Optimizing Connected Target Coverage in Wireless Sensor Networks Using Self-Adaptive Differential Evolution

O. GOKALP

Abstract—**Wireless Sensor Networks (WSNs) are advanced communication technologies with many real-world applications such as monitoring of personal health, military surveillance, and forest wildfire; and tracking of moving objects. Coverage optimization and network connectivity are critical design issues for many WSNs. In this study, the connected target coverage optimization in WSNs is addressed and it is solved using the self-adaptive differential evolution algorithm (SADE) for the first time in literature. A simulation environment is set up to measure the performance of SADE for solving this problem. Based on the experimental settings employed, the numerical results show that SADE is highly successful for dealing with the connected target coverage problem and can produce a higher performance in comparison with other widely-used metaheuristic algorithms such as classical DE, ABC, and PSO.**

Index Terms—**Connected Target Coverage, Metaheuristics, Optimization, Self-Adaptive, Wireless Sensor Networks.**

I. Introduction

A WIRELESS SENSOR Network (WSN) is one of the most widely used advanced communication technologies and have numerous application areas such as personal health monitoring [1], military surveillance [2], forest wildfire monitoring [3], air pollution monitoring [4], and moving object tracking [5].

As one of the main design issues in WSN, a sensor deployment plan that places individual sensors in a given region should be provided. A good deployment plan will be useful for coverage maximizing, connectivity maximizing, energy efficiency, and lifetime optimization [6]. Coverage is a fundamental property of every WSN that a network cannot detect events in its environment and may become useless without a sufficient sensing area. Therefore, it is important to increase the total coverage ratio as much as possible. Coverage problems can be divided into two classes, namely area coverage and target coverage that the former aims to cover the whole area of interest, whereas the latter aims to cover some specific points.

Connectivity is another critical requirement for WSNs, and without it, it is not possible to transfer information, no matter how high the coverage rate is. Two sensors are considered as connected if they can communicate with each other (e.g. send and receive data) either directly or via other sensor nodes. Fully connected networks require the existence of at least one path between any node and the sink node. Sometimes, intermittent connectivity is allowed when mobile sinks are provided to move and collect data from disconnected nodes [7].

As optimal sensor deployment problem for coverage is a hard optimization task, there have been so many studies proposed using metaheuristic algorithms such as Genetic Algorithm [8]–[11], Particle Swarm Optimization (PSO) [12]–[14], Artificial Bee Colony [15], Differential Evolution (DE) [16], [17], and Ant Colony Optimization [18]. However, the main drawback of optimization algorithms is the need for parameter tuning for each problem instance that is dealing with. Because of the fact that offline parameter tuning is an optimization problem itself and is mostly a time-consuming task [19], adaptive tuning or parameter control techniques can be used to overcome this issue.

In this study, self-adaptive differential evolution (SADE) [20], which is one of the main adaptive metaheuristic algorithms in continuous optimization domain, is used to solve the connected target coverage optimization in WSNs for the first time in the literature. The effectiveness of SADE has been analyzed with an experimental study that is based on a simulation environment set up. In addition, the performance of SADE has been compared with other metaheuristic optimization algorithms in the literature. The results obtained showed the effectiveness of using SADE for the purpose of solving the connected target coverage problem for WSNs.

The organization of the remaining sections of this paper is as follows. Section II formulates the connected target coverage problem in WSNs. Then, after reviewing the classical DE and SADE algorithms briefly, Section III defines how to use the SADE algorithm to solve this problem. After that, Section IV clearly explains the simulation environment that has been used to analyze the performance of the algorithm. Finally, Section V concludes the paper and gives some future work directions.

OSMAN GOKALP, is with Department of Computer Engineering, Ege University, Izmir, Turkey,(e-mail: osman.gokalp@ege.edu.tr).

https://orcid.org/ 0000-0002-7604-8647

Manuscript received September 25, 2019; accepted September 25, 2020.
DOI: 10.17694/bajece.624527

II. Problem Definition

Let $T = \{t_1 t_2, \dots, t_n\}$ is a set of targets, $S = \{s_1 s_2, \dots, s_k\}$ is a set of sensors. Also, all sensors are identical with each other and each one is capable of sensing objects within its sensing range, r_s. Assume that the elements of S and T are placed on an obstacle-free 2D Euclidean plane. When a target t_i is positioned at (x_i, y_i) and a sensor s_j is positioned at (x_j, y_j). Coverage of target t_i by sensor s_j is calculated as:

$$c(t_i, s_j) = \begin{cases} 1, if\ d(t_i, s_j) \le r_s \\ 0,\ otherwise \end{cases} \tag{1}$$

Where, $d(t_i, s_j)$ is the Euclidean distance between target t_i and sensor s_j, i.e. $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

A target is accepted as covered if it is sensed by at least one sensor:

$$cov(t_i) = \begin{cases} 1,\ if \sum_{j=1}^{k} c(t_i, s_j) \ge 1 \\ 0,\ otherwise \end{cases} \tag{2}$$

Another important property of each sensor is its communication range r_c that two sensors can communicate with each other only if the distance between them is within the r_c as:

$$con(s_i, s_j) = \begin{cases} 1, if\ d(s_i, s_j) \le r_c \\ 0,\ otherwise \end{cases} \tag{3}$$

A solution $S = \{s_1 s_2, \dots, s_k\}$ that offers a deployment plan for all sensors is valid if it is connected, i.e. there is at least one communication path between all pairs of sensors.

The goal of the problem is to find a valid (connected) solution that maximizes the target coverage ratio which is expressed as below:

$$CR = \frac{\sum_{i=1}^{n} cov(t_i)}{n} \tag{4}$$

Fig 1 demonstrates the example of WSN deployment with 3 sensors and 5 targets. Sensors s_1 and s_2 are connected with each other since they are positioned inside their r_c. On the other hand, sensor s_3 is not connected with s_1 and s_2, because both of them are outside of its r_c. So, this is an example of an unconnected WSN deployment plan. In addition, 3 of the 5 targets are covered since they are located within r_s of at least one of the sensors, whereas the remaining 2 are uncovered as they cannot meet this condition.

III. Solving the Problem with SADE

A. Review of DE and SADE Algorithms

DE [21] is a population-based metaheuristic algorithm that is used to solve continuous optimization problems. Each individual (solution) in the population is a vector of real numbers $\vec{X}_i = x_1, x_2, \dots, x_D$, where $i = 1,2, \dots, NP$ is a solution index, NP is the size of a population, and D is a dimension of a problem. The initial population is generally constructed randomly within the given upper and lower bounds. Then, at each iteration of the search, the following operations are done in order:

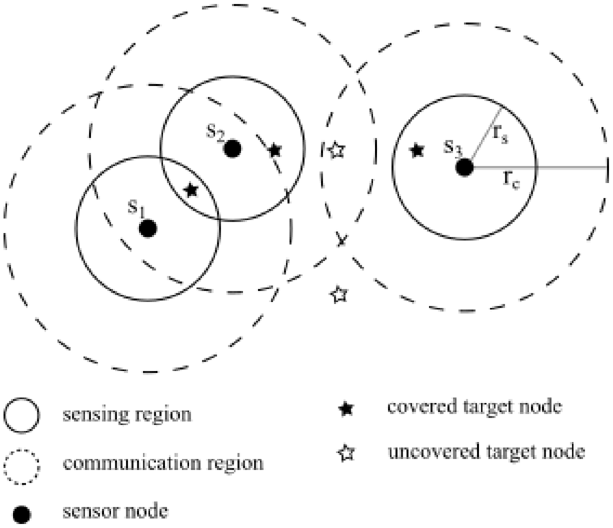


Fig.1. A WSN deployment example with 3 sensors and 5 targets

- **Mutation**: A mutant solution $\vec{V}_i$ is generated for each individual $\vec{X}_i$, which is also called a target vector, using a mutation strategy. In the basic DE algorithm, DE/rand/bin strategy is applied as follows:

$$\vec{V}_i = \vec{X}_{r1} + F(\vec{X}_{r2} - \vec{X}_{r3}) \tag{5}$$

Where i is the solution index, F is the differential weight, and $r1, r2$, and $r3$ are random solution indices that are different from each other as well as from i. Therefore, NP must be at least 4 to employ this strategy in the DE algorithm.

- **Crossover**: In crossover operation, each dimension j of the target vector i is tried to be changed and the trial vector $\vec{U}_i$ is generated as follows:

$$\vec{U}_{i,j} = \begin{cases} \vec{V}_{i,j},\ if\ r_{i,j} \le\ COR\ or\ j = I \\ \vec{X}_{i,j}, otherwise \end{cases} \tag{6}$$

Where, $I \in [1, D]$ is the randomly selected dimension index which is used to guarantee the change of at least one of the dimensions, COR is the crossover rate, $r_{i,j}$ is a random number that is selected uniformly in $[0,1]$.

- **Selection**: A trial solution is replaced with the target solution if its fitness value is better:

$$\vec{X}_i = \begin{cases} \vec{U}_i, if\ (\vec{U}_i) \le\ f(\vec{X}_i) \\ \vec{X}_i,\ \ otherwise \end{cases} \tag{7}$$

The self-adapted differential evolution algorithm (SADE) algorithm [20] maintains multiple DE mutation strategies to address optimization problems in different characteristics. Specifically, it includes DE/rand-to-best/2/bin, DE/rand/2/bin, and DE/current-to-rand/1 strategies, which are shown in (8) – (10), respectively, in addition to DE/rand/1/bin strategy, which was previously shown in (5).

$$\vec{V}_i = \vec{X}_i + F\left(\vec{X}_{best} - \vec{X}_i\right) + F\left(\vec{X}_{r1} - \vec{X}_{r2}\right) + F(\vec{X}_{r3} - \vec{X}_{r4})(8)$$

$$\vec{V}_i = \vec{X}_{r1} + F\left(\vec{X}_{r2} - \vec{X}_{r3}\right) + F(\vec{X}_{r4} - \vec{X}_{r5}) \qquad (9)$$

$$\vec{V}_i = \vec{X}_i + K\left(\vec{X}_{r1} - \vec{X}_{r2}\right) + F(\vec{X}_{r3} - \vec{X}_{r4}) \qquad (10)$$

SADE can select among these 4 strategies adaptively using success and failure memory tables that store successful and unsuccessful applications. That is, the more the strategy leads to an improvement over the current best solution, the more it has a chance to be selected again at later iterations. Furthermore, SADE can also adaptively decide parameter COR by maintaining a list that stores the successful applications of the parameter values. Then, new values of the COR are generated randomly by setting the median value of the list as a mean value of a normal distribution. As for parameter F, the parameter value is determined randomly by using a normal distribution with mean=0.5 and standard deviation=0.3. The interested reader should refer [20] to see other implementation details of this algorithm.

B. *Designing Problem-dependent Parts of the Algorithm*

Although SADE is a general-purpose numerical optimization method, the structure of solution representation and the fitness function evaluation subjects are left to the algorithm designer. Therefore, this section clearly explains how these algorithm components are designed to solve connected target coverage optimization problems for WSNs.

1) *Representation of solutions*

A solution to the problem should provide the locations of each individual sensor. In this study, the solution representation is designed as in Fig. 2, where S_i represents the sensor i and $x_i$and y_i represent the 2D Euclidean coordinates of that sensor. So, the problem dimension $D = 2 \times k$, where k is the total number of sensors.

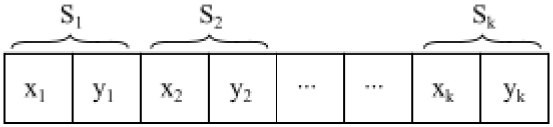


Fig.2. The solution representation

2) *Fitness function evaluation*

The fitness function evaluates the quality of a solution according to a given problem. In this study, the objective is to maximize the target coverage while preserving the overall network connectivity. So, the fitness function (maximization) is designed to handle these two conditions as in (11):

$$fit = W_1 \times CR + W_2 \times (1.0/CC) \qquad (11)$$

Where CR is the coverage ratio, CC is the connected component count (the number of unconnected sub-WSNs), W_1 and W_2 are the weighting factors. As it can be seen from the formula, higher values of CR are rewarded, whereas higher values of CC are penalized. Actually, the maximum fitness value is 2.0 and it is obtained when the full coverage of target points (e.g. $CR = 1.0$) and full connectivity (e.g. $CC = 1$) are both achieved.

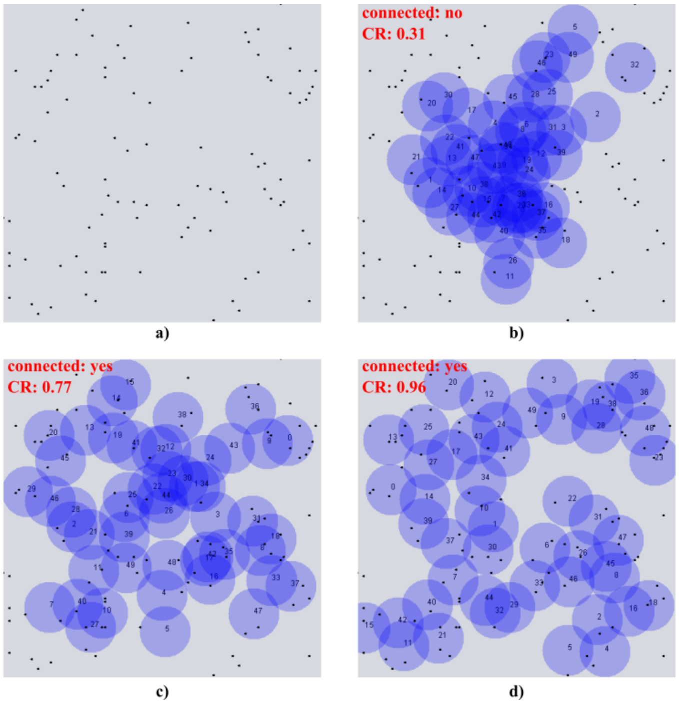
The solution will be useless if it does not provide full connectivity, regardless of how high coverage it provides. So, in order to ensure that the connectivity is achieved first, the fitness function is divided into two phases. In the first phase, the weight settings of $W_1 = 0.0$ and $W_2 = 1.0$ are used for making the algorithm work for only the connectivity purpose. After the connectivity is obtained, the weight settings of $W_1 = 1.0$ and $W_2 = 1.0$ are used for making the algorithm focus on increasing the coverage rate while preserving the connectivity.

IV. Simulation Results

A. *Simulation Environment*

To analyze the performance of SADE for solving the connected target coverage optimization problem in WSNs, the following simulation environment was set up on a computer with Intel® Core™ i7 6700 3.40 GHz CPU using a single core:

- simulation area boundaries: 100 x 100 unit^2
- k (# of sensors): 50, n (# of targets): 100
- deployment of targets: random
- r_s (sensing range): 8
- r_c (sensing range): {6, 8, 10, 12, 14}



a) b) c) d)

Fig.3. An example simulation steps for r_c=12. a) initial random positions of target points b) sensor positions after 500 FEs c) sensor positions after 50,000 FEs d) sensor positions after 250,000 FEs

For the implementation of the SADE algorithm, the guidelines in [20] were followed and the NP was set to 50. In addition, the maximum number of function evaluations (FEs) were set to $k \times 5{,}000$ for each run of the algorithm.

Fig. 3 shows an example of simulation steps with $r_c = 12$. Fig. 3-a shows the positions of target points that were placed randomly in the simulation area. Then, the sensor positions that were produced by the SADE algorithm at different FEs counts are shown in Fig. 3 b-d. It is seen that in earlier iterations of the algorithm, connectivity was not achieved yet. As the iterations passed, the connectivity was provided, and CR was improved.

B. Numerical Results

In this section, the numerical simulation results that were obtained with the SADE algorithm to solve the target coverage problem are provided and discussed. For this purpose, coverage and connectivity performances of the algorithm were measured with different r_c values.

The results obtained for 10 independent runs are presented in Table 1 that CR (Avg.) gives the average coverage rate, CR (S.D.) gives the standard deviation of coverage rates, and "# of succ." gives how many times the algorithm could produce a feasible solution (i.e. fully connected deployment plan).

It is seen from the table that the algorithm is very successful at producing connected solutions with 10/10 success rate, regardless of the communication ranges. The simulation results also reveal that as the communication range is increased the algorithm can produce solutions with higher CR values because it can cover more target points without violating the connectivity constraint. Indeed, when $r_c = 6$, which implies $r_c < r_s$, the CR is around 0.37. It increases to 0.57 when $r_c = r_s$ and can achieve 0.91 with $r_c \geq 1.5 \times r_s$.

TABLE I
SIMULATION RESULTS FOR VARIOUS COMMUNICATION RANGE VALUES

$r_c = 6$	CR (Avg.)	0.37
	CR (S.D.)	0.04
	# of succ.	10/10
$r_c = 8$	CR (Avg.)	0.57
	CR (S.D.)	0.04
	# of succ.	10/10
$r_c = 10$	CR (Avg.)	0.75
	CR (S.D.)	0.03
	# of succ.	10/10
$r_c = 12$	CR (Avg.)	0.91
	CR (S.D.)	0.03
	# of succ.	10/10
$r_c = 14$	CR (Avg.)	0.99
	CR (S.D.)	0.01
	# of succ.	10/10

C. Comparison with Other Optimization Algorithms

This section compares the performance of the SADE algorithm with other common metaheuristic algorithms in the continuous optimization domain. For this purpose the classical version of DE, which uses DE/rand/1/bin strategy, Artificial Bee Colony (ABC) [22], and Particle Swarm Optimization (PSO) [23].

In the experimental work, the population sizes of these 3 algorithms are determined as $NP = 50$, as they will be the same with the SADE algorithm. Also, the results were collected after 10 independent runs per algorithm with the same simulation environment that was defined in Section IV-A. After some preliminary testing, the following parameter values were used for the algorithms compared. For the DE algorithm, $COR = 0.3$ and $F = 0.5$; for ABC algorithm, $limit = (dim \times NP/2)$; and for PSO algorithm $w = 1.0/(2.0 * log_2)$, $c1 = c2 = 0.5 + log_2$, and $K = 3$.

Using the experimental setting above, the computational results obtained are presented in Table 2. One of the major findings is that SADE and PSO are both good at meeting the connectivity condition by producing 10/10 success rates for each r_c value. However, SADE outperforms PSO in terms of coverage rates in all cases. The results also reveal that SADE outperforms basic DE in terms of both CR and success rates. It can be seen that ABC can produce high CR values which are close to SADE, however, this is only valid for higher r_c values. Indeed, the ABC algorithm could not produce feasible solutions with $r_c \leq r_s$. As for the processing times of the algorithms, it is seen that SADE, DE, ABC, and PSO spend around 6.48 s., 6.53 s., 5.36 s., and 9.22 s., respectively. These results suggest that the SADE, DE, and ABC can produce results faster than PSO. Together, the results indicate that the SADE algorithm is preferable to solve the connected target coverage problem in comparison with its competitors that are considered in this experimental study.

TABLE II
COMPARISON THE PERFORMANCE OF SADE WITH OTHER COMMON METAHEURISTICS

		SADE	DE	ABC	PSO
$r_c = 6$	CR (Avg.)	**0.37**	0.28	N/A	0.22
	CR (S.D.)	0.04	0.02	N/A	0.03
	# of succ.	**10/10**	5/10	0/10	**10/10**
	Time (Avg. s.)	6.71	6.91	5.38	8.86
$r_c = 8$	CR (Avg.)	**0.57**	0.40	N/A	0.32
	CR (S.D.)	0.04	0.04	N/A	0.05
	# of succ.	**10/10**	8/10	0/10	**10/10**
	Time (Avg. s.)	6.54	6.87	5.85	9.50
$r_c = 10$	CR (Avg.)	**0.75**	0.53	0.71	0.42
	CR (S.D.)	0.03	0.03	0.09	0.06
	# of succ.	**10/10**	**10/10**	7/10	**10/10**
	Time (Avg. s.)	6.42	6.28	5.30	9.02
$r_c = 12$	CR (Avg.)	0.91	0.65	**0.93**	0.55
	CR (S.D.)	0.03	0.03	0.01	0.05
	# of succ.	**10/10**	**10/10**	**10/10**	**10/10**
	Time (Avg. s.)	6.35	6.30	5.27	8.89
$r_c = 14$	CR (Avg.)	**0.99**	0.76	**0.99**	0.67
	CR (S.D.)	0.01	0.02	0.01	0.06
	# of succ.	**10/10**	**10/10**	**10/10**	**10/10**
	Time (Avg. s.)	6.39	6.29	5.02	9.81

V. Conclusion

This paper uses the SADE algorithm to solve the connected target coverage optimization problem in WSNs for the first time in the literature. For this purpose, a fitness function with two stages that consider both connectivity and coverage rate is developed. To measure the performance of the SADE algorithm for solving this problem, the simulation environment was built, and the numerical results were obtained. Based on the experimental settings employed, the numerical results show that SADE is highly successful for dealing with the connected target coverage problem and can provide better performance in comparison with other common metaheuristic algorithms such as classical DE, ABC, and PSO.

Future work might extend this study by considering other variants of self-adaptive differential evolution algorithms such as [24] and [25]. In addition, the problem can be generalized to k-connected coverage variant in which targets must be covered by at least k sensors.

References

[1] A. Milenković, C. Otto, and E. Jovanov, “Wireless sensor networks for personal health monitoring: Issues and an implementation,” *Computer Communications*, vol. 29, no. 13–14, pp. 2521–2533, Aug. 2006.

[2] L. Lamont, M. Toulgoat, M. Deziel, and G. Patterson, “Tiered wireless sensor network architecture for military surveillance applications,” in *The Fifth International Conference on Sensor Technologies and Applications, SENSORCOMM*, 2011, pp. 288–294.

[3] M. A. Jan, P. Nanda, X. He, and R. P. Liu, “A Sybil attack detection scheme for a forest wildfire monitoring application,” *Future Generation Computer Systems*, vol. 80, pp. 613–626, Mar. 2018.

[4] W. Yi *et al.*, “A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems,” *Sensors*, vol. 15, no. 12, pp. 31392–31427, Dec. 2015.

[5] Chih-Yu Lin, Wen-Chih Peng, and Yu-Chee Tseng, “Efficient in-network moving object tracking in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, Aug. 2006.

[6] S. Abdollahzadeh and N. J. Navimipour, “Deployment strategies in the wireless sensor network: A comprehensive review,” *Computer Communications*, vol. 91–92, pp. 1–16, Oct. 2016.

[7] I. Khoufi, P. Minet, A. Laouiti, and S. Mahfoudh, “Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges,” *International Journal of Autonomous and Adaptive Communications Systems*, vol. 10, no. 4, pp. 341–390, 2017.

[8] Yourim Yoon and Yong-Hyuk Kim, “An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks,” *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, Oct. 2013.

[9] T. E. Kalayci and A. Uğur, “Genetic Algorithm-Based Sensor Deployment with Area Priority,” *Cybernetics and Systems*, vol. 42[1] T. E, no. 8, pp. 605–620, Nov. 2011.

[10] S. Mnasri, A. Thaljaoui, N. Nasri, and T. Val, “A genetic algorithm-based approach to optimize the coverage and the localization in the wireless audio-sensors networks,” in *2015 International Symposium on Networks, Computers and Communications (ISNCC)*, 2015, pp. 1–6.

[11] S. K. Gupta, P. Kuila, and P. K. Jana, “Genetic algorithm approach for k -coverage and m -connected node placement in target based wireless sensor networks,” *Computers & Electrical Engineering*, vol. 56, pp. 544–556, Nov. 2016.

[12] X. Wang, S. Wang, and D. Bi, “Virtual Force-Directed Particle Swarm Optimization for Dynamic Deployment in Wireless Sensor Networks,” in *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 292–303.

[13] Q. Ni, H. Du, Q. Pan, C. Cao, and Y. Zhai, “An improved dynamic deployment method for wireless sensor network based on multi-swarm particle swarm optimization,” *Natural Computing*, vol. 16, no. 1, pp. 5–13, Mar. 2017.

[14] X. Wang, S. Wang, J.-J. Ma, X. Wang, S. Wang, and J.-J. Ma, “An Improved Co-evolutionary Particle Swarm Optimization for Wireless Sensor Networks with Dynamic Deployment,” *Sensors*, vol. 7, no. 3, pp. 354–370, Mar. 2007.

[15] C. Ozturk, D. Karaboga, and B. Gorkemli, “Artificial bee colony algorithm for dynamic deployment of wireless sensor networks,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 20, no. 2, pp. 255–262, 2012.

[16] S. Kundu, S. Das, A. V. Vasilakos, and S. Biswas, “A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks,” *Soft Computing*, vol. 19, no. 3, pp. 637–659, Mar. 2015.

[17] N. Qin and J. Chen, “An area coverage algorithm for wireless sensor networks based on differential evolution,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 155014771879673, Aug. 2018.

[18] W.-H. Liao, Y. Kao, and R.-T. Wu, “Ant colony optimization based sensor deployment protocol for wireless sensor networks,” *Expert Systems with Applications*, vol. 38, no. 6, pp. 6599–6605, Jun. 2011.

[19] A. E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, Jul. 1999.

[20] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[21] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[22] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007.

[23] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, “Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements,” in *2013 IEEE Congress on Evolutionary Computation*, 2013, pp. 2337–2344.

[24] J. Zhang and A. C. Sanderson, “JADE: Self-adaptive differential evolution with fast and reliable convergence performance,” in *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 2007, pp. 2251–2258.

[25] R. Tanabe and A. Fukunaga, “Success-history based parameter adaptation for Differential Evolution,” in *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2013, pp. 71–78.

BIOGRAPHIES

OSMAN GOKALP was born in 1987. He received the B.S., M.S. and Ph.D. degrees in computer engineering from Ege University in 2010, 2012 and 2018. From 2011 to 2012, he was a research assistant in Yasar University - Department of Software Engineering. Since 2014, he has been a research assistant in Ege University - Department of Computer Engineering. His main research interests include evolutionary computing, metaheuristics and optimization.