Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research Paper

Dynamic Sensors Assignment to Improving Lifetime Wireless Sensor Networks

Ali Abdi Seyedkolaei*

Department of Computer Engineering, University of Mazandaran, Babolsar, Iran.

Article Info

Article History:
Received 09 December 2024
Revised 17 January 2025
Accepted 25 February 2025

DOI:10.22044/jadm.2025.15440.2655

Keywords:
Wireless Sensor Network, Sink Placement, Dynamic Assignment.

**Corresponding author: a.abdi@umz.ac.ir (A. Abdi Seyedkolaei).*

Abstract

Deploying multiple sinks instead of a single sink is one possible solution to improve the lifetime and durability of wireless sensor networks. Using multiple sinks leads to the definition of a problem known as the sink placement problem. In this context, the goal is to determine the optimal locations and number of sink nodes in the network to maximize the network's lifetime. In this paper, we propose a dynamic sensor assignment algorithm to address the sink placement problem and evaluate its performance against existing solution methods on a diverse set of instances. We conducted experiments in two stages. In the first stage, based on random instances and compared to the exact computational method using the CPLEX solver, and in the second stage, based on real-world instances compared to MC-JMSP (Model-Based Clustering- Joint Multiple Sink Placement) method. The results obtained in the first stage of the experiments indicate the superiority of the dynamic sensor assignment algorithm in runtime for all instances. Furthermore, the solution obtained by the dynamic sensor assignment algorithm is very close to the solution obtained by the CPLEX solver. In particular, the percentage error of the solution found by the proposed method compared to CPLEX in all experimented instances is less than 0.15%, indicating the effectiveness of the proposed method in finding the appropriate solution for assigning sensors to sinks. Also, the results of the second stage experiments show the superiority of the proposed method in both execution time and energy efficiency compared to the MC-JMSP method.

1. Introduction

Wireless sensor networks consist of a large number of sensor nodes that are dispersed in an environment. The primary function of these nodes is to gather information about their environment and wirelessly relay it to a central node, referred to as the sink. This information can include temperature, light, humidity, and other factors. Wireless sensor networks have characteristics, properties, and limitations that distinguish them from other networks. Among these limitations are energy consumption, processing speed, data storage capacity, and communication bandwidth. Wireless sensor networks can be implemented in applications such as target tracking, environmental monitoring, agriculture, industry, and military uses. Figure 1 illustrates the architecture of wireless sensor networks.

Given that sensors require energy consumption to receive information, energy efficiency by sensor nodes is considered a significant challenge in these networks. This is because nodes are randomly deployed in the network, and their distribution in inaccessible locations can lead to excessive energy consumption, putting them at risk of failure. When a node shuts down, its communication with other nodes will be interrupted, and the information received from the surrounding environment will be incomplete. Consequently, one of the main challenges in wireless sensor networks is to enhance the network's longevity by improving energy efficiency. For this purpose, deploying multiple sinks instead of using a single sink can be an effective solution.

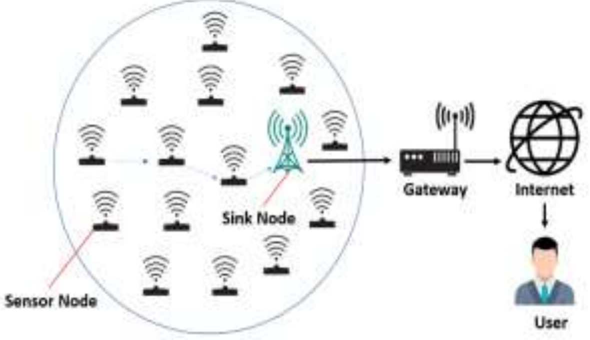

Figure 1. Wireless sensor network architecture.

Moreover, utilizing a single sink in the network has its drawbacks. For instance, network traffic in most areas will become excessively known, and the sink may enter an overflow state. Additionally, due to high energy consumption, the sink may also shut down. The use of multiple sinks leads to defining a problem known as the sink placement problem, where the number and placement of sinks become significant. Figure 2 illustrates an example of the sink placement architecture. As shown in Figure 2, the solid yellow nodes represent the sensors, the hollow circles denote potential sink locations, and the red circles indicate the selected sink locations.

Figure 2. An example of sinks placement architecture.

This paper aims to find the optimal placement and number of sink nodes in a way that meets defined constraints and prolongs the operational lifetime of the network. From a computational complexity perspective, the sink placement problem belongs to the class of NP-hard problems [3]. Our approach utilizes dynamic sensor assignment to tackle this issue. This algorithm employs the strategy of turning off active sinks to achieve an efficient network topology within an acceptable timeframe.
The proposed approach has significant practical implications across various domains, including smart cities, environmental monitoring, and industrial automation. In smart cities, WSNs (Wireless Sensor Networks) are used for applications like air quality monitoring, traffic management, and energy consumption optimization. By improving the energy efficiency and longevity of sensor networks, the proposed method reduces maintenance costs and ensures continuous service delivery in these urban applications.
In environmental monitoring, such as in agriculture, forest management, and climate change studies, the enhanced network lifetime enables long-term data collection, which is critical for ongoing analysis. The proposed solution minimizes energy consumption and ensures uninterrupted data flow, making it ideal for remote and inaccessible areas. Similarly, in industrial automation, the approach supports reliable monitoring of equipment and processes by reducing latency and ensuring consistent network performance, which is vital for operational efficiency and preventing downtime.
Overall, the proposed algorithm not only optimizes technical performance but also offers tangible benefits, such as reducing operational costs, enhancing reliability, and promoting environmental sustainability by decreasing electronic waste through extended network lifetimes. These advantages make the solution highly relevant for real-world applications. The main contributions of this paper are as follows:

- Using a novel algorithm for dynamically assigning sensors to sinks and selectively deactivating inefficient sinks to improve energy efficiency and extend network lifetime.
- Minimizing overall network energy consumption by reallocating sensors to the nearest active sink based on energy efficiency and balancing energy load across the network
- The proposed method is compared with the CPLEX solver and shown to provide solutions with an error of less than 0.15% in energy consumption, demonstrating efficiency and accuracy.
- The DSA (Dynamic Sensor Assignment) algorithm significantly reduces execution time compared to the exact computational method,

making it suitable for real-time applications in large-scale networks.

- The proposed method is scalable and can be effectively applied to various real-world scenarios such as smart cities, environmental monitoring, and industrial automation.

The structure of this paper is organized as follows: Section 2 reviews existing research related to the sink node placement problem. Section 3 explains the standard integer programming model for this issue. Section 4 introduces the proposed algorithm based on dynamic sensor allocation and examines its structure. Section 5 evaluates the performance of the proposed algorithm in comparison to a standard solver for integer programming models. Finally, Section 6 presents the results obtained from performance evaluation.

2. Related Works

This section reviews research conducted on the sink node placement problem. Jari and Avokh [4] focused on solving the sink placement problem. They implemented two algorithms, MPAR (Multi-sink Placement and Anycast Routing) and EMPAR (Extended Multi-sink Placement and Anycast Routing), within their proposed method, which concurrently addresses clustering challenges, the deployment of multiple sinks, and load-balanced routing among these sinks. Bouarourou et al. [5] modeled the deployment of multiple sinks and routing from sensors to sinks using clustering techniques to enhance the lifetime of wireless sensor networks. In their model, they identified the optimal locations for sinks and determined the most efficient routes for data transmission from sensors to these sinks. Their proposed method was inspired by ant clustering algorithms as an artificial intelligence approach.

Singh and Nagaraju [6] addressed the data transmission difficulties in multi-hop, multi-sink wireless sensor networks through network coding, aiming to minimize both delay and energy consumption. They utilized optimization algorithms such as network coding to improve network performance in terms of energy consumption, data transmission delay, and communication quality. Houssein et al. [7] applied Harris' Hawks optimization algorithm to address the sink node placement issue in large-scale wireless sensor networks. They also used Prim's shortest path algorithm to reconstruct the network by establishing minimum transmission paths from the sink node to other sensor nodes.

Yu et al. [8] introduced a method for concurrently determining the deployment of both sensors and sinks, aiming to minimize energy consumption while maximizing information effectiveness. In their research, they formulated constrained multi-objective optimization as a mixed-integer programming problem. Hanh et al. [9] examined the challenges associated with node deployment in wireless sensor networks with multiple sinks. They focused on how to deploy the minimum number of nodes to create a sensor network with multiple sinks. Al-Salti et al. [10] introduced a mathematical model for positioning multiple sinks in underwater wireless sensor networks, considering a three-dimensional mesh topology. The objective of this model was to reduce the total number of hops for each source-sink cell pair. Chen et al. [11] addressed the multi-sink placement problem with guaranteed delay and reliability in a wireless sensor network with data loss.

Tuba et al. [12] focused on determining the locations of multiple sinks with the aim of reducing energy consumption and increasing network lifetime. In their proposed method, they utilized brain storm optimization algorithms for sink placement. Sarwar and Chatterjee [13] investigated the optimal placement of multiple sinks in wireless sensor networks. The use of multiple sinks was validated as an efficient technique for extending network lifetime. They proposed distributed algorithms to determine the minimum number of required sinks and their optimal positions within the installation area while ensuring a specified delay and fault tolerance level. Furthermore, both random and deterministic sensor node installation strategies were explored in this research. Bose and Gurusamy [14] concentrated on solving the optimal multi-sink placement problem. They employed a Bacteria Foraging algorithm to determine the optimal positions of sinks. Their experimental results indicated that end-to-end delay was minimized and the average energy consumption of sensor nodes was reduced.

As shown in Table 1, the symbols '*' and '-' indicate whether a metric was considered or not in the research, respectively. Specifically, a metric marked with '*' in each row of Table 1 signifies that the research focuses on that particular metric, while '-' indicates the opposite.

3. Mathematical Model of the Problem

This section presents the mathematical framework for the sink placement problem, providing a precise characterization of the problem. The model encompasses a collection of sensors, a set of sinks, and various potential locations for sink deployment, denoted by the symbols N, S, and L, respectively.

The network is conceptualized as a graph, where each node represents either a sensor or a sink location. Specifically, a sensor node and a sink can be located next to each other at a single node. Each sensor is connected to only one of the sinks. By setting a connection capacity limit to the sink, multiple sensors can send data to a single sink. Additionally, due to the possibility of failures, each sink must maintain direct communication with other sinks. Consequently, the communication topology among the sinks is structured as a full mesh.

Table 1. Comparison of past research and our approach

Ref.	Proposed Method	Metrics: Energy	Lifetime	Reliability	Scalability	Dynamics
[4]	Using MPAR and EMPAR algorithms	*	*	-	-	-
[5]	Using the clustering technique	*	*	-	*	-
[6]	Using three different algorithms	*	*	-	-	-
[7]	Harris' Hawk Optimization algorithm	*	*	-	-	-
[8]	dual-population constrained multi-objective optimization algorithm	*	*	-	-	-
[9]	Using heuristic algorithms	*	*	-	-	-
[10]	Mathematical model and Partitioning Around Medoid approximation algorithm	*	*	-	-	-
[11]	Sink placement with guaranteed delay and reliability	*	*	*	-	-
[12]	Brain storm optimization algorithm	*	*	-	-	-
[13]	Distributed algorithms for multi-sink placement	*	*	*	-	-
[14]	Bacteria Foraging Algorithm	*	*	-	-	-
-	Use mathematical modeling and DSA algorithm	*	*	-	*	-

For each sensor $n \in N$, the number of packets transmitted to the sink is represented by a specific parameter σ^n. Each sink $s \in S$ has parameters α^s, μ^s, e^s and φ^s, which represent, respectively, the number of sensors that can connect to the sink, the number of packets that can be processed by the sink, the energy consumed by the sink, and the number of sinks of type *s*.

Each sensor node $n \in N$ consumes an amount of energy $E_{n(s)}$ to send *t* bits of data over a distance *d*, which is derived from Equation (1).

$$E_{n(s)} = \begin{cases} tE_{elect} + t\,\varepsilon_{fs} d^2 \quad , & d \le d_{co} \\ tE_{elect} + t\,\varepsilon_{mp} d^4 \quad , & d > d_{co} \end{cases} \tag{1}$$

The energy E_{elect} is required to activate the electronic circuits, while ε_{fs} and ε_{mp} represent the energy required to activate the power amplifier. The variable *d* calculates the Euclidean distance between each node, and d_{co} is a threshold limit that is approximately equal to 87 meters. Furthermore, the energy expended by the receiver to process *t* bits is derived from Equation (2). Finally, the total energy consumed by a node is calculated using Equation (3).

$$E_{n(r)} = tE_{elect} \tag{2}$$

$$E_n = E_{n(s)} + E_{n(r)} \tag{3}$$

Based on the definitions of sets and parameters, the decision variables for this problem are established:

$$X_{sl} = \begin{cases} 1 & \text{If sink } s \text{ is located at location } l, \\ 0 & \text{Otherwise,} \end{cases}$$

$$V_{nl} = \begin{cases} 1 & \text{If the connection between the sensor } n \text{ and the sink at location } l \text{ is established,} \\ 0 & \text{Otherwise,} \end{cases}$$

$$Z_{lk} = \begin{cases} 1 & \text{If there is a connection between place } l \text{ and place } k, \\ 0 & \text{Otherwise,} \end{cases}$$

The primary objective is to minimize energy consumption to extend the operational lifetime of the wireless sensor network. This energy encompasses that consumed by sinks, the energy used in communication between sensors and sinks, as well as the energy spent in interactions among sinks, denoted by symbols $E_s(X)$, $E_n(V)$, and $E_q(Z)$, respectively. The amount of energy consumed is for processing and transmitting information. Energy consumption is required for processing operations such as data extraction and processing, executing various algorithms, and performing calculations. Energy consumption is also needed for communication activities such as sending data, receiving data, managing data transmission media, and maintaining network connectivity. Equations (4), (5), and (6) describe how to calculate each of these energy consumptions.

$$E_s(X) = \sum_{s\in S} e^s \sum_{l\in L} X_{sl} \tag{4}$$

$$E_n(V) = \sum_{n\in N} \sum_{l\in L} E_n V_{nl} \tag{5}$$

$$E_q(Z) = \sum_{\substack{k\in L \\ k<l}} \sum_{l\in L} E_k Z_{lk} \tag{6}$$

The following section presents the objective function of the problem along with its associated constraints.

$$Minimize(E_s(X) + E_n(V) + E_q(Z)) \tag{7}$$

$$\sum_{s\in S} X_{sl} \le 1, \quad \forall l \in L \tag{8}$$

$$\sum_{k\in L} (Z_{lk} + Z_{kl}) + \sum_{n\in N} V_{nl} \le \sum_{s\in S} \alpha^s X_{sl}, \quad \forall l \in L \tag{9}$$

$$\sum_{l\in L} V_{nl} = 1, \quad \forall n \in N \tag{10}$$

$$\sum_{n\in N} \sigma^n V_{nl} \le \sum_{s\in S} \mu^s X_{sl}, \quad \forall l \in L \tag{11}$$

$$\sum_{l\in L} X_{sl} \le \varphi^s, \quad \forall s \in S \tag{12}$$

$$\sum_{s\in S} X_{sk} + \sum_{s\in S} X_{sl} \le Z_{lk} + 1, \tag{13}$$

$$(k < l, \forall k \in L, \forall l \in L)$$

$$X_{sl} \in \{0,1\}, \quad \forall s \in S, \forall l \in L$$

$$V_{nl} \in \{0,1\}, \quad \forall n \in N, \forall l \in L$$

$$Z_{lk} \in \{0,1\}, \quad \forall l \in L, \forall k \in L$$

Constraints (8) and (9) outline the restrictions on the number of sinks that can be deployed at each location, as well as the capacity limitations for sensors to connect to those sinks. Constraint (8) in the presented mathematical model states that only one sink can be deployed at each location. Constraint (10) indicates the connection limitation of sensors to sinks. This constraint specifies that each sensor must be connected to only one sink. Constraints (11) and (12) represent the limitation on the number of packets that can be processed by the sink and the limit on the number of available sinks for deployment, respectively. Constraint (13) shows the topology limitation of the sinks' connections, meaning that the topology among the sinks must be a complete mesh. Any binary assignment to the problem's variables that satisfies constraints (8) through (13) is termed a feasible solution to the problem, or simply a solution.

4. Proposed Algorithm: Dynamic Sensor Assignment

In this section, the Dynamic Sensor Assignment algorithm is used to solve the sink placement problem. This algorithm can select the best active sinks and assign sensors to them while turning off other sinks. The goal of this algorithm is to optimally allocate sensors to sinks in a way that reduces network energy consumption and increases network lifetime. The proposed algorithm can find an appropriate solution by searching through all $2^{|S|}$ subsets composed of sinks.

The steps of the DSA algorithm are as follows:

- *Initialization of Active Sinks*: The algorithm begins by selecting a subset of active sinks from the total available sinks. This is done by generating a random number (denoted as r) for each sink, and sinks with a random value $r>0.8$ are marked as active. The threshold (θ) for this process is set at 0.8.
- *Sensor Assignment to Active Sinks:* Once the active sinks are determined, each sensor is assigned to its nearest active sink based on the Euclidean distance and the energy consumption required for communication. This ensures that the sensor assignment is both energy-efficient and minimizes communication delays.
- *Evaluation of Active Sinks:* The next step involves evaluating whether any of the active sinks should be deactivated to reduce overall energy consumption. For each active sink, the algorithm checks if the sensors connected to it can be reassigned to other active sinks.

A sink will be turned off if the following conditions are met:

1. Reassignment Feasibility: The sensors connected to the sink can be reassigned to other active sinks without disrupting network performance.
2. Energy Optimization: Turning off the sink and reallocating its sensors leads to a decrease in total network energy consumption.

- *Reallocation of Sensors:* If a sink is to be turned off, the sensors assigned to that sink are reallocated to the other active sinks based on the same criteria—proximity to the sink and energy efficiency. After reassignment, the total energy consumption of the network is re-evaluated to ensure that the solution is still optimal or near-optimal.
- *Iteration and Convergence:* The process is repeated for a fixed number of iterations (denoted as k). During each iteration, the active

sinks are assessed, and sensors are reassigned as needed. The algorithm continues to adjust the active sinks dynamically until it converges to an optimal configuration.

4.1. Criteria for Decision Making:

- *Active Sink Selection:* The decision to activate or deactivate a sink is based on a random selection process (threshold r >0.8) and the energy consumption of sensors connected to the sink.
- *Sensor Reallocation:* Sensors are reassigned to active sinks that are closer or require less energy for communication. This is based on the Euclidean distance between the sensor and the sink, as well as the energy required to transmit data.
- *Sink Deactivation:* A sink is deactivated if its associated sensors can be reassigned to other sinks without increasing total energy consumption or network latency.

4.2. Assumptions:

- *Energy Consumption:* The energy consumption is primarily based on the distance between a sensor and the sink, as well as the energy required for transmission and reception of data. The assumption is that communication over shorter distances consumes less energy.
- *Fixed Number of Sinks:* The number of sinks is fixed at the start of the process, and the algorithm works with this predefined number of sinks to optimize their locations and energy usage.
- *Sensor Reallocation:* It is assumed that sensors can be easily reallocated to other active sinks without significantly impacting network performance. The algorithm ensures that each sensor is only connected to one sink at a time.

By dynamically assigning sensors to sinks and optimizing sink activation, the algorithm significantly improves the energy efficiency and operational lifetime of wireless sensor networks.
Finally, the details of the Dynamic Sensor Assignment algorithm are illustrated in Algorithm 1.
In each iteration from lines 1 to 19, active sinks are initially selected randomly from the set of sinks S, according to a designated r value. Sinks with an r value exceeding the threshold θ are designated as active sinks. Then, in each loop from lines 4 to 18, sensors are allocated to the active sinks based on the nearest distance and energy consumption. The active sink with the fewest connected sensors is selected. For all sensors served by this sink, it is checked whether they can be reallocated to other active sinks. If there are no sensors that can be transferred to other active sinks, this active sink will remain. Otherwise, it will be evaluated whether it can be turned off or not.

Algorithm 1 Dynamic Sensor Assignment (DSA) Algorithm

Input: $S, N, X_{sl}, V_{nl}, Z_{lk}, e^{s}, \alpha^{s}, \mu^{s}, \phi^{s}$
$\forall s \in S, \forall n \in N, \forall l \in L$
$\varepsilon_{fs}, \varepsilon_{mp}, d_{co};$

Output: $X^{*}_{sl}, V^{*}_{nl}, Z^{*}_{lk} \quad \forall s \in S, \forall n \in N, \forall l \in L, \forall k \in L;$

1: ***For*** *k = 1 … 1000* ***do***
2: *Compute* S'; // Select active sinks between set of sinks based on $r > 0.8$
3: *Best_Lifetime*$^{*} = \infty$, $S'' = S'$;
4: ***While*** $S'' \neq 0$ ***do***
5: *Compute* $X_{sl}, V_{nl}, Z_{lk} \quad \forall s \in S''$, $\forall l \in L, \forall k \in L, \forall n \in N;$
//allocation sensors with sinks
6: *Select* $S^{*} = \arg\min \sum_{n \in N} X_{sl} V_{nl}, \; \forall s \in S'', \forall l \in L;$
7: $S'' = S'' - \{S^{*}\};$
8: ***If*** *sensors connected to* S^{*} *can be moved to other sinks* ***then***
9: $S' = S' - \{S^{*}\};$
10: *Compute* X_{sl}, V_{nl}, Z_{lk}, $\forall s \in s'' \; \forall l \in L, \forall k \in L, \forall n \in N;$
11: *Compute Best_Lifetime;*
12: ***If*** *Best_Lifetime < Best_Lifetime** ***then***
13: *Best_Lifetime** = *Best_Lifetime,* $X^{*}_{sl} = X_{sl}, V^{*}_{nl} = V_{nl}, Z^{*}_{lk} = Z_{lk};$
14: ***else***
15: $S' = S' + \{S^{*}\};$
16: ***end if***
17: ***end if***
18: ***end while***
19: ***end for***

To assess each sink (lines 9 to 16), it is assumed that this sink is removed from the set of active sinks, and its sensors are reallocated to other active sinks based on the nearest distance and energy consumption. If the total energy consumption of the network decreases, then this active sink is turned off. Otherwise, it will remain active. This procedure continues until all active sinks have been evaluated for whether they can be turned off to reduce the overall energy consumption of the network. All steps for selecting active sinks and checking whether to turn them off or keep them active are executed in a loop for a total of k iterations. Thus, the DSA algorithm can quickly converge to the final optimal solution.

5. Simulation Results

This section compiles the experiments and calculations performed to assess the effectiveness of various methods for solving the placement problem. The experiments are conducted in two stages. In the first stage, based on random instances and compared with the exact computational method using CPLEX solver, and in the second stage, based on real-world instances compared with MC-JMSP method [5].

5.1. The First Stage Experiment:

In this regard, the execution time and solution quality metrics for each of the problem-solving methods using CPLEX [15] [16] [17] and the DSA algorithm are computed and reported in comparison with each other on a set of standard problem instances. The method for constructing the experimented instances is described as follows:

For each scenario, the network topology is randomly generated from a 20×20 grid network. This implies that each node within the specified grid has a probability p_r of being included as a node in the network graph, with p_r set to 0.25 for these instances. This means that each node from the specified grid will be a node in the network graph with probability p_r. The value of p_r is set to 0.25 for the instances. After selecting the designated nodes, the corresponding complete graph for the given network topology will be generated from the instance. The weight of each edge in the graph is determined by the Euclidean distance between its endpoints. The next step involves identifying nodes equipped with sensors on this graph. Specifically, for an instance with *i* sensors, *i* nodes are randomly chosen from the graph, and a sensor is installed at each selected node.

The parameter *i*, denoting the number of sensors installed in the network for constructing these instances, is selected from the set {5k | k=0, 1, 2, …, 20}.

The computations in this section were executed on a single-processor Intel Core i5 system running a Windows operating system with 8 GB of RAM. The dynamic sensor assignment algorithm was implemented using the C programming language. For CPLEX, a time limit of 3600 seconds was set. Additionally, due to the random selection of active sinks in the DSA algorithm, to achieve an optimal solution, this algorithm is executed 10 times, and the average of these runs is considered as the final optimal solution. The parameters used for solving the problem in CPLEX, as well as those for the proposed DSA algorithm, are detailed in Table 2.

Table 2. Problem parameters

The name of the parameter	Value
Electrical processing energy (E_{elect})	50 *nj/bit*
Energy of near-distance communication (ε_{fs})	10 *pj/bit/m²*
Energy of long-distance communication (ε_{mp})	0.001 *pj/bit/m⁴*
Energy consumed by the sink (e^s)	0.1 *j*
Number of sinks	Single & Multiple
Battery model	Constant
Simulation time	<3600 s
The number of sensors that can be connected to the sink (α^s)	20
The number of packets that can be processed by the sink (μ^s)	50000 *bits*
The number of sinks (φ^s)	10
Threshold (d_{co})	*87 m*
Amount of data sent *(t)*	2000 *bits*
Transmission type	Constant bit rate
Transmission range	5 m
Number of nodes (N)	100
Node distribution method in the area	Random
Simulation area	20 m × 20 m

The results obtained from these experiments are compiled in Table 3 based on the instances. In this table, for each problem instance, the following metrics are reported:

power: The optimal amount of energy consumed for these instances.
time: The duration taken for these instances.
% Err: The error percentage of the proposed DSA algorithm, in comparison to CPLEX regarding the optimal energy consumption for these instances, is calculated as follows.

$$\frac{power_{DSA}(\rho) - power_{CPLEX}(\rho)}{power_{CPLEX}(\rho)} * 100 \quad (14)$$

In the calculation of this quantity, each of the functions $power_{DSA}$ (.) and $power_{CPLEX}$ (.) respectively represent the optimal energy consumption obtained from the execution of the proposed DSA algorithm and the CPLEX solver.

In Table 3, the first column lists the names of the experimented instances. The second column shows the execution time of CPLEX on the instances. The third column presents the best energy consumption value achieved for each instance by CPLEX, under a maximum execution time of 3600 seconds. The fourth and fifth columns show the execution time and the optimal energy consumption value attained by the proposed DSA algorithm, respectively. Lastly, the sixth column indicates the error percentage of the proposed DSA algorithm relative to CPLEX in terms of identifying the best energy consumption value for each instance.

Table 3. The results of running the DSA and CPLEX algorithms on the instances.

	CPLEX		DSA		DSA vs CPLEX
Instance	**Time**	**Power**	**Time**	**Power**	**Err**
Sensors	**(sec.)**	**Joule**	**(sec.)**	**Joule**	**%**
5	0.31	5.15	0.53	5.15	0.03
10	0.30	10.22	0.53	10.24	0.14
15	3.33	15.25	0.53	15.25	0.00
20	0.39	20.40	0.56	20.40	0.00
25	41.78	26.44	0.54	26.45	0.03
30	20.05	31.40	0.67	31.42	0.07
35	365.42	36.52	0.60	36.53	0.04
40	3600	43.56	0.62	43.56	0.01
45	3600	48.55	0.60	48.56	0.01
50	3600	53.62	0.68	53.65	0.06
55	3600	61.68	0.71	61.73	0.07
60	3600	66.71	0.80	66.72	0.02
65	3600	71.73	0.76	71.76	0.05
70	3600	80.87	0.76	80.93	0.08
75	3600	85.89	0.81	85.98	0.10
80	3600	90.93	0.89	91.02	0.10
85	3600	101.16	0.84	101.12	0.00
90	3600	106.13	0.87	106.25	0.11
95	3600	117.32	0.90	117.47	0.12
100	3600	129.55	0.90	129.66	0.08

The results obtained from Table 3 demonstrate the superiority of the DSA algorithm in execution time for all instances. Additionally, the solution obtained by the DSA algorithm is very close to the solution derived from the exact computational method using the CPLEX solver. Notably, the error percentage of the solution derived from the proposed method remains below 0.15% for all instances experimented, indicating the optimal performance of the proposed method in finding a suitable solution for the problem of allocating sensors to sinks.

Figure 3 depicts the error percentage of the proposed DSA algorithm compared to CPLEX across various instances, with the horizontal axis representing instance sizes based on the number of sensors within the network, while the vertical axis shows the error percentage for each instance. According to this chart, the DSA algorithm performs well in finding solutions close to the exact answer, such that, based on the maximum value shown in the graph, the error percentage does not exceed 0.15. However, in most instances, this error percentage drops below 0.1%, and in some cases, it even reaches zero.

Figure 4 shows the execution time of the DSA algorithm compared to CPLEX. The horizontal axis in this figure represents the instance size based on the number of sensors, while the vertical axis indicates the execution time in seconds for the instances. The results presented in this figure underscore the time efficiency of the proposed method. In other words, the proposed algorithm can identify solutions with a minimal error percentage relative to the exact computational approach, all while optimizing the use of time resources. This notable benefit establishes the proposed method as an effective solution for addressing the sink location problem.

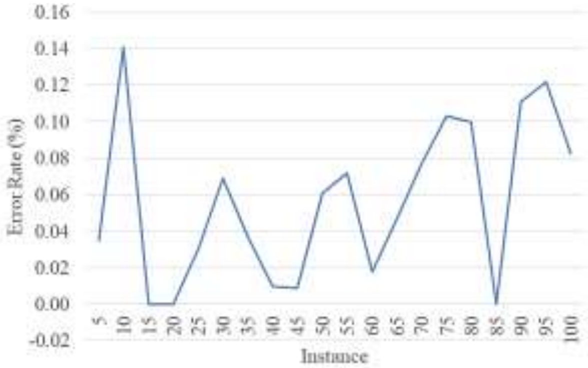

Figure 3. Error percentage of results obtained by DSA algorithm in comparison with CPLEX

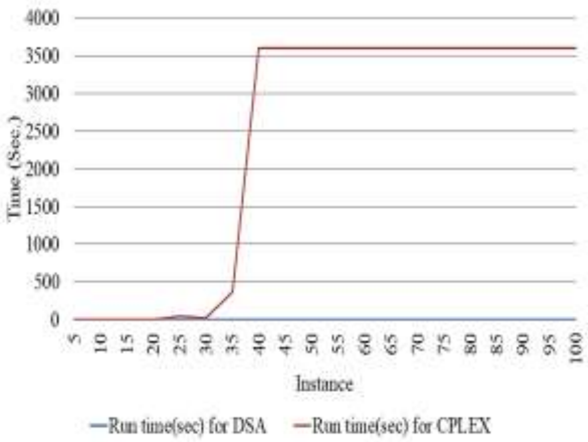

Figure 4. Execution time of the DSA algorithm on instances compared to CPLEX.

5.2. The Second Stage Experiment:

In this step, we use topology models obtained from Topology Zoo [18] to investigate the proposed solution in a real-world. Details of the topologies used in these experiments are given in Table 4.

Table 4. Topology information

Topology	|N|	|F|	|S|
iSTAR-A	18	3	15
iSTAR-B	18	4	14
iSTAR-C	18	5	13
NTELOS-A	48	5	43
NTELOS-B	48	10	38
NTELOS-C	48	15	33

The values of the problem-solving parameters and hardware equipment for performing the experiments are the same as in the first stage.

The results obtained from these experiments are compiled in Table 5.

% imp: The percentage improvement of the proposed formulation over the Joint Multiple Sink method is calculated as:

$$\frac{Power_{mc-jmsp} - Power_{pro}}{Power_{pro}} * 100 \quad (15)$$

In calculating this quantity, $Power_{mc\text{-}jmsp}$ and $Power_{pro}$ functions, give the optimal energy consumption found by the MC-JMSP method and the proposed method, respectively.

Table 5. Experimental results comparing the proposed, MC-JMSP method.

	DSA		MC-JMSP		DSA vs MC-JMSP
Topology	**Time (sec.)**	**Power Joule**	**Time (sec.)**	**Power Joule**	**Imp %**
iSTAR-A	0.50	17.65	0.92	17.65	0.00
iSTAR-B	0.49	16.56	2.96	16.56	0.00
iSTAR-C	0.53	15.34	4.13	15.61	1.76
NTELOS-A	0.80	49.01	53.23	49.02	0.02
NTELOS-B	0.77	41.01	103.50	41.02	0.02
NTELOS-C	0.73	36.01	102.57	36.02	0.03

In Table 5, the first column presents the names of the topologies. The second and third columns show the execution time and the best solution obtained by the proposed method for the tested examples, respectively. Similarly, the fourth and fifth columns display the corresponding values for the compared method.

As shown in Table 5, the proposed method generally performs better than MC-JMSP method, especially in topologies with more nodes. This is because the proposed method consumes less energy for connections compared to MC-JMSP method due to the proper assignment of sensors to sinks and the elimination of active sinks with too few sensors.

Figure 5 shows the average percentage improvement of the proposed method compared to MC-JMSP method for different topologies.

In Figure 5, the horizontal axis represents the names of the topologies and the vertical axis represents the percentage improvement. As shown in figure 5, from topology iSTAR-C onwards, the proposed method shows better performance. The fluctuation observed in the graph in Figure 5 can be attributed to the random placement of sinks and sensors in the experimented topologies. As a result, some sensors may be positioned closer to potential sink deployment locations, while others may be farther away. This variability directly impacts the overall energy consumption of the network.

Figure 6 shows the average runtime of the DSA algorithm compared to the MC-JMSP method. The horizontal axis represents the names of the topologies, and the vertical axis represents the average runtime. The results in Figure 6 show that the proposed method for larger topologies responds better than the compared method. As a result, it can be said that the proposed method is suitable for solving the problem of locating sinks in large-scale wireless sensor networks.

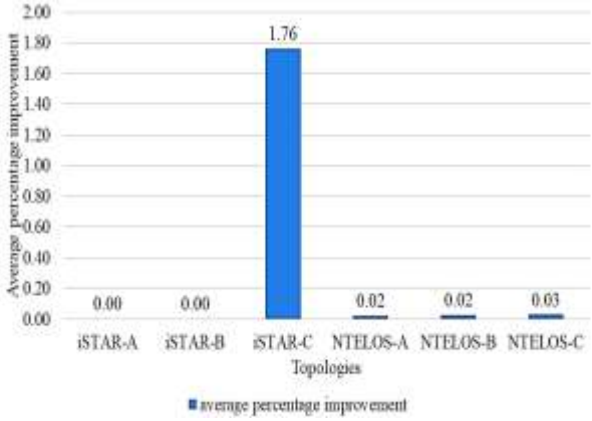

Figure 5. Average percentage improvement of the proposed method compared to MC-JMSP

Figure 6 shows the average runtime of the DSA algorithm compared to the MC-JMSP method. The horizontal axis represents the names of the topologies, and the vertical axis represents the average runtime. The results in Figure 6 show that the proposed method for larger topologies responds better than the compared method. As a result, it can be said that the proposed method is suitable for solving the problem of locating sinks in large-scale wireless sensor networks.

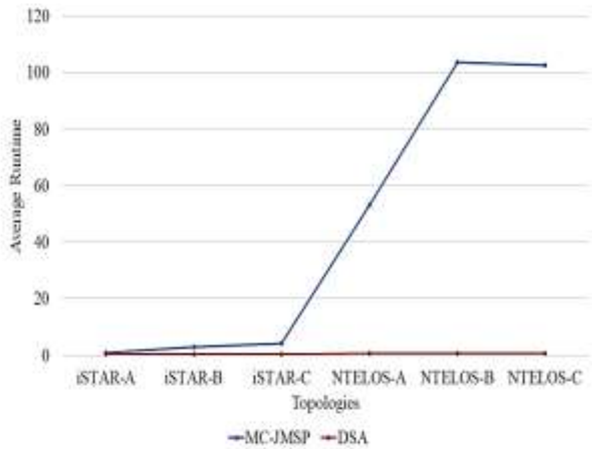

Figure 6. Average runtime of the proposed method compared to MC-JMSP

6. Conclusion

This paper examines the sink placement challenge in wireless sensor networks and introduces a dynamic sensor assignment-based algorithm to tackle it. The proposed algorithm utilizes the method of deactivating active sinks to achieve an efficient network topology within an acceptable timeframe. To evaluate the performance of the proposed algorithm, the experiments were conducted in two stages. In the first stage, various network instances were randomly generated. In the second stage, the topologies designed in Internet Topology Zoo were utilized. The experimental results of the proposed algorithm were then compared with those obtained from CPLEX and MC-JMSP method. The findings from the first stage revealed that the proposed method achieves optimal solutions in less time and with a lower error percentage compared to CPLEX. Additionally, the results from the second stage demonstrated the superiority of the proposed method over MC-JMSP method in terms of both execution time and minimizing energy consumption.

References

[1] K. Jaiswal and V. Anand, "EOMR: An energy-efficient optimal multi-path routing protocol to improve QoS in wireless sensor network for IoT applications." *Wireless Personal Communications*, Vol. 111, No. 4, pp. 2493-2515, 2020.

[2] L. Sitanayah, N. B. Kenneth, and J. S. Cormac, "Multiple sink and relay placement in wireless sensor networks." In *Proceedings of the 1st Workshop Artificial Intelligence for Telecommunications and Sensor Networks (WAITS'12), 20th European Conference on Artificial Intelligence (ECAI'12)*, 2012, pp. 18-23.

[3] S. T. Chandel, and S. Sharma, "Deployment of Multiple Sink in Different Monitoring Areas Based on the Optimization Technique." *International Journal of Sensors Wireless Communications and Control*, Vol. 10, No. 6, pp. 915-928, 2020.

[4] A. Jari and A. Avokh, "PSO-based sink placement and load-balanced anycast routing in multi-sink WSNs considering compressive sensing theory." *Engineering Applications of Artificial Intelligence*, Vol. 100, p. 104164, 2021.

[5] S. Bouarourou, A.Zannou,H. Nfaoui, and A. Boulaalam, "An efficient model-based clustering via joint multiple sink placement for WSNs." *Future Internet*, Vol. 15, No. 2, p 75, 2023.

[6] A. Singh and A. Nagaraju, "Low latency and energy efficient routing-aware network coding-based data transmission in multi-hop and multi-sink WSN." *Ad Hoc Networks*, Vol. 107, p 102182, 2020.

[7] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal Sink Node Placement in Large Scale Wireless Sensor Networks Based on Harris' Hawk Optimization Algorithm." *IEEE Access*, Vol. 8, No. 99, pp. 19381-19397, 2020.

[8] Q. Yu, C. Yang, G. Dai, L. Peng, and X. Chen. "Synchronous wireless sensor and sink placement method using dual-population co-evolutionary constrained multiobjective optimization algorithm." *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 6, pp. 7561-7571, 2022.

[9] N. T. Hanh, P. L. Nguyen, P. T. Tuyen, H. T. Thanh Binh, E. Kurniawan, and Y. Ji. "Node placement for target coverage and network connectivity in WSNs with multiple sinks." In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1-6. IEEE, 2018.

[10] F. Al-Salti, K. Day, N. Alzeidi, and A. Touzene. "Multiple sink placement strategy for underwater wireless sensor networks." In *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1-6. IEEE, 2018.

[11] Y. S. Chen, S. Y. Chang, T. W. Chang, and M. J. Tsai. "Multiple sink placement with latency and reliability guarantee in lossy wireless sensor networks." In *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-7. IEEE, 2018.

[12] E. Tuba, D. Simian, E. Dolicanin, R. Jovanovic, and M. Tuba. "Energy efficient sink placement in wireless sensor networks by brain storm optimization algorithm." In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 718-723. IEEE, 2018.

[13] M. S. Sarwar and P. Chatterjee. Optimal Sink Placement in Wireless Sensor Networks to Increase Network Performance. *Ind Interact Innov Sci Eng Technol [Internet]*. Vol. 11, pp. 423-434, 2018.

[14] P. Bose and M. Gurusamy. "Bacteria foraging algorithm based optimal multi sink placement in wireless sensor networks." *Journal of Intelligent Systems*, Vol. 27, No. 4, pp. 609-618., 2018.

[15] CPLEX: IBM's Linear Programming Solver.

[16] A. Moradi, A. Abdi Seyedkolaei, and Seyed A. Hosseini. "Controller placement in software defined network using iterated local search." *Journal of AI and Data Mining*, Vol 8, No. 1, pp. 55-65, 2020.

[17] B. Kabakulak, "Sensor and sink placement, scheduling and routing algorithms for connected coverage of wireless sensor networks." *Ad Hoc Networks*, Vol. 86, pp. 83-102, 2019.

[18] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and Matthew Roughan. "The internet topology zoo." *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 9, pp. 1765-1775, 2011.

تخصیص پویای حسگرها برای بهبود طول عمر شبکه‌های حسگر بی‌سیم

علی عبدی سیدکلایی

گروه مهندسی کامپیوتر، دانشگاه مازندران، بابلسر، ایران.

ارسال ۲۰۲۴/۱۲/۰۹؛ بازنگری ۲۰۲۵/۰۱/۱۷؛ پذیرش ۲۰۲۵/۰۲/۲۵

چکیده:

استفاده از چندین چاهک به جای یک چاهک، یکی از راه‌حل‌های ممکن برای بهبود طول عمر و دوام شبکه‌های حسگر بی‌سیم است. به‌کارگیری چندین چاهک منجر به تعریف مسئله‌ای به نام مسئله مکان‌یابی چاهک می‌شود. در این زمینه، هدف تعیین مکان‌های بهینه و تعداد چاهک‌ها در شبکه برای حداکثر کردن طول عمر شبکه است. در این مقاله، یک الگوریتم تخصیص پویای حسگر را برای حل مسئله مکان‌یابی چاهک پیشنهاد می‌کنیم و عملکرد آن را در مقایسه با روش‌های حل موجود بر روی مجموعه‌ای متنوع از نمونه‌ها ارزیابی می‌نماییم. آزمایش‌ها در دو مرحله انجام می‌شود: در مرحله اول، بر اساس نمونه‌های تصادفی و در مقایسه با روش محاسباتی دقیق با استفاده از حل‌کننده CPLEX، و در مرحله دوم، بر اساس نمونه‌های واقعی و در مقایسه با روش MC-JMSP. نتایج به‌دست‌آمده در مرحله اول آزمایش‌ها نشان‌دهنده برتری الگوریتم تخصیص پویای حسگر از نظر زمان اجرا برای تمامی نمونه‌ها است. علاوه بر این، جواب به‌دست‌آمده توسط این الگوریتم بسیار نزدیک به جواب حل‌کننده CPLEX است. به‌ویژه، درصد خطای جواب یافت‌شده توسط روش پیشنهادی در مقایسه با CPLEX در تمامی نمونه‌های آزمایش‌شده کمتر از ۰/۱۵ درصد است که نشان‌دهنده اثربخشی روش پیشنهادی در یافتن جواب مناسب برای تخصیص حسگرها به چاهک‌ها است. همچنین، نتایج مرحله دوم آزمایش‌ها برتری روش پیشنهادی را هم از نظر زمان اجرا و هم از نظر کارایی انرژی در مقایسه با روش MC-JMSP نشان می‌دهد.

کلمات کلیدی: شبکه حسگر بی‌سیم، مکان‌یابی چاهک، تخصیص پویا.