A Multi-Objective Genetic Algorithm for Constructing Load-Balanced Virtual Backbones in Probabilistic Wireless Sensor Networks

Jing (Selena) He[†], Shouling Ji[‡,♯], Raheem Beyah[♯], and Yingshu Li[‡,♮]
[†]Department of Computer Science, Kennesaw State University, Kennesaw, Georgia 30144, USA
[‡]Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
[♯]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA
[♮]School of Computer Science and Technology, Heilongjiang University, Harbin, 150080, China
Email: jhe4@kennesaw.edu, sji@gatech.edu, rbeyah@ece.gatech.edu, yli@cs.gsu.edu

***Abstract*—A Connected Dominating Set (CDS) is used as a Virtual Backbone (VB) for efficient routing and broadcasting in Wireless Sensor Networks (WSNs). Currently, almost all existing works focus on constructing Minimum-sized CDS under the Deterministic Network Model (DNM). However, due to the existence of many probabilistic *lossy links* in WSNs, it is more practical to obtain a VB under the realistic Probabilistic Network Model (PNM). Moreover, *load-balance* factor cannot be neglected when constructing a VB to prolong network lifetime. Hence, in this paper, we propose a Multi-Objective Genetic Algorithm (MOGA) to construct a Load-Balanced Virtual Backbone under PNM (LBVBP). Through simulations, we demonstrate that our proposed methods extend network lifetime by 65% on average compared with the existing state-of-the-art approaches.**

I. Introduction

Wireless Sensor Networks (WSNs) are emerging as the desired environment for increasing numbers of military and civilian applications, such as, disaster control, environment and habitat monitoring, battlefield surveillance, and health care applications [1]. Due to the infrastructure-less and dynamic nature in WSNs, most routing protocols in WSNs (*i.e.*, flooding) usually cause a serious broadcasting storm. A Connected Dominating Set (CDS) has been a well known approach for constructing a Virtual Backbone (VB) to alleviate the broadcasting storm thus improving the performance and increase the efficiency of routing protocols in WSNs. A Dominating Set (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 nodes in the set. If the induced subgraph by the nodes in a DS is connected, then this DS is called a CDS. The nodes in a CDS are called *dominators* denoted by set $\mathbb{B}$, otherwise, *dominatees* denoted by set $\mathbb{W}$.

Ever since the idea of employing a CDS as a VB for WSNs is introduced [2], a huge amount of approximation algorithms [3] have been proposed to construct an MCDS-based VB, which is a well-known NP-Hard problem [4]. After that, to make a CDS-based VB more resilient in mobile WSNs, the fault-tolerance of a VB is considered. In [5], k-connected and m-dominated sets are introduced as a generalized abstraction of a fault-tolerance VB. In [6], 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 qualified CDS in terms of size, diameter, and Average Backbone Path Length (ABPL) in [7].

Unfortunately, all of the above mentioned works are based on the ideal Deterministic Network Model (DNM), where any pair of nodes in a network is either fully connected or completely disconnected. In most real applications, however, the DNM model cannot fully characterize the behavior of wireless links due to the *transitional region phenomenon* [8]. Since beyond the "always connected" region, there is a *transitional region* where a pair of nodes are probabilistically connected via the so called *lossy links* [8]. As reported in [8], [9], there are often much more lossy links than fully connected links in a WSN. Therefore, a more practical network model for WSNs is the Probabilistic Network Model (PNM). Under this model, there is a *delivery ratio* (γ_{ij}) associated with each link connecting a pair of nodes v_i and v_j, which is used to indicate the probability that v_i can successfully deliver a package to v_j (an example is shown in Fig. 1). For convenience, the WSNs considered under the DNM/PNM are called deterministic/probabilistic WSNs.

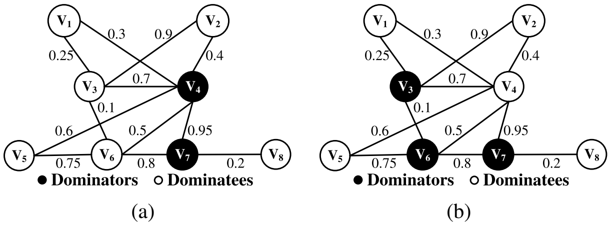


Fig. 1. Illustration of a regular VB and an LBVB.

On the other hand, all the aforementioned works did not consider the *load-balance* factor when they construct a CDS. Without considering balancing the traffic load among the nodes on each dominator, some heavy loaded nodes may quickly exhaust their energy (such as dominator v_4 shown in Fig. 1(a)), which might cause network partitions or malfunctions. To benefit from the CDS-based VB in WSNs and also take the load-balance factor into consideration, we constructed VBs in this manner in [10]–[12]. We proposed a greedy algorithm to build a Load-Balanced CDS (LBCDS) and then Load-Balancedly Allocate Dominatees (LBAD) based on *Expected Allocation Probability* for deterministic WSNs in [10]. Further, the performance ratio of the proposed algorithm are analyzed in [11]. Subsequently, we investigated the LBCDS and LBAD problems simultaneously for deterministic WSNs, and proposed a Genetic Algorithm to solve it in [12]. However, our previous works studied the Load-Balanced VB (LBVB) construction problem under the DNM rather than the more practical PNM, *i.e.*, for probabilistic WSNs. Actually, how to measure the traffic load on each node for probabilistic WSNs is different. For example, in Fig. 1, if $\gamma_{46} = 0.5$, which means the probability that v_4 can successfully deliver a packet to v_6 is 50%. Then the expected number of transmissions to guarantee v_4 delivered one packet to v_6 is $\frac{1}{0.5} = 2$. In other words, the less the γ_{ij} value, the more potential traffic load on the link from v_j to v_i. Therefore, in this research, we investigate how to construct an LBVB for probabilistic WSNs, denoted by LBVBP. In order to better control the trade-off between the size of the constructed VB and the balance of traffic loads among all the dominators, we propose a novel Multi-Objective Genetic Algorithm (MOGA) to solve this problem.

Particularly, the main contributions of this paper are summarized as follows: 1) We identify and highlight the use of lossy links when constructing a VB for Probabilistic WSNs. Moreover, in order to measure the load-balance of the nodes on a VB under the PNM, we define two new metrics *potential traffic load*, and *actual traffic load*, which measure the potential traffic load and actual traffic load of each node in the network, respectively. 2) In order to measure the load-balance factor of a constructed VB, we define two new metrics *VB p-norm* and *Partition p-norm* based on potential traffic load and actual traffic load of each dominator, respectively. 3) The LBVB construction problem under PNM (LBVBP) is an NP-Hard problem, thus we propose an effective Multi-Objective Genetic Algorithm (MOGA) to solve it, called LBVBP-MOGA. 4) We also conduct simulations to validate our proposed algorithms. The simulation results show that the constructed LBVB can extend network lifetime by 65% on average compared with the existing state-of-the-art approaches.

The rest of this paper is organized as follows: In Section II, we introduce the network model and formally define the LBVBP construction problem under PNM. The design of the LBVBP-MOGA algorithm is presented in Section III. The simulation results are presented in Section IV to validate our proposed algorithm. Finally, the paper is concluded in Section V.

II. Network Model and Problem Definition

A. Network Model

Under the *Probabilistic Network Model* (PNM), we model a WSN as an undirected graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \Upsilon(\mathbb{E}))$, where $\mathbb{V}$ is the set of n nodes, denoted by v_i, where $0 \leq i < n$. i is called the node ID of v_i in the paper. $\mathbb{E}$ is the set of lossy links. $\forall$ $v_i, v_j \in \mathbb{V}$, there exists an 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) $\gamma_{ij} > 0$. For each link $(v_i, v_j) \in \mathbb{E}$, γ_{ij} indicates the probability that node v_i can successfully directly deliver a packet to node v_j; and $\Upsilon(\mathbb{E}) = \{\gamma_{ij} \mid (v_i, v_j) \in \mathbb{E}, 0 < \gamma_{ij} \leq 1\}$.

Definition II.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: $N_1(v_i) = \{v_j | v_j \in \mathbb{V}, \gamma_{ij} > 0\}$.*

The physical meaning of 1-Hop Neighborhood is the set of the nodes that can be reached from node v_i via 1 hop neighbors with positive probability. In this paper, we use $|\mathbb{N}_1(v_i)|$ to represent the cardinality of the 1-Hop Neighborhood set of node v_i.

Definition II.2. *h-Hop Neighborhood($\mathbb{N}_h(v_i)$). $\forall v_i \in \mathbb{V}$, the h-Hop Neighborhood of node v_i is defined as: $\mathbb{N}_h(v_i) = \mathbb{N}_{h-1}(v_i) \cup \{v_k \mid \exists v_j \in \mathbb{N}_{h-1}(v_i),\ v_k \in \mathbb{N}_1(v_j),\ v_k \notin \bigcup_{i=1}^{h-1} \mathbb{N}_i(v_i)\}$.*

The physical meaning of the h-Hop Neighborhood is that the set of nodes that can be reached from node v_i by passing maximum h number of *lossy links* with positive probability.

B. Preliminary

Without knowing the communication protocol, 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 probabilistic WSNs. As we mentioned in the Section I, the less the γ_{ij} value, the more potential traffic load on v_j from v_i. Therefore, a more reasonable and formal definition of the potential traffic load is given as follows:

Definition II.3. *Potential Traffic Load (ι_i).* $\forall v_i \in \mathbb{V}$, the potential traffic load of v_i is defined as: $\iota_i = \sum_{v_j \in \mathbb{N}_1(v_i)} \frac{1}{\gamma_{ij}}$.

After knowing the *potential traffic load* of each node, how to measure load-balance of a constructed VB is another challenge. We use *p-norm* to measure load-balance in this paper.

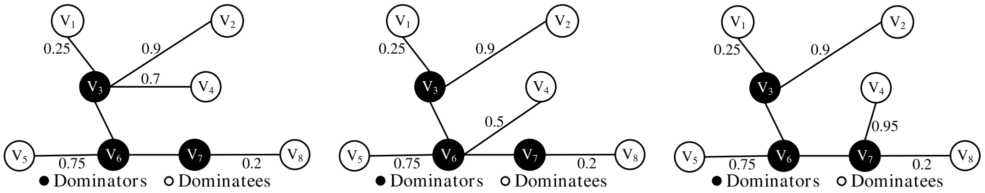
Definition II.4. *p-norm.* The p-norm of an $n \times 1$ vector $\mathbb{X} = (x_1, x_2, \cdots, x_n)$ is: $|\mathbb{X}|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$.

The authors in [13] 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 an 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 $\mathbb{X}$. For simplicity, we use $p = 2$ in this paper.

In this paper, we use *Potential Traffic Load* (*Definition II.3*) as the feature vector $\mathbb{X}$. According to Definition (II.4), we define the *VB p-norm* as follows:

Definition II.5. *VB p-norm ($|\mathbb{B}|_p$).* For WSN $\mathbb{G}(\mathbb{V}, \mathbb{E}, \Upsilon(\mathbb{E}))$, and a VB $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$. The *VB p-norm* of an $m \times 1$ vector $\mathbb{L} = (\iota_1, \iota_2, \cdots, \iota_m)$ is: $|\mathbb{B}|_p = (\sum_{i=1, v_i \in \mathbb{B}}^{m} |\iota_i - \bar{\iota}|^p)^{\frac{1}{p}}$, where m is the cardinality of set $\mathbb{B}$, ι_i represents the potential traffic load of each node in set $\mathbb{B}$, and $\bar{\iota} = (\sum_{j=1, v_j \in \mathbb{B}}^{m} \iota_j)/m$ is the average potential traffic load on set $\mathbb{B}$.

Actually, if one dominatee is adjacent to more than one dominator, one of the adjacent dominators is chosen by the dominatee to perform data transmission. Hence, it is considerably important to load-balancedly allocate dominatees to each dominator to further balance the traffic loads among each dominator. In a traditional/naive way [14], 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 *potential traffic load* can reflect the actual workload precisely. In a WSN with a CDS



(a) (b) (c)

Fig. 2. Illustration of unbalanced and balanced Dominator Partitions.

as the VB, only the dominator and dominatee links contribute to the actual traffic load. Based on this observation, we define the following concepts:

Definition II.6. *Dominator Partition ($\mathscr{P}$).* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \Upsilon(\mathbb{E}))$ and a VB $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$, m disjoint sets are identified on $\mathbb{V}$, *i.e*, $\mathbb{P}(v_1), \mathbb{P}(v_2), \cdots, \mathbb{P}(v_m)$, such that: 1) Each set $\mathbb{P}(v_i)$ $(1 \leq i \leq m)$ contains exactly one dominator v_i. 2) $\bigcup_{i=1}^{m} \mathbb{P}(v_i) = \mathbb{V}$, and $\mathbb{P}(v_i) \bigcap \mathbb{P}(v_j) = \emptyset$ $(1 \leq i \neq j \leq m)$. 3) $\forall v_u \in \mathbb{P}(v_i)$ $(1 \leq i \leq m)$ and $v_u \neq v_i$, such that $(v_u, v_i) \in \mathbb{E}$. A *Dominator Partition* is: $\mathscr{P} = \{\mathbb{P}(v_i) \mid v_i \in \mathbb{B}, 1 \leq i \leq m\}$.

We also use the WSN shown in Fig. 1 to explain the concept of *Dominator Partition*. Three different Dominator Partitions are shown in Fig. 2, in which only dominator and dominatee links are presented in the figure. According to Definition II.6, we have $\mathbb{P}(v_3) = \{v_1, v_2, v_4\}$, $\mathbb{P}(v_6) = \{v_5\}$, and $\mathbb{P}(v_7) = \{v_8\}$ for the partition $\mathscr{P} = \{\mathbb{P}(v_3), \mathbb{P}(v_6), \mathbb{P}(v_7)\}$ shown in Fig. 2(a). Without considering *delivery ration* on each dominator and dominatee link, it is obvious that the Dominator Partition shown in Fig. 2(a) is the most unbalanced of the workloads on each dominator. Moreover, without further information, it is hard to reveal that which partition is more balanced than the other shown in Fig. 2(b), and (c). According to above observations, we define the following concepts and metric to measure the load-balance of a Dominator Partition.

Definition II.7. *Authorized Link Set ($\mathbb{L}_i$).* $\forall v_i \in \mathbb{B}$, the *Authorized Link Set* of dominator v_i is the set of the dominator and dominatee links formed by nodes in $\mathbb{P}(v_i)$, *i.e.*, $\mathbb{L}_i = \{(v_i, v_j) \mid v_j \in \mathbb{P}(v_i), 1 \leq i \leq m\}$.

As we have already known, ι_i is only the indicator of the potential traffic load on each dominator v_i. The actual traffic load can be determined when a Dominator Partition is decided. In other words, the *Authorized Link Set* $\mathbb{L}_i$ along with the corresponding *Delivery Ratio* of each link are the indicators of the actual traffic load on each dominator v_i. According to this observation, we give the following definition:

Definition II.8. *Actual Traffic Load (l_i).* $\forall v_i \in \mathbb{B}$, the actual traffic load of v_i is defined as: $l_i = \sum_{(v_i,v_j)\in\mathbb{L}(v_i)} \frac{1}{\gamma_{ij}}$.

In this paper, we use *Partition p-norm* to measure the load-balance of different Dominator Partitions, in which, the *Actual Traffic Load* l_i of each dominator v_i is used as the feature vector $\mathbb{X}$ shown in Definition II.4. The definition of the *Partition p-norm* is given as follows:

Definition II.9. *Partition p-norm ($|\mathscr{P}|_p$).* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \Upsilon(\mathbb{E}))$, a VB $\mathbb{B} = \{v_1, v_2, \cdots, v_m\}$, and a Dominator Partition $\mathscr{P}$, the *Partition p-norm* is: $|\mathscr{P}|_p = (\sum_{i=1, v_i \in \mathbb{B}}^{m} |l_i - \bar{l}|^p)^{\frac{1}{p}}$. where $\bar{l} = (\sum_{j=1, v_j \in \mathbb{B}}^{m} l_j)/m$ is the average actual traffic load on set $\mathbb{B}$.

It is worth to mention that the smaller the *Partition p-norm* value, the more load-balanced the Dominator Partition. Fig. 2 illustrates unbalanced and balanced Dominator Partitions.

C. Problem Definition

Definition II.10. *Load-Balanced VB Problem in Probabilistic WSNs (LBVBP).* For a WSN represented by graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \Upsilon(\mathbb{E}))$, the *LBVBP* problem is to find a minimum-sized node set $\mathbb{B} \subseteq \mathbb{V}$ and a Dominator Partition $\mathscr{P}$, such that: 1) $\mathbb{G}[\mathbb{B}] = (\mathbb{B}, \mathbb{E}')$, where $\mathbb{E}' = \{e \mid e = (u, v), u \in \mathbb{B}, v \in \mathbb{B}, (u, v) \in \mathbb{E})\}$, 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) *minimize*$\{|\mathbb{B}|_p, |\mathscr{P}|_p\}$.

The LBVBP construction problem is NP-Hard, since it still belongs to the MCDS problem. Based on Definition II.10, the key issue of the LBVBP construction problem is to seek a VB that satisfies multiple constraints, *i.e.*, the minimum size, the minimum *VB p-norm*, and the minimum *Partition p-norm*. In reality, the multiple objectives are potentially in conflict. Conflicting objectives result in a set of compromised solutions, that is known as the Pareto-optimal set. Since none of the solutions in this set can be considered as better than the others with respect to all the objectives, the goal of multi-objective optimization problem is to find as many as Pareto-optimal solutions as possible. Multi-Objective Genetic Algorithms (MOGA) is a powerful tool to fulfill the above requirements due to its inherent parallelism and its ability to exploit the similarities among solutions by recombination [15]. Hence, in the following, a novel MOGA algorithm, named LBVBP-MOGA, is proposed to solve the LBVBP construction problem.

III. LBVBP-MOGA ALGORITHM

1) GA and MOGA Overview: 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. 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.

MOGA is a recently developed algorithmic tool to solve MOPs. MOGAs are very attractive because they have the ability to search partially ordered spaces for several alternative trade-offs. Additionally, an MOGA can track several solutions simultaneously via its population.

A. Design of LBVBP-MOGA

1) Representation of Chromosomes: A chromosome is a possible solution of the LBVBP problem. Hence, when designing the encoding scheme of chromosomes, we need to identify dominators and dominatees in a chromosome and a *Dominator Partition* in a chromosome as well. For convenience, the set of neighboring dominators of each dominatee $v_s \in \mathbb{W}$ is denoted by $\mathbb{D}(v_s) = \{v_r | v_r \in \mathbb{B}, (v_r, v_s) \in \mathbb{E}\}$. In the proposed LBVBP-MOGA, 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 dominator (*i.e.*, $\forall v_i \in \mathbb{B}$), the corresponding gene value is $g_i = 1$. Otherwise, the corresponding gene value is the two-tuples, which represent the allocated dominator of the dominatee v_i, and the neighboring dominators set of the dominatee v_i, respectively. Hence, a generation of chromosomes with gene values is denoted as: $\mathbb{C}^G = \{C_j \mid 1 \le j \le G, C_j = (g_1, g_2, \cdots, g_i, \cdots, g_n)\}$, where G is the number of chromosomes in each generation of population, and for $1 \le i \le n$,

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

Through the above description we know that, as long as choosing a specific dominator from the neighboring dominator set $\mathbb{D}(v_i), \forall v_i \in \mathbb{W}$, we can decide a specific *Dominator Partition*. Additionally, all the nodes with $g_i = 1$ form a VB $\mathbb{B} = \{v_i \mid g_i = 1, 1 \le i \le n\}$. We still use the probabilistic WSN shown in Fig.1(b) to illustrate the encoding scheme. There are 8 nodes and the VB is $\mathbb{B} = \{v_3, v_6, v_7\}$. Moreover, according to the topology shown in Fig. 1(b), we can get $\mathbb{D}(v_i), \forall v_i \in \mathbb{W}$ easily. Thus, the *Dominator Partition* shown in Fig. 2(b) can be encoded using 8 genes in a chromosome, *i.e.*, $C_1 = (v_3 \mid \{v_3\}, v_3 \mid \{v_3\}, 1, v_6 \mid \{v_3, v_6, v_7\}, v_6 \mid \{v_6\}, 1, 1, v_7 \mid \{v_7\})$ (shown in Fig. 3). In conclusion, C_j records one possible VB and one possible Dominator Partition associated with the VB, while $\mathbb{C}^G$ represents the G different solutions to the LBVBP problem.

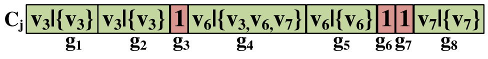


Fig. 3. A chromosome with meta-genes and genes.

2) 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 G 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 VB. Therefore we start to create the first chromosome C_1 by running an existing MCDS method, *e.g.*, the latest MCDS construction algorithm [14], [16], and then generate the population with G chromosomes by modifying C_1. We call the procedure, generating the whole population by modifying one specific chromosome, Inheritance Population Initialization (IPI) [12].

3) Fitness Function: Given a solution, its quality should be accurately evaluated by the fitness score, which is determined by the fitness function. In MOGAs, multiple conflict objectives need to be achieved. Hence, in our algorithm, three different fitness functions are defined as follows:

$$\begin{cases} f_1(C_j) = \min\{|\mathbb{B}|\}; \\ f_2(C_j) = \min\{|\mathbb{B}|_p\}; \\ f_3(C_j) = \min\{|\mathscr{P}|_p\}. \end{cases}$$

As we have mentioned, different from the relations of the solutions in SOPs, the relations of the solutions of MOPs have three possibilities. We use the following operator to summary the relations.

Definition III.1. *Fitter Operator*

$$Fitter(C_i, C_j) = \begin{cases} 1 & \boldsymbol{F}(C_i) \preceq \boldsymbol{F}(C_j); \\ -1 & \boldsymbol{F}(C_j) \preceq \boldsymbol{F}(C_i); \\ 0 & \text{non-dominated.} \end{cases} \quad (1)$$

where the objective vector is $\boldsymbol{F}(C_k) = (f_1(C_k), f_2(C_k), f_3(C_k)), 1 \le k \le G$.

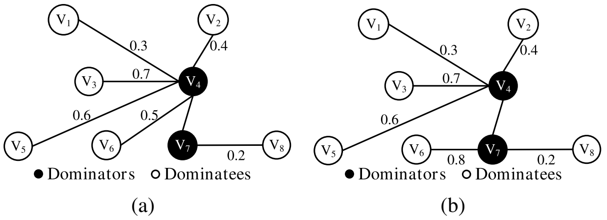


Fig. 4. Illustration of two different Dominator Partitions.

TABLE I
MULTIPLE OBJECTIVE FITNESS VALUES OF FIVE CHROMOSOMES

Chromosomes	f_1	f_2	f_3
C_1 (Fig. 4(a))	2	$\sqrt{68.8}$	$\sqrt{17.58}$
C_2 (Fig. 4(b))	2	$\sqrt{68.8}$	$\sqrt{3.59}$
C_3 (Fig. 2(a))	3	$\sqrt{34.68}$	$\sqrt{14.33}$
C_4 (Fig. 2(b))	3	$\sqrt{34.68}$	$\sqrt{1.99}$
C_5 (Fig. 2(c))	3	$\sqrt{34.68}$	$\sqrt{12.48}$

We will use the feasible solutions (encoded as chromosomes) shown in Fig. 4 and Fig. 2 to illustrate the *Fitter Operator*. The three fitness values of the five different chromosomes are listed in Table I. To be more specific, we draw the five chromosomes in the two-dimensional objective space (f_2 *vs.* f_3) in Fig. 5. From Fig. 5, we know C_2 Pareto dominates C_1 and C_5 Pareto dominates C_3. Moreover, C_4 Pareto dominates all other chromosomes, which means C_4 is the best solution among these 5 chromosomes.

4) Selection Scheme and Replacement Policy: 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, in MOPS, the selection operator needs to be more carefully formulated to ensure that better chromosomes (the chromosomes closer to the Pareto Optimal Set) of the population have a greater probability

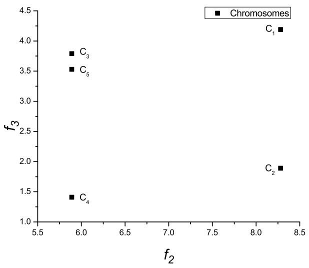


Fig. 5. Illustration of five chromosomes in the two-dimensional objective space.

of being selected for mating. We adopt *Dominating Tree (DT)* [17] to select parent chromosomes. A DT is a binary tree, in which each node has three fields: *id, left-link, and right-link*. The left-link field links to its left sub-tree whose root node is *dominated* by the node, and the right-link filed links to its right sub-tree whose root node is *non-dominated* by the node. A *Sibling Chain* of a DT is defined as a chain constituted by its root and the root's right-link nodes. A DT has some useful features [17]:

- The sibling chain of a DT consists of and only consists of all Pareto Optimal nodes in the DT.
- The root of a DT Pareto dominates all nodes in its left sub-tree.
- The leftmost node in the DT can be regarded as the "worst" node of the DT.

Fig. 6 demonstrates a DT consisting of the 5 encoded chromosomes shown in Fig. 4 and Fig. 2. According to the above features, we have:

- C_2 and C_4 (the *sibling chain*) are Pareto Optimal nodes in the DT.
- C_4 Pareto dominates C_3 and C_5 in the DT.
- C_3 can be considered as the "worst" node in the DT.

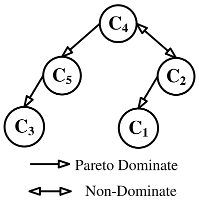


Fig. 6. Illustration of a Dominating Tree.

Based on the above description, in each generation, we random select parent chromosomes in the set of Pareto Optimal nodes (i.e., from *sibling chain*). New-generated offsprings will be inserted into the DT one by one and the "worst" (left-most) individual will be deleted from the DT each time. This process will be repeated until the stopping criterion (such as, a desired number of total generation is reached) is satisfied.

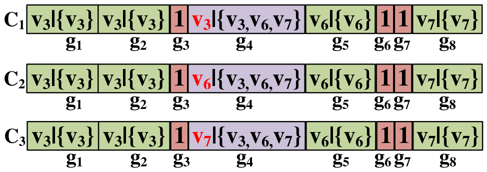
5) Genetic Operations: The performance of a GA relies heavily on two basic genetic operators, *crossover* and *mutation*. Crossover exchanges parts of the the parent chromosomes in order to find better ones. Mutation flips the values of genes, which helps a GA keep away from local optimum. In the LBVBP 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 [12] needs to be preformed to guarantee the validity of all the new generated offspring solutions.

The purpose of crossover operations is to produce more valid VBs represented by the offspring chromosomes. At this stage, we do not need to care about Dominator Partitions. Therefore, when performing crossover operations, we can logically assume all gene values of dominatees are 0. In the LBVBP-MOGA 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 select two chromosomes from the set of Poreto Optimal nodes as parents to perform one of the three crossover operators randomly. As mentioned early, after crossover operation, the new generated offsprings may not be a valid solution (the constructed VB is not a CDS). Thus we need to perform the correction mechanism. The mechanism starts with scanning each gene g_i on the offspring chromosome from the position of the crossover point, till the end of the chromosome. If g_i value is different from the corresponding value of its parent, then the mechanism corrects the value.

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 same correction mechanism needs to be preformed if the mutated chromosomes are not valid.

In order to increase the diversity of possible Dominator Partitions, we propose an additional step called *dominatee mutation* in LBVBP-MOGA to generate more feasible Dominator Partitions.

As known, as long as choosing a specific node from the neighboring dominator set $\mathbb{D}(v_i), \forall v_i \in \mathbb{W}$, we can easily explore a Dominator Partition. According to the observation, we design the following *dominatee mutation*. The original population without doing crossover and gene mutation operations will undergo the Dominatee mutation operation. If the number of neighboring dominators of a dominatee v_i is greater than 1, *i.e.*, $|\mathbb{D}(v_i)| \geq 2$, then randomly pick a node from the set $\mathbb{D}(v_i)$. We use the VB shown in Fig. 1(b) to illustrate the dominatee mutation. According to the topology, we get $|\mathbb{D}(v_4)| = |\{v_3, v_6, v_7\}| = 3 > 1$, which satisfies the condition to perform the dominatee mutation. Therefore, we randomly pick one dominator from the set $\mathbb{D}(v_4)$. If v_3 is selected from $\mathbb{D}(v_4)$, it means dominatee v_4 is allocated to dominator v_3. The corresponding Dominator Partition is shown in Fig. 2(a). Similarly, if dominatee v_4 is allocated to dominator v_6, or v_4 is allocated to dominator v_7, the Dominator Partitions are shown in Fig. 2(b), and (c), respectively. In summary, the process of dominatee mutation for v_4 is shown in Fig. 7.



	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
C_1	$v_3\|\{v_3\}$	$v_3\|\{v_3\}$	1	$v_3\|\{v_3,v_6,v_7\}$	$v_6\|\{v_6\}$	1	1	$v_7\|\{v_7\}$
C_2	$v_3\|\{v_3\}$	$v_3\|\{v_3\}$	1	$v_6\|\{v_3,v_6,v_7\}$	$v_6\|\{v_6\}$	1	1	$v_7\|\{v_7\}$
C_3	$v_3\|\{v_3\}$	$v_3\|\{v_3\}$	1	$v_7\|\{v_3,v_6,v_7\}$	$v_6\|\{v_6\}$	1	1	$v_7\|\{v_7\}$

Fig. 7. Illustration of Dominatee Mutation.

IV. Performance Evaluation

Since there are no existing works studying the LBVB construction problem for probabilistic WSNs currently, in the simulations, the results of LBVBP-MOGA (denoted by MOGA) are compared with the recently published Minimum-sized CDS construction algorithm [14] denoted by MCDS, and the LBCDS-GA algorithm proposed in [12] denoted by GA. We compare the three algorithms in terms of network lifetime, which is defined as the time duration until the first dominator runs out of energy.

A. 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, a random value between $[0.5, 0.98]$ is assigned to the Delivery Ratio (γ_{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 γ_{ij} associated to a pair of nodes beyond the transmission range. 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.

B. Simulation Results

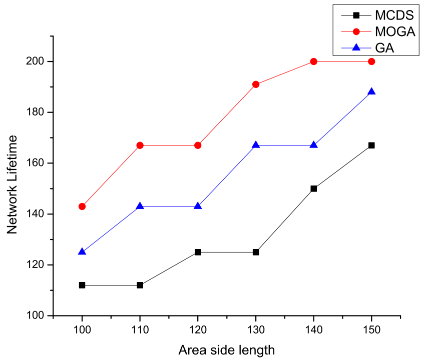


Fig. 8. Network Lifetime: the node transmission range is *20m*, the number of nodes is *100*, and the side length of the deployed area changes from *100m* to *150m*

Fig. 8 shows the network lifetime of three methods (MOGA, MCDS, GA)) under two different scenarios. From Fig. 8, we know that the network lifetime increases for all the three 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, the thinner the network is, the less number of neighbors of each dominator. In other words, the aggregated data are less on each dominator when the network becomes thinner. Hence, network lifetime is increasing for all the three algorithms. Additionally, we can see both MOGA and GA outperform MCDS. Furthermore, MOGA prolongs network lifetime by 42% on average compared with MCDS, and by 20% on average compared with GA. The results demonstrate that load-balancedly allocating dominatees to dominators can improve network lifetime significantly. On the other hand, MOGA outperforms GA, since MOGA takes multiple objectives into consideration simultaneously, which making the MOGA easier to converge to a global optimum. Additionally, the local optimal solution found by GA might not be the same as the global optimal solution. Hence, the results shown in Fig. 8 indicate our proposed MOGA can find a solution which is closer to the optimal solution than GA.

V. Conclusion

In this paper, we address the problem of constructing a Load-Balanced VB in a probabilistic WSN (LBVBP), which is a Minimum-sized CDS with the minimum $|\mathbb{B}|_p$ and $|\mathscr{P}|_p$ values in order to assure that the workload among each dominator is balanced. We propose an effective Multi-Objective Genetic Algorithm named LBVBP-MOGA to solve the problem. The simulation results demonstrate that using an LBVB can extend network lifetime significantly.

Acknowledgment

This research is partly supported by the National Science Foundation (NSF) under grant No. CCF-0545667, and the Kennesaw State University College of Science and Mathematics Faculty Summer Research award program.

References

[1] S. Ji and Z. Cai, *Distributed Data Collection in Large-Scale Asynchronous Wireless Sensor Networks under the Generalized Physical Interference Model*, to appear in the IEEE/ACM Transactions on Networking (ToN), 2012.

[2] 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), pp. 56-73, 1987.

[3] S. Cheng, J. Li, and Z. Cai, *$O(\varepsilon)$-Approximation to Physical World by Sensor Networks*, IEEE INFOCOM 2013.

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

[5] 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.

[6] 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.

[7] 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.

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

[9] S. Ji, R. Beyah, and Z. Cai, *Snapshot and Continuous Data Collection in Probabilistic Wireless Sensor Networks*, to appear in the IEEE Transactions on Mobile Computing (TMC), 2013.

[10] 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.

[11] J. He, S. Ji, Y. pan, and Z. Cai, *Approximation Algorithms for Load-Balanced Virtual Backbone Construction in Wireless Sensor Networks*, to appear in the Theoretical Computer Science (TCS), 2013.

[12] 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), 11(3): 166-178, 2012.

[13] Kalantari, M., Haghpanahi, M., and Shayman, M., *A p-norm Flow Optimization Problem in Dense Wireless Sensor Networks*. INFOCOM. pp. 341-345, 2008.

[14] P.J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia, *Minimumlatency aggregation scheduling in multihop wireless networks*, MobiHoc, 2009.

[15] C. Coello, *Evolutionary Multiobjective Optimization: a historical view of the field*, IEEE Computational Intelligence Magazine, 1(1):28 - 36, 2006.

[16] S. Ji, J. He, A. S. Uluagac, R. Beyah, and Y. Li, *Cell-based Snapshot and Continuous Data Collection in Wireless Sensor Networks*, to appear in the ACM Transactions on Sensor Networks (TOSN), 2012.

[17] C. Shi, Z. Yan, Z. Shi, and L. Zhang, *A Fast Multi-Objective Evolutionary Algorithm Based On A Tree Structure*, Applied Sof Computing, 10:468-480, 2010.

[18] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, Bristol, UK, IOP Publishing, 1997.