Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 37- 48

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Training and Learning Swarm Intelligence Algorithm (TLSIA) for Selecting the Optimal Cluster Head in Wireless Sensor Networks

Ali Sedighimanesh[1], Hessam Zandhessami[1*], Mahmood Alborzi[1], Mohammadsadegh Khayyatian[2]

[1].Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
[2].Institute for science and technology studies, shahid Beheshti university, Tehran, Iran

Received: 28 Nov 2020 / Revised: 29 Apr 2021/ Accepted: 29 May 2021
DOI: https://doi.org/10.52547/jist.15638.10.37.37

Abstract

Background: Wireless sensor networks include a set of non-rechargeable sensor nodes that interact for particular purposes. Since the sensors are non-rechargeable, one of the most important challenges of the wireless sensor network is the optimal use of the energy of sensors. The selection of the appropriate cluster heads for clustering and hierarchical routing is effective in enhancing the performance and reducing the energy consumption of sensors. Aim: Clustering sensors in different groups is one way to reduce the energy consumption of sensor nodes. In the clustering process, selecting the appropriate sensor nodes for clustering plays an important role in clustering. The use of multistep routes to transmit the data collected by the cluster heads also has a key role in the cluster head energy consumption. Multistep routing uses less energy to send information.

Methods: In this paper, after distributing the sensor nodes in the environment, we use a Teaching-Learning-Based Optimization (TLBO) algorithm to select the appropriate cluster heads from the existing sensor nodes. The teaching-learning philosophy has been inspired by a classroom and imitates the effect of a teacher on learner output. After collecting the data of each cluster to send the information to the sink, the cluster heads use the Tabu Search (TS) algorithm and determine the subsequent step for the transmission of information. Findings: The simulation results indicate that the protocol proposed in this research (TLSIA) has a higher last node dead than the LEACH algorithm by 75%, ASLPR algorithm by 25%, and COARP algorithm by 10%.

Conclusion: Given the limited energy of the sensors and the non-rechargeability of the batteries, the use of swarm intelligence algorithms in WSNs can decrease the energy consumption of sensor nodes and, eventually, increase the WSN lifetime.

Keywords: Hierarchical Routing; TLBO Algorithm; TS Algorithm; Wireless Sensor Network.

1- Introduction

The wireless sensor network consists of several non-rechargeable sensor nodes applied for particular purposes [1]. One of the most important issues and challenges related to wireless sensor networks is the use of methods to reduce the energy consumption of sensor nodes. One of the methods is the clustering of the sensor nodes; instead of the sensor nodes consuming a great deal of energy and transmitting the data directly to the sink, they fall into a group called the cluster and send the data to the cluster head, and the cluster heads are required to transmit the data, thus consuming less energy of the sensor nodes and extending the network's lifetime [2]. Cluster heads can either send the received data directly to the sink or work together to send the data to the sink in a hierarchical routing process. In general, transmitting data hierarchically reduces the energy consumption of cluster heads farther from the sink [3],[4].

The process of selecting cluster heads from available sensors and the routing between clusters to transmit data to the sink are of the optimization issues; therefore, the use of optimization algorithms has an effective role in the proper performance of these two processes, and ultimately, the efficiency of the wireless sensor network [5],[6]. Teaching-Learning-Based Optimization (TLBO) algorithm is one of the modern intelligent optimization algorithms implemented in two stages (phases) and can lead to optimization through being inspired by the learning and teaching process. In the teaching phase, the best member of the community is selected as the teacher and directs the

✉ **Corresponding Author**
Zandhessami@srbiau.ac.ir

average population towards himself/herself; this is similar to what a teacher does in the real world. In the learning phase, the people in the population work together to increase their knowledge, and it is similar to what happens in the company of friends and classmates [7].

The Tabu Search (TS) [8] algorithm is also one of the most powerful algorithms for solving optimization problems, especially graph-based and combinatorial optimization problems. The TS algorithm applies a list named the taboo list, which has been designed to prevent the algorithm from falling at the local optimal point. In summary, TS starts from a point or solution and searches for neighbors around that point, chooses the best neighbor and moves to that point, and continues this search until a stopping criterion to be satisfied. The optimal point is reported at the end of the search.

In the present article, the TLBO swarm intelligence algorithm is applied to select the appropriate cluster heads from the available sensor nodes. Once the cluster heads are identified, the members of each cluster become the member of the nearest cluster head and send the data to their cluster heads. The cluster heads receive data from their members and process and aggregate them subsequently. Then, the TS algorithm is used to transmit data to the sink by cluster heads until the best routes are formed for sending data, which reduces the energy consumption of cluster heads to transfer data. The rest of the article is structured as follows. Section 2 presents the previous work. Section 3 addresses the Proposed algorithm. Section 4 discusses the findings of the article. In Section 5, the authors present open problems for wireless sensor networks, and also the results are presented.

2- Previous Works

In this research, we will address several routing protocols that have attracted interest in recent years, namely the following: LEACH, ASLPR, and COARP[9][10].

2-1- Low- Energy Adaptive Clustering Hierarchy (LEACH)

In the LEACH protocol [11], there is a probability P for each sensor to be a cluster head (CH) in every round. In other words, LEACH creates groups using a distributed algorithm, in which the sensors automatically decide to become a cluster head and there is no centralized control. Each sensor can be a cluster head only once in 1/P consecutive rounds. First, each sensor makes a decision with a probability of P to become a cluster head. The cluster head roles changes in rounds between the group nodes, and this is to create an equilibrium in the energy consumption distribution. One can divide the performance of LEACH in each round into two phases. These phases are the setup and steady-state phases. A random number between 0 and 1 is chosen by every sensor in the setup phase. If that number is smaller than T(n), the sensor n becomes a CH for that round. The value of T(n) is computed based on (1), where P is the tendency of the sensor to be a node, and r represents the round number. Moreover, G denotes the set of all sensors that have not been chosen as a cluster head during the last 1/P rounds.

$$T_{(n)} = \{\frac{P}{1 - P \times [r \; mod \left(\frac{1}{P}\right)]} \qquad if \; n \in G \; 0 \qquad Otherwise \tag{1}$$

After the cluster heads are selected, they are announced to all the sensors in the network as cluster heads. When non-cluster head sensor receives an announcement from the cluster heads, it selects the cluster head closest in terms of communication.

2-2- Application- Specific Low Power Routing (ASLPR) protocol

The ASLPR protocol [12] collects specific pieces of information, such as remaining energy, distance from the base station, and distance between the CHs and sensor node, to select the cluster head nodes. Then, each node selects a random number between zero and 1. If the random number selected by a node is less than T_{ASLPR} in (2), this node is converted to a cluster head.

$$T_{ASLPR} = \begin{cases} Z(n) & if \; E(n) \geq t_1 \times \frac{1}{N}\sum_{i=1}^{N} E(i) \\ 0 & if \; E(n) < t_1 \times \frac{1}{N}\sum_{i=1}^{N} E(i) \end{cases} \tag{2}$$

$$Z(n) = \alpha_1 T_1(n) + \alpha_2 T_2(n) + \alpha_3 T_3(n) + \alpha_4 T_4(n) \tag{3}$$

In the above relationships, N represents the total number of live nodes in the current round, and E(n) equals the n remaining nodes.

In (3), $T_1(n)$ denotes the sub-threshold of the node energy, and α_1 refers to the weight of this sub-threshold. Moreover, $T_2(n)$ represents the sub-threshold for the distance between the nodes and the base station, and α_2 denotes the weight of this sub-threshold. In addition, $T_3(n)$ is the sub-threshold for the distance between the node and the cluster head, and α_3 refers to the weight of this sub-threshold. The sub-threshold $T_4(n)$ denotes the number of rounds where a node has been the cluster head, and α_4 represents the weight of this sub-threshold. Then, the cluster head nodes announce their existence to all the nodes in the network by issuing a message. After receiving this message from different cluster heads, the regular (non-cluster head) nodes select the closest cluster head to join. In this protocol, genetic algorithm (GA) combined with the simulated annealing (SA) algorithm has been used to optimize the special parameters utilized for determining the threshold for application-specific cluster heads. The objective functions of the GA and SA algorithms in this protocol are defined as follows:

$$Maximize{:}\,fitness = W_1 \times FND \times W_2 \times HND + W_3 \times LND \qquad (4)$$

$$0 \le \alpha_k \le 1 \qquad (k = 1,2,3,4),\ \sum_{k=1}^{4} \alpha_k = 1 \qquad (5)$$

$$0 \le t_s \le 2 \qquad (k = 1,2,3,4),\ t_1 \le t_2 \qquad (6)$$

$$0 \le W_u \le 1 \qquad (k = 1,2,3),\ \sum_{k=1}^{3} W_u = 1 \qquad (7)$$

In the above relationships, W_1, W_2, and W_3 denote the weights of the First Node Dead (FND), Half Node Dead (HND), and Last Node Dead (LND), respectively. The ranges of the mentioned weights are between 0 and 1, depending on the application, such that their sum equals 1 according to (7). Moreover, t_s refers to the sub-threshold values in (3), and α_k in (4) represents the weight of the sub-threshold in (3).

2-3- Cuckoo Optimization Algorithm - Based Routing Protocol (COARP)

In COARP [13], measurements to determine the CHs are performed within a centralized control system. The model of the network is a single-step model where the CHs communicate directly with the base station. During every round, the base station is aware of the position and energy level of the nodes in the network. During each round, every node sense and gathers the surrounding data. Then, it processes the data and sends it to the cluster head in a data packet form. The COARP clustering method involves the following steps: (1) the start-up phase, which involves determining the cluster head and creating the cluster, (2) the register phase, which involves creating a data scheduling and transmission plan. In CAORP, the CHs are accurately chosen by the cuckoo algorithm in the base station. Then, the cluster creation process and the register phase are performed. Every CH receives the information relating to all the nodes belonging to its own cluster. Then, it sends the received information to the base station in the form of a packet.

3- Proposed Algorithm

The appropriate selection of cluster heads from the available sensor nodes is one of the methods that lead to the reduction of the energy consumption of sensor nodes and cluster heads. Besides, the data transmission in a hierarchical manner instead of the one-step method highly affects the reduction of the energy consumption of sensors since the farther apart the two nodes are, the more energy they have to expend for data transmission. Therefore, selecting the appropriate cluster head from the available nodes and the hierarchical routing can lead to the reduction of the energy consumption of the sensor nodes, which will increase the lifetime of the wireless sensor network. For this purpose, there are various methods; the application of optimization methods for solving such problems will enhance decision-making and increase the efficiency of algorithms.

The proposed algorithm described in three sections: sensor node distribution, clustering process, routing. In the sensor node distribution section, the authors explain how to distribute the nodes in the simulation environment. In the clustering section, there is an attempt to classify sensor nodes into different clusters for the purpose of reducing energy consumption. For this purpose, a swarm intelligence algorithm called TLBO is employed to select the optimal cluster heads from the sensor nodes. In the routing section, the objective is to apply the best routes to transmit data hierarchically with less energy consumption; hence, the TS algorithm is used to choose the best route for data transmission. In the following, the authors will explain these steps step by step. The general algorithm of the proposed algorithm is as follows.

TLSIA Algorithm

```
Select nodes in sensing area for clustering
CHs= TLBO
For i=1: number of nodes
    If node(i) is in sensing area && node(i) is normal node
        node(i) joins to nearest CH
    end if
end for
Routing to send cluster head information
Route= TS
For i= Cluster heads
    CH(i) joins to route;
end for
```

3-1- Node Distribution and Sink Location

During the simulation, the sensor nodes are randomly distributed in an environment. Then, the location of the sink is determined, which is usually outside the environment.

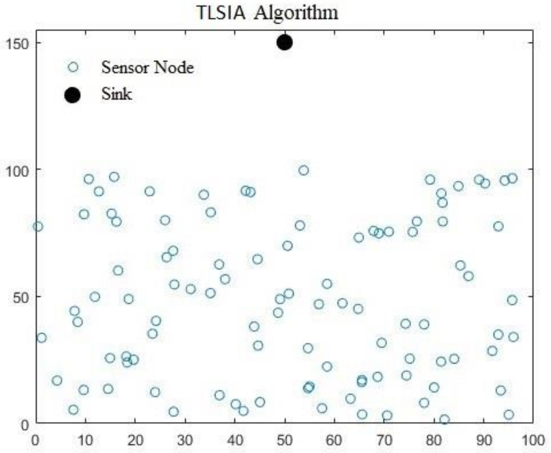

Fig. 1. Random distribution of the nodes in the environment

3-2- Cluster Head Selection

The process of choosing the optimal cluster heads from between the sensors in the network is performed using the Teaching-Learning-Based Optimization (TLBO) algorithm. The teaching-learning philosophy has been inspired by a classroom and imitates a teacher’s effect on the learner output. Similar to other swarm intelligence algorithms, the TLBO algorithm is a population-based evolutionary optimization algorithm and consists of a teaching phase and a learner phase.

In the teaching phase, the teacher has the main role and attempts to transfer their knowledge to all the learners in the classroom to increase the average score. The average result of the learners and the improvement in results completely depends on the teacher. In each step, the best learner in the population is selected as the teacher, and, accordingly, the cost function and the average position for improving the position of the learners are computed.

In the learning phase, the learners increase their knowledge either via the teacher or via interacting with each other. The main difference between the teaching and learning phases is that in the teaching phase, the teacher transfers the knowledge to the learners, but in the learning phase, the learners gain knowledge from the teacher and by communicating with each other. In population-based optimization methods, a population has a set of members, each of which has a number of variables. Every member of the population is a solution to the optimization problem. In this paper, we first form an initial population consisting of a number of members, named learners, to determine the cluster head. Each learner includes 2 variables: Position, which consists of a string of variables, and cost. The figure below shows an overview of a population.

Learner 01	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						
Learner 02	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						
				⋮			
Learner 0N	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						

Fig. 2. Overview of a population

First, the variables inside the position are given a random value between 0 and 1 ($0 \leq$ Position $(i) \leq 1$). The most important issue in optimization algorithms is how to determine the cost for the learners in the population. In this paper, the cost is equal to (8):

$$\text{Cost} = \text{Sum}(\text{Alpha} \times \text{RE}(x), \text{Beta} \times \text{Density}(x), \text{Gamma} \times \text{Centerality}(x)) \qquad (8)$$

In the above formula, x is the variable inside the population member, RE is the remaining energy of each variable, density is the ratio of the number of neighbors to the total number of nodes, centrality is the sum of distances of the nodes from the neighbors, Beta= -0.3, Alpha= -0.5, and Gamma=0.2.

In the TLBO method, every member of the population is considered a learner. In every iteration of the TLBO algorithm, we select the member with the lowest cost between the population members as the best member of the population. Then, we sort the variables inside the selected member in descending order and select 10% of these variables as the optimal cluster head. For example, if after the end of the maximum iteration of the algorithm, the output is as follows:

		01	02	03		n-1	n
Learner 01	Position	0.36	0.47	0.25	….	0.12	0.22
	Cost= -1.25						
Learner 02	Position	0.26	0.17	0.45	….	0.32	0.52
	Cost=-1.35						
				⋮			
Learner N	Position	0.14	0.32	0.54	….	0.33	0.63
	Cost=-1.05						

Learner 02 is selected as the best member of the population; hence, the variables inside this member are sorted in descending order, and 10% of them are considered as the cluster head.

In implementing the TLBO algorithm, 3 values have a vital role in the optimal performance of the algorithm: (1) initialization of the learners, (2) updating of the teaching phase, and (3) updating of the learning phase.

Learner initialization: In this method, we first create a random population and calculate the second population from the first using (9). Subsequently, we combine the 2 populations and compute and sort the costs of the learners. Then, we select from the learners with less cost a number equal to the learner members of the population[14], [15].

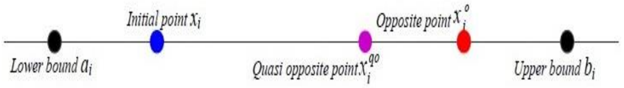

Fig. 3. Opposition-based learning and quasi-oppositional learning[15].

$$x_i^o = a_i + b_i - x_i \tag{9}$$

$$x_i^{qo} = \frac{a_i + b_i}{2} + rand.\left(x_i^o - \frac{a_i + b_i}{2}\right) \tag{10}$$

Teaching phase: In the teaching phase, the learners increase their knowledge via learning from the difference between the class average and the teacher. The update mechanism for the ith learner has been expressed as follows:

$$newX_i = X_i + rand.(Teacher - TF.Meam) \tag{11}$$

$$Mean = \frac{1}{NP}\sum_{i-1}^{NP} X_i \tag{12}$$

$newX_i$ is the learner's new state, X_i is the ith learner, Teacher is the learner with the best fitness, NP denotes the number of learners present in the population, and TF is a teaching factor that determines the value of the average that must be changed. Also, rand is a random vector the element of which is a random number in the range [0, 1].

Learning phase: During the learning phase, the learners also increase their knowledge interactively. The update mechanism for the ith learner has been expressed as follows:

$$newX_i = \begin{cases} X_i + rand.(X_i - X_k) & if\ f(X_i) < f(X_k) \\ X_i + rand.(X_i - X_k) & otherwise \end{cases} \tag{13}$$

where newX_i is the ith learner's position, X_k represents the learners chosen randomly from the class, and $f(X_i)$ and $f(X_k)$ respectively denote the fitness values of the learners X_i and X_k. In addition, rand denotes a random vector in the [0, 1] range.

TLSIA Clustering Algorithm

```
Initialize learners;
Evaluate learners;
For all learners
For i=each dimension
    $x_i^o = a_i + b_i - x_i$
    $x_i^{qo} = \frac{a_i+b_i}{2} + rand.\left(x_i^o - \frac{a_i+b_i}{2}\right)$
End_For
End_For
Combine first population and Quasi-opposite population;
Select best learners as new population;
Xteacher=best learner;
Xmean=average of learners;
While (stopping condition is not met)
        For i=all learners
        TF = round (1 + rand (0,1));
        Xnewi=Xi+rand*(Xteacher-TF*Xmean);
        End_For
        Evaluate new learners;
        If new learner is better than old one
            Xi=Xnewi;
        End_If
        For i=all learners
            Randomly select another learner which is
different from i (Xk);
            If Xi is better than Xk
                Xnewi=Xi+rand*(Xi-Xk);
            Else
                Xnewi=Xi+rand*(Xk-Xi);
            End_If
        End_For
        If new learner is better than existing
            Xi=Xnewi;
        End_If
        Xteacher=best learner;
        Xmean=average of learners;
End_While
```

3-3- Routing

We use the TS algorithm for routing and transferring the data collected by the cluster heads to the sink. The TS algorithm consists of a solution that includes a string of position and cost variables. The number of position variables equals the number of cluster heads minus 1 (N_Ch-1). The figure below shows a view of the solution in the TS algorithm.

Solution	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						

In the TS algorithm, a number of actions are performed on the solution variables so as to optimize the solution cost. These actions are reversion, swap, and insertion.

To optimize routing using the TS algorithm, we use the Prüfer algorithm [16] to create a tree between the cluster head nodes. This algorithm maps a sequence of numbers to the corresponding tree.

First, we create a solution that assigns a random number between 0 and 1 to each position variable. Then, the solution cost is computed. To calculate the cost of each solution, we first convert it to the corresponding tree using the Prüfer algorithm. Then, the routing is performed according to the obtained tree, and the cost is calculated from (14). E_1 is the network energy before applying the routing, and E_2 is the computed energy after applying the routing.

$$Cost = E_1 - E_2 \tag{14}$$

Given the actions considered in the TS algorithm, all the states relating to these actions are created in a list named Action List. We perform these actions on the obtained solution and update the cost and position for each action. If a lower cost results, it replaces the best solution, and the corresponding action is placed in the Tabu List and is not performed for a specific number of rounds. The desired number of actions is computed using (15).

$$\begin{aligned} & N_{Action} = N_{Swap} + N_{Reversion} + N_{Insertion} \\ & N_{Swap} = n \times \frac{(n-1)}{2} \\ & N_{Reversion} = n \times \frac{(n-1)}{2} \\ & N_{Insertion} = n \times n \qquad \text{(n is the number of position variables.)} \\ & Tabu\ List = Round(0.5 \times N_{Action}) \end{aligned} \tag{15}$$

This is continued until the best solution is obtained. Finally, the obtained solution is given to the Prüfer algorithm, the output of which is an optimal tree according to which the routing is performed. For example, assume the number of cluster heads is 10 in a known round. First, the number of variables inside the solution of the TS algorithm is equal to 9. We consider a random number between 0 and 1 for each variable and compute the initial solution cost.

		01	02	03	04	05	06	07	08	09
Solution	Position	0.37	0.26	0.88	0.76	0.44	0.55	0.45	0.35	0.25
	Cost= 2									

For the cost, we first convert this position into integers and obtain the corresponding array 5 4 10 9 5 7 6 5 4. This array is given to the Prüfer algorithm, and an equivalent tree is created. The cost is equal to the energy consumption of the cluster heads during the transmission of information to the sink according to this tree and route, and we seek to reduce the cost of the problem solution. After the initial solution is known, the desired actions are applied according to the obtained solution and the position and cost of the optimal solution are obtained. if a lower cost results, it replaces the best solution, and the action is not performed for a specific amount of time. This is continued until the cost is optimized. Finally, the obtained solution is converted to a tree via the Prüfer algorithm, which represents our optimal route.

TLSIA Routing Algorithm

```
Create initiate solution;
Sbest=best solution;
While (stopping condition is not met)
        Generate candidate solutions in the neighborhood of Sbest
        For i=candidate solutions
                If candidate_i is not in TabuList
                        If candidate_i is better than bestnewsol
                                Bestnewsol=candidate_i
                        End_If
                End_If
        End_For
        If bestnewsol is better than Sbest
                Sbest= bestnewsol
        End_If
        Push the bestnewsol to TabuList
        If TabuListSize>maxTabuListSize
                Remove the first element from TabuList;
        End_If
End_While
```

3-4- Network Operations and Energy Consumption Computation

The network operations in the proposed algorithm are divided into start-up and register phases. The energy consumption of every node in each round is computed by examining what has occurred in both phases.

3-4-1- Start-up Phase

The sink uses the k_{CP} control packet to communicate with the sensor nodes. These k_{CP} control packets contain short messages that request the ID, position, and the level of energy from each of the sensor nodes. The energy $E_{Rx}(k_{CP})$ is consumed in the process of receiving the control packets from the sink according to (16). Moreover, all the nodes utilize the energy $E_{Tx}(k_{CP}, d)$ to transfer to the sink the control packets that contain data relating to the IDs, positions, and levels of energy.

$$E_{Rx}(k) = kEelect \tag{16}$$

$$E_{Tx}(k,d) = \begin{cases} kE_{elect} + \varepsilon_{mp} k d^4, & if\ d > d_0 \\ kE_{elect} + \varepsilon_{fs} k d^2, & if\ d \leq d_0 \end{cases} \tag{17}$$

Where $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$ is the threshold distance. The amplifier energy ε_{mp} or ε_{fs} is based on the distance of the receiver and the acceptable bit error. The sink processes the control packets and, according to the proposed algorithm, determines which nodes will be cluster heads and which cluster head each node will become a member of. Moreover, all the nodes (CH or other nodes) use the energy $E_{Rx}(k_{CP})$ to receive their status information from the sink. The energy consumed by the CHs to send TDMA (Time-division multiple access) schedules to their respective members is obtained by the following relationship:

$$E_{Tx(ch_i)}(K_{CP}, d_{i-tomem}) = \sum_{i=1} ch_i * \begin{cases} K_{CP}E_{elect} + \varepsilon_{mp}K_{CP}d^4_{i-tomem}, & if\ d > d_0 \\ K_{CP}E_{elect} + \varepsilon_{fs}K_{CP}d^2_{i-tomem}, & if\ d \le d_0 \end{cases} \quad (18)$$

The member consumes energy to receive the TDMA schedules from the cluster head, which is computed from (16).

3-4-2- Register Phase

In the register phase, the active nodes send k-bit data to their respective cluster heads in terms of the TDMA schedule they have received from the sink. The cluster head is always ready to receive these sensed data from its members and processes and aggregates all the data received from its members before sending them to the sink. The energy consumed by the cluster head sensor transmitter to perform work, i.e., E_{DA}, is computed from (19).

$$E_{DA(m_i+1)}(k) = KE_{DA} * (\sum_{i=1} m_i + 1) \quad (19)$$

The energy lost in the transmission of the sensed data to the cluster head is calculated using the following relationship:

$$E_{Rx(m_i)}(k) = \sum_{i=1} m_i KE_{elec} \quad (20)$$

where m_i denotes the member nodes of the series $i = 1, 2, 3, \ldots, n - L$, and n and L represent the total numbers of sensor nodes and cluster heads, respectively. The energy consumed by the cluster head to collect the sensed data from the members and itself is determined via (19), as follows.

4- Findings:

All the experiments were conducted within MATLAB R2019b. To prove the efficiency, we compare the proposed algorithm to known protocols such as LEACH, ASLPR, and COARP based on FND, HND, LND, and the total number of data packets received at the sink from the start of the simulation to the end of the network lifetime.

4-1- Network Model Assumptions

The important assumptions for the network model and the radio model in the proposed algorithm are as follows:

- ❖ The sink is a fixed device and a rich source located outside the simulation environment.
- ❖ All the sensors are stable after deployment, and the average energy in the homogeneous or heterogeneous environment is constant.
- ❖ All the sensors are equipped with the Global Positioning System (GPS) or connected to other geographical positioning systems.
- ❖ The communication channel is considered to be symmetric.

Table 1: Adjusting the parameters of the TLBO algorithm

Parameter	Value
Population or Learner	50
Number of iterations	100
Number of Variables	length (Alive Nodes)
Variables Lower Bound	VarMin= 0
Variables Upper Bound	VarMax=1

Table 2: Adjusting the parameters of the TS algorithm

Parameter	Value
Population or Solution	1
Number of iterations	100
Number of Variables	Nch-1 (Nch= Number of Cluster Head)
Variables Lower Bound	VarMin= 0
Variables Upper Bound	VarMax=1
N_{Action}	N_{Swap}+$N_{Reversion}$+$N_{Insertion}$
N_{Swap} = $N_{Reversion}$	N × (N-1)/2
$N_{Insertion}$	N × N (N=Number of position variables)

Table 3: Simulation parameters

Parameter	Value
Initial energy of the nodes	1j
ε_{fs}	10 (pj/bit/m^2)
ε_{mp}	0.0013 (pj/bit/m^4)
E_{elec}	50 (nJ/bit)
E_{da}	5 (nJ/bit)
Data packet size	4100 (bit)

4-2- Simulation Results

In this section, the authors take into account eight scenarios according to Table (4) to evaluate the proposed algorithm. The number of sensors, the size of the environment, and the sink location are the parameters investigated in these scenarios to evaluate the algorithms in which the parameters change in each scenario.

Table 4: Used scenarios

Number	Number of sensors	Network size	Sink location
1	100	200m × 200m	(100m, 250m)
2	100	500m × 500m	(250m, 550m)
3	200	200m × 200m	(100m, 250m)
4	200	500m × 500m	(250m, 550m)
5	500	200m × 200m	(100m, 250m)
6	500	500m × 500m	(250m, 550m)
7	2000	200m × 200m	(100m, 250m)
8	2000	500m × 500m	(250m, 550m)

According to Table (4), the scenarios are simulated in two environments of sizes 200m×200m, and 500m×500m and the number of sensor nodes 100, 200, 500, and 2000, and their results are analyzed. Three factors are investigated in these scenarios: 1) the number of live nodes, 2) energy consumption of the network, 3) packets sent to the sink in each round.

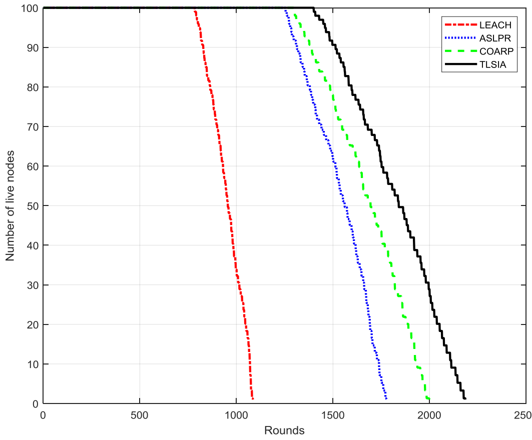

Fig. 4. Number of live nodes in each round in the first scenario.

According to the results obtained in Figure (4) in the first scenario, FND [1], HND [2] and LND [3] in the proposed algorithm are better compared to other approaches and indicates that in the Proposed algorithm, the energy consumption of sensors in each round is less than other methods. In Figure (5), the network's lifetime has been compared; in the Proposed algorithm, the networks' lifetime has increased compared to other methods, which shows the proper performance of the proposed algorithm in clustering and data transmission.

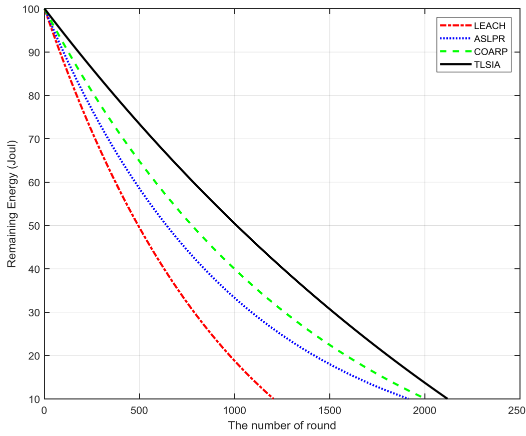

Fig. 5. Network's energy consumption in each round in the first scenario.

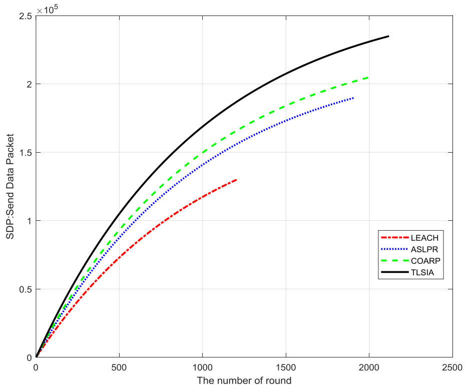

Fig. 6. Packets sent to the sink in each round in the first scenario.

In the simulations, the higher the number of intact packets sent to the sink, the better the performance of the sensor nodes and cluster heads, which leads to an increase in the performance of the wireless sensor network. As shown in Figure (6), in the Proposed algorithm, the number of packets sent to the sink in each round is more than other methods, which indicates the proper performance of the sensor nodes and cluster heads within the wireless sensor network in the TLSIA method.

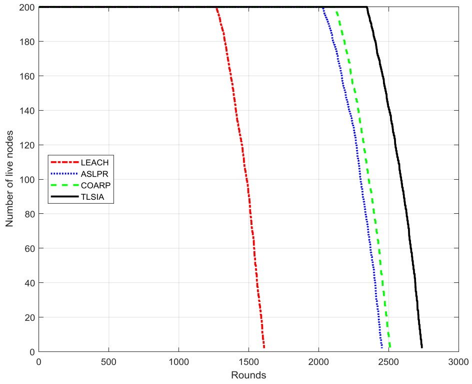

Fig. 7. Number of live nodes in each round in the third scenario.

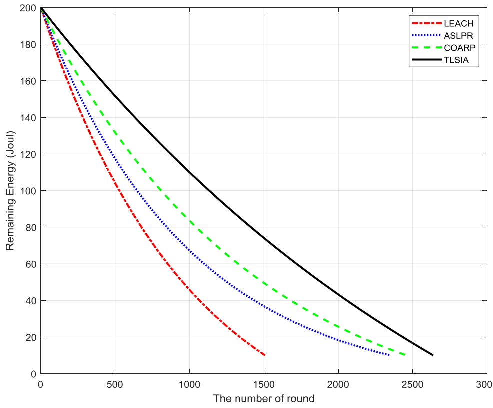

Fig. 8. Network's energy consumption in each round in the third scenario.

[1] First Node Dead
[2] Half Node Dead
[3] Last Node Dead

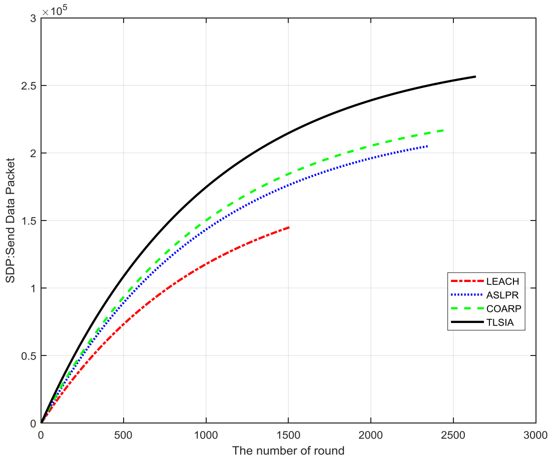

Fig. 9. Packets sent to the sink in each round in the third scenario.

The difference between the first and third scenarios is the number of nodes distributed in the simulation environment. The increase in the number of sensor nodes and the constant size of the simulation environment has led to an increase in the two factors of live nodes and packets sent to the sink in each round, which is true for all comparable algorithms. The results obtained from Figures 7, 8, and 9 indicate that the TLSIA algorithm outperforms the investigated algorithms. This performance includes the number of live nodes, the network's lifetime, and the number of packets sent to the sink in each round.

In Table 5, the authors compare and evaluate FND, HND, and LND factors of the proposed algorithm (TLSIA) compared to other methods in the first four scenarios.

Table 5: Comparison of FND, HND, and LND of TLSIA method with other methods in the first four scenarios.

			Network Size= 200 m × 200 m Sink location= (100 m, 250 m)			Network Size= 500 m × 500 m Sink location= (250 m, 500 m)		
			FND	**HND**	**LND**	**FND**	**HND**	**LND**
Number of sensor nodes	100	**LEACH**	780	1155	1204	2	27	43
		ASLPR	1248	1848	1914	4	43	69
		COARP	1294	1940	2004	5	47	73
		TLSIA	1388	2032	2119	8	54	85
	200	**LEACH**	1507	1431	1268	4	42	61
		ASLPR	2347	2289	2028	7	68	98
		COARP	2456	2361	2117	8	72	105
		TLSIA	2637	2547	2345	11	84	120

The difference between the first, second, third, and fourth scenarios is in the number of sensor nodes and the size of the simulation environment. Increasing the number of network's sensor nodes in these scenarios leads to an increase in the network's lifetime and the number of packets sent to the sink in each round, but increasing the size of the environment leads to a decrease in the network's lifetime and the number of packets sent to the sink in each round. As indicated in Table (5), with increasing the number of sensors as well as the network's size, the TLSIA method is better in terms of FND, HND, and LND in comparison with other techniques. These results mean that the TLSIA algorithm performs better in selecting cluster heads and routing the collected data compared to other methods, which reduces the energy consumption of sensor nodes and cluster heads.

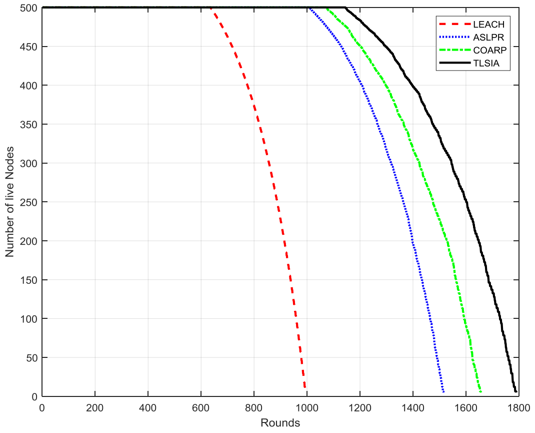

Fig. 10. Number of live nodes in each round in the fifth scenario.

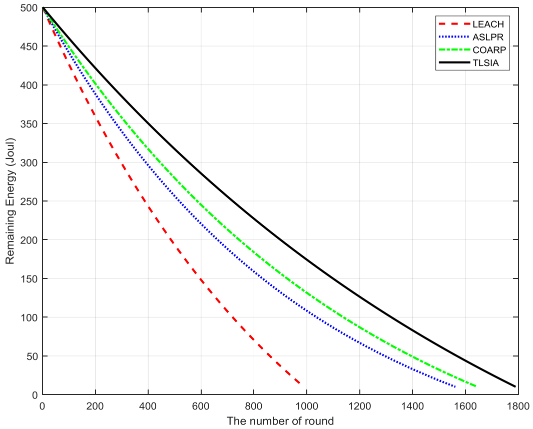

Fig. 11. Network's energy consumption in each round in the fifth scenario.

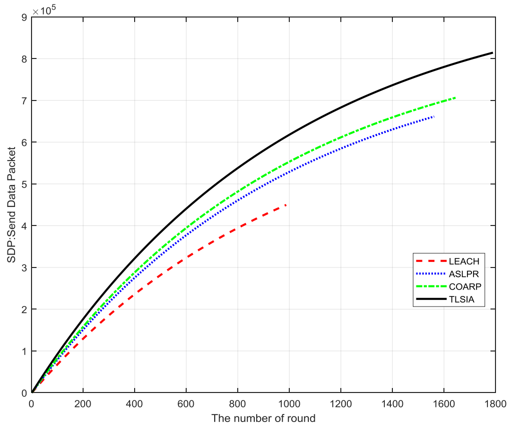

Fig. 12. Packets sent to the sink in each round in the fifth scenario.

In Scenario 5, the number of sensor nodes was considered to be 500, the size of the simulation environment to be 200m×200m, and the nodes being randomly distributed in the environment. According to the results obtained in Figures 10, 11, and 12, it can be concluded that an excessive increase in the number of sensors in the simulation environment has an adverse effect on network's performance since a large number of sensor nodes are distributed in a small environment which leads to an increase in the useless interactions between the sensors as well as an increase in the cluster heads' load, causing energy consumption and rapid discharge of cluster heads. Thus, there must be a tradeoff between selecting the number of sensor nodes and the size of the simulation environment to reach an optimal performance of this network.

In Scenario 7, by increasing the number of sensor nodes to 2000, the results indicate that the TLSIA method is more efficient than other approaches. According to Figures (13), (14), and (15), the TLSIA method outperforms other techniques in terms of the number of live nodes, network's lifetime, and the number of packets sent to the sink.

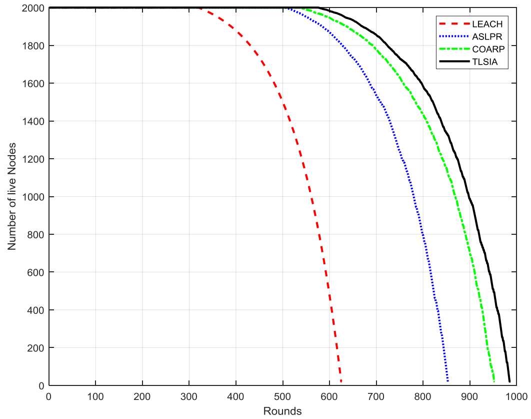

Fig. 13. Number of live nodes in each round in the seventh scenario.

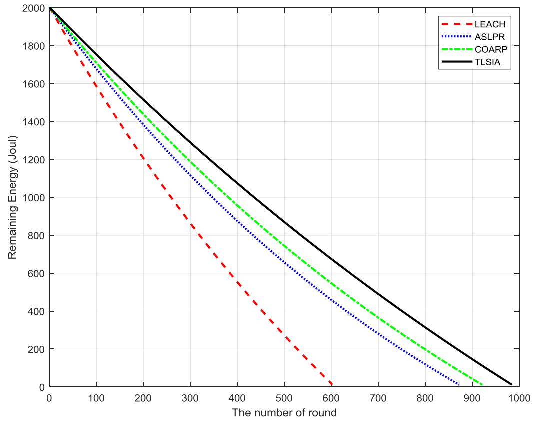

Fig. 14. Network's energy consumption in each round in the seventh scenario.

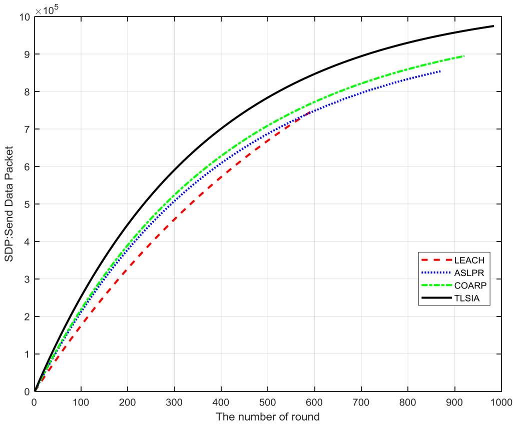

Fig. 15. Packets sent to the sink in each round in the seventh scenario.

In Table (6), the results of the proposed algorithm (TLSIA) are evaluated and compared to other methods in FND, HND, and LND modes. These results are related to the last four scenarios presented in Table (4).

Table 6: Comparison of FND, HND, and LND of TLSIA method with other methods in the second four scenarios.

			Network Size= 200 m × 200 m Sink location= (100 m, 250 m)			Network Size= 500 m × 500 m Sink location= (250 m, 500 m)		
			FND	**HND**	**LND**	**FND**	**HND**	**LND**
Number of sensor nodes	500	**LEACH**	636	910	988	60	102	127
		ASLPR	1005	1456	1562	95	164	205
		COARP	1072	1530	1645	100	174	212
		TLSIA	1145	1657	1790	112	185	234
	2000	**LEACH**	320	405	552	20	30	46
		ASLPR	505	645	872	32	49	74
		COARP	535	680	921	35	52	78
		TLSIA	575	720	984	48	64	92

The results of these four scenarios also indicate that by increasing the number of sensor nodes and also increasing the size of the environment, the TLSIA method has performed better compared to the other methods in terms of the three studied factors: FND, HND, and LND.

By evaluating the proposed scenarios, it can be concluded that some variables such as the size of the simulation environment and the number of sensor nodes distributed in the environment have a significant impact on the energy consumption of the sensors, cluster heads, and the performance of the wireless sensor network. Therefore, one of the significant challenges in such networks is establishing a proper fit between the network size and the number of sensors.

5- Discussion and Conclusion

The wireless sensor networks include a set of sensor nodes designed and applied for particular purposes; hence, energy-saving is considerably important due to the non-rechargeability of sensor nodes. Selecting the appropriate cluster head from the sensor nodes and the hierarchical routing has a significant effect on reducing the energy consumption of the sensor nodes. Different routing protocols, including the LEACH, ASLPR, and COARP protocols, have been proposed to achieve energy efficiency in wireless sensor networks. The purpose of all protocols is to extend the lifetime of wireless sensor networks. In order to achieve this objective, the authors applied the teaching-learning-based optimization algorithm, which consists of two phases: Teaching Phase, Learner Phase. The authors selected the appropriate nodes from the sensor nodes in the network using the TLBO swarm intelligence algorithm, which led to the formation of suitable clusters to reduce the energy consumption of the sensor nodes. After selecting the cluster head and also the clustering operations, the collected data was sent to the sink through a multistage (hierarchical) method by the TS algorithm. This method reduced the energy consumption of the cluster heads when sending data to the sink.

According to the simulation results of the proposed algorithm in this article, the TLSIA algorithm outperformed other compared algorithms in different conditions with increasing the number of sensor nodes and also the size of the simulation environment and also the network's lifetime has been increased. as well as, the proposed TLSIA algorithm can decrease the energy consumption of the nodes and increase the network life. In terms of HND, FND, and LND, the proposed algorithm has had an increase of about 75%, 15%, and 10% compared to the LEACH, ASLPR, and COARP algorithms, respectively. Moreover, the number of packets transmitted to the sink in the proposed algorithm has increased compared to that in other methods. The following suggestions can be made for future works to improve and develop the Proposed algorithm:

1) Mobilization of the sensor nodes inside the network to suitable clustering.
2) Movement of the sink around the network environment to collect the information sensed by the sensor nodes.
3) Use of ensemble learning algorithms in the selection of the cluster head.

6- References

[1] A. Belfkih, C. Duvallet, and B. Sadeg, “A survey on wireless sensor network databases,” Wirel. Networks, vol. 25, no. 8, pp. 4921–4946, 2019.

[2] M. Sedighimanesh* and H. Z. and A. Sedighimanesh, “Presenting the Hybrid Algorithm of Honeybee - Harmony in Clustering and Routing of Wireless Sensor Networks,” International Journal of Sensors, Wireless Communications and Control, vol. 9, no. 3. pp. 357–371, 2019.

[3] A. Kochhar, P. Kaur, P. Singh, and S. Sharma, “Protocols for wireless sensor networks: A survey,” Journal of Telecommunications and Information Technology. 2018.

[4] Z. Ullah, “A Survey on Hybrid, Energy Efficient and Distributed (HEED) Based Energy Efficient Clustering Protocols for Wireless Sensor Networks,” Wirel. Pers. Commun., vol. 112, no. 4, pp. 2685–2713, 2020.

[5] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, “Clustering objectives in wireless sensor networks: A survey and research direction analysis,” Comput. Networks, vol. 180, p. 107376, 2020.

[6] S. A. Susan T and B. Nithya, “Cluster Based Key Management Schemes in Wireless Sensor Networks: A Survey,” Procedia Comput. Sci., vol. 171, pp. 2684–2693, 2020.

[7] P. Sarzaeim, O. Bozorg-Haddad, and X. Chu, “Teaching-Learning-Based Optimization (TLBO) Algorithm BT - Advanced Optimization by Nature-Inspired Algorithms,” O. Bozorg-Haddad, Ed. Singapore: Springer Singapore, 2018, pp. 51–58.

[8] M. Gendreau, “An Introduction to Tabu Search,” in Handbook of Metaheuristics, 2006.

[9] U. E. Zachariah and L. Kuppusamy, “A hybrid approach to energy efficient clustering and routing in wireless sensor networks,” Evol. Intell., 2021.

[10] F. Fanian and M. K. Rafsanjani, “Cluster-based routing protocols in wireless sensor networks: A survey based on methodology,” J. Netw. Comput. Appl., vol. 142, pp. 111–142, 2019.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. p. 10 pp. vol.2, 2000.

[12] M. Shokouhifar and A. Jalali, “A new evolutionary based application specific routing protocol for clustered wireless sensor networks,” AEU - Int. J. Electron. Commun., vol. 69, no. 1, pp. 432–441, Jan. 2015.

[13] M. Khabiri and A. Ghaffari, “Energy-Aware Clustering-Based Routing in Wireless Sensor Networks Using Cuckoo Optimization Algorithm,” Wirel. Pers. Commun., vol. 98, no. 3, pp. 2473–2495, 2018.

[14] P. K. Roy, C. Paul, and S. Sultana, “Oppositional teaching learning based optimization approach for combined heat and power dispatch,” Int. J. Electr. Power Energy Syst., 2014.

[15] W. Shao, D. Pi, and Z. Shao, “An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem,” Appl. Soft Comput. J., 2017.

[16] X. Wang, L. Wang, and Y. Wu, “An Optimal Algorithm for Prufer Codes,” JSEA, vol. 2, pp. 111–115, Jan. 2009.