

# Multi-layer Genetic Algorithm for Maximum Disjoint Reliable Set Covers Problem in Wireless Sensor Networks

Mayyadah F. Abdulhalim · Bara'a A. Attea

Published online: 24 August 2014  
© Springer Science+Business Media New York 2014

**Abstract** Densely deployment of sensors is generally employed in wireless sensor networks to ensure complete target coverage for a long period of time. Many sensors scheduling techniques have been recently proposed for prolonging the network lifetime. Scheduling sensors into a maximum number of disjoint sets has been modeled, in the literature, as disjoint set covers (DSC) problem which is a well-known NP-hard optimization problem. Unlike other attempts which considers only a simple disk sensing model, this paper addresses the problem of finding the maximum number of set covers while considering a more realistic sensing model to handle uncertainty into the sensors' target-coverage reliability. The paper investigates the development of a simple multi-layer genetic algorithm (GA) whose main ingredient is to support selection of a minimum number of sensors to be assigned to maximum number of set covers. With the aid of the remaining unassigned sensors, the reliability of DSCs provided by the GA, can further be enhanced by a post-heuristic step. By identifying the upper bound of number of disjoint set covers, the GA can successively construct covers, each of which is of minimal sensor cost. Performance evaluations on solution quality in terms of both number of set covers and coverage reliability are measured and compared through extensive simulations, showing the effectiveness of the proposed GA.

**Keywords** Coverage · DSC · Energy efficiency · Genetic algorithm · Probabilistic sensing model · Wireless sensor networks

---

M. F. Abdulhalim  
Department of Computer Science, University of Bahrain, Sakhir, Bahrain  
e-mail: mfaisal@uob.edu.bh; fabdulhalim68@gmail.com

B. A. Attea (✉)  
Department of Computer Science, Baghdad University, Baghdad, Iraq  
e-mail: baraaali@yahoo.com; baraali@scbaghdad.edu.iq

## 1 Introduction

Recently, many applications, ranging from remote harsh field monitoring to surveillance and smart homes, have been directed towards studying and building their backbones based on Wireless sensor networks (WSNs). The dense ad-hoc deployment of such sensors from an aircraft into the monitoring area can result in network configurations with adequate targets coverage level. However, recharging or replacing a sensor's battery is generally infeasible. Hence, efficient utilization of the limited energy is one of the critical design considerations in WSNs. Energy-aware mechanism has been substantially pursued by the research community in order to form long lived WSNs. Energy saving techniques can generally be classified in the following categories:

1. Energy-efficient data aggregation, gathering and routing;
2. Power management by adjusting the transmission and/or sensing range of sensor nodes; and
3. Sensor wake-up scheduling to alternate between active and idle state.

In this paper, we will consider the third approach to prolong the WSN lifetime while completely monitoring the targets set. In this class of techniques, sensor activities are scheduled into disjoint sensor subsets, or *set covers*, and each set cover (hereinafter, interchangeably called, *sensor cover*) needs to satisfy the coverage constraints. At each interval of the whole WSN's lifetime, only one sensor cover (*active sensor cover*) is working to provide the required sensing functionality while the remaining sensor covers, with their sensors, are in the low-energy *sleep* mode. Once the active sensor cover runs out of energy, another sensor cover will be selected to enter the active mode and provide the functionality continuously. Thus, the more sensor covers we can find, the longer sensor network lifetime will be prolonged. It has been proven that this problem is a generalization of the minimum set cover problem [1] and proves its NP-completeness [2,3].

Many attempts in the literature have been proposed for solving DSC problem in WSNs using either heuristic or meta-heuristic (like genetic algorithms) approaches. 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 [4–11].

In [2], a heuristic approach called the “most constrained–minimally constraining covering (MCMCC)” is proposed to select and successively activate mutually exclusive sets of covers, where every set completely covers the entire area. Their method gives priority to sensors which cover a high number of uncovered fields, cover sparsely covered fields and do not cover fields redundantly. This method achieves energy savings by increasing the number of disjoint covers. The DSC problem has been solved in [12] using integer programming. The DSC problem is reduced to a maximum flow problem and solved using mixed integer programming. By a branch and bound method, the maximum covers based on mixed integer programming algorithm (MC-MIP) acts as an implicit exhaustive search to guarantee finding the optimal solution.

The definition of DSC problem has been re-formulated in [3,13], and [14] to include additional coverage constraints. The definition of DSC problem has been generalized in [3] to a maximum non-disjoint set covers (MSC) problem and solved it using, linear programming, and greedy heuristics. The extended problem in MSC lets the sensors to participate in multiple sets. In [13], the DSC problem has been extended to include connectivity constraint as well. Then, the Connected Set Covers (CSC) problem has as objective finding a maximum number of set covers such that each sensor to be activated should be connected to the base station.

In [14], DSC problem has been extended to include sensor coverage-failure probability. Each sensor is associated with sensor's failure probability (comes from several facts, e.g., manufacture, weather in the monitoring area, interferences to the sensors, or unexpected accidents). The proposed Maximum Reliability Sensor Covers (MRSC) problem has been solved in [14] using a heuristic greedy algorithm to compute the maximal number of set covers that satisfy a user specified coverage-reliability threshold.

The work in [15–17] also provides solutions to the DSC problem in WSNs but using the meta-heuristic framework of evolutionary and genetic algorithms. Like the previous mentioned heuristic methods, the genetic algorithms (GAs) proposed in [15–17] assume simple and common isotropic (i.e., disc) sensing model. Each sensor in this definite range law approximation model is associated with a sensing area which is represented by a circle and it successfully detects anything falling only within its sensing range. In a more realistic scenario, the sensing region of a sensor could be irregular, resulting in imperfect sensor approximation model. The coverage in this case could be expressed in probabilistic terms [18–20]. In probabilistic sensing model, there is a measure of uncertainty in sensor signal-detection being expressed by a value from 0 to 1. For reliable coverage with certainty threshold  $c_{th}$ , the detection uncertainty of each target should not exceed  $1 - c_{th}$ .

According to [21], the upper bound,  $ub$ , of the maximum number of disjoint complete set covers depends on the size of the target area, the total number of sensors, the sensors' locations, and their sensing ranges.  $ub$  can be estimated as the minimum number of sensors covering the *most sparsely* covered target, and thus can be computed by polynomial-time algorithms. It has been shown that for the same deployment of sensors and targets,  $ub$  can be increased by enlarging the sensing radius of each sensor [15–17, 21]. However, under probabilistic sensing models, the certainty threshold  $c_{th}$  should also effect on the number of sensors reliably covering the most sparsely covered target. As more reliability degree to cover a target is required,  $ub$  becomes smaller.

Unlike other related works, this paper concerns with the applicability of the genetic algorithm for solving the DSC problem while assuming a probabilistic sensing model to reflect the uncertainty in sensor readings. The main contributions of this paper are as follows:

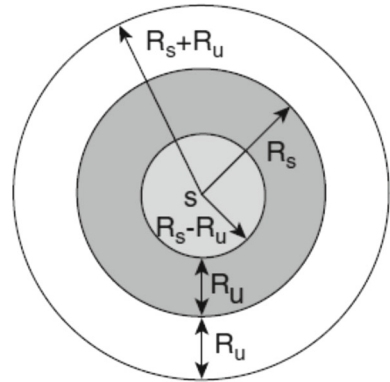
1. With the de-facto definition of the simple genetic algorithm, up to  $ub$  covers can be identified gradually, each of which should maintain low cost in terms of number of sensors to completely cover the targets within the specified certainty threshold.
2. With the incorporation of unassigned sensors, the coverage reliability of each set cover, and hence, the total network's coverage reliability can be further improved. A post-heuristic operator weighs each assigned and/or unassigned sensor to the membership of each set cover.

In what follow we first briefly describe the DSC problem in WSNs and its related system model. Then, in Sect. 3, we introduce the proposed multi-layered genetic algorithm and a post-heuristic operator tailored for solving DSC problem in WSNs. The results of the proposed genetic algorithm are then compared with a related work in Sect. 4. Finally, Sect. 5 concludes the current work and hints some further ramifications.

## 2 DSC Problem in WSNs

In order to model the system, we will assume that the investigated WSNs have 2D sensing area  $\mathcal{A}$  with known size  $(X_{max}, Y_{max})$ . We will also assume that  $\mathcal{A}$  has a set  $\mathcal{T}$  (i.e., target set) of  $n$  targets with known locations, i.e.,  $\mathcal{T} = \{(x_{t1}, y_{t1}), (x_{t2}, y_{t2}), \dots, (x_{tn}, y_{tn})\}$ . There

**Fig. 1** Probabilistic sensing model



are  $m$  homogenous sensors  $\mathcal{S} = \{(x_{s1}, y_{s1}), (x_{s2}, y_{s2}), \dots, (x_{sm}, y_{sm})\}$  having the same sensing range  $R_s$ . All the sensors are dropped randomly in  $\mathcal{A}$  ( $1 \leq \forall i \leq m \mid (x_{si}, y_{si}) = ([0, X_{max}], [0, Y_{max}])$ ). Depending on the sensing range  $R_s$ , each sensor will be responsible for sensing and covering a part of  $\mathcal{A}$ . We will consider a probabilistic sensing model [19,20] to define the notion of the probabilistic coverage of a target  $t_j = (x_{tj}, y_{tj})$  by a sensor  $s_i$ .

$$Coverage(s_i, t_j) = \begin{cases} 0 & \text{if } R_s + R_u \leq d(s_i, t_j) \\ e^{-\lambda a^\beta} & \text{if } R_s - R_u < d(s_i, t_j) < R_s + R_u \\ 1 & \text{if } R_s - R_u \geq d(s_i, t_j) \end{cases} \quad (1)$$

where  $R_u$  is a measure of the uncertainty in sensor detection.  $d(s_i, t_j)$  is the Euclidean distance  $\sqrt{(x_{si} - x_{tj})^2 + (y_{si} - y_{tj})^2}$  between sensor  $s_i$  and target  $t_j$ .  $a = d(s_i, t_j) - (r_s - r_u)$ , and  $\lambda$  and  $\beta$  are probabilistic detection parameters to measure detection strength when a target point lies within the interval  $\{R_s - R_u, R_s + R_u\}$ . It causes coverage value to exponentially decrease as the distance increase. All points that lie within a distance of  $R_s - R_u$  from the sensor are said to be 1-covered. Beyond the distance  $R_s + R_u$ , all the points have 0-coverage by this sensor (see Fig. 1).

To save energy and prolong WSN’s lifetime, sensors in the sensor set  $\mathcal{S}$  should be divided into duty-cycling *sensor cover* subsets, each of which can cover all the interested targets in  $\mathcal{T}$ . In the literature and under the traditional Boolean sensing model (Eq. 2),

$$Coverage(s_i, t_j) = \begin{cases} 1 & \text{if } d(s_i, t_j) \leq R_s \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

the definition of the sensor cover could be formulated as:

**Definition 1** (*Sensor Cover*). Given a WSN consists of target set  $\mathcal{T}$  and sensor set  $\mathcal{S}$ , where each sensor  $s_i \in \mathcal{S}$  can be represented as a subset  $\mathcal{T}_i \subset \mathcal{T}$ , such that  $t_j \in \mathcal{T}_i$  if and only if  $Coverage(s_i, t_j) = 1$ . Any subset  $\mathcal{S}_i \subset \mathcal{S}$  that can completely cover all the target set  $\mathcal{T}$  is termed as a sensor cover.

According to the above definition, the disjoint (sensor/set) cover problem (DSC) can be formulated as:

**Definition 2** (*Disjoint Sensor Covers Problem-DSC*). Given a finite collection  $C$  of subsets of a target set  $T$ , find the family of the maximum number of disjoint sensor covers

$S_1, S_2, \dots, S_{max}$  for  $T$ . Every sensor cover  $C_i$  is a subset of  $C$ ,  $C_i \subseteq C$ , such that every target element  $t_k$  of  $T$  belongs to at least one sensor member of  $C_i$ , and for any two sensor covers  $C_i$  and  $C_j$ ,  $C_i \cap C_j = \emptyset$ .

However, considering probabilistic sensing model, the definition of the traditional sensor cover needs to be re-formulated, here, as:

**Definition 3 (Reliable Sensor Cover).** Given a WSN consists of target set  $\mathcal{T}$  and sensor set  $\mathcal{S}$ , where each sensor  $s_i \in \mathcal{S}$  can be represented as a subset  $\mathcal{T}_i \subset \mathcal{T}$ , such that  $t_j \in \mathcal{T}_i$  if and only if  $Coverage(s_i, t_j) \geq c_{th}$ . Any subset  $\mathcal{S}_i \subset \mathcal{S}$  that can satisfy a user coverage constraint  $c_{th}$  to cover all the targets in  $\mathcal{T}$  is termed as a reliable sensor cover or reliable set covering. Formally speaking:

$$Cover(\mathcal{S}_i, \mathcal{T}) = \begin{cases} 1 & \text{if } \forall t \in \mathcal{T} \rightarrow \exists s \in \mathcal{S}_i | Coverage(s, t) \geq c_{th} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Now, the problem of finding the maximum number of disjoint set covers (DSC) could be turned into the problem of finding the maximum number of disjoint reliable sensor covers (DRSC), and can be formulated as:

**Definition 4 (Disjoint Reliable Sensor Covers Problem-DRSC).** Given a collection  $\mathcal{S}$  of reliable subsets of a finite set  $\mathcal{T}$ , find the maximum number of disjoint reliable covers for  $\mathcal{T}$ . Every cover  $\mathcal{S}_i$  is a subset of  $\mathcal{S}$ ,  $\mathcal{S}_i \subseteq \mathcal{S}$ , such that every element  $t_j$  of  $\mathcal{T}$  belongs to at least one member of  $\mathcal{S}_i$ , and for any two covers  $\mathcal{S}_i$  and  $\mathcal{S}_j$ ,  $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ . Now, the upper bound  $ub$  of the maximum number of disjoint reliable covers is determined by the most sparsely covered target and can be calculated as:

$$ub = |\mathcal{CS}| = \arg \min_{i=1,2,\dots,n} (|\mathcal{S}_i|) \tag{4}$$

where  $\mathcal{CS}$  denotes the set of sensors (called *Critical sensors Set*) covering target  $t_i$  with certainty equal to or larger than a user specified threshold  $c_{th}$ . As the most sparsely covered target only covered by  $ub$  sensors, the maximum number of disjoint reliable covers  $N$  is no larger than  $ub$  (i.e.,  $N \leq ub$ ).

### 3 The Proposed Multi-layer Genetic Algorithm

The GA simulates the biological processes of natural selection, reproduction, and mutation to iteratively evolve species of individual solutions to become more and more adapted to the problem environment. The proposed GA can be described as a process formulated in multi-layered fashion. Let  $\mathcal{S}_{sleep}^0 = \mathcal{S}$ ,  $\mathcal{S}_{active}^0 = \emptyset$  and  $GA^i : \mathcal{S}_{sleep}^{i-1} \rightarrow \{\mathcal{S}_{sleep}^i, \mathcal{S}_{active}^i\}$ ,  $i = 1, 2, \dots, ub$  be the  $i$ th GA process that iteratively evolves a population  $\rho$  of solutions, using genetic operators, toward the  $i$ th best *set cover* solution in terms of minimum number of sensors that reliably cover all targets. Thus, the objective function  $\Phi$  of the  $i$ th GA can be defined as:

$$\Phi^i : \text{Minimize } |\mathcal{S}_{active}^i| \tag{5}$$

where  $\mathcal{S}_{sleep}^i$  denotes the set of *alive* sensors that has no sensing functionality in the  $i$ th interval of the network’s lifetime.

Since no more than  $ub$  set covers can be generated, then we have no more than  $ub$  GA-layers (see Eq. 6). The  $i$ th layer (also called the  $i$ th GA) only explores the solution space

of the unassigned (i.e., sleep) sensors set resulted from the  $i - 1$ th GA. For the  $1^{st}$  GA, the number of unassigned sensors is the whole set of sensors  $\mathcal{S}$ .

$$GA = GA^{1\circ}GA^{2\circ} \dots \circ GA^l | l \leq ub \tag{6}$$

where:

$$l = \begin{cases} ub & \text{if } |\mathcal{S}_{sleep}^{ub-1}| > 0 \wedge Cover(\mathcal{S}_{sleep}^{ub-1}, \mathcal{T}) = 1 \\ < ub & \text{otherwise} \end{cases} \tag{7}$$

### 3.1 Space and Solution Configurations

The choice of a good solution representation is a critical issue for the applicability and performance of evolutionary algorithm. Solution representation is highly problem dependent and related to the evolution operations. In our algorithm design, each individual solution  $\mathbb{P}^i$  of  $GA^i$  is represented as a fixed-length vector of size  $m^i = |\mathcal{S}_{sleep}^{i-1}|$ , where each *gene* <sub>$j$</sub>  controls the *active/sleep scheduling* of the corresponding  $j$ th sensor in  $\mathcal{S}_{sleep}^{i-1}$ . Formally speaking, for population  $i$  of  $K$  individuals, each of  $m^i$  genes,

$$\begin{aligned} & \forall k \in \{1, \dots, K\} \text{ and } \forall j \in \{1, \dots, m^i\} : \\ & \mathbb{P}_k^i = \left( \mathbb{P}_{k,1}^i, \mathbb{P}_{k,2}^i, \dots, \mathbb{P}_{k,m^i}^i \right) \text{ s.t. :} \\ & \mathbb{P}_{k,j}^i = \begin{cases} 1, & \text{if } s_j \text{ is active} \\ 0, & \text{if } s_j \text{ is sleep} \end{cases} \end{aligned} \tag{8}$$

Then, the whole configuration space  $\delta^i$  for the  $i$ th GA can be created by the Cartesian product of activation/inactivation of all  $m^i$  unassigned sensors being resulted from in the  $i - 1$ th GA:

$$\delta^i = \prod_{i=1}^{m^i} (\{0, 1\}) = 2^{m^i} \tag{9}$$

where 0 means inactive (i.e., unassigned) sensor, while 1 means active (i.e., assigned) sensor. Let us to consider that  $GA^{i, 1 \leq i \leq ub}$  handles only  $K \ll \delta^i$  different individual solutions at a time. It starts with an initial random population  $\rho_0 \subset \delta^i, |\rho_0| = K$  and continues until a maximum number of generations  $max_{gen}$  has been reached. Each generation  $\Psi^i : \rho \rightarrow \rho'$  consists of four main operators: individual repair, parent selection, crossover, and mutation. Thus,  $\Psi$  can be decomposed into:

$$\Psi^i = \Psi_{rep}^i \circ \Psi_{sel}^i \circ \Psi_x^i \circ \Psi_{mu}^i \tag{10}$$

### 3.2 Repair Operator and the Fitness Function

Before evaluating each individual, infeasible set cover solutions should be transformed into feasible ones by means of a problem-specific repair operator. Infeasible solutions are those which suffer from either the existence of coverage-holes or let the targets to be over-covered by more than need sensors. The main idea of the repair operator is to make hole-free targets coverage with as less number of sensors as possible. The process of the proposed repair operator  $\Psi_{rep}^i$  of the  $i$ th GA is presented next (see Algorithm 1).

$$\Psi_{rep}^i : \mathbb{P}_k^i \rightarrow \mathbb{P}_{k'}^i \tag{11}$$

It takes as input the individual  $\mathbb{P}_k^i, 1 \leq k \leq K$  and the set of unassigned sensors  $\mathcal{S}_{sleep}^{i-1}$  resulted from the previous GA layer. First, it check whether the active sensors set  $\mathcal{S}_k$  selected by  $\mathbb{P}_k^i$  (i.e.,  $\mathcal{S}_k = \{s_j \mid \mathbb{P}_{kj}^i = 1\}$ ) forms coverage-hole or dense-coverage under the user-specified reliability threshold. In case of coverage-hole,  $\Psi_{rep}^i$  will randomly draw from  $\mathcal{S}_{sleep}^{i-1}$  set one sensor at a time and collect it with  $\mathcal{S}_k$  (i.e.,  $\mathcal{S}_k = \mathcal{S}_k \cup s \mid s \in \mathcal{S}_{sleep}^{i-1}$  and  $\mathcal{S}_{sleep}^{i-1} = \mathcal{S}_{sleep}^{i-1} - s$ ) until the new set form hole-free set cover. On the other hand, if  $\mathcal{S}_k$  forms dense-coverage,  $\Psi_{rep}^i$  will randomly deactivate one sensor at a time (i.e.,  $\mathcal{S}_k = \mathcal{S}_k - s \mid s \in \mathcal{S}_k$  and  $\mathcal{S}_{sleep}^{i-1} = \mathcal{S}_{sleep}^{i-1} + s$ ) until it can form complete coverage with less number of sensors.

**Algorithm 1: Repair Operator** ( $\mathbb{P}_k^i, \mathcal{S}_{sleep}^{i-1}, \mathbb{P}_k^i$ )

```

1:  set  $S$  to the active sensors selected by  $\mathbb{P}_k^i$ 
     $\forall j \in [1, m^{i-1}]: S \leftarrow \{\mathbb{P}_{kj}^i \mid \mathbb{P}_{kj}^i = 1\}$ 
2:  if  $Cover(S, \mathcal{T}) == 1$  /* no coverage hole */
3:  while  $Cover(S, \mathcal{T}) == 1$ 
4:  select a random sensor  $s \in S$ 
5:  set  $S \leftarrow S - s$ 
6:  set  $\mathcal{S}_{sleep}^{i-1} \leftarrow \mathcal{S}_{sleep}^{i-1} \cup s$ 
7:  end while
8:  set  $S \leftarrow S \cup s$  /* now  $Cover(S, \mathcal{T}) = 1$  */
9:  else /* coverage holes exist after using the sensors in  $S$  */
10: while  $Cover(S, \mathcal{T}) == 0$ 
11: select a random sensor  $s \in \mathcal{S}_{sleep}^{i-1}$ 
12: set  $S \leftarrow S \cup s$ 
13: set  $\mathcal{S}_{sleep}^{i-1} \leftarrow \mathcal{S}_{sleep}^{i-1} - s$ 
14: end while
15: end if
    
```

Then, to evaluate each individual solution  $\mathbb{P}_k^i$ , the fitness function  $\Phi$  simply sums the number of active sensors being selected by the corresponding solution.

$$\forall k \in [1, K]$$

$$\Phi \left( \mathbb{P}_k^i \right) = \sum_{j=1}^{m^i = |\mathcal{S}_{sleep}^{i-1}|} \mathbb{P}_{kj}^i \tag{12}$$

3.3 Selection, Crossover, and Mutation Operators

The remaining genetic operators follow the de-facto standard operators found in the simple genetic algorithms.. The binary tournament selection operator is used to choose one of two random individuals  $\mathbb{P}_1$  and  $\mathbb{P}_2$ . A proportion  $p_c = 0.6$  of pairs of parents are then selected for crossover. Two cut points  $r1, r2 \sim \{1, \dots, m^i - 1\}$ , are randomly selected, and the participating parent individuals,  $\mathbb{P}_1$  and  $\mathbb{P}_2$  are then swapped at  $gene_j \mid r1 \leq j \leq r2$ . Each  $gene_j \mid 1 \leq j \leq m^i$ . in the new individuals is then mutated with small probability  $p_m = 0.1$ .

$$\forall k \in \{1, \dots, K\}$$

$$\Psi_{sel} : \{\mathbb{P}_{k1}, \mathbb{P}_{k2}\} \rightarrow \mathbb{P}_k \tag{13}$$

$$\forall k \in \{1, \dots, K/2\}$$

$$\Psi_x : \{\mathbb{P}_{k1}, \mathbb{P}_{k2}\} \rightarrow \{\mathbb{P}'_{k1}, \mathbb{P}'_{k2}\}$$

$$\begin{aligned} \mathbb{P}'_{k1} &= (\mathbb{P}_{k1,1}, \dots, \mathbb{P}_{k1,r1}, \mathbb{P}_{k2,r1+1}, \dots, \mathbb{P}_{k2,r2}, \mathbb{P}_{k1,r2+1}, \dots, \mathbb{P}_{k1,m^i}) \\ \mathbb{P}'_{k2} &= (\mathbb{P}_{k2,1}, \dots, \mathbb{P}_{k2,r1}, \mathbb{P}_{k1,r1+1}, \dots, \mathbb{P}_{k1,r2}, \mathbb{P}_{k2,r2+1}, \dots, \mathbb{P}_{k2,m^i}) \end{aligned} \tag{14}$$

$$\forall k \in \{1, \dots, K\} \wedge \forall j \in \{1, \dots, m^i\}$$

$$\begin{aligned} \Psi_{mu} : \mathbb{P}_k &\rightarrow \mathbb{P}'_k \\ \mathbb{P}_{k,j} &= \begin{cases} \mathbb{P}_{k,j} & \text{if } r > p_m \\ 1 - \mathbb{P}_{k,j} & \text{if } r \leq p_m \end{cases} \end{aligned} \tag{15}$$

The mechanisms of the genetic operators being defined by repair, fitness evaluation, selection, crossover and mutation transform a complete population of solutions into another complete population and after a specified number of generations  $max_{gen}$ , the best individual solution (in terms of minimum  $\Phi$ ) will be produced. The definition of the best individual of the  $i$ th GA can be formulated as:

$$Best\mathbb{P}^i : \iff \nexists \mathbb{P}^i \in \rho_{max_{gen}} | \Phi(\mathbb{P}^i) < \Phi(Best\mathbb{P}^i) \tag{16}$$

### 3.4 Post-heuristic Operator

The best solution set  $Best\mathbb{P} = \{Best\mathbb{P}^i | i = 1, \dots, l\}$  provided by GA can further be improved in terms of coverage reliability by forwarding it to a post-heuristic operator dedicated for this purpose. Algorithm 2 presents the steps of this heuristic. It operates iteratively layer followed by layer (line 1), improving the reliability of each layer  $i$  by exploiting the existing unassigned sensors (being gathered in  $S^l_{sleep}$ ) and/or replacing the existing active sensors (being gathered in  $Best\mathbb{P}^i$ ).

---

**Algorithm 2: Post-Heuristic** ( $\{Best\mathbb{P}^i\}, S^l_{sleep}, l$ )

---

```

1: for  $i = 1$  to  $l$  do
2:     /* a new cover  $S^i_{active}$  will be formed */
   set  $S^i_{active} = \emptyset$ 
3:     set  $S$  to the sensors from both  $Best\mathbb{P}^i$  and  $S^l_{sleep}$ 
    $S \leftarrow \{Best\mathbb{P}^i_j | Best\mathbb{P}^i_j = 1\} \cup \{S^l_{sleep}\}$ 
4:     set contributed sensors set  $S' \leftarrow \emptyset$ 
5:     set  $Target \leftarrow \mathcal{T}$ 
6:     while  $Target \neq \emptyset$ 
7:         select a sensor  $s \in S$  that contributes to the most
8:         reliable coverage to a target  $t \in \mathcal{T}$ 
9:         set  $Target \leftarrow Target - t; S' \leftarrow S' \cup s$ 
10:    end while
11:    remove  $S' \rightarrow S^i_{active}$ 
12:    remove  $\{S - S'\} \rightarrow S^l_{sleep}$ 
13: end for

```

---

## 4 Performance Evaluations

In this section we will measure the performance of the proposed multi-layer GA (denoted hereinafter as  $mlGA$ ) for solving DRSC problem. The evaluation is presented in terms of number of set covers obtained, set covers' coverage reliability, and sensors cost. The results are obtained after setting WSNs and algorithm parameters into the following. The simulation area is square-shaped with side length  $X_{max} = 1,000$  m of 10 randomly distributed targets.



The simulation is mainly divided into nine groups according to nine different settings of sensor density: 25, 50, 75,  $\dots$ , 225. For each group, we will vary the sensing range of the sensor nodes  $R_s$  to five different values {400, 500, 600, 700, 800}, to get a different test instance (i.e., we have 45 different test instances). Each test instance  $TI^i$ ,  $i = 1, \dots, 45$  includes 10 random WSNs with different configurations. Thus the overall simulation examines a total of 450 random network configurations. As we have random WSN configurations, we may get different value for the upper bound of set covers  $ub$ , for the same parameter settings. Thus in each test instance, the presented results also indicate the average number of upper bound of set covers (denoted by  $\overline{ub}$ ). In other words,  $\overline{ub}$  is used to represents the maximum network's lifetime. Uncertainty level  $R_u$  is set to  $R_s * 0.5$  units, both  $\lambda$  and  $\beta$  are set to 0.5, and  $c_{th}$  is set to 0.001. The setting of the probabilistic coverage parameters also influences the overall network's coverage reliability. As studying the impact of varying these parameters is out of the scope of this paper, we fixed these parameters to one setting and evaluate the average coverage reliability  $\bar{r}$  (i.e., the average signal strength being detected from all the targets) of all network configurations in each test instance. Population size is set to 50 and will be allowed to evolve 500 times.

The performance of the proposed *mlGA* has been compared with the performance of Genetic Algorithm for Maximum Disjoint Set Covers (GAMDSC). GAMDSC has been proposed in [15] to extend the lifetime of WSNs. GAMDSC also is simple genetic algorithm but uses integer representation for chromosome representation (like all GA related work in the literature, e.g., [16] and [17]), traditional uniform crossover, traditional creep mutation, and a proposed scatter operator to evolve their chromosome solutions. The fitness function of GAMDSC has been devoted to maximize number of complete set covers in each chromosome. The performance of GAMDSC has been compared in [15] against the most constrained-minimum constraining heuristic (MCMCC) [2] and the exhaustive heuristic of Maximum Covers using Mixed Integer Programming (MCMIP) [12] for solving the DSC problem in WSNs. Simulation results in [15] show that GAMDSC can get near-optimal solutions and improve the performance of MCMCC by 16% in terms of the obtained sensor covers. As compared with the exhaustive search of MCMIP, GAMDSC can get results with acceptable computation time.

#### 4.1 Solution Quality of Multi-layer GA Versus GAMDSC

First, results are presented in Tables 1, 2 and 3 to compare the quality of results obtained by GAMDSC, and our algorithm. Tables 1, 2 and 3 quantitatively present the number of set covers (denoted by  $ub$ ) being obtained by both algorithms. To examine the quality of solutions, we present  $\overline{ub}$  as reference value. Unlike work in [15], which evaluates the performance of GAMDSC for small values of  $\overline{ub}$  ranged from 5 to 39, the results here evaluate both GAMDSC and *mlGA* for larger setting of  $\overline{ub}$  ranges from 3.7 to 92.1. In the tables, the solution providing the closest value to  $ub$  in each test instance has been bolded. Figures 2, 3 and 4 qualitatively depict the performance of GAMDSC and *mlGA*. Moreover, Figs. 5, 6 and 7 depict the accuracy percentage of obtained number of set covers of each algorithm. For further qualitative comparison, Fig. 8 depicts the obtained  $ub$  for three different settings of sensor density, i.e.,  $m = 25, 125$  and 225 and the five settings of sensing range. Additionally, Fig. 9 depicts  $ub$  for the two extremes of sensing range  $R_s = \{400, 800\}$ , and the nine different settings of sensor density.

Results in Tables 1, 2, 3 and Figs. 2, 3, 4, 5, 6, 7 reveal that the proposed *mlGA* significantly outperforms GAMDSC in terms of finding near optimal number of set covers. For  $\overline{ub} > 8.8$  (i.e., in  $TI^7$  to  $TI^{45}$ ), the performance of GAMDSC deteriorates, while *mlGA* continues

**Table 1** Comparison of the maximum number of the generated Set Covers for 10 WSNs in each test instance with number of sensors = {25, 50, 75} and  $R_s = \{400, 500, \dots, 800\}$

$TI$	$m$	$R_s$	$\overline{ub}$	GAMDSC	$mlGA$
$TI^1$	25	400	3.7	3.7	3.7
$TI^2$		500	4.3	4.3	4.3
$TI^3$		600	5.7	5.7	5.7
$TI^4$		700	7.5	<b>7.4</b>	7.2
$TI^5$		800	9.1	<b>9.1</b>	8.9
$TI^6$	50	400	8.8	<b>8.5</b>	8.4
$TI^7$		500	11.0	10.7	<b>10.8</b>
$TI^8$		600	13.1	12.9	<b>13.0</b>
$TI^9$		700	16.5	15.5	<b>15.6</b>
$TI^{10}$		800	19.2	17.9	<b>18.4</b>
$TI^{11}$	75	400	13.8	13.0	<b>13.6</b>
$TI^{12}$		500	18.0	16.6	<b>17.7</b>
$TI^{13}$		600	21.5	19.5	<b>21.2</b>
$TI^{14}$		700	24.9	22.6	<b>24.7</b>
$TI^{15}$		800	30.8	26.6	<b>29.8</b>

**Table 2** Comparison of the maximum number of the generated Set Covers for 10 WSNs in each test instance with number of sensors = {100, 125, 150} and  $R_s = \{400, 500, \dots, 800\}$

$TI$	$m$	$R_s$	$\overline{ub}$	GAMDSC	$mlGA$
$TI^{16}$	100	400	21.0	15.9	<b>20.2</b>
$TI^{17}$		500	25.4	19.6	<b>24.4</b>
$TI^{18}$		600	30.7	23.8	<b>29.6</b>
$TI^{19}$		700	36.0	28.5	<b>34.6</b>
$TI^{20}$		800	42.0	34.4	<b>39.1</b>
$TI^{21}$	125	400	22.4	18.2	<b>22.2</b>
$TI^{22}$		500	27.9	22.0	<b>27.4</b>
$TI^{23}$		600	33.7	26.4	<b>32.9</b>
$TI^{24}$		700	39.7	31.9	<b>38.7</b>
$TI^{25}$		800	46.8	38.1	<b>45.4</b>
$TI^{26}$	150	400	24.2	20.0	<b>24.2</b>
$TI^{27}$		500	30.7	24.7	<b>30.5</b>
$TI^{28}$		600	37.4	29.8	<b>37.4</b>
$TI^{29}$		700	45.2	35.0	<b>45.0</b>
$TI^{30}$		800	53.4	42.3	<b>52.5</b>

in getting solutions very slightly less than the optimal values. In almost all of the network configurations of these test instances,  $mlGA$  reaches the optimal solution. In other words, we can say that increasing  $\overline{ub}$  restricts the ability of GAMDSC and constraints the usefulness of its components (mainly characterized by the integer-based chromosome representation and the scatter operator) to find optimal or near optimal  $ub$  solutions. On the other hand, the characteristic components of the proposed  $mlGA$  (being specified mainly by the chromosome binary representation and the proposed repair operator) is found to be more robust to find very near optimal or even optimal  $ub$  solutions.

**Table 3** Comparison of the maximum number of the generated set covers for 10 WSNs in each test instance with number of sensors = {175, 200, 225} and  $R_s = \{400, 500, \dots, 800\}$

$TI$	$m$	$R_s$	$\overline{ub}$	GAMDSC	$mlGA$
$TI^{31}$	175	400	35.2	25.4	<b>35.0</b>
$TI^{32}$		500	43.5	30.5	<b>42.7</b>
$TI^{33}$		600	51.4	37.9	<b>50.9</b>
$TI^{34}$	200	700	60.6	45.2	<b>59.5</b>
$TI^{35}$		800	69.7	53.6	<b>67.7</b>
$TI^{36}$		400	35.4	27.4	<b>35.3</b>
$TI^{37}$		500	42.2	33.2	<b>41.9</b>
$TI^{38}$		600	49.0	39.5	<b>48.5</b>
$TI^{39}$	225	700	58.7	47.0	<b>57.4</b>
$TI^{40}$		800	70.5	56.9	<b>69.0</b>
$TI^{41}$		400	46.8	31.5	<b>46.8</b>
$TI^{42}$		500	57.4	39.8	<b>57.2</b>
$TI^{43}$		600	68.2	48.1	<b>67.3</b>
$TI^{44}$		700	81.6	58.5	<b>80.3</b>
$TI^{45}$		800	92.1	68.7	<b>89.0</b>

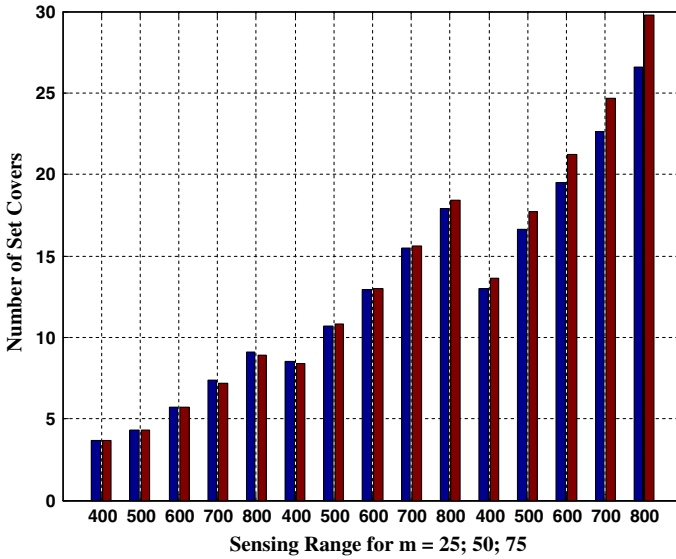
For small values of set covers (i.e.  $\overline{ub} \leq 8.8$ ), both GAMDSC and  $mlGA$  attained nearly an equal number of set covers. In test instances  $TI^1$  to  $TI^3$  both GAMDSC and  $mlGA$  get optimal number of set covers, while in  $TI^4$  to  $TI^6$  GAMDSC finds slightly more covers than  $mlGA$ . However, in all these test instances (i.e.,  $TI^1$  to  $TI^6$ ), we cannot say that both algorithm have an equal performance. As will be seen in Table 4 of the next section, the active sensors cost percentage  $SC\%$  used by GAMDSC is more than that of  $mlGA$ . While  $mlGA$  (in  $TI^1$  to  $TI^6$ ) costs only 47.6–68.8% of the total number of sensors, GAMDSC costs more than 99.0% of the total number of sensors.

As expected from the qualitative results depicted in Figs. 8 and 9, increasing sensor density and/or sensing range provides algorithms with more alternatives for constructing complete and reliable set covers. Moreover, as evident from the figures that increasing sensor density and/or sensing range further discriminates between the performance of both algorithms in terms of finding maximum number of set covers. The full results of both GAMDSC and  $mlGA$  are projected in 3D space in Figs. 10 and 11, respectively.

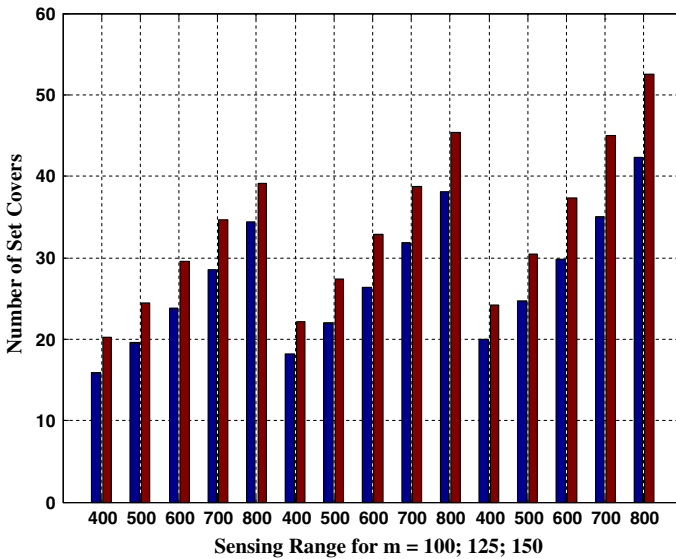
#### 4.2 Comparison of Sensors Cost of Multi-layer GA Versus GAMDSC

In this section, we want to compare the performance of the proposed  $mlGA$  and GAMDSC in terms of active sensors cost used in the generated DSCs (see Tables 4, 5, 6). The goal is to achieve the *highest* number of set covers possible with the *fewest* number of active sensors. Here, sensors cost percentage  $SC\%$  is defined as the ratio between the number of active sensors used in the generated best individual solution  $\mathbb{B}_{\text{estP}}$  and the total number of sensor nodes  $m$ , i.e.,:

$$SC = \frac{|\bigcup_{s \in \mathbb{B}_{\text{estP}}} s = 1|}{m} * 100\% \tag{17}$$



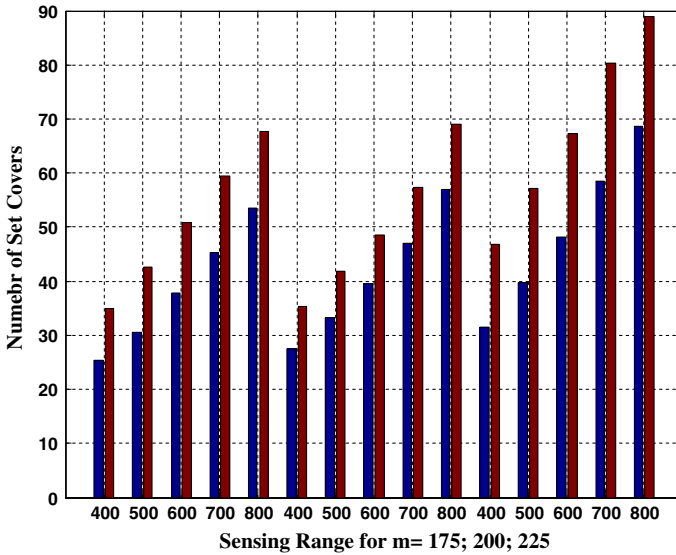
**Fig. 2** Maximum number of the generated Set Covers for GAMDSC: left versus mlGA: right where number of sensors  $m = \{25, 50, 75\}$  and  $R_s = \{400, 500, \dots, 800\}$



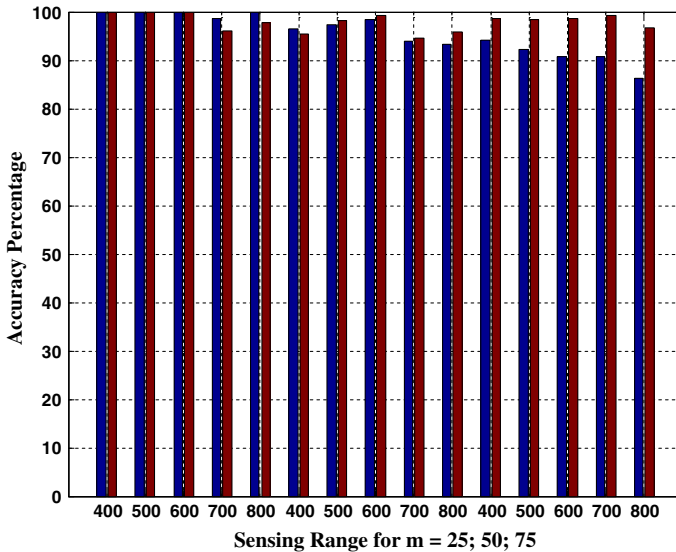
**Fig. 3** Maximum number of the generated Set Covers for GAMDSC: left versus mlGA: right where number of sensors  $m = \{100, 125, 150\}$  and  $R_s = \{400, 500, \dots, 800\}$

Also, here in the tables, the solution providing the smallest  $SC\%$  in each test instance has been bolded. Figures 12, 13 and 14 qualitatively depict the sensors cost percentage  $SC\%$  for GAMDSC and mlGA.

The results presented in Tables 4, 5, 6 and Figs. 12, 13, 14 clearly signify the superiority of mlGA over GAMDSC for activating less number of sensors for a large number of set

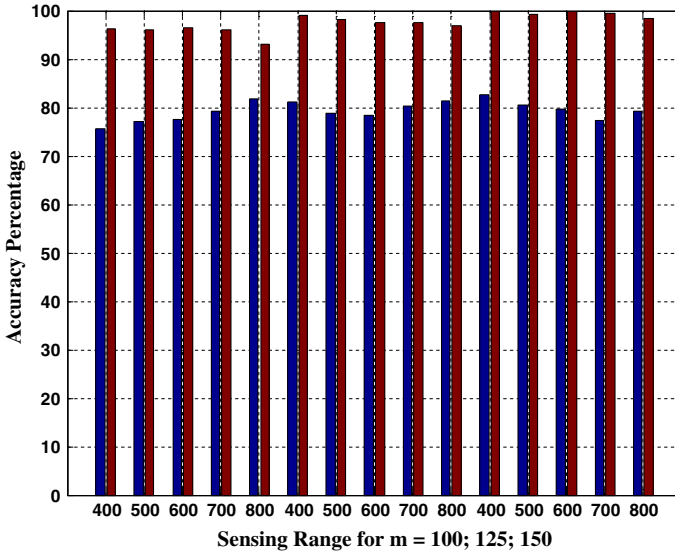


**Fig. 4** Maximum number of the generated Set Covers for GAMDSC: *left* versus *mlGA*: *right* where number of sensors  $m = \{175, 200, 225\}$  and  $R_s = \{400, 500, \dots, 800\}$

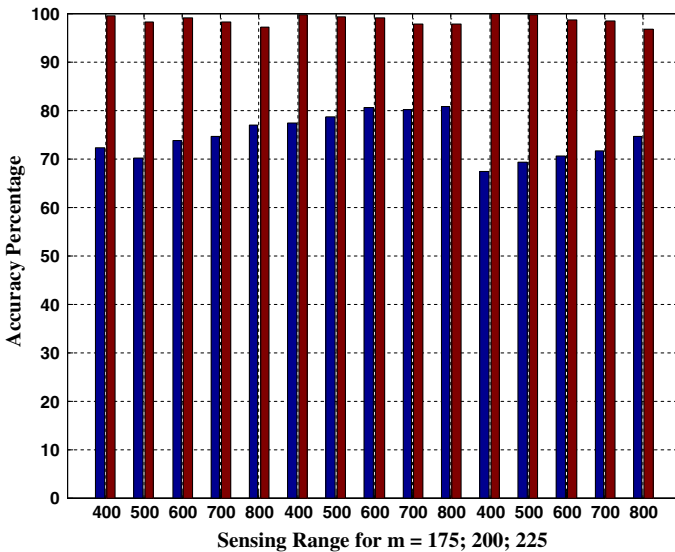


**Fig. 5** Accuracy percentage of the generated Set Covers for GAMDSC: *left* versus *mlGA*: *right* where number of sensors  $m = \{25, 50, 75\}$  and  $R_s = \{400, 500, \dots, 800\}$

covers (i.e., for a longer lived networks). In almost all of the 45 test instances (i.e., in 450 different network configurations), GAMDSC consumes more than 80% of the total number of sensors to achieve networks' lifetime less than that achieved by *mlGA*. For the purpose of a detailed comparison between the performance of the proposed *mlGA* versus GAMDSC under various sensor density and range settings, we will also here refer to the results depicted

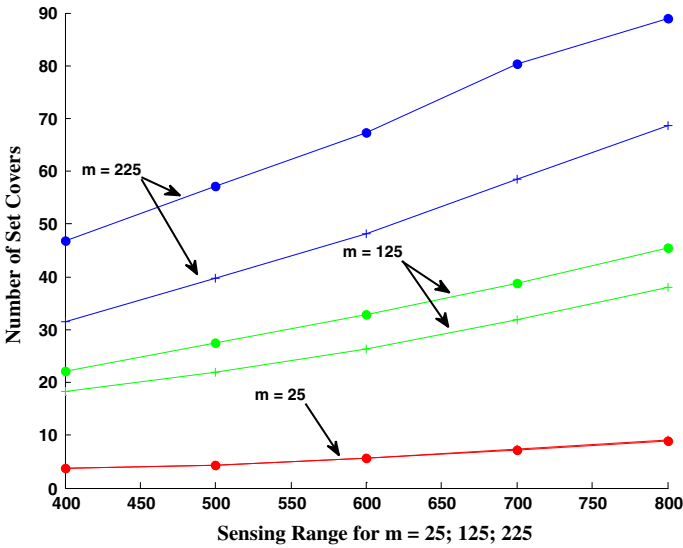


**Fig. 6** Accuracy percentage of the generated Set Covers for GAMDSC: *left* versus *mlGA*: *right* where number of sensors  $m = \{100, 125, 150\}$  and  $R_s = \{400, 500, \dots, 800\}$

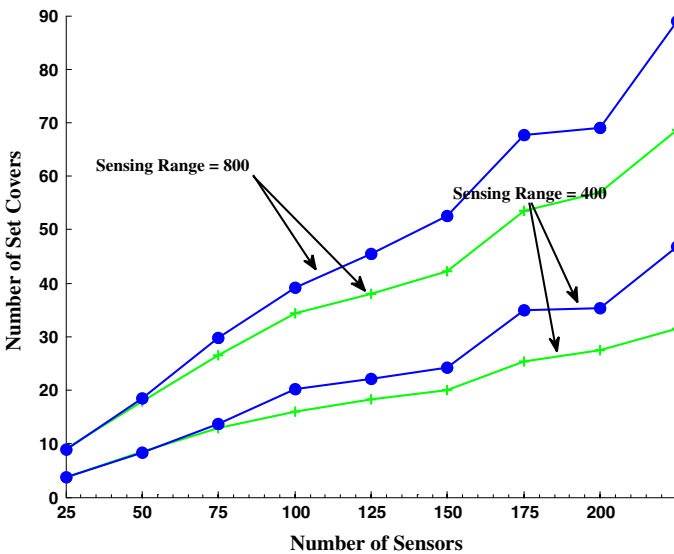


**Fig. 7** Accuracy percentage of the generated Set Covers for GAMDSC: *left* versus *mlGA*: *right* where number of sensors  $m = \{175, 200, 225\}$  and  $R_s = \{400, 500, \dots, 800\}$

in Figs. 5, 6, 7 and  $\bar{ub}$  values presented in Tables 1, 2 and 3. For small lived network instances of  $T^1 - T^{15}$  in Table 4 (where average lifetime, i.e., upper bound of set covers  $\bar{ub}$  ranges from 3.7 to 30.8 as presented in Table 1), GAMDSC consumes more than 90 % of the total sensors to achieve network’s lifetime accuracy above 85 % (as shown in Fig. 5). On the other hand, *mlGA* sacrifices no more than 77 % sensors to achieve lifetime very near to the expected

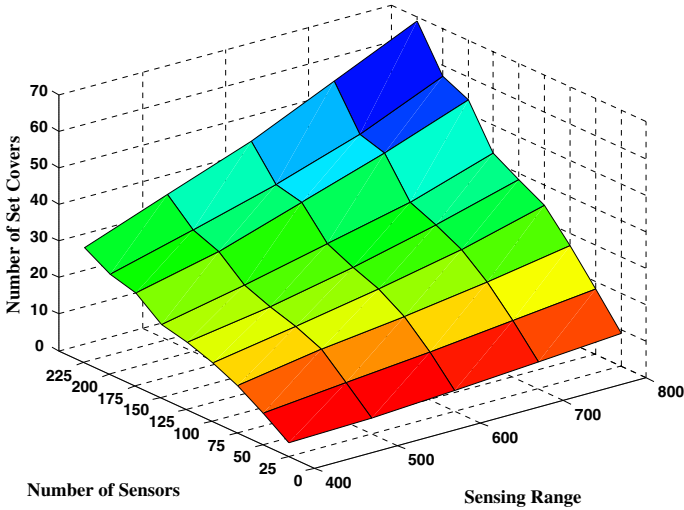


**Fig. 8** Average number of set covers of *GAMDSC* (+ sign) and *mlGA* (filled circle). The simulation is experimented under 150 networks with number of sensors = {25, 125, 225} and  $R_s = \{400, 500, 600, 700, 800\}$

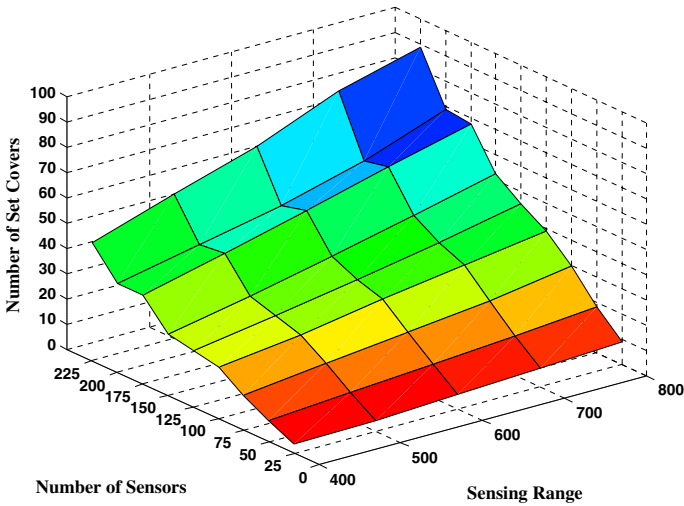


**Fig. 9** Average number of set covers of *GAMDSC* (+ sign) and *mlGA* (filled circle). The simulation is experimented under 180 networks with number of sensors = {25, 50, ..., 225} and  $R_s = \{400, 800\}$

$\overline{ub}$  value (with accuracy more than 94.5 % as shown in Fig. 5). For small to moderate lived networks of instances  $T^{16} - T^{30}$  in Table 5, the expected  $\overline{ub}$  value, in Table 2, ranges from 21.0 to 53.4. In these instances, more than 85 % of the total sensor density is activated by *GAMDSC* to extend the network’s lifetime to no more than 82 % of its upper bound (as shown in Fig. 6). However, for the same instance cases, *mlGA* continues to perform very properly, activating no more than 74 % of the total sensor density to achieve lifetime bound accuracy



**Fig. 10** 3D projection of the results of *GAMDSC* (*x*-coordinate: sensing range, *y*-coordinate: number of sensors, *z*-coordinate: average number of set covers). The simulation is experimented under 450 networks with *number of sensors* = {25, 50, ..., 225} and  $R_S = \{400, 500, \dots, 800\}$



**Fig. 11** 3D projection of the results of *mlGA* (*x*-coordinate: sensing range, *y*-coordinate: number of sensors, *z*-coordinate: average number of set covers). The simulation is experimented under 450 networks with *number of sensors* = {25, 50, ..., 225} and  $R_S = \{400, 500, \dots, 800\}$

no less than 93%. Finally, for the moderate to long lived networks of instances  $T^{31} - T^{45}$  in Table 6, we can see that the upper bound of network lifetime  $\overline{ub}$  (as shown in Table 3) should range from 35.2 to 92.1. In these instances, *GAMDSC* costs more than 75.7% sensors to get no more than 80.7% of  $\overline{ub}$  accuracy. On the other hand, *mlGA* is once again sufficient. It costs up to 77% sensors to get quite accurate number of set covers with 96.6% accuracy.

On overall, the results reflect that many sensors are redundantly scheduled by *GAMDSC* to form complete set covers. On the other hand, *mlGA* has the ability to correctly schedule



**Table 4** Comparison of sensors cost percentage for 10 WSNs in each test instance with *number of sensors* = {25, 50, 75} and  $R_s =$  {400, 500, ..., 800}

<i>TI</i>	<i>m</i>	$R_s$	GAMDSC	<i>mlGA</i>
<i>TI</i> <sup>1</sup>	25	400	100.0	<b>47.6</b>
<i>TI</i> <sup>2</sup>		500	100.0	<b>48.0</b>
<i>TI</i> <sup>3</sup>		600	100.0	<b>54.8</b>
<i>TI</i> <sup>4</sup>		700	99.2	<b>60.4</b>
<i>TI</i> <sup>5</sup>		800	100.0	<b>68.8</b>
<i>TI</i> <sup>6</sup>	50	400	99.0	<b>51.8</b>
<i>TI</i> <sup>7</sup>		500	98.6	<b>56.4</b>
<i>TI</i> <sup>8</sup>		600	99.0	<b>60.6</b>
<i>TI</i> <sup>9</sup>		700	97.2	<b>65.2</b>
<i>TI</i> <sup>10</sup>		800	96.2	<b>70.6</b>
<i>TI</i> <sup>11</sup>	75	400	96.8	<b>53.46</b>
<i>TI</i> <sup>12</sup>		500	95.2	<b>61.2</b>
<i>TI</i> <sup>13</sup>		600	94.66	<b>66.13</b>
<i>TI</i> <sup>14</sup>		700	93.73	<b>69.06</b>
<i>TI</i> <sup>15</sup>		800	91.73	<b>76.8</b>

**Table 5** Comparison of sensors cost percentage for 10 WSNs in each test instance with *number of sensors* = {100, 125, 150} and  $R_s =$  {400, 500, ..., 800}

<i>TI</i>	<i>m</i>	$R_s$	GAMDSC	<i>mlGA</i>
<i>TI</i> <sup>16</sup>	100	400	85.8	<b>61.4</b>
<i>TI</i> <sup>17</sup>		500	87.5	<b>65.9</b>
<i>TI</i> <sup>18</sup>		600	87.9	<b>69.8</b>
<i>TI</i> <sup>19</sup>		700	86.7	<b>72.3</b>
<i>TI</i> <sup>20</sup>		800	89.3	<b>73.4</b>
<i>TI</i> <sup>21</sup>	125	400	88.48	<b>56.48</b>
<i>TI</i> <sup>22</sup>		500	86.48	<b>63.28</b>
<i>TI</i> <sup>23</sup>		600	86.08	<b>67.52</b>
<i>TI</i> <sup>24</sup>		700	87.52	<b>69.28</b>
<i>TI</i> <sup>25</sup>		800	88.72	<b>72.4</b>
<i>TI</i> <sup>26</sup>	150	400	89.6	<b>53.2</b>
<i>TI</i> <sup>27</sup>		500	88.0	<b>60.0</b>
<i>TI</i> <sup>28</sup>		600	87.8	<b>64.93</b>
<i>TI</i> <sup>29</sup>		700	85.6	<b>69.46</b>
<i>TI</i> <sup>30</sup>		800	86.6	<b>72.6</b>

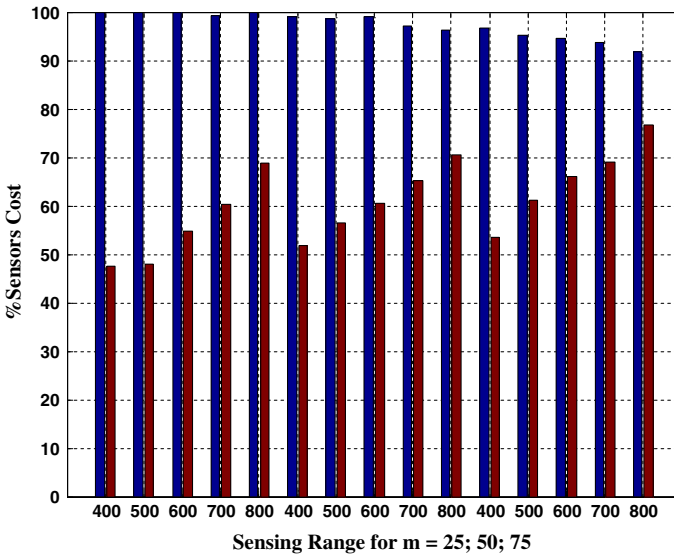
as less number of sensors as necessary to form as more as possible number of complete set covers.

#### 4.3 Impact of Post-heuristic Operator

The performance of the proposed *mlGA* is presented, here, *before* and *after* performing the post-heuristic operator with names *mlGA*<sup>-</sup> and *mlGA*<sup>+</sup>, respectively. Tables 7, 8 and 9

**Table 6** Comparison of sensors cost percentage for 10 WSNs in each test instance with *number of sensors* = {175, 200, 225} and  $R_s =$  {400, 500, ..., 800}

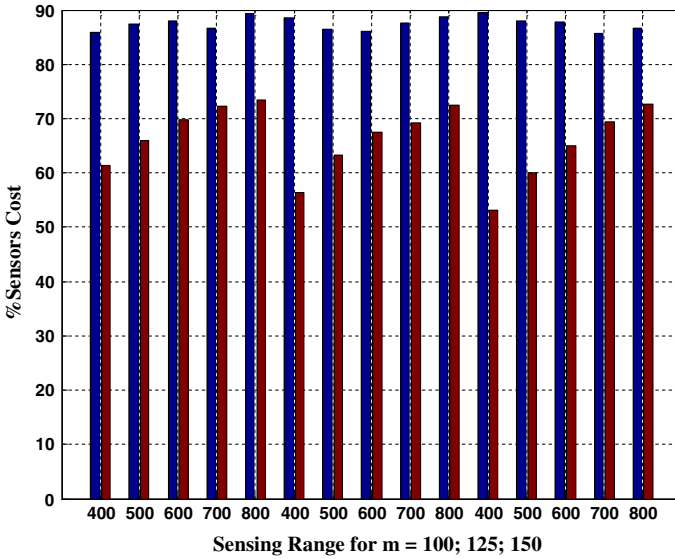
$TI$	$m$	$R_s$	GAMDSC	$mlGA$
$TI^{31}$	175	400	80.85	<b>63.02</b>
$TI^{32}$		500	80.17	<b>67.94</b>
$TI^{33}$		600	82.34	<b>71.65</b>
$TI^{34}$		700	83.88	<b>74.51</b>
$TI^{35}$		800	85.25	<b>76.85</b>
$TI^{36}$	200	400	85.75	<b>55.05</b>
$TI^{37}$		500	86.85	<b>57.6</b>
$TI^{38}$		600	87.55	<b>58.7</b>
$TI^{39}$		700	87.30	<b>62.35</b>
$TI^{40}$		800	88.50	<b>67.45</b>
$TI^{41}$	225	400	75.77	<b>63.64</b>
$TI^{42}$		500	80.00	<b>68.08</b>
$TI^{43}$		600	80.48	<b>71.73</b>
$TI^{44}$		700	81.42	<b>76.48</b>
$TI^{45}$		800	83.37	<b>77.02</b>



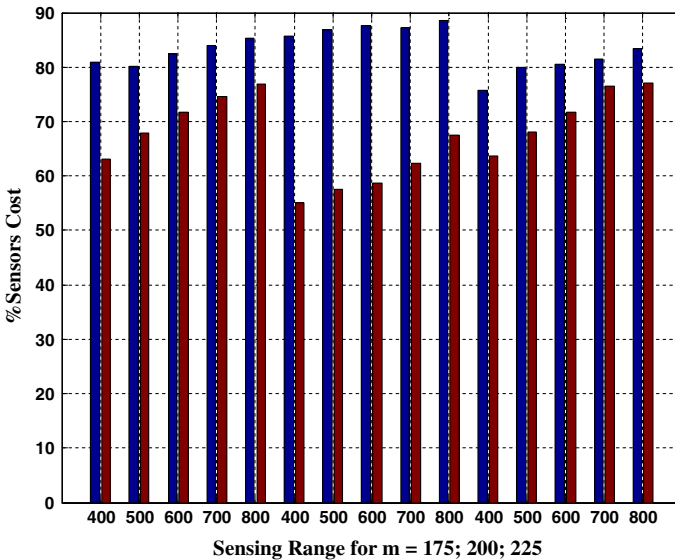
**Fig. 12** Sensors Cost percentage for the generated Set Covers for GAMDSC: *left* versus *mlGA*: *right* where *number of sensors*  $m = \{25, 50, 75\}$  and  $R_s = \{400, 500, \dots, 800\}$

present the average covers reliability  $r$  of the whole set covers solution of GAMDSC,  $mlGA^-$  and  $mlGA^+$ . Here,  $\bar{r}$  is presented as reference value. The solution providing the closest value to  $\bar{r}$  in each test instance has been bolded. Moreover, Figs. 15, 16 and 17 depict the sensors cost percentage  $SC\%$  being resulted from GAMDSC and  $mlGA^+$ .

$$r = \frac{\sum_{i=1}^l rDSC_i}{l} \tag{18}$$



**Fig. 13** Sensors Cost percentage for the generated Set Covers for GAMDSC: left versus *m*/GA: right where number of sensors *m* = {100, 125, 150} and *R<sub>s</sub>* = {400, 500, . . . , 800}



**Fig. 14** Sensors Cost percentage for the generated Set Covers for GAMDSC: left versus *m*/GA: right where number of sensors *m* = {175, 200, 225} and *R<sub>s</sub>* = {400, 500, . . . , 800}

where *l* is the maximum number of set covers and the coverage reliability of the *i*th set cover, *rDSC<sub>i</sub>*, can be formulated as:

$$rDSC_i = \frac{\sum_{j=1}^n \arg \max_{s_k \in DSC_i} (Coverage(s_k, t_j))}{n} \tag{19}$$

**Table 7** Comparison of covers reliability for 10 WSNs in each test instance with *number of sensors* = {25, 50, 75} and  $R_s = \{400, 500, \dots, 800\}$

<i>TI</i>	$\bar{r}$	GAMDSC	<i>mlGA</i> <sup>-</sup>	<i>mlGA</i> <sup>+</sup>
<i>TI</i> <sup>1</sup>	0.3426	0.5995	0.4380	<b>0.6121</b>
<i>TI</i> <sup>2</sup>	0.4228	0.6869	0.5357	<b>0.7477</b>
<i>TI</i> <sup>3</sup>	0.4747	0.7320	0.6073	<b>0.7574</b>
<i>TI</i> <sup>4</sup>	0.5262	0.7312	0.6612	<b>0.7758</b>
<i>TI</i> <sup>5</sup>	0.5742	0.7578	0.7076	<b>0.7828</b>
<i>TI</i> <sup>6</sup>	0.3310	0.5358	0.4026	<b>0.5745</b>
<i>TI</i> <sup>7</sup>	0.4003	0.6096	0.4775	<b>0.6312</b>
<i>TI</i> <sup>8</sup>	0.4655	0.6794	0.5846	<b>0.7029</b>
<i>TI</i> <sup>9</sup>	0.5145	0.7045	0.6278	<b>0.7419</b>
<i>TI</i> <sup>10</sup>	0.5812	0.7642	0.7071	<b>0.7742</b>
<i>TI</i> <sup>11</sup>	0.3458	0.5202	0.3847	<b>0.5491</b>
<i>TI</i> <sup>12</sup>	0.4159	<b>0.6100</b>	0.4803	0.5986
<i>TI</i> <sup>13</sup>	0.4846	0.6711	0.5981	<b>0.6850</b>
<i>TI</i> <sup>14</sup>	0.5379	0.7283	0.6562	<b>0.7369</b>
<i>TI</i> <sup>15</sup>	0.5848	0.7536	0.7138	<b>0.7659</b>

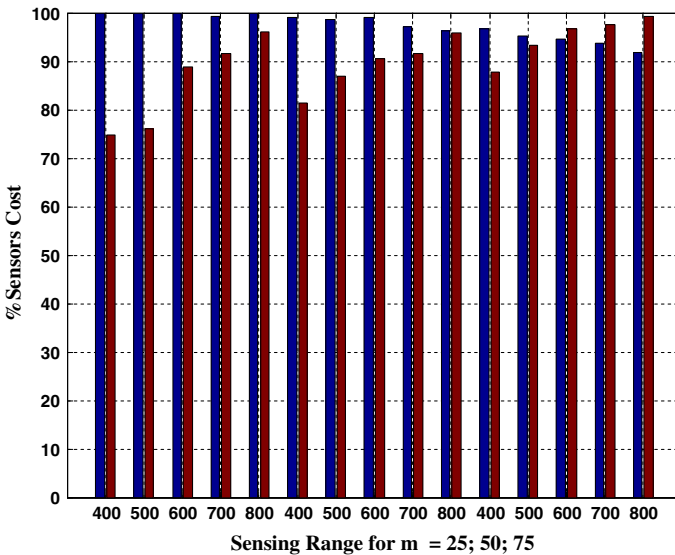
**Table 8** Comparison of covers reliability for 10 WSNs in each test instance with *number of sensors* = {100, 125, 150} and  $R_s = \{400, 500, \dots, 800\}$

<i>TI</i>	$\bar{r}$	GAMDSC	<i>mlGA</i> <sup>-</sup>	<i>mlGA</i> <sup>+</sup>
<i>TI</i> <sup>16</sup>	0.3538	0.5244	0.4298	<b>0.5369</b>
<i>TI</i> <sup>17</sup>	0.4306	<b>0.6219</b>	0.5047	0.6199
<i>TI</i> <sup>18</sup>	0.4886	<b>0.6694</b>	0.5777	0.6647
<i>TI</i> <sup>19</sup>	0.5367	0.7084	0.6540	<b>0.7195</b>
<i>TI</i> <sup>20</sup>	0.5762	0.7579	0.7006	<b>0.7593</b>
<i>TI</i> <sup>21</sup>	0.3575	<b>0.5590</b>	0.4028	0.5564
<i>TI</i> <sup>22</sup>	0.4261	<b>0.6382</b>	0.5127	0.6359
<i>TI</i> <sup>23</sup>	0.4832	0.6861	0.5936	<b>0.7011</b>
<i>TI</i> <sup>24</sup>	0.5321	0.7209	0.6416	<b>0.7355</b>
<i>TI</i> <sup>25</sup>	0.5761	<b>0.7683</b>	0.6967	0.7601
<i>TI</i> <sup>26</sup>	0.3525	0.5834	0.3975	<b>0.5905</b>
<i>TI</i> <sup>27</sup>	0.4257	0.6433	0.4911	<b>0.6569</b>
<i>TI</i> <sup>28</sup>	0.4808	0.6975	0.5698	<b>0.7035</b>
<i>TI</i> <sup>29</sup>	0.5349	0.7293	0.6360	<b>0.7336</b>
<i>TI</i> <sup>30</sup>	0.5765	<b>0.7660</b>	0.6862	0.7601

By comparing the results of *mlGA*<sup>-</sup> against *mlGA*<sup>+</sup> in Tables 7, 8 and 9, we can see that *mlGA*<sup>+</sup> can exploit the existence of the redundant sensors to improve its covers reliability. Also, the results verify that *mlGA*<sup>+</sup> can achieve covers reliability more or less than that achieved by GAMDSC. Again, this is expected as GAMDSC forms only a small number of complete set covers with many redundant sensors (as shown in Sects. 4.1, 4.2). This also demonstrates that there should be a tradeoff between the two contradictory criteria of getting maximum number of complete set covers and high covers reliability.

**Table 9** Comparison of covers reliability for 10 WSNs in each test instance with  $number\ of\ sensors = \{175, 200, 225\}$  and  $R_s = \{400, 500, \dots, 800\}$

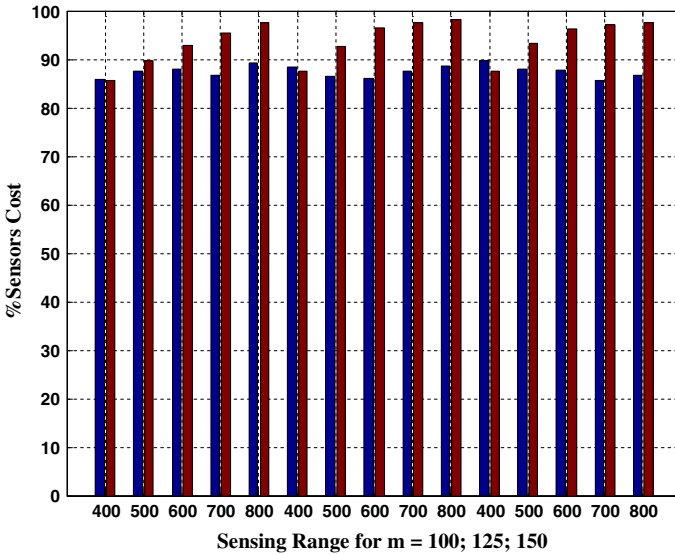
$TI$	$\bar{r}$	GAMDSC	$mlGA^-$	$mlGA^+$
$TI^{31}$	0.3538	<b>0.5324</b>	0.4157	0.5150
$TI^{32}$	0.4306	<b>0.6081</b>	0.5120	0.5962
$TI^{33}$	0.4886	<b>0.6872</b>	0.5927	0.6663
$TI^{34}$	0.5367	<b>0.7260</b>	0.6525	0.7089
$TI^{35}$	0.5762	<b>0.7671</b>	0.7066	0.7538
$TI^{36}$	0.3575	0.5736	0.4293	<b>0.5766</b>
$TI^{37}$	0.4261	0.6471	0.5157	<b>0.6527</b>
$TI^{38}$	0.4832	0.7192	0.6011	<b>0.7310</b>
$TI^{39}$	0.5321	0.7553	0.6592	<b>0.7595</b>
$TI^{40}$	0.5761	<b>0.7908</b>	0.7132	0.7839
$TI^{41}$	0.3525	<b>0.5358</b>	0.4183	0.5055
$TI^{42}$	0.4257	<b>0.6211</b>	0.5091	0.5860
$TI^{43}$	0.4808	<b>0.6827</b>	0.5928	0.6614
$TI^{44}$	0.5349	<b>0.7268</b>	0.6536	0.7046
$TI^{45}$	0.5765	<b>0.7667</b>	0.7154	0.7608



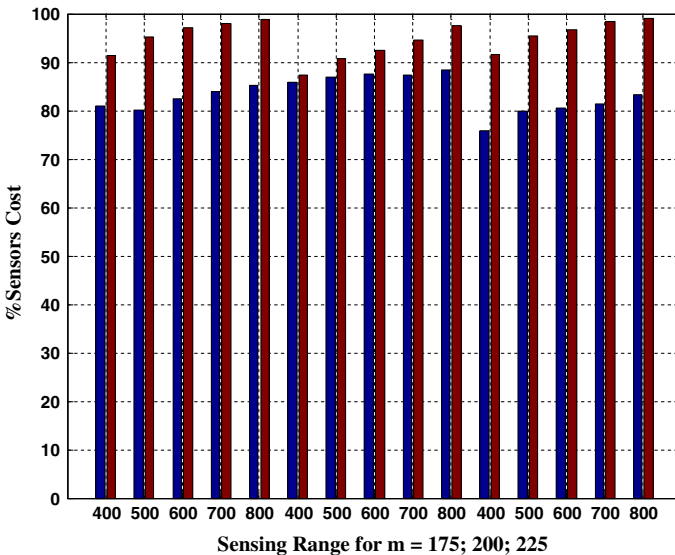
**Fig. 15** Sensors Cost percentage for the generated Set Covers for GAMDISC: *left* versus  $mlGA^+$ : *right* where  $number\ of\ sensors\ m = \{25, 50, 75\}$  and  $R_s = \{400, 500, \dots, 800\}$

#### 4.4 Comparison of Worst Case Time Complexity

Here we will compare the worst-case computational complexity of GAMDISC algorithm and the proposed multi-layer GA solving Disjoint Reliable Sensor Covers Problem (DRSC). Without loss of generality, the computational complexity of any single-objective genetic



**Fig. 16** Sensors Cost percentage for the generated Set Covers for GAMDSC: *left* versus *mlGA+*: *right* where number of sensors  $m = \{100, 125, 150\}$  and  $R_s = \{400, 500, \dots, 800\}$



**Fig. 17** Sensors Cost percentage for the generated Set Covers for GAMDSC: *left* versus *mlGA+*: *right* where number of sensors  $m = \{175, 200, 225\}$  and  $R_s = \{400, 500, \dots, 800\}$

algorithms is  $O(\max_i \times K)$  where  $K$  is the size of solutions to be evolved, via certain evolution operators, for  $\max_i$  iterations.

Now, let us consider the computation time needed for the most critical parts of the above algorithms when applied for solving DRSC problem. It is stated in [15] that in each generation of GAMDSC, the fitness function is the most critical part which represents the computation time bottleneck. The evaluation of the fitness function is linearly related to the number of

sensors and the number of targets. Thus, the worst-case time complexity of GAMDSC is:

$$O(GAMDSC) = \max_i K \times |n| \times |m| \quad (20)$$

where  $|n|$  is the number of targets and  $|m|$  is the number of sensors. On the other hand, for the multi-layer GA, the proposed repair operator (Algorithm 1) costs the most computation time and in the worst-case time it linearly related to the maximum number of sensors and targets. Also, the multi-layer GA has to repeat its whole operations for at most  $ub$  layers, and in the worst-case time it linearly related to the upper bound of sensor covers  $ub$ . Thus, the worst-case time complexity of the proposed multi-layer GA can be formally described in Eq. 21. Shortly speaking,  $O(mlGA) = O(GAMDSC) \times |ub|$ .

$$O(mlGA) = \max_i K \times |n| \times |m| \times |ub| \quad (21)$$

## 5 Conclusions

In this paper, we have addressed WSNs lifetime extension problem as a maximum Disjoint Set Covers (DSC) problem while introducing the concept of probabilistic coverage as a more realistic coverage model for constructing the set covers. A multi-layer genetic algorithm is proposed to maximize the number of set covers, where the total number of sensors used in each set cover is to be minimized. The result of multi-layer GA is then forwarded to a post-heuristic operator to improve the coverage reliability of each cover set. The performance of the proposed genetic algorithm is investigated in this paper under different simulation setting. The results of the simulations reveal that the proposed genetic algorithm outperforms GAMDSC in terms of number of disjoint set covers, covers reliability and sensors cost. The results of this paper currently motivate us to investigate the possibility of applying multi-objective evolutionary algorithms (like MOEA/D, NSGA-II, and MOPSO [22–25]) to the combined optimization problem of both disjoint set covers and covers reliability and handling both objective functions simultaneously instead of applying consecutive optimization mechanisms. Also, as a scope of further work, another quality of service (QoS) merit could be used to constraint the defined DRSC problem. For example, different targets may need different priority of sensing quality and thus the optimization problem should reflect this diversified QoS coverage constraint in its formulation.

**Acknowledgments** The authors gratefully acknowledge support from the Deanship of Scientific Research at University of Bahrain Grant 6/2013.

## References

1. Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability. A guide to the theory of NP-completeness*. New York: Freeman.
2. Slijepcevic, S., & Potkonjak, M. (2001). Power efficient organization of wireless sensor networks. In *Proceedings of the IEEE international conference communications* (Vol. 2, pp. 472–476). Finland.
3. Cardei, M., Thai, M., Li, Y., & Wu, W. (2005). Energy-efficient target coverage in wireless sensor networks. *IEEE INFOCOM 2005*, 1976–1984.
4. Tian, D., & Georganas, N. D. (2002). A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st association computing machinery international workshop wireless sensor network applications* (pp. 32–41).
5. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., & Gill, C. (2003). Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference embedded network sensor systems* (pp. 28–39) Los Angeles, CA.

6. Liu, Y., & Liang, W. (2005). Approximate coverage in wireless sensor networks. In *Proceedings of the IEEE conference local computation network 30th anniversary* (pp. 68–75).
7. Gupta, H., Zhou, Z., Das, S. R., & Gu, Q. (2006). Connected sensor cover: Self-organization of sensor networks for efficient query execution. *IEEE/Association Computing Machinery Transactions on Networking*, 14(1), 55–67.
8. Gupta, H., Zhou, Z., Das, S. R., & Gu, Q. (2006). Connected sensor cover: Self-organization of sensor networks for efficient query execution. *IEEE/Association Computing Machinery Transactions on Networking*, 14(1), 55–67.
9. Funke, S., Kesselman, A., Kuhn, F., Lotker, Z., & Segal, M. (2007). Improved approximation algorithms for connected sensor cover. *Wireless Networks*, 13(2), 153–164.
10. Zhou, Z., Das, S.R., & Gupta, H. (2009). Variable radii connected sensor cover in sensor networks. *IEEE/Association Computing Machinery Transactions on Networking*, 5(1), 1–36. Article 8.
11. Martins, F. V. C., Carrano, E. G., Wanner, E. F., Takahashi, R. H. C., & Mateus, G. R. (2009). A dynamic multiobjective hybrid approach for designing wireless sensor networks. In *Proceedings of the IEEE congress on evolutionary computation* (pp. 1145–1152).
12. Cardei, M., & Du, D.-Z. (2005). Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3), 333–340.
13. Cardei, I., & Cardei, M. (2008). Energy-efficient connected-coverage in wireless sensor networks. *International Journal of Sensor Networks*, 3(3), 201–210.
14. He, J., Xiong, N., Xiao, Y., & Pan, Y. (2010). A reliable energy efficient algorithm for target coverage in wireless sensor networks. *ICDCSW '10 Proceedings of the 2010 IEEE 30th international conference on distributed computing systems workshops* (pp. 180–188).
15. Lai, C., Ting, C., & Ko, R. (2007). An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In *Proceedings of the 2007 congress on evolutionary computation* (pp. 3531–3538).
16. Hu, X.-M., Zhang, J., Yu, Y., Chung, H. S.-H., Li, Y.-L., Shi, Y.-H., et al. (2010). Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. *IEEE Transactions on Evolutionary Computation*, 14(5), 766–781.
17. Gil, J.-M., & Han, Y.-H. (2011). A target coverage scheduling scheme based on genetic algorithms in directional sensor networks. *Sensors*, 11(2), 1888–1906. doi:10.3390/s110201888.
18. Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3), 249–265.
19. Ghosh, A., & Das, S. K. (2006). Coverage and connectivity issues in wireless sensor networks, Chap. 6. In R. Shorey, A. L. Ananda, M. C. Chan, & W. T. Ooi (Eds.), *Mobile, wireless, and sensor networks: Technology, applications, and future directions*. New York: Wiley.
20. Hossain, A., Biswas, P. K., & Chakrabarti, S. (2008). Sensing models and its impact on network coverage in wireless sensor network. *IEEE Region 10 Colloquium and the Third ICIS*. Kharagpur, India, Dec. 8–10.
21. Huang, C.-F., & Tseng, Y.-C. (2005). The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4), 519–528.
22. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.
23. Coello Coello, C. A., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. New York: Kluwer.
24. Zhang, Q., & Li, H. (2007). MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731.
25. Coello Coello, C. A., Toscano-Pulido, G., & Salazar-Lechunga, M. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 256–279.





**Mayyadah F. Abdulhalim** received her B.Sc. and M.Sc. degrees in computer science from University of Baghdad, Iraq in 1989 and 1996, respectively, and her Ph.D. degree in computer science from University of Technology, Iraq in 2002. She was working at University of Baghdad, College of Science, Department of Computer Science between 1989 and 2005 then she joined University of Bahrain, College of Information technology, Department of Computer Science in 2005 till now. She is associate professor and she has more than 25 publications. Her main research interests are computational intelligence, evolutionary algorithms, applications of bio-inspired algorithms in wireless sensor networks and e-learning management systems. Dr. Mayyadah Alani join different committees at University, College and Department levels. She is a member in scientific research committees at University, College and Department levels. In addition, she is a member in Department ABET committee.



**Bara'a A. Attea** received the BS and MS degrees in computer science from University of Baghdad, Baghdad, Iraq in 1993 and 1996, respectively, and the Ph.D. degree in computer science from University of Technology, Baghdad in 2002. She is an associate professor in Computer Science Department—University of Baghdad. She was a visiting researcher in Gazi University, Ankara, Turkey from 2011 to 2013. Her main research interests are computational intelligence, evolutionary algorithms, data mining, and applications of bio-inspired algorithms in wireless sensor networks.