An Energy Efficient Routing Algorithm using Chaotic Grey Wolf with Mobile Sink-based Path Optimization for Wireless Sensor Networks

Latifah ALharthi[1], Alaa E. S. Ahmed[2], Mostafa E. A. Ibrahim[3]
College of Computer and Information Sciences, Imam Muhamad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia [1,2,3]
Shoubra Faculty of Engineering, Benha University, Cairo, Egypt[2]
Department of Electrical Engineering, Benha Faculty of Engineering, Benha University, Benha, Egypt[3]

***Abstract*—The task of deploying an energy-conscious wireless sensor networks (WSNs) is challenging. One of the most effective methods for conserving WSNs energy is clustering. The deployed sensors are divided into groups by the clustering algorithm, and each group's cluster head (CH) is chosen to gather and combine data from other sensors in the group. Mobile Wireless Sensor Networks, which enable moving the sink node, aid in reducing energy consumption. Thus, this paper introduces an energy efficient clustering algorithm and optimized path for a mobile sink using a swarm intelligence algorithms. The Chaotic Grey Wolf Optimization (CGWO) approach is used to form clusters and identify CHs. While utilizing the Slime Mould Algorithm (SMA) for determining the shortest path between a mobile sink and CHs. The effectiveness of the suggested routing strategy is evaluated against that of other current, cutting-edge protocols. The findings demonstrate that in terms of overall energy consumption and network lifetime, the suggested algorithm performs better than others. While for stability period the proposed algorithm outperforms three of compared algorithms and was close to the fourth.**

Keywords—Wireless sensor network; clustering algorithm; grey wolf optimizer; slime mould algorithm; mobile sink

I. Introduction

Mobile Wireless Sensor Networks (MWSNs) enable the movement of entities within a network, functioning as sensor nodes or sinks through mechanisms like wheels, humans, animals, or robots [1, 2]. MWSNs offer a solution to the hotspot problem often encountered in traditional Wireless Sensor Networks (WSNs). In a hotspot scenario, sensor nodes situated near a sink used as a relay tend to deplete their energy rapidly, as the sink increases the communication load on these nearby sensors [1]. MWSNs find applications in various domains, including but not limited to the military, industrial monitoring, habitat observation, healthcare, home networks, disaster management, and security [3]. These applications encompass fire detection systems in forests, battlefield surveillance, traffic monitoring, smart homes and hospitals, pollution control, rescue missions [4] and oil well monitoring.

Clustering algorithms play a crucial role in reducing energy consumption within WSNs. These algorithms partition the sensor nodes into distinct groups or clusters, with each group having a designated cluster head (CH) responsible for coordinating communications between its members and the sink. Clustering can be implemented through various approaches, such as distributed, centralized, or hybrid methods [5]. Sensors consume a significant amount of energy due to their tasks, which include environmental sensing, data transmission, mobility, cluster head (CH) selection, and frequent cluster formation [6]. Additionally, energy demands increase with larger data sizes and greater distances between sensors and the sink.

Numerous algorithms have been proposed to mitigate energy consumption, with clustering being a widely adopted approach. Clustering involves selecting CHs and forming clusters to reduce the number of sensors communicating directly with the sink, thus optimizing communication. Therefore, the process of CH selection is pivotal in clustering. Recent research [6, 7] has explored the use of intelligent swarm algorithms to aid in CH selection, such as ant, firefly, and Grey Wolf Optimization (GWO) algorithms.

This paper investigates the reduction of energy consumption in MWSNs by introducing an enhanced clustering algorithm and optimizing the path for a mobile sink using a swarm intelligence algorithm. Specifically, it employs the Chaotic Grey Wolf Optimization (CGWO) algorithm [8] for CH selection and the Slime Mould Algorithm (SMA) [9] to determine the shortest path between a mobile sink and CHs to reduce energy dissipation, and hence extends the WSN's life cycle.

The proposed algorithm has the following contributions:

- Employing the CGWO algorithm as a clustering mechanism in MWSNs which to the best of our knowledge has not been investigated up to now in this field.
- Utilizing the SMA algorithm for sink node route determination in MWSNs has not been well studied up to now, and this study aims to fill this gap.
- The results of the proposed algorithm are compared to those of four other state-of-the-art algorithms GWO [10], ACO [11], FA [12], and PSO [13]. in terms of several performance metrics such as network lifetime, stability period, and total consumed energy.

The rest of this paper is structured in five sections. Section II presents the related work. Section III provides the mathematical models for GWO, the enhanced CGWO and Slim Mould Optimization algorithms. Section IV describes the network model and the methodology followed to develop the proposed algorithm. Section V illustrates the simulation results indicating the performance evaluation of the proposed protocol. Finally, Section VI sums up the paper and figure out future directions.

II. Related Work

A. Clustering Algorithms

This section reviews the state-of-the-art clustering algorithms that were recently used in MWSN. The earliest clustering algorithms for WSNs fell into the category of traditional clustering algorithms. These methods employed straightforward techniques for constructing clusters and selecting CHs. In essence, traditional methods designated CHs without the use of sophisticated, intelligent approaches [7]. An example of this approach is the Low Energy Adaptive Clustering Hierarchy (LEACH) [14] protocol.

Authors of [15] presented a heterogeneous clustering algorithm with multiple mobile data collectors (MDCs) to extend the network's lifespan. This algorithm selected CHs using a probability equation based on factors such as energy levels. The MDCs employed the Expectation-Maximization (EM) method to determine optimal paths for CHs based on their positions and energy levels. It demonstrated superior performance, particularly in small areas.

Authors of [11] introduced an enhanced clustering algorithm that employed multiple mobile sinks to improve energy efficiency. This enhanced clustering method incorporated the highest residual energy of sensors as a metric for CH selection, thereby enhancing the traditional LEACH [14] protocol. Mobile sinks utilized the Ant Colony Optimization (ACO) algorithm to identify optimal paths to CHs. Their algorithm defeated the LEACH, Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) in terms of network lifetime and energy consumption.

The authors of [16] proposed inter- and intra-clustering methods to optimize the movements of mobile sinks and conserve energy. The inter-clustering method involved calculating the sojourn time of mobile sinks in clusters, while the intra-clustering method determined the sojourn locations of mobile sinks within clusters. CHs were selected based on proximity to the cluster centre, the highest residual energy was also considered in subsequent CH selections. Their proposed algorithm outperformed the Energy-Efficient PSO-Based Routing algorithm with Mobile Sink (EPMS) and Two-Tier Data Dissemination (TTDD) algorithms in terms of inter-cluster movement.

Recent research papers [7, 17, and 18] have employed optimized clustering algorithms to enhance the network's lifetime. These optimized clustering algorithms leverage Computational Intelligence (CI) methodologies, encompassing fuzzy logic, swarm intelligence, Genetic Algorithms (GA), and petri nets.

In study [19] researchers introduced an algorithm for extending network lifetime and reducing transmission delays. A distributed fuzzy clustering algorithm was used to select CHs. The distributed fuzzy clustering algorithm integrated seven regular components of a fuzzy system into two elements. The first element characterized sensors based on their remaining energy, the number of neighbours, and distances from neighbours. The second element determined sensor and mobile gateway positions, considering factors such as the number of mobile gateways and distances to nearby and distant mobile gateways.

In research [20], authors proposed a fuzzy logic-based algorithm for the clustering process and employed multiple mobile sinks to reduce energy consumption in MWSNs. The fuzzy logic algorithm initially selected temporary CHs and then chose the final CHs from this group based on criteria including distances to the nearest Rendezvous Node (RN), remaining energy, and calculated cluster densities. The sensor areas were divided into regions, each served by a mobile sink. These mobile sinks collected data from the RNs and final CHs using a smart trajectory. Their results showed significant improvement in terms of first node dead, the time at which half of the nodes were still operational (HNA), and the total remaining energy (TRE).

The Particle Swarm Method [21, 22] is a type of swarm intelligence algorithm inspired by the food-searching strategies employed by animal flocks. It divides the swarm into groups, each following a distinct path [21]. Within each group, particles iteratively explore and update information to find the best positions while communicating with others.

In study [23] researchers proposed an enhanced fitness function for the Unequal Clustering PSO (UC-PSO) and Hybrid K-Means Clustering PSO (KC-PSO) algorithms. These improvements aimed to enhance energy efficiency and determine the optimal number of clusters and CHs. The new fitness function selected CHs based on factors such as mobility, residual energy, neighbour connectivity, and distance to the BS. While the KC-PSO algorithm employed UC-PSO in the CH selection process, it utilized a different cluster formation approach. Their results demonstrated that the KC-PSO algorithm outperformed the UC-PSO and LEACH algorithms.

Another example of swarm intelligence algorithms is the Firefly Method [12]. It is used for selecting CHs and introduced a mobile sink to enhance the network's lifetime. CH selection parameters included residual energy, node-to-node distances, and distances from nodes to the sink. The proposed method demonstrated superior performance compared to LEACH, Amend LEACH (A-LEACH), and GA-Based LEACH (LEACH-GA) methods. Additionally, it outperformed the Mobile Sink Improved Energy-Efficient PEGASIS-Based Routing Protocol and Mobile Sink-Based Adaptive Immune Energy-Efficient Clustering Protocol (MSIEEP) algorithms in terms of network lifespan, node residual energy, packet drop ratio (PDR), and packet delays.

The GWO method emulates the social hierarchy observed in grey wolf packs, consisting of alpha, beta, delta, and omega wolves [24]. Researcher in [25] introduced a layered and clustered structure based on the GWO method aimed at optimizing energy consumption.

TABLE I. COMPARISON OF STATE-OF-THE-ART CLUSTERING ALGORITHMS

Ref.	Mobility	Clustering Method/Size [1]	Cluster Head Selection Alg.	Cluster Head Selection Parameters [2]	Simulation Platform	Performance Metrics
[11]	Sink	Dist./dynamic	Traditional	- Residual energy	MATLAB	- Network lifetime - Energy consumption - Average packet loss ratio
[12]	Sink	Cent./dynamic	Firefly	- The residual energy - Distance concerning a node and other nodes - Distance from a node to the sink	MATLAB	- Network lifespan - Residual energy of nodes - Packet drop ratio - Packet delay
[13]	Sink	Cent./dynamic	Traditional	Based on the non-probability Method: - The number of neighbours - The rate at which packets are received	MATLAB	- Number of rendezvous points - Average memory utilization - Number of hops - Packet loss rate - Standard deviation - Throughput - Energy consumption
[15]	Sink	Dist./dynamic	Traditional	- Highest energy	MATLAB	- Number of cluster heads - Network lifetime - Stability period - Throughput
[16]	Sink	Dist./dynamic	Traditional	- The centre of a cluster - Residual energy	OMNet ++ Simulator	- Network lifetime - Residual energy
[19]	Sink	Dist./dynamic	Fuzzy Logic	- The remaining energy - The number of neighbours - The distance to neighbours - The position of a sensor and mobile gateways - The number of mobile gateways - The distance between a sensor and near and distant mobile gateways	OMNet ++ Simulator	- Number of dead sensor nodes - Average remaining energy - Delay in sending packets from the sensor to the base station
[20]	Sink	Dist./dynamic	Fuzzy Logic	- The distance to the closest rendezvous node - The remaining energy - The density	MATLAB	- First node failed - Nodes still operational - Total remaining energy
[22]	Sink	Cent./dynamic	Particle Swarm	- The remaining energy - Centre of the cluster	NA	- Network lifetime - Amount of packet delivery - Energy consumption - Average delivery delay
[23]	All Nodes	Cent./dynamic	Particle Swarm	- Mobility - Residual energy - Neighbours - Distance from the cluster head to the base station	NS2 Simulator	- Number of clusters formed - Network lifetime - Total energy consumption - Packet delivery ratio
[25]	All Nodes	Dist./dynamic	Grey Wolf Optimization	- Residual energy - RSSI - PRR	MATLAB	-Network lifetime - Energy consumption - Throughput
[26]	Sink	Dist./dynamic	Traditional	- Remaining energy - Distance - Data rate	MATLAB	- Network lifetime - Energy consumption - Throughput
[27]	Sink	Dist./dynamic	Traditional	- Residual energy - The ID of a sensor	NS2 Simulator	-Number of active nodes -Average residual energy -Total energy consumption
[28]	Sink	Dist./dynamic	Traditional	- Residual energy - Distance between cluster head and mobile sink	MATLAB	- Network lifetime - Energy consumption
[29]	Sink	Dist./dynamic	Traditional	- The centre of a cluster - Residual energy	Not Mentioned	- Network lifetime - Energy consumption - Packet delivery
[30]	Sink	Cent./dynamic	Traditional	-The centre of a cluster - Residual energy	NS2 Simulator	- Number of alive nodes - Number of delivered packets
[31]	Sink	Dist./dynamic	Traditional	- Residual energy - Distance	MATLAB	- Network lifetime

[1] Cent: Centralized, and Dist: Distributed

[2] Based on weighted probability:

This layered structure consisted of four tiers: alpha, beta, delta, and omega, with the alpha tier being the closest to the static BS. In this context, mobile sensors were analogous to grey wolves, and alpha wolves assumed the role of CHs. CHs were chosen using game theory principles, considering factors such as residual energy, Received Signal Strength Indexes (RSSIs), and Packet Reception Ratios (PRRs). Performance metrics encompassed network lifetime, energy consumption, and throughput. The presented method had better throughput. The results clearly demonstrated that the proposed method outperformed the LEACH protocol. Finally, Table I offers a comparative review of surveyed clustering algorithms indicating the platforms used and other parameters.

B. *Wireless Sensor Nodes Mobility*

Sink mobility contributes to energy conservation by allowing the sink to move. Path determination is a significant contributor to the energy consumption of WSN. Hence many researchers presented different methods for establishing energy efficient path between nodes [32]. Sink mobility allows for the movement of the sink, and there are three methods for selecting a path between the sink and nodes [2]: random, controlled, and predictable.

First, the easiest way is the random path technique. In the random path method, the mobile sink moves randomly to gather data from sensors [2]. This technique is used in [19, 26]. Second way of moving the sink node is the controlled path method. In the controlled path method, the mobile sink moves strategically within areas that meet certain constraints, such as high residual energy, the number of neighbors, and the number of hops.

Authors of [15] introduced a controlled adaptive mobility model using an EM algorithm. This algorithm assists the sink in collecting data from CHs with the lowest residual energy first. While in study [11] they implemented a controlled path based on the ACO algorithm. Authors of [29] also introduced a controlled route based on improved ACO, considering a distance heuristic factor to enhance its effect on the next node and improve global search ability.

Authors of [16] utilized a controlled trajectory determined by the GA. In [20] they presented a mobile sink that calculates the optimal trajectory by dividing the area of interest into 16 equal parts and considering the average remaining energy of each part. The mobile sink then follows a smart path based on RPs. Researchers of [13] employed the PSO algorithm to determine the path of the mobile sink to CHs. In [33] they presented another technique that integrated ACO and A* algorithms for finding the best energy efficient route between CHs and a base station.

Third way of sink movement is the predictable path. In the predictable path method, the mobile sink follows a predefined route to specific relay or data collector nodes responsible for gathering data from sensor nodes and transmitting it to the mobile sink [2]. Authors of [12] introduced a mobile sink that selects its path by dividing the network into four or eight areas and moves to each part using the centroid of the CHs. While [27] used a predictable trajectory. Authors of [28] used a predetermined route depending on the angular velocity.

III. TECHNICAL BACKGROUND

This section provides the mathematical models for Grey Wolf Optimization (GWO), the enhanced Chaotic GWO and Slim Mould Optimization algorithms.

A. *Original GWO*

The original GWO algorithm, introduced by [34], draws inspiration from the cooperative hunting and social hierarchy of grey wolves to tackle optimization problems. The core concept of the GWO algorithm involves locating a target, or "prey", by mimicking the leadership hierarchy of grey wolves. The inspiration is in the cooperative hunting and social hierarchy of grey wolves.

Grey wolves organize themselves into a dominant social hierarchy, featuring alpha, beta, delta, and omega wolves. The alpha wolves occupy the top tier of this hierarchy, where they make decisions regarding hunting and habitat selection. The beta wolves comprise the second tier and have the authority to issue commands to the delta and omega wolves. Delta wolves, in turn, follow the directives of the alpha and beta wolves and oversee the omega wolves. Ultimately, the omega wolves obediently follow the commands of all other members of the pack.

The population-based meta-heuristic method known as "grey wolf optimization" (GWO) mimics the natural hunting strategy and leadership structure of grey wolves. The GWO hunting process consists of several key phases:

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attacking the prey.

1) GWO mathematical model: The GWO algorithm [24, 34] considers the fittest solution as the alpha (α). As a result, the second and third-best solutions are designated as beta (β) and delta (δ), respectively. The remaining candidate solutions are assumed to be omega (ω). α, β, and δ guide the hunting process, with the ω wolves following these three leaders.

a) Encircling prey: Mathematically, grey wolves enclose and surround their prey as follows [34]:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{2}$$

where, $\vec{D}$ indicates the distance between prey and wolf, t corresponds to the present iteration, $\vec{X}(t)$ is the current position of the wolf, and $\vec{X}_p(t)$ is the position of the prey. $\vec{A}$ and $\vec{C}$ are coefficient-vectors, $\vec{X}p$ is the vector's location of the prey, and X is the vector's location of a grey wolf. vectors $\vec{A}$ and $\vec{C}$ can be computed as follows:

$$\vec{A} = 2\,\vec{a} \cdot \vec{r}_1 - \vec{a}, \qquad \vec{C} = 2 \cdot \vec{r}_2 \tag{3}$$

where, $\vec{r}_1$ and $\vec{r}_2$ are random vectors in range of [0,1], and the components of $\vec{a}$ are reduced linearly from 2 down to 0 throughout repeated iterations.

b) Hunting: The algorithm keeps the first three best solutions alpha α, beta β, and delta δ and obliges the omega ω wolves to adjust their positions based on positions of wolves α, β, and δ. Eq. (4) to Eq. (6) indicates how the distances from α, β, and δ wolves ($\vec{D}_\alpha, \vec{D}_\beta$ and $\vec{D}_\delta$) to each of the lasting wolfs, using positions α, β, and δ wolves ($\vec{X}_\alpha, \vec{X}_\beta$ and $\vec{X}_\delta$) and the position of lasting wolves($\vec{X}$):

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (4)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \quad (5)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (6)$$

c) Attacking prey: In study [34] Grey wolves conclude the hunt by attacking the prey when it stops moving. The GWO algorithm models approaching the prey mathematically by decreasing the value of $\vec{a}$ and narrowing the fluctuation range of $\vec{A}$ where |A|<1 force the wolves to attack the prey (exploitation).

d) Deletion: The search is based on the positions of the alpha, beta, and delta grey wolves. Grey wolves diverge from each other to search for prey and converge to attack prey. In the mathematical model of divergence, the GWO algorithm employs $\vec{A}$ where |A| > 1 to compel the search agents to diverge from the prey. The $\vec{C}$ vector contains random values in [0, 2], providing random weights for prey. This stochastic weighting either emphasizes (when C > 1) or deemphasizes (when C < 1) the attack. It reflects the effect of obstacles to approaching prey in nature. Depending on the position of a wolf, it can randomly assign weight to the prey, making it harder or easier for wolves to reach it.

B. Chaotic GWO

CGWO algorithm utilizes a varying number of wolves, which are regarded as leaders during each iteration. Rodrigues (2021) [8] proposed the use of a chaotic variable to determine the number of leaders in the pack during each iteration. To calculate the number of leader wolves for each iteration, the following formula is employed:

$$n(t) = \left\lceil \frac{M.z(t)}{2} \right\rceil \quad (7)$$

The number of leader wolves n(t) in each iteration ranges between 1 and half of the population size. The author used the Celling function for n(t) to round the result to the next integer. Here, M represents the number of wolves in the pack, z(t) is a chaotic variable within the interval [0,1] [8, 35]. In contrast to the way of updating the position of individual wolves in GWO described by Eq. (6), in CGWO the author utilized Eq. (8).

$$X(t+1) = \frac{\sum_{v=1}^{n(t)} X_v}{n(t)}, X_v = X_j(t) - A_j . D_j \quad (8)$$

where, Xj(t) represents the position of the wolf with the j-th best fitness value in iteration t, Aj is a random vector computed according to Eq. (4), (3) and D_j is calculated using the following equation:

$$D_j = |C_j . X_j(t) - X| \quad (9)$$

where, X is the current position of the wolf, and C_j is a random vector calculated according to Eq. (3).

Chaotic maps [8] represent the chaotic function, often referred to as an orbit. An orbit is an iterative function that generates a sequence of values in each iteration. The characteristics of an orbit are its aperiodic nature, boundedness (chaotic variables have upper and lower limits), and sensitivity to initial conditions.

The CGWO algorithm, in contrast to GWO, incorporates a chaotic sequence to determine a varying number of leaders in each iteration. This approach enhances the CGWO algorithm’s diversification capability and strikes a balance between diversification and intensification capabilities, which is crucial for the optimization algorithm’s overall performance. By involving a larger number of wolves as leaders, candidate solutions that are not near the optimal solution contribute to guiding the search process. In [8] nine chaotic maps are investigated.

C. Slim Mould Optimization Algorithm

The slime mould organism relies on creating an interconnected venous network to seek out food sources, allowing it to generate optimal paths for reaching food [9]. The SMA algorithm has two phases:

1) Approach food: The organic matter in slime mould seeks food, surrounds it, and secretes enzymes to digest it. The slime mould navigates towards a food source using a mathematical expression designed to mimic its contraction behavior as follows:

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(\vec{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \quad (10)$$

where, $\overrightarrow{vb}$ is a parameter with a range of $[-a, a]$, $\overrightarrow{vc}$ decreases linearly from 1 to 0. t indicates the current iteration, $\overrightarrow{X_b}$ denotes the individual location with the highest odour concentration currently found, $\vec{X}$ indicates the location of the slime mould, $\overrightarrow{X_A}$ and $\overrightarrow{X_B}$ indicate two individuals randomly selected from the swarm, and $\vec{W}$ indicates the weight of the slime mould. P is computed as follow:

$$p = \tanh|S(i) - DF| \quad (11)$$

where, $i \in 1,2,\dots,n$, S(i) denotes the fitness of $\vec{X}$ and DF indicates the best fitness obtained in all iterations. $\overrightarrow{vb}$ is computed as:

$$\overrightarrow{vb} = [-a, a], a = \operatorname{arctanh}\left(-\left(\frac{t}{max\_t}\right) + 1\right) \quad (12)$$

where, max_t indicates the maximum number of iterations. $\vec{W}$ is calculated as:

$$\overrightarrow{W(SI(i))} = \begin{cases} 1 + r \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), \text{condition} \\ 1 - r \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), \text{ others,} \end{cases} , \quad (13)$$

$$SI = sort(S)$$

Where condition denotes that S(i) ranks in the first half of the population, r represents a random value within the interval [0,1], bF stands for the optimal fitness obtained in the current iterative process, wF indicates the worst fitness value obtained in the current iterative process, and SI represents the sequence of fitness values sorted in ascending order.

2) Wrap food: It mimics the contraction mode of the venous tissue structure of slime mould during its search for food. As the concentration of food encountered by the vein increases, the bio-oscillator generates stronger waves, resulting in faster cytoplasmic flow and thickening of the vein [9]. The slime mould updates its location using the following mathematical formula:

$$\overrightarrow{X^*} = \begin{cases} \text{rand} \cdot (\text{UB} - \text{LB}) + \text{LB}, \text{rand} < z \\ \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(W \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \quad (14)$$

where, LB and UB represent the lower and upper boundaries of the search range, while rand and r denote random values in the range [0,1]. The constant z is set to 0.03.

3) Osillation: Slime mould relies on a biological oscillator to generate propagating waves that alter the flow of cytoplasm within its veins, allowing it to position itself more effectively in areas of higher food concentration. The vector value $\overrightarrow{vb}$ randomly oscillates within the range of $[-a, a]$ and gradually approaches 0 with increasing iterations. Similarly, the vector value $\overrightarrow{vc}$ oscillates within the range of [–1,1] and tends to 0 eventually [9]. Using the SMA algorithm, the mobile sink chooses the shortest path between itself and the CHs to collect sensed data.

IV. Methodology

This section dives into the design and implementation of the proposed algorithm. The proposed algorithm consists of two phases: cluster construction and path formation for a mobile sink. It combines the CGWO algorithm for the selection of CHs and the SMA for determining the shortest route between a mobile sink and the CHs.

A. WSN Model

The wireless sensor network model considered in this paper has the following assumptions:

- The network model is synchronous.
- All nodes are stationary but the mobile sink station.
- All nodes have the same initial battery capacity and can perform the same functions.
- Each sensor node is identified with a unique identifier.
- The links are symmetric.
- The distance between sensor nodes can be calculated based on received signal strength indicator (RSSI).

B. Cluster Structure Phase

The main functions of this phase are electing CHs using the CGWO algorithm and associating sensor nodes with respective CHs to form clusters.

The sensor nodes are randomly deployed in a region, and a mobile sink is located in the center of the network. Initially, the sensor nodes send their locations and remaining energy to a mobile sink. Then, the mobile sink selects the CHs based on the fitness function of the CGWO algorithm. The clusters have the following properties:

- Centralized clustering formation method: The mobile sink utilizes the CGWO algorithm to select CHs.
- Fixed cluster count: It specifies a fixed number of clusters, which is ten clusters in each round.
- Variable cluster size: The number of cluster members is not fixed in each round.
- Intra-cluster topology: The proposed algorithm relies on single hops to connect cluster members to their respective CHs.
- Inter-CH connectivity: The proposed algorithm establishes direct connections from CHs to the mobile sink.

This article introduced a new fitness function that relies on the following parameters for selecting CHs:

- Remaining energy of the CH.
- The CH's membership count.
- Euclidean distance from a mobile sink to a CH.
- CH centrality.

The fitness function for sensor node i is calculated as follows:

$$F(i) = aF_f(i) + (1 - a)F_u(i) \quad (15)$$

$$F_f(i) = R_e(i) + S_{m(i)} \quad (16)$$

$$F_u(i) = [E_u(i, M_s)]^{-1} + C_s(i) \quad (17)$$

where, a is a scaling factor with a value from 0.1 to 0.9, F_f is the fundamental fitness function, and F_u represents the non-fundamental fitness function.

The fundamental fitness function (F_f) calculates the sensor node's members (S_m) and its remaining energy (R_e). Sm signifies the number of connecting nodes to a particular node within its transmission range, while R_e is the ratio of the remaining energy to the initial energy of the node.

The non-fundamental fitness function (F_u) computes the Euclidean distance (E_u) and sensor centrality (C_s). E_u calculates the Euclidean distance from sensor i to the mobile sink M_s, while C_s determines the sensor node's centrality among its neighbors.

The sensor's members (Sm) is computed as follows:

$$N(i) = \sum_{j \in W} b_{ij} \quad (18)$$

where, W is the wireless sensor network, $b_{ij} = 1$ means that distributed sensor i is connected to the distributed sensor j; otherwise, $b_{ij} = 0$.

Algorithm 1: Cluster Formation Algorithm

A. Initialize WSN parameters: Area, number of nodes, initial mobile sink position (50,50), … etc.

Input:
Number of alive nodes
Number of packs P_i
Number of cluster heads in a pack: 10% of the total number of alive sensor nodes randomly.

Output:
Cluster heads

B. Cluster head selection using CGWO algorithm.

1. Initialize the population X_i, choose 10% of sensor nodes as cluster heads in a pack from alive nodes randomly.
2. Initialize a = 2.
3. Initialize vectors A and C.
4. Initialize the Tent chaotic map.
5. Compute the fitness value of each wolf according to Eq. (15).
6. Define the number of leaders n(t) according to Eq. (7).
7. **For** j = 1 : n(t) **do**
8. Compute D_j according to Eq. (9).
9. Compute X_v according to Eq. (8).
10. **End for**.
11. **While** t < maximum number of iterations **do**
12. **For** each wolf i **do**
13. Update its position X(t+1) according to Eq. (8).
14. **End for**.
15. Update a, A, C, and n(t).
16. Compute the new fitness value of each wolf.
17. **For** j = 1 : n(t) **do**
18. Compute D_j according to Eq. (9).
19. Compute X_v according to Eq. (8).
20. **End for**
21. Increment the iteration number (t = t + 1).
22. **End while**
23. **Return** the best solution (i.e. cluster head)
24. CH = Set of sensor nodes in P_i.

End Algorithm.

The sensor's members (C_s) is computed as follows:

$$C_s(i) = \frac{(N-1)}{\sum_{j=1}^{N} d_{ij}} \tag{19}$$

where, N is the number of sensors and d represents the shortest distance from sensor i to sensor j.

The remaining energy of sensor node (R_e) is computed as follows:

$$\mathbf{R_e} = \frac{\mathbf{1}}{\sum_{\mathbf{j=1}}^{\mathbf{m}}\left(\mathbf{E_{CH_j}}\right)} \tag{20}$$

where, m is the total number of CHs, l_j is the number of sensor nodes in cluster j, and ECH_j is the current energy of CH_j, $1 \leq j \leq m$.

The Euclidean distance (E_u) is computed as follows is computed as follows:

$$E_u = \sum_{j=1}^{m}\left(\frac{1}{l_j}\operatorname{dis}\left(CH_j, M_s\right)\right) \tag{21}$$

where, dis(CH_j, M_s) signifies the distance between CH_j and M_s.

Based on the experimental investigations, this paper employs the tent function Eq. (22) for the chaotic map function and set 'a' to 0.2 in the fitness function to achieve superior outcomes.

$$z(t+1) = \begin{cases} z(t)/0.4, & 0 < z(t) \leq 0.4 \\ \left(1 - z(t)\right)/0.6, & 0.4 < z(t) \leq 1 \end{cases} \tag{22}$$

In this study, the sensor nodes will transmit their locations and remaining energy to the mobile sink. Subsequently, the mobile sink will select CHs using the CGWO algorithm as described in Algorithm 1. Following this selection, the sensor nodes will align themselves with their respective CHs and begin transmitting the sensed data.

C. Path Formation Phase

Following the election of CHs, the mobile sink employs the SMA algorithm to move towards the nearest CH, followed by the second closest CH, and so forth, for data collection. This algorithm exerts control over the movement of the mobile sink.

Algorithm 2: Mobile Sink Path Formation Algorithm

1. **Input:** Number of cluster heads which is determined by the **Algorithm 1**

2. **Output:** The slime mould position = the mobile sink position

3. Initialize the population size = number of cluster heads.
4. Initialize the positions of the slime mould positions.
5. **While** t ≤ Max_iteration **do**
6. Calculate the fitness of all the slime moulds by Eq. (23).
7. Update bestFitness, x_b.
8. Calculate the W by Eq. (13).
9. **For each** search portion **do**

Update p, vp, vc.
Update positions by Eq. (14).

10. **End For**
11. t = t + 1.
12. **End while**
13. Return best Fitness, x_b = the mobile sink position.

A new fitness function is proposed for determining the path between a mobile sink and CHs. The following equation calculates the path distance (D) travelled by the mobile sink (M_S):

$$D(T_i) = \operatorname{dis}(M_S, ch_1) + \sum_{i=1}^{e-1}\operatorname{dis}(ch_i, ch_{i+1}) + \operatorname{dis}(ch_e, M_S) \tag{23}$$

where, ch is the number of CHs, ch1 is the nearest CH to the mobile sink MS, ch_e is the farthest CH, and $\operatorname{dis}(M_S, ch_1)$ indicates the distance between CHs or between a CH and an M_S. The M_S begins its journey from an initial position, visits all CHs, and returns to the starting point.

Algorithm 2 outlines the steps involved in constructing the mobile sink's route using the SMA algorithm.

D. Energy Consumption Model

This paper utilizes the first-order radio model as the energy consumption model for both sending and receiving data, as proposed by [36]. This model incorporates both the free space (f_s) and multi-path fading (m_p) models and is contingent on the distance between the sender and receiver. When the distance is less than the threshold value d_0, the authors employ the free space (f_s) model. Otherwise, the authors switch to the multi-path (m_p) model. The energy consumption for transmitting an l-bit message over a distance d is calculated as follows:

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + lE_{fs}\, d^2, & d < d_0 \\ lE_{elec} + lE_{mp}\, d^4, & d \geq d_0 \end{cases} \quad (24)$$

where, E_{Tx} represents the total energy required for transmission, E_{elec} denotes the energy dissipation per bit for circuit operation, including the transmitter or receiver, E_{fs} is the energy used for amplification in the free space model, and Emp pertains to the multi-path model and is significantly influenced by the transmitter amplifier model. The energy required to receive an l-bit message is calculated as:

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \quad (25)$$

where, E_{Rx} represents the energy consumption for data reception. The threshold distance d_0 is set as follows:

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (26)$$

V. Simulation Experments and Results

A. Experimintal Results

All the simulation experiments were conducted on a laptop with a processor speed of 1.70 GHz Intel Core i7 and memory of 16 GB. The operating system used was Windows 11, version 22H2. The simulation platform was the MATLAB 2023a. A WSN of area 100× 100 m^2 is considered. The nodes are uniformly deployed with the BS initially in the center of the network area. Table II presents the network model and simulation parameters used in the experiments.

TABLE II. Network Setup and Simulation Parameters

Parameter	Value
Simulation area	100 × 100 m^2
Total number of nodes	100
Initial energy	0.5 Joules
Number of rounds	2500
Packet size	2000 bytes
Initial position of the mobile sink	(50,50)
E_{elec}	$5.0e^{-8}$
E_{fs}	$1.0e^{-11}$
E_{mp}	$1.3e^{-15}$

B. Simulation Results

This section describes and compares the developed algorithm's performance with that of the following: FA [12], GWO [10], ACO [11], and PSO [13]. Several measures, including total residual energy, total energy consumption, network lifetime, and stability period are used for analysing and assessing the proposed algorithm.

The total residual energy versus time (in terms of rounds) is presented in Fig. 1. The suggested protocol exhibits larger residual energy than the other three algorithms, as the simulation results obviously reveal. Notably, the proposed algorithm outperforms compared ones, with the residual energy reaching 0 at rounds 1142, 2318, 2381, and 2381 for ACO [11], FA [12], PSO [13], and GWO [10], respectively. On the other hand, the proposed algorithm reaches 0 at round 2490.

The developed algorithm maintains the highest energy levels until 2489 rounds due to its use of the optimized clustering chaotic variable in conjunction with the GWO algorithm for CH selection. Additionally, the SMA algorithm aids the CGWO algorithm by facilitating the mobile sink's path determination to CHs.

Fig. 2 represents the total energy consumption versus time (in terms of rounds) of the proposed algorithm compared to the three other algorithms. The total energy consumption is measured during both transmission and reception and is calculated by dividing total energy consumption by total initial energy. Remarkably, the developed algorithm demonstrates significantly lower energy consumption compared to compare algorithms.

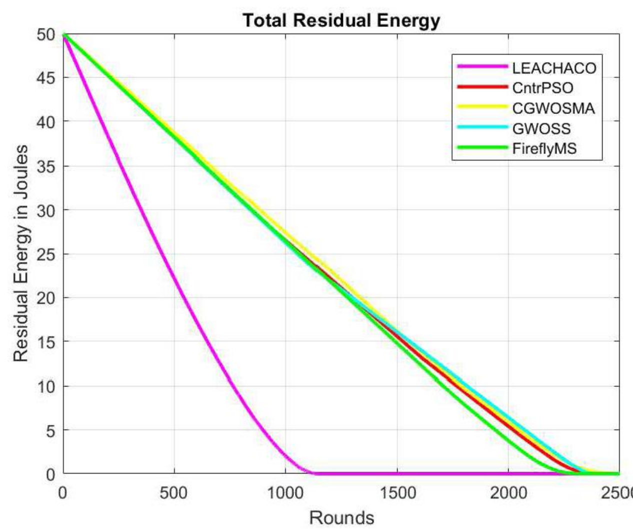

Fig. 1. Total residual energy versus time.

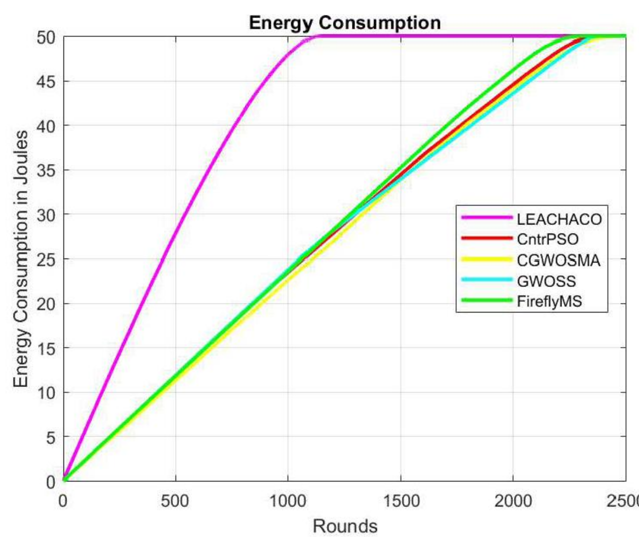

Fig. 2. Total energy consumption versus time.

On average, the energy consumption when using the proposed algorithm is lower than that of ACO [11], FA [12], PSO [13], and GWO [10] by 57.71%, 4.30%, 3.07%, and 2.46%, respectively.

Network lifetime refers to the duration for which nodes remain operational. It signifies the time span from the initiation of network operation until its conclusion, which is marked by the depletion of the last functioning sensor node [37].

Fig. 3 illustrates that it is evident that the developed algorithm sustains more alive nodes than other compared algorithms. The network lifetime is represented in terms of the percentage of alive nodes. ACO [11], PSO [13], GWO [10] algorithms, the proposed algorithm, and FA [12] algorithm, maintain 100% of alive nodes until rounds 159, 836, 1061, 1525, and 1619, respectively. However, ACO [11], FA [12], PSO [13] algorithms, the developed algorithm, and GWO [10] algorithm reach 50% of alive nodes at rounds 761, 2179, 2269, 2294, and 2343, respectively. Finally, ACO [11], FA [12], PSO [13], GWO [10] algorithms, and the implemented algorithm dwindle to 0% of alive nodes at rounds 1143, 2318, 2381, 2381, and 2490, respectively.

Thus, on average, when utilizing our proposed algorithm, the network lifetime surpasses that of ACO [11], FA [12], PSO [13], and GWO [10] by 318.17%, 34.31%, 6.85%, and 1.44%, respectively. The results highlight that employing the CGWO algorithm in the clustering formation significantly enhances network lifetime. The selection of CHs is influenced by chaotic variable, thereby extending the network's operational duration.

The primary driver behind the developed algorithm's superior performance is that the developed algorithm employs the CGWO algorithm, utilizing chaos variable. The developed algorithm utilizes a fitness function that considers both the residual energy and centrality of CHs when choosing them. While the SMA algorithm employs a fitness function based on the shortest distance between the mobile sink and CHs, contributing to enhanced energy retention. While the GWO algorithm [10] exhibits lower residual energy compared to the developed algorithm but fares better than the other compared algorithms. This is because it shares similarities with the developed algorithm in CH selection but uses the original GWO algorithm.

The PSO algorithm in [13] maintains higher residual energy than the FA and ACO algorithms, as it employs the PSO algorithm to determine the shortest path between the mobile sink and CHs. However, it lags behind the GWO and the developed algorithm because it does not use the optimized clustering approach for CH selection; it merely chooses nodes based on centrality. The FA algorithm [12] exhibits higher residual energy than the ACO algorithm as it selects CHs using the FA algorithm and employs a predictable path for the mobile sink. Nonetheless, it falls short of the PSO and GWO algorithms as well as the developed algorithm. Finally, the ACO algorithm [11] ranks the lowest due to its use of the traditional clustering approach LEACH. Although it utilizes three mobile sinks that employ the ACO algorithm to determine paths between mobile sinks and CHs, it lags significantly behind in energy retention compared to the other algorithms.

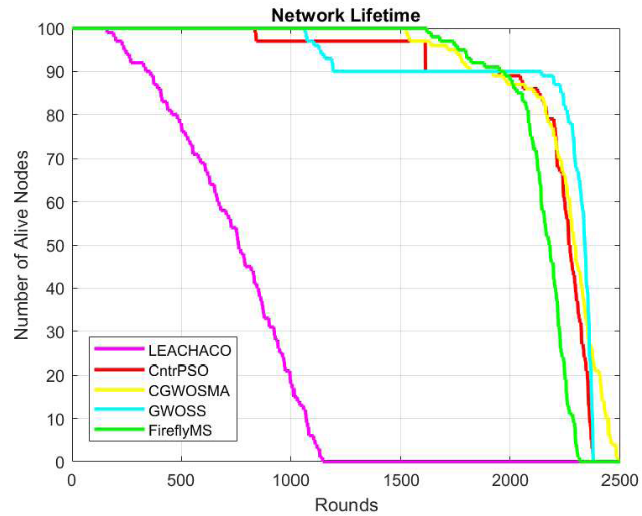

Fig. 3. Number of alive nodes versus time.

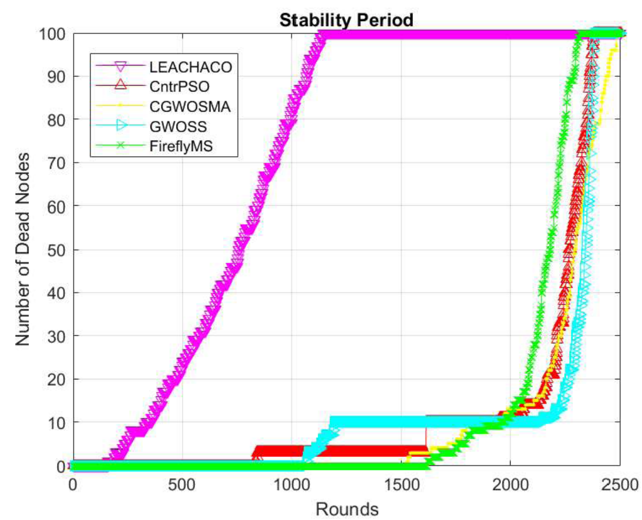

Fig. 4. Number of died nodes versus time.

At last, the stability period for all algorithms is measured. The stability period is defined as "The time when the first node died" [38]. Fig. 4 depicts when the first node in each algorithm died. Specifically, in the ACO [11], PSO [13], GWO [10] algorithms, the implemented algorithm, and FA [12] algorithm, the first node's energy depletion occurs at rounds 160, 837, 1062, 1526, and 1620, respectively.

Fig. 4 reveals that the FA [12] algorithm exhibits greater stability compared to the other algorithms. This enhanced stability can be attributed to its strategy of selecting CHs based on the FA algorithm and utilizing a predictable path for the mobile sink.

On the other hand, the implemented algorithm is less stable than the FA algorithm but more stable than the other two compared algorithms. This is primarily because it depends on node centrality in the fitness function, which is used in CH selection. This can lead to the early depletion of energy in nodes farthest from the CHs. Additionally, the fitness function of the SMA algorithm is based solely on the shortest distance from the mobile sink to the CHs.

VI. Conclusion and Future work

Clustering is one of the best techniques for WSN energy conservation. The clustering method divides the deployed sensors into groups, and each group's cluster head (CH) is selected to collect and aggregate data from other group members. Energy consumption can be decreased with the use of mobile wireless sensor networks, which allow the sink node to be moved. As a result, this paper suggests using swarm intelligence to cluster wireless sensor networks and select dynamic routes for mobile sinks.

To create clusters and locate CHs, the Chaotic Grey Wolf Optimization (CGWO) technique is employed. When figuring out the shortest route between a mobile sink and CHs using the Slime Mould Algorithm (SMA). This paper introduces a new fitness function that relies on remaining energy of the CH, the CH's membership count, Euclidean distance from a mobile sink to a CH, and CH centrality for selecting CHs and for determining the path between a mobile sink and CHs.

The performance of the proposed technique is compared with various state-of-the-art protocols. The results show that the recommended algorithm outperforms other compared algorithms in terms of overall energy consumption and network longevity. The developed approach performs better than three of the compared algorithms and is nearly as good as the fourth throughout the stability period.

As a future work, the limitations of the developed algorithm including the following aspects will be investigated. First, to improve stability period performance by refining the fitness function within the CGWO algorithm and exploring alternative swarm intelligence methods for both clustering and path formation, second to intend to employ the implemented algorithms with other mobility models where more than sink node is mobile.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding this work through Graduate Students Research Support Program.

References

[1] Obaidat, M., & Misra, S. (2014). Wireless mobile sensor networks. In Principles of Wireless Sensor Networks (pp. 248-281). Cambridge: Cambridge University Press. doi:10.1017/CBO9781139030960.012

[2] Sara, G. S., & Sridharan, D. (2014). Routing in mobile wireless sensor network: A survey. Telecommunication Systems, 57, 51-79. Available at: https://doi.org/10.1007/s11235-013-9766-2.

[3] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications magazine, 40(8), 102-114. Available at: https://doi.org/10.1109/MCOM.2002.1024422.

[4] Tunca, C., Isik, S., Donmez, M. Y., & Ersoy, C. (2013). Distributed mobile sink routing for wireless sensor networks: A survey. IEEE communications surveys & tutorials, 16(2), 877-897. Available at: https://doi.org/10.1109/SURV.2013.100113.00293.

[5] Jain, N., Sinha, P., & Gupta, S. K. (2013). Clustering protocols in wireless sensor networks: A survey. International Journal of Applied Information System (IJAIS), 5(2).

[6] Sumathi, J., & Velusamy, R. L. (2021). A review on distributed cluster based routing approaches in mobile wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 12, 835-849. Available at: https://doi.org/10.1007/s12652-020-02088-7.

[7] Wohwe Sambo, D., Yenke, B. O., Förster, A., & Dayang, P. (2019). Optimized clustering algorithms for large wireless sensor networks: A review. Sensors, 19(2), 322. Available at: https://doi.org/10.3390/s19020322.

[8] Rodrigues, L. R. (2023). A chaotic grey wolf optimizer for constrained optimization problems. Expert Systems, 40(4), e12719. Available at: https://doi.org/10.1111/exsy.12719.

[9] Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111, 300-323. Available at: https://doi.org/10.1016/j.future.2020.03.055.

[10] Agrawal, D., Wasim Qureshi, M. H., Pincha, P., Srivastava, P., Agarwal, S., Tiwari, V., & Pandey, S. (2020). GWO-C: Grey wolf optimizer-based clustering scheme for WSNs. International Journal of Communication Systems, 33(8), e4344. Available at: https://doi.org/10.1002/dac.4344.

[11] Krishnan, M., Yun, S., & Jung, Y. M. (2019). Enhanced clustering and ACO-based multiple mobile sinks for efficiency improvement of wireless sensor networks. Computer Networks, 160, 33-40. Available at: https://doi.org/10.1016/j.comnet.2019.05.019.

[12] Chauhan, V., & Soni, S. (2020). Mobile sink-based energy efficient cluster head selection strategy for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 11, 4453-4466. Available at: https://doi.org/10.1007/s12652-019-01509-6.

[13] Tabibi, S., & Ghaffari, A. (2019). Energy-efficient routing mechanism for mobile sink in wireless sensor networks using particle swarm optimization algorithm. Wireless Personal Communications, 104, 199-216. Available at: https://doi.org/10.1007/s11277-018-6015-8.

[14] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000, January). Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd annual Hawaii international conference on system sciences (pp. 10-pp). IEEE.

[15] Toor, A. S., & Jain, A. K. (2019). Energy aware cluster based multi-hop energy efficient routing protocol using multiple mobile nodes (MEACBM) in wireless sensor networks. AEU-International Journal of Electronics and Communications, 102, 41-53. Available at: https://doi.org/10.1016/j.aeue.2019.02.006.

[16] Gharaei, N., Bakar, K. A., Hashim, S. Z. M., & Pourasl, A. H. (2019). Inter-and intra-cluster movement of mobile sink algorithms for cluster-based networks to enhance the network lifetime. Ad Hoc Networks, 85, 60-70. Available at: https://doi.org/10.1016/j.adhoc.2018.10.020.

[17] Ahmed, A. E., & Ibrahim, M. E. (2018). Colored Petri Net Models for Clustered and Tree-Based Data Aggregation in Wireless Sensor Networks. International Journal on Information Technologies & Security, 10(3).

[18] Engelbrecht, A. P. (2007). Computational intelligence: an introduction. John Wiley & Sons.

[19] Abdolkarimi, M., Adabi, S., & Sharifi, A. (2018). A new multi-objective distributed fuzzy clustering algorithm for wireless sensor networks with mobile gateways. AEU-International Journal of Electronics and Communications, 89, 92-104. Available at: https://doi.org/10.1016/j.aeue.2018.03.020.

[20] Koosheshi, K., & Ebadi, S. (2019). Optimization energy consumption with multiple mobile sinks using fuzzy logic in wireless sensor networks. Wireless Networks, 25, 1215-1234. Available at: https://doi.org/10.1007/s11276-018-1715-2.

[21] Al Aghbari, Z., Khedr, A. M., Osamy, W., Arif, I., & Agrawal, D. P. (2020). Routing in wireless sensor networks using optimization techniques: A survey. Wireless Personal Communications, 111, 2407-2434. Available at: https://doi.org/10.1007/s11277-019-06993-9.

[22] Wang, J., Cao, Y., Li, B., Kim, H. J., & Lee, S. (2017). Particle swarm optimization based clustering algorithm with mobile sink for WSNs. Future Generation Computer Systems, 76, 452-457. Available at: https://doi.org/10.1016/j.future.2016.08.004.

[23] Sangeetha, M., & Sabari, A. (2018). Prolonging network lifetime and optimizing energy consumption using swarm optimization in mobile

wireless sensor networks. Sensor Review, 38(4), 534-541. Available at: https://doi.org/10.1108/SR-08-2017-0157.

[24] Cuevas, E., Fausto, F., González, A., Cuevas, E., Fausto, F., & González, A. (2020). An introduction to nature-inspired metaheuristics and swarm methods. New Advancements in Swarm Algorithms: Operators and Applications, 1-41. Available at: https://doi.org/10.1007/978-3-030-16339-6_1.

[25] Raajini, X. M., Kumar, R. R., Indumathi, P., & Praveen, V. Grey-Wolf Optimization Approach for Routing in Dynamic Wireless Sensor Network. International Journal of Applied Engineering Research, 10(32), 2015.

[26] Saranya, V., Shankar, S., & Kanagachidambaresan, G. R. (2019). Energy efficient data collection algorithm for mobile wireless sensor network. Wireless Personal Communications, 105, 219-232. Available at: https://doi.org/10.1007/s11277-018-6109-3.

[27] Amini, S. M., Karimi, A., & Shehnepoor, S. R. (2019). Improving lifetime of wireless sensor network based on sinks mobility and clustering routing. Wireless Personal Communications, 109, 2011-2024.

[28] Wang, J., Gao, Y., Liu, W., Sangaiah, A. K., & Kim, H. J. (2019). Energy efficient routing algorithm with mobile sink support for wireless sensor networks. Sensors, 19(7), 1494. Available at: https://doi.org/10.3390/s19071494.

[29] Wang, J., Cao, J., Sherratt, R. S., & Park, J. H. (2018). An improved ant colony optimization-based approach with mobile sink for wireless sensor networks. The Journal of Supercomputing, 74(12), 6633-6645. Available at: https://doi.org/10.1007/s11227-017-2115-6.

[30] Mechta, D., Harous, S., & Alem, I. (2018). Improving wireless sensor networks durability through efficient sink motion strategy: TMSRP. Wireless Personal Communications, 99, 1661-1682. Available at: https://doi.org/10.1007/s11277-018-5329-x.

[31] Pandey, S., & Anand, V. (2017). Load-balanced clustering scheme with sink mobility for heterogeneous wireless sensor networks. National Academy science letters, 40, 335-341. Available at: https://doi.org/10.1007/s40009-017-0590-1.

[32] Ibrahim, M. E., Ahmed, A. E., & Almujahed, H. (2019). A Comparative Study of Energy Saving Routing Protocols for Wireless Sensor Networks. International Journal on Information Technologies & Security, 11(2).

[33] Ibrahim, M. E., & Ahmed, A. E. (2022). Energy-aware intelligent hybrid routing protocol for wireless sensor networks. Concurrency and Computation: Practice and Experience, 34(3), e6601.

[34] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61. Available at: https://doi.org/10.1016/j.advengsoft.2013.12.007.

[35] Zheng, B., & Yang, J. (2021). Measurement method of distributed nodes in wireless sensor networks based on multiple attributes. Scientific Programming, 2021, 1-11. Available at: https://doi.org/10.1155/2021/9936337.

[36] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on wireless communications, 1(4), 660-670. Available at: https://doi.org/10.1109/TWC.2002.804190.

[37] Abo-Zahhad, M., Ahmed, S. M., Sabor, N., & Sasaki, S. (2014). A new energy-efficient adaptive clustering protocol based on genetic algorithm for improving the lifetime and the stable period of wireless sensor networks. International Journal of Energy, Information and Communications, 5(3), 47-72.

[38] Kumar, R., & Kumar, D. (2016). Hybrid swarm intelligence energy efficient clustered routing algorithm for wireless sensor networks. Journal of sensors, 2016. Available at: https://doi.org/10.1155/2016/5836913