cases} f(C_i^g) = \dfrac{n - |\mathbb{B}|}{w_1|\mathbb{B}|_p + w_2|\mathscr{A}|_p} \\ w_1 + w_2 = 1,\ 0 < w_1, w_2 < 1 \end{cases} \tag{4.4}$$

The purpose of doing a linear combination of $|\mathbb{B}|_p$ and $|\mathscr{A}|_p$ values in Equation 5.1 is that a user can change the weight of $|\mathbb{B}|_p$ and $|\mathscr{A}|_p$ values dynamically and easily. The denominator in Equation 5.1 needs to be minimized (the smaller the *p-norm* value, the more load-balanced the interested feature vector), while the numerator needs to be maximized (since we seek an MCDS). As a result, the fitness function value needs to be maximized.

4.3.5 Selection Scheme

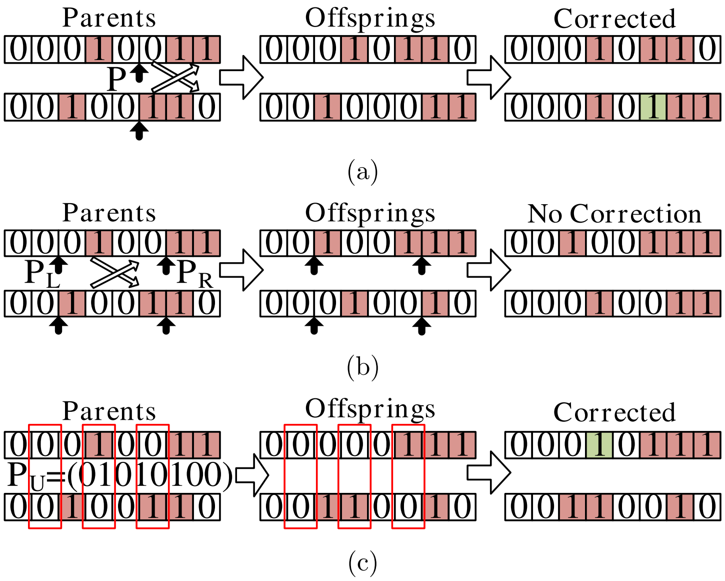
During the evolutionary process, election plays an important role in improving the average quality of the population by passing the high quality chromosomes to the next generation. Therefore, the selection operator needs to be carefully formulated to ensure that better chromosomes (higher fitness scores) of the population have a greater probability of being selected for mating, but that worse chromosomes of the population still have a small probability of being selected. Having some probability of choosing worse members is important to ensure that the search process is global and does not simply converge to the nearest local optimum. We adopt *Rank Selection* (RS) to select parent chromosomes. In order to prevent very fit chromosomes from gaining dominance early at the expense of less fit ones, which would reduce the population's genetic diversity, we set the rank value of each chromosome to be $R_i = log(1 + f(C_i^g))$. Thus, RS stochastically selects chromosomes based on R_i. A real-valued interval, S, is determined as the sum of the chromosomes' expected selection probabilities $P_i = \frac{R_i}{\sum_{j=1}^{k} R_j}$, thus, $S = \sum\limits_{i=1}^{k} P_i$. Chromosomes are then mapped one-to-one into contiguous intervals in the range $[0, S]$. To select a chromosome, a random number is generated in the interval $[0, S]$ and the chromosome whose segment spans the random number is selected. This process is repeated until a desired number of chromosomes have been selected.

4.3.6 Genetic Operations

The performance of a GA relies heavily on two basic genetic operators, *crossover* and *mutation*. Crossover exchanges parts of the current solutions (the parent chromosomes selected by the RS scheme) in order to find better ones. Mutation flips the values of genes, which helps a GA keep away from local optimum. The type and implementation of these two operators depend on the encoding scheme and also on the application. In the LBCDS problem, we can adopt classical operations, however, the new obtained solutions may not be valid (the dominator set represented by the chromosome is not a CDS) after implementing the crossover and mutation operations. Therefore, a correction mechanism needs to be preformed to guarantee the validity of all the new generated offspring solutions.

Crossover The purpose of crossover operations is to produce more valid CDSs represented by the new generated chromosomes. At this stage, we do not need to care dominatee allocations. Therefore, when performing crossover operations, we can logically assume all gene values of dominatees are 0, *i.e.*, $g_i = 0, \forall v_i \in \mathbb{W}$. After the new CDS is created, we can easily fill in all meta-gene values based on its original topology.

In the LBCDS-GA algorithm, we adopt three crossover operators called single-point crossover, two-point crossover, and uniform crossover respectively. With a crossover probability p_c, each time we use the RS scheme to select two chromosomes C_i^g and C_j^g as parents to perform one of the three crossover operators randomly. We use Fig. 4.3 to illustrate the three crossover operations. Suppose that two parent chromosomes (00010011) and (00100110) are selected by the RS scheme from the population. By the single-point crossover (shown in Fig.4.3(a)), the genes from the randomly generated crossover point $P = 6$ to the end of



(a)

(b)

(c)

Figure 4.3. Illustration of Crossover Operations: (a) single-point crossover; (b) two-point crossover; (c) uniform crossover.

the two chromosomes exchange with each other to get (00010110) and (00010111). After crossing, the first offspring (00010110) is a valid solution. However, the other one (00100011) is not valid, thus we need to perform the *correction mechanism*. The mechanism starts with scanning each gene on the offspring chromosome, denoted by C_o^g, till the end of the chromosome. If the value of the current scanned gene is 0, *i.e.*, $g_i = 0$ and the gene value is different from the original chromosome, denoted by C_s^g, without doing crossover and mutation operations, then change the gene value to 1. Whenever the DS represented by the corrected chromosome is a CDS, stop the mechanism. Otherwise, keep repeating the process till the end of C_o^g is reached. The idea behind the correction mechanism is that the DS represented by C_s^g is a CDS. If C_o^g is not valid, then add the dominators represented by C_s^g into the DS represented by C_o^g one by one. Finally, the corrected chromosome must be valid. For example, for the specific invalid offspring chromosome (00100011), when scanning the gene at position P, *i.e.* $g_6 = 0$, we find the value of g_6 is different after crossing. Therefore, we correct it by setting $g_6 = 1$. Then the corrected chromosome (00010111) is now a valid solution. Consequently, the correction mechanism stops and we get two valid offspring chromosomes (00010110) and (00010111). The correction mechanism is the same for crossover and mutation operations.

By the two-point crossover (shown in Fig.4.3(b)), the two crossover points are randomly generated which are $P_L = 3$ and $P_R = 6$; and then the genes between P_L and P_R of the two parent chromosomes are exchanged with each other. The two offsprings are (00100111) and (00010010) respectively. Since both of the offspring chromosomes are valid, we do not need to do any correction.

For the uniform crossover (shown in Fig.4.3(c)), a vector of uniform crossover P_U is randomly generated, which is $P_U = (01010100)$, indicating that g_2, g_4, and g_6 of the two parent chromosomes exchange with each other. Hence the two offsprings are (00000111) and (00110010). Since the first offspring is not a valid solution, we need to perform the correction mechanism mentioned before, and the corrected chromosome becomes to (00110010), which is a valid solution.

Gene Mutation The population undergoes the gene mutation operation after the crossover operation is performed. With a mutation probability p_m, we scan each gene g_i on the offspring chromosomes. If the mutation operation needs to be implemented, the value of the gene flips, *i.e.* 0 becomes to 1, and 1 becomes to 0. The correction mechanism mentioned before needs to be preformed if the mutated chromosomes are not valid.

4.3.7 Meta-gene Mutation

Differed from traditional GAs, in LBCDS-GA, we perform an additional operation named meta-gene mutation on k chromosomes in each generation. As mentioned before, the purpose of crossover operations is to produce more valid CDSs represented by the new offspring chromosomes. Moreover, the gene mutation operation after the crossover operation helps a GA keep away from local optimum. In summary, The aforementioned crossover and gene mutation operations only provide the chance to increase diversity of possible CDSs, however, till now nothing is aimed to create the diversity of dominatee allocation schemes.

In fact, to address the LBCDS problem, we need to find a load-balanced CDS and load-balancedly allocate dominatees to dominators. Therefore, Meta-gene mutation is proposed in LBCDS-GA to generate more possible dominatee allocation schemes.

As known, as long as choosing a specific node from each node set $\mathbb{H}(v_i), \forall v_i \in \mathbb{W}$, we can easily generate C_j^g from C_j^G. Thus, the procedure to determine gene values from meta-gene values is the procedure to specify a dominatee allocation scheme. According to the observation, we design the following described meta-gene mutation. The original population without doing crossover and gene mutation operations will undergo the meta-gene mutation operation. If the number of neighboring dominators of a dominatee v_i is greater than 1, *i.e.*, $|\mathbb{H}(v_i)| \geq 2$, then randomly pick a node from the set $\mathbb{H}(v_i)$ with a probability p_i. For example, the CDS shown in Fig. 4.1(b), and (c) is encoded as the chromosome with meta-genes $(\{v_3\}, \{v_3\}, 1, \{v_3, v_6, v_7\}, \{v_6\}, 1, 1, \{v_7\})$, which is shown in Fig. 4.2. Since $G_4 = \mathbb{H}(v_4) = \{v_3, v_6, v_7\}$, which means $|\mathbb{H}(v_4)| \geq 2$. We then randomly pick one dominator from the set $\mathbb{H}(v_4)$ with a probability p_i. If v_3 is selected from $\mathbb{H}(v_4)$, it means dominatee v_4 is allocated to dominator v_3. The dominatee allocation scheme is shown in Fig. 4.1(b), encoding as the chromosome with genes $(\{v_3\}, \{v_3\}, 1, \{v_3\}, \{v_6\}, 1, 1, \{v_7\})$. Similarly, if dominatee v_4 is allocated to dominator v_6, the dominatee allocation scheme is shown in Fig. 4.1(c), encoding as the chromosome with genes $(\{v_3\}, \{v_3\}, 1, \{v_6\}, \{v_6\}, 1, 1, \{v_7\})$.

To easily understand the traditional gene mutation and our proposed meta-gene mutation on chromosomes, we conclude the differences as follows:

1) The gene mutation operation is bit-wised, while the meta-gene mutation is performed at some positions i satisfying the condition $|\mathbb{H}(v_i)| \geq 2$, and $v_i \in \mathbb{W}$.

2) The gene mutation flips the logic gene values, *i.e.*, 0 becomes to 1, and 1 becomes to 0. In contrast, the meta-gene mutation only flips meta-gene values at some specific positions i, *i.e.*, randomly pick one node from the set $G_i = \mathbb{H}(v_i)$.

3) The purpose of gene mutation is to create diversity of all possible CDSs, while the purpose of meta-gene mutation is to provide more different dominatee allocation schemes. Since constructing a load-balanced CDS and load-balancedly allocating diminatees to domi-

nators are two critical challenges to solve the LBCDS problem, neither of the gene mutations and meta-gene mutations can be ignored in LBCDS-GA.

4.3.8 Replacement Policy

The last step of LBCDS-GA is to create a new population using an appropriate replacement policy. From crossover and gene mutation operations, we can get k offspring chromosomes. In addition, we can get another k chromosomes from the meta-gene mutation operation. In LBCDS-GA, we utilize the best k chromosomes (*i.e.*, the chromosome with the highest fitness score) among those $2k$ chromosomes to generate a new population. However, when creating new population by crossover, gene mutation, and meta-gene mutation, there is a chance to lose the fittest chromosome. Therefore, an elitism strategy, in which the best chromosome (or a few best chromosomes) is retained in the next generation's population, is used to avoid losing the best candidates. The LBCDS-GA stops and returns the current fittest solution until the number of total generations K is reached or the best fitness

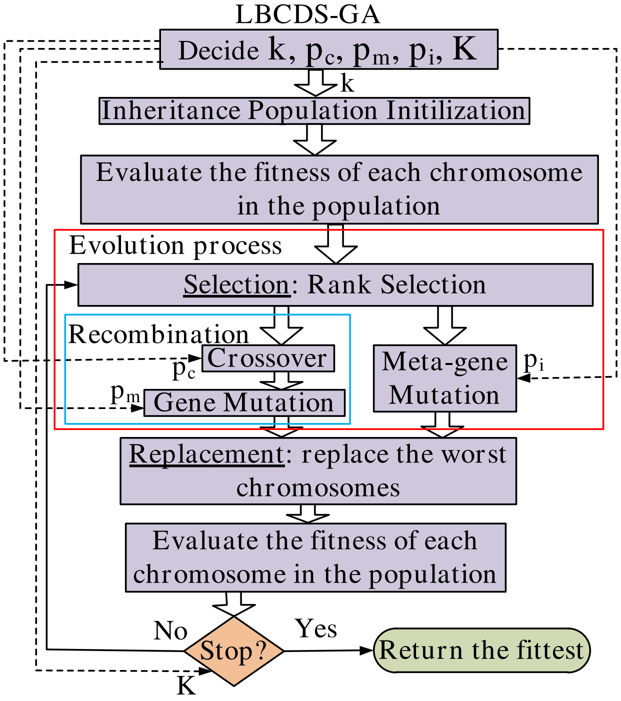

LBCDS-GA
Decide k, p_c, p_m, p_i, K
k
Inheritance Population Initilization
Evaluate the fitness of each chromosome in the population
Evolution process
Selection: Rank Selection
Recombination
Crossover
p_c
p_m
Gene Mutation
Meta-gene Mutation
p_i
Replacement: replace the worst chromosomes
Evaluate the fitness of each chromosome in the population
No
Stop?
Yes
Return the fittest
K

Figure 4.4. Procedure of LBCDS-GA

score does not change for continuous l generations. Figure 5.2 shows the flow chart of the LBCDS-GA algorithm.

4.4 Performance Evaluation

In the simulations, the results of LBCDS-GA are compared with the MCDS construction algorithm in [44] denoted by MIS, which is the latest and best MIS-based CDS construction algorithm. We compare the two algorithms in terms of the size of the constructed CDS, *CDS p-norm*, *Allocation Scheme p-norm*, the fitness score, network lifetime (which is defined as the time duration until the first dominator runs out of energy), and the average remaining energy over the whole network.

4.4.1 Simulation Environment

We build our own simulator where all nodes have the same transmission range of $50m$ and all nodes are deployed uniformly and randomly in a square area of $300m \times 300m$. n is incremented from 100 to 1000 by 100. For a certain n, 100 instances are generated. The results are averaged over 100 instances. Moreover, we use the CDS-based data aggregation as the communication mode. The simulated energy consumption model is that every node has the same initial 1000 unit energy. Receiving and transmitting a packet both consume 1 unit energy. Additionally, the particular GA rules and control parameters are listed in Table 5.2.

Table 4.1. GA Parameters and Rules

Population size (k)	50
Number of total generations (K)	100
Selection scheme	Rank Selection
Replacement policy	Elitism
Crossover probability (p_c)	1
Gene mutation probability (p_m)	0.2
Meta-gene mutation probability (p_i)	1

4.4.2 Simulation Results and Analysis

In Fig. 4.5, the X-axis represents the number of the sensor nodes n, while the Y-axis represents the evaluated factors, *i.e.*, the size of the constructed CDS $|\mathbb{B}|$, *CDS p-norm* $|\mathbb{B}|_p$, *Allocation Scheme p-norm* $|\mathscr{A}|_p$, the fitness score f, network lifetime T, and the average remaining energy E over the whole network respectively.

Fig. 4.5(a) shows the number of dominators $|\mathbb{B}|$ of the constructed CDSs by using LBCDS-GA and MIS. From Fig. 4.5(a), we can see that, with the increase of the number of the sensor nodes n, $|\mathbb{B}|$ almost keeps stable for the MIS scheme. This is because MIS aims to find a minimum-sized CDS and the area covered by the deployed sensors does not change. On the other hand, for LBCDS-GA, $|\mathbb{B}|$ increases when n increses. This is because the objective of LBCDS-GA is to balance energy consumption on each dominator. If more nodes are chosen as dominators, the energy consumption can be distributed to more components. More importantly, the more dominators, the more possibilities to create diversity of dominatee allocation schemes. Therefore, the load-balanced objective can be achieved.

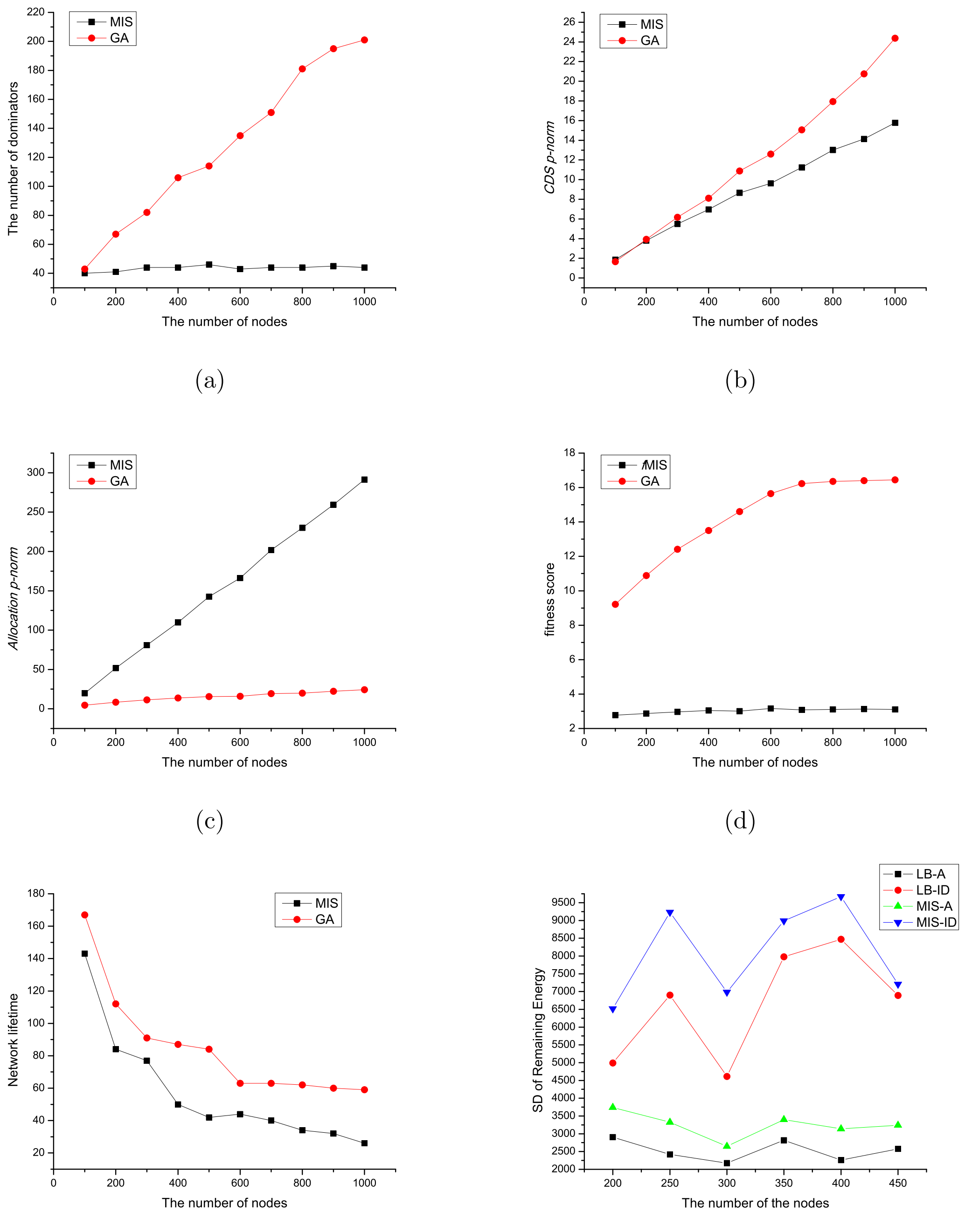
Fig. 4.5(b) shows the *CDS p-norm* $|\mathbb{B}|_p$ values of the constructed CDSs by using LBCDS-GA and MIS. With the increase of n, $|\mathbb{B}|_p$ increases correspondingly for both schemes. This is because when n increases, the area covered by the deployed sensors does not change. Therefore, the density of sensors increases, which means the degree of each dominator increases correspondingly. According to Definition 4.2.1, larger degrees of dominators imply larger subitem, thus $|\mathbb{B}|_p$ of both shemes increase. Moreover, the $|\mathbb{B}|_p$ of LBCDS-GA is larger than that of MIS. This is because we need more nodes to build an LBCDS (shown in Fig. 4.5(a)). More dominators imply more sum subitems based on Definition 4.2.1, thus, the $|\mathbb{B}|_p$ of LBCDS-GA is larger.

Fig. 4.5(c) shows the *Allocation Scheme p-norm* $|\mathscr{A}|_p$ values of the constructed CDSs by using LBCDS-GA and MIS. As mentioned before, the smaller the $|\mathscr{A}|_p$ value, the more load-balanced the dominatee allocation scheme $\mathscr{A}$. With the increase of n, $|\mathscr{A}|_p$ increases quickly for the MIS scheme. This is because, in MIS, dominatees are always allocated to the dominator with the smallest ID. The results also imply that $\mathscr{A}$ of MIS becomes more and more imbalance when n is getting larger. Nevertheless, for LBCDS-GA, $|\mathscr{A}|_p$ keeps almost the same, which means no matter how large the size of the set $\mathbb{B}$ is, LBCDS-GA always can find a load-balanced $\mathscr{A}$. Additionally, with the increase of n, the difference of $|\mathscr{A}|_p$ values between the two schemes becomes more and more obvious. This indicates LBCDS-GA becomes more and more effective to find an LBCDS in large scale WSNs.

Fig. 4.5(d) shows fitness scores f of the constructed CDSs by using LBCDS-GA and MIS. As mentioned before, the higher the fitness score is, the better quality the solution has. From Fig. 4.5(d), we can see that, with the increase of n, f does not change too much for MIS. However, for LBCDS-GA, f increases quickly. The results imply LBCDS-GA can find a more load-balanced CDS than MIS. This is because the MIS scheme does not consider the load-balance factor when building a CDS and allocating dominatees to dominators. Additionally, it is apparent that LBCDS-GA has more benefits when n becomes large.

Fig. 4.5(e) shows network lifetime of the two schemes. From Fig. 4.5(e), we know that the network lifetime decreases for both schemes with n increasing, since the WSN becomes denser and denser. Additionally, we can see LBCDS-GA prolongs network lifetime by 65% on average compared with MIS. In some extreme cases, such as $n = 1000$, network lifetime is extended by 100% compared with MIS. The result demonstrates that constructing an LBCDS and load-balancedly allocating dominatees to dominators can improve network lifetime significantly.

Fig. 4.5(f) shows the average remaining energy E over the whole network of the two schemes. With the increase of n, E increases for both schemes. As the WSN becomes denser and denser, a lot of redundant sensors exist in the WSN. From Fig. 4.5(f), we know that LBCDS-GA has less remaining energy than MIS. This is because LBCDS-GA considers the load-balance factor when building a CDS and allocating dominatees to dominators. Thus, the lifetime of the whole network is extended, which means the remaining energy of each node is less than MIS. This also indicates that constructing an LBCDS can balance the energy comsumption on each sensor node, making the lifetime of the whole network prolonged



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.5. Simulation results: (a) fitness score; (b) *CDS p-norm*; (c) *Allocation Scheme p-norm*; (d) the number of dominators; (e) network lifetime; (f) average remaining energy.

considerably.

4.5 Summary

In this chapter, we propose a novel concept — load-balanced CDS (LBCDS), which is an MCDS with the minimum $|\mathbb{B}|_p$ and $|\mathscr{A}|_p$ values in order to assure that the workload among each dominator is balanced and load-balancedly allocate dominatees to each dominator. We claim that constructing an LBCDS is an NP-Hard problem and propose an effective algorithm named LBCDS-GA to address the problem. The extensive simulation results demonstrate that using an LBCDS as a virtual backbone can balance the energy consumption among dominators. Consequently network lifetime is extended significantly. Particularly, when the node number changes from 100 to 1000, our proposed method prolong network lifetime by 65% on average compared with the latest MCDS construction algorithm [44].

CHAPTER 5

A GENETIC ALGORITHM WITH IMMIGRANTS SCHEMES FOR CONSTRUCTING A RELIABLE MCDS IN STOCHASTIC WIRELESS SENSOR NETWORKS

5.1 Motivation

WSNs are usually modeled using the *Deterministic Network Model* (DNM). Under this model, there is a transmission radius of each node. According to this radius, any specific pair of nodes are always connected to be neighbors if their physical distance is less than this radius, while the rest of the pairs are always disconnected. The Unit Disk Graph (UDG) model is a special case of the DNM model if all nodes have the same transmission radius. When all nodes are connected to each other, via a single-hop or multi-hop path, a WSN is said to have full connectivity. In most real applications, however, the DNM model cannot fully characterize the behavior of wireless links. This is mainly due to the *transitional region phenomenon* which has been revealed by many empirical studies [1, 4, 67]. Beyond the "always connected" region, there is a *transitional region* where a pair of nodes are probabilistically connected. Such pairs of nodes are not fully connected but reachable via the so called *lossy links* [1, 4]. As reported in [1, 4], there are often much more lossy links than fully connected links in a WSN. Additionally, in a specific setup [5], more than 90% of the network links are *lossy links.* Therefore, their impact can hardly be neglected.

In order to well characterize a WSN with lossy links, we propose a new network model called the *Stochastic Network Model* (SNM). Under this model, in addition to transmission radius, there is a *Transmission Success Ratio* (TSR) associated with each link connecting a pair of nodes, which is used to indicate the probability that one node can successfully directly deliver a package to another. Obviously, the core issue under the SNM model is how to guarantee the node-to-node delivery ratio of all possible node pairs satisfying the user requirement, in other words, how to guarantee the Transmission Quality (TQ). For constructing a MCDS under the SNM model, we propose *CDS reliability* to measure its TQ. Given a SNM model, *CDS reliability* is defined as the minimum node-to-node delivery ratio between any pair of dominators. Thus, how to find a reliable MCDS under the SNM model is the major concern of this chapter. The objective is to seek a MCDS whose reliability satisfies a certain application-dependent threshold denoted by σ (*e.g.*, $\sigma = 80\%$). If $\sigma = 100\%$, finding a reliable MCDS under the SNM model is the same as the traditional MCDS problem under the DNM model. However, a traditional MCDS algorithm may not find a reliable MCDS under the SNM mode. A counter-example is depicted in Fig. 5.1. By the latest algorithm proposed in [44], a spanning tree rooted at a specified initiator is first constructed, and then Maximal Independent Sets (MISs) are identified layer by layer. Finally a set of connectors to connect the MISs is ascertained to form a CDS. According to the topology shown in Fig. 5.1, the constructed CDS by [44] using s_4 as the initiator is $D = \{s_4, s_7, s_8\}$, whose reliability is 0.1. If the threshold $\sigma = 0.7$, the CDS D does not satisfy the constraint at all. The objective of our work is to find a MCDS whose reliability is greater than or equal to σ. One example of the satisfied reliable MCDS is $D' = \{s_3, s_6, s_7\}$ in Fig. 5.1. The key challenge finding a reliable MCDS under the SNM model is the computation of the *CDS reliability*. It is known that given a network topology, the calculation of the node-to-node delivery ratio is NP-Hard when network broadcast is used. Indeed, according to the reliability theory [68], the node-to-node delivery ratio is not practically computable unless

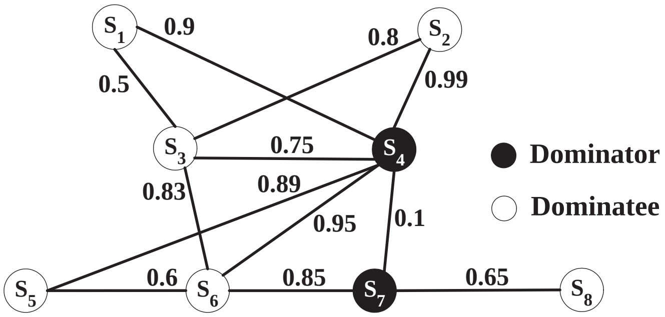


Figure 5.1. A WSN under the SNM model.

the network topology is basically series-parallel, namely, the graph representing a WSN can be reduced to a single edge by series and parallel replacements. Nevertheless, most network topologies are not series-parallel structures. Thus, instead of computing the accurate *CDS reliability*, we design a greedy based algorithm to approximate the *CDS reliability.* Another challenge is to find a minimum-sized CDS, which is also a NP-Hard problem [45]. Intuitively, the smaller the CDS is, the lower the reliability of the CDS is. The key issue then becomes how to find a proper trade-off between the minimum-sized CDS and the *CDS reliability* while satisfying the optimization constraint (*i.e.* the CDS reliability is no less than the threshold σ). To address this problem, we explore the Genetic Algorithm (GA) optimization approach. GAs are numerical search tools which operate according to the procedures that resemble the principles of nature selection and genetics [61]. Because of their flexibility and widespread applicability, GAs have been successfully used in a wide variety of problems in several areas of WSNs [26–29].

To the best of our knowledge, this work is the first one attempting to construct a MCDS under the SNM model for WSNs.

5.2 RMCDS Problem Statement

In this section, we give an overview of the reliable MCDS problem under the SNM model. We first present the assumptions, and then introduce the SNM model. Finally, we give the problem definition and make some remarks for the problem.

5.2.1 Assumptions

We assume a static WSN and all nodes in the WSN have the same transmission range. The Transmission Success Ratio (TSR) associated with each link connecting a pair of nodes is available, which can be obtained by periodic Hello messages, or be predicted using Link Quality Index (LQI) [69]. We also assume that the TSR values are fixed. This assumption is reasonable as many empirical studies have shown that LQI is pretty stable in a static environment [70]. Furthermore, no node failure is considered since it is equivalent to a link failure case. No duty cycle is considered either. We do not consider packet collisions or transmission congestion, which are left to the MAC layer. The degradation of the node-to-node delivery ratio is thus only due to the failure of wireless links.

5.2.2 Network Model

Under the *Stochastic Network Model* (SNM), we model a WSN as an undirected graph $G(V, E, P(E))$, where V is the set of n sensor nodes, denoted by $s_1, s_2, \ldots, s_n$; E is the set of m lossy links, $\forall\ u, v \in V$, there exists an edge (u, v) in G if and only if: 1) u and v are in each other's transmission range, 2) $TSR(e = \{u, v\}) > 0$, for each link $e = \{u, v\} \in E$, where $TSR(e)$ indicates the probability that node u can successfully directly deliver a packet to node v; and $P(E) = \{< e, TSR(e) > | e \in E, 0 \leq TSR(e) \leq 1\}$. We assume edges are undirected (bidirectional), which means two linked nodes are able to transmit and receive information from each other with the same TSR value.

Because of the introduction of $TSR(e)$, the traditional definition of the node neighborhood has changed. Hence, we first give the definition of the 1-hop neighborhood and then extend it to the r-Hop neighborhood.

Definition 5.2.1. 1-Hop Neighborhood. $\forall\ u \in V$, the 1-Hop Neighborhood of node u is defined as:

$$N_1(u) = \{v | v \in V, TSR(e = \{u, v\}) > 0\}$$

The physical meaning of 1-Hop Neighborhood is the set of the nodes that can be directly reached from node u.

Definition 5.2.2. r-Hop Neighborhood. $\forall\ u \in V$, the r-Hop Neighborhood of node u is defined as:

$$N_r(u) = N_{r-1}(u) \cup \{v | \exists w \in N_{r-1}(u), v \in N_1(w), v \notin \bigcup_{i=1}^{r-1} N_i(u)\}$$

The physical meaning of the r-Hop Neighborhood is that the set of the nodes that can be reached from node u by passing maximum r number of edges.

Definition 5.2.3. Node-to-Node Delivery Ratio. Given a source node u and a destination node v, one path between the node pair can be denoted by the edge permutation $\theta(u, v) = (e_1, e_2, \ldots, e_m)$, and the delivery ratio of the path is denoted by $DR_\theta = \prod_{i=1}^{m} e_i$. Furthermore, we use $\Theta(u, v)$ to denote the set of all the possible ways by which node v can be reached from node u. The Node-to-Node Delivery Ratio from node u to node v is then defined as:

$$DR^*(u, v) = max\{DR_\theta, \forall\ \theta(u, v) \in \Theta(u, v)\}$$

Clearly, $DR^*(u, v)$ is equivalent to $DR^*(v, u)$.

Definition 5.2.4. CDS Reliability. Given a WSN represented by $G(V, E, P(E))$ under the SNM model, and its CDS denoted by D, the reliability of D R^*_D is the minimum Node-to-Node Delivery Ratio between any pair of the nodes in the CDS, *i.e.*,

$$R^*_D = min\{DR^*(u, v), \forall\ u, v \in D, u \neq v\}$$

We use *CDS Reliability* to measure the quality of a CDS constructed under the SNM model. By this definition, when a CDS D has a reliability R^*_D satisfying a threshold σ (*i.e.* $R^*_D \geq \sigma$), we can state that for any pair of the nodes in the CDS the probability that they are connected is no less than the threshold.

According to the reliability theory [68], we know that the computation of the Node-to-Node Delivery Ratio is NP-Hard. Therefore, the computation of the CDS reliability is also NP-Hard. In summary, we claim that, given a WSN represented by $G(V, E, P(E))$ under the SNM model, a CDS for G denoted by D, and a pre-defined threshold $\sigma \in 0, 1]$, it is NP-Hard to verify whether $R^*_D \geq \sigma$.

Theorem 2. Given a WSN represented by $G(V, E, P(E))$ under the SNM model, a CDS for G denoted by D, and a pre-defined threshold $\sigma \in (0, 1]$, it is NP-Hard to verify whether

$R_D^* \geq \sigma$.

Proof: According to the reliability theory [68], we know that the computation of the Node-to-Node Delivery Ratio is NP-Hard. Therefore the computation of the CDS Reliability is also NP-Hard. □

5.2.3 Definition of RMCDS

After we introduce how to measure the quality of CDSs under the SNM model, we will give the formal definition of the problem we investigate in this chapter.

Definition 5.2.5. Reliable MCDS (RMCDS). Given a WSN represented by $G(V, E, P(E))$ under the SNM model, and a pre-defined threshold $\sigma \in (0, 1]$, the RMCDS problem is to find a minimum-sized node set $D \subseteq V$, such that

1. The induced graph $G[D] = (D, E')$, where $E' = \{e \mid e = (u, v), u \in D, v \in D, (u, v) \in E)\}$, is connected.
2. $\forall\ u \in V$ and $u \notin D$, $\exists\ v \in D$, such that $(u, v) \in E$.
3. $R_D^* \geq \sigma$.

We claim that the problem to construct a RMCDS for a WSN under the SNM model is NP-Hard. It is easy to see that the traditional MCDS problem under the DNM model is a special case of the RMCDS problem. By setting the TSR values on all edges to 1, we are able to convert the RMCDS problem to the traditional MCDS problem under the DNM model. Thus the RMCDS problem belongs to NP. The verification of the RMCDS problem needs to calculate the CDS Reliability. It is an NP-Hard problem, which is mentioned in Subsection 5.2.2. Therefore, the problem to construct a RMCDS for a WSN under the SNM model is NP-Hard.

Theorem 3. The problem to construct an RMCDS for a WSN under the SNM model is NP-Hard.

Proof: It is easy to see that the traditional MCDS problem under the DNM model is a special case of the RMCDS problem. By setting the TSR values on all edges to 1, we are able to convert the RMCDS problem to the traditional MCDS problem under the DNM model. Thus the RMCDS problem belongs to NP. The verification of the RMCDS problem needs to calculate the CDS Reliability. It is an NP-Hard problem, which is proved in Theorem 2. Therefore, the problem to construct an RMCDS for a WSN under the SNM model is NP-Hard. □

5.2.4 Remarks

As we already know, computing the Node-to-Node Delivery Ratio and the CDS reliability are NP-Hard problems. Therefore, instead of computing the accurate Node-to-Node Delivery Ratio, we design a greedy based algorithm to approximate the ratio denoted by $DR(u,v)$. Based on the approximate Node-to-Node Delivery Ratio, we then calculate the approximate *CDS Reliability* denoted by R_D. When there is no confusion, $DR^*(u,v)$ and $DR(u,v)$, R_D^* and R_D are interchangeable in the chapter.

Based on Definition 5.2.5, the key issue of the RMCDS problem is to seek a tradeoff between the minimum-sized CDS and the CDS reliability. GAs are population-based search algorithms, which simulate biological evolution processes and have successfully solved a wide range of NP-Hard optimization problems [26–29]. In the following, algorithm RMCDS-GA is proposed to solve the RMCDS problem to search the feasible domain more effectively and reduce the computation time.

5.3 RMCDS-GA Algorithm

In the following sections, we first provide some basics of the GA optimization approach, and then present the detailed design of the RMCDS-GA algorithm for the RMCDS problem.

5.3.1 GA Overview

GAs, first formalized as an optimization method by Holland [71], are search tools modeled after the genetic evolution of natural species. GAs encode a potential solution to a vector of independent variables, called chromosomes. The independent variables consisting of chromosomes are called genes. Each gene encodes one component of the target problem. A binary coding is widely used nowadays. GAs differ from most optimization techniques because of their global searching effectuated by one population of solutions rather than from one single solution. Hence, a GA search starts with the creation of the first generation, a random initial population of chromosomes, *i.e.*, potential solutions to the problem. Then, these individuals in the first generation are evaluated in terms of their "fitness" values, *i.e.*, their corresponding objective function values. Based on their fitness values, a ranking of the individuals in the first generation is dynamically updated. Subsequently, the first generation is allowed to evolve in successive generations through the following steps:

1. Reproduction: selection of a pair of individuals in the current generation as parents. The ranking of individual in the current generation is used in the selection procedure so that in the long run, the best individuals will have a greater probability of being selected as parents.

2. Recombination: crossover operation and mutation operation;

   (a) Crossover is performed with a crossover probability P_c by selecting a random gene along the length of the parent chromosomes and swapping all the genes of the selected parents chromosomes after that point. The operation generates two new children chromosomes.

   (b) Mutation is performed with a mutation probability P_m by flipping the value of one gene in the chromosomes (*e.g.*, 0 becomes 1, and 1 becomes 0, if binary coding is used).

3. Replacement: utilization of the fittest individual to replace the worst individual of

the current generation to create a new generation, so as to maintain the population number k a constant. Every time new children are generated by a GA, the fitness function is evaluated. And then a ranking of the individuals in the current generation is dynamically updated. The ranking is used in the replacement procedures to decide who, among the parents and the children chromosomes, should survive in the next population. This is to resemble the natural principles of the "survival of the fittest".

GAs usually stop when a certain number of total generations denoted by G are reached. Figure 5.2 shows the overview of the RMCDS-GA algorithm.

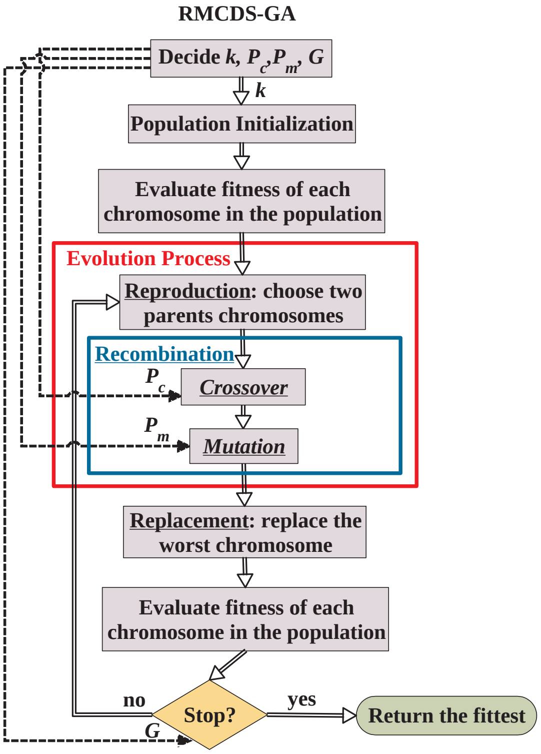

RMCDS-GA
Decide k, P_c, P_m, G
k
Population Initialization
Evaluate fitness of each chromosome in the population
Evolution Process
Reproduction: choose two parents chromosomes
Recombination
P_c
Crossover
P_m
Mutation
Replacement: replace the worst chromosome
Evaluate fitness of each chromosome in the population
no
Stop?
yes
G
Return the fittest

Figure 5.2. Procedure of RMCDS-GA

One important feature of GAs need to be emphasized here is that the optimization performance of GAs depends mainly on the convergence time of the algorithm. When using GAs, sufficient genetic diversity among solutions in the population should be guaranteed. Lack of such diversity would lead to a reduction of the search space spanned by the GA. Consequently, the GA may prematurely converge to a local minimum because mediocre individuals are selected in the final generation. Alternatively, an excess of genetic diversity, especially at later generations, may lead to a degradation of the optimization performance. In other words, excess genetic diversity may result in very late or even no convergence. In this chapter, genetic diversity is maintained by the crossover, mutation operations and immigrants schemes. In the following part of this section, we will explain RMCDS-GA step by step.

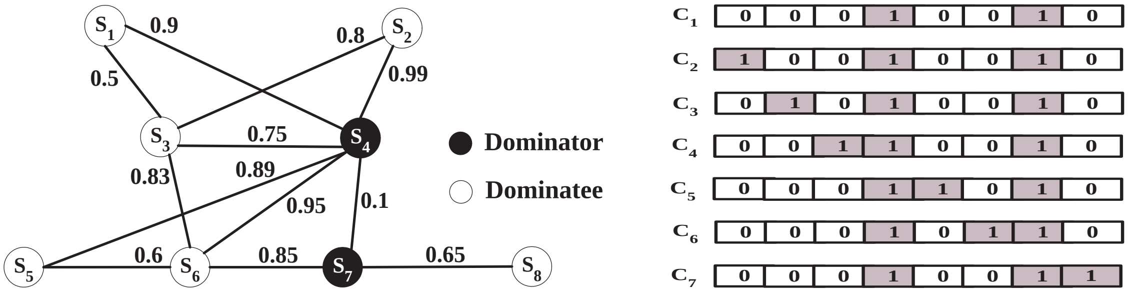
5.3.2 Representation of Chromosomes

In the proposed RMCDS-GA, each sensor is mapped to a gene in the chromosome. A gene value indicates whether the sensor represented by this gene is a dominator or not. Hence, a chromosome is denoted as: $C_i = (g_1, g_2, \cdots, g_j, \cdots, g_n)$, where $1 \leq i \leq k$ and k is the number of the chromosomes in the population; $1 \leq j \leq n$ and n is the total number of the sensors in a WSN.

$$\begin{cases} g_j = 1, \ \textit{node } s_j \textit{ is a dominator} \\ g_j = 0, \ \textit{node } s_j \textit{ is a dominatee} \end{cases}$$

All the sensors with $g_j = 1$ form a CDS denoted by $D = \{s_j | g_j = 1, 1 \leq j \leq n\}$.

An example WSN under the SNM model is shown in Fig.5.1 to illustrate the encoding scheme. There are 8 sensors and the CDS is $D = \{s_4, s_7\}$. Thus, the 8 sensors can be encoded using 8 genes in a chromosome, *e.g.* $C_1 = (g_1, g_2, \cdots, g_8)$, and then set the values of genes representing the dominators to 1. Finally, the encoded chromosome is $C_1 = (0, 0, 0, 1, 0, 0, 1, 0)$.



(a)

C_1	0	0	0	1	0	0	1	0
C_2	1	0	0	1	0	0	1	0
C_3	0	1	0	1	0	0	1	0
C_4	0	0	1	1	0	0	1	0
C_5	0	0	0	1	1	0	1	0
C_6	0	0	0	1	0	1	1	0
C_7	0	0	0	1	0	0	1	1

(b)

Figure 5.3. Illustration of Population of Initialization

5.3.3 Population Initialization

According to the flowchart of the proposed RMCDS-GA shown in Figure 5.2, after we decide the encoding scheme of the RMCDS problem, the first generation (a population with k chromosomes) should be created. This step is called population initialization in Figure 5.2. A general method to initialize the population is to explore the genetic diversity. That is, for each chromosome, all dominators are randomly generated. However, the dominators must form a CDS. Therefore we start to create the first chromosome by running an existing MCDS method, *e.g.*, Wan's work [44], and then generate the population with k chromosomes by modifying the first chromosome. We call the procedure, generating the whole population by modifying one specific chromosome, Inheritance Population Initialization (IPI) (see detail in 4.3.3).

An example is shown in Fig.5.1 to illustrate the IPI process. In Fig.5.1, the WSN and its CDS $D_1 = \{s_4, s_7\}$ are given. The values on the edges are TSR values and black nodes are dominators. Furthermore, we assume the CDS is constructed by a traditional MCDS method. According to the encoding scheme mentioned in subsection 5.3.2, $C_1 = (0, 0, 0, 1, 0, 0, 1, 0)$ represents the CDS generated by Wan's work [44] shown in Fig.5.1. Subsequently, we need to generate more chromosomes based on the first chromosome. The IPI algorithm is summarized as follows:

1. Start from the node with the smallest ID, reduce one dominator each time from the original CDS D_1 represented by C_1. If the new obtained node set is still a CDS D_i, then encode it as a chromosome C_i and add it into the initial population. Otherwise, remove the node with the second smallest ID from the original CDS D_1 and make the same checking process as for the node with the smallest ID. Repeating the process till no more new chromosome can be created. The CDS shown in Fig.5.1 is a minimum-sized CDS, *i.e.*, we cannot further reduce its size. Thus we go to step 2.

2. If the size of the original CDS D_1 cannot be reduced, and the number of the generated chromosomes is less than k, then for all the existing chromosomes $C_1, C_2, \cdots, C_i$ doing

the following steps till k non-duplicated chromosomes are generated.

(a) Let $t = 1$.

(b) In the CDS D_t represented by the chromosome C_t, start from node u with the smallest ID, and add one dominatee node in its 1-hop neighborhood $N_1(u)$ by the order of its ID into the CDS each time. If the new obtained node sets form CDSs, then encode them as chromosomes, and add them into the initial population. In Figure 5.3(b), the node with the smallest ID is s_4 in D. Therefore, the chromosomes from C_2 to C_6 are generated by adding one node from set $N_1(s_4) = \{s_1, s_2, s_3, s_5, s_6\}$ each time.

(c) Move to the node with the second smallest ID in CDS D_t till every node in D_t are checked. In Figure 5.3(b), the 1-Hop neighborhood of the node with the second smallest ID s_7 is $N_1(s_7) = \{s_6, s_8\}$. Since s_6 has already been marked as a dominator, we cannot add it to create a new CDS. By eliminating the duplicates, the chromosome C_7 is created.

(d) If all the dominators in the current D_t are checked, move to the next CDS by setting $t = t + 1$, repeat the step from 2b) to 2d).

Since each sensor has two choices: to be a dominator or a dominatee, consequently, there are $2^{n-|D|}$ possible ways to create new chromosomes, where $|D|$ is the size of the CDS denoted by D under the SNM model. Usually, k is much smaller than $2^{n-|D|}$. Hence the initial population $C_1, C_2, \cdots, C_k$ can be easily generated.

There are several merits that need to be pointed out here when using the above IPI algorithm to generate the initial population. First, we can guarantee every dominator set represented by a chromosome in the first generation is a CDS, *i.e.* each chromosome in the initial population is a feasible solution of the RMCDS problem. Second, the critical nodes (cut nodes), are dominators encoded in each chromosome of the initial population. When performing crossover operations, the critical nodes are still dominators in the new offspring chromosome in the successive generations. The illustration examples will be shown in Subsection 5.3.6. Finally, The IPI stops when k chromosomes are generated. Actually, we can obtain more valid solutions by continuously running the IPI algorithm. As we already know, the population diversity plays an important role on the optimization performance of GAs. Therefore, the extra valid solutions generated by keeping running the IPI algorithm can be used in the replacement process to bring more population diversity in new generations. We will give more detailed description of the replacement scheme in Section 5.4.

5.3.4 Fitness Function

Given a solution, its quality should be accurately evaluated by the fitness value, which is determined by the fitness function. In our algorithm, we aim to find a minimum-sized CDS D whose reliability R_D should be greater than or equal to a preset threshold σ. Therefore, the fitness function of a chromosome C_i in the population is defined as:

$$f(C_i) = \frac{R_D^2}{|D|^2} \tag{5.1}$$

The purpose of raising $|D|$ and R_D to the power of 2 in Equation 5.1 is to enlarge the weight of the size of the CDS D. The denominator in Equation 5.1 needs to be minimized while the numerator needs to be maximized. As a result, the fitness function value will be maximized.

As mentioned in the previous section, precisely calculating the CDS reliability is an NP-Hard problem. According to Definition 5.2.4, we can easily compute the CDS reliability based on the Node-to-Node Delivery Ratio of all possible dominator pairs in the CDS. Therefore, we propose a greedy based approximate algorithm to calculate the Node-to-Node Delivery Ratio. We adopt a greedy based routing protocol, Greedy Perimeter Stateless Routing (GPSR) [72], to find the path between all possible dominator pairs. In this work, we modified the greedy criterion to be the largest TSR values greater than or equal to σ based on GPSR, then we can guarantee that Node-to-Node Delivery Ratios between all possible dominator pairs are greater than or equal to σ.

For easier to understand, we first illustrate the idea by an example and then summarize the whole process. For the chromosome C_2 shown in Figure 5.3(b), the CDS represented by C_2 is $D = \{s_1, s_4, s_7\}$, in which there are three possible dominator pairs, i.e. $(s_1, s_4), (s_1, s_7), (s_4, s_7)$. Assume the reliability threshold is $\sigma = 60\%$. Clearly, the TSRs associated with the edges (s_1, s_4) and (s_4, s_7) are both greater than 60% in Figure 5.3, i.e. $TSR(e_1 = \{s_1, s_4\}) = 0.9$, and $TSR(e_2 = \{s_4, s_7\}) = 0.95$. According to the Definition 5.2.3, we know that $DR(s_1, s_4) = 0.9$ and $DR(s_4, s_7) = 0.95$ respectively. Therefore, the first greedy criterion comes out: the direct edges between sources and destinations with TSR values greater than δ have the highest priority to be chosen as the path between sources and destinations. For dominator pair (s_1, s_7), Obviously, there is no direct edge between them. Thus we need to find a multi-hop path between them. The search process starts from the destination s_7. The greedy criterion is based on the TSR values on the edges between s_7 and all its 1-hop neighborhood $N_1(s_7) = \{s_4, s_6, s_8\}$. Since $TSR(e_2 = \{s_4, s_7\}) = 0.95 > 0.6$ is the largest TSR values among all the nodes in $N_1(s_7)$, the edge $e_2 = \{s_4, s_7\}$ is chosen. Subsequently, we keep searching from s_4. Apparently, $TSR(e_3 = \{s_2, s_4\}) = 0.99 > 0.6$ is the highest TSR values on the edges from s_4 to all the nodes in $N_1(s_4)$. However, based on the direct edge greedy criterion, i.e. there is a direct edge between the source s_1 and the current search node s_4, therefore $e_1 = \{s_1, s_4\}$ is chosen. According to Definition 5.2.3, $\theta(s_1, s_7) = \{e_1, e_2\}$, $DR(s_1, s_7) = DR_\theta = \prod_{i=1}^{2} TSR(e_i) = 0.9 * 0.95 = 0.855$. Finally, based on Definition 5.2.4, we know $R(D) = min\{DR(s_1, s_4), DR(s_1, s_7), DR(s_4, s_7)\} = min\{0.9, 0.855, 0.95\} = 0.855$. The fitness of C_2 can then be calculated using Equation 5.1, $f(C_2) = \frac{0.855^2}{3^2} = 0.081225$.

5.3.5 Selection (Reproduction) Scheme

During the evolutionary process, election plays an important role in improving the average quality of the population by passing the high quality chromosomes to the next generation. The selection operator is carefully formulated to ensure that better chromosomes of the population (with higher fitness values) have a greater probability of being selected for mating, but that worse chromosomes of the population still have a small probability of being selected. Having some probability of choosing worse members is important to ensure that the search process is global and does not simply converge to the nearest local optimum. We adopt *Roulette Wheel Selection* (RWS) since it is simple and effective. RWS stochastically selects individuals based on their fitness values $f(C_i)$. A real-valued interval, S, is determined as the sum of the individuals' expected selection probabilities, *i.e.* $S = \sum_{i=1}^{k} P_i$, where $P_i = \frac{f(C_i)}{\sum_{j=1}^{k} f(C_j)}$. Individuals are then mapped one-to-one into contiguous intervals in the range [0, S]. The size of each individual interval corresponds to the fitness value of the associated individual. The circumference of the roulette wheel is the sum of all fitness values of the individuals. The fittest chromosome occupies the largest interval, whereas the least fit has correspondingly smaller interval within the roulette wheel. To select an individual, a random number is generated in the interval $[0, S]$ and the individual whose segment spans the random number is selected. This process is repeated until a desired number of individuals have been selected.

The pseudo-code is shown in Algorithm 4.

Algorithm 4 : Roulette Wheel Selection

```
Require: Population number $k$, each chromosome's fitness value $f(C_i)$.
$S = \sum_{i=1}^{k} f(C_i)$;
Generate random number $r$ from interval $(0, S)$;
Initialize $curS = 0$;
for $i = 1$ to $k$ do
    $curS += f(C_i)$;
    if $curS >= r$ then
        return $C_i$;
    end if
end for
```

We still use the WSN shown in Figure 5.3(a) to illustrate the RWAS scheme. The following Table 5.1 lists a sample population of 7 individuals (shown in Figure 5.3(b)). These individuals consist of 8 bit chromosomes. The fitness values are calculate by Equation 5.1. We can see from the table: C_1 is the fittest and C_7 is the weakest. Summing these fitness values we can apportion a percentage total of fitness. This gives the strongest individual a value of 35% and the weakest 6%. These percentage fitness values can then be used to configure the roulette wheel (shown in Figure 5.4). The number of times the roulette wheel is spun is equal to size of the population (i.e. k). As can be seen from the way the wheel is now divided, each time the wheel stops this gives the fitter individuals the greatest chance of being selected for the next generation and subsequent mating pool. According to the survival of the fittest in nature selection, individual $C_1 = (00010010)$ will become more prevalent in the general population because it is the fittest, and more apt to the environment we have put it in.

Table 5.1. Fitness of 7 chromosomes

No.	Chromosome	$f(C_i)$	% of total
C_1	00010010	$\frac{0.95^2}{2^2} = 0.226$	35
C_2	10010010	$\frac{0.855^2}{3^2} = 0.081$	12
C_3	01010010	$\frac{0.9405^2}{3^2} = 0.098$	15
C_4	00110010	$\frac{0.7125^2}{3^2} = 0.056$	9
C_5	000110101	$\frac{0.8455^2}{3^2} = 0.079$	12
C_6	00010110	$\frac{0.80725^2}{3^2} = 0.072$	11
C_7	00010011	$\frac{0.6175^2}{3^2} = 0.042$	6
Totals		0.654	100

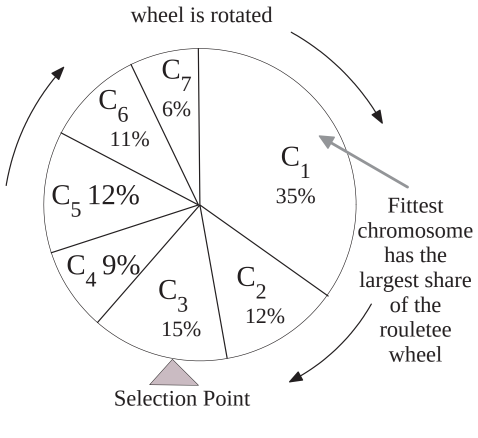


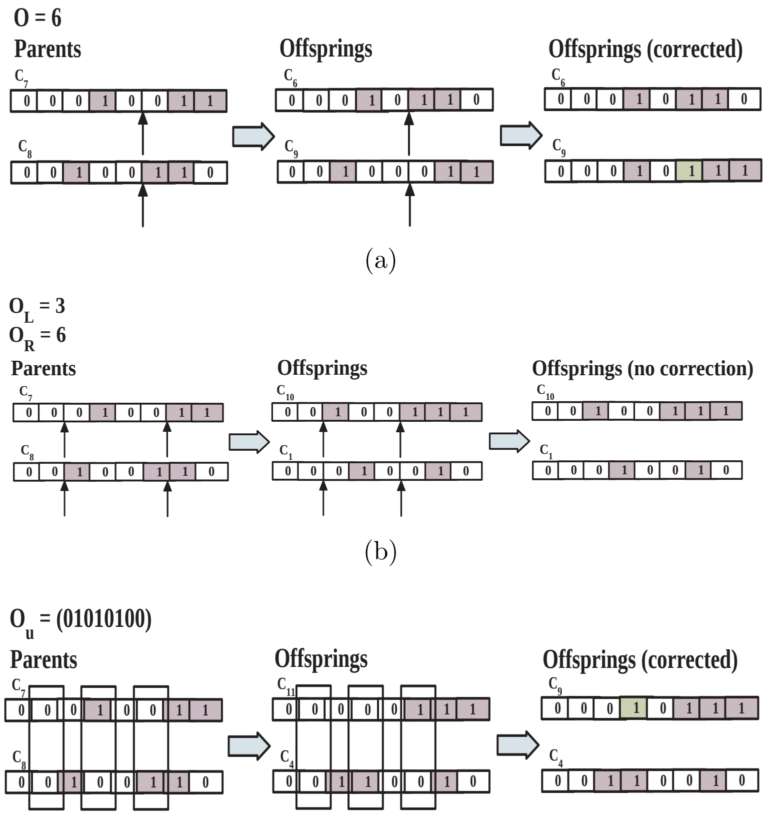
Figure 5.4. Roulette Wheel Selection

5.3.6 Genetic Operations

The performance of a GA relies heavily on two basic genetic operators, *crossover* and *mutation*. Crossover exchanges parts of the current solutions (the parent chromosomes selected by the RWS scheme) in order to find better ones. Mutation flips the values of genes, which helps a GA keep away from local optimum. The type and implementation of these two operators depend on the encoding scheme and also on the application. In the RMCDS problem, we use the binary coding scheme and all potential solutions must be CDSs. For crossover, we can adopt all classical operations, however, the new obtained solutions may not be valid (the dominator set represented by the chromosome is not a CDS) after implementing the crossover operations. Therefore, a correction mechanism needs to be preformed to guarantee validity of all the new generated solutions. Similarly, all traditional mutation operations can be adopted to the RMCDS problem, followed by a correction mechanism.

In this subsection, we introduce three crossover operators and their correction mechanism, followed by a mutation operator and its correction scheme.

Crossover In our algorithm, since a chromosome is expressed by binary codes, we adopt three crossover operators called single-point crossover, two-point crossover, and uniform crossover respectively. With a crossover probability P_c, each time we use the RWS scheme to select two chromosomes C_i and C_j as parents to perform one of the three crossover operators randomly. We use Fig.5.5 to illustrate the three crossover operations. Suppose that two parent chromosomes $C_7 = (00010011)$ and $C_8 = (00100110)$ are selected from the population. By the single-point crossover (shown in Fig.5.5(a)), the genes from the crossover point to the end of the two chromosomes exchange with each other to get $C_6 = (00010110)$ and $C_9 = (00010111)$. The crossover point denoted by $O = 6$ is generated randomly. After crossing, the first offspring $C_6 = (00010110)$ is a valid solution. However, the other one $C_9 = (00100011)$ is not valid, thus we need to perform the correction mechanism. The correction starts from the gene in the position of the crossover point O, *i.e.* g_6. Since g_6 is 1 in the parent chromosome C_8, it changes to 0 after crossing. We correct it by setting $g_6 = 1$.



(a)

(b)

(c)

Figure 5.5. Illustration of Crossover Operations: (a) single-point crossover; (b) two-point crossover; (c) uniform crossover.

Then $C_9 = (00010111)$ is now a valid solution. In general, we can keep correcting the genes till the end of the chromosome. By the two-point crossover (shown in Fig.5.5(b)), the two crossover points are randomly generated which are $O_L = 3$ and $O_R = 6$; and then the genes between O_L and O_R of the two parent chromosomes are exchanged with each other. The two offsprings are $C_{10} = (00100111)$ and $C_1 = (00010010)$ respectively. Since both of the offspring chromosomes are valid, we do not need to do any correction. As we already know, C_1 is the fittest in the population. This is a good illustration, we can obtain a fitter solution during the evolutionary process through genetic operations. For the uniform crossover (shown in Fig.5.5(c)), the vector of uniform crossover O_U is randomly generated which is $O_U = (01010100)$, indicating that g_2, g_4, and g_6 of the two parent chromosomes exchange with each other. Hence the two offsprings are $C_{11} = (00000111)$ and $C_4 = (00110010)$. Since C_{11} is not a valid solution, we need to perform the correction scheme, and the corrected chromosome becomes to $C_{10} = (00110010)$, which is a valid solution.

Mutation The population will undergo the mutation operation after the crossover operation is performed. With a mutation probability P_m, we scan each gene g_i on the parent chromosomes. If the mutation operation needs to be implemented, the value of the gene flips, *i.e.* 0 becomes to 1 and 1 becomes to 0.

An example shown in Figure 5.6, assume g_3 is mutated in chromosome C_7. The offspring $C_{11} = (00110011)$ is a valid solution, thus no correction needed. While g_6, g_8 are mutated in chromosome C_8, the offspring $C_{12} = (00100011)$ is not a valid solution. Therefore, we perform the similar correction mechanism mentioned in the crossover subsection to make the offspring C_{12} valid by correcting $g_6 = 1$.

Replacement Policy The last step of RMCDS-GA is to create a new population using an appropriate replacement policy. Usually, two chromosomes from the evolution process are utilized to replace the two worst chromosomes in the original population for generating a new population. However, when creating new population by crossover and mutation, we have a big chance to lose the fittest chromosome. Therefore, an elitism strategy, in which the best

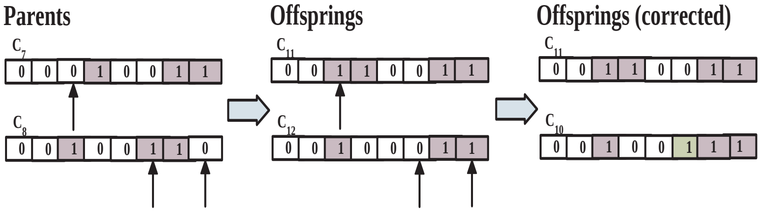


Figure 5.6. Illustration of Mutation operation

chromosome (or a few best chromosomes) is retained in the next generation's population, is used to avoid losing the best candidates.

The RMCDS-GA stops and returns the current fittest solution until the number of total generations G is achieved or the best fitness value does not change for continuous 10 generations. In the RMCDS-GA algorithm, we use G to stop the algorithm.

5.4 Genetic Algorithms with Immigrants Schemes

As mentioned in Section 5.3.1, the optimization performance of GAs depends mainly on the convergence time of the algorithm and appropriate population diversity may result in fast convergence time. In this section we investigate how the immigrants schemes affect the convergence time of the proposed RMCDS-GA algorithm.

In general, to converge at a proper pace is usually what we expect for GAs to find the optimal solutions for many optimization problems. However, for the RMCDS problem, the convergence becomes a challenge. GAs usually require to keep a certain population diversity level to maintain their adaptability. The crossover and mutation correction mechanisms in RMCDS-GA may reduce the population diversity. Thus it slows down the speed of convergence. To address this problem, the random immigrants approach is a quite natural and simple way [73–76], which is proposed with the inspiration from the flux of immigrants that wander in and out of a population between two generations in the nature. It maintains the diversity level of the population through replacing some individuals of the current population with random individuals, called random immigrants, in every generation. As to which individuals in the population should be replaced, usually there are two strategies: replacing random individuals or replacing the worst ones. In this chapter, GA with the random immigrants (GARI) uses the second replacement strategy, *i.e.*, utilize random immigrants to replace the worst individuals of the current population. The random immigrants can be obtained by keeping running the IPI algorithm or by randomly running another existing MCDS algorithm. In order to avoid significant disruption of the ongoing search progress by random immigrants, the ratio of the number of random immigrants to the population size denoted by P_{ri} is set to a small value, *e.g.*, $P_{ri} = 0.1$.

However, in some cases, random immigrants may not have any actual effect because individuals in the previous population may still be quite fit in the new population. In this case, random immigrants may thus degrade the performance. Based on the above consideration, GA with the elitism-based immigrants (GAEI), which uses elitism, *i.e.*, the best chromosome (or a few best chromosomes), to create immigrants and replace the worst individuals in the current population, is also used to address the RMCDS problem. The IPI algorithm can be performed to create immigrants from the elitism.

To further investigate the performance of GARI and GAEI, we propose the GA with hybrid immigrants (GAHI). In GAHI, in addition to the $P_{ri} \times k$ immigrants which are randomly created, $P_{ei} \times k$ immigrants are created from the elite of the previous generation, where P_{ei} is the ratio of the number of elitism-based immigrants to the population size. These two sets of immigrants will then replace the worst individuals in the current population.

The pseudo-code for GAEI and GAHI is shown in Algorithm 5.

5.5 Performance Evaluation

In the simulations, we implement the traditional GAs without immigrants and the three GAs with immigrants (GARI, GAEI, GAHI) to solve the RMCDS problem. These algorithms are compared with Wan's work [44] denoted by MIS, which is the latest and best MIS-based CDS construction algorithm.

Algorithm 5 : RMCDS-GA with Immigrants Schemes

```
Require: $k$, $P_{ri}$, $P_{ei}$, $G$.
$g = 0$; {$g$ represents the current generation number}
Initialize population $P(0)$ using IPI Algorithm;
while $g > G$ do
    Calculate the fitness of each chromosome in population $P(g)$;
    Select two parents chromosomes in $P(g)$ using RWS selection;
    Crossover with $P_c$
    Mutation with $P_m$
    Calculate the fitness of each chromosome in interim population $P^{'}(g)$
    {perform elitism-based immigrants}
    generate $P_{ei} \times k$ immigrants by modify $E(g-1)$; {$E(g-1)$ denotes the elite in $P(g-1)$}
    calculate the fitness of these immigrants;
    {perform hybrid immigrants}
    if GAHI is used then
        generate $P_{ri} \times k$ immigrants by modify $E(g-1)$;
        calculate the fitness of these immigrants;
    end if
    replace the worst individuals in $P^{'}(g)$ with the immigrants;
    $P(g+1) = P^{'}(g)$;
    g++;
end while
return  the fittest individual in population $P(G)$.
```

5.5.1 Simulation Environment

We build our own simulator where all nodes have the same transmission range ($10m$) and all nodes are deployed uniformly in a square area. Moreover, a random value between $[0.9, 0.98]$ is assigned to the TSR value associated to a pair of nodes inside the transmission range, otherwise, a random value between $(0, 0.8]$ is assigned to the TSR value associated to a pair of nodes beyond the transmission range. For a certain n, 100 instances are generated. The results are averaged among 100 instances. Additionally, the particular GA rules and control parameters are listed in Table 5.2.

Table 5.2. GA Parameters and Rules

Population size (k)	20
Number of total generations (G)	100
Selection scheme	Roulette Wheel Selection
Replacement policy	Elitism
Immigrants schemes	RI, EI, HI
P_{ri}	0.1
P_{ei}	0.1
Crossover probability (P_c)	1
Mutation probability (P_m)	0.001

5.5.2 Simulation Results

In Table 5.3, we show that traditional MCDS construction algorithms cannot solve the RMCDS problem under the SNM model, especially for large scale WSNs. In Table 5.3, we list the number of times that MIS and RMCDS-GA can find a CDS with a reliability greater than or equal to σ by running 100 simulations separately. σ is decreased from 0.6 to 0.4 by 0.1. From Table 5.3, we find that, with increasing n, the number of the times of satisfied CDSs for MIS and RMCDS-GA both decrease. This is because the sizes of CDSs increase which leads to a lower Node-to-Node Delivery Ratio. Moreover, RMCDS-GA can guarantee more satisfied CDSs than MIS, especially when $n \geq 200$. In other words, for large scale WSNs, it is hard to construct a satisfied CDS for MIS since the MIS algorithm does not consider reliability. Additionally, both MIS and RMCDS-GA can find more satisfied CDSs when σ decreases. In conclusion, traditional MCDS construction algorithms do not take reliability into consideration, while RMCDS-GA can find a satisfied reliable MCDS which is more practical in real environments.

In Table 5.4, R_{MIS} and R_{GA} represent the reliability of a CDS generated by MIS and RMCDS-GA, respectively. $|D_{MIS}|$ and $|D_{GA}|$ represent the size of the CDS constructed by MIS and RMCDS-GA, respectively. In Table 5.4, the reliability of CDSs decreases when the area size increases, since the number of the dominators increases. RMCDS-GA can

Table 5.3. MIS-based CDSs and RMCDS-GA generated CDSs

n	$\sigma = 0.6$		$\sigma = 0.5$		$\sigma = 0.4$	
	MIS	GA	MIS	GA	MIS	GA
50	100	100	100	100	100	100
80	94	100	100	100	100	100
120	57	100	98	100	100	100
160	21	100	90	100	100	100
200	5	96	44	100	88	100
250	2	91	12	93	56	100
400	1	90	4	17	10	100

guarantee to find a more reliable CDS than MIS, *i.e.*, $R_{GA} > R_{MIS}$. More importantly, the sizes of the CDSs obtained by MIS and RMCDS-GA are almost the same. On average, RMCDS-GA can find a CDS with 10% more reliability without increasing the size of a CDS than MIS. In summary, RMCDS-GA does not trade CDS size for CDS reliability.

Table 5.4. R & $|D|$ results of MIS and RMCDS-GA algorithms

Area (m^2)	n	R_{MIS}	R_{GA}	$\|D_{MIS}\|$	$\|D_{GA}\|$
40×40	50	0.65	0.77	17	18
50×50	80	0.59	0.72	24	26
60×60	120	0.51	0.68	33	33
70×70	160	0.46	0.62	40	44
80×80	200	0.44	0.58	51	51
90×90	250	0.39	0.53	63	62
100×100	400	0.32	0.49	78	78

5.6 Summary

In this chapter, we have investigated the RMCDS problem using a new network model called SNM. The SNM model is based on empirical studies show that most wireless links are lossy links which only probabilistically connect pairs of nodes. Different from the traditional DNM model which assumes that links are either connected or disconnected, the SNM model enable the employment of lossy links by introducing the TSR value on each lossy link. In this chapter we focus on constructing a minimum-sized CDS while its reliability satisfies a preset application-dependent threshold. We prove that RMCDS is an NP-Hard problem and propose a GA with immigrants schemes to address the problem. The simulation results show that compared to the traditional MCDS algorithm, RMCDS-GA can find a more reliable CDS without increasing the size of a CDS.

CHAPTER 6

THEORETIC ANALYSIS OF LOAD-BALANCED VIRTUAL BACKBONE CONSTRUCTION ALGORITHM FOR WIRELESS SENSOR NETWORKS

6.1 Introduction

As mentioned in Section 3.1, all the aforementioned works did not consider the *load-balance* factor when they construct a VB. For instance, when the MCDS-based VB is used in the network shown in Fig. 6.1(a), backbone node v_4 is adjacent to 5 different non backbone nodes, whereas, backbone node v_7 only connects to 2 non backbone nodes. If every non backbone node has the same amount of data to be transferred through the neighboring backbone node at a fixed data rate, then the number of neighboring non backbone nodes of each backbone node is a potential indicator of the traffic load on each backbone node. Hence, backbone nodes v_4 must deplete its energy much faster than backbone node v_7. A counter-example is shown in Fig. 6.1(b), the set $\{v_3, v_6, v_7\}$ is served as a VB. Compared with the VB constructed in Fig. 6.1(a), the numbers of neighboring non backbone nodes of all the backbone nodes in Fig. 6.1(b) are very similar. On the other hand, the criterion to allocate a non backbone node to a neighboring backbone node is also critical to balance traffic load on each backbone node. An illustration of the allocation schemes for non backbone nodes is depicted in Fig. 6.2, in which arrow lines represent that the non backbone nodes are allocated to the arrow pointed backbone nodes, while the dashed lines represent the communication links in the original network topological graph. Although the potential traffic load on each backbone node are evenly distributed in the VB constructed in Fig. 6.2 (as depicted in Fig. 6.1), different allocation schemes for non backbone nodes might break the balance. In Fig. 6.2, only the gray non backbone node v_4 is adjacent to more than one backbone node. Allocating v_4 to different backbone nodes leads to distinct traffic load on the allocated backbone node. In Fig. 6.2(a), v_4 is allocated to backbone node v_3, while in Fig. 6.2(b), v_4

is allocated to *backbone node* v_6. Apparently, backbone node v_3 has more traffic load than backbone nodes v_6 and v_7 in Fig. 6.2(a). However, traffic loads are balanced among backbone nodes in Fig. 6.2(b). Moreover, if the workloads on each backbone node are not balanced, some heavy-duty backbone nodes will deplete their energy quickly. Then, the whole network might be disconnected. Intuitively, compared with the WSN shown in Fig. 6.2(a), the VB and the allocation scheme for non backbone node v_4 shown in Fig. 6.2(b) can extend network lifetime notably. In summary, constructing a Load-Balanced VB (LBVB) and then load-balancedly allocate non backbone nodes to backbone nodes are equally important when considering the load-balance factor to form a VB in WSNs. Neither of these two aspects can be ignored.

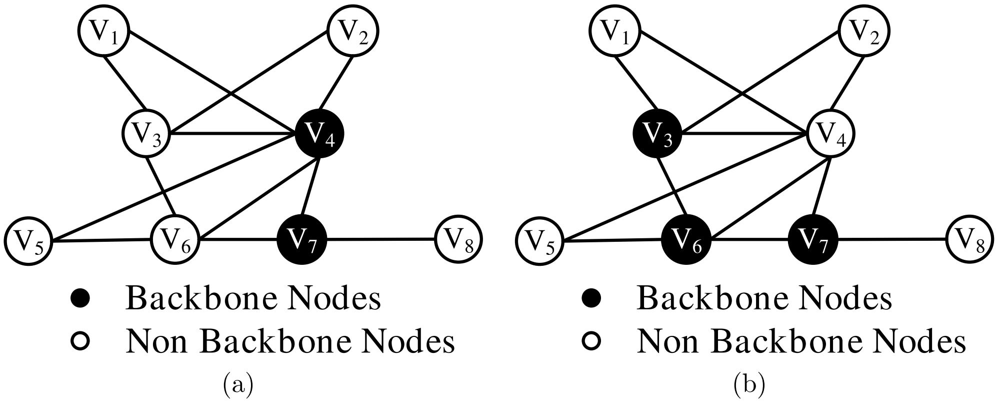


Figure 6.1. Illustration of a regular VB and a load balanced VB.

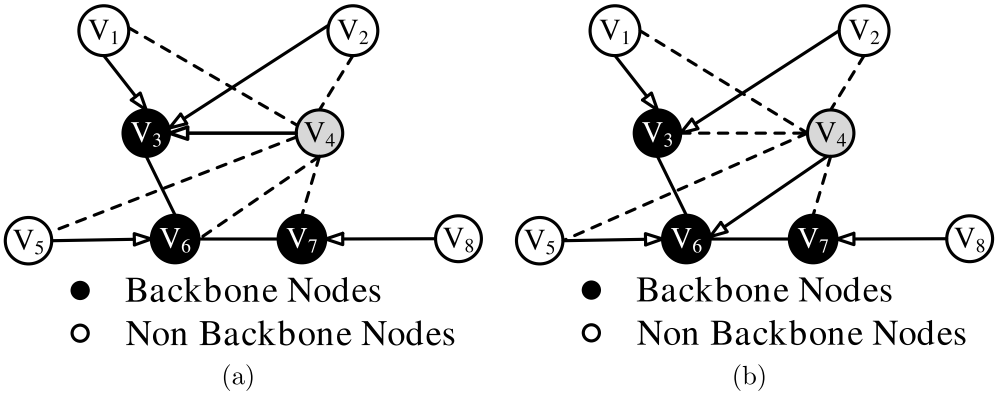


Figure 6.2. Illustration of a regular Allocation and a load balanced allocation.

To benefit from the CDS-based VB in WSNs and also take the load-balance factor into consideration, few attempts have been carried out to construct a VB in this manner [37]. In our previous work [37], we proposed a genetic-algorithm based method to build a load-balanced CDS (LBCDS) in WSNs. However, there is no performance ratio analysis in that chapter. In this research, we first investigate how to construct an LBVB. It is well known that in graph theory, a Maximal Independent Set (MIS) is also a DS. MIS can be defined formally as follows: given a graph $G = (V, E)$, an Independent Set (IS) is a subset $I \subset V$ such that for any two vertex $v_1, v_2 \in I$, they are not adjacent, *i.e.*, $(v_1, v_2) \notin E$. An IS is called an MIS if we add one more arbitrary node to this subset, the new subset will not be an IS any more. Therefore, we construct an LBVB with two steps. The first step is to find a MinMax Degree MIS (MDMIS), and the second step is to make this MIS connected. Subsequently, we explore how to load-balancedly allocate non backbone nodes to backbone nodes, followed by comprehensive performance ratio analysis.

Particularly, our contributions mainly include three aspects as follows:

1. We claim that the LBVB problem is an NP-Complete problem and therefore can not be solved in polynomial time unless P = NP. Hence, we solve the LBVB problem with two steps. First, we propose an approximation algorithm by using linear relaxation and random rounding techniques to solve the MinMax Degree Maximal Independent Set (MDMIS) problem. It is shown that this algorithm yields a solution upper bounded by $O(\Delta \ln(n))OPT_{MDMIS}$, where OPT_{MDMIS} is the optimal result of MDMIS, Δ is the maximum node degree in the network, and n is number of sensors in a WSN. Subsequently, the minimum-sized set of nodes are found to make the MDMIS connected. The theoretical upper bound of the size of the constructed LBVB is analyzed in this chapter as well.

2. We claim that the load-balancedly allocate non backbone nodes to backbone nodes problem is NP-Hard by formulating it as an equivalent binary programming. Consequently, we present a randomized approximation algorithm, which produces a solution in which the traffic load on each backbone node is upper bounded by

$O(\log^2(n))(OPT_{MVBA} + \frac{1}{\alpha^2})$ with probability $\frac{7}{8}$, where $\alpha = \log(n) + 3$, OPT_{MVBA} is the optimal result.

3. We also conduct extensive simulations to validate our proposed algorithms. The simulation results show that the constructed LBVB and the allocation scheme for non backbone nodes can extend network lifetime significantly compared with the stat-of-art algorithms. Particularly, when all nodes with the same transmission range $50m$ are deployed uniformly and randomly in a fixed square area $300m \times 300m$, and the number of nodes is incremented from 50 to 100 by 10, our proposed algorithms prolong network lifetime by 69% on average compared with the latest and best MCDS-based VB [44], by 47% on average compared with LBCDS [37].

6.2 LBVB Problem Formulation

6.2.1 Network Model

We assume a static connected WSN and all the nodes in the WSN have the same transmission range. Hence, we model a WSN as an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is the set of n sensor nodes, denoted by v_i, where $1 \leq i \leq n$, i is called the node ID of v_i in the chapter; $\mathbb{E}$ represents the link set $\forall\ u, v \in \mathbb{V}, u \neq v$, there exists a link (u, v) in $\mathbb{E}$ if and only if u and v are in each other's transmission range. In this chapter, we assume links are undirected (bidirectional), which means two linked nodes are able to transmit and receive data from each other. Moreover, the degree of a node v_i is denoted by d_i, whereas Δ denotes the maximum degree in the network graph $\mathbb{G}$.

6.2.2 LBVB Problem Definition

As we mentioned in Section 6.1, we will solve the LBVB problem in two steps. The first step constructs a MinMax Degree Maximal Independent Set (MDMIS), and the second step selects additional nodes which together with the nodes in the MDMIS induce a connected topology *LBVB*. In this subsection, we first formally define the MDMIS problem, followed by the problem definition of LBVB.

Definition 6.2.1. *MinMax Degree Maximal Independent Set (MDMIS) Problem.* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$, the *MDMIS* problem is to find a node set $\mathbb{D} \subseteq \mathbb{V}$ such that:

1. $\forall u \in \mathbb{V}$ and $u \notin \mathbb{D}$*,* $\exists\ v \in \mathbb{D}$*,* such that $(u, v) \in \mathbb{E}$*.*

2. $\forall u \in \mathbb{D}$*,* $\forall v \in \mathbb{D}$*,* and $u \neq v$*,* such that $(u, v) \notin \mathbb{E}$*.*

3. There exists no proper subset or superset of $\mathbb{D}$ satisfying the above two conditions.

4. Minimize $\max\{d_i \mid \forall v_i \in \mathbb{D}\}$*.*

Taking the load-balance factor into consideration, we are seeking an MIS in which the maximum degree of the nodes in the constructed MIS is minimized. In other words, the potential traffic load on each node in the MIS is as balance as possible. Now, we are ready to define the LBVB problem.

Definition 6.2.2. *Load-Balanced Virtual Backbone (LBVB) Problem.* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$ and an *MDMIS* $\mathbb{D}$, the *LBVB* problem is to find a node set $\mathbb{C} \subseteq \mathbb{V} \backslash \mathbb{D}$ such that:

1. The induced graph $G[\mathbb{D} \bigcup \mathbb{C}]$ *on* $\mathbb{G}$ is connected.

2. Minimize $|\mathbb{C}|$*,* where $|\mathbb{C}|$ is the size of set $\mathbb{C}$*.*

For convenience, the nodes in the set $\mathbb{D}$ are called *independent nodes*, whereas, the nodes in the set $\mathbb{C}$ are called *MIS connectors*. Moreover, $\mathbb{B} = \mathbb{D} \bigcup \mathbb{C}$ is an *LBVB* of $\mathbb{G}$. Specifically speaking, $\forall v_i \in \mathbb{B}$, v_i is a *backbone node*.

Constructing an LBVB is a part of the work to balance traffic load on each backbone node. One more important task needs to be resolved is how to allocate non backbone nodes to its neighboring backbone nodes. The formal definition of the non backbone node allocation scheme are given as follows:

Definition 6.2.3. *Non Backbone Node Allocation Scheme ($\mathscr{A}$).* For a WSN represented by graph $\mathbb{G}(\mathbb{V},\mathbb{E})$ and a VB $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$, we need to find m disjoint sets on $\mathbb{V}$, denoted by $\mathbb{A}(v_1), \mathbb{A}(v_2), \cdots, \mathbb{A}(v_m)$, such that:

1. Each set $\mathbb{A}(v_i)$ $(1 \leq i \leq m)$ contains exactly one backbone node v_i.

2. $\bigcup_{i=1}^{m} \mathbb{A}(v_i) = \mathbb{V}$, and $\mathbb{A}(v_i) \bigcap \mathbb{A}(v_j) = \emptyset$ $(1 \leq i \neq j \leq m)$.

3. $\forall v_u \in \mathbb{A}(v_i)$ $(1 \leq i \leq m)$ and $v_u \neq v_i$, such that $(v_u, v_i) \in \mathbb{E}$.

A *Non Backbone Node Allocation Scheme* is:

$$\mathscr{A} = \{\mathbb{A}(v_i) \mid \forall v_i \in \mathbb{B}, 1 \leq i \leq m\}$$

As we mentioned in Section 6.1, the potential traffic load indicator on each backbone node is the degree of the node, *i.e.*, d_i, for $\forall v_i \in \mathbb{B}$. However, d_i is not the actual traffic load. The actual traffic load only can be determined when a non backbone node allocation scheme $\mathscr{A}$ is decided. In other words, the number of allocated non backbone nodes is an indicator of the actual traffic load on each backbone node. According to this observation, we give the following definition:

Definition 6.2.4. *Valid Degree (d').* The *Valid Degree* of a backbone node v_i is the number of its allocated non backbone nodes, *i.e.*, $\forall v_i \in \mathbb{B}, d'_i = |\mathbb{A}(v_i)| - 1$, where $|\mathbb{A}(v_i)|$ represents the cardinality of the set $\mathbb{A}(v_i)$.

Finally, we are dedicated to find a load-balanced non backbone node allocation scheme $\mathscr{A}$, namely, the maximum *valid degree* of all the backbone nodes is minimized under $\mathscr{A}$.

Definition 6.2.5. *MinMax Valid-Degree non Backbone node Allocation (MVBA) Problem.* For a WSN represented by graph $\mathbb{G}(\mathbb{V},\mathbb{E})$ and an *LBVB* $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$, the *MVBA* problem is to find a backbone allocation scheme $\mathscr{A}^*$, such that: the $\max\{d'_i \mid \forall v_i \in \mathbb{B})\}$ is minimized under $\mathscr{A}^*$.

6.3 Load Balanced Virtual Backbone Problem

In this section, we first introduce how to solve the MinMax Degree Maximal Independent Set (MDMIS) Problem. Since finding an MIS is a well-known NP-complete problem [77] in graph theory, we claim the LBVB is NP-complete as well. Next, we formulate the MDMIS problem as an Integer Nonlinear Programming (INP). Subsequently, we show how to obtain an $O(\Delta \ln(n))$ approximation solution by using Linear Programming (LP) relaxation techniques. Finally, we present how to find a minimum-sized set of MIS connectors to form an LBVB $\mathbb{B}$.

6.3.1 INP Formulation of MDMIS

Consider a WSN described by graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$. First we define the 1-Hop Neighborhood of a node v_i and then extend it to the r-Hop Neighborhood.

Definition 6.3.1. *1-Hop Neighborhood ($\mathbb{N}_1(v_i)$).* $\forall v_i \in \mathbb{V}$, the 1-Hop Neighborhood of node v_i is defined as:

$$\mathbb{N}_1(v_i) = \{v_j \mid v_j \in \mathbb{V}, e_{ij} = (v_i, v_j) \in \mathbb{E}\}$$

The physical meaning of 1-Hop Neighborhood is the set of the nodes that can be directly reached from node v_i.

Definition 6.3.2. **r-Hop Neighborhood**. *$\forall v_i \in \mathbb{V}$, the r-Hop Neighborhood of node v_i is defined as:*

$$\begin{aligned}\mathbb{N}_r(v_i) = \mathbb{N}_{r-1}(v_i) \ \cup \ \{v_k \mid \exists v_j \in \mathbb{N}_{r-1}(v_i),\\ v_k \in \mathbb{N}_1(v_j), \ v_k \notin \bigcup_{i=1}^{r-1} \mathbb{N}_i(v_i)\}\end{aligned}$$

The physical meaning of the r-Hop Neighborhood is that the set of the nodes that can be reached from node v_i by passing maximum r number of links.

Next we formally model the MDMIS problem as an Integer Nonlinear Program (INP).

DS property constraint. As we mentioned early, an MIS is also a DS. Hence, we should formulate the DS constraint for the MDMIS problem. For convenience, we assign a decision variable x_i for each sensor $v_i \in \mathbb{V}$, which is allowed to be 0/1 value. This variable sets to 1 *iff* the node is an independent node, *i.e.*, $\forall v_i \in \mathbb{D}, x_i = 1$. Otherwise, it sets to 0. The DS property states that each non independent node must reside within the 1-hop neighborhood of at least one independent node. We therefore have

$$x_i + \sum_{v_j \in \mathbb{N}_1(v_i)} x_j \geq 1, \forall v_i \in \mathbb{V} \tag{6.1}$$

IS property constraint. Since the solution of the MDMIS problem is at least an IS, the IS property is also a constraint of MDMIS. The IS property indicates that no two independent nodes are adjacent, *i.e.*, $\forall v_i, v_j \in \mathbb{D}, (v_i, v_j) \notin \mathbb{E}$. In other words, we have

$$\sum_{v_j \in \mathbb{N}_1(v_i)} x_i \cdot x_j = 0, \forall v_i \in \mathbb{V} \tag{6.2}$$

Consequently, the objective of the MDMIS problem is to minimize the maximum degree of all the independent nodes. We denote z as the objective of the MDMIS problem, *i.e.*, $z = \max_{v_i \in \mathbb{D}}(d_i)$. Mathematically, the MDMIS problem can be formulated as an integer nonlinear programming INP_{MDMIS} as follows:

$$\begin{aligned}
\min \quad & z = \max\{d_i \mid \forall v_i \in \mathbb{D}\} \\
s.t. \quad & x_i + \sum_{v_j \in \mathbb{N}_1(v_i)} x_j \geq 1 \\
& \sum_{v_j \in \mathbb{N}_1(v_i)} x_i \cdot x_j = 0 \\
& x_i, x_j \in \{0, 1\}, \ \forall v_i, v_j \in \mathbb{V}
\end{aligned} \tag{INP_{MDMIS}}$$

Since the IS property constraint (formulated in Equation 6.2) is quadratic, the formulated integer programming INP_{MDMIS} is not linear. To linearize INP_{MDMIS}, the quadratic constraint is eliminated by applying the techniques proposed in [78]. More specifically, the product $x_i \cdot x_j$ is replaced by a new binary variable χ_{ij}, on which several additional constraints are imposed. As a consequence, we can reformulate INP_{MDMIS} exactly to an Integer Linear

Programming ILP_{MDMIS} by introducing the following linear constraints:

$$\begin{aligned}
&\sum_{v_j \in \mathbb{N}_1(v_i)} \chi_{ij} = 0 \\
&x_i \geq \chi_{ij} \\
&x_j \geq \chi_{ij} \\
&x_i + x_j - 1 \leq \chi_{ij} \\
&\chi_{ij} \in \{0,1\}, \ \forall v_i, v_j \in \mathbb{V}
\end{aligned} \tag{6.3}$$

For convenience, we assign a random variable l_{ij} for each edge in the graph $\mathbb{G}$ modeled from a WSN, *i.e.*,

$$l_{ij} = \begin{cases} 1, & \textit{if } (v_i, v_j) \in \mathbb{E}. \\ 0, & \text{otherwise.} \end{cases}$$

Thus, we obtain that $d_i = \sum_{v_j \in \mathbb{N}_1(v_i)} x_i l_{ij}, \forall v_i \in \mathbb{V}$. Moreover, by relaxing the conditions $x_j \in \{0,1\}$, and $\chi_{ij} \in \{0,1\}$ to $x_j \in [0,1]$, and $\chi_{ij} \in [0,1]$, correspondingly, we obtain the following relaxed linear programming LP^*_{MDMIS}:

$$\begin{aligned}
\min \ & z = \max\{1, \max\{d_i = \sum_{v_j \in \mathbb{N}_1(v_i)} x_i l_{ij} \mid \forall v_i \in \mathbb{V}\}\} \\
s.t. \ & x_i + \sum_{v_j \in \mathbb{N}_1(v_i)} x_j \geq 1 \\
& \sum_{v_j \in \mathbb{N}_1(v_i)} \chi_{ij} = 0 \\
& x_i \geq \chi_{ij} \\
& x_j \geq \chi_{ij} \\
& x_i + x_j - 1 \leq \chi_{ij} \\
& x_i, x_j, \chi_{ij} \in [0,1], \ \forall v_i, v_j \in \mathbb{V}
\end{aligned} \tag{LP^*_{MDMIS}}$$

6.3.2 Approximation Algorithm

Due to the relaxation enlarged the optimization space, the solution of LP^*_{MDMIS} corresponds to a lower bound to the objective of INP_{MDMIS}. Given an instance of MDMIS modeled by the integer nonlinear programming INP_{MDMIS}, the sketch of the proposed approximation algorithm (see Algorithm 6) is summarized as follows: first, solve the relaxed linear programming LP^*_{MDMIS} to get an optimal fractional solution, denoted by $(\mathbf{x}^*, z^*)$, where $\mathbf{x}^* =< x_1^*, x_2^*, \cdots, x_n^* >$, and then round x_i^* to integers $\widehat{x_i}$ according to the following five steps:

Algorithm 6 : Approximation Algorithm for MDMIS

```
Require: A WSN represented by graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$; Node degree $d_i$.
 1: Solve $LP^*_{MDMIS}$. Let $(\mathbf{x}^*, z^*)$ be the optimum solution, where $\mathbf{x}^* =< x_1^*, x_2^*, \cdots, x_n^* >$,
    $z^* = \max(1, \sum_{v_j \in \mathbb{N}_1(v_i)} x_i^* l_{ij})$.
 2: Sort all the sensor nodes by the $x_i^*$ value in the decreasing order. The sorted node ID $i$
    is stored in the array denoted by $A[n]$.
 3: for $i = 1$ to $n$ do
 4:   $\widehat{x_i} = 0$.
 5: end for
 6: $counter = 0$.
 7: while $counter \le \beta$, where $\beta = 3(\Delta + 1)\ln(n)$ do
 8:   $k = 0$.
 9:   while $k < n$ do
10:     $i = A[k]$.
11:     if $\forall v_j \in \mathbb{N}_1(v_i),\ \widehat{x_j} = 0$, then
12:       $\widehat{x_i} = 1$ with probability $p_i = \max(x_i^*, \frac{1}{d_i})$.
13:     end if
14:     $k = k + 1$.
15:   end while
16:   $counter = counter + 1$.
17: end while
18: return $(\widehat{\mathbf{x}}, \widehat{z} = \max(1, d_i = \sum_{v_j \in \mathbb{N}_1(v_i)} \widehat{x_i} l_{ij}))$.
```

1. Sort sensor nodes by the x_i^* value (where $1 \le i \le n$) in the decreasing order (line 2).

2. Set all $\widehat{x_i}$ to be 0 (line 3-5).

3. Start from the first node in the sorted node array A (line 8). If there is no node been selected as an independent node in v_i's 1-hop neighborhood (line 11), then let $\widehat{x_i} = 1$ with probability $p_i = \max(x_i^*, \frac{1}{d_i})$ (line 12).

4. Repeat step 3) till reaching the end of array A (line 9 - 15).

5. Repeat step 3) and 4) for $3(\Delta + 1)\ln(n)$ times (line 7 - 17).

Next the correctness of our proposed approximation algorithm (Algorithm 6) is proven, followed by the performance ratio analysis. Before showing the correctness of Algorithm 6, two important lemmas are derived as follows:

Lemma 2. For a WSN represented by $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, if a subset $\mathbb{S} \subseteq \mathbb{V}$ is a DS and meanwhile $\mathbb{S}$ is also an IS, then this subset $\mathbb{S}$ is an MIS of $\mathbb{G}$.

Proof: If $\mathbb{S} \subseteq \mathbb{V}$ is a DS of $\mathbb{G}$, it implies that $\forall v_i \in \mathbb{V} \backslash \mathbb{S}$, there exists at least one node $v_j \in \mathbb{S}$ in v_i's 1-hop neighborhood. Moreover, if $\mathbb{S}$ is also an IS, it implies that no two nodes in $\mathbb{S}$ are adjacent, *i.e.*, $\forall v_s, v_t \in \mathbb{S}, (v_s, v_t) \notin \mathbb{E}$.

Suppose $\mathbb{S}$ is **not** an MIS. In other words, we can find at least one more node, that does not violate the DS property and the IS property of $\mathbb{S}$, to be added into $\mathbb{S}$. Suppose v_k is such a node. Based on the DS property, we know that $\exists v_j \in \mathbb{S}$ and $v_j \in \mathbb{N}_1(v_k)$. According to the hypothesis, $v_k \in \mathbb{S}$, and considering the fact that $v_j \in \mathbb{N}_1(v_k)$, we conclude there are two nodes (v_j and v_k) are adjacent in $\mathbb{S}$ (*i.e.*, $(v_j, v_k) \in \mathbb{E}$), which is contradicted to the IS property. Hence, the hypothesis is false and Lemma 4 is true. ■

Lemma 3. The set $\mathbb{D} = \{v_i \mid \widehat{x_i} = 1, 1 \leq i \leq n\}$, where $\widehat{x_i}$ is derived from Algorithm 6, is a DS almost surely.

Proof: Suppose $\forall v_i \in \mathbb{V}$, $|\mathbb{N}_1(v_i)| = k_i$, where $|\mathbb{N}_1(v_i)|$ is the size of set $\mathbb{N}_1(v_i)$. Let the random variable X_i denote the event that no node in the set $\mathbb{N}_1(v_i) \bigcup \{v_i\}$ is selected as an independent node. Additionally, we denote $\mathscr{X} = \max\{\frac{1}{\Delta}, \max\{x_j^* | v_j \in \mathbb{N}_1(v_i) \bigcup \{v_i\}\}\}$, *i.e.*, $\mathscr{X}$ is the maximum $\{x_j^*, \frac{1}{\Delta}\}$ value, for $v_j \in \mathbb{N}_1(v_i) \bigcup \{v_i\}$. For the probability of X_i

happening, we have

$$\begin{aligned} P(X_i) &= (1-p_1)^\beta(1-p_2)^\beta\cdots(1-p_{k_i})^\beta && (6.4)\\ &= [(1-p_1)(1-p_2)\cdots(1-p_{k_i})]^\beta && (6.5)\\ &\le (1-\mathscr{X})^\beta && (6.6)\\ &\le (1-\frac{1}{\Delta+1})^\beta && (6.7)\\ &\le (e^{-\frac{1}{\Delta+1}})^\beta && (6.8)\\ &\le e^{-\frac{3(\Delta+1)\ln(n)}{\Delta+1}} = e^{-3\ln(n)} && (6.9)\\ &= \frac{1}{n^3} && (6.10) \end{aligned}$$

Inequality 6.7 follows the fact that $\mathscr{X} \ge \frac{1}{\Delta+1}$. Inequality 6.8 results from the inequality $1-x \le e^{-x}$, $\forall x \in [0,1]$. Since $\sum\limits_{n>0} \frac{1}{n^3}$ is a particular case of the Riemann Zeta function, then $\sum\limits_{n>0} \frac{1}{n^3}$ is bound, *i.e.*, $\sum\limits_{n>0} \frac{1}{n^3} < \infty$ by the result of the Basel problem. Thus, according to the Borel-Cantelli Lemma, $P(X_i) \sim 0$, it implies there exist one independent node in the set $\mathbb{N}_1(v_i) \bigcup \{v_i\}$ almost surely. From Lemma 5, it is almost surely that the set $\mathbb{D} = \{v_i \mid \widehat{x_i} = 1, 1 \le i \le n\}$ derived from Algorithm 6 is a DS. Then, it is reasonable that we consider $\mathbb{D}$ is a DS of $\mathbb{G}$ in the following [1]. Hence $\mathbb{D}$ holds the DS property almost surely.

■

Theorem 4. The set $\mathbb{D} = \{v_i|\widehat{x_i} = 1, 1 \le i \le n\}$, where $\widehat{x_i}$ is derived from Algorithm 6, is an MIS.

Proof: According to line 11- 13 of Algorithm 6, no two nodes can both be set as independent nodes in the 1-hop neighborhood. This guarantees the IS property of $\mathbb{D}$, *i.e.*, $\forall v_i, v_j \in \mathbb{D}, (v_i, v_j) \notin \mathbb{E}$. Moreover, $\mathbb{D}$ is a DS as proven in Lemma 5. Hence, based on Lemma 4, we conclude that $\mathbb{D}$ is an MIS. ■

From Theorem 4, the solution of our proposed approximation Algorithm 6 is an MIS. Subsequently, we analyze the approximation factor of Algorithm 6 in Theorem 5.

[1] It is almost impossible that $\mathbb{D}$ is not a DS of $\mathbb{G}$. If not, we repeat the entire rounding process.

Theorem 5. Let OPT_{MDMIS} denote the optimal solution of the MDMIS problem. The proposed algorithm yields a solution of $O(\Delta \ln(n))OPT_{MDMIS}$.

Proof: The expected d_i of the independent node v_i found by Algorithm 6 is as follows:

$$\begin{aligned} E[\sum_{v_j \in \mathbb{N}_1(v_i)} \widehat{x_i} l_{ij}] &\leq \sum_{v_j \in \mathbb{N}_1(v_i)} E[\widehat{x_i}]E[l_{ij}] && (6.11)\\ &\leq \sum_{v_j \in \mathbb{N}_1(v_i)} (\beta x_i^*)E[l_{ij}] && (6.12)\\ &= \beta \sum_{v_j \in \mathbb{N}_1(v_i)} x_i^* E[l_{ij}] && (6.13)\\ &\leq \beta z^* && (6.14) \end{aligned}$$

Inequality 6.11 holds because $\widehat{x_i}$ and l_{ij} are independent. Inequality 6.12 holds because the procedure, setting $\widehat{x_i} = 1$ with probability p_i, is repeated β times. By the union bound, we get $Pr[\widehat{x_i} = 1] = Pr[\bigcup_{t \leq \beta} \widehat{x_i} = 1 \textit{ at round } t] \leq \beta x_i^*$. This implies $E(\widehat{x_i}) \leq \beta x_i^*$. Inequality 6.14 follows from the fact that $\sum_{v_j \in \mathbb{N}_1(v_i)} x_i^* \cdot E[l_{ij}] \leq \max\{d_i \mid v_i \in \mathbb{D}\} = z^*$.

Applying the Chernoff bound, we obtain the following bound:

$$Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} \widehat{x_i} l_{ij} \geq (1+\mu)\beta z^*] \leq (\frac{e^{\mu}}{(1+\mu)^{1+\mu}})^{\beta z^*} \qquad (6.15)$$

for arbitrary $\mu > 0$. To simplify this bound, let $\mu = e - 1$, we get

$$\begin{aligned} Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} \widehat{x_i} l_{ij} \geq (1+\mu)\beta z^*] &\leq (\frac{e^{e-1}}{e^e})^{\beta z^*} && (6.16)\\ &\leq e^{-\beta} && (6.17)\\ &= e^{-3(\Delta+1)\ln(n)} && (6.18)\\ &\leq e^{-3\ln(n)} = \frac{1}{n^3} && (6.19) \end{aligned}$$

Inequality 6.17 holds since $z^* = \max\{1, \max\{d_i = \sum_{v_j \in \mathbb{N}_1(v_i)} x_i l_{ij} \mid \forall v_i \in \mathbb{V}\}\} \geq 1$. Applying the union bound, we get the probability that some independent node has a degree larger than $(1+\mu)\beta z^*$,

$$Pr[\widehat{z} \geq (1+\mu)\beta z^*] \leq n\frac{1}{n^3} = \frac{1}{n^2} \tag{6.20}$$

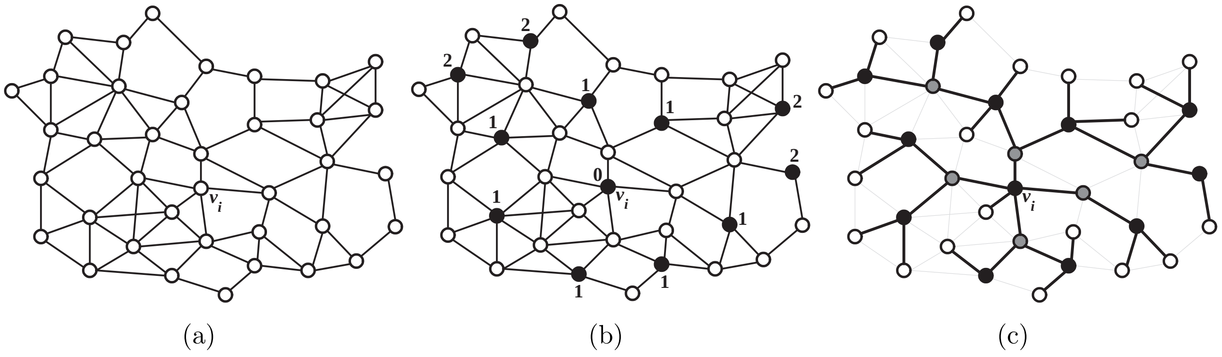
Again, since $\sum\limits_{n>0} \frac{1}{n^2}$ is a particular case of the Riemann Zeta function, then $\sum\limits_{n>0} \frac{1}{n^2}$ is bound, *i.e.*, $\sum\limits_{n>0} \frac{1}{n^2} < \infty$ by the result of the Basel problem. Thus, according to the Borel-Cantelli Lemma, $P[\widehat{z} \geq (1+\mu)\beta z^*] \sim 0$.

According to the probability of Inequality 6.10 and 6.20, we get

$$\begin{aligned} &Pr[\text{some node is selected to be an independent node} \\ &\quad \text{in 1-hop neighborhood} \bigcap \widehat{z} \leq (1+\mu)\beta z^*] \\ &\quad = 1 \cdot (1 - \tfrac{1}{n^2}) \sim 1, \ \textit{when}\ n \sim \infty \end{aligned} \tag{6.21}$$

where $\mu = e - 1$. ■

6.3.3 Connected Virtual Backbone



(a)

(b)

(c)

Figure 6.3. Illustration of LBVB construction process.

To solve the LBVB problem, one more step is needed after constructing an MDMIS, which is to make the MDMIS connected. Next, we introduce how to find a minimum-sized set of MIS connectors to connect the MDMIS.

We first divide the MDMIS $\mathbb{D}$ into disjoint node sets according to the following criterion:

$$\mathbb{D}_0 = \{v_i \mid \forall v_i \in \mathbb{D} \text{ and } v_i \text{ has the minimized node ID among all the nodess in } \mathbb{D}\}$$

$$\mathbb{D}_\iota = \{v_i \mid v_i \in \mathbb{D}, \exists v_j \in \mathbb{D}_{\iota-1}, v_i \in \mathbb{N}_2(v_j), v_i \notin \bigcup_{k=0}^{\iota-1} \mathbb{D}_k\}$$

The independent node with smallest node ID is put into $\mathbb{D}_0$. Clearly, $|\mathbb{D}_0| = 1$. All the independent nodes in the 2-Hop Neighborhood of the nodes in $\mathbb{D}_{\iota-1}$ are put into $\mathbb{D}_\iota$. Hence, ι is called the *level* of an independent node. $\mathbb{D}_\iota$ represents the set of independent nodes of level ι in $\mathbb{G}$ with respect to the node in $\mathbb{D}_0$. Additionally, suppose the maximum level of an independent node is L. For each $0 \leq i \leq L-1$, let $\mathbb{S}_i$ be the set of the nodes adjacent to at least one node in $\mathbb{D}_i$ and at least one node in $\mathbb{D}_{i+1}$. Subsequently, compute a minimum-sized set of nodes $\mathbb{C}_i \subseteq \mathbb{S}_i$ cover node set $\mathbb{D}_{i+1}$. Let $\mathbb{C} = \bigcup_{i=0}^{L-1} \mathbb{C}_i$ and therefore $\mathbb{B} = \mathbb{D} \bigcup \mathbb{C}$ is a *Load Balanced Virtual Backbone* of the original graph $\mathbb{G}$.

We use the WSN shown in Fig. 6.3 (a) as an example to explain the construction process of an LBVB. In Fig. 6.3 (a), each circle represents a sensor node. As we mentioned early, the construction process consists of two steps. In the first step, it solves the MDMIS problem by Algorithm 6 to obtain $\mathbb{D}$ which is shown in Fig. 6.3 (b) by black circles. In $\mathbb{D}$, suppose v_i is the node with the smallest node ID. Then, the number besides each independent node is the level of that node with respect to v_i. In the second phase, we choose the appropriate MIS connectors ($\mathbb{C}$), shown by gray nodes in Fig. 6.3 (c), to connect all the nodes in $\mathbb{D}$ to form an LBVB ($\mathbb{B}$).

Next, we analyze the number of backbone nodes $|\mathbb{B}|$ produced by our algorithm.

Theorem 6. The number of backbone nodes $|\mathbb{B}| \leq 2|\mathbb{D}|$.

Proof: According to the above proposed algorithm, each MIS connector connects the independent nodes in $\mathbb{D}_i$ and $\mathbb{D}_{i+1}$. Hence, $|\mathbb{C}| = |\bigcup_{i=0}^{L-1} \mathbb{C}_i| \leq \sum_{i=0}^{L-1} \max\{|\mathbb{D}_i|, |\mathbb{D}_{i+1}|\} \leq |\mathbb{D}|$. Finally, we get $|\mathbb{B}| = |\mathbb{D} \bigcup \mathbb{C}| \leq 2|\mathbb{D}|$. ■

6.4 MinMax Valid-Degree non Backbone node Allocation

In this section, we first claim that the MVBA problem is NP-Hard. Subsequently, we formulate the MVBA problem as an Integer Linear Programming (ILP). Then, we present an approximation algorithm by applying the linear relaxation and random rounding technique.

6.4.1 ILP Formulation of MVBA

According to Definition 6.2.5, the MVBA problem can be modeled by a binary problem with an linear objective functions, which is a known NP-Hard problem. In this subsection, we first model the MVBA problem as an ILP.

We define a binary variable b_i to indicate whether the sensor v_i is a backbone node or not. b_i sets to be 1 *iff* the sensor v_i is a backbone node. Otherwise, b_i sets to be 0 *iff* the sensor v_i is a non backbone node. Additionally, we assign a random variable a_{ij} for each edge connecting a backbone node v_i and a non backbone node v_j on the graph $\mathbb{G}$ modeled from a WSN, *i.e.*,

$$a_{ij} = \begin{cases} 1, \text{ if non backbone node } v_j \text{ is allocated to backbone node } v_i. \\ 0, \text{ otherwise.} \end{cases}$$

Consequently, the MVBA problem can be formulated as an Integer Linear Programming ILP_{MVBA} as follows:

$$\begin{aligned} \min \quad & y = \max\{d'_i \mid \forall v_i \in \mathbb{B}\} \\ s.t. \quad & \sum_{v_i \in \mathbb{N}_1(v_j)} b_i a_{ij} = 1, \ \forall v_j \notin \mathbb{B} \\ & a_{ij} \in \{0,1\} \end{aligned} \qquad (ILP_{MVBA})$$

The objective function y is the maximum valid degree (d') of all the backbone nodes. The first constraint states that each non backbone node can be allocated to only one backbone node, whereas the second constraint indicates that a_{ij} is a binary variable. By relaxing variable $a_{ij} \in \{0,1\}$ to $a_{ij} \in [0,1]$, we get the relaxed formulation which falls into a standard

Linear Programming (LP) problem, denoted by LP^*_{MVBA} as follows:

$$\begin{aligned}
\min \quad & y = \max\{1, \max\{\sum_{v_j \in \mathbb{N}_1(v_i)} b_i a_{ij} \mid \forall v_i \in \mathbb{B}\}\} \\
s.t. \quad & \sum_{v_i \in \mathbb{N}_1(v_j)} b_i a_{ij} = 1, \ \forall v_j \notin \mathbb{B} \\
& a_{ij} \in [0, 1]
\end{aligned} \tag{LP^*_{MVBA}}$$

Due to the relaxation enlarged the optimization space, the solution of LP^*_{MVBA} corresponds to a lower bound to the objective of ILP_{MVBA}.

6.4.2 Randomized Approximation Algorithm

Given an instance of MVBA modeled by the integer linear programming ILP_{MVBA}, the sketch of the randomized approximation algorithm (see Algorithm 7) is summarized as follows: first, solve the relaxed linear programming LP^*_{MVBA} to get an optimal fractional solution, denoted by $(\mathbf{a}^*, y^*)$, where $\mathbf{a}^* =< a^*_{11}, \cdots, a^*_{1n}, a^*_{21}, \cdots, a^*_{2n}, \cdots, a^*_{m1}, \cdots, a^*_{mn} >$, and then round a^*_{ij} to integers $\widehat{a_{ij}}$ by a random rounding procedure, which consists of four steps:

Algorithm 7 : Approximation Algorithm for MVBA

```
Require: A WSN represented by graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$.
 1: Solve $LP^*_{MVBA}$. Let $(\mathbf{a}^*, y^*)$ be the optimum solution.
 2: $\widehat{a_{ij}} = 0$.
 3: while $k \leq \alpha^2$, where $\alpha = \log(n) + 3$ do
 4:     $\widehat{a_{ij}} = 1$ with probability $a^*_{ij}$
 5:     $k = k + 1$
 6: end while
 7: if $((v_i, v_j) \in \mathbb{E})$ and $(v_i \in \mathbb{B}$ or $v_j \in \mathbb{B})$ then
 8:     $\widehat{a_{ij}} = 1$ with probability $\frac{1}{\Delta}$.
 9: end if
10: repeat
11:     line 3 - 6
12: until $\sum_{v_i \in \mathbb{N}_1(v_j)} b_i \widehat{a_{ij}} = 1$
13: return $(\widehat{\mathbf{a}}, \widehat{y} = \max(1, \sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}}))$.
```

1. Set all $\widehat{a_{ij}}$ to be 0 (line 2).
2. Let $\widehat{a_{ij}} = 1$ with probability a^*_{ij} and execute this step for α^2 times (line 3 - 6), where $\alpha = \log(n) + 3$.
3. Let $\widehat{a_{ij}} = 1$ with probability $\frac{1}{\Delta}$ (line 7).
4. To ensure $(\widehat{a_{ij}}, \widehat{y})$ is a feasible solution to ILP_{MVBA}, repeat steps 2) and 3) until every non backbone node is assigned a backbone node.

Subsequently, we analyze the approximation factor of Algorithm 7 in Theorem 7.

Theorem 7. Let OPT_{MVBA} denote the optimal solution of the MVBA problem. The proposed algorithm yields an optimal fractional solution of $O(\log^2(n))(OPT_{MVBA} + \frac{1}{\alpha^2})$ with probability $\frac{7}{8}$, when $\alpha = \log(n) + 3$.

Proof: Considering any backbone node v_i and non backbone node v_j, the expected valid degree of v_i is as follows:

$$E[\sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}}] \tag{6.22}$$

$$= \sum_{v_j \in \mathbb{N}_1(v_i)} b_i E[\widehat{a_{ij}}] \tag{6.23}$$

$$\leq \sum_{v_j \in \mathbb{N}_1(v_i)} b_i (\alpha^2 a^*_{ij} + \frac{1}{\Delta}) \tag{6.24}$$

$$= \alpha^2 \sum_{v_j \in \mathbb{N}_1(v_i)} b_i a^*_{ij} + \frac{1}{\Delta} \sum_{v_j \in \mathbb{N}_1(v_i)} b_i \tag{6.25}$$

$$\leq \alpha^2 y^* + \frac{\Delta}{\Delta} \tag{6.26}$$

$$= \alpha^2 (y^* + \varphi), where\ \varphi = \frac{1}{\alpha^2} \tag{6.27}$$

Equation 6.23 holds because b_i and $\widehat{a_{ij}}$ are independent. Inequality 6.24 holds since the random rounding technique used in Algorithm 7. Applying the *union bound*, we have the probability, that a non backbone node v_j is allocated to a backbone node v_i (*i.e.*, $\widehat{a_{ij}} = 1$) when the random rounding shown in Algorithm 7 is done, is: $Pr[\widehat{a_{ij}} = 1] = Pr[\bigcup_{k \leq \alpha^2+1} \widehat{a_{ij}} = 1 \text{ at iteration } k] \leq \alpha^2 a^*_{ij} + \frac{1}{\Delta}$. This implies $E[\widehat{a_{ij}}] \leq \alpha^2 a^*_{ij} + \frac{1}{\Delta}$. Inequality 6.26 holds because there is at most Δ non backbone nodes in a backbone node's 1-Hop Neighborhood.

Applying the Chernoff bound, we obtain the following bound:

$$\begin{aligned} &Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}} \geq (1+\mu)\alpha^2(y^* + \varphi)] \\ &\leq (\tfrac{e^\mu}{(1+\mu)^{1+\mu}})^{\alpha^2(y^*+\varphi)} \end{aligned} \tag{6.28}$$

for arbitrary $\mu > 0$. To simplify this bound, suppose $\mu \geq 2e - 1$, then

$$\begin{aligned} Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}} &\geq (1+\mu)\alpha^2(y^* + \varphi)] \\ &\leq (\tfrac{e^\mu}{(1+\mu)^{1+\mu}})^{\alpha^2(y^*+\varphi)} \\ &\leq (\tfrac{e^\mu}{(2e)^{1+\mu}})^{\alpha^2(y^*+\varphi)} \\ &= 2^{-(1+\mu)} e^{-\alpha^2(y^*+\varphi)} \\ &\leq 2^{-(1+\mu)\alpha^2(y^*+\varphi)} \\ &\leq 2^{-\mu\alpha^2(y^*+\varphi)} \end{aligned} \tag{6.29}$$

Since $y^* = \max\{1, \max\{\sum_{v_j \in \mathbb{N}_1(v_i)} b_i a_{ij} \mid \forall v_i \in \mathbb{B}\}\} \geq 1$, we have

$$\begin{aligned} Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}} &\geq (1+\mu)\alpha^2(y^* + \varphi)] \\ &\leq 2^{-\mu\alpha^2(y^*+\varphi)} \\ &\leq 2^{-\alpha^2} \leq e^{-\alpha} \\ &\leq \tfrac{1}{e^{\ln(n)+3}} \leq \tfrac{1}{16n} \end{aligned} \tag{6.30}$$

For arbitrary $0 < \mu < 2e - 1$, we let $\mu = e - 1$

$$\begin{aligned} & Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}} \geq (1+\mu)\alpha^2(y^* + \varphi)] \\ & \quad \leq (\tfrac{e^{\mu}}{(1+\mu)^{1+\mu}})^{\alpha^2(y^*+\varphi)} = (\tfrac{e^{e-1}}{e^e})^{\alpha^2(y^*+\varphi)} \\ & \quad = e^{-\alpha^2(y^*+\varphi)} \end{aligned} \tag{6.31}$$

Similarly, since $y^* \geq 1$, we get

$$\begin{aligned} & Pr[\sum_{v_j \in \mathbb{N}_1(v_i)} b_i \widehat{a_{ij}} \geq (1+\mu)\alpha^2(y^* + \varphi)] \\ & \qquad \leq e^{-\alpha^2(y^*+\varphi)} \leq e^{-\alpha^2} \leq e^{-\alpha} \\ & \qquad \leq \tfrac{1}{e^{\ln(n)+3}} \leq \tfrac{1}{16n} \end{aligned} \tag{6.32}$$

In both of the above cases (Inequality 6.30 and 6.32), summing over all backbone nodes $v_i \in \mathbb{B}$, we obtain the probability that some backbone node has a *valid degree* larger than $(1+\mu)\alpha^2(y^* + \varphi)$ as follows:

$$Pr[\widehat{y} \geq (1+\mu)\alpha^2(y^* + \varphi)] = n\frac{1}{16n} = \frac{1}{16} \tag{6.33}$$

Subsequently, we consider the probability that a non backbone node $v_j \in \mathbb{V} \backslash \mathbb{B}$ is not allocated to a backbone node in its 1-hop neighborhood at iteration j. That is,

$$\prod_{v_i \in \mathbb{N}_1(v_j),\ v_i \in \mathbb{B}} Pr[b_i \widehat{a_{ij}} = 0 \text{ at iteration } j] \tag{6.34}$$

$$= \prod_{v_i \in \mathbb{N}_1(v_j),\ v_i \in \mathbb{B}} (1 - b_i a^*_{ij}) \tag{6.35}$$

$$\leq \prod_{v_i \in \mathbb{N}_1(v_j),\ v_i \in \mathbb{B}} e^{-b_i a^*_{ij}} \tag{6.36}$$

$$= e^{-\sum_{v_i \in \mathbb{N}_1(v_j),\ v_i \in \mathbb{B}} b_i a^*_{ij}} = \frac{1}{e} \tag{6.37}$$

Inequality 6.36 results from the inequality $(1-x) \leq e^{-x},\ \forall x \in [0,1]$. Equation 6.37 follows the fact that each non backbone node can be allocated to only one backbone node, *i.e.*,

$\sum\limits_{v_i \in \mathbb{N}_1(v_j)} b_i a^*_{ij} = 1,\ \forall v_j \notin \mathbb{B},\ \forall v_i \in \mathbb{B}$. Now, the probability that any non backbone node is not allocated to a backbone node in its 1-hop neighborhood after the random rounding is $e^{-\alpha^2}$.

$$\begin{aligned} &Pr[\text{Some non backbone node} \\ &\text{has no neighboring backbone node}] \\ &\le ne^{-\alpha^2} = n\tfrac{1}{16n} = \tfrac{1}{16} \end{aligned} \tag{6.38}$$

Based on the probability in Inequality 6.33 and Inequality 6.38, we have

$$\begin{aligned} &Pr[\text{each non backbone node is allocated to a} \\ &\text{backbone node} \bigcap \widehat{y} \le (1+\mu)\alpha^2(y^* + \varphi)] \\ &\ge (1 - \tfrac{1}{16})(1 - \tfrac{1}{16}) \ge \tfrac{7}{8} \end{aligned} \tag{6.39}$$

for arbitrary $\mu > 0$ ■

From theorem 7, the solution of our proposed random approximation Algorithm 7 yields a solution upper bounded by $O(\log^2 n)(OPT_{MVBA} + \frac{1}{\alpha^2})$. Moreover, this bound can be verified in polynomial time. If not, we repeat the entire rounding process. The expected number of repetitions is at most $\lceil \frac{8}{7} \rceil = 2$, when $\alpha = \log(n) + 3$.

6.5 Performance Evaluation

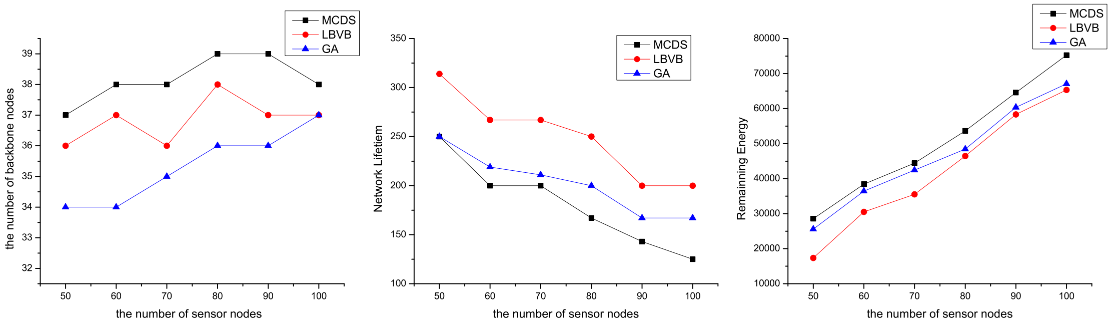
In the simulations, the results of LBVB are compared with the latest and best MCDS construction algorithm [44] denoted by MCDS, and the LBCDS-GA algorithm proposed in [37] denoted by GA. We compare the three algorithms in terms of the number of backbone nodes, network lifetime, which is defined as the time duration until the first backbone node runs out of energy, and the remaining energy over the whole network.

6.5.1 Simulation Environment

We build our own simulator where all the nodes have the same transmission range and all the nodes are deployed uniformly and randomly in a square area. For each specific setting, 100 instances are generated. The results are averaged over these 100 instances (all results are rounded to integers). Moreover, we use the VB-based data aggregation as the communication mode. The simulated energy consumption model is that every node has the same initial 1000 units of energy. Receiving and transmitting a packet both consume 1 unit of energy. In the simulation, we consider the following tunable parameters: the node transmission range, the total number of nodes deployed in the square area, and the side length of the square area. Subsequently, we show the simulation results in three different scenarios.

6.5.2 Scenario 1: Change the total number of nodes

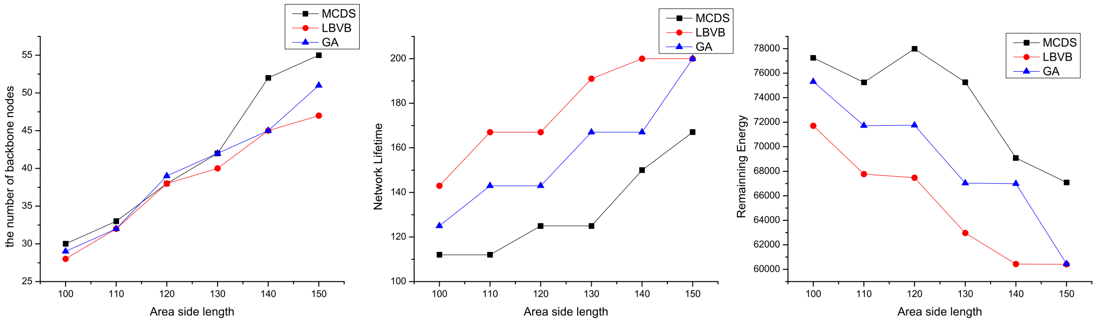
In this scenario, all nodes have the same transmission range of $50m$ and all nodes are deployed uniformly and randomly in a square area of $300m \times 300m$. The number of nodes is incremented from 50 to 100 by 10. The simulation results are shown in Fig. 6.4, where the X-axis represents the number of nodes, while the Y-axis represents the number of backbone nodes, network lifetime, and the remaining energy over the whole network respectively.



(a) (b) (c)

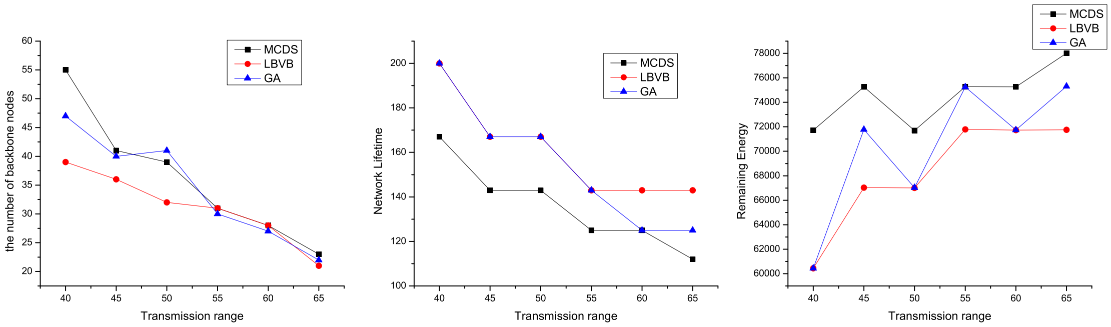
Figure 6.4. Simulation results for a square area of *300m x 300m*, the node transmission range is *50m*, and the number of nodes changes from *50* to *100*: (a) The number of backbone nodes; (b) Network Lifetime; (c) Remaining Energy.

From Fig. 6.4(a), we can see that, with the increase of the number of the sensor nodes, the number of backbone nodes keeps stable (from 34 to 39) for all the three algorithms (MCDS, LBVB, and GA). This is because the area of the network deployed region keeps



(a) (b) (c)

Figure 6.5. Simulation results for the node transmission range is *20m*, the number of nodes is *100*, and the side length of the deployed area changes from *100m* to *150m*: (a) The number of backbone nodes; (b) Network Lifetime; (c) Remaining Energy.



(a) (b) (c)

Figure 6.6. Simulation results for a square area of *300m x 300m*, the number of nodes is *100*, and the node transmission range changes from *40m* to *65m*: (a) The number of backbone nodes; (b) Network Lifetime; (c) Remaining Energy.

fixed. The results implies that, if the network deployed area keeps unchanged, the density of the WSN does not affect too much on the size of the constructed VB.

Though few changes in the number of backbone nodes, different non backbone node allocation schemes do affect network lifetime as shown in Fig. 6.4(b). From Fig. 6.4(b), we know that the network lifetime decreases for all algorithms with the number of nodes increasing. This is because we use data aggregation communication mode in a more and more crowded network. Intuitively, the denser the network is, the more number of neighbors of each backbone node. With the number of neighbors increasing, the aggregated data on each backbone node becomes more and more heavier. Hence, the network lifetime decreases for all the three algorithms. Additionally, we can see both LBVB and GA outperform than MCDS. Furthermore, LBVB prolongs network lifetime by 69% on average compared with MCDS, and by 47% on average compared with GA. The results demonstrate that load-balancedly allocating non backbone nodes to backbone nodes can improve network lifetime significantly. On the other hand, LBVB outperforms than GA, since GA takes constructing an LBVB and finding a load-balancedly allocation scheme into consideration simultaneously, whereas our proposed algorithm formulate the whole process as two separate optimization problems. Moreover, GA searches the best solution in a limited searching space. The local optimal solution found by GA might not be the same as the global optimal solution. The results shown in Fig. 6.4(b) indicate our proposed algorithms can find a solution which is closer to the optimal solution than GA.

Fig. 6.4(c) shows the remaining energy over the whole network of the three algorithms. With the increase of the number of nodes, the remaining energy increases for all algorithms. As the WSN becomes denser and denser, a lot of redundant sensor nodes exist in the WSN. From Fig. 6.4(c), we know that LBVB and GA have less remaining energy than MCDS. Additionally, LBVB has less remaining energy than GA. This is because both LBVB and GA consider the load-balance factor when building a VB and allocating non backbone nodes to backbone nodes. Thus, the lifetime of the whole network is extended, which means the remaining energy of the network is less than MCDS. On the other hand, LBVB has more network lifetime than GA as shown in Fig. 6.4(b), hence LBVB has less remaining energy than GA. In summary, Fig. 6.4 indicates that constructing an LBVB can balance the energy consumption on each backbone node, and make the lifetime of the whole network prolonged considerably.

6.5.3 Scenario 2: Change the side length of the square area

In this scenario, all nodes have the same transmission range of $20m$ and 100 nodes are deployed uniformly and randomly in a square area. The side length of the square area is incremented from 100 to 150 by 10. The simulation results are presented in Fig. 6.5, where the X-axis represents the side length of the square area, while the Y-axis represents the number of backbone nodes, network lifetime, and the remaining energy over the whole network respectively.

From Fig. 6.5(a), we can see that, with the increase of the area of the network deployed region, the number of backbone nodes increases for all the three algorithms (MCDS, LBVB, and GA). This is because the WSN becomes more and more thinner, more backbone nodes are needed to maintain the connectivity of the constructed VB. There is no obvious trend that which algorithm might produce more backbone nodes when constructing a VB. It implies that the sizes of the constructed VBs are all considered for all the three algorithms.

Though no obvious trend in the number of backbone nodes, different non backbone node allocation schemes still affect network lifetime as shown in Fig. 6.5(b). From Fig. 6.5(b), we know that the network lifetime increases for all algorithms with the side length of the deployed area increasing. It is obvious that the density of the network becomes more thinner with the side length of the deployed area increasing. As to a data aggregation communication mode, the thinner the network is the less number of neighbors of each backbone node. In other words, the aggregated data are less on each backbone node when the network becomes more and more thinner. Hence, network lifetime is increasing for all the three algorithms. Additionally, we can see both LBVB and GA outperform than MCDS. More specifically, LBVB prolongs network lifetime by 42% on average compared with MCDS, and by 20% on average compared with GA. The reasons are the same as analyzed in Scenario 1.

Fig. 6.5(c) shows the remaining energy over the whole network of the three algorithms. With the increase of the side length of the deployed area, the remaining energy decreases for all algorithms. As the WSN becomes thinner and thinner, more nodes are selected as backbone nodes to maintain the connectivity of the constructed VB. Additionally, the traffic load on each backbone node is less as mentioned early, hence, the remaining energy decreases with the area of the deployed area increasing. From Fig. 6.5(c), we know that LBVB and GA has less remaining energy than MCDS. Furthermore, LBVB has less remaining energy than GA. The reasons are the same as analyzed in Scenario 1.

6.5.4 Scenario 3: Change the node transmission range

In this scenario, 100 nodes are deployed uniformly and randomly in a square area of $300m \times 300m$. The node transmission range is incremented from 40 to 65 by 5. The simulation results are recorded in Fig. 6.6, where the X-axis represents the node transmission range, while the Y-axis represents the number of backbone nodes, network lifetime, and the remaining energy over the whole network respectively.

From Fig. 6.6(a), we can see that, with the increase of the node transmission range, the number of backbone nodes decreases for all the three algorithms (MCDS, LBVB, and GA). This is because there are more and more nodes in the circle with the node transmission range as the radius, when the node transmission range increasing. This is equivalent to the network become more denser. Hence, the connectivity of the constructed VB can still be maintained even using less backbone nodes. There is still no obvious trend that which algorithm might produce more backbone nodes when constructing a VB.

From Fig. 6.6(b), we know that the network lifetime decreases for all algorithms with the node transmission range increasing. The fact is that the network becomes denser with the node transmission range increasing. The denser the network is, the more number of neighbors on each backbone node. Since we use data aggregation as the communication mode in the simulations, the aggregated data are increasing on each backbone nodes when the network becomes more and more denser. Hence, network lifetime is decreasing for all the three algorithms. Similar results we can derive that both LBVB and GA outperform than MCDS. To be specific, LBVB prolongs network lifetime by 25% on average compared with MCDS, and by 6% on average compared with GA. The reasons are the same as analyzed in Scenario 1.

Fig. 6.6(c) shows the remaining energy over the whole network of the three algorithms. With the increase of the node transmission range, the remaining energy increases for all algorithms. A bunch of redundant sensors exist in the more and more crowded network, thus the remaining energy increases for all the three algorithms. From Fig. 6.6(c), we know that LBVB and GA have less remaining energy than MCDS. Furthermore, LBVB has less remaining energy than GA. The reasons are the same as analyzed in Scenario 1.

6.6 Summary

In this chapter, we address three fundamental problems of constructing a load-balanced VB in a WSN. More specifically, we solve the LBVB problem which is claimed to a NP-Complete problem with two steps. First, the MDMIS problem aims to find the optimal MIS such that the maximum degree of all the independent nodes is minimized. To solve this problem, a near optimal approximation algorithm is proposed, which yields an $O(\Delta \ln(n))$ approximation factor. Subsequently, the minimum-sized set of MIS connectors are found to make the MDMIS connected. The theoretical upper bound of the number of backbone nodes is analyzed in this chapter as well. In the end, the MVBA problem is dedicated to allocate non backbone nodes to proper backbone nodes with an objective to minimize the maximum valid degree of all the backbone nodes, which is a NP-Hard problem. To solve this problem, we propose an approximation algorithm by using linear relaxing and random rounding techniques, which yields a solution of $O(\log^2(n))$ approximation factor of traffic load on each backbone node. Simulations show that the proposed algorithms can extend network lifetime significantly.

CHAPTER 7

CONSTRUCTING LOAD-BALANCED DATA AGGREGATION TREES IN STOCHASTIC WIRELESS SENSOR NETWORKS

7.1 Introduction

Wireless Sensor Networks (WSNs) are emerging as the desired environment for increasing numbers of military and civilian applications. In such applications, sensor nodes periodically sense their monitored environment and send the information to the sink (or base station), at which the gathered/collected information can be further processed for end-user queries. In this *data gathering* process, *data aggregation* can be used to fuse data from different sensors to eliminate redundant transmissions, since the data sensed by different sensors are spatially correlated to some extent [79]. Hence, through this in-network data aggregation technique, the amount of data that needs to be transmitted by a sensor is reduced, which in turn decreases each sensor's energy consumption so that the whole network lifetime is extended. For continuous monitoring applications with a periodic traffic pattern, a tree-based topology is often adopted to gather and aggregate sensing data because of its simplicity. Compared with an arbitrary network topology, a tree-based topology conserves the cost of maintaining a routing table at each node, which is computationally expensive for the sensor nodes with limited resources. For clarification, data gathering trees with aggregation are also referred to as *Data Aggregation Trees* (DATs), which are directed trees rooted at the sink and have a unique directed path from each node to the sink. Additionally, in a DAT, sensing data from different sensors are combined at intermediate sensors according to certain aggregation functions including COUNT, MIN, MAX, SUM, and AVERAGE [80].

Due to the dense sensor deployment, many different DATs can be constructed to relay data from the monitored area to the sink. According to the diverse requirements of different applications, the DAT related works can be roughly classified into three categories: Energy-

Efficient Aggregation Scheduling [81, 82], Minimum-Latency Aggregation Scheduling [44, 83], and Maximum-Lifetime Aggregation Scheduling[86, 87]. It is worth mentioning that aggregation scheduling attracts a lot of interests in the current literatures. However, unlike most of the existing works which put lots of efforts on aggregation scheduling, we mainly focus on the DAT construction problem.

Furthermore, most of the existing DAT construction works are based on the ideal *Deterministic Network Model* (DNM), where any pair of nodes in a WSN is either connected or disconnected. Under this model, any specific pair of nodes are neighbors if their physical distance is less than the transmission range, while the rest of the pairs are always disconnected. However, in most real applications, the DNM cannot fully characterize the behavior of wireless links due to the existence of the *transitional region phenomenon* [4]. It is revealed by many empirical studies [1, 4] that, beyond the "always connected" region, there is a *transitional region* where a pair of nodes are probabilistically connected via the so called *lossy links* [4]. Even without collisions, data transmissions over the lossy links cannot be guaranteed. Moreover, as reported in [4], there are often much more lossy links (sometimes [5], 90% more) than fully connected links in a WSN. Therefore, in order to well characterize WSNs with lossy links, a more practical network model is the *Stochastic Network Model*(SNM). Under this model, there is a *transmission success ratio* (ι_{ij}) associated with each link connecting a pair of nodes v_i, v_j, which is used to indicate the probability that one node can successfully deliver a package to another. An example is shown in Fig.7.2(a), in which the number over each link represents its corresponding transmission success ratio, and s_0 is the sink. For convenience, the WSNs considered under the DNM are called *Deterministic WSNs*, whereas, the WSNs considered under the SNM are called *Stochastic WSNs.* Clearly, DNM is only a specific case of SNM, when $\iota_{ij} = 1$.

On the other hand, all the aforementioned works did not consider the *load-balance* factor when they construct a DAT. Without considering balancing the traffic load among the nodes on a DAT, some heavy loaded nodes may quickly exhaust their energy, which might cause network partitions or malfunctions. For instance, for aggregating the sensing data from

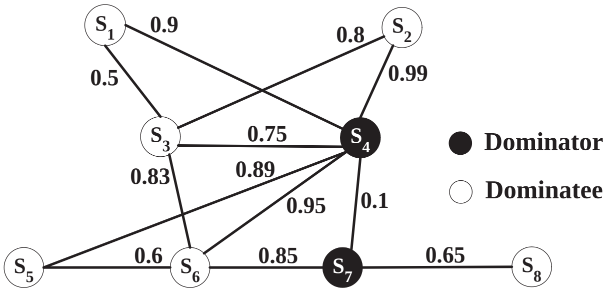


Figure 7.1. A Stochastic WSN

8 different nodes to the sink node v_0, a shortest-path-based DAT for the stochastic WSN (Fig.7.1) is shown in Fig.7.2(a). The intermediate node v_4 aggregates the sensing data from four different nodes, whereas, v_7 only aggregates one sensing data from v_8. For simplicity, if every link shown in Fig.7.1 is always connect and every node has the same amount of data to be transferred through the intermediate nodes at a fixed data rate, heavy loaded v_4 must deplete its energy much faster than v_7. From Fig.7.2(a), we know that the intermediate nodes usually aggregate the sensing data from neighboring nodes in a shortest-path-based DAT. Actually, the number of neighboring nodes of an intermediate node is a potential indicator of the traffic load on each intermediate node. However, it is not the only factor to impact the traffic load on each intermediate node. The criterion to assign a parent node, to which data is aggregated for each node on a DAT, is also critical to balance traffic load on each intermediate node. We refer the procedure, that assign a unique parent node for each node in the network, to as the Parent Node Assignment (PNA) in this chapter. An illustration of a PNA differed from Fig.7.2(a) is depicted in Fig.7.2(b). Instead of assigning v_4 as v_3 and v_6's parent, assigning v_2 as v_3's parent and v_7 as v_6's parent, a more load-balanced DAT is shown in Fig.7.2(b). Apparently, the traffic loads on the intermediate nodes v_2, v_4, and v_7 shown in Fig.7.2(a) are much more balanced than Fig.7.2(a). Therefore, constructing a Load-Balanced DAT (LBDAT) and then seeking a Load-Balanced PNA (LBPNA) are equally important when considering the load-balance factor to form an LBDAT under the SNM. Neither of these two aspects can be ignored. Finally, a perfect LBDAT is shown in Fig.7.2(c). Intuitively, compared with the DATs shown in Fig.7.2(a), and Fig.7.2(b), the

LBDAT shown in Fig.7.2(c) can extend network lifetime notably, since the traffic load are evenly distributed over all the intermediate nodes on the LBDAT.

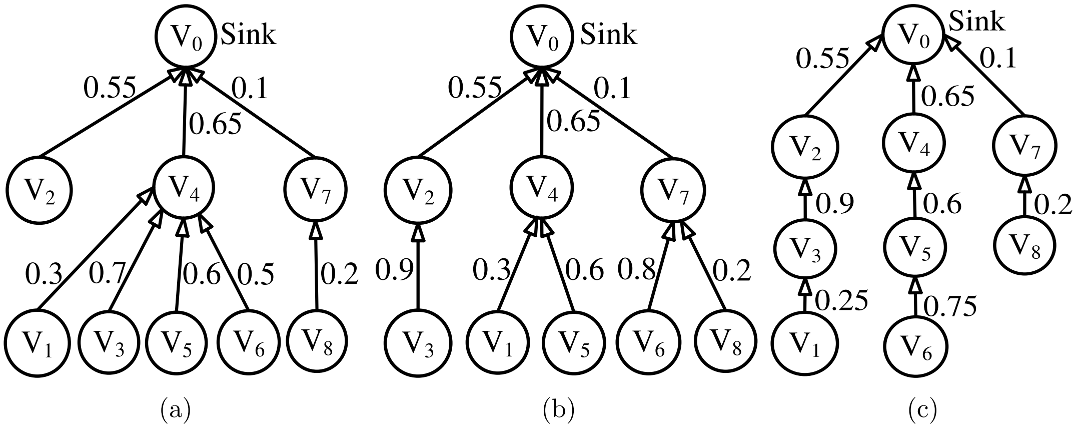


Figure 7.2. A Stochastic WSN, DATs and an LBDAT.

In summary, our research problem in this chapter is distinguished from all the prior researches in three aspects. First, most of the current literatures investigated the DAT construction problem under the DNM, whereas ours work is suitable for both DNM and SNM. Second, the *load-balance* factor is not considered when constructing a DAT in most of the aforementioned works. Finally, the DAT construction problem is our major concern, whereas the prior researches were focused on the aggregation scheduling problem. Therefore, in this chapter, we explore the DAT construction problem under the SNM with balancing the traffic load among all the nodes on a DAT. To be specific, in this chapter, we construct a Load-Balanced DAT (LBDAT) under the SNM in three phases. We first investigate how to construct a Load-Balanced Maximal Independent Set (LBMIS). A MIS can be defined formally as follows: given a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, an Independent Set (IS) is a subset $\mathbb{I} \subseteq \mathbb{V}$ such that for any two vertex $v_1, v_2 \in \mathbb{I}$, they are not adjacent, *i.e.*, $(v_1, v_2) \notin \mathbb{E}$. An IS is called an MIS if we add one more arbitrary node to this subset, the new subset will not be an IS any more. After obtaining an LBMIS, we attempt to find a minimum-sized set of nodes called *LBMIS connector set* $\mathbb{C}$ to make this LBMIS $\mathbb{M}$ connected, which is called the Connected MIS (CMIS) problem. Finally, we seek a Load-Balanced Parent Node Assignment (LBPNA).

After an LBPNA is decided, by assigning a direction to each link in the constructed tree structure, we obtain an LBDAT. Comprehensive performance ratio analysis are given in the chapter as well.

The main contributions of this chapter are summarized as follows:

- We identify and highlight the use of lossy links when constructing a DAT. Moreover, in order to measure the load-balance of the nodes on a DAT under the SNM, we define two new metrics *potential load*, and *actual load*.

- The LBDAT construction problem is an NP-Hard problem. Then, we solve the LBDAT construction problem with three phases. First, we propose an approximation algorithm by using linear relaxation and random rounding techniques to solve the LBMIS problem, which is an NP-Complete problem. Theoretical analysis shows that this algorithm yields a solution upper bounded by $O(\Delta \ln(n))opt_{LBMIS}$, where opt_{LBMIS} is the optimal result of LBMIS, Δ is the maximum node degree of the network, and n is number of sensors in a WSN. Subsequently, the minimum-sized set of nodes are found to make the LBMIS connected. Finally, to solve LBDAT, we present a randomized approximation algorithm to find a Load-Balanced Parent Node Assignment (LBPNA). The approximation algorithm produces a solution in which the actual traffic load on each intermediate node is upper bounded by $O(\Delta \log(n))(opt_{LBPNA} + \lceil \frac{B}{\mathcal{R}} \rceil)$, where opt_{LBPNA} is the optimal result, B is the data package size, and $\mathcal{R}$ is the maximum data receiving rate of all the n nodes.

- We also conduct simulations to validate our proposed algorithms. The simulation results demonstrate that the constructed LBDAT can extend network lifetime significantly compared with the stat-of-the-art algorithms.

7.2 Related Work

The problem of data gathering and aggregation in WSNs has been extensively investigated in the literatures. Moreover, a tree-based topology to periodically aggregate collected data in WSNs is widely adopted because of its simplicity. However, most of existing works concentrated on the aggregation scheduling problem in deterministic WSNs, which is very different from our research problem. To be specific, we focus on constructing an LBDAT to perform data aggregation in stochastic WSNs in this paper. Therefore, in this section, we review the most related works to our work. Based on the different user requirements, the existing DAT related works can be roughly divided into three categories: Energy-Efficient Aggregation Scheduling [46, 81, 82], Minimum-Latency Aggregation Scheduling [18, 44, 83–85], and Maximum-Lifetime Aggregation Scheduling [86–90].

7.2.1 Energy-Efficient Aggregation Scheduling

As to battery powered sensors in WSNs, energy-effici-ent is always the major concern. Hence, it is important to minimize the total energy consumed by a WSN when designing a DAT. The authors proposed a Power Efficient Data Gathering and Aggregation Protocol (PEDAP) in [81], in which a near optimal minimum energy cost spanning tree is constructed for data aggregation. At first, only the sink node is included in the tree. Then, it keeps selecting nodes not in the tree one by one to join the current tree iteratively. The selected node is the one that can transmit packets to one of the nodes in the current tree with the minimum energy cost. However, PEDAP does not consider each node's energy and cannot achieve energy-awareness. Therefore, PEDAP - Power Aware (PEDAP-PA) is proposed in [81] to improve PEDAP by considering the remaining energy of the sender. Later, the authors tried to construct an energy-balanced Minimum Degree Spanning Tree (MDST) in [46]. It starts from an arbitrary tree and tries to balance degree of nodes in the tree according to their energy. However, a node with fewer children in a DAT does not mean it can relay fewer data. Differed from the previous centralized algorithms, in [82], the authors proposed a localized, self organizing, robust, and energy-efficient DAT for a WSN called Localized Power-Efficient Data Aggregation Protocol (L-PEDAP). The proposed approach consists of two phases. In the first phase, it computes a sparse topology over the original graph of the WSN using the one-hop neighborhood information. In the second phase, it constructs a DAT over the edges of the computed sparse topology. Moreover, L-PEDAP is adaptive since it considers the dynamic changes when constructing a routing tree.

7.2.2 Minimum-Latency Aggregation Scheduling

The minimum data aggregation time problem was proved to be an NP-Hard problem in [83]. Moreover, Chen et al. [83] designed a $(\Delta - 1)R$-approximation algorithm based on a shortest path tree for data aggregation, where Δ is the maximum degree of the network graph and R is the network radius. Subsequently, the First-Fit algorithm is proposed by Huang [84], in which a Connected Dominating Set (CDS) based tree is first constructed, and then maximal interference-free set of links is scheduled in each time slot. The latency of Huang's approach is bounded by $23R + \Delta - 18$. However, the already scheduled transmissions could also interference with the candidate links which is neglected in [84]. Hence, as a successor, Wan [44] developed a $15R + \Delta - 4$ approximation algorithm called Sequential Aggregation Scheduling (SAS) to solve the Minimum-Latency Aggregation Schedule (MLAS) problem. Similar to Huang's work, Wan et al. in [44] also divided the aggregation process into the tree construction phase and the scheduling phase. The main difference is that the parents of leaf nodes are dynamically determined during the scheduling process. Subsequently, Xu et al. [85] developed an approximation algorithm with bound $16R' + \Delta - 14$, where R' is the inferior network radius which is smaller than R. Recently, Li et al. proposes a distributed scheduling algorithm named Clu-DDAS based on a novel cluster-based aggregation tree in [18] whose latency bound is $4R' + 2\Delta - 2$.

All the above works devoted efforts to find a *data aggregation schedule* for each link on the constructed DAT which leads to the minimum *data aggregation latency.* Hence, all these researches are differed than our work in this paper. We mainly focus on the load-balanced tree construction in stochastic WSNs.

7.2.3 Maximum-Lifetime Aggregation Scheduling

Wu et al. [46] proved that constructing an arbitrary aggregation tree with the maximum lifetime is NP-Hard. Hence, huge amount of approximation algorithms are proposed to construct a DAT with maximum lifetime. Xue et al. in [86], using linear programming formulation, modeled this problem as a multi-commodity flow problem, where a commodity represents the data generated from a sensor node and delivered to a base station. A fast approximate algorithm is presented, which is able to compute $(1 - \epsilon)$-approximation to the optimal lifetime for any $\epsilon > 0$. Lin et al. considered a more general network model in which the transmission power levels of sensors are heterogeneous and adjustable in [87]. The proposed algorithm starts from an arbitrary spanning tree rooted at the base station. Subsequently, one of the heavily loaded nodes is reduced to normalized load by partially rearranging the current tree to create a new tree. The upper bound on the lifetime of the constructed DAT is also presented in [87]. The authors in [88] proposed a combinatorial iterative algorithm for finding an optimal continuous solution to the Maximum Lifetime Data Gathering with Aggregation (MLDA) problem that consists of up to $n-1$ aggregation trees and achieves lifetime T_0. They obtained an α-approximate optimal integral solution, where $\alpha = \frac{T_0-n+1}{T_0}$, and n is the number of sensors in a WSN. The Decentralized Lifetime Maximizing Tree (DLMT) with energy consideration is proposed in [89]. Recently, Luo et al. proposed a distributed shortest-path based DAT in [30]. The authors transformed the problem of maximizing the lifetime of DATs into a general version of semi-matching problem, and showed that the problem can be solved by a min-cost max-flow approach in polynomial time.

7.2.4 Remarks

All the above mentioned existing works consider to construct a DAT under the DNM. To the best knowledge of us, however, none of them attempt to construct a load-balanced DAT under the SNM, which is more realistic. This is the major motivation of this research work. Moreover, all the aforementioned works were focused on constructing energy-efficient aggregation scheduling, minimum-latency aggregation scheduling, or maximum-lifetime aggregation scheduling. Unfortunately, they do not consider the load-balance factor when constructing a DAT. In contrast, in this paper, we first show an example to illustrate that an imbalanced DAT cannot prolong network lifetime by reducing the communication cost. Instead, it actually leads to the reduction of network lifetime. Based on this observation, we then study to build an LBDAT for more practical stochastic WSNs. Approximation algorithms are proposed in the paper followed by comprehensive theoretical analysis.

7.3 Network Model and LBDAT Problem Definition

In this section, we give an overview of the LBDAT construction problem under the SNM. We first present the assumptions, and then introduce the SNM. Finally, we give the problem definitions and make some remarks for the proposed problems.

7.3.1 Assumptions

We assume a static connected WSN with the set of n nodes $\mathbb{V}_s = \{v_1, v_2, \cdots, v_n\}$ and one sink node v_0. All the nodes have the same transmission range. The *transmission success ratio* ι_{ij} associated with each link connecting a pair of nodes v_i, v_j is available, which can be obtained by periodic Hello messages, or be predicted using Link Quality Index (LQI) [69]. We also assume that the ι_{ij} values are fixed. This assumption is reasonable as many empirical studies have shown that LQI is pretty stable in a static environment [70]. Furthermore, no node failure is considered since it is equivalent to a link failure case. No duty cycle is considered either. We do not consider packet collisions or transmission congestion, which are left to the MAC layer.

We further assume that the n nodes monitor the environment in the deployed area and periodically report the collected data to the sink node v_0 along the LBDAT $\mathbb{T}$ (the formal definition of LBDAT will be given later). Every node produces a data package of B bits during each report interval. Moreover, an intermediate node can aggregate multiple incoming B-bit packets, together with its own package into a single outgoing B-bit package.

Furthermore, we assume the data receiving rate of each node v_i is γ_i, and $\mathcal{R}$ denotes the maximum data receiving rate of all n nodes. Finally, the degree of a node v_i is denoted by d_i, whereas δ/Δ denotes the minimum/maximum degree in the network.

7.3.2 Network Model

Under the *Stochastic Network Model* (SNM), we model a WSN as an undirected graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$, where $\mathbb{V} = \mathbb{V}_s \bigcup \{v_0\}$ is the set of $n+1$ nodes, denoted by v_i, where $0 \leq i \leq n$. i is called the node ID of v_i in the chapter. $\mathbb{E}$ is the set of lossy links. $\forall\, v_i, v_j \in \mathbb{V}$, there exists a link (v_i, v_j) in $\mathbb{G}$ if and only if: 1) v_i and v_j are in each other's transmission range, and 2) $\iota_{ij} > 0$. For each link $(v_i, v_j) \in \mathbb{E}$, ι_{ij} indicates the probability that node v_i can successfully directly deliver a packet to node v_j; and $\mathbb{P}(\mathbb{E}) = \{\iota_{ij} \mid (v_i, v_j) \in \mathbb{E}, 0 \leq \iota_{ij} \leq 1\}$. We assume the links are undirected (bidirectional), which means two linked nodes are able to transmit and receive information from each other with the same ι_{ij} value.

7.3.3 Definition of LBDAT

Since the load-balance is the major concern of this work, the measurement of the traffic load balance under the SNM is critical to solve the LBDAT construction problem. Hence, in this subsection, we first define a novel metric called *potential load* to measure the potential traffic load on each node.

As we mentioned in Section 7.1, the number of neighboring nodes of a node (*i.e.*, $|\mathbb{N}_1(v_i)|$) is a potential indicator of the traffic load on each node. However, it is not the only factor to indicate the potential traffic load on each node in stochastic WSNs. For example, if $\iota_{ij} = 0.5$, then the expected number of transmissions to guarantee v_i to deliver one packet to v_j is $\frac{1}{0.5} = 2$. The less the ι_{ij} value, the more potential traffic load on v_j from v_i. Therefore, a more reasonable and formal definition of the potential load is given as follows:

Definition 7.3.1. Potential Load (ρ_i). $\forall v_i \in \mathbb{V}_s$, the potential load of v_i is defined as: $\rho_i = \sum\limits_{v_j \in \mathbb{N}_1(v_i)} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}}$.

We solve the LBDAT construction problem in three phases in this chapter. First, we construct a Load-Balanced Maximal Independent Set (LBMIS), and then we select additional nodes to connect the nodes in LBMIS, denoted by the Connected MIS (CMIS) problem. Finally, we acquire a Load-Balanced Parent Node Assignment (LBPNA). After LBPNA is decided, by assigning a direction of each link in the constructed tree structure, we obtain an LBDAT. In this subsection, we formally define the LBMIS, CMIS, LBPNA, and LBDAT construction problems sequentially.

Definition 7.3.2. Load-Balanced Maximal Independent Set (LBMIS) Problem. For a s-tochastic WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$, the *LBMIS* problem is to find a node set $\mathbb{M} \subseteq \mathbb{V}$ such that:

1. $v_0 \in \mathbb{M}$.

2. $\forall u \in \mathbb{V}$ and $u \notin \mathbb{M}$, $\exists\ v \in \mathbb{M}$, such that $(u, v) \in \mathbb{E}$.

3. $\forall u \in \mathbb{M}$, $\forall v \in \mathbb{M}$, and $u \neq v$, such that $(u, v) \notin \mathbb{E}$.

4. There exists no proper subset or superset of $\mathbb{M}$ satisfying the conditions 1, 2, and 3.

5. Maximize $\min\{\rho_i \mid \forall v_i \in \mathbb{M}\}$[1].

Taking the load-balance factor into consideration, we are seeking an MIS in which the minimum potential load of the nodes in the constructed LBMIS is maximized. In other words, the potential traffic load on each node in the LBMIS is as balance as possible. Now, we are ready to define the CMIS problem.

Definition 7.3.3. Connected Maximal Independent Set (CMIS) Problem. For a probablistic WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$ and an *LBMIS* $\mathbb{M}$, the *CMIS* problem is to find a node set $\mathbb{C} \subseteq \mathbb{V} \backslash \mathbb{M}$ such that:

1. The induced graph $G[\mathbb{M} \bigcup \mathbb{C}]$ on $\mathbb{G}$ is connected.

[1] MaxMin and MinMax can achieve the load-balance objective similarly according to [**?**]. In this chapter, MinMax is also applicable.

2. Minimize $|\mathbb{C}|$, where$|\mathbb{C}|$ is the cardinality of set $\mathbb{C}$.

For convenience, the nodes in set $\mathbb{M}$ are called *independent nodes*, whereas, the nodes in set $\mathbb{C}$ are called *LBMIS connectors*. Moreover, the nodes in the set $\mathbb{G}\backslash(\mathbb{M}\bigcup\mathbb{C})$ are called *leaf nodes*. Furthermore, $\forall v_i \in \mathbb{M}\bigcup\mathbb{C}$, v_i is also called a *non-leaf node*. Hence, the set of non-leaf nodes are denoted by $\mathbb{D} = \mathbb{M}\bigcup\mathbb{C}$.

Constructing a load-balanced connected topology is just one part of the work to build an LBDAT. In order to measure the actual traffic load, one more important task needed to be resolved is how to do parent node assignment for leaf nodes in the network. Since the actual traffic load of each node in a DAT is depended on the number of its children, which are composed of leaf nodes and non-leaf nodes, we give the formal definition of the parent node assignment for leaf nodes to non-leaf nodes as follows:

Definition 7.3.4. Parent Node Assignment for leaf nodes ($\mathcal{A}_L$). For a stochastic WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$ and a CMIS $\mathbb{D} = \{v_1, v_2, \cdots, v_m\}$, we need to find m disjoint sets on $\mathbb{V}$, denoted by $\mathbb{L}(v_1), \mathbb{L}(v_2), \cdots, \mathbb{L}(v_m)$, such that:

1. Each set $\mathbb{L}(v_i)$ $(1 \le i \le m)$ contains exactly one non-leaf node v_i.

2. $\bigcup_{i=1}^{m} \mathbb{L}(v_i) = \mathbb{V}$, and $\mathbb{L}(v_i)\bigcap\mathbb{L}(v_j) = \emptyset$ $(1 \le i \ne j \le m)$.

3. $\forall v_u \in \mathbb{L}(v_i)$ $(1 \le i \le m)$ and $v_u \ne v_i$, such that $(v_u, v_i) \in \mathbb{E}$.

4. Assign v_i $(1 \le i \le m)$ as the parent node of the nodes in $\mathbb{L}(v_i)\backslash\{v_i\}$.

A *Parent Node Assignment for leaf nodes* is: $\mathcal{A}_L = \{\mathbb{L}(v_i) \mid \forall v_i \in \mathbb{D}, 1 \le i \le m\}$.

Definition 7.3.5. Parent Node Assignment for non-leaf nodes ($\mathcal{A}_I$). For a stochastic WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$ and a CMIS $\mathbb{D} = \{v_1, v_2, \cdots, v_m\}$, we need to find m sets on $\mathbb{D}$, denoted by $\mathbb{I}(v_1), \mathbb{I}(v_2), \cdots, \mathbb{I}(v_m)$, such that:

1. $\forall v_i \in \mathbb{M}$, the set $\mathbb{I}(v_i)$ contains exactly one independent node v_i.

2. $\forall v_j \in \mathbb{C}$, the set $\mathbb{I}(v_j)$ contains exactly one LBMIS connector v_j.

3. $\forall v_i \in \mathbb{D}, 1 \leq |\{\mathbb{I}(v_j) \mid v_i \in \mathbb{I}(v_j), j = 1, 2, \cdots, m\}| \leq 2.$

4. $\bigcup_{i=1}^{m} \mathbb{I}(v_i) = \mathbb{D}.$

5. $\forall v_u \in \mathbb{I}(v_i)$ $(1 \leq i \leq m)$ and $v_u \neq v_i$, such that $(v_u, v_i) \in \mathbb{E}$.

6. Assign v_i $(1 \leq i \leq m)$ as the parent node of the nodes in $\mathbb{I}(v_i)\backslash\{v_i\}$.

A Parent Node Assignment for non-leaf nodes *is:* $\mathcal{A}_I = \{\mathbb{I}(v_i) \mid \forall v_i \in \mathbb{D}, 1 \leq i \leq m\}$.

$\mathcal{A}_L$ and $\mathcal{A}_I$ together is called a Parent Node Assignment (PNA) $\mathcal{A}$. According to the above definitions, as to each set $\mathbb{L}(v_i)$ in $\mathcal{A}_L$, v_i is the *parent node* of the nodes in set $\mathbb{L}(v_i)\backslash\{v_i\}$, whereas, the nodes in set $\mathbb{L}(v_i)\backslash\{v_i\}$ are called the *leaf children nodes* of v_i. Similarly, as to each set $\mathbb{I}(v_i)$ in $\mathcal{A}_I$, v_i is the *parent node* of the nodes in set $\mathbb{I}(v_i)\backslash\{v_i\}$, whereas, the nodes in set $\mathbb{I}(v_i)\backslash\{v_i\}$ are called the *non-leaf children nodes* of v_i. As we have already known, ρ_i is only the indicator of the potential traffic load on each non-leaf node. The actual traffic load only can be determined when a PNA, *i.e.*, $\mathcal{A} = \{\mathcal{A}_L, \mathcal{A}_I\}$, is decided. In other words, the number of leaf children and non-leaf children nodes (*i.e.*, $|\mathbb{L}(v_i)| - 1$ and $\mathbb{I}(v_i) - 1$) along with the corresponding ι_{ij} are the indicators of the actual traffic load on each non-leaf node v_i. According to this observation, we give the following definition:

Definition 7.3.6. Actual Load (α_i). The actual load of a *non-leaf node* v_i is: $\forall v_i \in \mathbb{D}, \alpha_i = \sum\limits_{v_j \in \{\mathbb{L}(v_i) \bigcup \mathbb{I}(v_i) \mid i \neq j\}} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}}$.

Load-Balance is our major concern, hence, when doing parent node assignment, we still need taking it into consideration. The formal definition of Load-Balanced Parent Node Assignment is as follows:

Definition 7.3.7. Load-Balanced Parent Node Assignment (LBPNA $\mathcal{A}^*$). For a stochastic WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$ and a *CMIS* $\mathbb{D} = \{v_1, v_2, \cdots, v_m\}$, the *LBPNA* problem is to find a Parent Node Assignment $\mathcal{A}^*$ for $\mathbb{V}$, such that: $\min\{\alpha_i \mid \forall v_i \in \mathbb{D}\}$ is maximized under $\mathcal{A}^*$.

After a $\mathcal{A}^*$ is decided, every node in the network has a unique parent node. Hence, a tree structure is established. The physical meaning of $\mathcal{A}^*$ is the minimum actual load among all the non-leaf nodes is maximized in the constructed DAT. Finally, we are dedicated to construct a load-balanced DAT. The formal definition of an LBDAT is:

Definition 7.3.8. Load-Balanced Data Aggregation Tree (LBDAT $\mathbb{T}$). For a stochastic WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$, a *CMIS* $\mathbb{D} = \{v_1, v_2, \cdots, v_m\}$, and a $\mathcal{A}^*$, *LBDAT* $\mathbb{T}$ is such that:

1. $\mathbb{T}$ is rooted at v_0.

2. For each link in $\mathcal{A}^*$, assigning its direction from the children node to the parent node.

Since finding an MIS is a well-known NP-Complete problem [77] in graph theory, CMIS is NP-Complete as well. Therefore CMIS cannot be solved in polynomial time unless P = NP. Consequently, we propose an approximation algorithm by using linear relaxation and random rounding technique to obtain an approximate solution. Additionally, the key aspect to solve the LBDAT construction problem is to find an LBPNA $\mathcal{A}^*$. We claim that obtaining an LBPNA is NP-Hard by formulating it as an equivalent binary programming. Consequently, we present a randomized approximation algorithm to find the approximate solution to $\mathcal{A}^*$. After specifying the direction of each link in $\mathcal{A}^*$, we obtain an LBDAT $\mathbb{T}$.

7.4 Connected Maximal Independent Set

In this section, we first introduce how to solve the Load-Balanced Maximal Independent Set (LBMIS) Problem. We formulate the LBMIS problem as an Integer Nonlinear Programming (INP). Subsequently, we show how to obtain an $O(\Delta \ln(n))$ approximation solution by using Linear Programming (LP) relaxation techniques. Finally, we present how to find a minimum-sized set of LBMIS connectors to form a CMIS $\mathbb{D}$.

7.4.1 INP Formulation of LBMIS

For convenience, we assign a decision variable ω_i for each sensor $v_i \in \mathbb{V}$, which is allowed to be 0/1 value. This variable sets to 1 if and only if the node is an independent node, *i.e.*, $\forall v_i \in \mathbb{M}, \omega_i = 1$. Otherwise, it sets to 0.

It is well known that in graph theory, an MIS is also a Dominating Set (DS). A DS is defined as a subset of nodes in a WSN such that each node in the network is either in the set or adjacent to some node in the set. Hence, we formally model the LBMIS problem as an Integer Nonlinear Programming (INP) as follows:

Sink node constraint. All aggregated data are reported to the sink node, hence the sink node is deliberately set to be an independent node, *i.e.*, $\omega_0 = 1$.

DS property constraint. Since an MIS is also a DS, we should formulate the DS constraint for the LBMIS problem first. The DS property states that each non independent node must reside within the 1-hop neighborhood of at least one independent node. We therefore have

$$\omega_i + \sum_{v_j \in \mathbb{N}_1(v_i)} \omega_j \geq 1, \forall v_i \in \mathbb{V}. \tag{7.1}$$

IS property constraint. Since the solution of the LBMIS problem is at least an IS, the IS property is also a constraint of LBMIS. The IS property indicates that no two independent nodes are adjacent, *i.e.*, $\forall v_i, v_j \in \mathbb{M}, (v_i, v_j) \notin \mathbb{E}$. In other words, we have

$$\sum_{v_j \in \mathbb{N}_1(v_i)} \omega_i \cdot \omega_j = 0, \forall v_i \in \mathbb{V}. \tag{7.2}$$

Consequently, the objective of the LBMIS problem is to maximize the minimum potential load (ρ_i) of all the independent nodes ($\forall v_i \in \mathbb{M}$). We denote ν as the objective of the LBMIS problem, *i.e.*, $\nu = \min_{v_i \in \mathbb{M}}(\rho_i)$. Mathematically, the LBMIS problem can be formulated as an integer nonlinear programming INP_{LBMIS} as follows:

$$\begin{aligned} \max \quad & \nu = \min\{\rho_i \mid \forall v_i \in \mathbb{M}\} \\ s.t. \quad & \omega_0 = 1; \\ & \omega_i + \sum_{v_j \in \mathbb{N}_1(v_i)} \omega_j \geq 1; \\ & \sum_{v_j \in \mathbb{N}_1(v_i)} \omega_i \cdot \omega_j = 0; \\ & \omega_i, \omega_j \in \{0,1\}, \ \forall v_i, v_j \in \mathbb{V}. \end{aligned} \tag{INP_{LBMIS}}$$

Since the IS property constraint (formulated in Equation (7.2)) is quadratic, the formulated integer programming INP_{LBMIS} is not linear. To linearize INP_{LBMIS}, the quadratic constraint is eliminated by applying the techniques proposed in [78]. More specifically, the product $\omega_i \cdot \omega_j$ is replaced by a new binary variable ϖ_{ij}, on which several additional constraints are imposed. As a consequence, we can reformulate INP_{LBMIS} exactly to an Integer Linear Programming ILP_{LBMIS} by introducing the following linear constraints:

$$\begin{aligned} & \sum_{v_j \in \mathbb{N}_1(v_i)} \varpi_{ij} = 0 \\ & \omega_i \geq \varpi_{ij}; \omega_j \geq \varpi_{ij} \\ & \omega_i + \omega_j - 1 \leq \varpi_{ij}; \varpi_{ij} \in \{0,1\}, \ \forall v_i, v_j \in \mathbb{V}. \end{aligned} \tag{7.3}$$

According to Definition 7.3.1, we obtain that the potential load of an independent node v_i is $\rho_i = \sum_{j:\omega_i \iota_{ij} > 0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}}$. Moreover, by relaxing the conditions $\omega_j \in \{0,1\}$ and $\varpi_{ij} \in \{0,1\}$ to $\omega_j \in [0,1]$ and $\varpi_{ij} \in [0,1]$, correspondingly, we obtain the following relaxed linear programming LP^*_{LBMIS}:

$$\begin{aligned}
&\max\ \nu = \min\{\ \rho_i = \sum_{j:\omega_i \iota_{ij}>0} \lceil \tfrac{B}{\gamma_i} \rceil \tfrac{1}{\iota_{ij}} \mid \forall v_i \in \mathbb{V}_s\} \\
&s.t.\ \ \omega_0 = 1; \\
&\qquad \omega_i + \sum_{v_j \in \mathbb{N}_1(v_i)} \omega_j \geq 1; \\
&\qquad \sum_{v_j \in \mathbb{N}_1(v_i)} \varpi_{ij} = 0; \\
&\qquad \omega_i \geq \varpi_{ij}; \qquad \omega_j \geq \varpi_{ij}; \\
&\qquad \omega_i + \omega_j - 1 \leq \varpi_{ij}; \\
&\qquad \omega_i, \omega_j, \varpi_{ij} \in [0,1],\ \forall v_i, v_j \in \mathbb{V}_s.
\end{aligned} \tag{LP^*_{LBMIS}}$$

7.4.2 Approximation Algorithm

Due to the relaxation enlarged the optimization space, the solution of LP^*_{LBMIS} corresponds to an upper bound to the objective of INP_{LBMIS}. Given an instance of LBMIS modeled by the integer nonlinear programming INP_{LBMIS}, we propose an approximation algorithm as shown in Algorithm 8 to search for an LBMIS.

The sketch of Algorithm 8 is summarized as follows: first, solve the relaxed linear programming LP^*_{LBMIS} to get an optimal fractional solution, denoted by $(\boldsymbol{\omega}^*, \nu^*)$, where $\boldsymbol{\omega}^* =< \omega^*_1, \omega^*_2, \cdots, \omega^*_n >$, and then round ω^*_i to integer $\widehat{\omega_i}$ according to the following six steps:

1. Sort sensor nodes by the ω^*_i value (where $1 \leq i \leq n$) in the decreasing order (line 2).

2. Set the sink node to be the independent node, *i.e.*, $\widehat{\omega_0} = 1$ (line 3).

3. Set all $\widehat{\omega_i}$ to be 0 (line 4 - 6).

4. Start from the first node in the sorted node array A (line 9). If there is no node been selected as an independent node in v_i's 1-hop neighborhood (line 12), then let $\widehat{\omega_i} = 1$ with probability $p_i = \max(\omega^*_i, \frac{1}{d_i})$ (line 13).

5. Repeat step 4 till reach the end of array A (line 10 - 16).

Algorithm 8 : Approximation Algorithm for LBMIS

```
Require: A stochastic WSN represented by graph $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$.
 1: Solve $LP^*_{LBMIS}$. Let $(\mathbb{\omega}^*, \nu^*)$ be the optimum solution, where $\mathbb{\omega}^* =< \omega_1^*, \omega_2^*, \cdots, \omega_n^* >$,
    $\nu^* = \min\{ \sum_{j:\omega_i^* \iota_{ij} > 0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} \mid \forall v_i \in \mathbb{V} \}$.
 2: Sort all the sensor nodes by the $\omega_i^*$ value in the decreasing order. The sorted node ID $i$
    is stored in the array denoted by $A[n]$.
 3: $\widehat{\omega_0} = 1$.
 4: for $i = 1$ to $n$ do
 5:    $\widehat{\omega_i} = 0$.
 6: end for
 7: $counter = 0$.
 8: while $counter \leq \tau$, where $\tau = 3(\Delta + 1)\ln(n)$ do
 9:    $k = 0$.
10:    while $k < n$ do
11:       $i = A[k]$.
12:       if $\forall v_j \in \mathbb{N}_1(v_i),\ \widehat{\omega_j} = 0$, then
13:          $\widehat{\omega_i} = 1$ with probability $p_i = \max(\omega_i^*, \frac{1}{d_i})$.
14:       end if
15:       $k = k + 1$.
16:    end while
17:    $counter = counter + 1$.
18: end while
19: return $(\widehat{\mathbb{\omega}}, \widehat{\nu} = \min\{ \sum_{j:\widehat{\omega_i} \iota_{ij} > 0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} \mid \forall v_i \in \mathbb{V} \})$.
```

6. Repeat step 4 and 5 for $3(\Delta + 1)\ln(n)$ times (line 8 - 18).

Next the correctness of our proposed approximation algorithm (Algorithm 8) is proven, followed by the performance ratio analysis. Before showing the correctness of Algorithm 8, two important lemmas are given as follows.

Lemma 4. For a stochastic WSN represented by $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$, if a subset $\mathbb{S} \subseteq \mathbb{V}$ is a DS and meanwhile $\mathbb{S}$ is also an IS, then this subset $\mathbb{S}$ is an MIS of $\mathbb{G}$.

Proof. If $\mathbb{S} \subseteq \mathbb{V}$ is a DS of $\mathbb{G}$, it implies that $\forall v_i \in \mathbb{V} \backslash \mathbb{S}$, there exists at least one node $v_j \in \mathbb{S}$ in v_i's 1-hop neighborhood. Moreover, if $\mathbb{S}$ is also an IS, it implies that no two nodes in $\mathbb{S}$ are adjacent, *i.e.*, $\forall v_s, v_t \in \mathbb{S}, (v_s, v_t) \notin \mathbb{E}$.

Suppose $\mathbb{S}$ is **not** an MIS. In other words, we can find at least one more node, that does not violate the DS property and the IS property of $\mathbb{S}$, to be added into $\mathbb{S}$. Suppose v_i is such a node. Based on the DS property, we know that $\exists v_j \in \mathbb{S}$ and $v_j \in \mathbb{N}_1(v_i)$. According to the hypothesis, $v_i \in \mathbb{S}$, and considering the fact that $v_j \in \mathbb{N}_1(v_i)$, we conclude there are two nodes (v_i and v_j) are adjacent in $\mathbb{S}$ (*i.e.*, $(v_i, v_j) \in \mathbb{E}$), which is contradicted to the IS property. Hence, the hypothesis is false and Lemma 4 is true. ■

Lemma 5. The set $\mathbb{M} = \{v_i \mid \widehat{\omega_i} = 1, 0 \leq i \leq n\}$, where $\widehat{\omega_i}$ is derived from Algorithm 8, is a DS almost surely.

Proof. Suppose $\forall v_i \in \mathbb{V}$, $|\mathbb{N}_1(v_i)| = k_i$, where $|\mathbb{N}_1(v_i)|$ is the cardinality of the set $\mathbb{N}_1(v_i)$. Let the random variable W_i denote the event that no node in the set $\mathbb{N}_1(v_i) \bigcup \{v_i\}$ is selected as an independent node. Additionally, we denote $\mathscr{W} = \max\{\frac{1}{\Delta}, \max\{\omega_j^* \mid v_j \in \mathbb{N}_1(v_i) \bigcup \{v_i\}\}\}$, *i.e.*, $\mathscr{W}$ is the maximum $\{\omega_j^*, \frac{1}{\Delta}\}$ value, for $v_j \in \mathbb{N}_1(v_i) \bigcup \{v_i\}$. For the probability of W_i happening, we have

$$P(W_i) = [(1-\omega_1)(1-\omega_2)\cdots(1-\omega_{k_i})(1-\omega_i)]^\tau \tag{7.4}$$

$$\leq (1-\mathscr{W})^\tau \leq (1-\frac{1}{\Delta+1})^\tau \leq (e^{-\frac{1}{\Delta+1}})^\tau \tag{7.5}$$

$$\leq e^{-\frac{3(\Delta+1)\ln(n)}{\Delta+1}} = e^{-3\ln(n)} = \frac{1}{n^3}. \tag{7.6}$$

Thus, according to the Borel-Cantelli Lemma, $P(W_i) \sim 0$, it implies there exists one independent node in the set $\mathbb{N}_1(v_i) \bigcup \{v_i\}$ almost surely, *i.e.*, it is almost surely that the set $\mathbb{M} = \{v_i \mid \widehat{\omega_i} = 1, 0 \leq i \leq n\}$ derived from Algorithm 8 is a DS. Then, it is reasonable that we consider $\mathbb{M}$ is a DS of $\mathbb{G}$ in the following[2]. ■

Theorem 8. The set $\mathbb{M} = \{v_i | \widehat{\omega_i} = 1, 0 \leq i \leq n\}$, where $\widehat{\omega_i}$ is derived from Algorithm 8, is an MIS.

Proof. According to line 11- 13 of Algorithm 8, no two nodes can both be set as independent nodes in the 1-hop neighborhood. This guarantees the IS property of $\mathbb{M}$, *i.e.*, $\forall v_i, v_j \in \mathbb{M}, (v_i, v_j) \notin \mathbb{E}$. Moreover, $\mathbb{M}$ is a DS as proven in Lemma 5. Hence, based on Lemma 4, we conclude that $\mathbb{M}$ is an MIS. ■

[2] It is almost impossible that $\mathbb{M}$ is not a DS of $\mathbb{G}$. If not, we repeat the entire rounding process.

From Theorem 8, we know that the solution of Algorithm 8 is an MIS. Subsequently, we analyze the approximation factor of Algorithm 8 in Theorem 9.

Theorem 9. Let opt_{LBMIS} denote the optimal solution of the LBMIS problem. The proposed algorithm yields a solution of $O(\Delta \ln(n))opt_{LBMIS}$.

Proof. The expected ρ_i of the independent node v_i found by Algorithm 8 is as follows:

$$E[\sum_{j:\widehat{\omega_i}\iota_{ij}>0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}}] \geq E[\sum_{j:\widehat{\omega_i}\iota_{ij}>0} \lceil \frac{B}{\gamma_i} \rceil]. \tag{7.7}$$

Inequality (7.7) holds because $0 \leq \iota_{ij} \leq 1$. From the above formula, we know the expected ρ_i of the independent node v_i mainly depends on the number of the neighboring nodes of v_i. Hence, we obtain:

$$E[\sum_{j:\widehat{\omega_i}\iota_{ij}>0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}}] \geq E[\lceil \frac{B}{\gamma_i} \rceil \sum_{v_j \in \mathbb{N}_1(v_i)} \widehat{\omega_i}\iota_{ij}] \tag{7.8}$$

$$\geq \lceil \frac{B}{\gamma_i} \rceil \sum_{v_j \in \mathbb{N}_1(v_i)} E[\widehat{\omega_i}]E[\iota_{ij}] \tag{7.9}$$

$$\geq \lceil \frac{B}{\gamma_i} \rceil \sum_{v_j \in \mathbb{N}_1(v_i)} (\omega_i^*)E[\iota_{ij}] \geq \lceil \frac{B}{\gamma_i} \rceil \nu^* \geq \lceil \frac{B}{\mathcal{R}} \rceil \nu^*. \tag{7.10}$$

Since $0 \leq \iota_{ij} \leq 1$, we get Inequality (7.8). Inequality (7.9) holds because $\widehat{\omega_i}$ and ι_{ij} are independent. The first inequality of (7.10) holds because the procedure, setting $\widehat{\omega_i} = 1$ with probability p_i, is repeated τ times. This implies $E(\widehat{\omega_i}) \geq \omega_i^*$. The second Inequality of (7.10) follows from the fact that $\sum_{v_j \in \mathbb{N}_1(v_i)} \omega_i^* E[\iota_{ij}] \geq \nu^*$.

Applying the Chernoff bound, we obtain the following bound:

$$Pr[\sum_{j:\widehat{\omega_i}\iota_{ij}>0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} \leq (1-\sigma)\tau \lceil \frac{B}{\mathcal{R}} \rceil \nu^*] \leq (\frac{e^{-\sigma}}{(1-\sigma)^{1-\sigma}})^{\tau \lceil \frac{B}{\mathcal{R}} \rceil \nu^*}.$$

for arbitrary $0 < \sigma < 1$. To simplify this bound, let $\sigma = \frac{1}{\Delta+1}$, we get

$$Pr[\sum_{j:\widehat{\omega}_i \iota_{ij}>0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} \leq (1-\sigma)\tau \lceil \frac{B}{\mathcal{R}} \rceil \nu^*] \tag{7.11}$$

$$\leq (e^{-\sigma}(1-\sigma)^{\sigma-1})^{\tau \lceil \frac{B}{\mathcal{R}} \rceil \nu^*} \tag{7.12}$$

$$= e^{-\sigma\tau\nu^*}(1-\sigma)^{(\sigma-1)\tau\lceil \frac{B}{\mathcal{R}} \rceil \nu^*} \tag{7.13}$$

$$\simeq e^{-\sigma\tau\nu^*} \leq e^{-\frac{1}{\Delta+1}3(\Delta+1)\ln(n)} = e^{-3\ln(n)} = \frac{1}{n^3}. \tag{7.14}$$

The first approximate equal of (7.14) holds since when $\Delta \to \infty$ (*i.e.*, $n \to \infty$), $\sigma = \frac{1}{\Delta+1} \to 0$. It implies $1 - \sigma = 1$. The second Inequality of (7.14) holds because $\nu^* = \min\{\sum_{j:\omega_i^* \iota_{ij}>0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} \mid \forall v_i \in \mathbb{V}\} > 1$. Applying the union bound, we get the probability that some independent node has the potential load $\widehat{\nu}$ less than $(1-\sigma)\tau\nu^*$,

$$Pr[\widehat{\nu} \leq (1-\sigma)\tau\nu^*] \leq n\frac{1}{n^3} = \frac{1}{n^2}. \tag{7.15}$$

Again, since $\sum_{n>0} \frac{1}{n^2}$ is a particular case of the Riemann Zeta function, then $\sum_{n>0} \frac{1}{n^2}$ is bounded, *i.e.*, $\sum_{n>0} \frac{1}{n^2} < \infty$ by the result of the Basel problem. Thus, according to the Borel-Cantelli Lemma, $P[\widehat{\nu} \leq (1-\sigma)\tau\nu^*] \sim 0$.

According to the probability of Inequality (7.6) and (7.15), we get

$$\begin{aligned} &Pr[\text{a node is selected to be an independent node} \\ &\quad \text{in 1-hop neighborhood} \bigcap \widehat{\nu} \geq (1-\sigma)\tau\nu^*] \\ &\quad = 1 \cdot (1 - \tfrac{1}{n^2}) \sim 1, \ \textit{when } n \sim \infty, \end{aligned} \tag{7.16}$$

where $\sigma = \frac{1}{\Delta+1}$, and $\tau = 3(\Delta+1)\ln(n)$.

Based on Inequality (7.16), the minimum potential load on all the independent nodes produced by Algorithm 8 is upper bounded by $(1-\sigma)\tau\nu^*$ with probability 1, where ν^* is the solution of LP^*_{LBMIS}, and $\tau = 3(\Delta+1)\ln(n)$. Hence, Theorem 9 is proven. ■

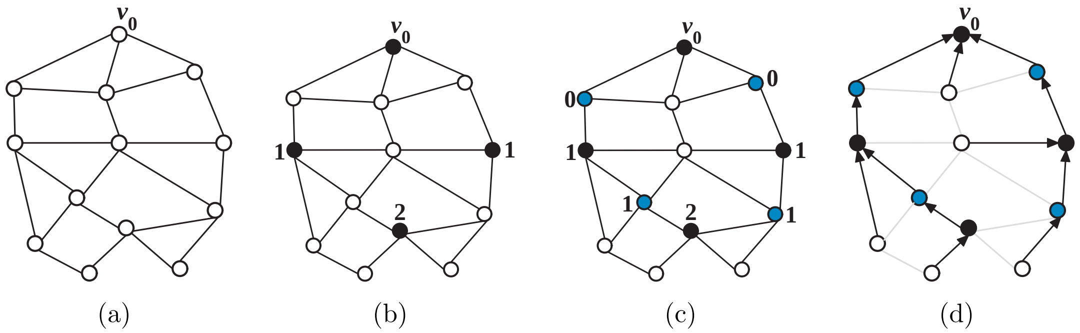


Figure 7.3. Illustration of a CMIS construction process.

7.4.3 Connecting LBMIS

To solve the CMIS problem, one more step is needed after constructing an LBMIS, which is making the LBMIS connected. Next, we introduce how to find a minimum-sized set of LBMIS connectors to connect the constructed LBMIS.

We first divide the LBMIS $\mathbb{M}$ into disjoint node sets according to the following criterion: $\mathbb{M}_0 = \{v_0\}$ and $\mathbb{M}_l = \{v_i \mid v_i \in \mathbb{M}, \exists v_j \in \mathbb{M}_{l-1}, v_i \in \mathbb{N}_2(v_j), v_i \notin \bigcup_{k=0}^{l-1} \mathbb{M}_k\}$. The sink node is put into $\mathbb{M}_0$. Clearly, $|\mathbb{M}_0| = 1$. All the independent nodes in the 2-Hop Neighborhood of the nodes in $\mathbb{M}_{l-1}$ are put into $\mathbb{M}_l$. Hence, l is called the *level* of an independent node. $\mathbb{M}_l$ represents the set of independent nodes of level l in $\mathbb{G}$ with respect to the node in $\mathbb{M}_0$. Additionally, suppose the maximum level of an independent node is L. For each $0 \leq i \leq L-1$, let $\mathbb{S}_i$ be the set of the nodes adjacent to at least one node in $\mathbb{M}_i$ and at least one node in $\mathbb{M}_{i+1}$. Subsequently, compute a minimum-sized set of nodes $\mathbb{C}_i \subseteq \mathbb{S}_i$ cover the nodes in set $\mathbb{M}_{i+1}$. Let $\mathbb{C} = \bigcup_{i=0}^{L-1} \mathbb{C}_i$ and therefore $\mathbb{D} = \mathbb{M} \bigcup \mathbb{C}$ is a CMIS of the original graph $\mathbb{G}$.

We use the WSN shown in Fig.7.3 (a) as an example to explain the construction process of a CMIS. In Fig.7.3 (a), each circle represents a sensor node. As we mentioned early, the construction process consists of two phases. In the first phase, it solves the LBMIS problem by Algorithm 8 to obtain $\mathbb{M}$ which is shown in Fig.7.3 (b) by black circles. The number besides each independent node is the level of that node with respect to the sink node v_0. In the second phase, we choose the appropriate LBMIS connectors ($\mathbb{C}$), shown by gray nodes in Fig.7.3 (c), to connect all the nodes in $\mathbb{M}$ to form a CMIS ($\mathbb{D}$).

Next, we analyze the number of non-leaf nodes $|\mathbb{D}|$ produced by our proposed algorithms. The Lemma below presents some additional properties of the constructed CMIS [44]:

Lemma 6. The following statements are true.

1. For each $0 \leq i < L$, each LBMIS connector in $\mathbb{C}_i$ is adjacent to at most 4 independent nodes in $\mathbb{M}_{i+1}$.
2. For each $1 \leq i < L-1$, each independent node in $\mathbb{M}_i$ is adjacent to at most 11 LBMIS connector in $\mathbb{C}_i$.
3. $|\mathbb{C}_0| \leq 12$.

Based on Lemma 6, we have the following theorem.

Theorem 10. The number of non-leaf nodes satisfies $|\mathbb{M}| + \lceil \frac{\mathbb{M}}{4} \rceil \leq |\mathbb{D}| \leq 2|\mathbb{M}|$.

Proof. According to the above proposed algorithm, each LBMIS connector connects the independent nodes in $\mathbb{M}_i$ and $\mathbb{M}_{i+1}$. Hence, $|\mathbb{C}| = |\bigcup_{i=0}^{L-1} \mathbb{C}_i| \leq \sum_{i=0}^{L-1} \max\{|\mathbb{M}_i|, |\mathbb{M}_{i+1}|\} \leq |\mathbb{M}|$. Moreover, according to Lemma 6, $|\mathbb{C}| \geq \lceil \frac{\mathbb{M}}{4} \rceil$. Finally, we get $|\mathbb{M}| + \lceil \frac{\mathbb{M}}{4} \rceil \leq |\mathbb{M} \bigcup \mathbb{C}| = |\mathbb{D}| \leq |\mathbb{M}| + |\mathbb{C}| \leq 2|\mathbb{M}|$. ■

7.5 Load-Balanced Data Aggregation Tree

A tree structure is decided after the Load-Balanced Parent Node Assignment (LBPNA) $\mathcal{A}^*$ is produced. By assigning a direction of each link in the constructed tree from the children node to the parent node, we obtain an LBDAT. Hence, in this section, we first formulate the LBPNA problem as an Integer Linear Programming (ILP). Then, we present an approximation algorithm by applying the linear relaxation and random rounding technique. Finally, we use an example to illustrate how to build an LBDAT.

7.5.1 ILP Formulation of Load-Balanced Parent Node Assignment

Finding an LBPNA can be modeled by a binary problem with a linear objective functions, which is a known NP-Hard problem. Hence, according to Definition 7.3.8, the LBDAT construction problem is a NP-Hard problem. In this subsection, we first model LBPNA as an ILP.

We define a binary variable β_i to indicate whether the sensor v_i is a non-leaf node or not. β_i sets to be 1 *iff* the sensor v_i is a non-leaf node. Otherwise, β_i sets to be 0. Additionally, we assign a random variable ξ_{ij} for each link connecting a non-leaf node v_i and a leaf node v_j on the graph $\mathbb{G}$ modeled from a stochastic WSN, *i.e.*, $\xi_{ij} = 1$, *if non-leaf node v_i is assigned to be the parent of leaf node v_j*; or $\xi_{ij} = 0$, *otherwise.*

Consequently, LBPNA can be formulated as an Integer Linear Programming ILP_{LBPNA} as follows:

$$\begin{aligned}
\max \quad & \vartheta = \min\{\alpha_i = \sum_{v_j \in \{\mathbb{L}(v_i) \bigcup \mathbb{I}(v_i) \ \mid \ i \neq j\}} \lceil \tfrac{B}{\gamma_i} \rceil \tfrac{1}{\iota_{ij}} \\
& \qquad\qquad \mid \forall v_i \in \mathbb{D}\} \\
s.t. \quad & \sum_{v_i \in \mathbb{N}_1(v_j)} \beta_i \xi_{ij} = 1, \ \forall v_j \notin \mathbb{D} \\
& \xi_{ij} \in \{0, 1\}.
\end{aligned} \qquad (ILP_{LBPNA})$$

The objective function ϑ is the minimum actual load (α_i) among all the non-leaf nodes. The first constraint states that each leaf node can be allocated to only one non-leaf node, whereas the second constraint indicates that ξ_{ij} is a binary variable. According to Definition 7.3.6, the number of *leaf children nodes* and the number of *non-leaf children nodes* are both contributed to the actual load of a non-leaf node. The leaf children nodes of parent node v_i can be represented by $v_j : \beta_i \xi_{ij} > 0$. Moreover, as stated in Lemma 6, the number of non-leaf children nodes of a independent parent node $\forall v_i \in \mathbb{M}$ is no more than 12, whereas, the number of non-leaf children nodes of an LBMIS connector parent node $\forall v_i \in \mathbb{C}$ is no more than 4. For simplicity, we consider the total actual load of leaf children nodes is approximated to $12\lceil \frac{B}{\mathcal{R}} \rceil$, (*i.e.*, $\sum_{v_j \in \{\mathbb{I}(v_i) \ \mid \ i \neq j\}} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} \simeq 12\lceil \frac{B}{\mathcal{R}} \rceil$)[3]. Therefore, by relaxing variable $\xi_{ij} \in \{0, 1\}$ to $\xi_{ij} \in [0, 1]$, we get the relaxed formulation which falls into a standard Linear Programming

[3] It loses only a constant factor.

(LP) problem, denoted by LP^*_{LBPNA} as follows:

$$\begin{aligned} \max \quad & \vartheta = \min\{\alpha_i = \sum_{j:\beta_i\xi_{ij}>0} \lceil \tfrac{B}{\gamma_i} \rceil \tfrac{1}{\iota_{ij}} + 12\lceil \tfrac{B}{\mathcal{R}} \rceil \mid \forall v_i \in \mathbb{D}\}\} \\ s.t. \quad & \sum_{v_i \in \mathbb{N}_1(v_j)} \beta_i \xi_{ij} = 1, \ \forall v_j \notin \mathbb{D} \qquad\qquad (LP^*_{LBPNA}) \\ & \xi_{ij} \in [0,1]. \end{aligned}$$

Due to the relaxation enlarged the optimization space, the solution of LP^*_{LBPNA} corresponds to a upper bound of the objective of ILP_{LBPNA}.

Algorithm 9 : Approximation Algorithm for LBPNA

```
Require: A stochastic WSN represented by graph $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{P}(\mathbb{E}))$, a CMIS $\mathbb{D}$.
1: Solve $LP^*_{LBPNA}$. Let $(\mathfrak{x}^*, \vartheta^*)$ be the optimum solution.
2: $\widehat{\xi_{ij}} = 0$, $k = 0$.
3: while $k \leq \kappa$, where $\kappa = 3\Delta \log(n)$ do
4:     $\widehat{\xi_{ij}} = 1$ with probability $\xi^*_{ij}$
5:     $k = k + 1$
6: end while
7: if $((v_i, v_j) \in \mathbb{E})$ and $(v_i \in \mathbb{D}$ or $v_j \in \mathbb{D})$ then
8:     $\widehat{\xi_{ij}} = 1$ with probability $\frac{1}{\delta}$.
9: end if
10: repeat
11:     line 3 - 6
12: until $\sum_{v_i \in \mathbb{N}_1(v_j)} \beta_i \widehat{\xi_{ij}} = 1$
13: return $(\widehat{\mathfrak{x}}, \widehat{\vartheta} = \min(\sum_{j:\beta_i\widehat{\xi_{ij}}>0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} + 12\lceil \frac{B}{\mathcal{R}} \rceil))$.
```

7.5.2 Randomized Approximation Algorithm

Given an instance of LBPNA modeled by the integer linear programming ILP_{LBPNA}, the sketch of the randomized approximation algorithm is shown in Algorithm 9. We summarize Algorithm 9 as follows: first, solve the relaxed linear programming LP^*_{LBPNA} to get an optimal fractional solution, denoted by $(\mathfrak{x}^*, \vartheta^*)$, where $\mathfrak{x}^* =< \xi^*_{11}, \cdots, \xi^*_{1n}, \xi^*_{21}, \cdots, \xi^*_{2n}, \cdots, \xi^*_{m1}, \cdots, \xi^*_{mn} >$, and then round ξ^*_{ij} to integers $\widehat{\xi_{ij}}$ by a random rounding procedure, which consists of four steps:

1. Set all $\widehat{\xi_{ij}}$ to be 0 (line 2).

2. Let $\widehat{\xi_{ij}} = 1$ with probability ξ_{ij}^* and execute this step for κ times (line 3 - 6), where $\kappa = 3\Delta \log(n)$.

3. Let $\widehat{\xi_{ij}} = 1$ with probability $\frac{1}{\delta}$ (line 7).

4. To ensure $(\widehat{\xi_{ij}}, \widehat{y})$ is a feasible solution to ILP_{LBPNA}, repeat steps 2 and 3 until every leaf node is assigned a non-leaf node.

Subsequently, we analyze the approximation factor of Algorithm 9 in Theorem 11.

Theorem 11. Let opt_{LBPNA} denote the optimal solution of LBPNA. Algorithm 9 yields an optimal fractional solution of $O(\Delta \log(n))(opt_{LBPNA} + \lceil \frac{B}{\mathcal{R}} \rceil)$ with probability 1.

Proof. Considering any non-leaf node v_i and leaf node v_j, the expected actual load of v_i is as follows:

$$E[\sum_{j:\beta_i \widehat{\xi_{ij}} > 0} \lceil \frac{B}{\gamma_i} \rceil \frac{1}{\iota_{ij}} + 12 \lceil \frac{B}{\mathcal{R}} \rceil] \geq E[\sum_{j:\beta_i \widehat{\xi_{ij}} > 0} \lceil \frac{B}{\gamma_i} \rceil] + 12 \lceil \frac{B}{\mathcal{R}} \rceil. \tag{7.17}$$

Inequality (7.17) holds because $0 \leq \iota_{ij} \leq 1$. From the above formula, we know the expected α_i of the non-leaf node v_i mainly depends on the number of the children nodes of v_i. Hence, we have:

$$
\begin{aligned}
& E[\sum_{j:\beta_i\widehat{\xi_{ij}}>0} \lceil\frac{B}{\gamma_i}\rceil\frac{1}{\iota_{ij}} + 12\lceil\frac{B}{\mathcal{R}}\rceil] && (7.18)\\
& \geq \lceil\frac{B}{\gamma_i}\rceil E[\sum_{v_j\in\mathbb{N}_1(v_i)} \beta_i\widehat{\xi_{ij}}] + 12\lceil\frac{B}{\mathcal{R}}\rceil && (7.19)\\
& = \lceil\frac{B}{\gamma_i}\rceil \sum_{v_j\in\mathbb{N}_1(v_i)} \beta_i E[\widehat{\xi_{ij}}] + 12\lceil\frac{B}{\mathcal{R}}\rceil && (7.20)\\
& \geq \lceil\frac{B}{\gamma_i}\rceil \sum_{v_j\in\mathbb{N}_1(v_i)} \beta_i(\xi^*_{ij} + \frac{1}{\delta}) + 12\lceil\frac{B}{\mathcal{R}}\rceil && (7.21)\\
& = \lceil\frac{B}{\gamma_i}\rceil \sum_{v_j\in\mathbb{N}_1(v_i)} \beta_i\xi^*_{ij} + \lceil\frac{B}{\gamma_i}\rceil\frac{1}{\delta} \sum_{v_j\in\mathbb{N}_1(v_i)} \beta_i + 12\lceil\frac{B}{\mathcal{R}}\rceil && (7.22)\\
& \geq \vartheta^* + \lceil\frac{B}{\gamma_i}\rceil\frac{\delta}{\delta} \geq \vartheta^* + \lceil\frac{B}{\mathcal{R}}\rceil && (7.23)\\
& = \vartheta^* + \varphi, \text{ where } \varphi = \lceil\frac{B}{\mathcal{R}}\rceil. && (7.24)
\end{aligned}
$$

Since $0 \leq \iota_{ij} \leq 1$, we obtain the Inequality (7.18). Equation (7.20) holds because β_i and $\widehat{\xi_{ij}}$ are independent. Inequality (7.21) holds since the random rounding technique used in Algorithm 9. More specifically, the procedure, a non-leaf node v_i is assigned to be the parent node of a leaf node v_j (*i.e.*, $\widehat{\xi_{ij}} = 1$) with probability ξ^*_{ij}, is repeated κ times. Moreover, the same procedure is run one more time with probability $\frac{1}{\delta}$. This implies $E[\widehat{\xi_{ij}}] \geq \xi^*_{ij} + \frac{1}{\delta}$. Inequality (7.23) holds because there is at least δ leaf nodes in a non-leaf node's 1-Hop Neighborhood.

Applying the Chernoff bound, we obtain the following bound:

$$
Pr[\sum_{j:\beta_i\widehat{\xi_{ij}}>0} \lceil\frac{B}{\gamma_i}\rceil\frac{1}{\iota_{ij}} + 12\lceil\frac{B}{\mathcal{R}}\rceil \leq (1-\lambda)\kappa(\vartheta^* + \varphi)] \leq (\frac{e^{-\lambda}}{(1-\lambda)^{1-\lambda}})^{\kappa(\vartheta^*+\varphi)}
$$

for arbitrary $0 < \lambda < 1$. To simplify this bound, suppose $\lambda = \frac{1}{\Delta}$, then

$$Pr[\sum_{j:\beta_i\widehat{\xi_{ij}}>0} \lceil\frac{B}{\gamma_i}\rceil\frac{1}{\iota_{ij}} + 12\lceil\frac{B}{\mathcal{R}}\rceil \leq (1-\lambda)\kappa(\vartheta^* + \varphi)] \tag{7.25}$$

$$\leq [e^{-\lambda}(1-\lambda)^{\lambda-1}]^{\kappa(\vartheta^*+\varphi)} \tag{7.26}$$

$$\leq e^{-\lambda\kappa(\vartheta^*+\varphi)}(1-\lambda)^{(\lambda-1)\kappa(\vartheta^*+\varphi)} \tag{7.27}$$

$$\leq e^{-\lambda\kappa(\vartheta^*+\varphi)}. \tag{7.28}$$

Since $\vartheta^* = \min\{\sum_{j:\beta_i\xi^*_{ij}>0} \lceil\frac{B}{\gamma_i}\rceil\frac{1}{\iota_{ij}} + 12\lceil\frac{B}{\mathcal{R}}\rceil \mid \forall v_i \in \mathbb{D}\} \geq 1$, we have

$$Pr[\sum_{j:\beta_i\widehat{\xi_{ij}}>0} \lceil\frac{B}{\gamma_i}\rceil\frac{1}{\iota_{ij}} + 12\lceil\frac{B}{\mathcal{R}}\rceil \leq (1-\lambda)\kappa(\vartheta^* + \varphi)] \leq e^{-\lambda\kappa(\vartheta^*+\varphi)} \tag{7.29}$$

$$\leq e^{-\lambda\kappa} = e^{-\frac{3\Delta\log(n)}{\Delta}} \tag{7.30}$$

$$\leq \frac{1}{e^{3\ln(n)}} \leq \frac{1}{n^3}. \tag{7.31}$$

Summing over all non-leaf nodes $v_i \in \mathbb{D}$, we obtain the probability that some non-leaf node has the actual load less than $(1-\lambda)\kappa(\vartheta^* + \varphi)$ as follows:

$$Pr[\widehat{\vartheta} \leq (1-\lambda)\kappa(\vartheta^* + \varphi)] = n\frac{1}{n^3} = \frac{1}{n^2}. \tag{7.32}$$

Subsequently, we consider the probability that a leaf node $v_j \in \mathbb{V}\backslash\mathbb{D}$ is not assigned a parent non-leaf node in its 1-hop neighborhood at iteration j. That is,

$$\prod_{v_i\in\mathbb{N}_1(v_j),\ v_i\in\mathbb{D}} Pr[\beta_i\widehat{\xi_{ij}} = 0 \text{ at iteration } j] \tag{7.33}$$

$$= \prod_{v_i\in\mathbb{N}_1(v_j),\ v_i\in\mathbb{D}} (1-\beta_i\xi^*_{ij}) \tag{7.34}$$

$$\leq \prod_{v_i\in\mathbb{N}_1(v_j),\ v_i\in\mathbb{D}} e^{-\beta_i\xi^*_{ij}} \tag{7.35}$$

$$= e^{-\sum_{v_i\in\mathbb{N}_1(v_j),\ v_i\in\mathbb{D}}\beta_i\xi^*_{ij}} = \frac{1}{e}. \tag{7.36}$$

Inequality (7.35) results from the inequality $(1-x) \leq e^{-x}$, $\forall x \in [0,1]$. Equation (7.36) follows the fact that each leaf node can be allocated to only one non-leaf node, *i.e.*, $\sum\limits_{v_i \in \mathbb{N}_1(v_j)} \beta_i \xi^*_{ij} = 1$, $\forall v_j \notin \mathbb{D}$, $\forall v_i \in \mathbb{D}$. Now, the probability that a leaf node is not assigned a parent non-leaf node in its 1-hop neighborhood after the random rounding is $e^{-\kappa}$, which implies $Pr[a\ leaf\ node\ has\ no\ neighboring\ non\text{-}leaf\ node] \leq ne^{-\kappa} = n\frac{1}{n^3} = \frac{1}{n^2}$. Then, considering Inequality (7.32), we have $Pr[each\ leaf\ node\ is\ assigned\ to\ a\ parent\ non\text{-}leaf\ node \bigcap \widehat{y} \geq (1-\lambda)\kappa(\vartheta^* + \varphi)] \geq (1-\frac{1}{n^2})(1-\frac{1}{n^2}) \sim 1$, *when* $n \sim \infty$, for $\lambda = \frac{1}{\Delta}$, and $\kappa = 3\Delta \log(n)$. ■

From theorem 11, the solution of our proposed random approximation Algorithm 9 yields an solution upper bounded by $O(\Delta)(opt_{LBPNA} + \lceil \frac{B}{\mathcal{R}} \rceil)$. Moreover, this bound can be verified in polynomial time.

After $\mathcal{A}^*$ is decided, a tree can be obtained by assigning each link a direction from the children to the parent.

7.6 Performance Evaluation

Since there are no existing works studying the LBDAT construction problem for s-tochastic WSNs currently, in the simulations, the results of LBDAT are compared with the recently published DS-based data aggregation algorithm [44] denoted by DAT. We compare both algorithms in terms of the number of non-leaf nodes, and network lifetime, which is defined as the time duration until the first non-leaf node runs out of energy.

We build our own simulator where all the nodes have the same transmission range and all the nodes are deployed uniformly and randomly in a square area. For each specific setting, 100 instances are generated. The results are averaged over these 100 instances (all results are rounded to integers). Moreover, a random value between $[0.5, 0.98]$ is assigned to the Transmission Success Ratio (ι_{ij}) value associated to a pair of nodes (v_i and v_j) inside the transmission range. Otherwise, a random value between $(0, 0.5]$ is assigned to ι_{ij} associated to a pair of nodes beyond the transmission range. Furthermore, every sensor node produces

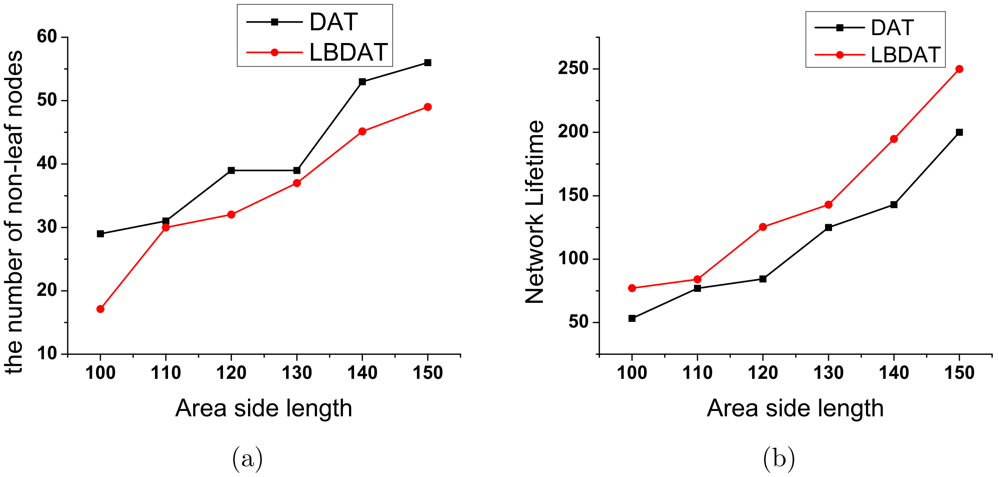


Figure 7.4. Simulation results: (a) The number of non-leaf node; (b) Network Lifetime.

a packet with size one (*i.e.*, $B = 1$) during each report time interval. The data receiving rate γ_i of each node v_i is randomly generated from the value between $(0, 10]$. The energy consumption model is that every node has the same initial 1000 units of energy. Receiving and transmitting a packet both consume 1 unit of energy. Additionally, all nodes have the same transmission range of $20m$ and 100 nodes are deployed uniformly and randomly in a square area. The side length of the square area is incremented from 100 to 150 by 10. The simulation results are presented in Fig.7.4.

From Fig.7.4 (a), we can see that, with the increase of the area of the network deployed region, the number of non-leaf nodes increases for both algorithms (DAT, and LBDAT). This is because the stochastic WSN becomes thinner, more non-leaf nodes are needed to maintain the connectivity of the constructed CMIS. There is no obvious trend that which algorithm might produce more non-leaf nodes when constructing a DAT.

From Fig.7.4 (b), we know that the network lifetime increases for both algorithms with the side length of the deployed area increasing. It is obvious that the density of the network becomes more thinner with the side length of the deployed area increasing. As to data aggregation, the thinner the network is, the less number of neighbors of each non-leaf node. In other words, the aggregated data are less on each non-leaf node when the network becomes thinner. Hence, network lifetime is increasing for both algorithms. Additionally, LBDAT prolongs network lifetime by 32% on average compared with DAT. The results demonstrate that load-balanced parent node assignments can improve network lifetime significantly.

7.7 Summary

In this chapter, we address fundamental problems of constructing a load-balanced DAT in stochastic WSNs. We first solve the CMIS problem, which is NP-Complete, with two phases. In the first phase, we aim to find the optimal MIS such that the minimum *potential load* of all the independent nodes is maximized. To this end, a near optimal approximation algorithm is proposed, which yields an $O(\Delta \ln(n))$ approximation factor. In the second phase, the minimum-sized set of LBMIS connectors are found to make the LBMIS connected. The theoretical lower and upper bounds of the number of non-leaf nodes are analyzed as well. Subsequently, we study the LBDAT construction problem which is to find a load-balanced Parent Node Assignment (LBPNA) with an objective to maximize the minimum *actual load* of all the non-leaf nodes. Since this problem is NP-hard, we propose an approximation algorithm by using linear relaxing and random rounding techniques, which yields a solution of $O(\Delta \log(n))$ approximation factor of actual traffic load on each non-leaf node. After LBPNA is decided, by assigning a direction to each link, we obtain an LBDAT. Simulations show that the proposed algorithms can extend network lifetime significantly.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusions

In this dissertation, we first briefly introduce the background knowledge of WSNs and the topology control techniques in WSNs. Since sensor nodes are tightly constrained in terms of energy, processing, and storage capacities, restricting topology in WSNs is very challenging due to these inherent characteristics that distinguish WSNs from other wireless networks. Due to such difference, many new algorithms have been proposed for controlling topology in WSNs. Connected Dominating Set (CDS) based topology control which is one kind of hierarchical methods has received more attention to reduce redundant and unnecessary communication overhead. Having such a CDS reduces network topology by restricting the main communication tasks to the dominators only. Then, we summarize the CDS constructing algorithms in Chapter 2.

Unfortunately, to the best of our knowledge, all the related work did not consider the *load-balance* factor when they construct a CDS. In chapter 3, we propose a new concept the Load-Balanced CDS (LBCDS) and a new problem the Load-Balanced Allocate Dominatee (LBAD) problem. Consequently, a greedy-based approximation algorithm is proposed to construct an LBCDS in a WSN. Moreover, we propose an optimal centralized algorithm and an efficient probability-based distributed algorithm to solve the LBAD problem. If there is a given CDS constructed by any method, the upper bound and lower bound of the performance ratio of the distributed algorithm are analyzed in the dissertation. Through extensive simulations, we demonstrate that our proposed methods extend network lifetime by 80% compared with the best and latest CDS construction algorithm.

In chapter 4, we investigate constructing an LBCDS and load-balancedly allocating dominatees to dominators simultaneously. A Genetic Algorithm (GA) based strategy called

LBCDS-GA is proposed to construct an LBCDS in a WSN. As a matter of fact, constructing a CDS as a virtual backbone in a WSN is an efficient way to extend network lifetime through reducing the number of the nodes involved in communication, while building an LBCDS and load-balancedly allocating dominatees to dominators can further prolong network lifetime through balancing the workloads of all the dominators. We also demonstrate by simulation that our proposed method extend network lifetime by 65% on average compared with the best and latest MCDS construction algorithm. On the other hand, in the current related literature, network are deterministic where two nodes are assumed either connected or disconnected. In most real applications, however, there are many intermittently connected wireless links called lossy links, which only provide probabilistic connectivity. For WSNs with lossy links, we propose a Stochastic Network Model (SNM). Under this model, we measure the quality of CDSs using CDS reliability defined as the minimum upper limit of the node-to-node delivery ratio between any pair of dominators in a CDS.

In chapter 5, we attempt to construct a MCDS while its reliability is above a preset application-specified threshold, called Reliable MCDS (RMCDS). We claim that constructing a RMCDS is NP-Hard under the SNM model. We propose a novel Genetic Algorithm (GA) with immigrants schemes called RMCDS-GA to solve the RMCDS problem. To evaluate the performance of RMCDS-GA, we conduct comprehensive simulations. The simulation results show that compared with the traditional MCDS algorithms, RMCDS-GA can construct a more reliable CDS without increasing the size of a CDS.

In chapter 6, we perform comprehensive performance ratio analysis of the Load-Balanced Virtual Backbone (LBVB) construction algorithms. To be specific, the MinMax Degree Maximal Independent Set (MDMIS) problem, the Load-Balanced Virtual Backbone (LBVB) problem, and the MinMax Valid-Degree non Backbone node Allocation (MVBA) problem are investigated and analyzed. We claim that MDMIS and LBVB are NP-Complete problems and MVBA is an NP-Hard problem. Approximation algorithms and comprehensive theoretical analysis of the approximation factors are presented in the chapter. Moreover, our theoretical analysis and simulation results show that the proposed algorithms outperform the existing state-of-the-art approaches.

In chapter 7, we apply the constructed LBCDS to a practical application under the realistic SNM model, namely data gathering. Data Gathering is a fundamental task in WSNs. For applications where each sensor continuously monitors the environment and periodically reports to the sink, a tree-based topology is often used to gather and aggregate sensing data from sensor nodes. Thus, data gathering trees with aggregation are also referred to as Data Aggregation Trees (DATs). To be specific, a new problem, Load-Balanced Data Aggregation Tree (LBDAT), is introduced finally. Our simulation results show that the proposed algorithms outperform the existing state-of-the-art approaches significantly.

8.2 Future Works

There are several directions for our next step works.

- Consider *mobility* when constructing an LBVB.
- Design distributed algorithms for LBVB construction.
- Consider different packages arriving on each node, which is more realistic than our current single package assumption, when constructing an LBVB.
- Consider *Linear Fractional Programming* to get a tighter performance ratio.
- Consider not only network lifetime, but also time delay, and other performances simultaneously.

REFERENCES

[1] M. Zuniga and B. Krishnamachari, *Analyzing the transitional region in low power wireless links*, SECON, Santa, Clara, Oct. 4-7, 2004.

[2] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic, *Impact of radio irregularity on wireless sensor networks*, Mobisys, Boston, Massachusetts, Jun. 6-9, 2004.

[3] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, *Temporal properties of low power wireless links: Modeling and implications on multi-hop routing*, MobiHoc, Nrbana-Champaign, IL, May 25-28, 2005.

[4] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, *Statistical model of lossy links in wireless sensor networks*, IPSN'05, Los Angeles, CA, 2005.

[5] Y. Liu, Q. Zhang, L.-M. Ni, *Opportunity-Based Topology Control in Wireless Sensor Networks*, TPDS'10, 21(9), 2010.

[6] R. Rajaraman, *Topology Control and Routing in Ad Hoc Networks: A Survey*, ACM SIGACT News, 33(2): 60C73, 2002.

[7] A. Ephremides, J. Wieselthier, and D. Baker, *A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling*, proceeding of the IEEE, 75(1):56-73, 1987.

[8] S. Guha, and S. Khuller, *Approximation Algorithms for Connected Domimating Sets*, Algorithmica, vol. 20, pp. 374-387, 1998.

[9] J. Wu, and H. Li, *On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks*, DIALM, pp. 7-14, 1999.

[10] F. Dai, and J. Wu, *An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks*, TPDS, 15(10): 908-920, 2004.

[11] P. Wan, K. Alzoubi, and O. Frieder, *Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks*, Mobile Networks and Applications, 9(2): 141-149, 2004.

[12] M. Cardei, M. Cheng, X. Cheng, and D. Zhu, *Connected Dominating Set in Ad Hoc Wireless Networks*, Int'l Conf. Computer Science and Informatics, 2002.

[13] Y. Li, M. Thai, F. Wang, C. Yi, P. Wang, and D. Du, *On Greedy Construction of Connected Dominating Sets in Wireless Networks*, Wireless Comm. and Mobile Computing, 5(8): 927-932, 2005.

[14] K. Alzoubi, P. Wan, and O. Frieder, *Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks*, MobiHoc, 2002.

[15] Y. Li, S. Zhu, M. Thai, and D. Du, *Localized Construction of Connected Dominating Set in Wireless Networks*, TAWN, Chicago, June, 2004.

[16] R. Misra, and C. Mandal, *Rotation of CDS via Connected Domatic Partition in Ad Hoc Sensor Networks*, TMC, 8(4), 2009.

[17] D. Kim, W. Wang, X. Li, Z. Zhang, and W. Wu, *A New Constant Factor Approximation for Computing 3-Connected m-Dominating Sets in Homogeneous Wireless Networks*, INFOCOM, 2010.

[18] L. Ding, X. Gao, W. Wu, W. Lee, X. Zhu, and D. Z. Du, *Distributed Construction of Connected Dominating Sets with Minimum Routing Cost in Wireless Networks*, ICDCS, 2010.

[19] D. kim, Y. Wu, Y. Li, F. Zou, and D. Z. Du, *Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks*, TPDS, 20(2), 2009.

[20] S. Biswas, and R. Morris, *Opportunistic Routing in Multi-Hop Wireless Networks*, SIGCOMM, 2005.

[21] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, *Trading Structure for Randomness in Wireless Opportunistic Routing*, SIGCOMM, 2007.

[22] J. Ma, Q. Zhang, C. Qian, and L. Ni, *Energy-efficient opportunistic topology control in wireless sensor networks*, ACM/SIGMOBILE MobiOpp, 2007.

[23] J. Ma, C. Qian, Q. Zhang, and L. Ni, *Opportunistic Transmission Based QoS Topology Control in Wireless Sensor Networks*, MASS, 2008.

[24] Y. Liu, Q. Zhang, and L. Ni, *Opportunity-based Topology Control in Wireless Sensor Networks*, ICDCS, Beijing, China, June, 2008.

[25] Y. Liu, and L. Ni, *A Generalized Probabilistic Topology Control for Wireless Sensor Networks*, INFOCOM Minisymposium, Rio de Janeiro, Brazil, Apr., 2009.

[26] M. Al-Obaidy, A. Ayesh, and A. Sheta, *Optimizing the Communication Distance of an Ad Hoc Wireless Sensor Networks by Genetic Algorithms*, Artifical Intelligence, vol. 29, pp. 183-194, 2008, .

[27] J. Wang, C. Niu, and R. Shen, *Priority-based Target Coverage in Directional Sensor Networks using a Genetic Algorithm*, Computers and Mathematics with Applications, vol. 57, pp. 1915-1922, 2009.

[28] A. Bari, S. Wazed, A. Jaekel, and S. Bandyopadhyay, *A Genetic Algorithm Based Approach for Energy Efficient Routing in Two-Tiered Sensor Networks*, Ad Hoc Networks, pp. 665-676, 2009.

[29] X. Hu, J. Zhang, Y. yu, H. Chung, Y. Lim Y. Shi, and X. Luo, *Hybrid Genetic Algorithm Using a Forward Encoding Scheme for Lifetime Maximization of Wireless Sensor Networks*, ITEC'10, 14(5):766 - 781, October, 2010, 2010.

[30] S. Ji, Y. Li, and X. Jia, *Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks for Continuous Data Collection*, Infocom 2011, Shanghai, China, Apr. 10-15, 2011.

[31] B. K. Polat, P. Sachdeva, M. H. Ammar, and E. W. Zegura, *Message Ferries as Generalized Dominating Sets in Intermittently Connected Mobile Networks*, Proceedings of the Second International Workshop on Mobile Opportunistic Networking, MobiOpp'10, Pisa, Italy, February 22-23, 2010.

[32] B. Deb, S. Bhatnagar, and B. Nath, *Multi-resolution state retrieval in sensor networks*, Proc. of IEEE International Workshop on Sensor Network Protocols and Applications, 2003.

[33] J. He, S. Ji, Y. Pan, and Y. Li, *Greedy Construction of Load-Balanced Virtual Backbones in Wireless Sensor Networks*, to appear in the Wireless Communications and Mobile Computing (WCMC), 2012.

[34] J. He, S. Ji, M. Yan, Y. Pan, and Y. Li, *Load-Balanced CDS Construction in Wireless Sensor Networks Via Genetic Algorithm*, International Journal of Sensor Networks (IJSNET), Vol. 11, No. 3, pp(s): 166-178, 2012.

[35] J. He, S. Ji, P. Fan, Y. Pan, and Y. Li, *Constructing a Load-Balanced Virtual Backbone in Wireless Sensor Networks*, ICNC 2012, Maui, Hawaii, USA, January 30-February 2, 2012.

[36] M. Yan, J. He, S. Ji, and Y. Li, *Minimum Latency Scheduling for Multi-Regional Query in Wireless Sensor Networks*, IPCCC 2011, Orlando, Florida, USA, November 17- 19, 2011.

[37] J. He, S. Ji, M. Yan, Y. Pan, and Y. Li, *Genetic-Algorithm-Based Construction of Load-Balanced CDSs in Wireless Sensor Networks*, MILCOM 2011, Baltimore, Maryland, USA, November 7-10, 2011.

[38] J. He, S. Ji, Y. Pan, and Z. Cai, *A Genetic Algorithm with Immigrants Schemes for Constructing a σ-Reliable MCDS in Probabilistic Wireless Networks*, to appear in the Ad Hoc & Sensor Wireless Networks (AHSWN), 2012.

[39] J. He, Z. Cai, S. Ji, R. Beyah, and Y. Pan, *A Genetic Algorithm for Constructing a Reliable MCDS in Probabilistic Wireless Networks*, WASA 2011, Chengdu, China, August 11-13, 2011.

[40] H.M. Ammari, and J. Giudici, *On the Connected k-Coverage Problem in Heterogeneous Sensor Nets: The Curse of Randomness and Heterogeneity*, 29th IEEE International Conference on Distributed Computing Systems (ICDCS'09), Montreal, Québec, Canada, June 22-26, 2009.

[41] J. He, S. Ji, Y. Pan, and Y. Li, *Reliable and Energy Efficient Target Coverage for Wireless Sensor Networks*, Tsinghua Science and Technology Special Issue on Wireless Mobile Computing and Networking, Vol. 16, No. 5, pp: 464-474, October, 2011.

[42] J. He, N. Xiong, Y. Xiao, and Y. Pan, *A Reliable Energy Efficient Algorithm for Target Coverage in Wireless Sensor Networks*, ICDCS workshop on Sensor Networks, Genoa, Italy, June 21-25, 2010.

[43] Z. Cai, S. Ji, J. He, and A.G. Bourgeois, *Optimal Distributed Data Collection for Asynchronous Cognitive Radio Networks*, ICDCS 2012, Macau, China, June 18-21, 2012.

[44] P.J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, X. Jia, *Minimum-latency aggregation scheduling in multihop wireless networks*, MobiHoc 2009, New Orleans, Louisiana, May 18-21, pp. 185-194, 2009.

[45] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, WH Freeman & Co., New York, 1979.

[46] M. Kalantari, M. Haghpanahi, and M. Shayman, *A p-norm Flow Optimization Problem in Dense Wireless Sensor Networks*, INFOCOM'08, Phoenix, AZ, USA, April 13-18, 2008.

[47] S. R. Kulkarni, and P. Viswanath, A Deterministic Approach to Throughput Scaling in Wireless Networks, *TIT*, 2004.

[48] G. Hadim, and N. Mohamed, *Middleware Challenges and Approaches for Wireless Sensor Networks*, IEEE Distributed Systems, 7(3): 1-1, 2006.

[49] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, *The Broadcast Storm Problem in a Mobile Ad Hoc Network*, MOBICOM, Washington, pp. 152-162, 1999.

[50] B. Das, and V. Bharghavan, *Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets*, ICC, 1997.

[51] B. Deb, S. Bhatnagar, and B. Nath, *Multi-resolution state retrieval in sensor networks*, IWSNPA, 2003.

[52] H.M. Ammari, and J. Giudici, *On the Connected k-Coverage Problem in Heterogeneous Sensor Nets: The Curse of Randomness and Heterogeneity*, ICDCS, Montreal, Quebec, Canada, June 22-26, 2009.

[53] S. Ji, A.S. Uluagac, R. Beyah, and Z. Cai, *Practical Unicast and Convergecast Scheduling Schemes for Cognitive Radio Networks*, To appear in the Journal of Combinatorial Optimization (JCO), 2012.

[54] Z. Cai, S. Ji, and J. Li, *Data Caching Based Query Processing in Multi-Sink Wireless Sensor Networks*, International Journal of Sensor Networks (IJSNet), Vol. 11, No. 2, pp(s): 109-125, 2012.

[55] Y. Duan, S. Ji, and Z. Cai, *Generalized Broadcast Scheduling in Duty-Cycle Multi-Hop Wireless Networks*, IPCCC 2011, Orlando, Florida, USA, November 17-19, 2011.

[56] S. Ji, Z. Cai, Y. Li, and X. Jia, *Continuous Data Collection Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks*, To appear in the IEEE Transactions on Parallel and Distributed Systems (TPDS), 2011.

[57] S. Ji, R. Beyah, and Z. Cai, *Snapshot/Continuous Data Collection Capacity for Large-Scale Probabilistic Wireless Sensor Networks*, IEEE INFOCOM 2012, Orlando, Florida USA, March 25-30, 2012.

[58] S. Ji, and Z. Cai, *Distributed Data Collection and Its Capacity in Asynchronous Wireless Sensor Networks*, IEEE INFOCOM 2012, Orlando, Florida USA, March 25-30, 2012.

[59] S. Ji, R. Beyah, and Y. Li, *Continuous Data Collection Capacity of Wireless Sensor Networks Under Physical Interference Model*, IEEE MASS 2011, Valencia, Spain, October 17-22, 2011.

[60] S. Ji, Y. Li, and X. Jia, *Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks for Continuous Data Collection*, IEEE INFOCOM 2011, Shanghai, Chian, April 10-15, 2011.

[61] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, 1989.

[62] W. Di, Q. Yan, and T. Ning, *Connected dominating set based hybrid routing algorithm in ad hoc networks with obstacles*, ICC'06, Turkey, Istanbul, June 11-15, 2006.

[63] E. H. Wassim, A. F. Ala, G. Mohsen, and H. H. Chen, *On Efficient Network Planning and Routing in Large-Scale MANETs*, IEEE Transactions on Vehicular Technology, Vol. 58, No. 7, September, 2009.

[64] L. Ding, Y. Shao, and M. Li, *On reducing broadcast transmission cost and redundancy in ad hoc wireless networks using directional antennas*, WCNC'08, pp. 2295-2300, 2008.

[65] A. Helmy, S. Garg, P. Pamu, and N. Nahata, *CARD: A Contact-based Architecture for Resource Discovery in Ad Hoc Networks*, MONET, 10(1):99-113, 2004.

[66] K.P. Shih, D.J. Deng, R.S. Chang, and H.C. Chen, *On Connected Target Coverage for Wireless Heterogeneous Sensor Networks with Multiple Sensing Units*, Sensors, Vol. 9, No. 7, pp. 5173-5200, 2009.

[67] K.-W. Chin, J. Judge, A. Williams, and R. Kermode, *Implementation experience with manet routing protocols*, ACM SIGCOMM Computer Communications Review, vol. 32, pp. 49-59, 2002.

[68] A. Agrawal. and R. Barlow, *A Survey of Network Reliability and Domination Theory*, Operations Research, 32:298-323, 1984.

[69] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. Stankovic, *Atpc: Adaptive transmission power control for wireless sensor networks*, Sensys'06, Boulder, Colorado, Oct. 31 - Nov. 3, 2006.

[70] D. Son, B. Krishnamachari, and J. Heidemann, *Experimental study of concurrent transmission in wireless sensor networks*, SenSys'06, Boulder, Colorado, Oct. 31 - Nov. 3, 2006.

[71] J. H. Holland, *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, MI, 1975.

[72] B. Karp, H. Kung, *GPSR: greedy perimeter stateless routing for wireless networks*, MobiCom'00, New York, 2000.

[73] J. Grefenstette, *Genetic Algorithms for changing environments*, Second International Conference on Parallel Problem Solving from Nature, pp. 137-144, 1992.

[74] S. Yang, and X. Yao, *A hybrid immigrants scheme for genetic algorithms in dynamic environments*, JAC, vol. 4, no. 3, pp.243-254, 2007.

[75] X. Yu, K. Tang, and X. Yao, *An immigrants scheme based on environmental information for genetic algorithms in changing environmentt*, Proceedings of the 2008 Congree on Evolutionary Computing, pp. 1141-1147, 2008.

[76] X. Yu, K. Tang, T. Chen, and X. Yao, *Empirical analysis of evolutionary algorithms with immigrants schemes for dynamic optimization*, Memetic Computing, vol. 1, no. 1, pp. 3-24, 2009.

[77] M. Garey, and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1983.

[78] S. Gueyea, P. Michelonb, *A linearization framework for unconstrained quadratic (0-1) problems*, Discrete Applied Mathematics 157:1255 - 1266, 2009.

[79] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, *On Network Correlated Data Gathering*, INFOCOM, pp. 2571-2582, 2004.

[80] S. Madden, R. Szewczyk, M.J. Franklin, and D. Culler, *Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks*, Workshop on Mobile Computing and Systems Applications, 2002.

[81] H.O. Tan, and I. Korpeogle, *Power Efficient Data Gatering and Aggregation in Wireless Sensor Networks*, SIGMOD Record, 32(3):66-71, 2003.

[82] H.O. Tan, I. Korpeoglu, and I. Stojmenovic, *Computing Localized Power-Efficient Data Aggregation Trees for Sensor Networks*, TPDS'11.

[83] X. Chen, X. Hu, and J. Zhu, *Minimum Data Aggregation Time Problem in Wireless Sensor Networks*, Lecture Notes in Computer Science, 3794:133-142, 2005.

[84] C. Huang, P. Wan, T. Vu, Y. Li, and F. Yao, *Nearly Constant Approximation for Data Aggregation Scheduling in Wireless Sensor Networks*, INFOCOM, 2007.

[85] X. Xu, X. Li, X. Mao, S. Tang, and S. Wang, *A Delay-Efficient Algorithm for Data Aggregation in Multihop Wireless Sensor Networks*, TPDS, 2011.

[86] Y. Xue, Y. Cui, and K. Nahrstedt, *Maximizing Lifetime for Data Aggregation in Wireless Sensor Networks*, MONET, 2005.

[87] H. Lin, F. Li, and K. Wang *Constructing Maximum-Lifetime Data Gathering Trees in Sensor Networks with Data Aggregation*, ICC, 2010.

[88] K. Kalpakis, and S. Tang, *A Combinatorial Algorithm for the Maximum Lifetime Data Gathering with Aggregation Problem in Sensor Networks*, Computer Communication, 2009.

[89] D. Virmani, and S. Jain, *Construction of Decentralized Lifetime Maximizing Tree for Data Aggregation in Wireless Sensor Networks*, World Academy of Science, Engineering and Technology.

[90] D. Luo, X. Zhu, X. Wu, and G. Chen, *Maximizing Lifetime for the Shortest Path Aggregation Tree in Wireless Sensor Networks*, INFOCOM, 2011.