Hybrid Genetic Algorithm Using a Forward Encoding Scheme for Lifetime Maximization of Wireless Sensor Networks

Xiao-Min Hu, *Student Member, IEEE,* Jun Zhang, *Senior Member, IEEE,* Yan Yu, Henry Shu-Hung Chung, *Senior Member, IEEE,* Yuan-Long Li, Yu-Hui Shi, *Senior Member, IEEE,* and Xiao-Nan Luo

***Abstract*—Maximizing the lifetime of a sensor network by scheduling operations of sensors is an effective way to construct energy efficient wireless sensor networks. After the random deployment of sensors in the target area, the problem of finding the largest number of disjoint sets of sensors, with every set being able to completely cover the target area, is nondeterministic polynomial-complete. This paper proposes a hybrid approach of combining a genetic algorithm with schedule transition operations, termed STHGA, to address this problem. Different from other methods in the literature, STHGA adopts a forward encoding scheme for chromosomes in the population and uses some effective genetic and sensor schedule transition operations. The novelty of the forward encoding scheme is that the maximum gene value of each chromosome is increased consistently with the solution quality, which relates to the number of disjoint complete cover sets. By exerting the restriction on chromosomes, the forward encoding scheme reflects the structural features of feasible schedules of sensors and provides guidance for further advancement. Complying with the encoding requirements, genetic operations and schedule transition operations in STHGA cooperate to change the incomplete cover set into a complete one, while the other sets still maintain complete coverage through the schedule of redundant sensors in the sets. Applications for sensing a number of target points, termed point-coverage, and for the whole area, termed area-coverage, have been used for evaluating the effectiveness of STHGA. Besides the number of sensors and sensors' sensing ranges, the influence of sensors' redundancy on the performance of STHGA has also been analyzed. Results show that the proposed algorithm is promising and outperforms the other existing approaches by both optimization speed and solution quality.**

***Index Terms*—Coverage, disjoint set covers problem, encoding scheme, evolutionary algorithm, genetic algorithm, memetic algorithm, redundancy, schedule, SET *k*-cover problem, wireless sensor network.**

I. Introduction

WIRELESS sensor networks (WSNs) use a large quantity of sensors in a target area for performing surveillance tasks such as environmental monitoring, military surveillance, animal tracking, and home applications [1]–[6]. Each sensor collects information by sensing its surrounding region and transfers the information to a sink (also called a data center) via wireless transmission. Because of the features of sensors, WSNs have been implemented in harsh environments such as in the deep sea, arctic areas, and hazardous war zones. Different from other battery-powered apparatuses, recharging a sensor's battery is generally impossible. Although solar and wind energy can be used, such energy supplies are not reliable. Equipped with limited energy supplies, WSNs are much more demanding on energy conservation than the other kinds of networks. How to maximize the network's lifetime is a critical research topic in WSNs.

Various methods have been proposed in the literature for organizing energy efficient WSNs [7]–[15], in which sensing coverage and network connectivity are two fundamental issues. Since it has been proven that the network connectivity of active sensors in complete coverage is guaranteed by having the communication range of each sensor at least twice of its sensing range [16]–[18], only the sensing coverage problem is considered in this paper. There are two ways for deploying sensors to completely cover a target area, i.e., controlled deployment and random deployment [2]. Controlled deployment is to deploy sensors based on a well-designed plan. Examples for designing a required deployment plan can be referred to [19]–[23]. Most of the controlled deployment methods aim at assigning the smallest number of sensors under the cost limitation in an area [19]–[23], whereas some methods consider dispatching a set of mobile sensors to satisfy the coverage and connectivity requirements [23]–[25]. However,

Manuscript received May 8, 2009; revised August 30, 2009 and December 3, 2009. Date of publication April 22, 2010; date of current version October 1, 2010. This work was supported in part by the National Natural Science Foundation of China Joint Fund with Guangdong under Key Project U0835002, by the National High-Technology Research and Development Program ("863" Program) of China No. 2009AA01Z208, by the National Science Foundation of China under Project 60 975 080, by the Suzhou Science and Technology Project under Grant SYJG0919, by the Sun Yat-Sen Innovative Talents Cultivation Program for Excellent Tutors No. 35000-3126202, and by the Cultivation Fund of the Key Scientific and Technical Innovation Project of the Ministry of Education of China under Grant 706045.

X.-M. Hu, J. Zhang, Y.-L. Li, and X.-N. Luo are with the Department of Computer Science, Sun Yat-Sen University, Guangzhou 510275, China, and also with the Key Laboratory of Digital Life (Sun Yat-Sen University), Ministry of Education (J. Zhang is the corresponding author. e-mail: junzhang@ieee.org).

Y. Yu is with the Institute of Microelectronics, Dalian University of Technology, Dalian 116024, China.

H. S.-H. Chung is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong.

Y.-H. Shi is with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2010.2040182

1089-778X/$26.00 © 2010 IEEE

sometimes it is difficult to place sensors to the positions prearranged, because the number of sensors is too large to place the sensors one by one, or the position is difficult to be approached. In those cases, random deployment is performed to place the sensors (e.g., dropping from a plane) to the target area.

In order to ensure complete coverage to the target area, a large number of sensors are deployed densely to the area by random deployment. Scheduling is an important mechanism for controlling sensors' activities so as to prolong a network's lifetime [1], [26]. A sensor generally has two operation modes, i.e., an active mode and a sleep mode. When in an active mode, a sensor can carry out its full operations, such as sensing, computation, and communication. To maintain those operations, sensors need to consume a relatively large amount of energy. In contrast, a sensor in a sleep mode uses only a small amount of energy and can be awoken in a scheduled working interval for full operations. Since a subset of sensors in the area can already cover the target area completely, the other sensors can be scheduled to be in the sleep mode to save energy.

Different scheduling rules determine when sensors change to be active or sleep. In localized and distributed realizations, sensors periodically investigate their neighborhood and decide whether to change their operation modes [18], [27]–[30]. In those situations, the lifetime of a WSN for accomplishing the surveillance task cannot be guaranteed. Moreover, in order to save energy, many existing scheduling methods (e.g., [27]–[33]) only consider selecting a subset of sensors which can satisfy the surveillance task using the minimum energy, but the influence of the selected sensors for the lifetime of the WSN is neglected. If the selected sensors happen to be the critical points in the network, the WSN may fail to work after the selected sensors run out of energy. Therefore, exploiting the redundancy in a WSN and finding out the possible scheduling sequence of sensors can maximize the lifetime of the network. This paper aims at finding the maximum number of disjoint complete sensor cover sets in a WSN and assigns the sensors to the sets for maximizing the lifetime of the network. Each sensor cover set forms complete coverage to the target area and the WSN can fulfill the surveillance task with only one set of sensors active at any time. Finding the maximum number of disjoint complete sensor cover sets in a WSN is called the disjoint set covers problem or the SET k-cover problem, which has been proven to be nondeterministic polynomial (NP)-complete [34], [35].

Finding the optimal complete coverage scheme in WSNs is not easy, because the number of sensors in a target area is so huge that the computation is time-consuming. Cardie and Du [34] proposed a "maximum covers using mixed integer programming (MC-MIP)" algorithm to find the maximum number of disjoint complete cover sets for covering a set of target points. They transformed the problem into a maximum-flow problem and then formulated it as a mixed integer programming. By using a branch and bound method, MC-MIP acts as an implicit exhaustive search which guarantees finding the optimal solution. However, as the numbers of sensors and targets become larger, the running time of MC-MIP increases exponentially. Slijepcevic and Potkonjak [35] proposed a greedy deterministic approach called the "most constrained–minimally constraining covering (MCMCC)" heuristic to completely cover the target area. MCMCC cannot guarantee finding the optimum, but it works much faster than MC-MIP for problems in a large scale. In MCMCC, a function is defined favoring the sensor which covers the most constrained field, whereas the other fields covered by the sensor are minimally constraining. Whether a field is constrained or not depends on the number of sensors that can cover the field. Each complete cover set in MCMCC is constructed by selecting sensors according to the heuristic objective function. The work in [36]–[41] also provides methods for constructing working subsets of sensors, but their solutions trade off complete coverage in exchange for prolonging the network lifetime. For example, in the randomized scheduling methods in [36], sensors are randomly assigned to multiple working subsets of sensors. For each subset of sensors, the algorithm used an extra-on rule for guaranteeing network connectivity and then updated the working schedule accordingly. Lin and Chen [37] later improved the approach of [36] by detecting and eliminating coverage holes in the subsets. Abrams *et al.* [38] designed three approximation algorithms for a variation of the SET k-cover problem. However, none of the three algorithms guarantees complete coverage.

In addition to heuristic methods, genetic algorithms (GAs) have also been applied. GAs [42] are population based search algorithms, which simulate biological evolution processes and have successfully solved a wide range of NP-hard optimization problems [43]–[47]. Compared with MC-MIP and MCMCC, using a GA for finding the maximum number of disjoint complete cover sets is expected to search the domain more effectively and reduce the computation time. Lai *et al.* [48] introduced a GA for point-coverage problems. They termed it the genetic algorithm for maximum disjoint set covers (GAMDSC) and encoded each gene in the chromosome as an integer index of the set that the sensor joined. Using traditional genetic operations and a scatter operator, their algorithm was reported to be able to get near-optimal solutions. However, it can be observed that their algorithm lacks the consideration for redundant sensors in cover sets and the guidance for joining sensors to form complete coverage. Their algorithm is only suitable when the numbers of targets and sensors are small. Moreover, the problems addressed by MC-MIP and GAMDSC are point-coverage problems, whereas MCMCC can be applied to both point-coverage and area-coverage problems. Area-coverage involves a much larger number of coverage targets than point-coverage, because each field in the target area must be completely covered.

In this paper, an enhanced GA is proposed, aiming at solving disjoint set covers problems for maximizing the WSN lifetime. The proposed algorithm, termed the schedule transition hybrid genetic algorithm (STHGA), can be applied to both point-coverage and area-coverage disjoint set covers problems. The distinct feature of STHGA is that it adopts a forward encoding scheme for the representation of chromosomes in the population and uses some effective genetic and sensor schedule transition operations. The forward encoding scheme is novel because the maximum gene value of each chromosome is increased consistently with the solution quality, which relates to the number of disjoint complete cover sets. In the forward encoding scheme, each gene in a chromosome maps to a sensor. The forward encoding scheme exerts a restriction on each chromosome that, except for the largest gene value, the genes having the same value form a complete sensor cover set respectively. Such an encoding scheme not only reflects the structural features of feasible schedules of sensors in chromosomes, but also provides guidance for further advancement, because the primary task of the algorithm is to schedule sensors to change the unique incomplete cover set into a complete one, without influencing the other sets' complete coverage to the targets. The genetic operations and schedule transition operations in the proposed STHGA are designed based on the forward encoding scheme and they cooperate to search for the maximum number of disjoint complete cover sets. Moreover, the schedule transition operations utilize the redundancy information among the scheduled sensors for finding better scheduling schemes. The performance of the proposed STHGA has been compared with the state-of-the-art MCMCC heuristic [35] and GAMDSC [48]. Results show that the proposed algorithm can achieve high-quality solutions with a much faster optimization speed.

The remainder of this paper is organized as follows. Section II presents the definition of the optimization problem and makes some discussions on the estimation of the upper limit of the maximum number of disjoint complete cover sets, the redundancy rate of a deployment, and the way of measuring the coverage percentage. Section III describes the implementation of the proposed algorithm in detail, including the encoding method of chromosomes, the design of the fitness function, and the operations. A series of experiments are conducted and the results are analyzed in Section IV to illustrate the performance of the proposed algorithm. Finally, conclusions of our research and guidelines for future work are given in Section V.

II. Problem Definition and Discussions

A. Problem Definition

In order to prolong the lifetime of WSNs, the number of disjoint complete cover sets of sensors should be maximized. Suppose a set $\boldsymbol{S} = \{s_1, s_2, ..., s_N\}$ of sensors are deployed in an $L \times W$ area, the objective of the sensor set covers problem is to find the maximum number T of disjoint complete cover sets and the corresponding cover sets S_i, satisfying:

1) each set $\boldsymbol{S}_i = \{s_{i_1}, s_{i_2}, ..., s_{i_{|S_i|}}\} \subseteq \boldsymbol{S}$ of sensors forms complete coverage to the target area, where $|\boldsymbol{S}_i|$ is the number of sensors that are activated in the ith schedule, $i = 1, 2, ..., T$;
2) each sensor belongs to no more than one cover set, that is

$$\boldsymbol{S}_i \bigcap \boldsymbol{S}_j = \phi \tag{1}$$

where $i \neq j$, i, $j = 1, 2, ..., T$.

As each sensor s_i monitors a certain area, complete coverage to a target area means that the whole area is under-monitored by sensors. An illustration of complete coverage to an $L \times W$ area is shown in Fig. 1. Each sensor s_i in the figure covers the area within a sensing range r_i. Note that in practical applications, the sensing area covered by a sensor may not be a circular area, but can be any irregular shapes. For simplicity, in this paper we only consider the case that the sensing area of a sensor is a circle.

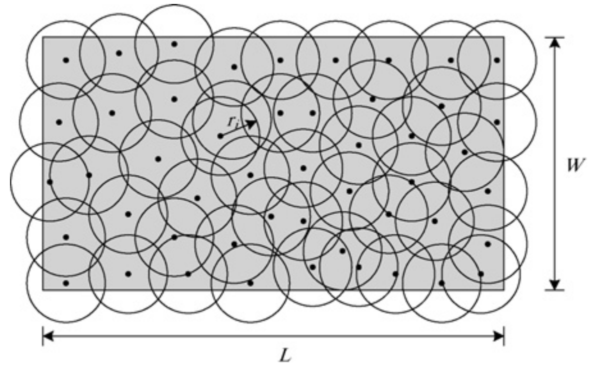


Fig. 1. Illustration of complete coverage to an $L \times W$ area. Each sensor s_i is marked as a dot and has a sensing range r_i. Each circle represents the sensing area of the corresponding sensor.

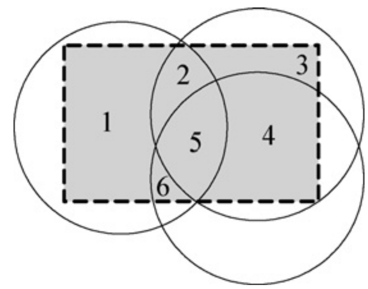


Fig. 2. Example of six fields formed by three sensors. The area contained in the same field is covered by the same set of sensors.

B. Discussions

The maximum disjoint complete cover set number T of the above maximization problem depends on the size of the target area, the total number of sensors, the sensors' locations, and their sensing ranges. When all of the sensors are activated, the whole target area must be completely covered. Otherwise, the deployment of sensors fails.

1) *Upper Limit of T:* It can be observed from Fig. 1 that the areas covered by sensors overlap each other, forming separate fields [35]. The area contained in the same field is covered by the same set of sensors. An example of fields is shown in Fig. 2, where six fields are formed by three sensors. When all of the sensors are activated and the fields are formed, the upper limit of T, which is denoted as $\tilde{T}$, can be estimated as the minimum number of sensors that cover a field in the target area as

$$\tilde{T} = \min_{j=1,2,...,n_F} (|\boldsymbol{F}_j|) \tag{2}$$

where $\boldsymbol{F}_j$ denotes the set of sensors that cover the field j, n_F is the number of fields formed by all of the sensors. In the

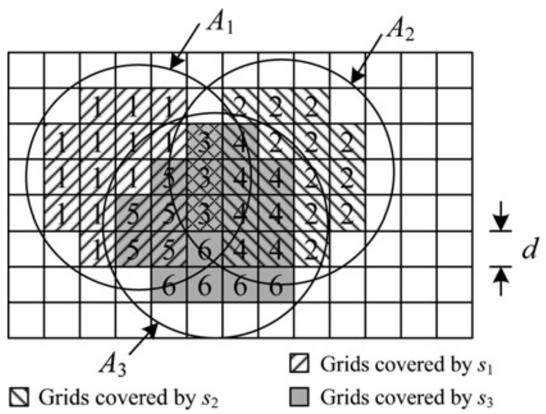


Fig. 3. Example for evaluating the coverage percentage of sensors to an area by dividing the target area into grids with a size d. The three circular sensing areas A_1, A_2, A_3 are covered by s_1, s_2, and s_3 respectively. The number in grids indicates their field indexes. Note that the grids contained in the same field are covered by the same set of sensors.

case of Fig. 2, the values of $|\boldsymbol{F}_1|, |\boldsymbol{F}_2|, ..., |\boldsymbol{F}_6|$ are 1, 2, 1, 2, 3, 2, respectively, so the value of $\tilde{T}$ is 1.

In fact, the calculation of $\tilde{T}$ is also a well-known K-coverage problem [20], which can be addressed by polynomial-time algorithms. Because there are fields only covered by $\tilde{T}$ sensors, the maximum number of disjoint complete cover sets is no larger than $\tilde{T}$ (i.e., $T \leq \tilde{T}$).

2) *Redundancy Rate of a Deployment:* According to Williams [49], the minimum number M of sensors for complete coverage to an $L \times W$ area satisfies

$$\frac{M\pi R^2}{LW} = \frac{2\pi}{\sqrt{27}} \tag{3}$$

where all of the sensors have the same sensing range as R. Therefore, for the N sensors in a successful deployment in this paper, we have

$$\eta = \frac{N}{\tilde{T}}\frac{\pi R^2}{LW} \geq \frac{2\pi}{\sqrt{27}} \approx 1.21 \tag{4}$$

where η is termed the redundancy rate. It will be shown in Section IV-C-3 that disjoint complete cover sets are much easier to be found in a problem with a larger η than in a problem with a smaller η.

3) *Calculation of the Coverage Percentage:* In practice, it is difficult to calculate the exact coverage percentage of the sensors to an $L \times W$ area. Therefore, we divide the target area into grids to approximate the coverage percentage. Only the grids being totally inside a sensor area are considered as being covered. For example (as Fig. 3), there are three active sensors s_1, s_2, and s_3 in a target area, which is composed of grids with a size d. The sensors' circular sensing areas are denoted as A_1, A_2, and A_3. The numbers of grids covered by the three sensors are counted as 21, 21, and 20, respectively. The total number of grids covered by the sensors is 44. Therefore, the coverage percentage of the sensors to the target area with 13×8 grids is approximately $\frac{44}{13\times 8} \approx 42.3\%$. Although the estimated coverage percentage is smaller than the actual coverage percentage, there are no *blind points* in the target area when the estimated coverage percentage is 100%.

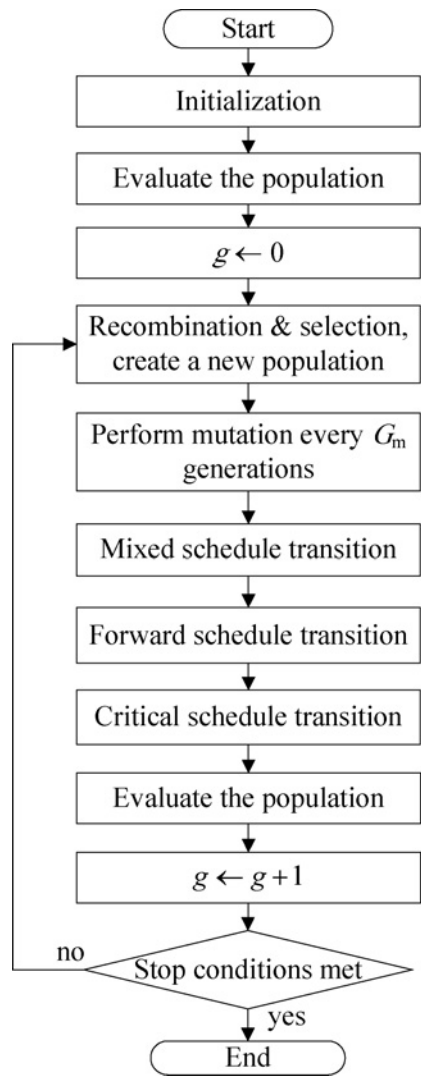


Fig. 4. Flowchart of the proposed STHGA, where g represents the generation index.

The indexes of fields have also been exemplified in Fig. 3. The grids in the same field are covered by the same set of sensors. The higher the resolution of grids is, the higher the accuracy is for representing fields in the target area. In order to approximate fields better, the grid size is selected based on the sensors' sensing ranges. After determining the coverage percentage of each field, the fields can be regarded as targets [35] and can be used for computing the coverage percentage of sensors instead of using grids.

III. Proposed Schedule Transition Hybrid Genetic Algorithm

This paper proposes a hybrid genetic algorithm with schedule transition operations for maximizing the lifetime of a sensor network. In this section, the forward encoding scheme is firstly introduced to the representation of chromosomes. Then the implementation of the proposed STHGA is presented step by step, including the initialization, the evaluation of the population, the genetic operations, and the schedule transition

TABLE I
NOTATIONS

Symbol	Descriptions
$\boldsymbol{S}$	Set of sensors deployed in the target area
s_i	Sensor i
N	Total number of sensors
L	Length of the target area
W	Width of the target area
$\boldsymbol{S}_i$	Cover set to the target area
T	Maximum number of disjoint complete cover sets
r_i	Sensing range of s_i
$\tilde{T}$	Upper limit of T
$\boldsymbol{F}_j$	Set of sensors that cover the field j
n_{F}	Number of fields formed by all of the sensors
M	Minimum number of sensors for complete coverage to an area
R	Sensing range
η	Redundancy rate
d	Size of the grid for discritizing the target area
m	Number of chromosomes in the population
C_i	Chromosome i
c_i	Maximum number of disjoint complete cover sets of chromosome C_i
g_{ij}	jth gene of chromosome C_i
$g_{i_{\max}}$	Maximum gene value in chromosome C_i
ω_1	Predefined weight for the maximum number of complete coverage
ω_2	Predefined weight for the coverage percentage
p_{c_i+1}	Coverage percentage of the (c_i+1)th cover set
G_m	Generation interval for performing mutation
μ	Mutation rate
K_1	Number of genes selected in the initialization and in the forward schedule transition operation
K_2	Number of genes selected in the mixed schedule transition operation

operations. A complete flowchart of the proposed STHGA is shown in Fig. 4. Table I tabulates the notations used in this paper.

A. Representation of Chromosomes

Each gene in the chromosome is mapped to a sensor and the gene value indicates the sensor's scheduling number for activation. Each chromosome C_i in the population is represented as

$$C_i = (g_{i1}, g_{i2}, ..., g_{iN}) \tag{5}$$

where $g_{ij} \in \{1, 2, ..., T+1\}$ represents sensor S_j's scheduling number, $j = 1, 2, \ldots, N$, N is the total number of sensors in the target area, $i = 1, 2, ..., m$, and m is the number of chromosomes in the population. The sensors with the same scheduling number form a disjoint cover set.

The forward encoding scheme proposed in this paper requires that the sensors with the same scheduling number, which is smaller than $g_{i_{\max}} = \max(g_{i1}, g_{i2}, ..., g_{iN})$, form complete coverage to the target area, respectively. That is, all the cover sets with the scheduling number from 1 to $g_{i_{\max}} - 1$ (if $g_{i_{\max}} > 1$) are disjoint complete cover sets to the target area. Whether the $g_{i_{\max}}$th schedule forms complete coverage depends on its coverage percentage. If the sensors in the $g_{i_{\max}}$ th schedule cannot completely cover the target area, the number of disjoint complete cover sets of chromosome C_i is

$$c_i = g_{i_{\max}} - 1. \tag{6}$$

Otherwise, the number of disjoint complete cover sets of chromosome C_i is

$$c_i = g_{i_{\max}}. \tag{7}$$

For example, suppose a chromosome is $C_1 = (1, 1, 2, 2, 1, 3, 2, 1)$. It means that there are eight sensors in the target area and set 1 and set 2 are complete cover sets. If the sensors with the gene value 1 are scheduled to be activated at first, the other sensors will keep in a sleep mode until the second set of sensors is activated.

Although the chromosome in the traditional and the proposed encoding schemes is both a series of numbers, the meaning behind is different. For example, the representation of chromosomes in GAMDSC [48] uses integers to represent the sets that the sensors joined, but it does not use the forward encoding scheme. The main difference between the encoding scheme of our method and the one in [48] is that any sensors with the same scheduling number in [48] do not guarantee forming complete coverage to the target area. Moreover, according to [48], there may be no complete cover set represented by the chromosomes, even though the sensors have been assigned with different numbers, resulting in an invalid solution.

The forward encoding scheme not only indicates the number of disjoint complete cover sets in a chromosome, but also provides guidance for fulfilling the $(c_i + 1)$th schedule of sensors into a complete cover set. Such an encoding scheme has a proactive effect in search of better chromosomes, because at most one set of sensors is not a complete cover set and the other sensors in the complete cover sets can be adjusted to achieve a higher coverage percentage for the incomplete cover set.

The forward encoding scheme actually exerts a restriction on chromosomes so that they can reflect the structural features of solutions. The set covers problem considered in this paper is to identify the maximum number of subsets that can completely cover the target area, forming a pattern in the solution structure. Therefore, the forward encoding scheme is used to represent the pattern of the sensor subsets in chromosomes, with each subset being able to completely cover the target area. The forward encoding scheme is especially suitable for the problems that have the feature of regular patterns in the solution. Encoding the structural features of solutions in chromosomes can accelerate the optimization speed.

The operations in the proposed algorithm are designed based on the forward encoding scheme so that the new chromosomes always comply with the scheme. The effectiveness of the encoding scheme and the operations in STHGA will be tested via a series of experiments in this paper.

B. Initialization

Initially, a population with m chromosomes is created. Based on the forward encoding scheme, the scheduling numbers of all genes are firstly assigned as 1, i.e., $C_i = (1, 1, ..., 1)$,

$i = 1, 2, ..., m$, meaning that all of the sensors are activated. If the schedule does not form complete coverage to the target area, the deployment of sensors fails and the proposed algorithm will not be started.

In a successful sensor deployment, redundant sensors can be turned to sleep without influencing the coverage percentage. For every chromosome, K_1 genes are randomly chosen, where K_1 is a predefined parameter. If the selected genes are redundant sensors for the initial schedule, the values of the selected genes add 1, which means that the selected sensors will be activated in the next schedule. After resetting the genes for each chromosome, the initial population $C_1, C_2, ..., C_m$ is generated.

C. Evaluation of the Population

The fitness function of a chromosome C_i in the population is defined as

$$f_i = \omega_1 c_i + \omega_2 p_{c_i+1} \tag{8}$$

where c_i ($c_i \geq 1$) is the number of disjoint complete cover sets $\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_{c_i}$ and p_{c_i+1} ($p_{c_i+1} \in (0, 1)$) is the coverage percentage of the $(c_i + 1)$th cover set, which is an incomplete cover set, $i = 1, 2, ..., m$. The parameters ω_1 and ω_2 are the predefined weights for c_i and p_{c_i+1}. The larger the value of c_i is, the higher the fitness of chromosome C_i is. If the values of c_i and c_j for two chromosomes C_i and C_j are equal, the chromosome with a larger coverage percentage in the incomplete cover set is the better one. Because the values of c_i and p_{c_i+1} are already sufficient for distinguishing the two parts, the values of ω_1 and ω_2 are both set as 1 in this paper.

D. Genetic Operations

1) *Recombination and Selection:* Different from the implementation in classical GAs, the recombination and selection operations in STHGA are combined for taking advantage of the interactions of chromosomes to generate a better population. The process is described as follows.

Randomly select two chromosomes C_i and C_j from the population $C_1, C_2, ..., C_m$. Then select each gene with equal probability from the two chromosomes and recombine the genes to form a new offspring C_k as

$$g_{kl} = \begin{cases} g_{il}, & \text{if } q_0 < 0.5 \\ g_{jl}, & \text{if } q_0 \geq 0.5 \end{cases} \tag{9}$$

where q_0 is a uniform random number in $[0, 1)$, $k = m+1, m+2, ..., 2m$ is the index of the newly generated chromosome, $l = 1, 2, ..., N$, $i, j \in \{1, 2, ..., m\}$. The recombination technique is actually a kind of uniform crossover and it generates a new offspring for every two parents. Using the uniform crossover can combine the genes from two chromosomes in a more uniform way. An example for generating a new offspring is illustrated in Fig. 5. The selected genes are marked gray in the figure.

After recombination, the fitness of the new offspring C_k is then evaluated according to (8) and compared with its parents. Only the offspring that has no worse fitness than its parents are selected in the new population. Otherwise, the offspring C_k is replaced by the better parental chromosome.

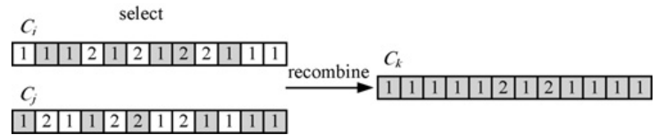


Fig. 5. Randomly choose two chromosomes C_i and C_j and select genes to recombine a new chromosome C_k.

```
/* generate m new chromosomes */
for k := m+1 to 2m do
    i := rand(1, m);
    do j := rand(1, m); while j=i;
/* recombine genes to form a new offspring */
    for l := 1 to N do
        q0 := rand01( );
        if q0 < 0.5
            gkl := gil;
        else
            gkl := gjl;
        end if
    end for
/* select the best chromosome to the new population */
    evaluate Ck = (gk1, gk2,…, gkN) according to (8);
    if fk < fi
        Ck ← Ci; fk := fi;
    end if
    if fk < fj
        Ck ← Cj; fk := fj;
    end if
end for
```

Fig. 6. Pseudocode of the proposed combination and selection, where rand(1, m) is a function returning a uniform random integer number in $\{1, 2, \ldots, m\}$, rand01( ) is a function returning a uniform random float-point number in [0, 1).

The above recombination and selection process repeats for m times, and thus a new population $C_{m+1}, C_{m+2}, ..., C_{2m}$ is generated. Because each chromosome in the new population is no worse than its parents and the largest gene value is no larger than that of its parents, the chromosomes in the new population still satisfy the forward encoding scheme. The pseudocode for the selection and combination is depicted in Fig. 6.

2) *Reverse Mutation:* The reverse mutation, as its name indicates, performs reverse operations by changing the sensors in the incomplete cover set back to one of the complete cover sets. Note that the forward encoding scheme confines that only the set with the largest scheduling number can be an incomplete cover set. The other sets are disjoint complete cover sets to the target area. Because the reverse mutation may reduce the fitness of chromosomes (that is, reduce the coverage percentage of the incomplete cover set but not influence the already complete cover sets), it is carried out only once every G_{m} generations.

For the best chromosome C_i in the current population, if its $g_{i_{\max}}$ th set is an incomplete cover set (i.e., $c_i = g_{i_{\max}} - 1$), every gene g_{ij} that corresponds to the incomplete cover set will be mutated according to a mutation rate μ. If the $g_{i_{\max}}$ th set is already a complete cover set, the mutation operation is not performed. For every gene g_{ij} in the incomplete cover

```
/* perform mutation every G_m generations */
if g mod G_m = 1
    b = 0;
/* find out the iteration best chromosome in the m chromosomes */
    for i := m+1 to 2m
        if f_i > b
            b := f_i;
        end if
    end for
    for i := m+1 to 2m
/* perform mutation to the iteration best chromosome that has an incomplete cover set */
        if f_i = b and c_i = g_imax − 1
            for j := 1 to N
                if g_ij = g_imax and rand01( ) < μ
                    g_ij := ⌊c_i rand01()⌋+1;
                end if
            end for
            evaluate C_i = (g_i1, g_i2,…, g_iN) according to (8);
        end if
    end for
end if
```

Fig. 7. Pseudocode of the proposed reverse mutation, where g represents the generation index, rand01( ) is a function returning a uniform random float-point number in [0, 1).

set (i.e., $g_{ij} = g_{i_{\max}}$), if $q_1 < \mu$, the mutated gene value is generated as

$$g_{ij} = \lfloor c_i q_2 \rfloor + 1 \tag{10}$$

where q_1 and q_2 are uniform random numbers in [0, 1), the $\lfloor c_i q_2 \rfloor$ represents the largest integer that is less than or equal to $c_i q_2$.

The reverse mutation is an important operation for maintaining diversity in the population so as to avoid trapping in local optima. It eliminates search bias by reducing the fitness of the current scheduling scheme. The pseudocode of the reverse mutation is shown in Fig. 7.

E. Schedule Transition Operations

The schedule transition operations, including the mixed schedule transition, the forward schedule transition, and the critical schedule transition, utilize the redundancy information among the scheduled sensors for each chromosome. The three schedule transition operations have their own characteristics and cooperate to search for a better scheduling scheme.

1) *Mixed Schedule Transition:* The mixed schedule transition operation schedules redundant sensors from a cover set to the other cover set. For every chromosome C_i, $i = m + 1, m+2, ..., 2m$, each schedule from 1 to c_i forms complete coverage to the target area. Suppose the disjoint complete cover sets of chromosome C_i are denoted as $\boldsymbol{S}_{i1}, \boldsymbol{S}_{i2}, ..., \boldsymbol{S}_{i,c_i}$, and the incomplete cover set is $\boldsymbol{S}_{i,c_i+1}$. Possible transition directions of the mixed schedule transition operation are illustrated in Fig. 8. Redundant sensors can be changed between an incomplete coverage schedule and a complete coverage schedule, or between the complete coverage schedules. This operation helps reschedule sensors among different cover sets without influencing the complete cover sets. For every chromosome

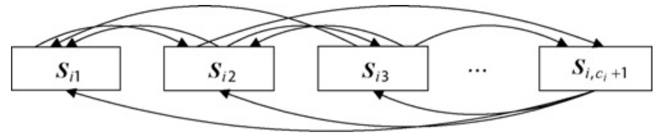


Fig. 8. Illustration of the transition directions of the mixed schedule transition operation for chromosome C_i.

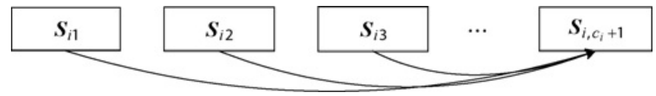


Fig. 9. Illustration of the transition directions of the forward schedule transition operation for chromosome C_i.

C_i, $i = m+1, m+2, ..., 2m$, the process of the mixed schedule transition is described as follows.

Firstly, randomly select a sensor. If the sensor is redundant to the corresponding cover set, we change the schedule of the sensor to a randomly selected complete coverage schedule. If the selected scheduling number is the same as that of the sensor, the sensor is scheduled to the incomplete cover set, making it different from the sensor's original schedule. The above process is repeated for K_2 times, where K_2 is a predefined parameter.

The difference between the reverse mutation operation and the mixed schedule transition operation is obvious. In mutation, sensors are selected by a probability, regardless whether it is redundant to the incomplete cover set. Therefore, after mutation, the fitness of the chromosome may be reduced. However, in the mixed schedule transition operation, only redundant sensors can be rescheduled and the fitness of the chromosome is not reduced.

2) *Forward Schedule Transition:* The forward schedule transition operation is used for enhancing the coverage percentage of the incomplete cover set. After rescheduling sensors among different cover sets by the mixed schedule transition, the forward schedule transition operation schedules some redundant sensors from complete cover sets to the incomplete cover set. For every chromosome C_i, $i = m + 1, m + 2, ..., 2m$, K_1 genes are selected. If the corresponding sensors are redundant, the selected sensors are rescheduled into the set $\boldsymbol{S}_{i,c_i+1}$. Fig. 9 presents the possible transition directions of the forward schedule transition.

Note that when $c_i = g_{i_{\max}}$, initially the set $\boldsymbol{S}_{i,c_i+1} = \phi$. By transferring redundant sensors to the set $\boldsymbol{S}_{i,c_i+1}$, the coverage percentage of $\boldsymbol{S}_{i,c_i+1}$ is increased so that the fitness of the chromosome is enhanced.

3) *Critical Schedule Transition:* Sensors are not uniformly distributed in the target area by a random sensor deployment. Some fields are covered densely, while some fields are covered sparsely. Sparsely covered fields restrict the maximum number of disjoint complete cover sets. In order to accelerate the algorithm, the fields that are covered by the minimum number of sensors, termed *critical fields*, are checked.

The critical schedule transition operation is used for facilitating the critical fields to be covered by at least one sensor in the incomplete cover set. Once critical fields are covered, there

TABLE II
SUMMARY OF THE FUNCTIONS OF OPERATIONS IN STHGA

	Operations	Functions
Genetic Operations	Recombination and Selection	Use interactions between chromosomes to improve solution quality
	Reverse Mutation	Eliminate search bias by reducing the fitness of the current scheduling scheme
Schedule Transition Operations	Mixed Schedule Transition	Schedule redundant sensors from one cover set to the other cover set
	Forward Schedule Transition	Schedule redundant sensors to enhance the coverage of the incomplete cover set
	Critical Schedule Transition	Ensure that critical fields can be covered in the incomplete cover set

are higher chances for scheduling sensors to the incomplete sensor set to form complete coverage.

The fields covered by only $\tilde{T}$ sensors are critical fields, which can be determined in the initialization step. Every chromosome C_i, $i = m+1, m+2, ..., 2m$, performs the critical schedule transition as follows. For every critical field, if it is not covered in the incomplete cover set, we randomly choose a redundant sensor that covers the critical field from the complete cover sets and reschedule the sensor to the incomplete cover set. After performing the critical schedule transition, critical fields are covered in the incomplete coverage schedule if corresponding redundant sensors are found.

F. Summary Discussions on STHGA

Table II summarizes the functions of the operations used in the proposed STHGA. The genetic operations and schedule transition operations cooperate to search for the best scheduling scheme of sensors. The generated chromosomes by the operations comply with the requirements of the forward encoding scheme. The effectiveness of the operations in STHGA will be analyzed in the next section via a series of experiments.

After performing the genetic operations and the schedule transition operations in every generation, the new population is evaluated according to the fitness function (8). The best-so-far chromosome is thus updated and the chromosomes in $C_{m+1}, C_{m+2}, ..., C_{2m}$ replace the chromosomes in $C_1, C_2, ..., C_m$ respectively. When $\tilde{T}$ disjoint complete cover sets are found or the predefined maximum number of fitness function evaluations is reached, the proposed STHGA terminates. If the termination condition is not satisfied, a new population will be generated in the next generation.

The number of fitness function evaluations in each generation of the proposed STHGA is analyzed as follows. The evaluation number used in the recombination and selection operation is m, because m new chromosomes are generated. Since the reverse mutation is performed every G_m generations and the worst case is to mutate all the chromosomes in the population, the average evaluation number for every generation is thus no larger than m/G_m. After the three schedule transitions, the population is evaluated again, so that m function evaluations are needed. In summary, the number of fitness function evaluations in every generation is approximately $(2 + 1/G_m)m$.

IV. Experimental Study and Discussions

In this section, the proposed STHGA is tested with different sensor deployments for point-coverage and area-coverage problems. The performance of STHGA will be compared with the state-of-the-art algorithms, i.e., MCMCC [35] and GAMDSC [48]. Analysis and discussions on the operations of the proposed STHGA are also presented.

If not specially stated, all experiments for STHGA use the same parameters settings as the population size $m = 3$, the interval for performing mutation $G_m = 100$, the mutation rate $\mu = 0.5$, and the parameters $K_1 = K_2 = 5$. These parameter values are set empirically and their influences to the performance of STHGA will be analyzed. Parameter settings of MCMCC and GAMDSC can be referred in [35] and [48]. For STHGA and GAMDSC, each case is tested 100 times independently. The sensors are deployed in a 50×50 rectangle area and the coordinates of sensors' locations are randomly generated as float-point values in [0, 50]. All cases are run by a computer with a Pentium IV 2.8 GHz CPU.

A. Experiments on Point-Coverage Problems

As has been stated in Section I, GAMDSC is proposed for solving point-coverage problems, whereas MCMCC and STHGA can be used for both point-coverage and area-coverage problems. Seven point-coverage cases with different numbers N of sensors are tested. The number of targets is fixed as 10 and the sensing range R is 22 for all the sensors. Using the same stopping criterion as GAMDSC in [48], the maximum number of fitness function evaluations for both GAMDSC and STHGA is 20 100. If the number of disjoint complete cover sets reaches $\tilde{T}$, the algorithm also stops.

Table III tabulates the results computed by STHGA, GAMDSC, and MCMCC. The $\tilde{T}$ in the table represents the upper limit of the maximum number of disjoint complete cover sets. Because MCMCC is a deterministic algorithm, it is run only once and the result and the time used for computation are recorded. From the table, MCMCC obtains results that are equal to $\tilde{T}$ in four out of the seven cases. In contrast, the proposed STHGA achieves results that are equal to $\tilde{T}$ in all of the seven cases. The time used by STHGA is much shorter than MCMCC in most of the cases except for Cases 5 and 6. However, MCMCC cannot achieve the optima of the two cases but STHGA can by using a slightly longer time. In comparison with GAMDSC, the advantage of STHGA is obvious. STHGA can find the optima in all of the 100 independent runs, so that only the mean results are tabulated. However, GAMDSC cannot always obtain the optima within the predefined maximum number of function evaluations except for Case 3, which is the case with the smallest $\tilde{T}$ value. The best and mean results of GAMDSC are presented in the table, plus the average number of function evaluations (*avgE*)

TABLE III

TEST RESULTS FOR POINT-COVERAGE CASES WITH DIFFERENT NUMBERS N OF SENSORS

Cases			STHGA				GAMDSC					MCMCC	
No.	N	$\tilde{T}$	Mean	avgE	ok%	Time (ms)	Best	Mean	avgE	ok%	Time (ms)	Result	Time (ms)
1	90	30	**30**	596	100	17	**30**	28.63	19 822	8	965	29	31
2	100	23	**23**	214	100	5	**23**	22.83	11 939	84	569	**23**	31
3	110	21	**21**	156	100	4	**21**	**21**	6141	100	303	**21**	31
4	120	35	**35**	422	100	15	**35**	33.50	19 899	5	1249	**35**	62
5	130	41	**41**	1856	100	84	**41**	40.99	10 297	99	741	40	78
6	140	44	**44**	3568	100	172	43	40.72	20 100	0	1550	43	93
7	150	42	**42**	532	100	25	**42**	40.82	19 132	24	1517	**42**	109

The number of targets is 10 and the sensing range R is 22 for all the sensors. The best results among the three algorithms for each case are bold.

TABLE IV

TEST RESULTS FOR AREA-COVERAGE CASES WITH DIFFERENT NUMBERS N OF SENSORS AND SENSING RANGES R

Cases					STHGA				GAMDSC					MCMCC	
No.	N	R	n_F	$\tilde{T}$	Mean	avgE	ok%	Time (ms)	Best	Mean	avgE	ok%	Time (ms)	Result	Time (ms)
1	100	20	385	7	**7**	93	100	33	**7**	**7**	874	100	126	**7**	1438
2	300	15	673	16	**16**	509	100	400	15	13.19	20 100	0	9713	**16**	33 922
3	300	20	400	32	**32**	713	100	468	29	26.06	20 100	0	10 080	**32**	44 047
4	400	10	1556	9	**9**	598	100	797	8	6.24	20 100	0	13 764	**9**	54 844
5	400	15	676	23	**23**	800	100	767	20	16.90	20 100	0	13 268	**23**	81 766
6	500	8	2400	7	**7**	878	100	1588	6	4.04	20 100	0	17 413	**7**	76 296
7	500	10	1586	15	**15**	9223	100	11 386	8	5.81	20 100	0	18 527	**15**	124 922
8	1000	5	6076	5	**5**	890	100	4534	3	0.89	20 100	0	38 105	**5**	263 469
9	1000	8	2498	17	**17**	1925	100	5901	6	3.19	20 100	0	37 830	**17**	683 890

The best results among the three algorithms for each case are bold.

and the average time in microsecond (ms) used for obtaining the best result in each run, and the successful percentage (ok%). STHGA outperforms GAMDSC both in the solution quality and the optimization speed.

Using the point-coverage Case 1 as an example, we analyze how STHGA performs better than GAMDSC. Fig. 10 shows the average optimization curves of STHGA and GAMDSC when solving Case 1 within the maximum function evaluation number. It can be seen that STHGA finds high-quality results much faster than GAMDSC. Note that GAMDSC does not use the proposed forward encoding scheme to chromosomes so that the initialization of STHGA and GAMDSC is different. In STHGA, all sensors are initially in the same complete cover set and redundant sensors are then scheduled to form a new cover set. So the initial number of complete cover sets is small. In GAMDSC, each sensor is initially assigned to a random cover set. From the inner figure in Fig. 10, the initial number of complete cover sets found by GAMDSC is 9, which is bigger than that of STHGA. However, STHGA soon catches up and then surpasses GAMDSC because the incomplete set is continuously completed through the operations in STHGA. The above results demonstrate that STHGA is very efficient.

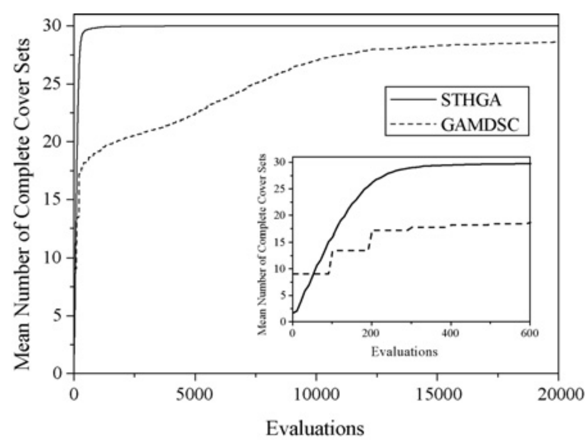


Fig. 10. Average optimization curves of STHGA and GAMDSC when solving the point-coverage Case 1 (N = 90 and the maximum number of disjoint cover sets is 30). The inner figure shows more details within the first 600 evaluations.

B. Experiments on Area-Coverage Problems

The characteristics of the nine area-coverage cases with different numbers N of sensors and sensing ranges R are presented in Table IV. In the area-coverage cases, the grid size d is set as $L/\lfloor L/(R/8)\rfloor$ so that the resolution of grids is big enough. The value of n_F shows that the number of target fields to be covered in the area-coverage problems is much larger than the number of targets in the point-coverage problems in the previous subsection.

The results in Table IV show that GAMDSC is not suitable for solving area-coverage problems, because it can only find the optimum of the smallest Case 1 within the predefined maximum number of function evaluations. In contrast, STHGA and MCMCC can find the optima of the nine area-coverage cases successfully. By comparing the time used by STHGA and MCMCC for achieving the optimal solution, the computation speed of STHGA is much faster than that of MCMCC. Take Case 5 as an example, the average time used by STHGA is

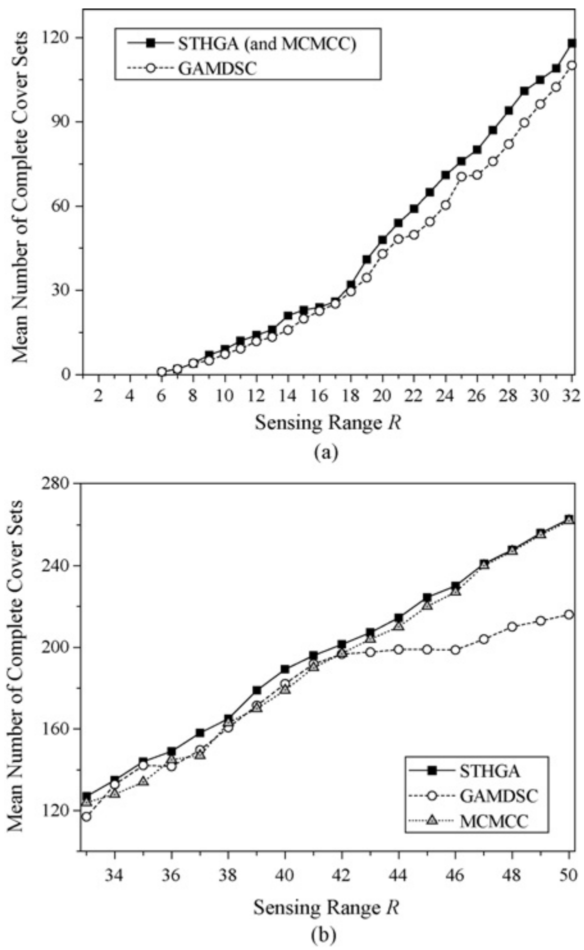


Fig. 11. Mean number of disjoint complete cover sets with R increasing from 1 to 50 for the test case $N = 400$. The optimization results of STHGA and GAMDSC are obtained by calculating the average number of disjoint complete cover sets in 100 independent runs. The predefined maximum number of function evaluations for both STHGA and GAMDSC is 150 000. (a) Illustration of the sensing ranges from 1 to 32. The results obtained by the STHGA and the MCMCC are identical, so their curves are drawn together. (b) Illustration of the sensing ranges from 33 to 50.

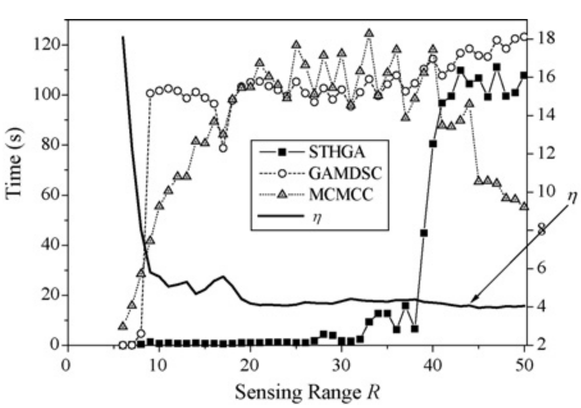


Fig. 12. Comparison of the time used for obtaining the optimization results by STHGA, GAMDSC, and MCMCC when R increases from 1 to 50 for the test case $N = 400$. The left axis shows the time in seconds, whereas the right axis shows the value of η for the cases with different sensing ranges R. The curve of η is marked by an arrow line in the figure.

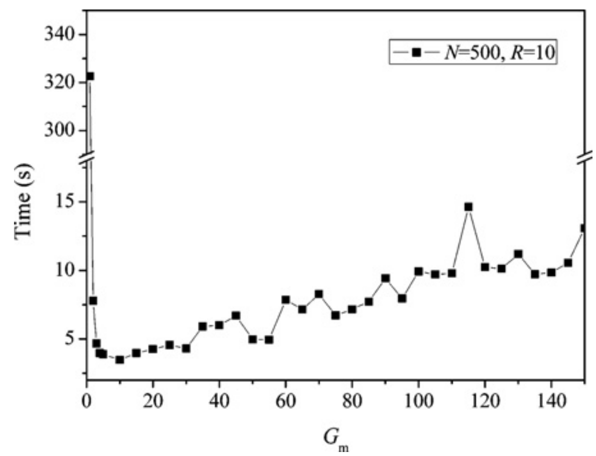


Fig. 13. Comparison of the time used by STHGA with different interval values G_m (from 1 to 150) for performing mutation on the test case $N = 500$ and $R = 10$. The algorithm is run for 100 times independently with every G_m value.

767 ms, which is shorter than 1 s, whereas MCMCC needs 81 766 ms, which is approximately 82 s. The reason why MCMCC uses a longer time than STHGA is because of its mechanism in building cover sets. MCMCC builds each cover set by successively adding a sensor to the set until all fields have been covered. Before each step for choosing a sensor, a new critical field that is formed by the remaining sensors to the uncovered fields must be determined. Then each of the sensors that cover the new critical field is evaluated by an objective function. The sensor that has the maximum function value is selected into the cover set. The worst runtime of MCMCC is $O(N^2)$ and it needs to calculate the values of its objective function several times for choosing an unselected sensor to a new cover set. Because the update of the critical field and the calculation of the objective function are time-consuming, MCMCC generally uses a long time before termination.

When the number and positions of sensors are fixed in the target area, the sensing ranges of sensors influence the lifetime of the WSN. Without considering the energy consumption by different sensing ranges, a sensor deployment with 400 sensors using sensing ranges from 1 to 50 is checked. Fig. 11 compares the results computed by STHGA, GAMDSC, and MCMCC. According to Fig. 11(a), when the sensing range is smaller than 6, the sensor deployment is a failure for no complete coverage to the target area. When the range R increases, the maximum number of complete cover sets also increases. The numbers of complete cover sets that are achieved by the three algorithms are the same when $R = 6$ to 8. However, for the larger sensing ranges ($R > 8$), the performance of GAMDSC becomes worse than both STHGA and MCMCC. Fig. 11(a) shows that the results obtained by STHGA and MCMCC are identical when $R = 6$ to 32. When $R = 33$ to 50 [shown in Fig. 11(b)], STHGA performs better than MCMCC.

Fig. 12 shows the average time used by the three algorithms for the above experiment. The time used by STHGA is always shorter than GAMDSC. When the sensing range $R \leq 40$, the time used by STHGA is much shorter than MCMCC. The computation time of STHGA increases rapidly after $R > 33$, during which MCMCC fails to find the optimal solution. Even though MCMCC has terminated with a suboptimal solution, STHGA continues to search for better solutions. The results show that STHGA is more robust to different sensing applications than the other algorithms.

C. Further Analysis on the Performance of STHGA

Note that the proposed STHGA is a hybrid of a GA and some schedule transition operations. Therefore, in this subsection, we firstly analyze the influence of different parameter settings in the genetic operations on the performance of STHGA and then evaluate the effects of the schedule transition operations through experiments. We also discuss the performance of STHGA in solving problems with different redundancy in cover sets.

1) *Influence of Parameter Settings in Genetic Operations:* The parameters in the genetic operations of STHGA include the population size m, the mutation rate μ, and the interval G_m for performing mutation. Table V tabulates the results of STHGA with different numbers of population size m when solving area-coverage problems. The results show that using a big population size slows down the convergence speed of the proposed algorithm by taking more evaluation steps (*avgE*). The best performance of STHGA is achieved when the population size $m = 3$. By hybridizing with other effective search methods in a GA, the results show that a small population is enough for evolving better chromosomes in STHGA.

The mutation operation is used to avoid trapping in local optima. When a mutation procedure starts, the mutation rate determines the probability of mutating the gene in the incomplete cover set into another randomly selected complete cover set. Table VI compares the results of STHGA when using different mutation rates μ for solving area-coverage problems. When $\mu = 0.5$, the genes have 50% probability to be mutated. No gene is mutated when $\mu = 0$, whereas all genes in the incomplete cover set are mutated when $\mu = 1$. The performances of STHGA using the three mutation rates are similar except for Case 7. Without the mutation operation ($\mu = 0$), the algorithm can only achieve the optimum with 21% successful percentage for Case 7. Meanwhile, using the mutation operation can obtain 100% success. Therefore, the mutation operation is useful for STHGA.

Different values of the interval G_m for performing mutation have also been tested. Fig. 13 illustrates the comparison results by testing different interval values (G_m) from 1 to 150 on the test Case 7 with $N = 500$ and $R = 10$. Although STHGA can find the optimal solution using all of the tested G_m values in the test case, the time used by the algorithm is the longest when $G_m = 1$. It means that if mutation is performed in every generation, the mutation operation is so destructive that the algorithm takes much more steps for finding the optimal solution. It is recommended to use a relatively large G_m so as to avoid being trapped and lowering the performance of the algorithm.

2) *Effects of Schedule Transition Operations in STHGA:* Among the three schedule transition operations, the forward schedule transition is the major operation for enhancing the incomplete cover set. The mixed and critical schedule transition operations facilitate searching for better solutions. Table VII compares the results of STHGA with default settings, without the mixed schedule transition, and without critical schedule transition. Although the three algorithms can find the optimal solutions for the test cases, the average numbers of function evaluations used by the latter two algorithms are larger than those used by STHGA with default settings. It shows that the mixed and critical schedule transition operations can accelerate searching for the optimal solution.

Different values of K_1 and K_2 have been checked for evaluating the effectiveness of the forward and mixed schedule transition operations, respectively. Fig. 14 shows the average time used by STHGA for finding the optimal solutions with $K_1 = 1$ to 100. When the value of K_1 increases from 1, the time decreases for achieving the optimal solution. As the value of K_1 continues to increase, the time gradually becomes larger instead. The tolerance of STHGA to a large K_1 is higher for the cases with a larger number of sensors. In the cases with a large number of sensors, more sensors are redundant and can be scheduled to the incomplete cover set by the forward schedule transition operation. However, if the value of K_1 exceeds a threshold, the performance of the algorithm deteriorates because of the lack of redundant sensors in complete cover sets. The best parameter setting of K_1 is in the range of [20, 40] for most cases. Note that the default setting of K_1 in this paper is $K_1 = 5$, which is not in the best value range. It means that although the performance of STHGA using the default parameter settings has outperformed other state-of-the-art algorithms, the performance of STHGA for the test cases in this paper still can be further enhanced.

Fig. 15 compares the time used by STHGA with different values of K_2. All the curves in the figure show a significant turning point near the value $K_2 = 5$. The larger the number of sensors in the problem, the best value of K_2 is slightly bigger. For example, the turning points for the cases in the figure appear when $K_2 = 3, 5, 6, 7$, respectively, for the cases with $N = 100, 300, 400$, and 1000. The existence of a turning point shows that an appropriate scheduling of sensors to the other cover sets is necessary but more mixed scheduling may waste time.

3) *Influence of Redundancy in Set Covers Problems:* Besides the number and positions of the sensors and their sensing range, the redundancy rate η is also an important factor for measuring the optimization difficulty of a set covers problem. Note that the three schedule transition operations are designed for utilizing the redundancy information among sensors. When the value of η reduces, redundant sensors are fewer and the proposed schedule transition operations are harder to take effect. The values of η for the $N = 400$ cases with different sensing ranges have been illustrated in Fig. 12. As the sensing range increases, the number of disjoint complete cover sets becomes bigger and the value of η shows a decreasing trend. When the value of η suddenly increases (e.g., when $R = 13$ or 17), the time used by the three algorithms also reduces. It can be observed from the figure that the problem with a larger value of η is easier to be solved than

TABLE V

TEST RESULTS OF STHGA WITH DIFFERENT NUMBERS OF POPULATION SIZES $m = 3, 6$, OR 10

Cases					$m = 3$			$m = 6$			$m = 10$		
No.	N	R	n_F	$\tilde{T}$	Mean	avgE	ok%	Mean	avgE	ok%	Mean	avgE	ok%
1	100	20	385	7	7	93	100	7	165	100	7	256	100
2	300	15	673	16	16	509	100	16	851	100	16	1304	100
3	300	20	400	32	32	713	100	32	1167	100	32	1791	100
4	400	10	1556	9	9	598	100	9	947	100	9	1393	100
5	400	15	676	23	23	800	100	23	1282	100	23	1964	100
6	500	8	2400	7	7	878	100	7	1269	100	7	1775	100
7	500	10	1586	15	15	9223	100	15	13 395	100	15	14 161	100
8	1000	5	6076	5	5	890	100	5	1426	100	5	2153	100
9	1000	8	2498	17	17	1925	100	17	3086	100	17	4585	100

TABLE VI

TEST RESULTS OF STHGA WITH DIFFERENT MUTATION RATES $\mu = 0, 0.5$, OR 1

Cases					$\mu = 0.5$			$\mu = 0$			$\mu = 1$		
No.	N	R	n_F	$\tilde{T}$	Mean	avgE	ok%	Mean	avgE	ok%	Mean	avgE	ok%
1	100	20	385	7	7	93	100	7	95	100	7	96	100
2	300	15	673	16	16	509	100	16	531	100	16	513	100
3	300	20	400	32	32	713	100	32	691	100	32	710	100
4	400	10	1556	9	9	598	100	9	592	100	9	600	100
5	400	15	676	23	23	800	100	23	788	100	23	784	100
6	500	8	2400	7	7	878	100	7	884	100	7	797	100
7	500	10	1586	15	15	9223	100	14.21	17 306	21	15	10 214	100
8	1000	5	6076	5	5	890	100	5	858	100	5	943	100
9	1000	8	2498	17	17	1925	100	17	1866	100	17	1933	100

TABLE VII

TEST RESULTS OF STHGA WITH DEFAULT SETTINGS, WITHOUT MIXED SCHEDULE TRANSITION (NO_MIX_SCHEDULE), AND WITHOUT CRITICAL SCHEDULE TRANSITION (NO_CRIT_SCHEDULE)

Cases					STHGA			no_mix_schedule			no_crit_schedule		
No.	N	R	n_F	$\tilde{T}$	Mean	avgE	ok%	Mean	avgE	ok%	Mean	avgE	ok%
1	100	20	385	7	7	93	100	7	298	100	7	142	100
2	300	15	673	16	16	509	100	16	971	100	16	774	100
3	300	20	400	32	32	713	100	32	1318	100	32	934	100
4	400	10	1556	9	9	598	100	9	949	100	9	834	100
5	400	15	676	23	23	800	100	23	1326	100	23	1109	100
6	500	8	2400	7	7	878	100	7	1887	100	7	1890	100
7	500	10	1586	15	15	9223	100	15	16 753	100	15	19 953	100
8	1000	5	6076	5	5	890	100	5	1225	100	5	1062	100
9	1000	8	2498	17	17	1925	100	17	2709	100	17	2489	100

TABLE VIII

TEST RESULTS FOR TWO GROUPS OF AREA-COVERAGE CASES WITH DIFFERENT VALUES OF η

	Test Cases				STHGA			MCMCC	
	N	R	η	$\tilde{T}$	Mean	avgE	Time (ms)	Result	Time (ms)
Group 1	1000	5	6.28	5	**5**	1087	4976	**5**	255 094
	650	5	4.08	5	**5**	1578	3728	**5**	67 390
	564	5	3.54	5	**5**	3214	5774	**5**	41 859
	482	5	3.03	5	**5**	20 379	27 701	4	22 266
	427	5	2.68	5	**5**	556 083	613 942	4	13 797
Group 2	560	8	5.00	9	**9**	1693	2975	**9**	113 453
	372	8	3.32	9	**9**	4399	4454	**9**	32 062
	305	8	2.73	9	**9**	67 634	50 586	7	16 250
	280	8	2.50	9	**9**	394 940	267 337	7	11 937

The best results among the three algorithms for each case are bold.

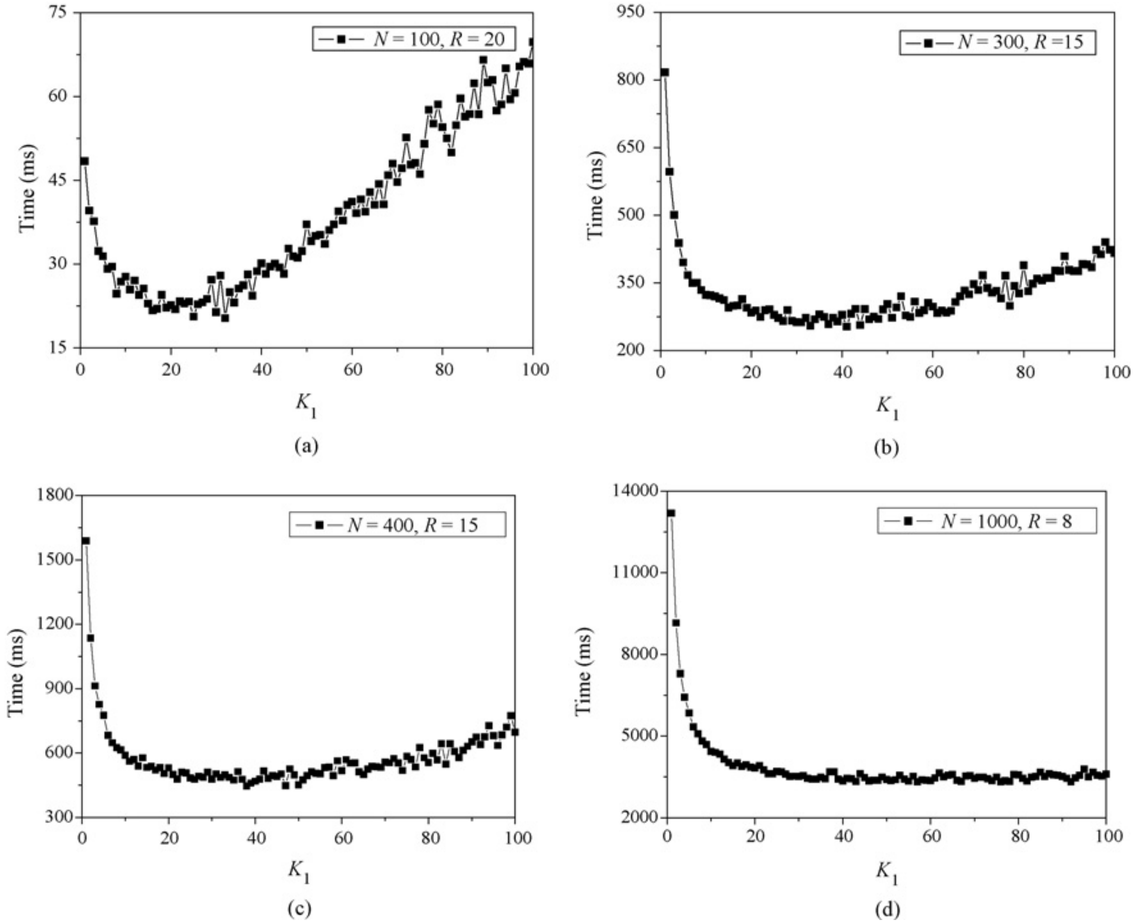


Fig. 14. Comparison of the time used by STHGA when K_1 increases from 1 to 100. The time [in microseconds (ms)] is an average of 100 independent runs for each test case. (a) N = 100, R = 20. (b) N = 300, R = 15. (c) N = 400, R = 15. (d) N = 1000, R = 8.

the one with a smaller η. The above conclusion fits to all the three algorithms.

It can be seen that the value of η for the N = 400 cases is no smaller than 4.0 in Fig. 12. In order to reduce the value of η to smaller values so as to analyze its influence, test cases are generated as follows. Firstly, a random sensor deployment with a large number of sensors is generated. The sensing ranges are set to be small but there must be complete coverage to the target area. When the optimum of the set covers problem is found by STHGA, keep running the algorithm for 200 more generations. Suppose the final best chromosome is C_i. The sensors in the set S_{i,c_i+1} are redundant sensors, which are then removed. Thus a new deployment that has fewer sensors but with the same maximum number of disjoint complete cover sets is generated. Repeat the above process several times and a series of test cases with smaller numbers of sensors and smaller η values are built.

In this experiment, two groups of area-coverage cases are generated from different initial sensor deployments. The characteristics of the generated cases and the η values are listed in Table VIII. As GAMDSC works not as well as the other two algorithms, which have been shown in the previous experiments, only the performances of STHGA and MCMCC are compared. From Table VIII, the proposed STHGA can find the best solutions in all test cases, whereas MCMCC cannot find the best solutions in test cases when the value of η drops below 3.1. For the cases that both algorithms can find the best solutions, STHGA performs better than MCMCC by using a much shorter time.

Fig. 16 illustrates the average optimization results computed by STHGA for solving the first group of area-coverage cases. The maximum number of disjoint complete cover sets is 5 for all the cases with different values of N and η. When the value of η decreases, the number of function evaluations needed for achieving the optimum increases rapidly. In the first 600 evaluations, STHGA finds better results faster when the number N is smaller. As the optimization goes on, STHGA spends more time on the case with a smaller η to find the optimum. A reason for this phenomenon is that when the value of η decreases, redundant sensors are more and more difficult to be selected by the schedule transition operations for constructing a new complete cover set successfully. It should be noted that STHGA uses fewer than 1500 function evaluations to obtain four disjoint complete cover sets in the first group of area-coverage cases. Compared with the results of Cases N = 482 and N = 427 in Table VIII, STHGA still performs much faster than MCMCC for achieving the same sub-optimal solution.

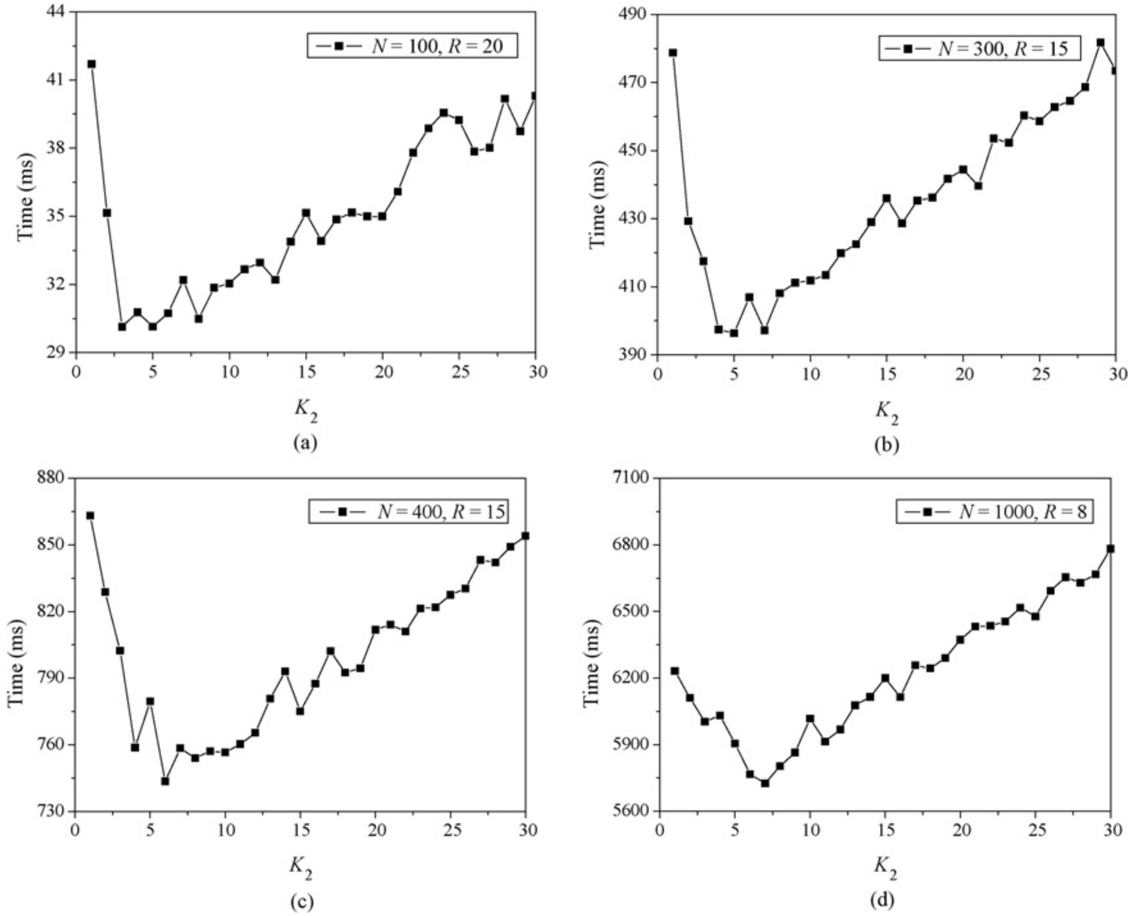


Fig. 15. Comparison of the time used by STHGA when K_2 increases from 1 to 30. The time [in microseconds (ms)] is an average of 100 independent runs for each test case. (a) $N = 100$, $R = 20$. (b) $N = 300$, $R = 15$. (c) $N = 400$, $R = 15$. (d) $N = 1000$, $R = 8$.

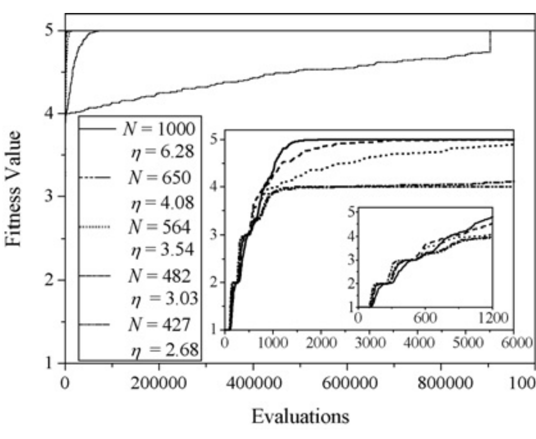


Fig. 16. Average optimization curves of STHGA when solving the first group of area-coverage cases in Table VIII. The cases are generated by removing redundant sensors so that the value of η decreases. The maximum number of disjoint complete cover sets is five for the cases.

V. Conclusion

In this paper, a GA based approach termed STHGA has been proposed to find the maximum number of disjoint complete cover sets of sensors for maximizing the lifetime of a WSN. Its distinct features and importance are concluded as follows.

1) The advantage of STHGA in solving the disjoint set covers problems lies in its adoption of a forward encoding scheme and some well-designed operations. These operations are very suitable for finding the maximum number of disjoint complete cover sets for maximizing the lifetime of WSNs.
2) Encoding the structural features of solutions in chromosomes so that the maximum gene value of each chromosome is increased consistently with the solution quality is the novelty of our encoding scheme of chromosomes. The forward encoding scheme for representing chromosomes not only reflects the best scheduling scheme of sensors that has been found, but also provides guidance for further advancement of chromosomes.
3) Complying with the forward encoding scheme, the genetic operations and schedule transition operations in STHGA cooperate to search for the best scheduling scheme of sensors. In particular, the usage of redundancy information among the scheduled sensors has been shown to be efficient in this paper.
4) STHGA is applicable to both point-coverage and area-coverage problems in WSNs. Even though area-coverage problems are more difficult than point-coverage problems, because area-coverage problems involve a whole area instead of only a few target points in

point-coverage problems, the proposed algorithm can achieve high-quality solutions in a fast optimization speed and outperform the other state-of-the-art algorithms.

Further applications of the proposed algorithm to other similar problems are important parts of our future research work. First, the problems addressed in this paper can be regarded as a type of constrained subset optimization problems, in which the maximum number of subsets that can meet the objective requirements is needed to be identified. In this paper, the objective requirement of each subset is to form complete coverage to the targets. Therefore, the proposed STHGA also has presented a new implementation method of GAs for the problems. On the other hand, we will investigate the application of STHGA by considering the energy consumption of sensors by using different sensing ranges, and find out the best configuration between different working modes and the lifetime of complete cover sets.

References

[1] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[2] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, Jun. 2008.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[4] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensory networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[7] H. Y. Shwe, X.-H. Jiang, and S. Horiguchi, "Energy saving in wireless sensor networks," *J. Commun. Comput.*, vol. 6, no. 5, pp. 20–28, May 2009.

[8] C.-Y. Chang and H.-R. Chang, "Energy-aware node placement, topology control and MAC scheduling for wireless sensor networks," *Comput. Netw.*, vol. 52, no. 11, pp. 2189–2204, Aug. 2008.

[9] H. Leung, S. Chandana, and S. Wei, "Distributed sensing based on intelligent sensor networks," *IEEE Circuits Syst. Mag.*, vol. 8, no. 2, pp. 38–52, Apr.–Jun. 2008.

[10] S. S. Iyengar, H.-C. Wu, N. Balakrishnan, and S. Y. Chang, "Biologically inspired cooperative routing for wireless mobile sensor networks," *IEEE Syst. J.*, vol. 1, no. 1, pp. 29–37, Sep. 2007.

[11] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, "Cross-layer energy and delay optimization in small-scale sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3688–3699, Oct. 2007.

[12] Y. Yu, V. K. Prasanna, and B. Krishnamachari, "Energy minimization for real-time data gathering in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3087–3096, Nov. 2006.

[13] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Comput. Commun.*, vol. 29, no. 4, pp. 413–420, Feb. 2006.

[14] M. Cardei and J. Wu, "Coverage in wireless sensor networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds. Boca Raton, FL: CRC Press, 2004, pp. 432–446.

[15] S. J. Baek, G. de Veciana, and X. Su, "Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1130–1140, Aug. 2004.

[16] D. Tian and N. D. Georganas, "Connectivity maintenance and coverage preservation in wireless sensor networks," *Ad Hoc Netw.*, vol. 3, no. 6, pp. 744–761, Nov. 2005.

[17] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 1, nos. 1–2, pp. 89–124, Mar. 2005.

[18] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, Los Angeles, CA, 2003, pp. 28–39.

[19] F. Y. S. Lin and P. L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 43–45, Jan. 2005.

[20] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Netw. Appl.*, vol. 10, no. 4, pp. 519–528, 2005.

[21] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 13, pp. 1448–1453, Dec. 2002.

[22] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proc. IEEE Wireless Commun. Netw.*, vol. 3, 2003, pp. 1609–1614.

[23] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. Mobile Comput.*, vol. 7, no. 2, pp. 262–274, Feb. 2008.

[24] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, Sep. 2002.

[25] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 1, pp. 78–92, Jan. 2005.

[26] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mobile Netw. Appl.*, vol. 11, no. 5, pp. 723–740, Oct. 2006.

[27] S. Funke, A. Kesselman, F. Kuhn, Z. Lotker, and M. Segal, "Improved approximation algorithms for connected sensor cover," *Wireless Netw.*, vol. 13, no. 2, pp. 153–164, Apr. 2007.

[28] H. Gupta, Z. Zhou, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," *IEEE/Assoc. Comput. Machinery Trans. Netw.*, vol. 14, no. 1, pp. 55–67, Feb. 2006.

[29] Y. Liu and W. Liang, "Approximate coverage in wireless sensor networks," in *Proc. IEEE Conf. Local Comput. Netw. 30th Anniversary*, Nov. 2005, pp. 68–75.

[30] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. 1st Assoc. Comput. Machinery Int. Workshop Wireless Sensor Netw. Appl.*, 2002, pp. 32–41.

[31] Z. Zhou, S. R. Das, and H. Gupta, "Variable radii connected sensor cover in sensor networks," *Assoc. Comput. Machinery Trans. Sensor Netw.*, vol. 5, no. 1, article 8, pp. 1–36, Feb. 2009.

[32] F. V. C. Martins, E. G. Carrano, E. F. Wanner, R. H. C. Takahashi, and G. R. Mateus, "A dynamic multiobjective hybrid approach for designing wireless sensor networks," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 1145–1152.

[33] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *J. Comb. Optim.*, vol. 11, no. 1, pp. 21–41, Feb. 2006.

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

[35] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 2. Finland, 2001, pp. 472–476.

[36] C. Liu, K. Wu, Y. Xiao, and B. Sun, "Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 6, pp. 562–575, Jun. 2006.

[37] J.-W. Lin and Y.-T. Chen, "Improving the coverage of randomized scheduling in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4807–4812, Dec. 2008.

[38] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2004, pp. 424–432.

[39] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Efficient energy management in sensor networks," in *Ad Hoc and Sensor Networks, Wireless Networking and Mobile Computing*, Y. Pan and Y. Xiao, Eds. New York: Nova Science Publishers, 2005, pp. 71–90.

[40] H. Zhang and J. C. Hou, "Maximising α-lifetime for wireless sensor networks," *Int. J. Sensor Netw.*, vol. 1, nos. 1–2, pp. 64–71, 2006.

[41] M. Cardei, D. MacCallum, M. X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, "Wireless sensor networks with energy efficient organization," *J. Interconnect. Netw.*, vol. 3, nos. 3–4, pp. 213–229, 2002.
[42] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.
[43] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1996.
[44] A. Brabazon and M. O'Neill, "An introduction to evolutionary computation in finance," *IEEE Comput Intell. Mag.*, vol. 3, no. 4, pp. 42–55, Nov. 2008.
[45] J. Zhang, H. S.-H. Chung, and W.-L. Lo, "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 326–335, Jun. 2007.
[46] M. Zhang, X. Gao, and W. Lou, "A new crossover operator in genetic programming for object classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1332–1343, Oct. 2007.
[47] J. Zhang, H. S.-H. Chung, W.-L. Lo, S. Y. Hui, and A. K.-M. Wu, "Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithms," *IEEE Trans. Power Electron.*, vol. 16, no. 6, pp. 752–763, Nov. 2001.
[48] C.-C. Lai, C.-K. Ting, and R.-S. Ko, "An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 3531–3538.
[49] R. Williams, *The Geometrical Foundation of Natural Structure: A Source Book of Design*. New York: Dover, 1979, pp. 51–52.

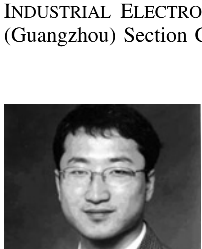
Xiao-Min Hu (S'06) received the B.S. degree in network engineering in 2006, from Sun Yat-Sen University, Guangzhou, China, where she is currently working toward the Ph.D. degree in computer application and technology from the Department of Computer Science.

Her current research interests include artificial intelligence, evolutionary computation, and swarm intelligence, and their applications in design and optimization on wireless sensor networks, network routing technologies, and other intelligence applications.

Jun Zhang (M'02–SM'08) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2002.

From 2003 to 2004, he was a Brain Korean 21 Postdoctoral Fellow at the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Since 2004, he has been with Sun Yat-Sen University, Guangzhon, China, where he is currently a Professor and the Ph.D. Supervisor with the Department of Computer Science. He has authored seven research books and book chapters, and over 80 technical papers in his research areas. His research interests include computational intelligence, data mining, wireless sensor networks, operations research, and power electronic circuits.

Dr. Zhang is currently an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He is currently the Chair of the IEEE Beijing (Guangzhou) Section Computational Intelligence Society Chapter.



Yan Yu received the B.S. degree in industrial automation from Harbin University of Civil Engineering and Architecture, Harbin, China, in 1999. He received the M.S. degree in architecture technique science, and the Ph.D. degree in disaster prevention and reduction engineering from the Harbin Institute of Technology, Harbin, China, in 2001 and 2006, respectively.

He is currently an Assistant Professor with the Institute of Microelectronics, Dalian University of Technology, Dalian, China. He is the author of more than 30 published papers. His research interests include wireless sensor networks, data fusion, ant colony systems, structural health monitoring, and intelligent building.

Henry Shu-Hung Chung (M'95–SM'03) received the B.Eng. and Ph.D. degrees in electrical engineering from Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, in 1991 and 1994, respectively.

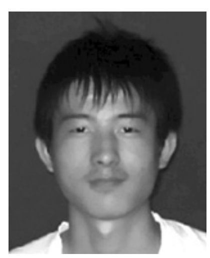
Since 1995, he has been with the City University of Hong Kong (CityU), Hong Kong. He is currently a Professor with the Department of Electronic Engineering and Chief Technical Officer with e.Energy Technology Ltd., an associated company of CityU. He has authored four research book chapters and over 250 technical papers, including 110 refereed journal papers in his research areas, and holds 10 patents. His research interests include time and frequency-domain analysis of power electronic circuits, switched-capacitor-based converters, random-switching techniques, control methods, digital audio amplifiers, soft-switching converters, and electronic ballast design.

Dr. Chung was awarded the Grand Applied Research Excellence Award in 2001 from the CityU. He was the IEEE Student Branch Counselor and was the Track Chair of the Technical Committee on power electronics circuits and power systems of the IEEE Circuits and Systems Society from 1997 to 1998. He was an Associate Editor and a Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART I: FUNDAMENTAL THEORY AND APPLICATIONS from 1999 to 2003. He is currently an Associate Editor of the IEEE TRANSACTIONS ON POWER ELECTRONICS and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART I: FUNDAMENTAL THEORY AND APPLICATIONS.

Yu-Hui Shi (SM'98) received the Ph.D. degree in electrical engineering from South-East University, Nanjing, China, in 1992.

He is currently a Professor with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China, where he also holds the position of the Director of the Research and Postgraduate Office. Before joining XJTLU, he was with the Electronic Data Systems Corporation, Indianapolis, IN. His research interests include the areas of computational intelligence techniques (including swarm intelligence) and their applications.

Prof. Shi is the Editor-in-Chief of the *International Journal of Swarm Intelligence Research*, and an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He is the Chair of the IEEE Computational Intelligence Society Task Force on Swarm Intelligence.



Yuan-Long Li received the B.S. degree in mathematics from Sun Yat-Sen University, Guangzhou, China, in 2009. He is currently working toward the M.S. degree in computer science from the Department of Computer Science, Sun Yat-Sen University.

His research interests include metaheuristics and evolutionary computation.

Xiao-Nan Luo received the B.S. degree from Jiangxi University, China, in 1983, the M.S. degree from Xi'an University of Electronic Science, Xi'an, China, in 1985, and the Ph.D. degree from Dalian University of Technology, Dalian, China, in 1991.

He is currently a Professor and Ph.D. Supervisor with the Department of Computer Science, School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, China, where he is also the Chairman of the Computer Application Institute. His research interests include mobile computing, computer graphics and computer-aided design (CAD), and 3-D CAD.

He was granted the government special allowance by the State Council of China, received the National Science and Technology Progress Prize awarded by the Ministry of Science and Technology of China, and received the National Science Fund for Distinguished Young Scholars granted by the National Nature Science Foundation of China.