BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 65, No. 2, 2017
DOI: 10.1515/bpasts-2017-0030

Minimizing sensor movement in target coverage problem: A hybrid approach using Voronoi partition and swarm intelligence

A.M. JAGTAP* and N. GOMATHI

VEL-TECH Dr. RR & Dr. SR Technical University, Chennai 600 062, India

Abstract. This paper addresses the major challenges that reside on target coverage problem, which is one among the two primary sub-problems of node deployment problem. In order to accomplish a cost-efficient target coverage, a Voronoi partition-based, velocity added artificial bee colony algorithm (V-VABC) is introduced. The V-VABC is an advancement over the traditional, target-based Voronoi greedy algorithm (TV-greedy). Moreover, the VABC component of V-VABC is a hybrid, heuristic search algorithm developed from the context of ABC and particle swarm optimization (PSO). The V-VABC is an attempt to solve the network, which has an equal number of both sensors and targets, which is a special case of TCOV. Simulation results show that V-VABC performs better than TV-greedy and the classical and base algorithms of V-VABC such as ABC and PSO.

Key words: sensor, target, Voronoi, heuristic, ABC, PSO, VABC.

1. Introduction

Recently, wireless sensor networks (WSNs) are applied in various fields, such as sensing, surveillance, and monitoring purposes [1] with two main objectives: targeted coverage and connectivity [2]. So, to improve the coverage and connectivity of the network, sensor techniques are used [3, 4]. Deploying the sensor nodes to attain the maximum coverage with a minimum number of sensors is a challenging task. The surveillance of the WSN and quality of the detection power can be measured using the sensor coverage metric measure [5]. Network lifetime is also considered an important factor which determines the efficiency of a wireless sensor network. To achieve enhanced lifetime, energy usage should be reduced because of the battery-powered sensor nodes. Coverage problem includes the area coverage problem (covering the entire region) and target coverage problem (covering the specific area of interest). Target coverage is divided into simple coverage, k-coverage and Q-coverage [6]. Two types of sensor node deployments are introduced: random deployment (in an inaccessible region) and deterministic deployment (in an accessible region) [7, 8]. In the study, the optimal deployment of sensor nodes identification and scheduling is achieved using the artificial bee colony method. This method optimizes the target coverage and maximizes the lifetime of a network [6].

However, more studies have been done in detecting the target coverage with high detection probability, lowering the false alarm rate and detection delay. To prolong the network lifetime and minimize the mobility of the sensor nodes, many methods have been introduced [9]. Many heuristic approaches are used to maximize sensor coverage. Deterministic methods have been used to find the sensor coverage of 2D and 3D flat surfaces [5]. On the other hand, deterministic deployment seems to be complex in large networks. Based on immune algorithm, a new centralized algorithm is used to maximize the coverage area with less energy utilization. So, the centralized algorithm has an advantage over the very low processing power of the sensor nodes [10].

Sensor deployment is the basic problem in WSNs. It is divided into two kinds of methods: based on continued points and based on grid [11]. The system cost must be low and the connected system must cover the sensing field within WSNs. So, the minimum-cost and connectivity-guaranteed grid coverage (MCGC) occurs as an optimization problem [12]. This paper addresses the challenges in the first problem of the node deployment problem, termed target coverage (TCOV) problem [5]. In order to overcome the problem, a meta-heuristic search-based node deployment algorithm is proposed. The algorithm moves the sensors to cover the target at minimum movement distance, based on Voronoi diagram and heuristic search. The heuristic search is performed by adopting a hybrid version of artificial bee colony (ABC) and particle swarm optimization (PSO). Hence, the key contributions of this paper can be presented as follows.

We addressed the scenario of searching for servers (nodes) in the higher-order neighbour targets and the failure of the existing algorithm to handle it.

1. We propose a Voronoi partition-based velocity added ABC, abbreviated as V-VABC.
2. We hybridize the ABC with PSO to introduce velocity added ABC (VABC).
3. We have simulated the scenario of higher order neighborhood searching for servers (nodes) and the superiority of V-VABC over TV-greedy algorithm.

The rest of the paper is organized as follows. Section 2 reviews the related works and Section 3 gives the preliminaries of the system model and the objective of the TCOV solution. Section 4 explains the TV-greedy algorithm and the proposed V-VABC algorithm, and Section 5 details the hybrid version of

*e-mail: artap2016@gmail.com

ABC, termed VABC. Section 6 discusses the results and Section 7 concludes the paper. In this paper, server, chief server, and aid servers represent nodes, and the targets are equivalent to sinks of a network.

2. Literature review

2.1. Related works. In 2012, Liu [12] addressed the issue of minimum-cost and connectivity guaranteed grid coverage (MCGC) for the implementation of wireless sensor networks (WSNs). To find a solution to the problem of MCGC, he proposed a new algorithm, termed ant colony optimization [13], with three classes of ant transitions (ACO-TCAT). The algorithm decreased the inferior solutions and narrowed the searching range of the algorithm. In 2014, Mini et al. [6] have found that the sensor deployment is done based on the area coverage [14–18], which yields less network lifetime. So they focused on target coverage and optimized the sensor node deployment and scheduling to achieve the maximum target coverage and increasing the network life time. For computing the deployment locations, they exploited three methods, such as artificial bee colony algorithm (ABC), heuristic, and particle swarm optimization. After computing the optimal locations, they scheduled the sensor nodes to avoid the battery drain of all nodes using the heuristic approach for achieving the theoretical upper bound of network lifetime. Thus, the study confirmed that the network lifetime and maximum target coverage can be extended by using the ABC method of deploying the sensor nodes at optimized locations, and then scheduling them to achieve the theoretical upper bound.

However, in 2014, Liao et al. [9] have found a small number of works on minimizing the sensor mobility. Hence, they started investigating the mobile sensor deployment (MSD) problem to obtain the minimal movement and the maximum network connectivity. For minimizing the sensor movement, the target coverage (TCOV) and network connectivity (NCON) problems are taken into consideration. An algorithm based on the Hungarian method for special cases, the basic algorithm based on clique partition for general cases, and the TV-greedy algorithm based on Voronoi partition of the deployment region are proposed to minimize the movement distance of sensors. Steiner minimum tree method is constructed to solve the network connectivity problem. In 2014, Temel et al. [5] studied works related to deterministic sensor deployment [9], and found out that the implementation of that method in 3D environment has not yet been accomplished. So, they proposed a deterministic sensor deployment method, which is based on wavelet transform (WT) to maximize the quality of coverage of a WSN with a minimum number of sensors on a 3D surface. For achieving the solution, they exploited the probabilistic sensing model and Bresenham's line of sight algorithm. The cat swarm optimization (CSO) algorithm is modified for finding the sensor deployment problems on 3D terrains. The modified algorithm is compared with the Delaunay triangulation and genetic algorithm-based methods.

In 2015, Zahhad et al. [10] noted the coverage problem that is mentioned in Hui et al. [11] work on hybrid sensor deployment strategy. For relocating the mobile nodes after the initial configuration to maximise the coverage area, they proposed a new centralised deployment algorithm based on the immune optimisation algorithm. Using the Matlab simulation technique, the proposed algorithm is compared with the previous algorithms.

2.2. Problem statement. Traditionally, various methods are used for improving the sensor mobility, targeted coverage, and increasing the network lifetime and connectivity. The methods used are ACO with three classes of ant transitions (ACO-TCAT), the greedy algorithm, and particle swarm optimisation (PSO). ACO-TCAT [11] algorithm is used to improve the quality of the solution space, narrow the searching speed, and solve the combinatorial optimization problem of minimum-cost and connectivity guaranteed grid coverage (MCGC). The drawbacks of this method include high storage and computing cost [19, 20]. Also, the position affection of the sink, node communication problem, and the connectivity problem are not considered. The TV-greedy algorithm based on Voronoi partition of the deployment region is used to minimize the movement distance of sensors to obtain the targeted coverage. Sometimes they fail to produce the optimal solution and may even produce the unique worst possible solution. PSO approach is used to maximize the coverage rate based on a probabilistic sensing model in mobile WSNs (MWSNs). It is a continuous technique that is very poorly suited to combinatorial problems.

Since traditional methods have a lot of drawbacks, various innovative methods have been proposed. The innovative methods include immune algorithm (IA), artificial bee colony (ABC) algorithm, and cat swarm optimization (CSO) algorithm. IA is an innovative centralised deployment algorithm used to maximise the coverage area with minimum mobility energy consumption. The IA method requires very low power consumption from the sensor nodes and it also works well for networks with obstacles. ABC algorithm was found to be effective and it is suggested to use this robust technique for sensor node deployment. Constrained optimization, data clustering, and benchmark functions are the drawbacks of the ABC system. CSO method is a deterministic sensor deployment method, which is based on wavelet transform (WT) used to maximize the quality of coverage of a WSN with a minimum number of sensors on a 3D surface. These methods have drawbacks, too. So, finding an optimized method for the enhancement of network lifetime, increasing the connectivity, targeting the coverage and minimizing the mobility of a sensor are essential. This is quite challenging and shows a wide, open area for research in the field of WSN. Since the basic algorithm [9] exhibits sequential programming, the convergence to minimum distance often becomes slow. As a result, the targeted coverage cannot be accomplished within a stipulated period.

3. System model and TCOV problem

3.1 System model. The system model is composed of N_t targets and N_s sensors, the representations of which are

$\{T_i\}$: $i = 1, 2, \ldots, N_t$ and $\{S_j\}$: $j = 1, 2, \ldots, N_s$, respectively. All the mobile sensors are kept informed on the details of their position through the use of a GPS unit residing in them, or a localization service concerned with the network. A control center also exists to gather information regarding the location of the sensors or to broadcast movement orders, which are to be sent to the mobile sensors at a later stage, whereas the computations are performed in the servers. The task area is completely free from any sort of obstacles that are caused by the motion. But when the obstacles are present, a sensor helps to overcome them through selecting the most suitable, shortest path leading to the destination.

1) Disk model: The disk model portrayed in [21] allows the sensors to sense, as well as communicate, using a sensing radius of r_s and the communication radius of r_c. All the sensors are capable of covering any number of targets at a particular instant.

Condition 1: A target is acknowledged as covered when one or more sensors are found to surely reside in a disk of radius r_s with its center on the target. The disk can be now stated the target's coverage disk, while the circle formed from the coverage disk can be termed the target's coverage circle [9].

2) Mobility model: The network uses a free mobility model [22], wherein the sensors are rendered with complete freedom to move without disruption or cease at any place. The distance covered when the sensor is under motion can aid in yielding the amount of energy that the sensor has consumed to establish the movement. The move distance of the sensor S_j for covering a target T_i is $D(S_j, T_t)$, where $D(\bullet)$ specifies the Euclidean distance between S_j and T_i. Likewise, $D(S_j, S_k)$ refers to the movement distance between the sensors S_j and S_k. Under a no-obstacle condition, a sensor has to travel in a straight line from its initial position to the target to reach the coverage circle of a target, and hence, it minimizes the movement distance of a sensor to the target.

3.2. Objective model. The objective of this paper is to determine or identify the sensors to be moved and the location to which they should be deployed to cover the target, as well as to establish network connectivity. This is basically called mobile sensor deployment (MSD) problem [9]. There are two main issues in MSD, namely target coverage and network connectivity. Thus, the solutions for the MSD problem can be divided into two sub-tasks, namely placing sensors at the minimum cost of movement to cover the target and to deploy the rest of sensors to enable connectivity among the coverage sensors and the sink. More details about the problem and the definitions can be found in [9].

3.3. TV-greedy algorithm for TCOV problem model. The TV-Greedy algorithm aims at minimizing the total distance for the sensors to move from their current position to cover the targets. Algorithm 1 refers to the pseudo-code of the TV-greedy algorithm, which is explained in brief. More information about the algorithm can be acquired from [9].

Algorithm 1. TV Greedy Algorithm

Input: T_i // Target information
S_j // Sensor information
r_s // Sensing radius
Output: C_M // Cost of movement
Perform Voronoi partition (*VP*) on T targets;
Determine neighbors for each target according to their Voronoi polygon;
Determine the *OSG* for each target according to S_j and *VP*
for each OSG_i **do**
  Determine the *chief server*
  Identify the aid server for T_i 's neighbor
for every T // i.e. T_i **do**
  if T_i has already been covered **then**
    Return $C(T_i) = 0$
  else
    Produce GSG_i of T_i
  If $GSG_i \neq 0$ **then**
    Move the nearest server to cover T_i
  Return $C(T_i)$ = moving distance
  else
    if neighbor's *chief server* could be shared **then**
      Move the nearest *chief server* to cover T_i;
      Return $C(T_i)$ = moving distance;
    else
      Regenerate the *GSG* of T_i by searching for aid servers of the T_i 's 2nd or higher order neighbors
      Move the nearest aid server to cover T_i;
      Return $C(T_i)$ = moving distance;
$C_M = C_M + C(T_i)$;
Return C_M

The principle behind the TV-greedy is to deploy the closest server to the uncovered target. Voronoi partitions are used to cluster the sensors based on their distance to the target, because the sensors available in a Voronoi polygon of a target are closer to the target than other sensors [9]. Since the sensors are aware of the coordinates of the targets, Voronoi partition of the targets is done based on the coordinates. The partitioned Voronoi polygons are used to determine the neighbors of each target, followed by constructing an own server group (OSG) of every target. From every OSG, a chief server is determined by identifying the own server that is closest to the target. The other servers are subjected to determining their distance to the neighbors, for selecting them as aid servers of the respective neighbors. When a target is initially covered, its OSG stands by and remains to collect orders, otherwise a candidate server group (CSG), which is a union of chief servers and the possible aid servers of its neighbors, is constructed. If the constructed

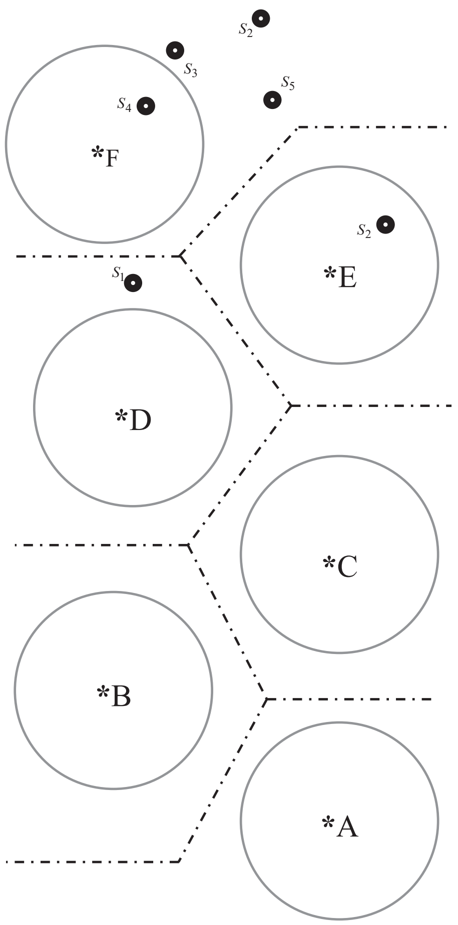

Fig. 1. A special scenario of the TCOV problem, in which a target needs to get an aid server from a higher degree of neighbors

CSG is not empty, the closest server is moved to cover the target, otherwise an exit neighbor of the target is determined for possible sharing of its chief server with our target. Such chief server moves from its initial position to a new position, which is within the coverage disk of both targets.

If no such possibility of sharing persists, higher-order neighborhood of T_i is determined and hence, its CSG is regenerated. Subsequently, the nearest aid server is moved to the new position of the coverage disk of T_i.

Under a circumstance of regenerating the CSG using higher-order neighbors, the TV-greedy algorithm faces numerous practical challenges. Consider a scenario of targets and sensors as illustrated in Fig. 1. Here, T_A is not covered by any servers with empty CSG. Hence, the CSG has to be regenerated for second-order neighborhood. For second-order neighborhood, the CSG becomes $\{S_1\}$. When the CSG is exploited, according to step 12, then TD loses its chief server. Hence, the CSG has to be built for higher-order neighborhoods. This increases the complexity of finding suitable servers for Target T_A. This situation can be handled by the proposed V-VABC algorithm searching for suitable sensors from the search space of all sensors. It also avoids the problem of assigning the covering server to other targets by memorizing them.

4. Proposed solutions for TCOV problem model

4.1. Proposed V-VABC algorithm. The proposed V-VABC exploits the heuristic nature of hybrid features of PSO and ABC when higher-order neighborhood search is required in the TV-greedy algorithm. Hence, the regeneration phase of TV-greedy algorithm is replaced by the proposed VABC phase. The basic definitions used in V-VABC are given below, followed by the pseudo-code and its description.

Algorithm 2. V-VABC Algorithm

Input	T_i // Target information
	S_j // Sensor information
	r_s // Sensing radius
Output	C_M // Cost of movement
1	$CVSG = \phi$;
2	Perform Voronoi partition (*VP*) on *T* targets;
3	Determine neighbors for each target according to their Voronoi polygon;
4	Determine the *OSG* for each target according to S_j and *VP*;
5	**for** each GSG_i **do**
6	Determine the *chief server*;
7	Identify the aid server for T_i's neighbor;
8	**for** every *T* // i.e. T_i **do**
9	**if** T_i has already been covered **then**
10	Update *CVSG*;
11	Return $C(T_i) = 0$;
12	**else**
13	Produce CSG_i of T_i;
14	**if** $CSG_i \neq 0$ **then**
15	Move the nearest server to cover T_i;
16	Update *CVSG*;
17	Return $C(T_i) =$ moving distance;
18	**else**
19	**if** neighbor's *chief server* could be shared **then**
20	Move the nearest *chief server* to cover T_i;
21	Update *CVSG*
22	Return $C(T_i) =$ moving distance
23	**Else**
24	$VABC(T_i, S_j, r_s, c_s)$ // Determines the sensors to be moved and returns the moving distance;
25	Update *CVSG*;
26	Return $C(T_i) =$ moving distance;
27	$C_M = C_M + C(T_i)$;
28	Return C_M

Definition 1. A covering server group (CVSG) is a group of servers that are already deployed to cover a target.

Lemma 1. A server from CVSG can be shared between two targets only if both targets share a common Voronoi polygon edge.

The major difference between the TV-greedy algorithm and the proposed V-VABC resides in assigning the sensors when there is either no sensor to cover the target from its OSG or

CSG, or sharing from its neighbors' chief server. However, the CVSG is being updated: (1) when the target is covered by the server which is in the coverage disk of the target, (2) when the target is covered by the nearest server of its CSG, (3) when the target is covered by the chief server of the neighbor under a sharing basis, and (4) when the target is covered by servers which are not based on the above three constraints. The updating process is nothing but a set union operation between the server that is covering the target and the CVSG. Since the VABC is a meta-heuristic search algorithm, it requires an objective model to be minimized. The adopted objective model is of conditional type, as given in (1), where $D(\bullet)$ is the Euclidean distance between the target and the sensor. The $D(\bullet)$ is considered only if the sensor is not available in CVSG, otherwise $F(S_j, T_t)$ gets the Euclidean distance of the farthest sensor.

$$S_k^* = \operatorname{argmin} F(S_j, T_t) \tag{1}$$

$$F(S_j, T_t) = \begin{cases} D(S_j, T_t);\ if\ S_j \notin CVSG \\ \max(D(S_j, T_t)\forall j);\ otherwise \end{cases} \tag{2}$$

4.2. Hybridization of ABC and PSO.

4.2.1 PSO. PSO is a renowned swarm intelligence algorithm introduced in 2001 [23]. Since then, PSO has been applied in solving global optimization problems [24, 25]. The pseudo-code of PSO when applying for solving the TCON problem is given below. The PSO attempts to minimize the objective function given in (1).

Algorithm 3. PSO for server movement	
Input	L // Solution length
	$F(\bullet)$ // Objective function
Output	S_k^* // Movement information of the sensor
1	Generate S_p // Initial solutions
2	Generate V_p // Initial velocity
3	Determine $F(S_p)$ // Evaluation of initial solutions
4	Define N_{swarm} // Number of searching swarms
5	Set *swarm* to 1
6	While $swarm < N_{swarm}$ do
7	Determine $P_{swarm,p}^{best}$ and G_{swarm}^{best}
8	Update V_p // Update velocity
9	Update S_p // Update solution
10	Determine $F(S_p)$
11	$S_k^* = G_{swarm}^{best}$
12	Return S_k^*

The objective function $F(\bullet)$ and the number of solution variables L are given as input to the PSO. The PSO returns the server and its new position to cover the target as its output.

In Step 7, the $P_{swarm,p}^{best}$ is the best solution that has been achieved so far by identifying the S_p (sensor information), whereas the G_{swarm}^{best} is the best value obtained so far among all the solution variables in the population.

In Steps 8 and 9, the V_p and S_p are determined using (3) and (4), respectively, where w refers to inertia weight, c_1 and c_2 are acceleration constants, and r_1 and r_2 are arbitrary integers within the interval [0, 1]. In every cycle, S_k is updated (in Step 11) by the determined G_{swarm}^{best}.

$$\begin{aligned} V_p(l) = wV_p(l) + c_1 r_1\left(P_{swarm,p}^{best}(l) - S_p(l)\right) + \\ + c_2 r_2\left(G_{swarm}^{best} - S_p(l)\right) \end{aligned} \tag{3}$$

$$S_p(l) = S_p(l) + V_p(l) \tag{4}$$

4.2.2. ABC. ABC is a meta-heuristic search algorithm introduced by Karaboga [26, 27]. Since its introduction, the algorithm has had a huge number of applications [28, 33], including the sensor deployment problem [34]. The ABC adopted here for validating the applicability is explained in the pseudo-code given in Algorithm 4.

$$S_p^u(l) = \begin{cases} S_p(l) + \phi_p(l)\left(S_p(l) - S_p(k)\right);\ r_l < M_R \\ S_p(l);\ otherwise \end{cases} \tag{5}$$

Algorithm 4. ABC for server movement	
Input	L // Solution length
	$F(\bullet)$ // Objective function
Output	S_k^* // Movement information of the sensor
1	Generate S_p // Initial solutions
2	Determine $F(S_p)$ // Evaluation of initial solutions
3	Define N_{cycle} // Number of cycles of searching
4	Set *cycle* to 1
5	While $cycle < N_{cycle}$ do
6	Determine S_p^u // for employee bee
7	Perform greedy selection between S_p and S_p^u
8	Determine $P(S_p)$
9	Determine S_p^u // for onlookers
10	Perform greedy selection between S_p and S_p^u
11	If $F(S_p)$ is not improved for long time, then
12	Reinitialize S_p // Scout bee phase
13	Memorize best among $F(S_p)$
14	$S_k^* = S_p$
15	Increment *cycle* by one
16	Return S_k^*

The searching process of the ABC algorithms begins in Step 6 with S_p^u being determined based on S_p using (5), where S_p^u is the updated employed bee, S_p is the previous employed bee, $S_p(l)$ is the l^{th} solution variable of the p^{th} employed bee, M_R refers to the modification rate, usually set as greater than 0.5 to achieve a high degree of recombination, $\varphi(\bullet)$ and r_l are random functions to generate arbitrary integer within the interval [0, 1]. The similar modification is applied for onlooker bees also (in Step 9). The greedy selection process selects either S_p^u or S_p, based on its objective function given in (1), and considers S_p for further processes. $P(S_p)$ in Step 8 determines the probability of S_p to be selected for the onlooker bee. When no significant improvements are achieved in the onlooker bee phase, S_p is considered a scout bee, and hence reinitialized. More details on ABC can be found in [25].

4.2.3 VABC. VABC is a hybrid version of ABC and PSO, introduced to achieve a performance improvement over both ABC and PSO. Despite the context of VABC having been reported in [35], our VABC incorporates a velocity update in the employed bee phase, rather than in the onlooker bee phase. As a result, the convergence can be improved because of the global searching principle of PSO being applied earlier (in the employed bee phase), followed by local searching using the onlooker phase. The pseudo-code of the VABC for server movement is given in Algorithm 5.

Algorithm 5. VABC for server movement

Input	L // Solution length
	$F(\bullet)$ // Objective function
Output	S_k^* // Movement information of the sensor
1	Generate S_p // Initial solutions
2	Generate V_p // Initial velocity
3	Determine $F(S_p)$ // Evaluation of initial solutions
4	Define N_{cycle} // Number of cycles of searching
5	Set *cycle* to 1
6	While $cycle < N_{cycle}$ do
7	Determine $P_{swarm,p}^{best}$ and G_{swarm}^{best}
8	Update V_p
9	Determine S_p^u
10	Perform greedy selection between S_p and S_p^u
11	Determine $P(S_p)$
12	Determine S_p^u for onlookers
13	Perform greedy selection between S_p and S_p^u
14	If $F(S_p)$ is not improved for long time, then
15	Reinitialize S_p
16	Memorize best among $F(S_p)$
17	$S_k^* = S_p$
18	Increment *cycle* by one
19	Return S_k^*

The VABC differs from ABC in updating its employed bees. In VABC, velocities for every employed bee are initialized in Step 2 by generating arbitrary integers within the interval [0, 1] for each solution variable.

Similar to PSO, $P_{swarm,p}^{best}$ and G_{swarm}^{best} are determined for the employed bees, and based on those, the velocities are updated and the employed bees are using (5), respectively. The updated employed bees S_p^u are subjected to greedy selection to continue the rest of the regular ABC processes.

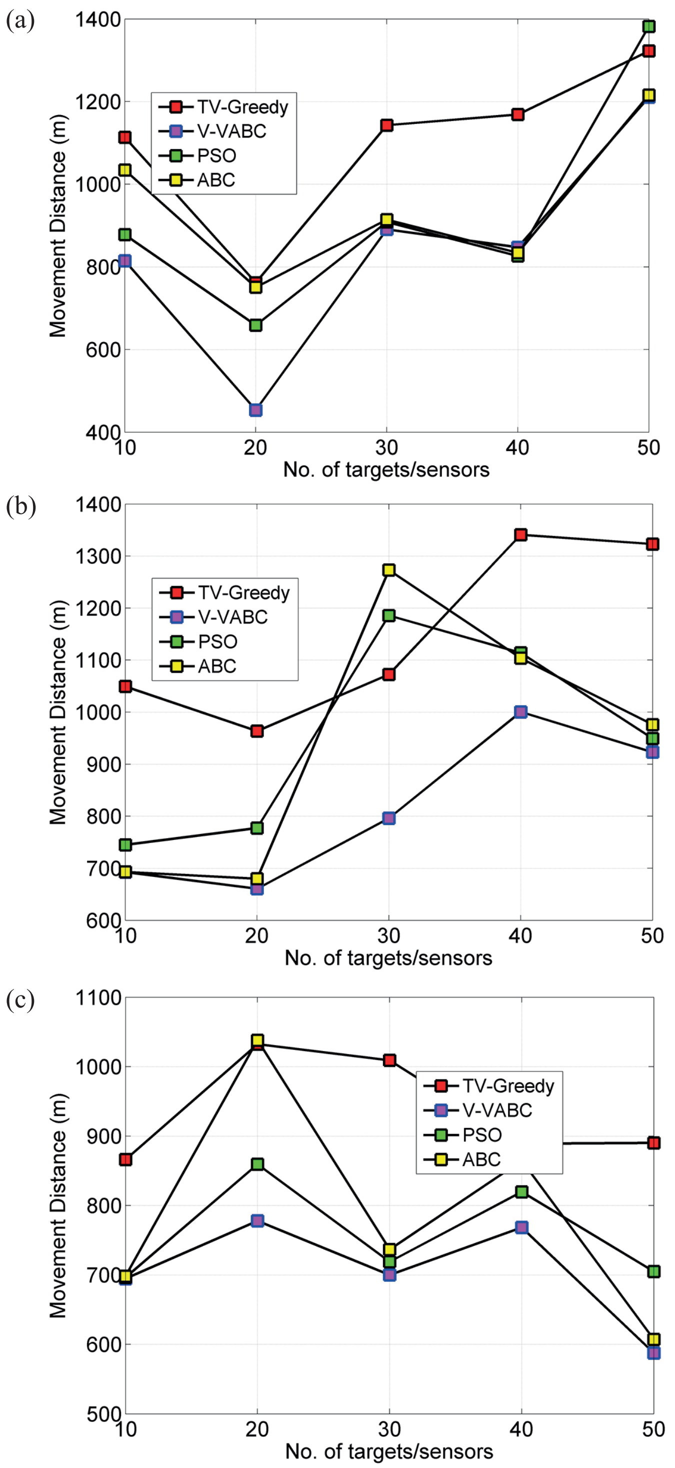

Fig. 2. Comparative study on varying number of targets and sensors, when the sensing radius is set to (a) 5 m, (b) 10 m, and (c) 15 m

5. Simulation results

5.1. Simulation setup. The sensor network is simulated in MATLAB with the dimension of 400 m × 400 m and the simulation is conducted between the TV-greedy algorithm, VVAC algorithm, ABC algorithm, and PSO algorithm to solve the TCOV problem. The source code of the ABC algorithm has been acquired from http://mf.erciyes.edu.tr/abc/software.htm.

The sensors are initially deployed in the network randomly. In order to match a more realistic environment, the targets are placed randomly. The movement cost exhibited by each algorithm is observed for varying network configurations. Since a heuristic approach is adopted in the proposed V-VABC algorithm, each algorithm is executed 50 times and the mean costs are observed. The PSO parameters, such as w, c_1, and c_2, are set to 2, whereas the common parameters, such as N_{cycle} and N_{swarm}, are set to 100. Since the primary intention of all these algorithms is to move the nodes to cover the targets, it does not focus on determining the rate of data exchange and the respective computation overhead.

5.2. Comparative analysis. The comparative study can be made in two phases. In the first phase, the V-VABC is compared with TV-greedy algorithm to demonstrate the superiority of V-VABC. In the second phase, the V-VABC is compared against the basic heuristic algorithms such as PSO and ABC to demonstrate the superior performance of hybridization in solving the TCOV problem over the existing algorithms.

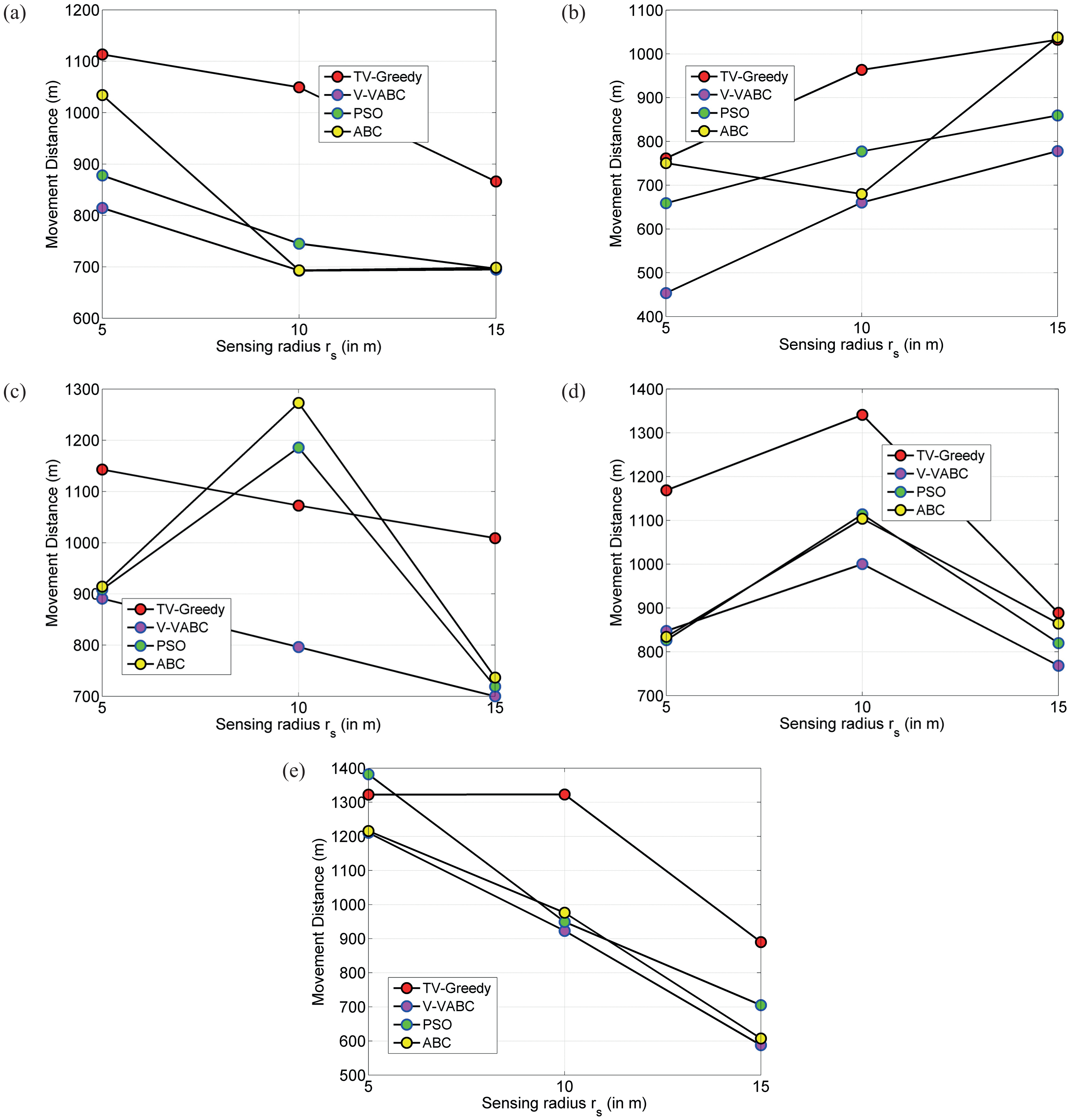

Fig. 3. Comparative study on varying rs, when the number of targets and sensors are set to (a) 10, (b) 20, (c) 30, (d) 40, and (e) 50

Table 1
Mean and standard deviation of the proposed V-VABC over TV-greedy, ABC, and PSO algorithms

r_s (m)	TV-greedy		ABC		PSO		V-VABC	
	μ	*Stdev*	*μ*	*Stdev*	*μ*	*Stdev*	*μ*	*Stdev*
5	927.4	27.304	937.19	26.967	814.77	22.586	734.57	21.432
6	984.41	27.883	874.81	23.809	801.52	26.586	721.22	17.892
7	1010.1	29.751	811.27	25.492	791.2	23.059	715.06	21.813
8	1043.8	31.585	762.59	21.824	768.91	23.879	704.63	19.606
9	1047.8	31.692	723.96	24.545	760.17	23.606	693.56	20.341
10	1052.6	30.852	693.58	20.94	746.52	20.809	689.15	18.446
11	1036.7	29.87	691.42	17.608	736.1	22.179	670.45	18.704
12	1016.5	32.633	684.9	20.556	719.22	20.78	667.6	19.075
13	973.3	26.134	690.53	17.296	711.79	21.82	665.42	16.555
14	928.56	24.676	691.99	20.798	702.2	17.769	678.22	17.915
15	865.51	25.53	698.97	21.033	692.83	19.91	697.14	18.203

r_s – Sensing radius, μ – mean, Stdev – standard deviation

When r_s was 5 m, the V-VABC has achieved better performance on moving 10 and 20 sensors to cover 10 and 20 targets, respectively. The cost of movement is considerably smaller than the cost incurred by other algorithms. When attempting to move 30 sensors to cover 30 targets, the V-VABC has achieved not far better, yet better performance than the other algorithms. When attempting to move 40 sensors, the V-VABC could gain only the third position, but retains first position when moving 50 sensors.

When r_s is set to 10 and 15 m, the V-VABC remains in the first position in assigning the definite number of sensors to the targets. Few instances such as 10 targets with an r_s of 10 m and 15 m, the performance of ABC is closer to V-VABC, yet not dominating. The similar results have been portrayed in the perspective of r_s in Fig. 3. According to Fig. 3, first let us consider only 10 sensors and 10 targets. When r_s is minimum, the V-VABC dominates other algorithms, while ABC comes closer to V-VABC, and PSO accompanies further, when r_s is increased to 10 m and 15 m, respectively. When the number of sensors and targets in the network was 20, 30, and 50, the V-VABC presented remarkable performance over other algorithms. However, when the network had 40 sensors and targets, the V-VABC gained the third position with least r_s, yet it retains the first position in the rest of r_s settings.

Considering Figs. 2 and 3, the V-VABC has achieved superior performance over TV-greedy. There is no compromise on it at any instance. This infers that the V-VABC has the ability to move sensors from higher-order neighbourhood at a smaller distance than TV-greedy. On the other hand, the V-VABC has a few compromises with the heuristic search algorithms. This is because of the meta-heuristic effect of ABC and PSO on the V-VABC. However, the majority of the simulation instances have been dominated by V-VABC over PSO and ABC. From Table 1, our proposed method outperforms the other existing methods in mean and standard deviation.

5.3. Order of targets and sensors. In [9], the simulation constraints have been set to varying numbers of targets and sensors separately. The assumption is that there are more sensors in the network than the targets. In a scenario of an equal number of sensors and targets, higher order neighbourhood plays key role. During the simulation, we have observed a frequent occurrence of such scenarios and illustrated it in Fig. 4.

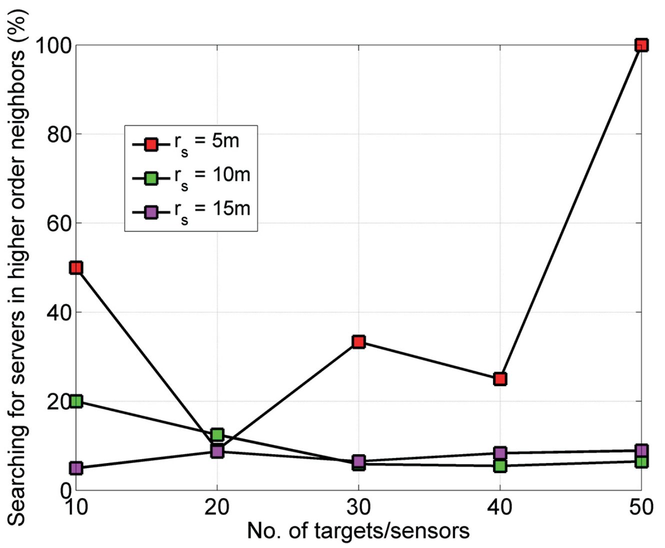

Fig. 4. The percentage of simulations in which the sensors need to be moved from a target's Voronoi partition of higher-order neighborhood

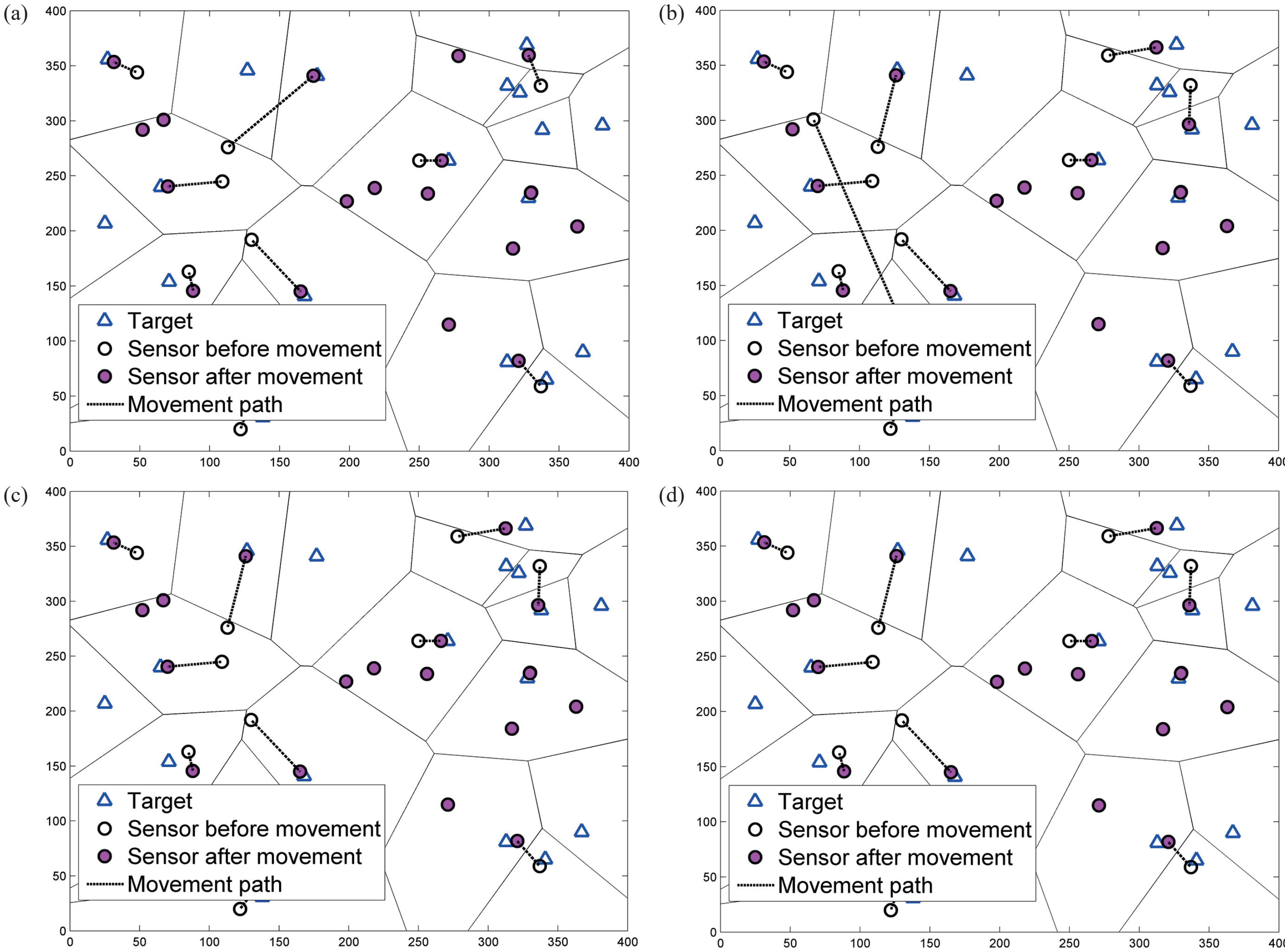

Fig. 5. An illustration of possible sensor movements in a network with 20 targets and 20 sensors, exhibited by (a) TV-greedy, (b) V-VABC, (c) PSO, and (d) ABC

When r_s is set at minimum, the occurrence of the need for higher-order neighbourhood arises, irrespective of the number of targets and sensors. For instance, 100% searching has been done in the higher-order neighbours, where the network has 50 sensors and targets. The TV-greedy algorithm finds a huge challenge in handling such scenarios. Moreover, the movement distance has been reported as inversely proportional to the number of sensors and targets. However, a search in the higher-order neighbour area often leads to an increase of the movement cost, although there are less sensors and targets. This can be interpreted from Figs. 2 and 3, where there is no reciprocal relationship maintained between the movement distance and the number of sensors and targets. The movement exhibited by the algorithms in such a scenario is illustrated in Fig. 5.

6. Conclusions and future work

This paper has addressed a special scenario of the TCOV problem, under which a sensor has to be moved from the Voronoi partition of higher-order neighbourhood of a target. The proposed V-VABC has been investigated for its performance over the traditional TV-greedy algorithm and the heuristic search algorithms such as ABC and PSO. Such special scenario occurs when the number of sensors and targets are equal to each other. The simulation results reveal that this scenario occurs in 20–100% of the simulations. The V-VABC has handled such scenarios efficiently and hence the sensors were moved to cover the targets at minimum cost. The simulated investigations have been done with a varying network configuration, and the competing performance of V-VABC over TV-greedy, PSO, and ABC was observed. Hence, the TCOV problem, which is to enable the connectivity between the targets and the sensors, has been solved. Since TCOV is one of two sub-problems of the node deployment problem, our future plans are to solve the second sub-problem, the network connectivity (NCON) problem. Unlike the TCOV problem, the NCON problem connects the target with the base station (master sink or the control centre) by placing series of sensors between the coverage sensor and the base station and hence the respective target can be connected.

This requires additional algorithms to specially handle those connecting sensors, termed as rest sensors. So, the future work is intended to develop an algorithm, such that will lead to effective node deployment.

References

[1] C.-Y. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges", *Proc. of the IEEE* 91 (8), 1247–1256 (2003).

[2] M. Lu, J. Wu, M. Cardei, and M. Li, "Energy-efficient connected coverage of discrete targets in wireless sensor networks", *Networking and Mobile Computing* 3619, 43–52 (2005).

[3] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks", *IEEE Transactions on Parallel and Distributed Systems* 24 (2), 301–311 (2013).

[4] R.C. Luo and O. Chen, "Mobile sensor node deployment and asynchronous power management for wireless sensor networks", *IEEE Transactions on Industrial Electronics* 59 (5), 2377–2385 (2012).

[5] S. Temel, N. Unaldi, and O. Kaynak, "On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform", *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44 (1), 111–120 (2014).

[6] S. Mini, S.K. Udgata, and S.L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor network", *IEEE Sensors Journal* 14 (3), 636–644 (2014).

[7] Y. Gu, H. Liu, and B. Zhao, "Target coverage with QoS requirements in wireless sensor networks", *Proc. Intell. Pervas. Comput*., 35–38 (2007).

[8] M. Chaudhary and A.K. Pujari, "Q-coverage problem in wireless sensor networks", *Proc. Int. Conf. Distrib. Comput. Netw*., 325–330 (2009).

[9] Z. Liao, J. Wang, S. Zhang, J. Cao, and G. Min, "Minimizing movement for target coverage and network connectivity in mobile sensor networks", *IEEE Transactions on Parallel and Distributed Systems* 26 (7), 1971–1983 (2015).

[10] M.A. Zahhad, S.M. Ahmed, N. Sabor, and S. Sasaki, "Utilisation of multi-objective immune deployment algorithm for coverage area maximisation with limit mobility in wireless sensors networks", *IET Wireless Sensor Systems* 5 (5), 250–261 (2015).

[11] W.C. Ke, B.H. Liu, and M.J. Tsai, "Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is NP-Complete", *IEEE Trans. Computers* 56 (5), 710–715 (2007).

[12] X. Liu, "Sensor deployment of wireless sensor networks based on ant colony optimization with three classes of ant transitions", *IEEE Communications Letters* 16 (10), 1604–1607 (2012).

[13] M. Dorigo and T. Stutzle, *Ant Colony Optimization,* MIT Press, 2004.

[14] C. Liu, K. Wu, and V. King, "Randomized coverage-preserving scheduling schemes for wireless sensor networks", *Proc. NETWORKING 2005*, 956–967 (2005).

[15] L.-H. Yen and Y.-M. Cheng, "Range-based sleep scheduling (RBSS) for wireless sensor networks", *Wireless Pers. Commun*. 48 (3), 411–423 (2009).

[16] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks", *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput*., 322–333 (2006).

[17] A. Makhoul and C. Pham, "Dynamic scheduling of cover-sets in randomly deployed wireless video sensor networks for surveillance applications", *Proc. 2nd IFIP Conf. Wireless Days*, 73–78 (2009).

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

[19] S. Dhillon, K. Chakrabarty, and S.S. Ivengar, "Sensor placement for grid coverage under imprecise detections", *Proc. 2002 International Conference on Information Fusion*, 1581–1587 (2002).

[20] S.S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks", *Proc. 2003 IEEE Conference on Wireless Communications and Networking*, 1609–1614 (2003).

[21] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T.H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity", *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 131–142 (2006).

[22] R. Tan, G. Xing, J. Wang, and H.C. So, "Exploiting reactive mobility for collaborative target detection in wireless sensor networks", *IEEE Trans. Mobile Comput.* 9 (3), 317–332 (2010).

[23] J. Kennedy, R.C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, 2001.

[24] K.E. Parsopoulos and M.N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization", *Natural Computing: An International Journal* 1 (2–3), 235–306 (2002).

[25] M. Clerc and J. Kennedy, "The particle swarm – explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. Evolutionary Computation* 6 (1), 58–73 (2002).

[26] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm", *Journal of Global Optimization* 39 (3), 459–171 (2007).

[27] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", *Applied Soft Computing* 8 (1), 687–697 (2008).

[28] M. Sonmez, "Artificial bee colony algorithm for optimization of truss structures", *Applied Soft Computing* 11 (2), 2406–2418 (2011).

[29] S.N. Omkar, J. Senthilnath, R. Khandelwal, G.N. Naik, and S. Gopalakrishnan, "Artificial bee colony (ABC) for multi-objective design optimization of composite structures", *Applied Soft Computing* 11 (1), 489–499 (2011).

[30] A.R. Yildiz, "A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing", *Applied Soft Computing* 13 (5), 2906–2912 (2013).

[31] A.R. Yildiz, "Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach", *Information Sciences* 220, 399–407 (2013).

[32] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding", *Applied Soft Computing* 13 (6), 3066–3091 (2013).

[33] I. Develi, Y. Kabalci, and A. Basturk, "Artificial bee colony optimization for modelling of indoor PLC channels: A case study from Turkey", *Electric Power Systems Research* 127, 73–79 (2015).

[34] C. Ozturk, D. Karaboga, and B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm", *Sensors* 11 (6), 6056–6065 (2011).

[35] N. Imanian, M.E. Shiri, P. Moradi, "Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems", *Engineering Applications of Artificial Intelligence* 36, 148–163 (2014).