Heuristic Search Algorithm (HSA) for Enhancing the Lifetime of Wireless Sensor Networks

Tripatjot S. Panag, J. S. Dhillon

***Abstract*—**The lifetime of a wireless sensor network can be effectively increased by using scheduling operations. Once the sensors are randomly deployed, the task at hand is to find the largest number of disjoint sets of sensors such that every sensor set provides complete coverage of the target area. At any instant, only one of these disjoint sets is switched on, while all other are switched off. This paper proposes a heuristic search method to find the maximum number of disjoint sets that completely cover the region. A population of randomly initialized members is made to explore the solution space. A set of heuristics has been applied to guide the members to a possible solution in their neighborhood. The heuristics escalate the convergence of the algorithm. The best solution explored by the population is recorded and is continuously updated. The proposed algorithm has been tested for applications which require sensing of multiple target points, referred to as point coverage applications. Results show that the proposed algorithm outclasses the existing algorithms. It always finds the optimum solution, and that too by making fewer number of fitness function evaluations than the existing approaches.

***Keywords*—**Coverage, disjoint sets, heuristic, lifetime, scheduling, wireless sensor networks, WSN.

I. Introduction

A Wireless Sensor Network (WSN) consists of a large number of densely deployed sensors which work collectively to provide information about the target field. The applications of wireless sensor networks include monitoring the battlefields, exploring the deep oceans, tracking the animals in their natural habitats etc. [1]-[8]. Most of these applications require the sensor nodes to be placed in hostile environments, making the replacement of their batteries difficult or practically impossible. The lifetime of a WSN with limited energy resources can be increased if these resources are used in a controlled and intelligent way. This is an important topic of research [1], [2], [8]-[11].

The first step in setting up of a WSN is the deployment of sensor nodes. Many methods [12]-[17] have been proposed to deploy the sensors in a planned manner in order to achieve complete coverage of the monitored area with minimum number of sensors. It has also been proposed to dispatch a set of mobile sensor nodes to satisfy the coverage and connectivity requirements [7], [16], [18], [19]. For applications where the area to be monitored is too large or where the batteries of the sensor nodes cannot be replaced, random deployment is the preferred choice. It is achieved by spreading the sensors randomly all over the monitored area [9], [10], [20]. In random deployment, the number of sensors required to achieve complete coverage is more than the planned methods. As the technological advancements have considerably reduced the cost of sensor nodes [1], [8], [10], random deployment has become economically feasible. Practically, the number of sensor nodes deployed is much more than the number required for achieving complete coverage. This adds redundancy. Redundant deployment helps in prolonging the lifetime of a WSN. Many methods [21]-[29] have been proposed to intelligently use this redundancy for setting up of energy efficient wireless sensor networks. Redundant deployment also allows identification of redundant backbones for data communication tasks and helps setting up of minimum k-connected, m-dominating set wireless sensor networks [30].

With redundant deployment, more sensors than one monitor any portion of the area under surveillance. This implies that the region can be monitored by switching on any one of them and switching off the rest. Thus only a subset of the deployed sensors needs to be active to completely monitor any area [8]-[11], [31]-[35] at a time. Therefore, the sensors are divided into disjoint sets, each providing a complete coverage of the monitored area. Many methods [2], [8]-[11], [31], [34]-[45] have been proposed to divide the sensors into such mutually exclusive sets. Instead of using any one subset till its energy exhausts and then switching on to the other, it is better to use them in a cyclic manner in order to increase the lifetime of the wireless sensor network. This follows from the fact that the battery life doubles, if it is used in short bursts separated by significant off time than in a continuous mode of operation [46]. Using various disjoint sets one after the other in a cyclic manner is referred to as scheduling. For maximizing the lifetime of a WSN, scheduling requires that the sensors be divided into the maximum number of sets subject to the condition that each of these sets provides complete coverage of the target area.

Many methods [9], [10], [31], [47] have been proposed for scheduling of sensors in a WSN. Cardie and Du [10] proposed a "maximum covers using mixed integer programming (MC-MIP)" algorithm. It is based on a branch and bound method to make an implicit exhaustive search and guarantees to find an optimal solution to the problem. However, the execution time of MC-MIP increases exponentially as the number of sensors and targets becomes larger [9]. Another heuristic, named

Tripatjot Singh Panag is an assistant professor in the Electronics & Communication Engineering Department Baba Banda Singh Bhadur Engineering College, Fatehgarh Sahib-140407, Punjab, India (phone: +91-9855764499; e-mail: panagtripat@gmail.com)

J. S. Dhillon is a professor in the Electrical & Instrumentation Engineering Department, Sant Longowal Institute of Engineering & Technology, Longowal-148106, Sangrur, Punjab, India (e-mail: jsdhillonp@yahoo.com)

"most constrained minimally constraining covering (MCMCC)" heuristic has been proposed by [31]. It uses a greedy deterministic approach. Hu et al. [9] suggest that it works much faster than MC-MIP for large scale problems, but it does not guarantee to find an optimum solution.

Genetic algorithms (GAs) [48] are population based search algorithms that simulate biological evolution process and have successfully been used to solve a wide range of NP-hard optimization problems [49]-[53]. Lai et al. [47] proposed a GA named "genetic algorithm for maximum disjoint set covers (GAMDSC)". It encodes each gene in the chromosome as an integer index of the set that the sensor has joined. GAMDSC is suitable only when the number of targets and sensors is small. Hu et al. [9] proposed a "schedule transition hybrid genetic algorithm (STHGA)" that uses a combination of genetic algorithm and schedule transition operations to address the problem of finding the largest number of disjoint complete sets. STHGA starts by placing all the sensors in a single set so that it is definitely a complete set. It then uses a forward encoding scheme to move the redundant sensors of the first set to the second set. STHGA clubs forward encoding with genetic algorithm and sensor schedule transition operations to complete the second set. It then moves the redundant sensors of these two sets to the third set. The process continues until the maximum number of disjoint complete cover sets has been created. Hu et al. [9] have compared the performance of STHGA with GAMDSC and MCMCC. It has been highlighted that STHGA always achieves the desired results whereas GAMDSC and MCMCC do not always do so. STHGA has also been reported to be faster than the other two. The number of fitness function evaluations made by STHGA is always considerably less than GAMDSC.

In this paper, a heuristic search algorithm (HSA) is proposed to solve the disjoint set covers problem for maximizing the lifetime of a wireless sensor network. The algorithm uses a group of members to explore the entire search space in order to find an optimum solution. An optimum solution is the one which divides the sensors into maximum number of disjoint complete cover sets. The number of dimensions of the search space is equal to the number of sensors deployed. Each dimension in the search space corresponds to a sensor and its value indicates the set in which the sensor is scheduled to be switched on. Thus each dimension of a member can have an integer value in the range from 1 to the maximum number of possible disjoint complete sets (determined by the number of sensors covering a critical field as discussed later).

The HSA starts by randomly initializing all the members. Then the coverage of all the cover sets in each of the members is evaluated. This is followed by fitness function evaluation of each member. The fitness function has been designed to give more weight to the complete sets so that a member having more number of complete sets MUST have higher fitness. Then three moves are applied heuristically one after the other. These moves have been designed to move the redundant sensors from one set to another. The selection of sensors for movement and the determination of new sets is done randomly to explore the entire space. The directed moves aim at improving the convergence of the algorithm. The performance of the proposed HSA has been compared with STHGA [9]. Results show that HSA always finds the optimum solution like STHGA, but at a much faster optimization speed.

II. Problem Definition

Let N sensors be randomly deployed in a rectangular region $A \times B$ (Fig. 1). Each sensor is assumed to have a circular coverage area with radius R. Each sensor has a finite lifetime. The aim is to maximize the time for which the area is monitored. Let the number of sensors deployed be too large than the number of sensors required to achieve complete coverage of the area. Thus only a subset of the sensors deployed can provide complete coverage.

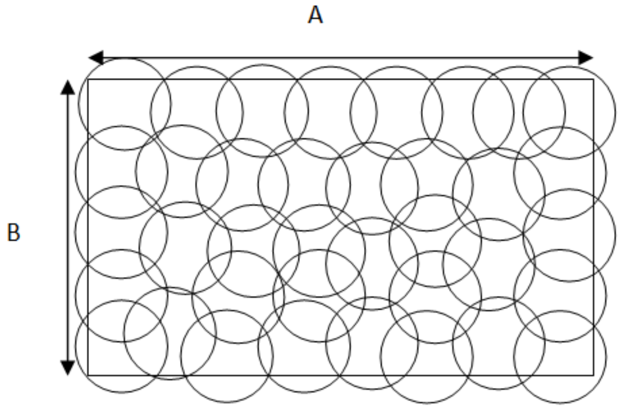

Fig. 1 N sensors with circular coverage deployed in A × B rectangular area

If the deployed sensors are represented by a set $\boldsymbol{S} = \{s_1, s_2, \ldots\ldots, s_N\}$, and the maximum number of disjoint sets that can be created by β, the objective is to find the maximum number of subsets $\boldsymbol{S_k}$, $k = 1, 2, \ldots, \beta$, of $\boldsymbol{S}$ so that each subset provides complete coverage of the target area. Further, $\boldsymbol{S_k} \cap \boldsymbol{S_l} = \varnothing$; $k \neq l$ and $k, l = 1, 2, \ldots., \beta$.

Once the sensors are divided into subsets $\boldsymbol{S_k}$, $k = 1, 2, \ldots., \beta$, as per the criteria given above, sensors in only one of the subsets shall be activated at any instant of time. After some time that subset of sensors shall be switched off and another subset shall be switched on. Thus sensors in a sensor subset $\boldsymbol{S}_k$ shall be activated in the k^{th} schedule.

In the illustration of Fig. 1, each sensor s_i has been assumed to have a circular coverage area of radius R. In practice, however, the sensor coverage area can be of any shape. In this work, for simplicity the coverage area has been assumed to be circular.

Field: Referring to Figs. 1 and 2 it can be observed that the areas covered by sensors overlap each other, forming separate fields [32]. A field can thus be defined as the area that is covered by the same set of sensors e.g. in Fig. 2 three sensors combine together to form seven fields.

TABLE I
NOTATIONS

s_k	Sensor k
f_i	Fitness value of member i, $i = 1, 2, ..., M$
f_{best}	Fitness value of the best member
A	Width of the target area
B	Length of the target area
N	Number of sensors
R	Sensing range of each sensor
M	Number of members in a population
K_1	Number of sensors selected during the redundant move
K_2	Number of sensors selected during the completion move
NF	Total number of fields in the region that are to be covered
NCF	Number of critical fields i.e. the number of fields that are covered by β sensors
$MAXt$	Maximum number of iterations
C_i	Number of complete sets in member i, $i = 1, 2, ..., M$
C_{best}	Number of complete sets in the best member
$V_{i,y}$	Coverage percentage of y^{th} incomplete set in member i
$I(i,j)$ $i = 1,2,...,M$ $j=1,2,...,N$	Gives the location of the i^{th} member in the N dimensional search space. Each dimension represents a sensor and its value (in the range 1 to β) indicates the set in which the sensor is scheduled to be switched on
$\boldsymbol{S}$	Set of sensors deployed in the target area
$\boldsymbol{S_k}$ $k = 1, 2, ..., \beta$	Subset of $\boldsymbol{S}$
β	Number of sensors covering a critical field

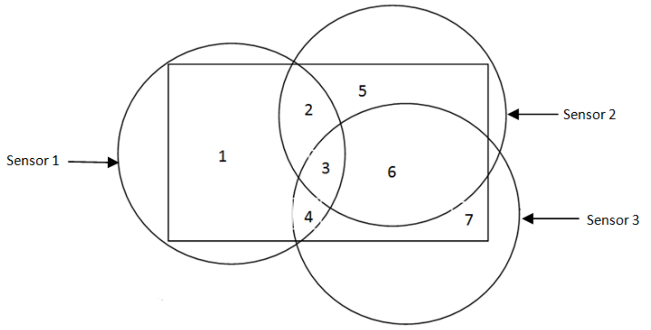

Fig. 2 Circular coverage areas of 3 sensors combine together to form 7 fields

Critical Field: A field that is covered by the minimum number of sensors is referred to as the critical field and the sensors covering that critical field are called critical sensors.

If a critical field is covered by β sensors, then at least one of these β sensors must be active at any instant of time to provide complete coverage of the target area. Thus we can have at most β number of disjoint sets of sensors which provide complete coverage to the target area.

Let N sensors combine together to form NF fields. Let F_k, k=1, 2, …, NF represent the number of sensors that cover the k^{th} field, then

$$\beta = \text{min. } \{F_k; k = 1, 2, \ldots\ldots, NF\} \quad (1)$$

In Fig. 2 the values of F_1, F_2, …, F_7 respectively are 1, 2, 3, 2, 1, 2 and 1. Thus the value of β is 1.

III. Proposed Directed Random Search Algorithm

The algorithm being proposed in this paper solves the problem at hand by using random search clubbed with some directive moves. A complete flow chart of the proposed algorithm is shown in Fig. 3. The algorithm uses M members to explore an N-dimensional space, where N is the number of sensors deployed. In this section, first the representation and initialization of the members is discussed, followed by the calculation of coverage percentage of a set, evaluation of the population and the directive moves.

A. Representation of Members

The initial population comprises of M members, each having N dimensions. Each member is a candidate solution for the problem. Every dimension of a member corresponds to a sensor. A member is represented as:

$$I(i) = [\alpha_{i1}, \alpha_{i2}, ..., \alpha_{In}]^T \; ; \; (i = 1, 2, \ldots, M) \quad (2)$$

where $\alpha_{ij} \in \{1, 2, \ldots, \beta\}$; $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, N$. The value α_{ij} indicates the set or the schedule in which the j^{th} sensor shall be switched on in member i. Thus each member divides the N sensors into β disjoint sets.

B. Initialization

As the critical field is covered by β sensors, the sensors may at the most be divided into β disjoint sets so that each of the sets completely covers all the fields. The algorithm starts by randomly placing the sensors in a member into any of the β sets as:

$$I(i,j) = \text{roundoff}\,(1 + (\beta - 1) \times \text{uniran}()\,) \; ; \quad (i = 1, 2, \ldots, M \; ; j = 1, 2, \ldots, N) \quad (3)$$

uniran() is a function which returns a uniform random number in the range [0-1]. Thus, the sensors are divided into β disjoint sets.

C. Fitness Evaluation

The percentage coverage of each of the sets is to be evaluated which in turn requires the calculation of the percentage coverage of each sensor. As the determination of exact percentage coverage of the sensors is difficult, an approximation is made by dividing the area into grids [9]. A set having hundred percent coverage is termed as a complete set. The aim of the algorithm is to make all the sets complete i.e. to divide the sensors into β disjoint complete cover sets.

The fitness of the i^{th} member in the population is evaluated as

$$f_i = C_i + \sum_{y=1}^{(\beta - C_i)} (0.1 \times V_{i,y}) \quad (4)$$

where C_i is the number of complete sets in member i, (β-C_i) the number of incomplete sets in member i, and $V_{i,y}$ the coverage percentage of the y^{th} incomplete set in member i.

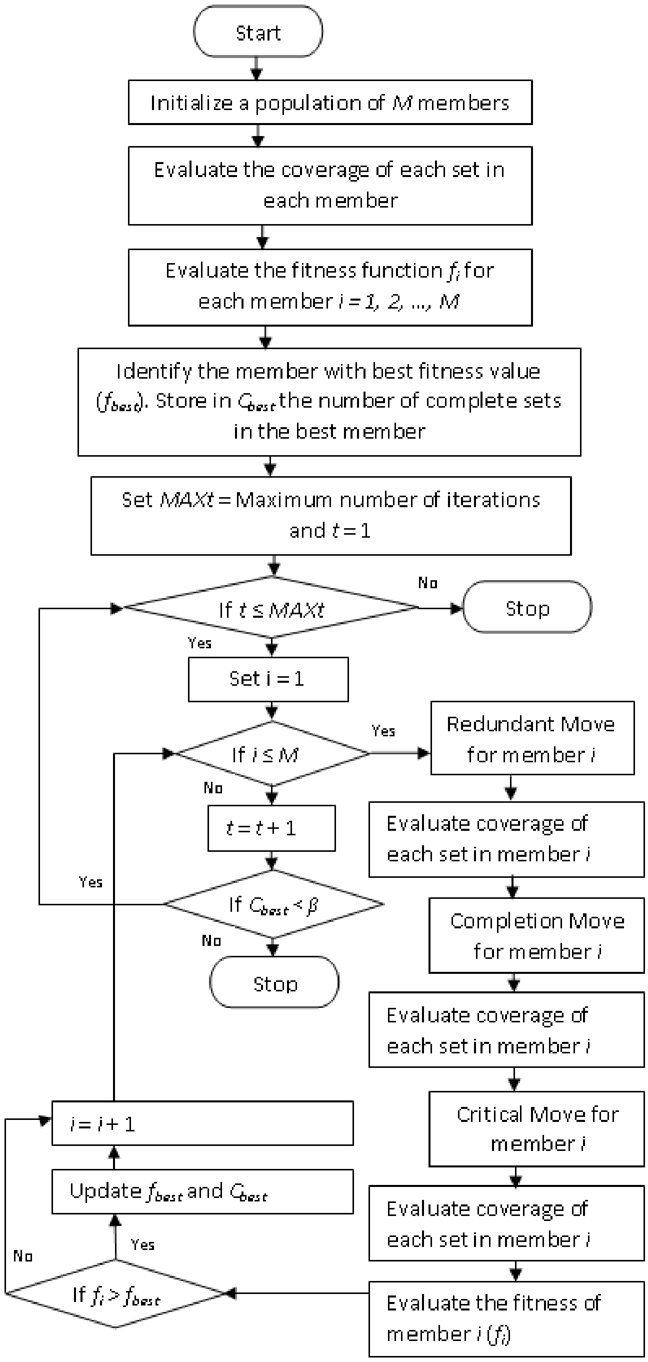
Start
Initialize a population of M members
Evaluate the coverage of each set in each member
Evaluate the fitness function f_i for each member $i = 1, 2, ..., M$
Identify the member with best fitness value (f_{best}). Store in C_{best} the number of complete sets in the best member
Set $MAXt$ = Maximum number of iterations and $t = 1$
If $t \leq MAXt$
No
Stop
Yes
Set i = 1
If $i \leq M$
Yes
Redundant Move for member i
Evaluate coverage of each set in member i
Completion Move for member i
Evaluate coverage of each set in member i
Critical Move for member i
Evaluate coverage of each set in member i
Evaluate the fitness of member i (f_i)
No
$t = t + 1$
Yes
If $C_{best} < \beta$
No
Stop
$i = i + 1$
Update f_{best} and C_{best}
Yes
No
If $f_i > f_{best}$

Fig. 3 Flowchart of the proposed HAS

D. Directive Moves

1) Redundant Move

The fields are covered by multiple sensors. More than one sensor covering a field may have been scheduled in one set. All but one of these sensors are redundant as removing them from the set will not change the percentage coverage of the set. The purpose of this move is to change the schedule of such redundant sensors. In every iteration, the move is made K1 times for every member as: Randomly select a sensor. If the selected sensor is redundant to its current set, randomly reschedule it to any of the other sets.

The move makes K_1 efforts to reschedule the sensors. But this does not imply that K_1 sensors are rescheduled for every member in each of the iterations. The sensor may be rescheduled if and only if it is redundant to its current schedule. A pseudo-code for the redundant move is given in Fig. 4.

```
FOR ($k = 1, 2, \ldots, K_1$)
  $j$ = roundoff ($1 + (N - 1) \times$ uniran() )
  $T = I(i,j)$
  IF ($j$ is redundant to the set $T$)
    $T^{new}$ = roundoff ($1 + (\beta - 1) \times$ uniran() )
    $I(i,j) = T^{new}$
  END IF
END FOR
```

Fig. 4 Pseudo-code for redundant move

2) Completion Move

The completion move aims at improving the coverage percentage of incomplete sets to make them complete. It moves the redundant sensors from the complete sets to the incomplete sets. This move is also performed K2 times for every member in each iteration. The procedure for this move is explained below and a pseudo-code for the same is given in Fig. 5.

Randomly select a sensor. If the sensor belongs to a complete set, check whether it is redundant to its current set. If yes, reschedule it to a randomly selected incomplete set.

```
FOR ($k = 1, 2, \ldots, K_2$)
  $j$ = roundoff ($1 + (N - 1) \times$ uniran() )
  $T = I(i,j)$
  IF (the sensors of set $T$ provide complete coverage)
    IF ($j$ is redundant to the set $T$)
      $x$ = roundoff ($1 + ((\beta - C_i) - 1) \times$ uniran() )
      /* $(\beta - C_i) \rightarrow$ number of incomplete sets in member
      $i$ */
      $T^{new}$ = schedule of the $x^{th}$ incomplete set
      $I(i,j) = T^{new}$
    END IF
  END IF
END FOR
```

Fig. 5 Pseudo-code for completion move

Like in the redundant move, performing the completion move K_2 times does not mean that K_2 sensors will be displaced. Probability (the sensor will be rescheduled) = Probability ((the randomly selected sensor belongs to a complete set) *and* (the selected sensor is redundant to its current set)).

3) Critical Move

The upper limit on the number of disjoint complete sets is fixed by the number of sensors covering the critical fields. In order to classify the sensors into the maximum number of disjoint complete sets, the sensors covering the critical fields must be placed in different sets. In other words, no set can be complete if it does not contain a sensor for each of the critical fields. This move tries to displace the multiple critical sensors scheduled in a common set to those incomplete sets which do not cover the corresponding critical field. The critical move is performed for all the incomplete sets of every member in each iteration as given below:

For each critical field, check whether it has been covered in the incomplete set. If not, randomly select a sensor from among the sensors covering that critical field. Check whether the selected sensor is redundant to its current set. If yes, reschedule it to the current incomplete set.

Fig. 6 presents the pseudo-code for the critical move.

```
FOR ($k$ = 1, 2, …, ($\beta$ – $C_i$))   /* for every incomplete set in member $I$ */
  $T$ = schedule of the $k^{th}$ incomplete set
  FOR ($l$ = 1, 2, …, $NCF$) ;  /* for every critical field */
    IF(the $l^{th}$ critical field is not covered in schedule $T$)
      Randomly select a sensor $j$ from among the sensors covering the $l^{th}$ critical field
      IF(sensor $j$ is redundant to its current schedule in the member $i$)
        $I(i,j) = T$    /* reschedule the sensor $j$ to the complete set $T$} */
      END IF
    END IF
  END FOR
END FOR
```

Fig. 6 Pseudo-code for critical move

After performing the heuristics, the fitness of the new population is evaluated using the fitness function given by (4). The best-so-far member is then updated. The HSA terminates either when β disjoint complete cover sets are found or when the predefined maximum number of iterations is reached. If none of the termination conditions is met, the HSA again starts with the redundant move for the new population and the process is repeated.

With M members in the population, the number of coverage evaluations as well as the fitness function evaluations made during initialization is M. Further, in every iteration, coverage evaluation is made three times for every member and the fitness function evaluation is done once for every member. Thus for t iterations, the number of coverage evaluations made shall be $M(1+3t)$ and the number of fitness function evaluations shall be $M(1+t)$.

IV. Results and Discussion

The proposed HSA has been tested with different sensor deployments for point coverage problems. The performance of the HSA has been compared with STHGA [9]. This section presents the analysis and discussion of the results.

The population size for all the experiments has been fixed at $M = 5$. The parameters K_1 and K_2 used in the heuristics are set as $K_1 = K_2 = (N / 3)$ where N is the number of sensors deployed. These parameters have been set empirically. Each case been tested 100 times for the HSA. The parameter settings and other details regarding STHGA can be referred in [9]. The monitored area is a 50 × 50 unit square. The sensor node location coordinates have been randomly generated as float point values in [1], [51]. All the simulations have been carried out on a computer having Intel® Core™ i3 CPU @2.4GHz with 4GB RAM.

Experiments have been conducted for different cases with different numbers N of the sensors deployed. The number of targets and the sensing range R of the sensors has been fixed at 10 and 22 respectively. Table II tabulates the results computed by HSA and compares it with the results for STHGA documented by Hu et al. [9]. To facilitate the comparison, the results for the same values of N, R and β have been presented as [9]. Both STHGA and HSA create the maximum number of disjoint sets in all the cases. Similar to STHGA [9], HSA also finds the optima in all the 100 independent runs for each case. The results listed in the table are the mean values. However, HSA reaches the optima much faster than STHGA in all the cases. This is reflected by the considerably smaller average number of fitness function evaluations made by HSA in comparison with STHGA for all the seven cases. Fig. 7 graphically shows the variation in number of fitness function evaluations by both STHGA and HSA for the seven cases listed in Table II. It can be noted that both STHGA and HSA show similar variation trends. However, the number of fitness evaluations for HSA is smaller than STHGA in all the cases.

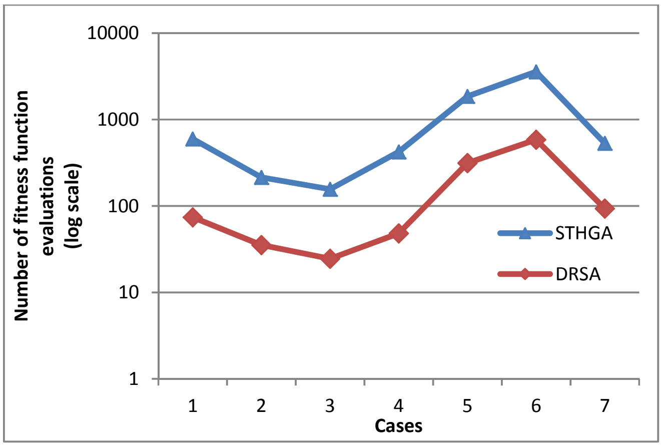

Fig. 7 Comparison of number of fitness function evaluations for STHGA and HSA

TABLE II
TEST RESULTS FOR POINT COVERAGE WITH DIFFERENT VALUES OF N

Cases			STHGA			HSA				
Sr.	N	R	β	Nsets[@]	avgFE[*]	β	Nsets[@]	Number of Iterations	avgFE[*]	avgCE[**]
1	90	22	30	30	596	30	30	13.77	73.83	211.50
2	100	22	23	23	214	23	23	6.07	35.33	96.00
3	110	22	21	21	156	21	21	3.90	24.50	63.50
4	120	22	35	35	422	35	35	8.67	48.33	135.00
5	130	22	41	41	1856	41	41	61.80	314.00	932.00
6	140	22	44	44	3568	44	44	115.97	584.83	1744.50
7	150	22	42	42	532	42	42	17.67	93.33	270.00

* avgFE = average of number of fitness function evaluations
** avgCE = average of number of coverage evaluations
@ Nsets = Average of number of disjoint complete sets created

V. CONCLUSIONS AND FUTURE SCOPE

In this paper a directed random search algorithm has been presented to find the maximum number of disjoint complete cover sets of sensors for point coverage in order to maximize the lifetime of a WSN. HSA uses a population to explore the entire solution space in search of the optimum solution. The members are initially randomly scattered all over the search space. Each member explores the search space in its neighborhood and reports the best position acquired by it. Keeping in view, the requirements of the application for which the search algorithm is designed, a set of heuristics has been introduced. These heuristics speed up the convergence of the algorithm towards the optima. The proposed algorithm achieves high quality solutions at fast optimization speeds and outperforms the STHGA which has been proven to be the most suitable algorithm for this problem [9].

The results show that HSA always finds the optima and that too at a high pace. Further research needs to be carried out to analyze the effect of variation in the population size and the number of times the redundant and completion moves are made. Also the suitability of HSA to increase the lifetime of WSNs in area coverage problems needs to be explored.

REFERENCES

[1] Zheng, J., and Jamalipour, A. (2009). Wireless sensor networks – a networking perspective. John Wiley & Sons, ISBN: 978-0-470-16763-2.

[2] Anastasi, G., Conti, M., Francesco, M. Di, and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. Ad Hoc Networks, 7(3), 537-568.

[3] Younis, M., and Akkaya, K. (2008). Strategies and techniques for node placement in wireless sensor networks: A survey. Ad Hoc Networks, 6(4), 621-655.

[4] Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. Computer Networks, 52(12), 2292-2330.

[5] Akyildiz, I. F., Melodia, T., and Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. Computer Networks, 51(4), 921-960.

[6] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. IEEE Communications Magazine, 40(8), 102-114.

[7] Wand, X. and Wang, S. (2011). Hierarchical deployment optimization for wireless sensor networks. IEEE Transactions on Mobile Computing, 10(7), 1028-1041.

[8] Lai, C.-C., Ting, C.-K., and Ko, R.-S. (2007). An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In proceedings of IEEE Congress on Evolutionary Computation, pp. 3531-3538, September 25 – 28.

[9] Hu, X.-M., Zhang, J., Yu, Y., Chung, H. S.-H., Li, Y.-L., Shi, Y.-H., and Luo, X.-N. (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.

[10] Cardei, M., and Du, D.-Z. (2005). Improving wireless sensor network lifetime **through** power aware organization. Wireless Networks, 11(3), 333-340.

[11] Gaur, B., and Kumar, P. (2013). Wireless sensor deployment using modified discrete binary PSO method. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 1(3), 82-89.

[12] Lin, F. Y. S., and Chiu, P. L. (2005). A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks. IEEE Communications Letters, 9(1), 43-45.

[13] Huang, C.-F., and Tseng, Y.-C. (2005). The coverage problem in a wireless sensor network. Mobile Networks and Applications, 10(4), 519-528.

[14] Chakrabarty, K., Iyengar, S. S., Qi H., and Cho, E. (2002). Grid coverage for surveillance and target location in distributed sensor networks. IEEE Transactions on Computers, 51(12), 1448-1453.

[15] Dhillon, S. S., and Chakrabarty, K. (2003). Sensor placement for effective coverage and surveillance in distributed sensor networks. In proceedings of IEEE Wireless Communications and Networking Conference, WCNC 2013, LA USA, vol. 3, pp. 1609-1614, March 20.

[16] Wang, Y.-C., Hu, C.-C., and Tseng, Y.-C. (2008). Efficient placement and dispatch of sensors in a wireless sensor network. IEEE Transactions on Mobile Computing, 7(2), 262-274.

[17] Khanjary, M., Sabaei, M., and Meybodi, M. R. (2014). Critical density for coverage and connectivity in two-dimensional aligned-orientation directional sensor networks using continuum percolation. IEEE Sensors Journal, 14(8), 2856-2863.

[18] Howard, A., Matarić, M. J., and Sukhatme, G. S. (2002). An incremental self deployment algorithm for mobile sensor networks. Autonomous Robots, 13(2), 113-126.

[19] Heo, N., and Varshney, P. K. (2005). Energy-efficient deployment of intelligent mobile sensor networks. IEEE Transactions on Systems, Man, Cybernetics, Part A: Systems and Humans, 35(1), 78–92.

[20] Kulkarni, R.V., and Venayagamoorthy, G.K. (2010). Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40(6), 663-675.

[21] Shwe, H. Y., Jiang, X.-H., and Horiguchi, S. (2009). Energy saving in wireless sensor networks. Journal of Communication and Computer, 6(5), 20-28.

[22] Chang, C.-Y., and Chang, H.-R. (2008). Energy-aware node placement, topology control and MAC scheduling for wireless sensor networks. Computer Networks, 52(11), 2189-2204.

[23] Leung, H., Chandana, S., and Wei, S. (2008). Distributed sensing based on intelligent sensor networks. IEEE Circuits and Systems Magazine, 8(2), 38-52.

[24] Iyengar, S. S., Wu, H.-C., Balakrishnan, N., and Chang, S. Y. (2007). Biologically inspired cooperative routing for wireless mobile sensor networks, IEEE Systems Journal, 1(1), 29-37.

[25] Cui, S., Madan, R., Goldsmith, A. J., and Lall, S. (2007). Cross-layer energy and delay optimization in small-scale sensor networks. IEEE Transactions on Wireless Communication, 6(10), 3688-3699.

[26] Yu, Y., Prasanna, V. K., and Krishnamachari, B. (2006). Energy minimization for real-time data gathering in wireless sensor networks. IEEE Transactions on Wireless Communication, 5(11), 3087-3096.

[27] Cardei, M., and Wu, J. (2006). Energy-efficient coverage problems in wireless ad-hoc sensor networks. Computer Communications, 29(4), 413-420.

[28] Cardei, M., and Wu, J. (2004). Coverage in wireless sensor networks. Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, Ilyas M. and Mahgoub I., CRC Press (pp. 432-446), ISBN 0-8493-1968-4.

[29] Baek, S. J., Veciana, G. de, and Su, X. (2004). Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation. IEEE Journal on Selected Areas in Communication, 22(6), 1130-1140.

[30] Ahn, N., and Park, S. (2014). An optimization algorithm for minimum k-connected m-dominating set problem in wireless sensor networks. Wireless Networks, doi: 10.1007/s11276-014-0819-6.

[31] Slijepcevic, S., and Potkonjak, M. (2001). Power efficient organization of wireless sensor networks. In proceedings of IEEE International Conference on Communications, Helsinki, vol. 2, pp. 472-476.

[32] Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M. (2002). Optimizing sensor networks in the energy-latency-density design space. IEEE Transactions on Mobile Computing, 1(1), 70-80.

[33] Raghunathan, V., Schurghers, C., Park, S., and Srivastava, M.B. (2002). Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine, 19(2), 40-50.

[34] Lin, Y., Zhang, J., Chung, H.S.-H., Ip, W.H., Li, Y., and Shi, Y.-H. (2012). An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 42(3), 408-420.

[35] Ashouri, M., Zali, Z., Mousvi, S.R., and Hashemi, M.R. (2012). New optimal solution to disjoint set k-coverage for lifetime extension in wireless sensor networks. IET Wireless Sensor Systems, 2(1), 31-39.

[36] Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., and Gill, C. (2003). Integrated coverage and connectivity configuration in wireless sensor networks. In proceedings of the First ACM Conference on Embedded Networked Sensor Systems, SenSys'03, Los Angeles, USA, pp. 28-39, November 5 – 7.

[37] Wang, L., and Xiao, Y. (2006). A survey of energy-efficient scheduling mechanisms in sensor networks. Mobile Networks and Applications, 11(5), 723-740.

[38] Funke, S., Kesselman, A., Kuhn, F., Lotker, Z., and Segal, M. (2007). Improved approximation algorithms for connected sensor cover. Wireless Networks, 13(2), 153-164.

[39] Martins, F.V.C., Carrano, E.G., Wanner, E.F., Takahashi, R.H.C., and Mateus, G.R. (2009). A dynamic multiobjective hybrid approach for designing wireless sensor networks. In proceedings of IEEE Congress on Evolutionary Computation, CEC'09, Trondheim, pp. 1145-1152, May 18 - 21.

[40] Liu, C., Wu, K., Xiao, Y., and Sun, B. (2006). Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks. IEEE Transactions on Parallel Distributed Systems, 17(6), 562-575.

[41] Lin, J.-W., and Chen, Y.-T. (2008). Improving the coverage of randomized scheduling in wireless sensor networks. IEEE Transactions on Wireless Communications, 7(12), 4807–4812.

[42] Abrams, Z., Goel, A., and Plotkin, S. (2004). Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In proceedings of 3rd International Symposium on Information Processing in Sensor Networks, Berkley, USA, pp. 424–432, April.

[43] Berman, P., Calinescu, G., Shah, C., and Zelikovsky, A. (2005). Efficient energy management in sensor networks. in Ad Hoc and Sensor Networks, Wireless Networking and Mobile Computing, Edited by Pan Y. and Xiao Y., Nova Science Publishers (pp 71–90), ISBN: 1-59454-396-8.

[44] Cardei, M., MacCallum, D., Cheng, M. X., Min, M., Jia, X., Li, D., and Du, D.-Z. (2002). Wireless sensor networks with energy efficient organization. Journal of Interconnection Networks, 3(3–4), 213–229.

[45] Mohamadi, H., Salleh, S., Ismail, A. S., and Marouf S. (2014). Scheduling algorithms for extending directional sensor network lifetime. Wireless Networks, doi: 10.1007/s11276-014-0808-9.

[46] Benini, L., Castelli, G., Macii, A., Macii, E., Poncino, M., and Scarsi, R. (2000). A discrete-time battery model for high-level power estimation. In proceedings of Design, Automation and Test in Europe Conference and Exhibition, Paris, pp 35-39.

[47] Lai, C.-C., Ting, C.-K., and Ko R.-S. (2007). An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In proceedings of IEEE Congress on Evolutionary Computation, CEC 2007, Singapore, pp. 3531–3538

[48] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI.

[49] Michalewicz Z. (1996). Genetic Algorithm + Data Structures = Evolution Programs. Springer-Verlag, Berlin, Germany.

[50] Brabazon A. and O'Neill M. (2008). An introduction to evolutionary computation in finance. IEEE Computational Intelligence Magazine, 3(4), 42–55.

[51] Zhang 'J., Chung H. S.-H., and Lo W.-L. (2007). Clustering-based adaptivecrossover and mutation probabilities for genetic algorithms. IEEE Transactions on Evolutionary Computation, 11(3), 326–335.

[52] Zhang M., Gao X., and Lou W. (2007). A new crossover operator in genetic programming for object classification. IEEE Transaction on Systems, Man, Cybernetics, Part B: Cybernetics, 37(5), 1332–1343.

[53] Zhang J., Chung H. S.-H., Lo W.-L., Hui S. Y., and Wu A. K.-M. (2001). Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithms. IEEE Transactions on Power Electronics, 16(6), 752–763.