Optimal Node Scheduling for the Lifetime Maximization of Two-tier Wireless Sensor Networks

Ying Lin, *Student Member, IEEE*, Xiao-min Hu, *Student Member, IEEE*,

Jun Zhang, *Senior Member, IEEE*, Ou Liu and Hai-lin Liu

Abstract—**Research into maximizing the network lifetime is one of the most significant and challenging areas in wireless sensor networks (WSNs). By arranging sensors and sinks to realize target coverage and network connectivity respectively, an efficient schedule of sensors and sinks can prolong the network lifetime. However, the arrangements of sensors and sinks correlate with each other because each sensor needs to send its data to a sink, making the problem of finding the optimal schedule difficult. Instead of using a single process to optimize the entire schedule of sensors and sinks, this paper proposes a scheduling method which uses two separate processes to schedule operations of sensors and sinks respectively. The first process organizes sensors in the network into disjoint sets, with each set being able to fully cover the targets. Based on the arrangement of sensors, a novel genetic algorithm (GA) is adopted in the second process to allocate sinks to each set of sensors. When the number of full cover sets that ensure both connectivity of sensors to sinks and connectivity of the network composed of sinks is maximized, a schedule that maximizes the network lifetime can be obtained. The proposed method has been applied to a number of WSN cases. Results demonstrate that the method is effective and efficient in prolonging the lifetime of WSNs.**

I. INTRODUCTION

WIRLESS sensor networks (WSNs) are composed of a set of battery-powered nodes that are deployed in an area of interest. Each node senses the surrounding environment and delivers the sensed data via wireless transmission to the remote base station (BS) for further analysis or applications. In this way, WSNs are able to provide reliable, accurate, and real-time observations over a vast area, encouraging usage in many civil and military applications [1][2]. However, due to the large quantity of nodes and the unpredictability of the environment, replacing or recharging the battery of each node is difficult. The energy resource is thus the fundamental constraint in the applications of WSNs. How to conserve energy and prolong the network lifetime is a critical issue in the research of WSNs.

This work was supported in part by the NSFC Joint Fund with Guangdong under Key Project U0835002 and the National High Technology Research and Development Program ("863" Program) of China, No. 2009AA01Z208.

Ying Lin, Xiao-min Hu, and Jun Zhang are with the Department of Computer Science, Sun Yat-sen University, Guangzhou, China. Jun Zhang is the corresponding author (email: junzhang@ieee.org)

Ou Liu is with The Hong Kong Polytechnic University, Hung Hom, Hong Kong.

Hai-lin Liu is with Guangdong University of Technology, Guangzhou, China.

When nodes in a WSN are densely deployed, the sensing ranges of neighboring nodes usually overlap. Only a subset of the nodes can already fulfill the sensing task and the rest nodes can be scheduled into a sleep state for conserving energy. Therefore, the maximization of network lifetime can be considered as a problem of finding the maximum number of disjoint sets of nodes, with each set being able to realize target coverage and network connectivity independently.

In the literature, the above idea has been applied to WSNs of flat architectures and encouraging results have been reported [3]-[6]. However, WSNs in real-world applications generally have *n*-tier (n>1) architectures. One of the most widely used *n*-tiered WSNs is the two-tier WSN [7]-[12]. A two-tier WSN comprises a number of clusters and one or more BS's. Each cluster is composed of several member nodes and a head node. The member nodes of different clusters compose the lower tier of the WSN while the cluster heads (CHs) and the BS's compose the upper tier. In a cluster, the member nodes are responsible for the sensing task over the corresponding area, whereas the CH collects the data from the sensors and routes them to the BS. Such cluster-based architecture offers some inherent advantages against the flat architecture in terms of energy conservation [13]-[15]. First, since only CHs are involved in routing data to the remote BS's and the member nodes only transmit the sensed data to a CH nearby, the energy consumed in data transmission is substantially reduced [13][14]. Second, the fact that only the CH transmits data out of the cluster also helps save energy by avoiding collisions between cluster members [15]. Due to the popularity and advantages of using two-tier WSNs, we consider designing energy-efficient algorithms or protocols for prolonging the lifetime of two-tier WSNs deserves more in-depth investigations.

Generally speaking, there are two types of clusters in two-tier WSNs: homogeneous and heterogeneous [16][17]. In this paper, we focus on the WSNs comprised of heterogeneous clusters. Different from the homogeneous clusters consisting of a single type of nodes, there are two types of nodes in the heterogeneous clusters. The nodes used as CHs are sinks equipped with high capacity (e.g., data aggregation and routing ability), whereas the nodes used as non-CHs are common sensors with limited capacity (e.g., application-specified sensing ability). Apparently, the WSNs

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

composed of heterogeneous clusters can save substantial cost by reducing the number of high-capacity nodes that require expansive hardware. Besides, management is easier because there is no need to alternate the modes of nodes between being CHs and non-CHs [13][16]. Literatures [15], [20], and [21] have proposed three different node scheduling schemes to maximize the lifetime of a WSN with homogeneous clusters. However, the research on optimal node scheduling schema for maximizing the lifetime of heterogeneous two-tier WSNs is still at an early stage.

This paper proposes a new node scheduling methods for maximizing the lifetime of heterogeneous two-tier WSNs. A heterogeneous two-tier WSN must satisfy three constraints to function properly, i.e., the coverage constraint of sensors, the routing constraint of sinks, and the connectivity constraint between sensors and sinks. The coverage constraint requires sensors on the lower tier to form a satisfying coverage to the deployed area. The routing constraint requires sinks on the upper tier to form a connected network so that each CH can transmit data to a BS either directly (in one hop) [13] or indirectly (in multiple hops) [17]. The connectivity constraint ensures that every lower-tier sensor can communicate with one upper-tier sink. Thus every piece of information of the deployed area can be transmitted from sensors to sinks.

Indeed, it is difficult to use a single global optimization process for finding the optimal schedule that maximizes the number of sets that satisfy all the three constraints simultaneously. Therefore, this paper introduces a divide-and-conquer method. First, the sensors are scheduled for finding the maximum number of disjoint full cover sets. Second, based on the optimal schedule of sensors, sinks are allocated to the existing sets of sensors, aiming to build as many admissible sets that comply with the connectivity and routing constraints as possible. By combining the solutions in the above two steps, an optimal or near-optimal schedule can be obtained. Several approaches have been proposed for addressing the sensor scheduling problem in the first step [3]-[6]. In this paper, we focus on the second step of sink scheduling and propose a novel genetic algorithm (GA) [22] to address it. The proposed GA is designed based on the idea of minimizing the number of sinks in every admissible set, so that inadmissible sets have more opportunities to be improved. In order to evaluate the performance of the proposed GA, a series of random WSN cases are generated. Experimental results show that the proposed method can find optimal or near-optimal schedules at a fast speed.

The remainder of this paper is organized as follows. Section II describes the network model and defines the problem considered in this paper. Section III details the implementation of the proposed GA. Section IV shows the experimental study and related discussions. The conclusion of the whole paper is drawn in Section V.

II. Network Model and Problem Definition

A. Network Model

Fig. 1 (a) shows an example of the heterogeneous two-tier WSNs considered in this paper. As can be observed, the two-tier WSN contains a BS, a set of sensors, and a set of sinks.

The application-specified sensors are responsible for the sensing task. They are grouped in clusters and constitute the lower tier of the network. After capturing the data within the sensing range, sensors transmit the data to the sink in its own cluster in one hop. Sinks are equipped with data aggregation and forwarding abilities. In the network model of this paper, there is only one sink as the cluster head in each cluster of sensors. After receiving data from sensors in the cluster, the sink aggregates the data and forwards the aggregation results to the BS. The BS receives messages from the sinks and obtains a complete scene of the whole deployed area by analyzing the messages. Since the BS is generally remote from the other nodes, it is more energy-efficient to organize the sinks into a relay network that is connected with the BS at the closest point. Through the relay network, a sink that is far from the BS can forward its message in multiple hops. Sinks and BSs constitute the upper tier of the network. Fig. 1 (b) shows a hierarchical view of the above two-tier WSN.

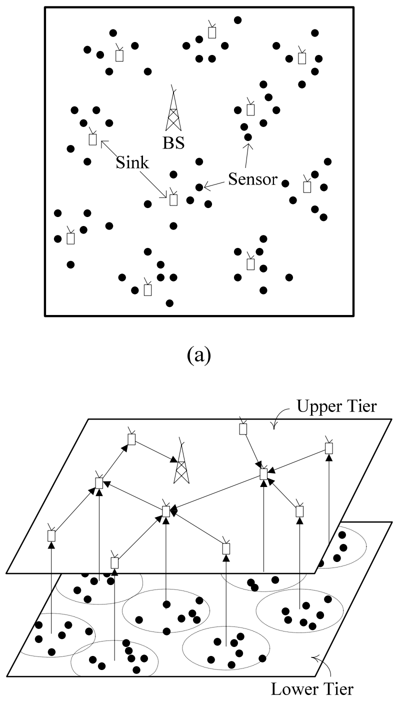


Fig. 1. An example of (a) the two-tier WSN and (b) its hierarchical view. In (b), the dot circles indicate the clusters; the arrows show the direction of data flow.

Both sensors and sinks are battery-powered. The batteries of sinks are generally more powerful because sinks consume energy at a higher rate. In this paper, we assume the battery of sensors and sinks can support them for the same length of time. Different from the sensors and sinks, the energy provision of BS is generally not constrained. Therefore, the energy conservation of BS is not a concern of our investigation.

B. Problem Definition

In order to maximize the network lifetime, the scheduling scheme should maximize the number of admissible sets. Suppose N sensors $\boldsymbol{SE}=\{SE_1,SE_2,\ldots,SE_N\}$ and M sinks $\boldsymbol{SI}=\{SI_1,SI_2,\ldots,SI_M\}$ have been deployed in an area A. The nodes are stationary and their positions are known to the schedule scheme through the Global Positioning System (GPS) or other GPS-free location algorithms. The problem considered in this paper can then be stated as maximizing the number U of admissible set with each set $\boldsymbol{S}_i$ (i=1,2,…,U) subject to the following constraints.

1) Coverage constraint. In this paper, the coverage constraint requires sensors in $\boldsymbol{S}_i$ to form a full coverage of the deployed area A. In other words, any point P in A must be covered by at least one sensor in $\boldsymbol{S}_i$. Assume all the sensors have the same sensing range r_s and their detection follows the deterministic disk model [13]. Then $\boldsymbol{S}_i$ satisfies the coverage constraint if and only if

$$\forall P\in A,\ \exists SE_j\in \boldsymbol{SE}_i,\ \|SE_j - P\| \le r_s, \tag{1}$$

where $\boldsymbol{SE}_i=\{SE_j \mid SE_j \in \boldsymbol{S}_i, j=1,2,\ldots,N\}$ is the set of sensors in $\boldsymbol{S}_i$ and $\|SE_j - P\|$ indicates the distance between SE_j and P. An illustration of the set satisfying the coverage constraint is shown in Fig. 2.

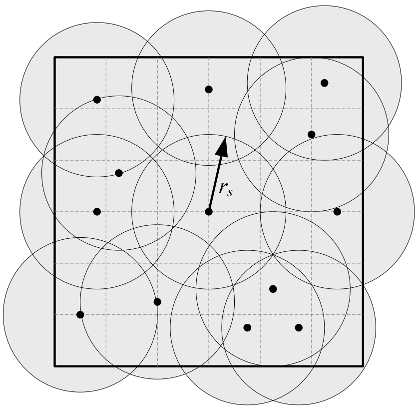


Fig. 2. Illustration of a set that satisfies the coverage constraint.

2) Routing constraint. As discussed in the previous part, the sinks compose a relay network for routing data to the BS efficiently. To guarantee the existence of the relay network, there must be a spanning tree that links all the sinks in $\boldsymbol{S}_i$ (as shown in Fig. 3). Consider a weighted graph G with the vertex set $\boldsymbol{V}=\boldsymbol{SI}_i=\{\boldsymbol{SI}_j \mid \boldsymbol{SI}_j\in\boldsymbol{S}_i,\ j=1,2,\ldots,M\}$, the edge set $\boldsymbol{E}=\{(SI_j,SI_k) \mid SI_j,SI_k\in\boldsymbol{SI}_i\}$, and the cost function $C{:}\boldsymbol{E}\rightarrow\Re^+$ as $C((SI_j,SI_k))=\|SI_j - SI_k\|$. Then S_i satisfies the routing constraint if and only if the minimum spanning tree T constructed from G is subject to

$$\forall(SI_j,SI_k)\in T,\ \|SI_j - SI_k\| \le r_t^{SI}, \tag{2}$$

where r_t^{SI} is the transmission range of sinks.

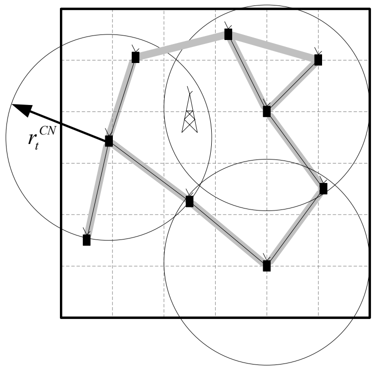


Fig. 3. Illustration of a set that satisfies the routing constraint. The gray bold lines connect the sinks that can communicate with each other directly and indicate the relay network constituted by the sinks in the set. The black thin lines indicate a spanning tree in the relay network.

3) Connectivity constraint. The connective constraint is to insure that all the information obtained by the sensors is delivered to the sinks. For this, every sensor must be able to transmit its data to at least one sink (as shown in Fig. 4). Suppose all the sensors have the same transmission range r_t^{SE}. $\boldsymbol{S}_i$ satisfies the connectivity constraint if and only if

$$\forall SE_j\in\boldsymbol{SE}_i,\ \exists SI_k\in\boldsymbol{SI}_i,\ \| SE_j - SI_k \| \le r_t^{SE}. \tag{3}$$

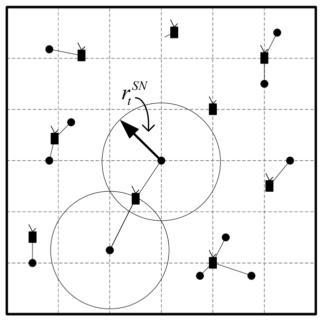


Fig. 4. Illustration of a set that satisfies the connectivity constraint. Each sensor in the set is connected to a sink in its transmission range with a solid line

Note that the routing and connectivity constraints do not address any specified routing protocols or clustering rules. It is the existence of the routing topology and the feasibility of clustering that are concerned in this paper. Besides the above three constraints, the schedule forbids any node to appear in two or more sets simultaneously, i.e.,

$$\forall i, j \in \{1,2,\ldots,U\},\ i \ne j,\ \boldsymbol{S}_i \cap \boldsymbol{S}_j = \varnothing. \tag{4}$$

This is because we have assumed the energy of sensors and sinks can support their activation for the same length of time. Once a set is activated, it continues to monitor the deployed area until a node in the set exhausts its energy.

Based on the idea of divide-and-conquer, the above problem can be divided into two sub-problems: 1) schedule sensors to maximize the number of full cover sets that satisfy the coverage constraint and 2) schedule sinks into the full cover sets to maximize the number of admissible sets that satisfy all the three constraints. The former sub-problem can be addressed by scheduling methods in flat WSNs [3]-[6]. This paper focuses on the later sub-problem and proposes a GA to solve it.

III. A Genetic Algorithm for Scheduling Sinks on the Upper-tier

In this section, a GA is proposed to schedule sinks for maximizing the lifetime of two-tier WSNs. There are three key components in a GA: 1) representation of chromosomes, 2) construction of fitness function, and 3) design of genetic operators. In this section, we detail the design of these three components and then give the complete process of the proposed GA.

A. The Key Components of the Proposed GA

1) Representation of Chromosomes

In a GA, a chromosome represents a solution to the optimization problem. Therefore, a chromosome in the proposed GA should define a schedule of sinks. Suppose U full cover sets have already been found in the sensor scheduling step. With every gene being mapped to a sink, the chromosome C_i can be represented as

$$C_i = \{g_{i1}, g_{i2}, \ldots, g_{iM}\},\ i = 1,2,\ldots,PS, \tag{5}$$

where $g_{ij}\in[1,U+1]$ is the index of the set that the sink SI_j belongs to, M is the number of sinks, and PS is the size of population. Note that when $g_{ij}=U+1$, the sink SI_j is actually excluded from the WSN and would not be activated at any time. With such a representation scheme, schedules found by the GA are forced to obey the disjoint constraint in (4). Moreover, the representation scheme can cover the whole solution space and make each chromosome represent a unique schedule.

2) Fitness Function

The fitness function in a GA evaluates the quality of chromosomes. A good fitness function can make the optimization process more efficient. Consider the aim of the proposed GA being maximizing the number of admissible sets. The fitness function is defined as

$$f(C_i) = \omega_1 \sum_{k=1}^{U} I(\boldsymbol{S}_k) + \omega_2 \sum_{k=1}^{U} P(\boldsymbol{S}_k), \tag{6}$$

It can be observed that the fitness function is composed of two components. The first component contains a predefined positive weight ω_1 and an indicator function $I(\boldsymbol{S}_k)$, which is defined as (7) to indicate whether $\boldsymbol{S}_k$ is an admissible set.

$$I(\boldsymbol{S}_k) = \begin{cases} 1, & \text{Eq. (1) - (3) satisfied} \\ 0, & \text{otherwise} \end{cases},\ k=1,2,\ldots,U. \tag{7}$$

The first component of the fitness function summarizes the number of admissible sets in the chromosome. Chromosomes that represent schedules with more admissible sets are supposed to have better fitness values. In this way, the fitness function encourages the chromosomes to evolve towards schedules with more admissible sets.

The second component of the fitness function is constituted by a predefined negative weight ω_2 and a penalty function $P(\boldsymbol{S}_k)$. The penalty function embeds an idea: the more redundant sinks the admissible sets contain, the less opportunity for the remaining inadmissible sets to be improved. Thus, suppose the sensor SE_t ($SE_t \in \boldsymbol{S}_k$) can transmit data to v_t sinks in $\boldsymbol{S}_k$, the penalty of $\boldsymbol{S}_k$ is calculated by

$$P(\boldsymbol{S}_k) = \frac{1}{|\boldsymbol{SE}_k|} \sum_{SE_t \in \boldsymbol{S}_k} H(v_t), \tag{8}$$

where $H(\cdot)$ is a function defined as

$$H(v_t) = \begin{cases} v_t - 1, & v_t \geq 1 \\ |\boldsymbol{SI}_k|, & v_t = 0 \end{cases}. \tag{9}$$

A special case is that no sink has been scheduled into $\boldsymbol{S}_k$, i.e., $|\boldsymbol{SI}_k|=0$. In that situation, the penalty is set to one. From (8) and (9), it can be deduced that a set receives heavy penalty under two situations: 1) the number of unconnected sensors in the set is large and 2) the sinks in the set have serious redundancy. Generally speaking, the number of unconnected sensors will reduce when more sinks are assigned to the set. If a set involves many sinks but still has some sensors unconnected, the schedule is considered especially inefficient and the penalty of this set would be particularly large. By adding up the penalties of all the sets, the second component of the fitness function can bias the search process towards schedules that allocates sinks more efficiently.

In order to insure the effectiveness of the fitness function, the values of ω_1 and ω_2 must be adjusted. The definition of the penalty function determines that the penalty value of a set will not exceed the range of $[0, |\boldsymbol{SI}_k|]$. Therefore, if ω_1 and ω_2 satisfy

$$\left|\frac{\omega_1}{\omega_2}\right| > \max_{1\le k\le U}(|\boldsymbol{SI}_k|), \tag{10}$$

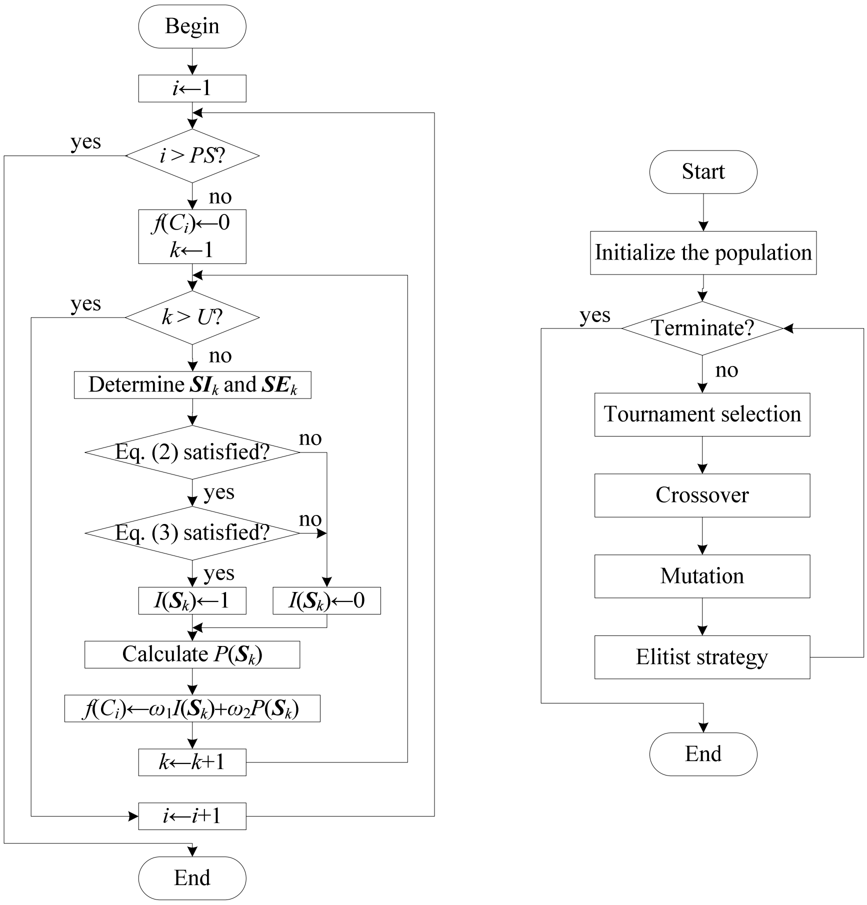
an admissible set will always contribute a positive increment to the fitness value because $\omega_1+\omega_2\cdot P(\boldsymbol{S}_k)>\omega_1+\omega_2\cdot \max_{1\le k\le U}(|\boldsymbol{SI}_k|)>0$. In the opposite, an inadmissible set will reduce the fitness by a negative value $\omega_2\cdot P(\boldsymbol{S}_k)$. By doing so, the fitness value of a chromosome increases as the number of admissible sets rises.

The process of evaluating the population with the above fitness function is shown in Fig. 5 (a).

3) Genetic Operators

The proposed GA comprises three genetic operators: selection, crossover, and mutation, which are defined as follows.

- Selection. The classical tournament selection [22] is adopted. The best chromosome within the TS chromosomes that are randomly selected from the population is chosen as a parent chromosome.
- Crossover. As shown by (11), the genes from two randomly selected parents C_i and C_j (i, j=1,2,…,PS) are combined with equal probability to generate a new chromosome $C'_k = (g'_{k1}, g'_{k2}, \ldots, g'_{kM})$,



(a) (b)

Fig. 5. Flowchart of (a) the population evaluation process and (b) the complete process of the proposed GA.

$$g'_t = \begin{cases} g_{it}, & q < 0.5 \\ g_{jt}, & q \ge 0.5 \end{cases}, \; t=1,2,\ldots,M, \tag{11}$$

where q is a uniformly distributed random number in (0,1). For each generation, the above procedure is repeated until PS offspring $C'_1, C'_2, \ldots, C'_{PS}$ are generated.

- Mutation. After the crossover, the genes in an offspring C'_k are changed as

$$g''_{kt} = \begin{cases} g'_{kt}, & q > P_m \\ g, & q \le P_m \end{cases}, \; t=1,2,\ldots,M, \tag{12}$$

where P_m is the mutation probability, q is a uniformly distributed random number in (0,1), and $g \neq g'_t$ is a random value in $[1,U+1]$. In every generation, (12) is applied to all the offspring generated by the crossover. The mutated chromosomes $C''_k = (g''_1, g_2, \ldots, g''_M)$ ($k=1,2,\ldots,PS$) constitute a new population.

B. The Complete Process of the Proposed GA

With the key components specified as above, the complete process of the proposed GA is summarized in the flowchart of Fig. 5 (b). The following paragraphs explain the process step by step.

Step 1) Initialization: The population is initialized by assigning random numbers in the range of $[1,U+1]$ to the genes of each chromosome. The chromosomes are then evaluated by the fitness function. The best chromosome C_b is used to initialize the historical best record C_{hb}.

Step 2) Termination check: If the evolution termination criterion, e.g., the maximum number of fitness function evaluations, has been met, the schedule of C_{hb} is returned as the result and the optimization process terminates. Otherwise, the optimization process continues with Step 3).

Step 3) Selection: The tournament selection is performed.

Step 4) Reproduction: The crossover and mutation operators defined in Section III-A are performed to generate a new population.

Step 5) Elitist strategy: After reproduction, the chromosomes in the new population are evaluated with the fitness function. The currently worst chromosome C_w is replaced by the historical best record C_{hb}. The currently best chromosome C_b is compared with C_{hb}. If $f(C_b) > f(C_{hb})$, C_{hb} is updated with C_b.

Step 6) After Steps 3) to 5), the population of the next generation has been formed. The optimization process returns to Step 2) and begins another generation of evolution.

In the above GA process, the time complexity of selection and reproduction is no more than $O(PS\times M)$. The most time-consuming component is the evaluation of the population. For each set $\boldsymbol{S}_k$: the time complexity for checking the routing constraint is $|\boldsymbol{SI}_k|^2$; the time complexity for calculating the penalty is $|\boldsymbol{SI}_k|\times|\boldsymbol{SE}_k|$. Therefore, the time complexity to evaluate the whole population is

$$PS\times\sum_{k=1}^{U}|\boldsymbol{SI}_k|(|\boldsymbol{SI}_k|+|\boldsymbol{SE}_k|), \tag{13}$$

which is no larger than $PS\times(M+N)^2$.

IV. Experimental Study

In this section, the proposed GA is evaluated by a series of simulation on some randomly generated test cases. The test cases are designed with the optimal solutions known. Therefore, the effectiveness and the performance of the proposed algorithm can be directly examined.

A. Experimental Setups

Procedure TEST_CASE_GENERATION
Input:
L: the length of the deployed area
W: the width of the deployed area
r_s: the sensing range of sensors
N: the number of sinks
per: the proportion between the numbers of sensors and sinks
Output:
U: the number of covering sets in the schedule of sensors
$\boldsymbol{SE}$ $[SE_i]$: SE_i=<x_i,y_i> indicates the position of the i-th sensor
D $[d_i]$: d_i is the index of the set that SE_i belongs to
M : the number of sinks
$\boldsymbol{SI}$ $[SI_i]$: SI_i=<x_i,y_i> indicates the position of the i-th sink
r_t^{SE} : communication range of sensors
r_t^{SI} : communication range of sinks
Begin
  //randomly deploy N sinks in the $L\times W$ area
  For i:=1 **to** N
    SI_i:=RAND(L,W);
  End
  //schedule the sensors
  <D,U>:=SE_SCHEDULIN(L,W,r_s,$\boldsymbol{SE}$);
  //schedule the sinks
  M:=0; r_t^{SN} :=0; r_t^{CN} :=0;
  For i:=1 **to** U
    $\boldsymbol{SE}_i$:=$\{SN_j \mid d_j=i, j=1,2,\ldots,N\}$;
    K:=$per\times|\boldsymbol{SN}_i|$;
    $\boldsymbol{C}$:=K_CLUSTERING($\boldsymbol{SE}_i$,K);
    $tr_t^{SN} := \max_{1\le k\le K, s_j\in \boldsymbol{SN}_i}(\|C_k - SE_j\|)$;
    tr_t^{CN} :=PRIM_MST(C);
    If $tr_t^{SE} > r_t^{SE}$ **Then** $r_t^{SE} := tr_t^{SE}$; **End**
    If $tr_t^{SI} > r_t^{CN}$ **Then** $r_t^{SI} := tr_t^{SI}$; **End**
    For j:=M+1 **to** M+K
      SI_j=<$C_{j-K,1}$, $C_{j-K,2}$>;
    End
    M:=M+K;
  End
End

Fig. 6. Pseudo-code of the test case generation method. RAND(L,W) is a function generating random numbers in the range of $[0,L]\times[0,W]$; SE_SCHEDULIN(L,W,r_s,$\boldsymbol{SE}$) is a function based on [6] to schedule sensors and to return the schedule D as well as the number U of the full cover sets; K_CLUSTERING($\boldsymbol{SE}_i$,K) is a function to cluster the nodes in $\boldsymbol{SE}_i$ into K clusters and return the array $\boldsymbol{C}$ of the positions of the cluster centers

With the deployed area set as a 50×50 rectangle, the sensing range of sensors set to R, and the proportion between the numbers of sensors and sinks fixed at 0.2, nine test cases are generated by the method introduced in Fig. 6. Table I tabulates the configurations of the nine test cases, including the numbers of sensors and sinks (M and N), the sensing range of sensors (r_s), the transmission range of sensors and sinks (r_t^{SE} and r_t^{SI}), and the number of admissible sets (U) in the optimal schedule.

There are a total of five parameters in the proposed GA: the size of population PS, the size of tournament selection TS, the mutation probability P_m, and the weights ω_1 and ω_2 in the fitness function. Among these parameters, PS, TS, and P_m are common parameters of GAs. As recommended in literature [22], these three parameters are set as PS=20, TS=8, and P_m=0.05. The relationship between ω_1 and ω_2 has been discussed in Section III-A. In order to satisfy the constraint in (10), we set ω_1=M and ω_2=−1.0, which has been empirically proven good. The proposed GA terminates when the number of fitness function evaluations exceeds the predefined upper boundary 50,000. For fair evaluation, the proposed algorithm is executed for 30 independent times and the statistical results are used for analysis. All the experiments are run on a Dell computer with an Intel Core 2 Duo 2.33GHZ CPU and a ram of 966 MB.

TABLE I
TEST CASES USED IN EXPERIMENTAL STUDY

Case No.	N	M	r_s	r_t^{SE}	r_t^{SI}	U
1	100	21	20	29	31	7
2	300	60	15	30	35	16
3	300	60	20	32	36	32
4	400	80	10	16	25	9
5	400	84	15	30	34	23
6	500	100	8	15	22	7
7	500	98	10	22	28	15
8	1000	200	5	7	12	5
9	1000	198	8	15	21	17

TABLE II
RESULTS OF THE PROPOSED GA

Case No.	Best	Worst	Avg	avgFEs	SR(%)
1	7	7	7	236	100
2	16	16	16	728	100
3	32	32	32	3906	100
4	9	8	8.87	11019	86.67
5	23	23	23	1108	100
6	7	7	7	2452	100
7	15	15	15	4617	100
8	5	4	4.43	23945	43.33
9	17	15	16.33	24454	43.33

B. Experimental Results

Table II tabulates the results of the experiments. In the table, 'Max', 'Min', and 'Avg' represent the maximum, minimum, and average numbers of the admissible sets found by the algorithm. 'SR' is the rate that the algorithm successfully found the optimal solution in 30 runs. 'avgFEs' indicates the average number of fitness function evaluations

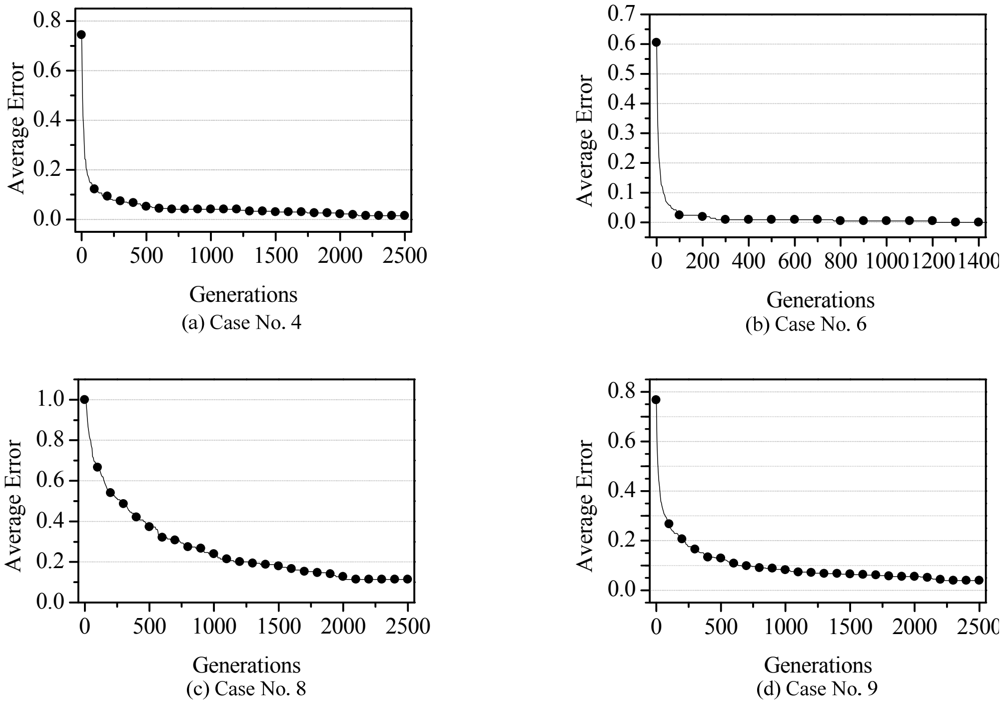


Fig. 7. Evolution curves of the proposed GA on (a) Case No. 4, (b) Case No. 6, (c) Case No. 8, and (d) Case No. 9.

for the algorithm to obtain the optimal solution.

From Table II, it can be observed that the proposed GA is able to find the optimal solution in every test case. Actually, the algorithm promises to find the optimal solution with 100% successful rate for all the test cases except for cases No. 4, 8, and 9. Even on cases No. 4, 8, 9, the successful rates are no less than 40%. On the other hand, on cases No. 1 to 7, the value of 'avgFEs' is smaller than 12,000, that is, 600 generations, suggesting that the algorithm is able to find the optimal solutions quickly when the scale of the problem is under a thousand nodes.

To better understand the optimization process, Fig. 7 depicts the evolution curves for cases No. 4, 6, 8, and 9. In this figure, the horizontal axis indicates the number of generations, while the vertical axis is the relative error between the numbers of admissible sets in the best-so-far solution and the optimal solution averaged over 30 runs. Therefore, the larger the value on the vertical axis, the worse the solution is. It can be observed that at the beginning of the algorithm, the random initialization generates poor solution with relative error over 0.6. Then in the early phase of the evolution, as shown by the steep curves in Fig. 7, the algorithm quickly improves the solution quality and reduces the error. However, the evolution curve becomes smooth in the later phase, meaning that the algorithm slows down in the later evolution.

Fig. 8 displays the average CPU time for the proposed GA to find the optimal solution in 30 runs. It can be observed that the average CPU time spent on cases No. 1 to 7 is less than 1 second.

Concluded from the above, the proposed GA is able to find optimal or near-optimal schedules in limited computation time. The experimental results have confirmed that the proposed GA is effective and efficient for scheduling the upper-tier nodes in the two-tier WSN.

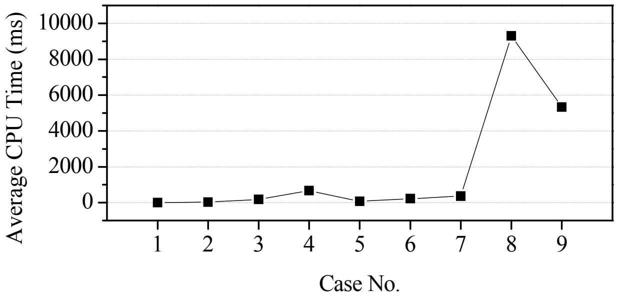


Fig. 8. Illustration of the average CPU time cost by the proposed GA on each test case.

V. CONCLUSION

This paper considers a node scheduling method to maximize the lifetime of the two-tier WSNs. Based on the idea of divide-and-conquer, the proposed method suggests scheduling sensors on the lower tier and sinks on the upper tier respectively. Since the scheduling of sensors can be analogous to the node scheduling in flat WSNs and some approaches have already been provided, this paper focuses on the scheduling of sinks and proposes a GA to address the problem. The proposed GA is based on the idea of minimizing the number of sinks in the admissible sets so that the inadmissible sets gain more opportunities to be improved. Experiments on a range of random test cases show that the proposed GA is effective and efficient in finding the optimal or near optimal schedule of sinks.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor network: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] I. F. Akyildiz, T. Melodia, K. R. Chowdury, "Wireless multimedia sensor network: a survey," *IEEE Wireless Communication*, vol. 14, no. 6, pp. 32-39, Dec. 2007.

[3] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, pp. 89-124, Mar. 2005.

[4] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless and C.Gill, "Integrated coverage and connectivity configuration and energy conservation in sensor networks," *ACM Trans. on Sensor Networks*, vol. 1, no. 1, pp. 36-72, Aug. 2005.

[5] D. Tian and N. D. Georganas, "Connectivity maintenance and coverage preservation in wireless sensor network," *Ad Hoc Network*, vol. 3, No. 6, pp. 744-761, Nov. 2005.

[6] Y. –L. Li, X. –M. Hu, and J. Zhang, "A new genetic algorithm for the SET *k*-cover problem in wireless sensor networks," *Proc. IEEE SMC 2009*, Oct. 2009, in press.

[7] D. J. Baker and A. Ephremides, "The architecture organization of a mobile radio network via a distributed algorithm," *IEEE Trans. on Communications*, vol. 29, no. 11, pp. 1694-1701, Nov. 1981.

[8] J. Pan, L. Cai, Y. T. Hou, Y. Shi, and S. X. Shen, "Optimal base-station locations in two-tiered wireless sensor network," *IEEE Trans. on Mobile Computing*, vol. 4, no. 5, Sept. 2005.

[9] Y. T. Hou, Y. Shi, and H. D. Sherali, "Optimal base station selection for anycast routing in wireless sensor networks," *IEEE Trans. on Vehicular Technology*, vol. 55, no. 3, pp. 813-821, May 2006.

[10] Y. T. Hou, Y. Shi, J. Pan, and S. F. Midkiff, "Maximizing the lifetime of wireless sensor networks through optimal single-session flow routing," *IEEE Trans. on Mobile Computing*, vol. 5, no. 9, pp. 1255-1276, Sept. 2006.

[11] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provision and relay node placement for wireless sensor networks," *IEEE Trans. on Wireless Communication*, vol. 4, no. 5, pp. 2579-2590, Sept. 2005.

[12] J. –H. Ryu, S. Song, and D. –H. Cho, "New clustering schemes for energy conservation in two-tiered mobile ad hoc networks," *IEEE Trans. on Vehicular Technology*, vol. 51, no. 6, Nov. 2002.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," *Proc. IEEE Hawaii Int'l Conf. System Science*, pp. 1-10, 2002.

[14] K. Akkaya and M. Younis, "A survey of routing protocols in wireless sensor networks," *Ad Hoc Network*, vol. 3, no. 3, pp. 325-349, May, 2005.

[15] S. F. Hwang, Y. Y. Su, Y. Y. Lin, and C. R. Dow, "A cluster-based coverage-preserved node scheduling scheme in wireless sensor networks," *Proc. 3rd Ann. Int'l Conf. Mobile and Ubiquitous Systems: Networking and Services*, pp. 1-7, 2006.

[16] L. M. Arboleda and N. Nasser, "Comparison of clustering algorithms and protocols for wireless sensor networks," *Proc. Canadian Conf. Electrical and Computer Eng.*, pp. 1787-1792, 2006.

[17] V. Mhatre and C. Rosenberg, "Homogeneous vs Heterogeneous clustered sensor networks: a comparative study," *Proc. IEEE Int'l Conf. Comm.*, vol. 6, pp. 3646-3651, Jun. 2004.

[18] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer, T. W. Kenny, K. H. Law, and Y. Lei, "Two-tiered wireless sensor network architecture for structural health monitoring," *Proc. Int'l Symposium on Smart Structures and Materials*, 2003.

[19] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient distributed clustering approach for Ad Hoc sensor network," *IEEE Trans. on Mobile Computing*, vol. 3, no. 4, Oct. 2004.

[20] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," *Proc. IEEE Int'l Workshop Wireless Sensor Networks and Applications (WSNA'02)*, pp. 32-41, 2002.

[21] A. Chaman and S. Pierre, "On the planning of wireless sensor networks: energy-efficient clustering under the joint routing and coverage constraint," *IEEE Trans. on Mobile Computing*, vol. 8, no. 8, Aug. 2009.

[22] Z. Michalewicz, Genetic Algorithms + Data Structure = Evolution Programs, Springer, 1998.