Genetic-Algorithm-Based Construction of Load-Balanced CDSs in Wireless Sensor Networks

Jing (Selena) He, Shouling Ji, Mingyuan Yan, Yi Pan, Yingshu Li
Department of Computer Science
Georgia State University
Atlanta, Georgia, USA
Email: {jhe9, sji, myan2, pan, yli}@cs.gsu.edu

Abstract—A Connected Dominating Set (CDS) is used as a virtual backbone for efficient routing and broadcasting in Wireless Sensor Networks (WSNs). Most existing works focus on constructing a Minimum CDS (MCDS), a $k$-connect $m$-dominating CDS, a minimum routing cost CDS or a bounded-diameter CDS. However, no work considers the load-balance factor of CDSs in WSNs. In this paper, we propose a novel problem — the Load-Balanced CDS (LBBCDS) problem, in which constructing an LBBCDS and load-balancedly allocating dominatees to dominators are investigated simultaneously. A Genetic Algorithm (GA) based strategy called LBBCDS-GA is proposed to construct an LBBCDS 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 LBBCDS and load-balancedly allocating dominatees to dominators can further prolong network lifetime through balancing the workloads of all the dominators. Through extensive simulations, we demonstrate that our proposed methods extend network lifetime by 65% on average compared with the best and latest MCDS construction algorithm.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are deployed for monitoring and controlling systems where human intervention is not desirable or feasible. Therefore, WSNs are widely used in many military and civilian applications such as battlefield surveillance, health care applications, environment and habitat monitoring, and traffic control [1]. Due to the lack of physical backbone infrastructure in WSNs, most routing protocols in WSNs, such as flooding, usually cause a serious broadcasting storm [2]. A Connected Dominating Set (CDS) has become 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 node in the set. If the induced graph 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 $\mathbb{D}$, otherwise, dominatees denoted by set $\mathcal{W}$. In a WSN with a CDS as its VB, dominatees may forward their data only to their neighboring dominators. With he help of a CDS, the average message burden of a WSN could be reduced so that routing becomes much easier and can adapt quickly to network topology changes [3]. Furthermore, using dominators as forwarding nodes can efficiently reduce energy consumption, which is also a critical concern in WSNs. In addition to routing protocols, a CDS has many other applications in WSNs, such as topology control [4], coverage [5], data collection [6], and data aggregation [7]. Clearly, the benefits of a CDS can be magnified by making its size (the number of the nodes in the CDS) smaller. In general, the smaller the CDS is, the less communication and storage overhead are incurred. Hence, it is desirable to build a Minimum-sized CDS (MCDS).

Ever since the idea of employing a CDS for WSNs was introduced [8], a huge amount of effort has been made to construct different CDSs for different applications, especially MCDSs. In the seminal work [9], Guha and Kuller first modeled the problem of constructing the smallest CDS as the MCDS problem in a general graph, which is a well-know NP-Hard problem [10]. Subsequently, many polynomial-time approximation algorithms for constructing an MCDS construction have been proposed in recent literatures, such as abstraction-based algorithms [11], [12], and addition-based algorithms [7]. After that, to make a CDS more resilient in mobile WSNs, the fault-tolerance of a VB is considered. In [13], $k$-connected and $m$-dominated sets are introduced as a generalized abstraction of a fault-tolerance VB. In [14], 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 [15].

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 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 (LBBCDS). An illustration of an LBBCDS is depicted in Fig. 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 $v_i \in \mathbb{D}$ is denoted by $\mathbb{U}(v_i)$. The set of dominatees allocated to a dominator $v_i$ is denoted by $\mathcal{A}(v_i) = \{v_j | v_j \in \mathcal{W}, v_j \text{ forwards its data only to } v_i\}$. According to the traditional MCDS construction algorithms, a CDS $\{v_1,v_2\}$ with size 2 is obtained for the network shown in Fig. 1(a). However, There are two severe drawbacks of the CDS shown in Fig. 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 $\bigcup (v_5) = \{v_6, v_8\}$. 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. 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. 1(b), where the constructed CDS is $\{v_3, v_6, v_7\}$. According to the topology shown in Fig. 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. 1(a), the numbers of neighboring dominatees of all the dominators in Fig. 1(b) are very similar. On the other hand, we have two different dominatee allocation schemes shown in Fig. 1(b) and Fig. 1(c) respectively. One is: $A_{(v_3)} = \{v_1, v_2, v_4\}$, $A_{(v_6)} = \{v_5\}$, and $A_{(v_7)} = \{v_8\}$. The other one is: $A_{(v_3)} = \{v_1, v_2\}$, $A_{(v_6)} = \{v_4, v_5\}$, and $A_{(v_7)} = \{v_8\}$. Apparently, the workload on each dominator is almost evenly distributed in the CDS constructed in Fig. 1(c). Intuitively, the construction algorithm and dominatee allocation scheme shown in Fig. 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. To solve the workload imbalance problem of an MCDS, in this paper, we investigate how to construct an LBCDS and how to load-balancedly allocate dominatees to dominators simultaneously.

Particularly, our contributions mainly include four aspects as follows: 1) To the best of our knowledge, this is the first work that studies the LBCDS problem. We point out the disadvantages and advantages of the existing MCDS construction algorithms. Based on the observation of workload imbalance in MCDS, we propose to take the load-balance factor into consideration when building a CDS and allocating dominatees to each dominator. 2) In order to measure the load-balance factor of a CDS and a dominatee allocation scheme, we define two new metrics CDS $p$-norm and Allocation Scheme $p$-norm based on the degrees of dominators and the number of allocated dominatees to each dominator, respectively. 3) We claim that the LBCDS problem is an NP-Hard problem, thus we propose an effective GA to solve the LBCDS problem, called LBCDS-GA. Based on the characteristics of GAs, the searching heuristic, lots of possible solutions are searched and the best one is selected as the final result. Additionally, GAs have been proven to be effective at escaping local optima and discovering the global optimum in even a very complex searching space [16]. 4) We also conduct extensive simulations to validate our proposed algorithm. The simulation results show that the constructed LBCDS can extend network lifetime significantly compared with the existing algorithms. Particularly, when the node number changes from 100 to 1000, our proposed method extends network lifetime by 65% on average compared with the work in [7], which is the latest MCDS construction algorithm.

The rest of this paper is organized as follows: in Section II, we introduce the network model and formally define the LBCDS problem. The design of a GA for the LBCDS problem 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. PROBLEM DEFINITION

A. 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 $G(V, E)$, where $V$ is the set of $n$ sensor nodes, denoted by $v_1, v_2, \ldots, v_n$; $E$ represents the link set, $\forall u, v \in V$, there exists an link $(u, v)$ in $E$ if and only if $u$ and $v$ are in each other’s transmission range. In this paper, we assume edges are undirected (bidirectional), which means two linked nodes are able to transmit and receive data from each other.

B. Terminologies

Definition II.1. $p$-norm ($\|\Theta\|_p$). The $p$-norm of an $n \times 1$ vector $\Theta = (\theta_1, \theta_2, \ldots, \theta_n)$ is: $\|\Theta\|_p = (\sum_{i=1}^{n} |\theta_i|^p)^{\frac{1}{p}}$.

The authors in [17] 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 $\theta_i$. The smaller the $p$-norm value is, the more load-balanced the interested feature vector $\Theta$ is.

Definition II.2. CDS $p$-norm ($\|B\|_p$). For a WSN represented by graph $G(V, E)$, and a CDS $B = \{v_1, v_2, \ldots, v_m\}$. The CDS $p$-norm of an $m \times 1$ vector $D = (d_1, d_2, \ldots, d_m)$ is: $\|B\|_p = (\sum_{i=1}^{m} |d_i - d|^p)^{\frac{1}{p}}$, where $m$ is the number of dominators in the set $B$, $d_i$ represents the node degree of each dominator in the set $B$, and $d$ is the mean degree of $G$.

We use the WSN shown in Fig. 1 to illustrate how to calculate the CDS $p$-norm. For simplicity, we use $p = 2$ in this paper. Without specific explanation, $p$ and 2 are interchangeable in this paper. Two different CDSs for the same network are identified in Fig. 1. The degree of node $v_i$ is denoted by $d_i$ in Fig. 1. From the topology shown in Fig. 1, we can get $\bar{d} = 3$. Therefore, the CDS...
p-norm of the CDS shown in Fig. 1(a) is \( \sqrt{9} \). Similarly, in Fig. 1(b), the CDS p-norm value is \( \sqrt{2} \). Clearly, \( \sqrt{2} < \sqrt{9} \), which implies that the CDS in Fig. 1(b) is more load-balanced than the CDS in Fig. 1(a). The result also matches the observation we mentioned in Section I.

When constructing an LBCDS, it is considerably important to allocate dominatees to each dominator. In a traditional/naive way, such as the work in [7], 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 as the CB, only the dominator and dominatee links contribute to workload. Based on this observation, we define the following concepts:

**Definition II.3. Dominatee Allocation Scheme (\( \mathcal{A} \)).** For a WSN represented by graph \( G(V, E) \) and a CDS \( B = \{v_1, v_2, \ldots, v_m\} \), we need to find \( m \) disjoint sets on \( V \), i.e., \( A(v_1), A(v_2), \ldots, A(v_m) \), such that: 1) Each set \( A(v_i) \) (\( 1 \leq i \leq m \)) contains exactly one dominator \( v_i \), 2) \( \bigcup_{i=1}^{m} A(v_i) = V \), and \( A(v_i) \cap A(v_j) = \emptyset \) (\( 1 \leq i \neq j \leq m \)). 3) \( \forall v_u \in A(v_i) \) \((1 \leq i \leq m)\) and \( v_u \neq v_i \), such that \((v_u, v_i) \in E \).

A Dominatee Allocation Scheme is: \( \mathcal{A} = \{A(v_i) \mid v_i \in B, 1 \leq i \leq m \} \).

**Definition II.4. Valid Degree (\( d^p \)).** The Valid Degree of dominator \( v_i \) is the number of its allocated dominatees, i.e., \( \forall v_i \in B, d_i^p = |A(v_i)| \), where \( |A(v_i)| \) represents the number of dominatees in the set \( A(v_i) \).

**Definition II.5. Allocation Scheme p-norm (\( |A|^p \)).** For a WSN represented by graph \( G(V, E) \), a CDS \( B = \{v_1, v_2, \ldots, v_m\} \), and a dominatee allocation scheme \( \mathcal{A} \), the Allocation Scheme p-norm is: \( |A|^p = (\sum_{i=1}^{m} d_i^p - |E|^p)^{\frac{1}{p}} \), where \( d_i^p \) represents the valid degree of each dominator in the set \( B \), and \( |E|^p = n^p - m \) is the expected allocated dominatees on each dominator.

Fig. 1(b) and Fig. 1(c) illustrate an imbalanced and a balanced dominatee allocation scheme respectively. The valid Degree of dominator \( v_i \) is denoted by \( d_i^p \) in Fig. 1. From the topology shown in Fig. 1(b) and (c), we can get \( E = \frac{n}{2} \). Therefore, the Allocation Scheme p-norm of the dominatee allocation scheme shown in Fig. 1(b) is \( \sqrt{\frac{7}{6}} \). Similarly, in Fig. 1(c), the Allocation Scheme p-norm is \( \sqrt{\frac{3}{2}} \), which implies the dominatee allocation scheme shown in Fig. 1(c) is more load-balanced than the scheme shown in Fig. 1(b). The result further confirms the observation we mentioned in Section I.

**C. Problem Definition**

**Definition II.6. Load-balanced CDS (LBCDS) Problem.** For a WSN represented by graph \( G(V, E) \), the LBCDS problem is to find a minimum-sized node set \( B \subseteq V \) and a dominatee allocation scheme \( \mathcal{A} \), such that: 1) \( G[B] = (B, E') \), where \( E' = \{e \mid e = (u, v), u \in B, v \in B, (u, v) \in E \} \), is connected. 2) \( \forall u \in V \) and \( u \notin B, \exists v \in B, \text{ such that } (u, v) \in E \). 3) min(|B|,|A|^p).

We claim that the LBCDS problem is NP-Hard, since it still belongs to the MCDS problem. Based on Definition II.6, 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 [18], [19]. 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.

**III. LBCDS-GA ALGORITHM**

**A. GA Overview**

GAs are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics [20]. 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.

**B. 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 \( v_s \in W \) is denoted by \( \mathcal{H}(v_s) = \{v_r ; v_r \in B, (v_r, v_s) \in E \} \).

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: \( \mathcal{C}^0 = \{C_j^0 \mid 1 \leq j \leq k, C_j^0 = (g_1, g_2, \ldots, g_{n}), \} \), where \( k \) is the number of the chromosomes in each generation of population, and for \( 1 \leq i \leq n \),

\[
g_i = \begin{cases} 1, & v_i \in \mathbb{B} \backslash \mathbb{B}' \in \mathbb{W} \end{cases}
\]

Additionally, beyond the aforementioned gene value, there is a meta-gene value \( G_j \) to store \( \mathcal{H}(v_s), \forall v_s \in \mathbb{W} \). Thus, a generation of chromosomes with meta-gene values is denoted as: \( \mathcal{C}^G = \{C_j^G \mid 1 \leq j \leq k, C_j^G = (G_1, G_2, \ldots, G_i, \ldots, G_n) \} \), and for \( 1 \leq i \leq n \),
Through the above description we know, as long as choosing a specific node from each node set $H(v_i), \forall v_i \in \mathbb{W}$, we can easily generate $C^n_g$ from $C^n_G$. 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.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. 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. 2. Based on the dominatee allocation scheme shown in Fig. 1(c), i.e., dominatee $v_4$ is allocated to dominator $v_6$, the chromosome with 8 genes is $C^g = (\{v_3\}, \{v_3\}, 1, \{v_3, v_6, v_7\}, \{v_6\}, 1, 1, \{v_7\})$. In conclusion, $C^G$ stores all neighboring dominators of each dominatee, while the corresponding $C^g$ records one CDS and one specific dominatee allocation scheme.

$G_i = \left\{ \begin{array}{ll} 1, & v_i \in \mathbb{B}, \\ \mathbb{H}(v_i), & v_i \in \mathbb{W}. \end{array} \right.$

$\mathbb{H}(v_i), v_i \in \mathbb{W}$.


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 $|\mathcal{A}|_p$ values. Therefore, the fitness function of a chromosome $C^n_g$ is defined as:

$$f(C^n_g) = \frac{n - |\mathbb{B}|}{w_1|\mathbb{B}|_p + w_2|\mathcal{A}|_p}$$

$$w_1 + w_2 = 1, \quad 0 < w_1, w_2 < 1$$

The purpose of doing a linear combination of $|\mathbb{B}|_p$ and $|\mathcal{A}|_p$ values in Equation 1 is that a user can change the weight of $|\mathbb{B}|_p$ and $|\mathcal{A}|_p$ values dynamically and easily. The denominator in Equation 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.

**E. Selection Scheme**

During the evolutionary process, selection 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 $R_i = \log(1 + f(C^n_g))$. 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 $P_i = \frac{R_i}{\sum_{i=1}^{k} R_i}$, thus, $S = \sum_{i=1}^{k} P_i$. 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.

**F. 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. 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.

1) 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, and two-point crossover respectively. With a crossover probability $p_c$, each time we use the RS scheme to select two chromosomes $C^n_g$ and $C^n_b$ as parents to perform one of the two crossover operators randomly. We use Fig. 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.3(a)), the genes from the randomly generated crossover point $P = 6$ to the end of the two chromosomes exchange with each other to get (00011010) and (00101111). After crossing, the first offspring (00011010) 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 $C^p$, 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 $C^o$, 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 correction mechanism. Otherwise, keep repeating the process till the end of $C^p$ is reached. The idea behind the correction mechanism is that the DS represented by $C^p$ is a CDS. If $C^p$ is not valid, then add the dominator represented by $C^p$ into the DS represented by $C^o$ 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.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 offspring are (00100111) and (00010011) respectively. Since both of the offspring chromosomes are valid, we do not need to do any correction.

2) 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.

G. 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 $H(v)$, $\forall v \in W$, we can easily generate $C^o$ from $C^p$. 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., $|H(v)| \geq 2$, then randomly pick a node from the set $H(v)$ with a probability $p_i$. For example, the CDS shown in Fig. 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. 2. Since $G_4 = \mathcal{H}(v_4) = \{v_3, v_6, v_7\}$, which means $|\mathcal{H}(v_4)| \geq 2$. We then randomly pick one dominator from the set $\mathcal{H}(v_4)$ with a probability $p_i$. If $v_3$ is selected from $\mathcal{H}(v_4)$, it means dominatee $v_4$ is allocated to dominator $v_3$. The dominatee allocation scheme is shown in Fig. 1(b), encoding as the chromosome with genes $(\{v_3\}, \{v_3\}, 1, \{v_3, v_6, v_7\})$. Similarly, if dominatee $v_4$ is allocated to dominator $v_6$, the dominatee allocation scheme is shown in Fig. 1(c), encoding as the chromosome with genes $(\{v_3\}, \{v_3\}, 1, \{v_6\}, 1, 1, \{v_7\})$.

H. 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 score does not change for continuous $l$ generations.

IV. PERFORMANCE EVALUATION

In the simulations, the results of LBCDS-GA are compared with the MCDS construction algorithm in [7] denoted by MIS, which is the latest and best MIS-based CDS construction algorithm. We compare the two algorithms in terms of Allocation Scheme p-norm, and 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 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 I.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>GA PARAMETERS AND RULES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size (K)</td>
<td>50</td>
</tr>
<tr>
<td>Number of total generations (K)</td>
<td>100</td>
</tr>
<tr>
<td>Selection scheme</td>
<td>Rank Selection</td>
</tr>
<tr>
<td>Replacement policy</td>
<td>Elitism</td>
</tr>
<tr>
<td>Crossover probability (pc)</td>
<td>1</td>
</tr>
<tr>
<td>Gene mutation probability (pm)</td>
<td>0.2</td>
</tr>
<tr>
<td>Meta-gene mutation probability (pj)</td>
<td>1</td>
</tr>
</tbody>
</table>

B. Simulation Results and Analysis

In Fig. 4, the X-axis represents the number of the sensor nodes n, while the Y-axis represents the evaluated factors, i.e., Allocation Scheme p-norm |A|p, and network lifetime T respectively.

Fig. 4(a) shows the Allocation Scheme p-norm |A|p values of the constructed CDSs by using LBCDS-GA and MIS. As mentioned before, the smaller the |A|p value, the more load-balanced the dominatee allocation scheme A. With the increase of n, |A|p increases quickly for the MIS scheme. This is because, in MIS, dominatesses are always allocated to the dominant with the smallest ID. The results also imply that the A of MIS becomes more and more imbalanced when n is getting larger. Nevertheless, for LBCDS-GA, |A|p keeps almost the same, which means no matter how large the size of the set B is, LBCDS-GA always can find a load-balanced A. Additionally, with the increase of n, the difference of |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(b) shows network lifetime of the two |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 [7].

ACKNOWLEDGMENT

This research is partly supported by the National Science Foundation (NSF) under grant No. CCF-0545667, CNS-0831634, and by the 111 project of China under the grant No. 111-2-14.

REFERENCES