Particle Swarm Optimization for Wireless Sensor Network Lifespan Maximization

Souad Kamel
College of Computer Science and Engineering, University of Jeddah, Saudi Arabia
skamel@uj.edu.sa (corresponding author)

Abeer Al Qahtani
College of Computer Science and Engineering, University of Jeddah, Saudi Arabia
abera.stu@uj.edu.sa

Abdullah Saad Musaed Al-Shahrani
College of Computer Science and Engineering, University of Jeddah, Saudi Arabia
aslshahrani@uj.edu.sa

Received: 14 December 2023 | Revised: 1 March 2024 | Accepted: 9 March 2024

 DOI: https://doi.org/10.48084/etasr.6752

ABSTRACT

Despite the deployment of wireless sensor networks in diverse fields (health, environment, military applications, etc.) for tracking or monitoring, several challenges, such as extending the lifetime of the network under energy constraints, still need to be resolved. Lifetime is the operational time of the network during which it can perform dedicated tasks and satisfy the application requirements. The energy constraints dictate that the energy consumption of sensors should be minimized since in most cases the sensors are battery-powered. Various methods have been proposed to work around this problem using scheduling approaches. In this paper, particle swarm optimization-based scheduling was designed and implemented to maximize the lifetime of wireless sensor networks formulated as a Non-Disjoint Sets Cover (NDSC) problem. The experimental findings show that the proposed approach is extremely competitive to the state-of-the-art algorithms, as it is able to find the optimal and best-known solutions in the instances investigated.

Keywords-scheduling; target coverage problem; non-disjoint set covers; wireless sensor networks; lifespan; particle swarm optimization

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are a type of wireless and ad hoc network. WSNs combine sensing, processing, and networking over miniaturized sensor nodes (often hundreds or thousands). They are typically deployed to monitor large or hazardous areas [1]. WSNs offer some significant advantages, including the fact that they are less expensive to deploy than wired networks. Sensor nodes can be added and withdrawn with ease. Furthermore, the node's location can be modified without rewiring. Finally, WSNs can be configured into different network topologies (star, tree, mesh, etc.). Despite the distinct features of WSNs, particularly their simplicity and effective cost, they have an anomalous character related to their resource restrictions in terms of computing power and energy. In fact, in most cases, the sensor nodes are powered by batteries. Manual configuration, maintenance and battery replacement are often impossible. So, to overcome these energy constraints and to save energy consumption and therefore extend the network lifetime, many methods have been proposed, such as optimal deployment, clustering, multi-hop routing, data aggregation, energy harvesting, and sleep-wake scheduling [2-4]. Much research is conducted in the direction of scheduling sensor activities. The deployed sensors are divided into a number of sensor sets each of which can cover all the targets and can send all the sensed data to the base station. These sensor sets can be disjoint or non-disjoint and are activated successively one by one. In each round, only one set is active. Only sensors in the active set are used to collect data from the surrounding environment and to relay them to the sink, whereas all the other sensors go into energy-saving sleep mode. Thus, each node or a cover of nodes should save as much power as possible by turning off the radio transmission when there is nothing to transmit. The process of clustering sensor nodes into cover sets and scheduling them to maximize the lifespan of the network belongs to the NP-hard problems. This kind of problem can be solved using either exact methods, like linear programming and branch-bound or metaheuristics, namely genetic, invasive weed optimization, and Particle Swarm Optimization (PSO) algorithms [4-8]. The former approach requires an exponential computation time depending on the size of the problem to be solved, while the latter tries to obtain good solutions (not necessarily optimal) in a reasonable time. We focused on utilizing PSO since it has been proved to be effective in tackling a plethora of problems and it has not been investigated enough in extending the lifetime of WSNs.

In this paper, a PSO algorithm is used to solve WSN lifespan maximization problem, formulated as a non-disjoint set cover problem.

II. RELATED WORKS

This section reviews the related work on scheduling sensor activities to maximize network lifetime under application requirements that include target coverage. Lifetime is the operational time of the network during which it is able to perform the dedicated tasks while the coverage is a measure of the physical space the sensors are able to observe. Authors in [9] presented a greedy algorithm for solving the maximum lifetime coverage problem and the energy consumed in the sleep/active schedule. An empty blacklist is created when constructing a new set cover. Iteratively, a minimal energy sensor, which is not present in the blacklist and covers at least a new target is chosen. When there are many sensors with minimal energy, the selected sensor will be the one that covers the largest number of new targets. Unfortunately, this approach is not suitable for homogeneous networks. Authors in [10] suggested an improved cuckoo search algorithm that partitions sensors into a maximum number of NDSC with a sensing range that can be adjustable. Each cover provides k-target coverage to maximize the lifetime of the WSN. After calculating the upper bounds of the maximum number of NDSC, the authors find the covers that will be scheduled and activated one by one. Unfortunately, the k-target coverage consumes more energy than the single-target coverage. Authors in [11] proposed the bidirectional mutation hybrid Genetic Algorithm (GA) to find the maximum NDSC that completely covers all the targets and maximizes the lifetime of the WSN. This algorithm differs from the traditional GA in the chromosomes representation and in the use of a greedy technique for initialization rather than the random initialization. Also, a novel bidirectional mutation was utilized to speed up the convergence. The proposed algorithm is suitable for heterogeneous WSNs that contain sensors with different levels of initial energy. Authors in [12] recommended an exact method (LP formulation) and two polynomial-time greedy heuristics for target coverage problem in Directional Sensor Networks (DSNs). Their aim was to balance between maximizing DSN lifetime and fault tolerance. They provide optimal solutions when using linear programming but with high cost. However, at a lower cost, with the use of heuristics, the solutions obtained were suboptimal. Authors in [13] proposed two methods to extend the lifetime of WSNs adopting the NDSC approach rather than DSC. First, they used the binary coverage relations matrix to sort the randomly deployed sensors nodes and to find the binary relations that link sensors and targets to construct a maximum number of non-disjoint set covers. These obtained covers are scheduled utilizing an exact method and a GA with a novel gene coding to get a near-optimal solution in reasonable time. The exact method gives the optimal solution, but requires a lot of computational time. Also, the metaheuristic employed is not efficient in a network that has a large number of sensors. Authors in [14] suggested a column generation method that finds the maximum NDSC to extend the lifetime of WSN. It is an exact algorithm that seeks to discover valid covers by a new integer linear programming model. The process is repeated iteratively and stopped only if there is no column with a positive reduced cost. The computation cost is decreased through the use of a branch-and-cut method. This reaches the best solution, but it is costly. Authors in [15] recommended a novel scheduling called Energy-Efficient Connected Coverage (EECC). EECC increased the level of coverage and connectivity of the sensors. It considers the remaining energy of each sensor and tries to avoid redundant coverage of critical points in the monitoring area. Unluckily, EECC was only effective with homogeneous WSNs containing sensors with the same initial energy. Authors in [16] proposed a novel local wake-up scheduling based on ant colony optimization. They constructed a first layer which contains a set of active sensors that completely cover the targets. Then, they manufactured multiple successor sets to mitigate the problem that some sensors in the first layer set run out of energy. This approach is effective, but it is suitable only for small and medium-scale networks.

Complexity theory classifies most scheduling problems as NP-hard [5]. This justifies the use of meta-heuristics (approximate methods), which provide acceptable (not necessarily optimal) solutions in a reasonable time. In this paper, PSO is used to solve the problem of randomly deployed sensors network lifespan maximization formulated as a scheduling problem. PSO has been proved to be an effective method for many optimization problems, and in some cases, it does not suffer from the difficulties experienced by other metaheuristics.

III. PROBLEM FORMULATION

The issue of WSN lifetime optimization based on the maximum number of NDSCs and their optimal scheduling belongs to the NP-hard family [5]. It presents two difficulties. The first one is the assignment of each sensor to a cover set, and the second one is the scheduling of these cover sets to optimize the network lifetime. Data, constraints, and objectives of WSN scheduling problem are defined below.

A. Data

- S represents a set of n sensor nodes. A sensor node is labeled as s_i $(i = 1, \dots, n)$.
- T represents a set of m targets. A target is labeled as t_j $(j = 1, \dots, m)$.
- s_i can monitor a subset of targets $T(s_i) \in T$ located in its coverage range r.
- Each target t_j could be monitored by a subset of sensors $S(t_j) \in S$.
- A collection of elements of S denoted C_l is a cover for the subset of targets denoted $T(C_l)$ if it can sense all the targets of $T(C_l) \in T$.
- C_l is considered as a cover if $T(C_l) = T$.
- Each sensor node S_i has an initial energy E_i.

- $E_i(k)$ is the total energy consumed by S_i during a period of time k.

B. Constraints

- Each target t_j is covered by at least one sensor of S.
- A sensor can be included into at most one set cover in the case of Disjoint Set Cover (DSC).
- A sensor can be included into more than one cover in the case of Non-Disjoint Set Covers (NDSC).

C. Criteria

- We have to maximize the lifespan, which is the time elapsed until all the available sensor nodes do not succeed to satisfy the targeted requirements. It can be expressed as [17]:

$$L = k \times \sum_{l=1}^{q} y_l \quad (1)$$

D. Example

To illustrate the benefit of using NDSC instead of DSC, let us assume that the three targets ($Target_1$, $Target_2$, and $Target_3$) in Figure 1 can be monitored by three sensor nodes ($Sensor_1$, $Sensor_2$, and $Sensor_3$).

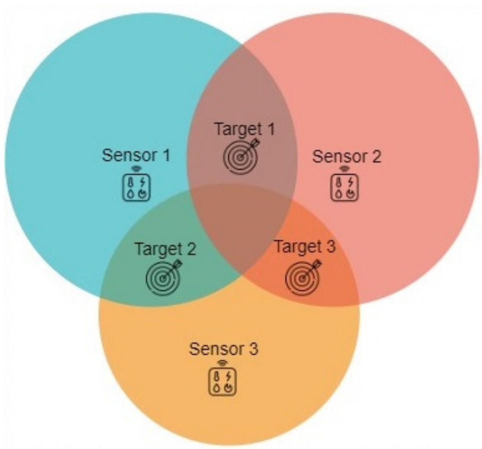

Fig. 1. Example of topology of three targets monitored by three sensors.

$Sensor_1$ covers $Target_1$ and $Target_2$. $Sensor_2$ covers $Target_1$ and $Target_3$, and $Sensor_3$ covers $Target_2$ and $Target_3$. If all the sensor nodes were to be activated simultaneously, then the network lifetime would be equal to the standard lifetime h of a single sensor. By dividing the sensors into disjoint sets, the resulting network lifetime would still be h, since for this topology a disjoint algorithm can only produce one cover set (e.g. $Cover_1 = \{Sensor_1, Sensor_2\}$ or $Cover_2 = \{Sensor_1, Sensor_3\}$, or $Cover_3 = \{Sensor_2, Sensor_3\}$). However, if a sensor node in Figure 1 can be part of two cover sets, then the network lifetime can be extended. By creating three non-disjoint cover sets (see Figure 2), each one activated for $0.5 \times h$ hours, the total network lifetime can be extended to $1.5 \times h$ (see Figure 3), assuming that the energy consumption during the sleep mode is negligible.

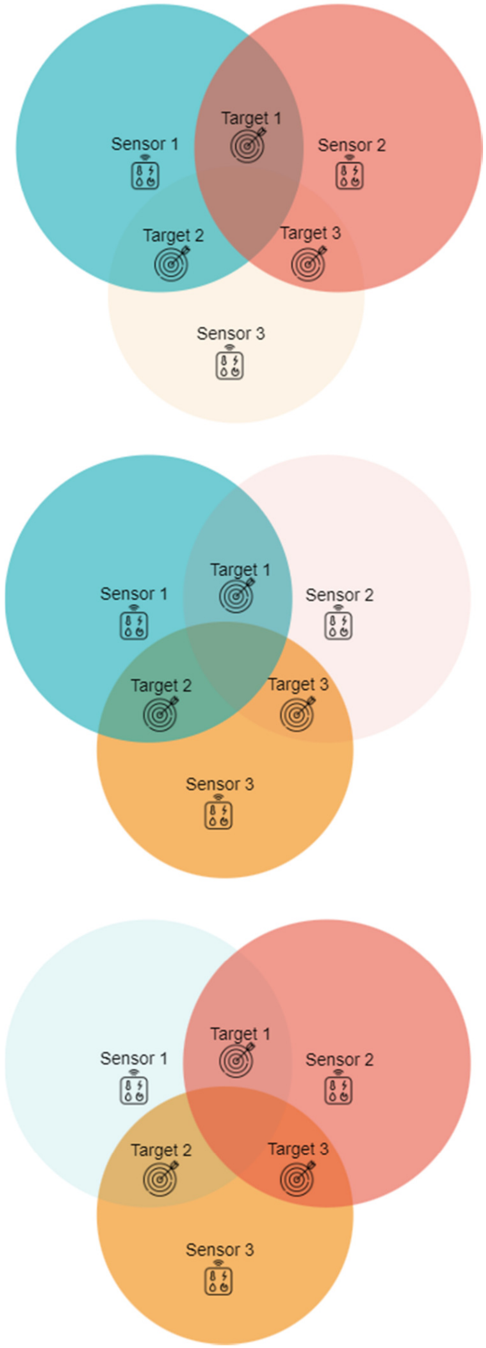

Fig. 2. Three obtained non-disjoint cover sets.

From the above observation, a sensor node can spend part of its energy within one cover and another part within another. So, finding the optimal lifespan requires solving two sub-problems: (1) finding the optimal number of NDSC and (2) maximizing lifespan by scheduling. This problem belongs to the NP-hard family and that is why metaheuristics are considered in solving it. This paper aims to develop a new PSO-based method that can efficiently find the maximum number of NDSC for a set of sensors that can be effectively scheduled to prolong the WSN lifespan.

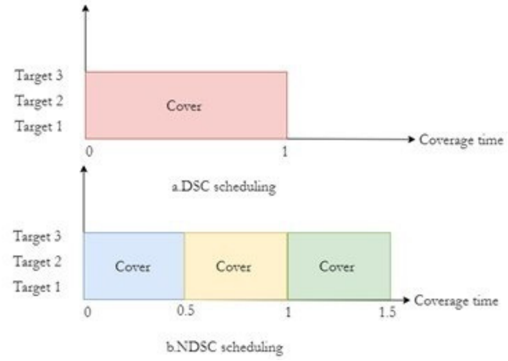

Fig. 3. GANTT in case of DSC and NDSC.

IV. PROPOSED APPROACH

Given that metaheuristic algorithms are competent in finding the best solution in an acceptable amount of time for especially hard problems, the following NDSC-based PSO is proposed to figure out the optimal number of non-disjoint cover sets and then to schedule them in order to maximize the lifespan of the network.

A. Particle Swarm Optimization

PSO was inspired by the social behavior of fish and birds. Each particle of the swarm (individual solution in a population), which is a candidate solution for the optimization problem, performs an individual search (cognitive search) and a global search to minimize an error function. PSO is built on the concept of progressively evolving a swarm of possible solutions to an optimization problem. Each particle has a position denoted by $X^i \in R^n, i = 1, \dots, n$, where n is the number of particles (swarm size), and a velocity to determine and move to the next particle denoted by $V^i \in R^n$. Every particle has also a fitness value to evaluate its quality. During the search process and over each iteration t, the particles move and update their velocities and positions according to (2) and (3) [18]:

$$V_{t+1}^i = \omega V_t^i + C_1 r_1 \left(p_t^i - X_t^i\right) + C_2 r_2 \left(g_t^i - X_t^i\right) \quad (2)$$

$$X_{t+1}^i = X_t^i + V_{t+1}^i \quad (3)$$

where ω is the inertia weight, C_1 and C_2 are the cognition learning and the social learning rates, respectively, r_1 and r_2 are uniformly random numbers in the range of $[0,1]$, p_t^i and g_t^i are the best personal and global solutions, equivalently, X_{t+1}^i is the modified position of the i^{th} particle, and t is the current iteration.

PSO can be used to solve problems with continuous and discrete variables. It is also applicable to multi-objective and constraint satisfaction problems. More details on PSO, its variants and applications, are referred to [16-26].

1) Coding

The first step in using PSO to solve any optimization problem is to map between the particle position of the swarm evolution concept and the special case of the investigated problem. For a set S of sensors used to monitor a set T of targets, the input of the PSO algorithm is the coverage relation matrix, which explains the targets $T(i)$ from T that are covered by each sensor s_i from S. The sum of the monitoring periods will be equal to the network lifespan. Figure 4 represents the structure of a given particle.

Fig. 4. A particle coding.

2) Initialization

The particle swarm (or population) can be either generated using a random initialization or can be issued from an output of any other heuristic method.

3) Position and Velocity Update

At each iteration t, the position and velocity of every particle (i.e. a potential solution of the considered optimization problem) are updated according to (2) and (3).

4) Fitness Evaluation

For each particle, the fitness value is the sum of the whole scheduled period y_l multiplied by the period *k*. It can be mathematically formulated as in (1).

5) Termination

The simplest condition of termination is to let the PSO simulation stop once it reaches a maximum number of iterations, which is predefined before starting the optimization process.

V. NUMERICAL RESULTS

The algorithms were coded implementing python programming language on Widows10 and with i5 8th gen processor.

A. PSO-based Method for finding the NDSC

This study simulated a set of randomly deployed sensors S = 10, 20, 30, …., 500, which have a constant sensing range equal to 3 and are used to monitor 5 targets in a 10 × 10 area. The hyperparameters of PSO are $C_1 = C_2$ =2 and ω =0.75. A different number of iterations and population sizes were investigated to find the NDSCs. Table I presents the number of NDSCs found engaging different small numbers of sensors utilized to monitor 5 targets, with number of iterations = 20 and population size = 500. It can be clearly noticed that the number of non-disjoint cover sets increases when the number of deployed sensors increases.

TABLE I. NDSCS FOR A SMALL NUMBER OF SENSOR NODES

Sensors	10	20	30	40	50
NDSCs	1397	2965	5556	5985	8224

Table II depicts the number of NDSCs for a large number of sensors used to monitor 5 targets, number of iterations = 20, and population size =500. Through the previous table it can be observed that the number of non-disjoint cover sets increases when the number of deployed sensors raises and especially when this number exceeds the range of hundreds (100, 200, 300,…500).

TABLE II. NDSCS FOR A LARGE NUMBER OF SENSOR NODES

Sensors	100	200	300	400	500
NDSCs	9778	9999	10000	10000	10000

Table III presents the effect of the population size on the results (using the same number of iterations = 20). It can be detected that the number of non-disjoint cover sets increases when the number of particles used in the swarm raises. This can lead to the conclusion that the number of the non-disjoint cover sets can be maximized, but this can influence the execution time of PSO.

TABLE III. NDSCS WITH DIFFERENT POPULATION SIZE

Population size	100	200	300	400	500	600
NDSCs	275	600	817	1101	1379	1613

Table IV provides the running time (ms) for different numbers of sensors with a different number of iterations. It can be spotted that for a fixed size of population, runtime raises when the number of deployed sensors increases. In the same direction, when the number of sensors were fixed and the size of the swarm was varied, the runtime increases.

TABLE IV. INSTANCES RUNTIME

Population Iterations	**100 sensors**	**200 sensors**	**300 sensors**	**400 sensors**
500	10	13	17.51	20.613
1000	19.84	25	34.25	41.185
1500	29.67	41.4	50.54	61.01
2000	40.5	53.6	69.09	78.92

B. PSO-based Method for finding the Optimal Scheduling

Table V compares the results obtained by the proposed NDSC-PSO approach and those provided by NDSC-GA and the exact method (ILP) on the same eight instances [13].

TABLE V. COMPARISON OF ILP, NDSC-GA, AND NDSC-PSO ALGORITHMS

Instances	**Lifetime**		
$\boldsymbol{S, T, r, E_i, E_i(k)}$	**ILP**	**GA**	**PSO**
5,5,3,160,16	20	20	20
5,5,3,160,8	40	40	40
5,5,3,160,4	80	80	80
5,5,3,160,2	160	160	160
10,5,3,160,8	30	30	30
10,5,3,160,4	60	**59**	**60**
10,5,3,160,4	120	**117**	**120**
10,5,3,160,2	240	**238**	**240**

In Table V, it can be observed that the results obtained by the proposed NDSC-PSO approach are better than those acquired with the NDSC-GA method. Indeed, the findings of NDSC-PSO reached the optimal solution for all the considered instances.

VI. DISCUSSION

After the conduction of many experiments, some conclusions can be drawn:

- To improve the acquired results, the swarm size could be enlarged, at the cost of execution speed.
- The second finding concerns the effect of the number of sensors deployed on the complexity of the problem being addressed.
- NDSC-PSO outperforms other effective approaches known in the literature.

The first and second observations can be considered obvious. The third one is the main contribution and the most important finding of the current study. In fact, in all instances tested in this work, the NDSC-PSO reaches the maximum lifetime. This result can be further generalized by testing other new instances that address the same kind of problem (WSN), but with different constraints and objectives (e.g. sensor type: fixed, mobile, random, homogeneous, heterogeneous, fully/partially connected, mono/multi criteria problem, etc.).

VII. CONCLUSION

In this paper, a NDSC-based PSO was investigated and integrated into a scheduling approach to maximize the lifespan of WSNs. The experimental results were implemented in the case of random deployment of sensor nodes. The obtained outcomes were compared with those derived through an exact method, namely the ILP model, and those acquired via a metaheuristic, namely NDSC-GA, known in the literature. For all the cases considered, the NDSC-PSO approach outperformed NDSC-GA and provided results close to those of ILP. The experimental findings are very encouraging as the proposed method is able to discover the best network lifetime for the studied instances. A more comprehensive study should be conducted on a larger number of cases. Further research could focus on additional constraints of the sensor network, such as the mobility and heterogeneity of nodes. Another future aim would be to test the usage of a fully adaptive PSO algorithm called TRIBES or the hybridization of PSO and other heuristics.

ACKNOWLEDGMENT

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-20-119-DR). The authors thank the University of Jeddah for the technical and financial support.

REFERENCES

[1] H. T. T. Binh and N. Dey, Eds., *Soft Computing in Wireless Sensor Networks*, 1st ed. Boca Raton, FL, USA: CRC Press, 2018.

[2] Y. Emami and R. Javidan, "An Energy-efficient Data Transmission Scheme in Underwater Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 6, no. 2, pp. 931–936, Apr. 2016, https://doi.org/10.48084/etasr.629.

[3] S. P. Singh and S. C. Sharma, "A Novel Energy Efficient Clustering Algorithm for Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 7, no. 4, pp. 1775–1780, Aug. 2017, https://doi.org/10.48084/etasr.1277.

[4] A. Rajab, "Genetic Algorithm-Based Multi-Hop Routing to Improve the Lifetime of Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7770–7775, Dec. 2021, https://doi.org/10.48084/etasr.4484.

[5] V. Kesavan, R. Kamalakannan, R. Sudhakarapandian, and P. Sivakumar, "Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system NP-hard problems: A comprehensive review," *Materials Today: Proceedings*, vol. 21, pp. 66–72, Jan. 2020, https://doi.org/10.1016/j.matpr.2019.05.363.

[6] S. Mekni, B. Chaar, and M. Ksouri, "A Novel Particle Swarm Optimization Approach for Multiobjective Flexible Job Shop Scheduling Problem," in *ICINCO-ICSO 2008*, 2008.

[7] S. Mekni and B. C. Fayech, "A Modified Invasive Weed Optimization Algorithm for MultiObjective Flexible Jobe Shop Scheduling Problems," *Computer Science & Information Technology*, vol. 10, pp. 51–60, Nov. 2014, https://doi.org/10.5121/csit.2014.41106.

[8] S. Mekni and B. Chaar Fayech, "Multiobjective Flexible Job Shop Scheduling Using A Modified Invasive Weed Optimization," *International Journal on Soft Computing*, vol. 6, no. 1, pp. 25–36, Feb. 2015, https://doi.org/10.5121/ijsc.2015.6103.

[9] N. T. Hanh, H. T. T. Binh, N. V. Son, N. T. Trang, and P. N. Lan, "Optimizing wireless sensor network lifetime through K-coverage maximization and memetic search," *Sustainable Computing: Informatics and Systems*, vol. 40, Dec. 2023, Art. no. 100905, https://doi.org/10.1016/j.suscom.2023.100905.

[10] D. Arivudainambi, R. Pavithra, and P. Kalyani, "Cuckoo search algorithm for target coverage and sensor scheduling with adjustable sensing range in wireless sensor network," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 4, pp. 975–996, May 2021, https://doi.org/10.1080/09720529.2020.1753301.

[11] J. Li, Z. Luo, and J. Xiao, "A Hybrid Genetic Algorithm With Bidirectional Mutation for Maximizing Lifetime of Heterogeneous Wireless Sensor Networks," *IEEE Access*, vol. 8, pp. 72261–72274, 2020, https://doi.org/10.1109/ACCESS.2020.2988368.

[12] V. R. Ekhlas, M. Hosseini Shirvani, A. Dana, and N. Raeisi, "Discrete grey wolf optimization algorithm for solving *k*-coverage problem in directional sensor networks with network lifetime maximization viewpoint," *Applied Soft Computing*, vol. 146, Oct. 2023, Art. no. https://doi.org/10.1016/j.asoc.2023.110609.

[13] Y. E. E. Ahmed, "Modeling, Scheduling and Optimization of Wireless Sensor Networks lifetime," Ph.D. dissertation, Université de Lorraine, Lorraine, France, 2016.

[14] M. E. Keskin, "A column generation heuristic for optimal wireless sensor network design with mobile sinks," *European Journal of Operational Research*, vol. 260, no. 1, pp. 291–304, Jul. 2017, https://doi.org/10.1016/j.ejor.2016.12.006.

[15] D. K. Sah, S. Srivastava, R. Kumar, and T. Amgoth, "An energy efficient coverage aware algorithm in energy harvesting wireless sensor networks," *Wireless Networks*, vol. 29, no. 3, pp. 1175–1195, Apr. 2023, https://doi.org/10.1007/s11276-022-03125-3.

[16] M. Suresh Kumar and G. A. Sathish Kumar, "Enhanced ant colony optimization algorithm for packet delivery with improved energy efficiency in wireless sensor networks," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 5, pp. 7909–7917, Jan. 2023, https://doi.org/10.3233/JIFS-221856.

[17] Y. E. E. Ahmed, K. H. Adjallah, R. Stock, I. Kacem, and S. F. Babiker, "NDSC based methods for maximizing the lifespan of randomly deployed wireless sensor networks for infrastructures monitoring," *Computers & Industrial Engineering*, vol. 115, pp. 17–25, Jan. 2018, https://doi.org/10.1016/j.cie.2017.09.049.

[18] M. Clerc, *Particle Swarm Optimization*, 1st ed. Hoboken, New Jersey, USA; London, UK: Wiley-ISTE, 2006.