



Intelligent Approaches for Enhancing Networked Routing Protocol

Ruwaida M. Yas*¹, Soukaena H. Hashem²

¹Iraqi Commission for Computer & Informatics-Information Institute for Postgraduate Studies

²University of Technology-Computer Science Department

Received:3/11/2020

Accepted: 23/1/2021

Abstract

The rapid evolution of wireless networking technologies opens the door to the evolution of the Wireless Sensor Networks (WSNs) and their applications in different fields. The WSN consists of small energy sensor nodes used in a harsh environment. The energy needed to communicate between the sensors networks can be identified as one of the major challenges. It is essential to avoid massive loss, or loss of packets, as well as rapid energy depletion and grid injustice, which lead to lower node efficiency and higher packet delivery delays. For this purpose, it was very important to track the usage of energy by nodes in order to improve general network efficiency by the use of intelligent methods to reduce the energy used to extend the life of the WSN and take successful routing decisions. For these reasons, designing an energy-efficient system that utilizes intelligent approaches is considered as the most powerful way to prolong the lifetime of the WSN. The proposed system is divided into four phases (sensor deployment phase, clustering phase, intra-cluster phase, and inter-cluster phase). Each of these phases uses a different intelligent algorithm with some enhancements. The performance of the proposed system was analyzed and evaluations were elaborated with well-known existing routing protocols. To assess the proficiency of the proposed system and evaluate the endurance of the network, efficiency parameters such as network lifetime, energy consumption, and packet delivery to the Sink (Base station) were exploited. The experimental outcomes justify that the proposed system surpasses the existing mechanisms by 50%.

Keywords: wsn, routing, ai moea, aco ga

استخدام خوارزميات ذكية لتحسين بروتوكول التوجيه الشبكي

رويدة محمد ياس*، سكينه حسن هاشم

¹الهيئة العراقية للحاسبات والمعلوماتية للدراسات العليا

²قسم علوم الحاسبات، الجامعة التكنولوجية، بغداد، لعراق

الخلاصة:

إن التطور السريع لتقنيات الشبكات اللاسلكية يفتح الباب أمام تطور شبكات الاستشعار اللاسلكية وتطبيقاتها في مختلف المجالات. وتتكون الشبكة من عقدات صغيرة لاستشعار الطاقة تستخدم في بيئة صعبة. وتعتبر الطاقة اللازمة للاتصال بين شبكات الاستشعار من أهم التحديات الرئيسية التي يجب معالجتها لتفادي فقدان الحزم أو فقدانها على نطاق واسع، وسرعة استنفاد الطاقة، مما يؤدي إلى انخفاض كفاءة الشبكة وزيادة التأخيرات في تسليم الحزم. ولهذا الغرض، كان من المهم جداً مراقبة الطاقة المستخدمة في المتحسسات من أجل تحسين كفاءة الشبكة العامة عن طريق استخدام أساليب ذكية للحد من الطاقة المستخدمة فبذلك يتم تمديد

*Email: roueidam@yahoo.com

عمر الشبكة واختيار مسارات توجيه ناجحة للأرسال البيانات للمحطة. ولهذه الأسباب ، فقد تم تصميم نظام فعال يستخدم الطرق الذكية لإطالة عمر الشبكة. وهذا النظام يقسم إلى أربع مراحل (مرحلة نشر أجهزة الاستشعار ، ومرحلة التجميع وتكوين العقد، ومرحلة المجموعات الداخلية ، ومرحلة المجموعات المشتركة). كل مرحلة من هذه المراحل تستخدم خوارزمية ذكية مختلفة مع بعض التحسينات. ويجري تحليل أداء النظام المقترح ، ويجري إعداد التقييمات مع بروتوكولات تحديد المسار المعروفة جيدا. ولتقييم كفاءة النظام المقترح لتقييم مدى تحمل الشبكة ، يتم استغلال بارامترات الكفاءة مثل عمر الشبكة ، واستهلاك الطاقة ، وتوصيل الحزم إلى محطة سينك (القاعدة). وتبرر النتائج التجريبية أن النظام المقترح يتجاوز الآليات القائمة بنسبة %50.

1. Introduction

Wireless Sensor Network (WSN) has been well-known over the years for their environmental applications, including healthcare, disaster warning, agriculture, monitoring, etc.[1]. A large number of wireless sensor nodes must be deployed to perform such tasks in such applications in a tough environment [2]. These nodes are small, sensing-limited, and transmission-limited. They are also limited in energy resource and memory and cannot be substituted. For these reasons, the key problem is how to optimize the life of the WSN and reduce energy consumption [3].

Many energy-efficient approaches have been suggested to solve the WSN issues, including energy-efficient MAC layer protocol, clustering, routing, scheduling, etc. [4, 6]. In a cluster-based protocol, the sensor nodes are organized into clusters, with a predetermined number of clusters, where only the Cluster Head (CH) is responsible for communicating with the sink / base station. It is one of the effective methods used to increase energy efficiency by minimizing and balancing the energy consumption of the sensor nodes [7]. In the sensor node scheduling, a few of the total nodes deployed are enabled for a few rounds to monitor the target area, while the remaining nodes are in sleep mode; this method preserves the node energy and prolongs the lifetime of the network [8]. As mentioned above, the nodes are of restricted range and can communicate only if they are within the same range of transmission. For these reasons, there are also problems of how the entire coverage of the target area can be guaranteed and of the connections between the nodes and the sink (base station) [8].

Another issue of the cluster-based protocol is that the cluster head consumes more battery power than the rest of the sensor nodes. The explanation for this unequal energy consumption is that the head of the cluster has to remain active most of the time, while the remaining nodes are only active when they process and transmit the information. This is why, relative to other nodes, the Cluster Head uses up the energy sources faster and thereby decreases the network's lifespan. The balances of the energy consumption may be restored by periodic (i.e. rotation) transmission of the position of the cluster head to different sensor nodes [9, 10], or by using the re-clustering method.

The present research has proposed an efficient system to extend the life-time of the abovementioned WSNs, through the combination of different enhanced intelligent methods.

The remaining parts of the present research will be organized as follows. Section 2 presents the related work. Section 3 shows the proposed system. Section 4 provides the results. Finally, Section 5 will conclude the paper and describe the future work.

2. Related work

In WSN, there are many approaches that have been suggested for energy efficient clustering to maximize the network life-time. LEACH (Low Energy Adaptive Clustering Hierarchy) was the first approach for clustering suggested by Heinzelman *et al.* [11]. In this method, the selection of CH is based on numbers that are generated randomly between 0 to 1. The numbers are selected and compared with the threshold (T), which is predetermined. If the random number chosen is $< (T)$, then the node becomes a CH. This is the weakness point of this protocol; i.e. the single hop (one-hop) communication between CHs and the sink (Base

station). An enhanced LEACH approach, called Low Energy adaptive clustering hierarchy-Centralized (LEACH-C) was described [15]. The selection of CHs in this approach depends on parameters that will be added, like nodes' location and the remaining energy. These parameters lead to better selection of CHs, where the parameters are sent to the sink, then the sink will find the average node energy, the candidate nodes as CH if they have energy higher than the average energy. The main disadvantage is that the CH is selected randomly from candidate CHs. Thus, this approach do not always guarantee that the higher residual energy become a CHs. In a previous work [12], another improved version of LEACH was proposed by Raghavendra & Lindsey, which was called PEGASIS] (i.e. Power-Efficient Gathering in Sensor Information System). In this protocol of routing, better results can be achieved by balancing energy and prolonging the network lifetime. Problems of this approach include the generation of long chain among the neighboring nodes, due to chain is depending upon the closest neighbors. Also, early death of some nodes can occur during the mechanism of CH rotation, particularly the ones which are far from the sink, as a result of long transmission distance. Communication overhead will be maximizing cause the resection of the chain-head after every round. There have been many cluster-based routing algorithms developed for WSN, depending on evolutionary algorithms. Chakraborty *et al.* [13] proposed a protocol of routing GROUP that forms a chain to communicate with the sink, where each node is allowed for a non-periodical message transmission with the sink. Transmission is based on the remaining energy and the location of every node. Hence, the network life-time will be increased and the greedy chain will be avoided. In another study [14], Jong-Myoung *et al.* enhanced the LEACH protocol by introducing a new CHEF-Cluster Head Election mechanism by using FL. The researchers in this algorithm proposed that with the use of FL, the collection and the calculation time of the overheads may be decreased, and ultimately, the Sensor Networks' life-time may be increased. Ran *et al.* [16] improved the LEACH protocol with the use of the Fuzzy Logic (LEACH-FL) that takes distance, battery level, and node density into account for Cluster Head election, but they did not solve issue of the fast energy depletion in the cluster head. Ataul *et al.* developed a new WSN routing protocol which is based on GA for the data transmission between the gateways in the two tiers WSN [17]. This approach used Roulette Wheel for the individual selection, and the fitness function was based on the network life-time in terms of each round. A new routing approach, called GAR, which is based on GA was proposed [18] for improving the life time if the network. This algorithm used the overall distance in each round for defining the fitness function. Gupta and Jana [19] developed a new clustering and routing approach which is based on GA. The clustering phase depends on two factors, namely the remaining energy of the gateways and the distance between the sensor nodes and their cluster head. While, the routing phase depends on the remaining energy of the gateways. It also makes a tradeoff between the forwards' number and the distance of the transmission.

However, the above algorithms neither take into consideration the rotation of the cluster head within each clusters, nor the scheduling of the sensor nodes in the region of interest with cluster formation. Hence, the major contributions in the present study are as follows:

- The utilization of Voronoi Diagram for sensor deployment in the Deployment Phase.
- The utilization of a FCM, GA, FGA algorithm to form balanced clusters in the Clustering Phase.
- In the Intra-Cluster Phase, MOGA-Scheduling is used for sensor nodes scheduling. Also, FCHR is employed for solving the CHs' energy depletion, where the FL is used for making decisions for the determination of the new CH (N-CH). FL will use three parameters, namely the residual energy of the cluster member (CM), the distance between CM and the sink, and the average distance.

- The utilization of MOEA/D, MOACO, and Dijkstra for efficient route selection from each CH to BS in the Inter-Cluster Phase.

3. The Proposed System

Essentially, the proposed system has nine proposed routing protocols. Each of these protocols is implemented in two versions (With Scheduling and Without Scheduling). The following points illustrate the names of the proposed routing protocols and the used algorithms [20]:

1. FCMC-FCHR-MOEA/D-RP

This protocol uses Fuzzy C Mean for clustering the sensor nodes, Fuzzy Cluster Head Rotation for selecting another CH when depleting the energy of the first CH, and MOEA/D for path selection.

2. FCMC-FCHR-MOACO-RP

This protocol uses Fuzzy C Mean for clustering the sensor nodes, Fuzzy Cluster Head Rotation for selecting another Cluster Head (CH) when depleting the energy of the first CH, and MOACO for path selection.

3. FCMC-FCHR-Dijkstra-RP

This protocol uses Fuzzy C Mean for clustering the sensor nodes, Fuzzy Cluster Head Rotation for selecting another Cluster Head (CH) when depleting the energy of the first CH, and Dijkstra for path selection.

4. GAC-FCHR-MOEA/D-RP

This protocol uses Genetic Algorithm to cluster the sensor nodes, Fuzzy Cluster Head Rotation to select another Cluster Head (CH) when depleting the energy of the first CH, and MOEA/D to select the route.

5. GAC-FCHR-MOACO-RP

This protocol uses Genetic Algorithm to cluster the sensor nodes, Fuzzy Cluster Head Rotation to select another Cluster Head (CH) when depleting the energy of the first CH, and MOACO to select the route.

6. GAC-FCHR-Dijkstra-RP

This protocol uses Genetic Algorithm to cluster the sensor nodes, Fuzzy Cluster Head Rotation to select another Cluster Head (CH) when depleting the energy of the first CH, and Dijkstra to select the route.

7. FGAC-FCHR-MOEA/D-RP

This protocol uses Fuzzy Logic with Genetic Algorithm for sensor nodes' clustering, Fuzzy Cluster Head Rotation to select another Cluster Head (CH) when depleting the energy of the first CH, and MOEA/D to select the route.

8. FGAC-FCHR-MOACO-RP

This protocol uses Fuzzy Logic with Genetic Algorithm for sensor nodes' clustering, Fuzzy Cluster Head Rotation to select another Cluster Head (CH) when depleting the energy of the first CH, and MOACO to select the route.

9. FGAC-FCHR-Dijkstra-RP

This protocol uses Fuzzy Logic with Genetic Algorithm for sensor nodes' clustering, Fuzzy Cluster Head Rotation to select another Cluster Head (CH) when depleting the energy of the first CH, and Dijkstra to select the route.

4. The System Phases

The system is divided into Four Phases; each phase is consisted of different intelligent approaches. Before the explanation of all phases, the number of the sensor nodes, the area of the region, and the location of the sink must be determined.

4.1 Deployment Phase

This phase includes the ways that the sensor nodes are distributed in the area of interest. In this work, the sensors are deployed randomly, where the sensor nodes are distributed normally based on the Pseudo Random Number Generator (PRNG). Then, the Voronoi

Diagram (VD) was applied. It is a geometrical method where the area R is partitioned into polygons, each containing one sensor node. The polygons are generated from a point generator, so that each sensor node inside the polygon will be closest to this generator than to the other nodes. To generate VD, a number of nodes $X_i, i=1, 2, \dots, N$ are generated, where N is the number of sensor nodes in the network. Also, a point Z is generated for the polygons, where $Z \in R$ is the area. The Voronoi region V_i that corresponds to each node X_i is defined as:

$$V_i = \{Z \in R \mid \|Z - X_i\| < \|Z - X_j\|, \forall i, j = 1, 2, \dots, N, i \neq j\} \quad (1)$$

After the deployment of the sensor node using one of the above methods, the sink sends a start message (*Start-Msg*) for all nodes in the network as a sign for the beginning of the network operation. Then, every node computes its distance to the sink, depending on the Received-Signal-Strength Indicator (RSSI) technique. Every node calculates the distance with its neighboring nodes and gathers their energy by sending a request information message (*Reqst-Info-Mesg* (ID, Location, Energy)) to every neighboring node for gathering information about the location and the energy of each neighboring node. The nodes within the sensing range receive this message and then reply by sending a self-information message (*Self-Info-Msg* (ID, Location, Energy)) [20].

4.2 Clustering phase

This stage of the network's operation is dedicated to divide the network into clusters and select a CH to collect the sensed data from the CMs. For achieving this stage, an intelligent clustering technique is used for minimizing the energy consumption of the network and for balancing the distribution of the workload of CH. In this phase, the number of the needed clusters is determined and the clusters are formed. Three clustering techniques are included, which are described below.

4.2.1 Fuzzy C Mean (FCM) Based Clustering

In this method, all the sensor nodes will participate in cluster formation. At the beginning, the centers (c) of clusters are selected randomly by the sink from the sensor nodes deployed in the area of interest. Next, the following steps are conducted [21]:

1. Compute the distance between each sensor node in the area and all selected cluster centers, using the Euclidian distance:

$$d(x_i, c_j) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

where:

- (x_2, y_2) are the coordinates of the sensor nodes.
- (x_1, y_1) are the coordinates of the cluster center.
- i is the number of the sensor nodes.
- j is the number of the clusters center.

2. Calculate the membership degree for each sensor node to each cluster center based on equation (3), where each node is assigned to each cluster center. When a node has a highest membership degree to one of the cluster centers, this means that it belong to this cluster.

$$\mu_{ij} = \frac{1}{\sum_{r=1}^M \left(\frac{d(x_i, c_j)}{d(x_i, c_r)} \right)^{2/(m-1)}} \quad (3)$$

where:

- $x_i = [x_{i1}, x_{i2}]$ is the coordinate of the node i .
- $c_j = [c_{j1}, c_{j2}]$ is the center coordinating of the cluster j .
- m is the exponential weight used for adjusting the degree of fuzziness, which has a value greater than 1.
- $r = 1, 2, \dots, M$, M is the number of clusters.
- μ_{ij} is the degree of membership of node x_i in the j th cluster.

3. Then, the new cluster centers for each cluster are calculated by using the following equation:

$$c_j = \frac{\sum_{i=1}^N (\mu_{ij}^m \times x_i)}{\sum_{i=1}^N \mu_{ij}^m} \quad (4)$$

4. Update the membership degree for each node belonging to each cluster center according to equation (3).

5. Continue steps 2, 3 to 4 until the centroids are not changing, as indicated in algorithm 1.

Algorithm (1). The steps for the FCM cluster formation.

Algorithm (1): FCM cluster formation
Input: - N = No. of nodes - M = No. of clusters - m = weighting exponent Output: - V = fuzzy c cluster matrix - C = centroid matrix
Begin Step 1: Initialize clusters' center vector $C = c_1, c_2, \dots, c_M$ Step 2: Repeat - compute the distance between all sensor nodes and all clusters center - Initialize membership matrix μ using Eq. (3) - Calculate new cluster centers using Eq. (4) Until the centroids are not changing Step 3: Return V, C End

4.2.2 GA-based clustering

In this method, the cluster centers are selected randomly from the deployed sensor nodes, then the cluster formation is performed using the following steps [22].

1. The chromosome representation

Each chromosome is represented as an integer number, which represents the center of the cluster. The length of the chromosome is the number of the selected clusters. Each gene in the chromosome represents the X, Y dimensions of the cluster center that is stored in the center dimension's matrix.

Example: Let $N = 2$, i.e. the two dimensional space, and $K = 4$, i.e. the number of the clusters is four, so that each gene contains the value of X, Y dimensions of the cluster center Z.

$$CH = \begin{bmatrix} Z_1 & Z_2 & Z_3 & Z_4 \end{bmatrix}$$

where Z represents the dimensions of cluster center.

$$CH = \begin{bmatrix} 50 & 60 & 31 & 22 & 15 & 29 & 32 & 40 \end{bmatrix}$$

where (50 60) is the first cluster's center, where 50 represents the value of x-axis, while 60 represents the value of y-axis. Also, (31 22) is the second cluster's center, (15 29) is the third cluster's center, and (32 40) is the fourth cluster's center.

2. Initialization of the population

Each chromosome is initialized randomly and the K cluster centers (Z_1, Z_2, \dots, Z_K) are also selected randomly. This operation is repeated for the number of population (P).

3. The Cluster formation

The cluster is created according to the cluster centers that were selected randomly in the

chromosome. Each node (X_i) is assigned to one cluster (C_j) in the chromosome, with a center (Z_j), if and only if

$$dist(X_i, Z_j) < dist(X_i, Z_p), (5)$$

$$i = 1, 2, \dots, n, j \& p = 1, 2, \dots, k, \text{ but } j \neq p$$

where:

- $dist$ represents the distance computed using equation (2)
- X_i is the dimension of the node.
- Z_j is the center of the cluster.

After the cluster formation, each cluster center (Z) in the chromosome is replaced by the mean value of nodes in this cluster. Therefore, for cluster C_i , a new cluster center Z_i^* is computed by using the following equation:

$$Z_i^* = \frac{1}{n_i} \sum X_j \tag{6}$$

where:

- Z_i^* is the new cluster center.
- $X_j \in C_i, i = 1, 2, \dots, K.$
- n_i is the number of nodes that belong to the C_i cluster.

4. The Fitness computation

The fitness function f is depending on the static dimensions of the nodes and the variable clusters, computed as follows:

$$(\max)f = 1/m \tag{7}$$

$$m = \sum_{i=0}^K m_i \tag{8}$$

$$m_i = \sum_{X_j \in C_i} \|X_j - C_i\| \tag{9}$$

where m is the clustering matrix.

Hence, maximum fitness function will lead to minimize the clustering matrix.

5. GA-operation

- Selection:

Roulette Wheel selection is used, where the chromosome with the large fitness value will survive to the next generation.

A two point - crossover with a predefined probability (P_c) was used.

- Mutation:

Each chromosome is mutated based on a fixed probability (P_m). The mutation is achieved for the selected gene by generating a random number depending on the number of nodes in the cluster.

6. Termination condition

These operations will terminate if the centroid is not changing or after a maximum number of iterations. The best chromosome in the last generation represents the best locations of cluster centers.

Algorithm (2). The steps for the GA cluster formation.

Algorithm (2): GA cluster formation
Input: N = No. of nodes M = No. of clusters P _c = probability of crossover P _m = probability of mutation nPop = population size MaxIt = maximum iteration
Output: best centroid, set of M clusters
begin
Step 1: Initialize clusters' center position (chooses randomly from the nodes)

```

Step 2: For it = 1 to nPop
  For j = 1 to N
    For k = 1 to M
      1. Compute the distance between each node and each cluster center
      2. Assign each node to a single cluster
      3. Compute new cluster center using Eq. (6)
      4. Evaluate each individual using Eqs. (7)(8)(9) and store the best solution
      5. Select by using Roulette Wheel
      6. Apply crossover with PC
      7. Apply mutation with Pm
      8. Evaluate the new generation and store the new best solution
      9. IF (new best solution == best solution) stop and get the output
    Else continue
  End IF
End For
End For
End For
End

```

4.2.3 FGA-based clustering

In this method, the cluster formation has two steps; in the first step, the sink uses a fuzzy logic system with three inputs (residual energy of the nodes at the current round, distance from the sink, and centrality degree of the node) and one output to select the initial cluster centers from the deployed sensor nodes; while in the second step, the clusters are formed using the GA.

The calculation of the centrality degree is based on the following equation:

$$C_i^{cl} = \frac{1}{n-1} \sum_{j \neq i} \frac{1}{d(i,j)} \quad (10)$$

where n is the no. of the nodes in the network and $d(i,j)$ is the distance between each node and the other nodes, which is computed using equation (2).

First step: Each node in the network will calculate its chance to be an initial CH, where three input variable are considered. Every input function has 3 Gaussian membership functions, namely (Low, Medium, High) for Residual Energy, (Far, Medium, Close) for the Distance to the Sink, and (Far, Medium, Near) for the Centrality degree. Also, 9 triangular membership functions are used for the output function, namely Very Weak (VWeak), Weak, Rather Weak (RWeak), Rather Medium (RMedium), Medium, High Medium (HMedium), Rather Strong (RStrong), Strong, and High Strong (HStrong)). The node with High Residual Energy, Close distance to Sink, and near centrality will have a higher chance to be the initial CH. The membership function for the three inputs is Gaussian, while for the output membership function it is triangular. In the Fuzzy Inference System (FIS), crisp input values are translated into linguistic variables. The fuzzy if-then rules are used to convert the input variables into the corresponding fuzzy variables depending on the Mamdani process. The total number of rules, i.e. 27 (3^3), will be used. The method for defuzzification (i.e. for converting linguistic variables into a crisp output value) will be applied using the center of area (CoA).

Second step: Then, the cluster formation is performed by using GA, as described previously, based on the initial CH center that was selected using step 1.

Algorithm (3) shows the main steps of how the sensor nodes are clustered depending on the FGA method [20].

Algorithm (3): FGA cluster formation
Input: N = No. of nodes M = No. of clusters P_C = probability of crossover P_m = probability of mutation nPop = population size MaxIt = maximum iteration Output: best centroid, set of M clusters
<pre> //First Step// begin For i = 1 to N 1. Compute the nodes Residual Energy 2. Compute the nodes Centrality using Eq. (10) 3. Compute the nodes' Distance to Sink 4. Initial Cluster Center $_{chance} = FIS$ (node[i]. Residual Energy, node[i]. Centrality, node[i]. Distance to Sink) End //Second Step// begin Initialize clusters' center position (chooses randomly from the nodes) For it = 1 to nPop For j = 1 to N For k = 1 to M 1. Compute the distance between each node and each initial cluster center 2. Assign each node to a single cluster 3. Compute new cluster centers using Eq. (6) 4. Evaluate each individual using Eqs. (7)(8)(9) and store the best solution 5. Select by using Roulette Wheel 6. Apply crossover with P_C 7. Apply mutation with P_m 8. Evaluate the new generation and store the new best solution 9. IF (new best solution == best solution) stop and get the output 10. Else continue End IF End For End For End For End </pre>

After completion of the above steps in the three methods, the nodes are divided into clusters, each containing a different number of nodes. The node that is close to the cluster center with higher energy will be a CH. The CH sends an advertisement message, called Final CH message (*Final-CH-Msg* (ID, Distance, Energy)), which contains its dimensions, energy, and distance information to its cluster member (CM) nodes for informing them. After the CM nodes receive the CH message, they send a Join request message (*Join-Reqst-Msg*) to their CH for informing that they are the members of the cluster. Every CH sends a TDMA schedule table by sending a scheduling message (*Sched-Msg* (TDMA)) to its CMs, and begins to send the sensed data to its CH. Finally, each CH sends a CH advertisement message (*CH-Adv-MSg* (ID, Residual Energy, Distance)) to the sink, then the sink sends this message to all CHs in the NW for using them during the routes selection [20].

4.3 Intra cluster energy optimization phase

In this phase, the energy of each cluster in the network is optimized using two proposed methods, namely the Multi objectives Nodes Scheduling and CH-Rotation. Each method will be explained briefly in sections 4.3.1 and 4.4.2, respectively [20].

4.3.1 Multi Objective GA Nodes Scheduling (MOGA-Scheduling)

After the clusters formation, it is assumed that all sensor nodes have started to operate and send the sensed data to the CHs. But in this process, the node's power will be consumed and this node will quickly die. Therefore, CMs scheduling is suggested to maximize the clusters lifetime. In this mechanism, a multi-objective nodes scheduling based on GA is used so that a subset of these nodes will be activated while the remaining will be inactive. The objective is to minimize the number of nodes that sense the target area while maintaining the full coverage of the regions, selecting the sensors with higher residual energy. When one or more of the selected sensor nodes have lower level of energy, they die quickly and thus the network is uncovered. In this case, a new set of the remaining sensor nodes in each cluster has to be rescheduled in order to complete the task. According to this mechanism, GA- based scheduling can be explained in the following steps [22].

1. Chromosome representation

The chromosome is represented as a binary string, and the chromosome's length (N) represents the number sensor nodes (s) in one cluster in the WSN.

$CH = \{g_i | g_i \in \{0,1\}, \forall i, 1 \leq i \leq N\}$. The value of g_i represents the case whether the selected S_i is active or sleeps in the current round.

where:

$$g_i = \begin{cases} 1, & \text{if } s_i \text{ is active node} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

2. Population initialization

The creation of the population will be achieved by generating the chromosomes in a random form. Here, we must mention that each chromosome in the population represents a CM nodes. The generated number is either 0 or 1.

Calculation of the Proposed Fitness Function

The evaluation of the chromosomes depends on the fitness function. The proposed fitness function will consider the above explained objectives, and it which will be described as follows:

First objective:

Select the chromosome with minimum number of sensor nodes for scheduling. So Hence, the first fitness function will be as follows:

$$\text{MIN } f_1 = \sum_{i=1}^N g_i \quad \{g_i, \{g_i=1, i \leq N\}\} \quad (12)$$

Where where g_i Represent represents the sensor node in the selected chromosome and N is the number of the nodes in each cluster.

Second objective:

Select the sensor nodes g_j that covered the target t_i . So Hence, the second fitness function will be as follows:

$$\text{max} f_2 = \sum_{i=1}^K [\delta_{\text{cost}}(t_i)] \quad (13)$$

$$\delta_{\text{cost}}(t_i) = \begin{cases} +1 & \text{if } |q_{\text{cover}}(t_i)| \geq 1 \\ -1 & \text{otherwise} \end{cases} \quad (14)$$

$$q_{\text{cover}}(t_i) = \{s_j | g_j=1 \ \& \ \text{dist}(t_i, s_j) \leq R_{\text{sens}}\} \quad (15)$$

Where where K is the number of numbers of targets, dist is the Euclidian distance, R_{sens} is the sensor's sensing range.

Third objective:

Select the sensor nodes with the higher energy. So Hence, the third fitness function will be as follows:

$$\max_{i \in N} \{ f_3 = \min \{ E_{res}(s_i) | g_i = 1 \}, i \in N \} \quad (16)$$

Where where E_{res} is the node's residual energy.

Therefore, to obtaining a single function will be achieved by summation summing of the above functions and will be formulated as follows:

$$\text{Fitness (max)} = \{ W_1 \times (1.0 - 1/N \sum_{i=1}^N g_i) + W_2 \times 1/K \sum_{i=1}^K [\delta_{cost}(S_i)] + W_3 \times E_{MIN}/E_{MAX} \} \quad (17) \text{ Wherewhere:}$$

W_i is the weight value, $0 \leq W_i \leq 1, \forall i, 1 \leq i \leq 3$.

E_{MAX} is the maximum energy of the sensor nodes.

E_{MIN} is the current minimum energy, and it can be obtained using f_3 .

GA operators

For the GA operators, the following steps will be conducted:

Selection: - The the chromosomes with the highest fitness values are selected for the new generation. The selection is done by using Roulette Wheel selection.

Crossover: – two two-point crossover is applied for producing the new chromosomes.

Mutation: – in this step, the gene is randomly selected and converted from (0 to 1) or from (1 to 0).

Stopping condition

The algorithm is stopped after a number of iterations.

After the completion of the above steps, the chromosome with the highest fitness function is selected and the number of ones in this chromosome represents the number of the sensors that are selected to be active in the cluster. In each cluster, the active sensor nodes send their sensed data to their CHs using a single hop. This process is repeated for each cluster and for each round. Algorithm (4) describes the main steps for MOGA-Scheduling [21].

Algorithm (4): GA cluster formation

Input: N = No. of nodes in the cluster

M = No. of targets

P_C = probability of crossover

P_m = probability of mutation

$nPop$ = population size

$MaxIt$ = maximum iteration

E_{MAX} = maximum nergy of the nodes

R_{sens} = sensing range of the nodes-

$W_1 = 0.3, W_2 = 0.2, W_3 = 0.3$.

Output: minimum No. of active nodes that cover the targets

Begin:

Step 1: Generate initial population (chooses randomly)

Step 2: Compute the residual energy of each node and the distances between the nodes and the targets

Step 3: For Iter = 1 to $MaxIt$

For $i = 1$ to $nPop$

For $j = 1$ to N

For $k = 1$ to M

IF (the value of gene == 1)

Sum = sum +1

End IF

IF (the value of gene == 1) && (distance between node and target $< R_{sens}$)

Coverage cost = +1

Else Coverage cost = -1

```

End IF
Sum of cost coverage = Sum of cost coverage + 1
IF (the value of gene == 1)
  Select the minimum residual energy
End IF
Compute the proposed fitness function using Eq. (17)
Select using Roulette Wheel
Apply crossover with  $P_C$ 
Apply mutation with  $P_m$ 
Evaluate new generation and store the new best solution
IF (Iter == MaxIt) break and get the output
Else continue
End IF
End For
End For
End For
End For
End

```

4.3.2 Fuzzy based Cluster Head Rotation (FCHR)

After several rounds, the CHs will deplete their energy faster than the Cluster members (CMs), and this which leads to reduce the Network network lifetime. If the CH's energy is fully consumed, so this leads to making create some area that cannot be covered, and the Network cannot complete its mission. So to To solve this problem, a CH-Rrotation is used. In the proposed method, each CH will do the following [21]:

Checking: Tthe CHs check their energy at each round, . if If the energy of the CH becomes equal to threshold, then CH role is reassigning (rotating) to different sensor nodes (CMs). Then, each CHs sends a request message (Req-Msg) to each of theirits own CMs.

Collection: the CMs replay their current CH (C-CH) by computing their Residual residual energy (E_{Res}), Distance the distance between the each CM to and the sink, / (BS) (d_{BS}), and the distance from their cluster center (d_{CC}) CMs, which is computed using equation (3.2). Then, the CMs send these this information to their corresponding C-CH.

New CH selection: The the proposed method uses the FL for determining the New-CH (N-CH). Mamdani is used in FL algorithm due to its simplicity. The common Mamdani's rule with two inputs and one output is shown as follows:

if $in(1)=A$ and $in(2)=B$ then $out=C$ (18)

Where where A, B, and C are the membership functions of the inputs and output.

For the Fuzzification stage, three attributes of the CM will be considered, t. These three attributes will be the input function for the Fuzzy system. Every input function has 3 Gaussian membership functions; these are the (Low, Medium, High) for Residual Energy, (Far, Medium, Close) for the Distance to the Sink, and (Far, Medium, Near) for the Distance to cluster center, . Also, and use 9 triangular membership functions are used for the output function; these are the (Very Weak (VWeak), Weak, Rather Weak (RWeak), Rather Medium (RMedium), Medium, High Medium (HMedium), Rather Strong (RStrong), Strong, and High Strong (HStrong)). After the Fuzzification process and getting the membership values, the predetermined fuzzy-rules, for which their total number is 27 (33), are evaluated via the fuzzy inference engine. The rules will be in the if-then form and the Fuzzy operator (and) operator is used for identifying new fuzzy output sets.

Finally, the aggregation process is applied on the fuzzy outputs, and converts them to a single crisp output, for. For this process, the fuzzy operator (OR) can be used. For determining the

chance of the CM to be the N-CH, the Centroid of Area (COA) Defuzzification technique can be used as follows:

$$C_CM(i) = \frac{\int_{x_j} x_j * \mu(x_j) dx}{\int \mu(x_j) dx} \quad (19)$$

Where where x_j is the element that has maximum membership value, and j is the fuzzy rule number.

After selecting N-CH, the C-CH moves its current state to the N-CH and turns its rule to become a CM. The N-CH informs their its CMs to joint it by sending the New CH-message (NEWCH-Msg). Then, each CMs starts to send their its sensed data to the N-CH. The N-CH Sends sends the rotation transference message (Rot-Trans-Msg) to the BS and to all other CHs for informing them about the N-CH. The CHs receive this message to take into consideration the N-CHs during routes selection [20]. Algorithm (5) describes the steps of the FCHR [21].

Algorithm (5): FCH-Rotation
Input: N = The No. of nodes in the cluster (CMs).
Output: New cluster head (N-CH)
Begin Step 1: checking the Residual Energy (E_{Res}) of the Current Cluster Head (C-CH) IF (E_{Res} of the C-CH == 0) For i = 1 to N Compute The Distance between each CM and the BS (d_{BS}) Compute The Distance between each CM and the cluster center (d_{CC}). Compute The Residual Energy of each CM (E_{Res}). Compute CM.Chance = FIS (CM. E_{Res} , CM. d_{BS} , CM. d_{CC}) End For Step 2: finding the N-CH N-CH = Max CM.Chance N-CH sends (NEWCH-Msg) to CMs Else C-CH continues to receive CM data End IF End

The procedure of rotating the CH in each cluster will continue until no other CMs are able to become N-CH (i.e. every CM is selected as CH) [20].

Inter Inter-cluster energy optimization phase

This phase is concerned with the data routing scheme, . after After the CMs send their sensed data to their corresponding CHs, each CH needs to forward its collected and aggregated data to the BS. Each CH can transmit the data using single-hop or multi-hops, the choice is depending on the distance between the CHs and the BS. If the BS is within the CHs' communication range, the CHs send their aggregated data directly to the BS (single-hop), otherwise each CH must select an efficient relay CHs to construct the route towards the BS. The routes are selected depending on minimum energy and distance (cost) routing. At the beginning, each CH sends messages to its neighboring CHs, called (NEIGHBOURS_MESSAGE), which is contains the id of the CH, residual energy, and CH's distance to BS. Based on the strength of the received signal, each CH calculates its the distance between itself and the sender of the message. Each CH receiveds this message from other CHs within the radius, then useds the received information for the selecting the optimal route. For Selecting the selection of optimal routing scheme, two three methods are proposed, which are the weight weight-based ACO Routing (WACO-Routing), Multi-objectives GA Routing (MOGA-Routing), and Dijkstra based Routing. These three techniques will be explained briefly in sections (4.4.1), (4.4.2), and (4.4.3) [20].

Multi Objectives ACO-Routing (MOACO-Routing)

In this method, the ants are used as a model to find the optimal route between the source and the destinations. In WSN, there is a single destination node (BS), and it uses two ant's types, namely the forward and backward ants. The forward ants are responsible for new routes discovery, and information gathering, while the backward ants return back from the destination to the source for updating the sensor node information when moving. They do that by releasing pheromones for routes establishment. Pheromone values decide how ants that originate in a source node will move along the multi-hops until they reach the BS. Every forward ants established their its route by selecting a next-hop CH from their its neighbors, which are placed in the same communication radius of the CH. In the proposed MOACO-Routing, each ant can travel from CH_i to CH_j based on the proposed equation (20) [20]:

$$P_{((i,j))^m} = (\tau_{(i,j)}(t))^{\alpha} \times [W_{(i,j)}(t)] / (\sum_{CH \in N_i} (\tau_{(i,j)}(t))^{\alpha} \times [W_{(i,j)}(t)]) \quad (20)$$

Wherewhere:

$P_{((i,j))^m}$ is the probability of the ant m to travel from CH_i to CH_j .

$\tau_{(i,j)}(t)$ is the value of the pheromone trail deposited for transition from CH_i to CH_j at iteration t .

α, β are control constants are theirwith values between (0,1).

N_i is the set of the CH_i neighbors, which the ant m can be selected from the next-hop CH.

$W_{(i,j)}(t)$ is the weight information of searching for the route segment CH_i, CH_j , and can be computed using equation (21):

$$W_{ij}(t) = [E_{(i,j)}]^{-\beta} \times [E_{(i,j)}]^{-\delta} \times [D_{(i,j)}]^{-\theta} \times [D_{(j,BS)}]^{-\sigma} \quad (21)$$

Wherewhere:

$E_{(i,j)}$ is ensure to select the CH with higher residual energy, and can be found using equation (22):

$$E_{(i,j)} = (e_j(t)) / (\sum_{l \in N_i} (e_l(t))) \quad (22)$$

Where where $e_j(t)$ is residual energy of CH_j , and $e_l(t)$ is the residual energy of the l th CH neighbors.

$D_{(i,j)}$ Eensure selecting the shortest route segment between CH_i and CH_j , and it is computed using equation (23).

$$D_{(i,j)} = d_{(i,j)} / (\sum_{s \in N_i} d_{(i,s)}) \quad (23)$$

Where where $d_{(i,j)}$ is the distance between CH_i and CH_j , and $d_{(i,s)}$ is the distance of the l th CH neighbors.

$D_{(j,BS)}$ Eensure selecting the shortest route segment between CH_j and BS, and it is computed using equation (24).

$$D_{(j,BS)} = d_{(j,BS)} / (\sum_{t \in N_j} d_{(t,BS)}) \quad (24)$$

Where where $d_{(j,BS)}$ is the distance between CH_j and the BS, and $d_{(t,BS)}$ is the distance of the l th CH neighbors [CH_j] to the BS.

$\beta, \delta, \theta, \sigma$ are constants and their values are between (0,1).

When the forward ant finds the best next-hop CH, it will establish the route between the sending CH and the next-hop CH. The next-hop CH then produces a backward ant that is sent to the reverse node (source node) and releases the pheromone as it returns. At the end of the searching time the pheromone level will be updated using equation (25).

$$\tau_{(i,j)}(t+1) = (1-\rho) \times \tau_{(i,j)}(t) + (\Delta\tau_{(i,j)}(t)) / (\omega \times [hop]_{(count)}_{-m}) \quad (25)$$

Wherewhere:

$\tau_{(i,j)}(t+1)$ is the value of the pheromone trail at iteration $t+1$.

ρ is the evaporation factor of the pheromone.

ω is a constant.

$[\text{hop}]_{-}([\text{count}]_{-}m)$ is the hops number for the m th ant visited from source CH to the BS.

$[\Delta\tau]_{-}(i,j)(t)$ is the amount of pheromone deposited by m th ant, and it is computed using equation (26)

$$[\Delta\tau]_{-}(i,j)(t) = b \times ([\text{hop}]_{-}\text{max} - [\text{hop}]_{-}([\text{count}]_{-}m)) \times e_{-}([\text{ave}]_{-}m) \quad (26)$$

Wherewhere:

b is a constant (0,1)

$[\text{hop}]_{-}\text{max}$ is the maximum hops number that is allowed for ants.

$e_{-}([\text{ave}]_{-}m)$ is the average energy of the nodes visited by the ant m .

After the arriving arrival of every ant to the BS node, every ant matches its route to the BS. So Hence, for selecting the optimal route a fitness function, equation (27) will be used:

$$\text{Max } f_m^k = (e_{-}([\text{ave}]_{-}m) \times [e_{-}\text{min}]_{-}m) / (L_m^k) \quad (27)$$

Wherewhere:

$[e_{-}\text{min}]_{-}m$ is the minimum energy of node that the m th ant is passing through during its route.

L_m is the tour length of the m th ant route of k iteration.

The higher the fitness value, the better route for the MOACO-Routing, as shown in algorithm (6) [20].

Algorithm (6): MOACO-Routing

Input:

- The Residual Energy of each CHs.
- The Distances between each CHs.
- The Distance between the CHs and the BS.
- N: The No. of CHs.
- M: the No. of ants.
- iter.: the No. of iteration.
- $\alpha, \beta, \delta, \theta, \sigma$ = the parameters

Output: L_{best} which is representing the best route.

Begin

Step 1: initialize the default pheromone, and place the ants in the source node.

Step 2: For $i = 1$ to iter

For $j = 1$ to M

CCH (Current CH) = source CH

while Cnode != destination node

Calculate the weight information using Eq. (21)

Calculate $P_{(i,j)}^m$ probability function for selecting the next CH of the CCH according to Eq. (20)

Choose the next-hop CH

CCH= next-hop CH

End while

End for

Modify the path density of pheromone according to Eq. (26)

End for

Calculate the best route according to Eq. (27)

End

4.3.3 Multi-objectives Evolutionary Algorithm based on Decomposition Routing (MOEA/D-Routing)

In this method, the MOEA/D is used to find the optimal route between the CH and the sink, as shown in algorithm (7). The optimal route will be found by finding the best relay CH towards the sink. The best path will be calculated based on two conflicting goals, which are selecting the maximum residual energy path and minimum distance path, by using the proposed fitness function (28). Because of these conflicting objectives, a single route that optimizes all targets is extremely difficult to find at the same time. Therefore, the MOEA/D will be used to provide a set of paths that is not dominated by any of the others, called the Pareto - optimal set. MOEA with the decomposition of Tchebycheff approach is used for decomposing the Pareto – Front into sub-problems of scalar optimization. In this method, the chromosomes are represented as integer numbers which represent the dimensions of the CH in the NW [20].

$$\text{Maximize MOEA}=[F_RE (I),F_dist (I)] \tag{28}$$

where:

$$F_RE (I)=\sum_{j=2}^{m-1} [RE(CH(j))] \tag{29}$$

$$F_dist (I)=\sum_{i=1}^{m-1} d_((CH(i),CH(i+1))) \tag{30}$$

$F_RE (I)$ is the fitness function that computes the total residual energy of the CHs in the I^{th} chromosome.

$F_dist (I)$ is the fitness function that computes the distance between the CHs in the I^{th} chromosome.

m is the chromosome’s length.

$d_((CH(i),CH(i+1)))$ is the distance between the CHs, which is computed using Euclidean distance.

As explained above, the MOEA/D is used to find the optimal path. Hence, let N indicates the number of sub-problem ($N=2$), $\{\lambda^1, \lambda^2\}$ indicates weight vectors of the two sub-problem, $Z^*=\{Z^1, Z^2\}$ represents the reference point which is the best value that can be found for the two objective functions, T is the number of the neighborhoods of the weight vectors, and (EP) is the external population used for storing the non-dominating solution, . Each chromosome is represented as a variable length vector. The Euclidean distances should be calculated between two weight vectors and then the T vectors closest to each weight vector should be stored in $B[i]$ as an index. , where T is the number of neighboring weight vectors for each weight vector stored in $B[i]$. Each individual will be evaluated using the proposed fitness function (28). At each generation, two parents are randomly selected from $B[i]$. After that, genetic operators (crossover, mutation), are used to generate a new solution (new individual). Subsequently, the new individual is evaluated by applying the proposed fitness function (28). Then, the reference point Z^* for each objective function has to be updated. Lastly, the neighboring solutions and EP are updated to identify the solutions that are non-dominated [20].

Algorithm (7): MOEA/D-Routing	
Input:	
-	The Residual Energy of each CHs.
-	The Distances between CHs.
-	The Distance between the CHs and the BS.
-	MOP: $Maximize MOEA = [F_{RE}(I), F_{dist}(I)]$
-	N : the No. of sub problem.
-	T : the No. of neighboring solutions.
-	P_C = probability of crossover
-	P_m = probability of mutation

- gen_{MAX} = maximum No. of generation.
 Output: External Population (EP) which represents the best route.

Begin:

Step 1: Setup:

1. EP= \emptyset .
2. $gen_{MAX}=0$.

Step 3: Initialization:

1. initialize the reference point $Z = \{Z_1 = 0, Z_2 = 500\}$
2. Generate initial population $P_0 = \{X_1, \dots, X_N\}$
3. Find the weight $\lambda^1 = \frac{i}{N}, \lambda^2 = \frac{N-i}{N}$.
4. Compute the Euclidian distances between two weight vectors, then the T closest weight vectors to each weight vector, store in $B(i), \forall i = 1, \dots, N$, set $B(i) = \{i, \dots, i_T\}$, where $\lambda^{i1}, \dots, \lambda^{iT}$ are the T closest weight vector to λ^i .

Step 4: Update: For $i = 1, \dots, N$

1. Apply proposed FF and evaluate P_0 by using equation (3.28). Select two indexes V, Q from $B(i)$, then generate a new solution (y) by applying (crossover, mutation). Apply the proposed FF and evaluate the new solution (y) by using equation (28).
2. Update of Z: if $Z_i \leq MOEA(y)$, then $Z_i = MOEA(y)$.
3. Update of Neighboring: if $(g^{tc}(y|\lambda^j, Z) \leq g^{tc}(X^j|\lambda^j, Z))$ then $X^j = y, MOEA(x) = MOEA(y)$
4. Update of external population(EP):
 - a. Delete from EP all the vectors dominated by $MOEA(y)$.
 - b. Add $MOEA(y)$ into EP if no vector in EP is dominated.

Step 5: Condition to stop
 If (New generation == Old generation) then stop,
 Output external population (EP). Else, go to step 2.
 End If
 END

4.3.4 Dijkstra Algorithm-based Routing

In this method, the optimal route between the CH and the sink is found based on Dijkstra algorithm, as shown in algorithm (8). The optimal route will be found by selecting the best relay CH towards the sink. This algorithm is based on a graph to select the shortest path from one CH (source) to the sink (destination). Consider that all CHs and the sink are connected to form a fully connected graph $G(V, A, C)$, where V is the vertices' set (CHs set), A is the Arcs' set, and C is the cost of the arcs which represent the distance between vertices (CHs). V is divided into visited CH set and unvisited CH set. First of all, the unvisited node set contains all the CHs except the source CH. The shortest paths between two CHs are calculated by Dijkstra algorithm as follows [20]:

1. Create a set S which contains the CHs that will participate in the route. Firstly, this set contains only the source CH (CH_{source}), and a path to the neighboring CHs (CH_N) that represent the link cost (distance between them).

$$S = \{CH_{source}\}$$

$$C(CH_{source}, CH_N) = \text{dist}(CH_{source}, CH_N)$$

$$\text{Set } CH_{source} = 0, \text{ and } CH_N = \infty.$$

2. For the respective source CH (CH_{source}), accept and calculate all its neighbors' costs ($CH_N(\text{cost})$). Compare the new cost with the value assigned and select the smallest cost.

3. Delete this (CH_N) from the unvisited CHs, then add it to the visited CHs (S).

4. Repeat steps 2 and 3 until reaching the destination (Sink).

Algorithm (8): Dijkstra based Routing

<p>Input:</p> <ul style="list-style-type: none"> - Source CH - N = No. of CHs in the Network <p>Output:</p> <ul style="list-style-type: none"> - S = shortest route
<p>Begin</p> <p>Step 1: initialization</p> <p style="padding-left: 20px;">S = {source CH}</p> <p>Q = list contain the unvisited CHs</p> <p>Distance of the Source CH = 0</p> <p>Step 2: For i = 1 to N</p> <p style="padding-left: 20px;">dist [i] = infinity</p> <p style="padding-left: 40px;">End For</p> <p>Step 3: creating the Graph</p> <p>1. For i = 1 to N</p> <p style="padding-left: 20px;">For j = 1 to N</p> <p style="padding-left: 40px;">Compute the distance between the (CH(i), CH(j))</p> <p style="padding-left: 20px;">End For</p> <p style="padding-left: 40px;">End For</p> <p>Step 4:</p> <p>1. While Q ≠ ∅</p> <p style="padding-left: 20px;">u = CH in Q with smallest dist []</p> <p style="padding-left: 20px;">Remove u from Q</p> <p style="padding-left: 20px;">For v = 1 to N</p> <p style="padding-left: 40px;">If (dist [v] > (cost (u, v) + dist [u]))</p> <p style="padding-left: 60px;">Dist [v] = dist [u] + cost [u, v]</p> <p style="padding-left: 40px;">S = u</p> <p style="padding-left: 20px;">End If</p> <p style="padding-left: 40px;">End For</p> <p>End While</p> <p>Step 5: return S</p>

5. Simulation results and analysis

The suggested algorithm was tested with the use of MATLAB R2019b with the following parameters, shown in table_2.

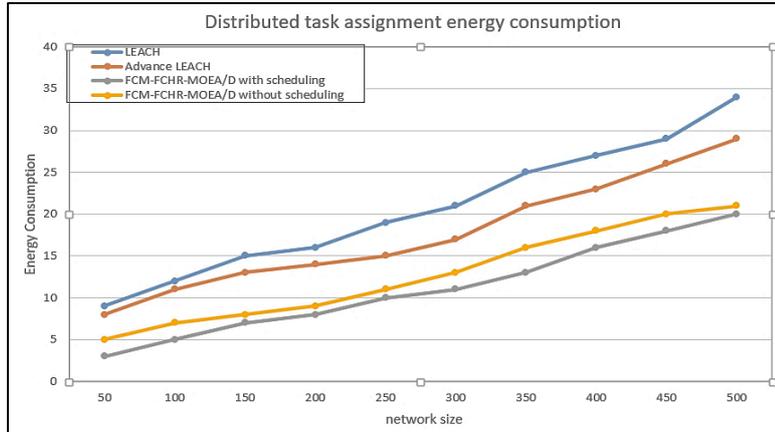
[Table2: Parameters of simulation](#)

~~Table2: parameters of simulation~~

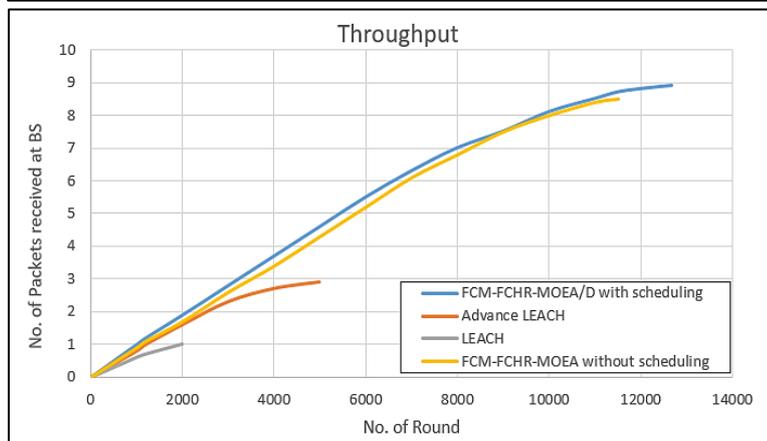
Parameters	Value
Initial sensor nodes' energy	0.5 J
Location of the Sink	(50,100)
Maximum communication range	100
$E_{elec}, \epsilon_{fs}, \epsilon_{mp}$	50nJ/bit, 10pJ/bit/m ² , 0.0013pJ/bit/m ⁴
d_0 (reference distance)	square root (E_{fs} / E_{amp}) = 87 m
Data Packet	4000 bit

The efficiency of the suggested system was assessed based on the network's life-time, the network's throughput, and network's energy consumption. The results of the suggested algorithms, using different scenarios, were compared to the LEACH, Advance-LEACH. The simulation results of the FCMC-FCHR-MOEA/D-RP, FCMC-FCHR-MOACO-RP, and FCMC-FCHR-Dijkstra-RP considered a network size of 100 × 100, nodes' number of 100,

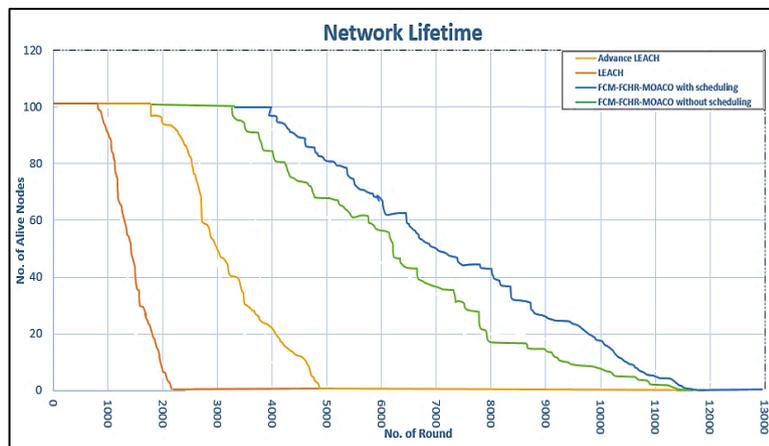
and clusters' number of 5, and are presented in Figures1, 2, and 3, respectively. The results show that the proposed protocols with the two versions (with and without scheduling) outperform the other protocols.



(a)

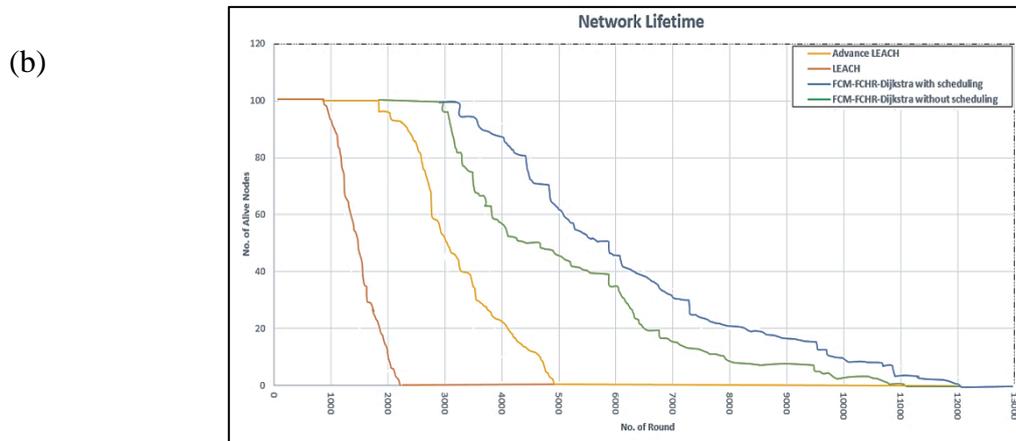
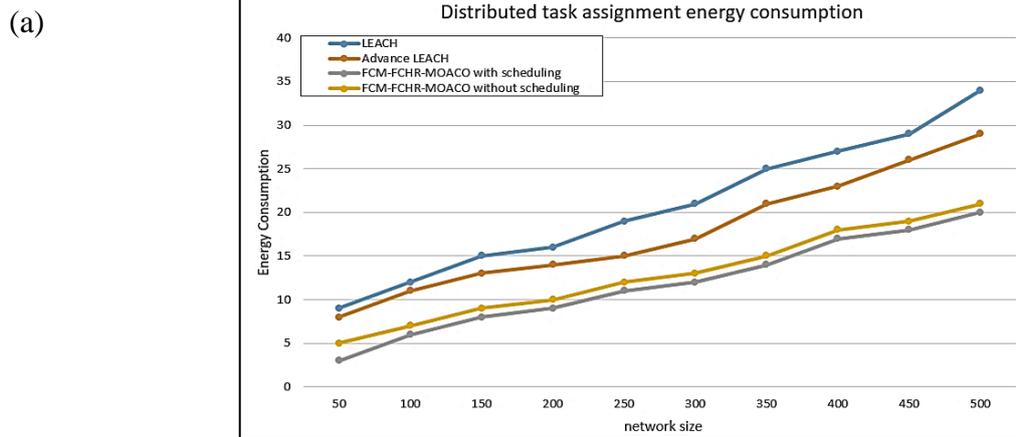
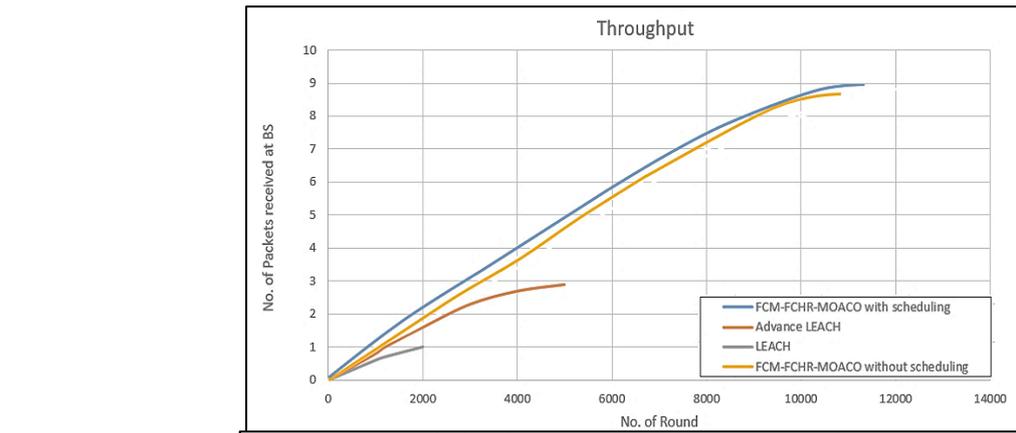


(b)

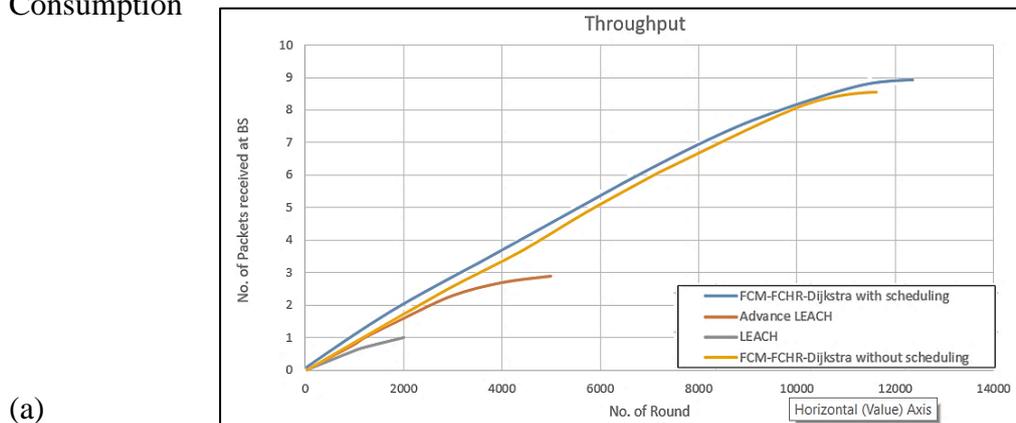


(c)

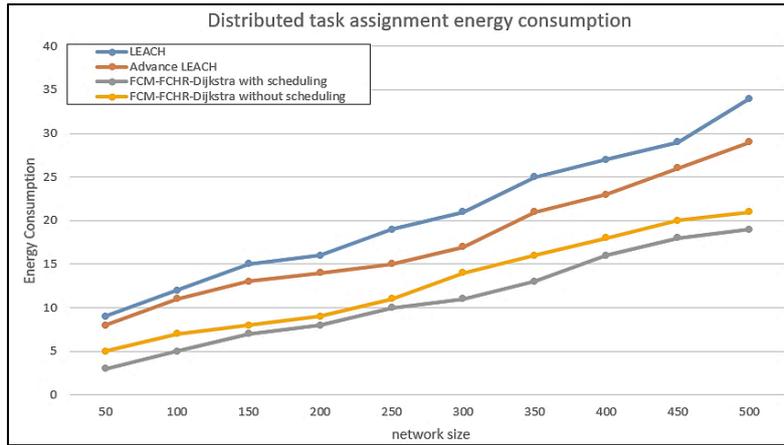
Figure 1. FCM-FCHR-MOEA/D-RP (a) Network lifetime (b) Throughput (c) Energy Consumption



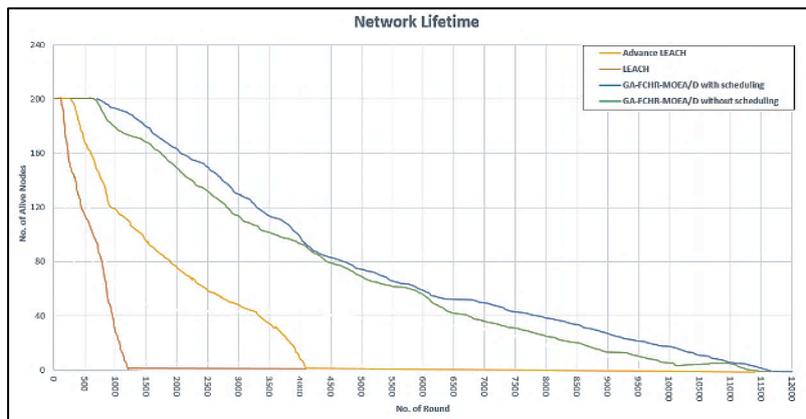
(c) **Figure 2-FCM-FCHR-MOACO-RP** (a) Network lifetime (b) Throughput (c) Energy Consumption



(a)

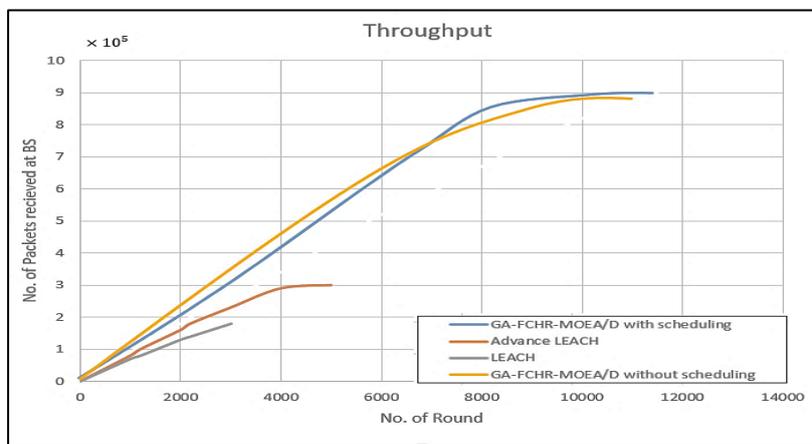


(b)

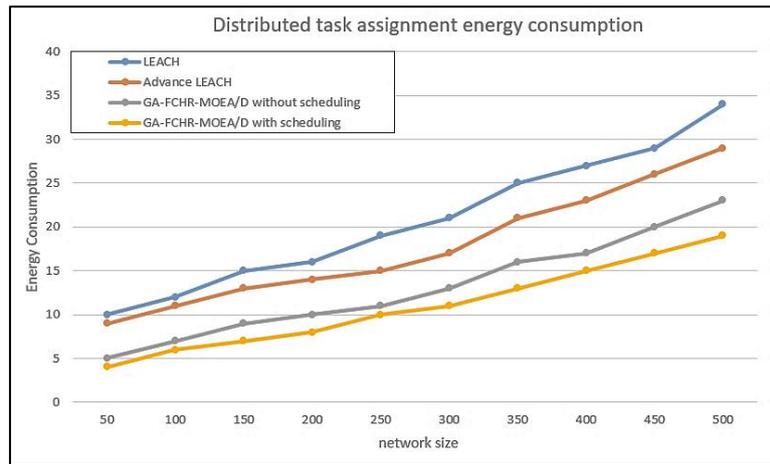


(c)

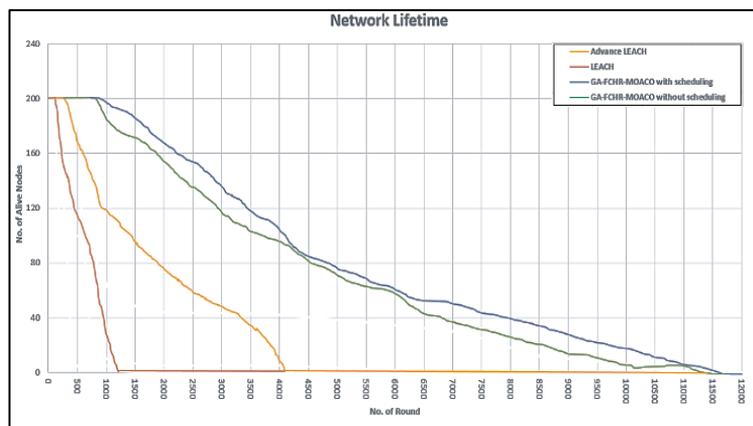
Figure 3-FCM-FCHR-Dijkstra-RP (a) Network lifetime (b) Throughput (c) Energy Consumption



(a)



(b)

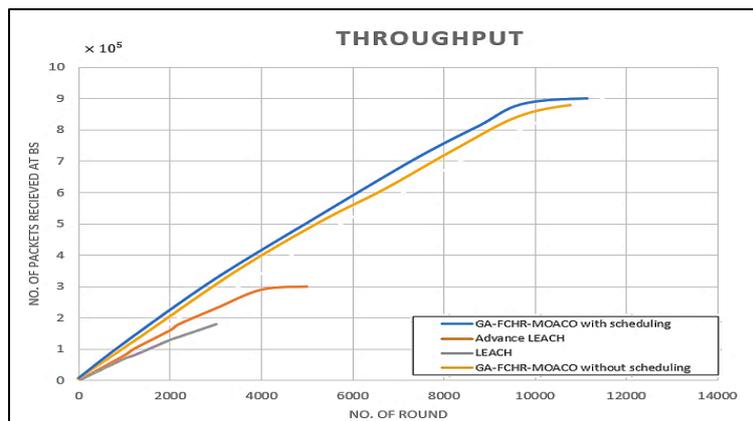


(c)

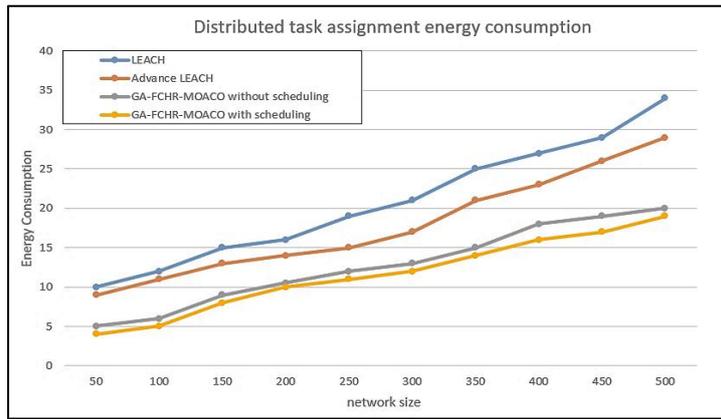
Figure 4. GA-FCHR-MOEA/D-RP (a) Network lifetime (b) Throughput (c) Energy

Consumption

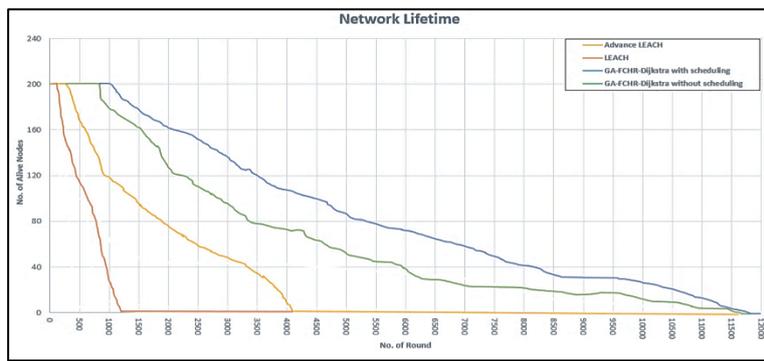
The results of the GA-FCHR-MOEA/D-RP, GA-FCHR-MOACO-RP, and GA-FCHR-Dijkstra-RP are presented in Figures 4, 5, 6, respectively, with network size of 200×200 , nodes' number of 200, and clusters' number of 10.



(a)

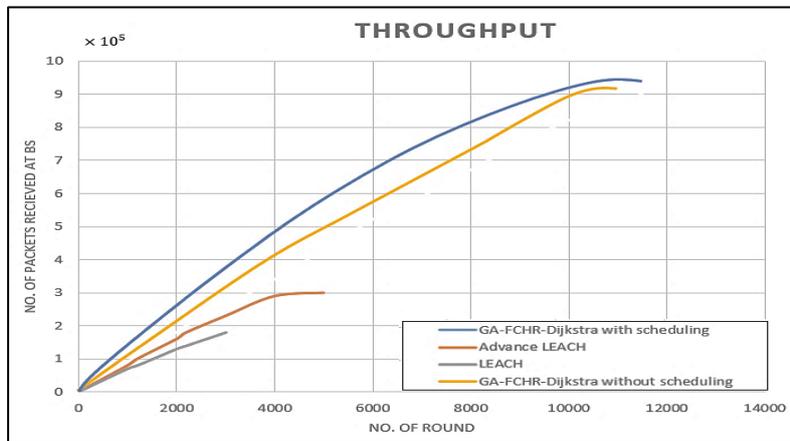


(b)

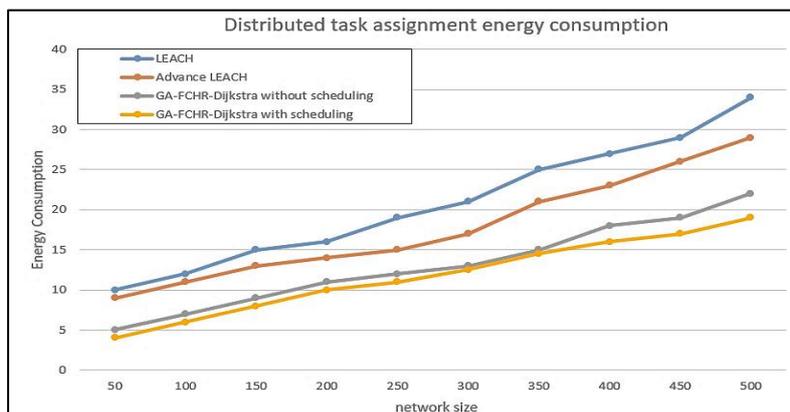


(c)

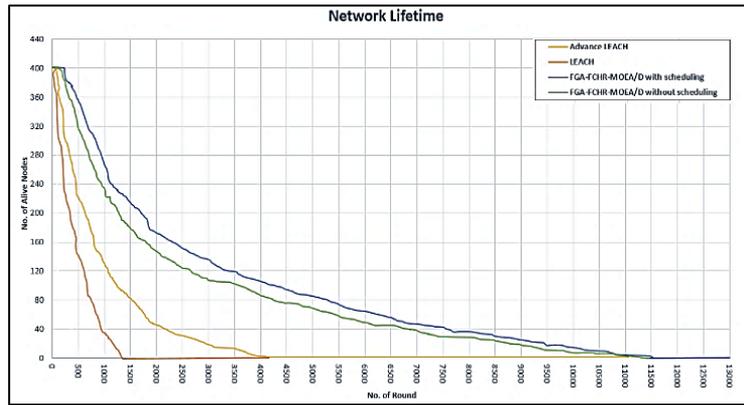
Figure 5-GA-FCHR-MOACO-RP (a) Network lifetime (b) Throughput (c) Energy Consumption



(a)



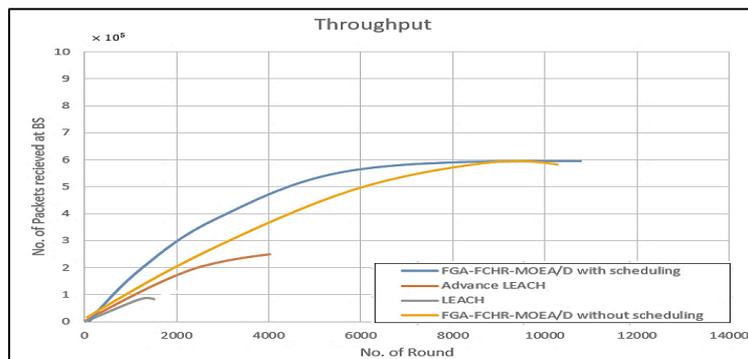
(b)



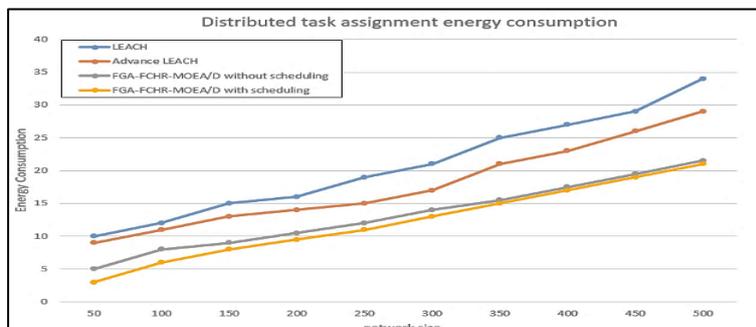
(c)

Figure 6-GA-FCHR-Dijkstra-RP (a) Network lifetime (b) Throughput (c) Energy Consumption

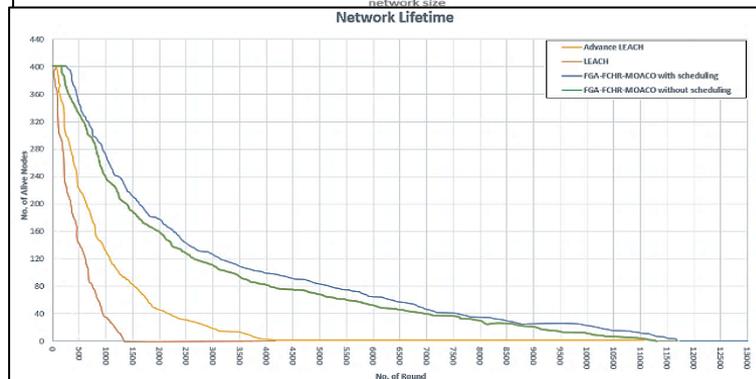
The results of the FGA-FCHR-MOEA/D, FGA-FCHR-MOACO, and FGA-FCHR-Dijkstra are presented in Figures 7, 8, 9, respectively, with network size of 400×400 , nodes' number of 400, and clusters' number of 15. Each figures exhibits the comparison results in terms of Network-lifetime, throughput, and Energy energy consumption of network.



(a)



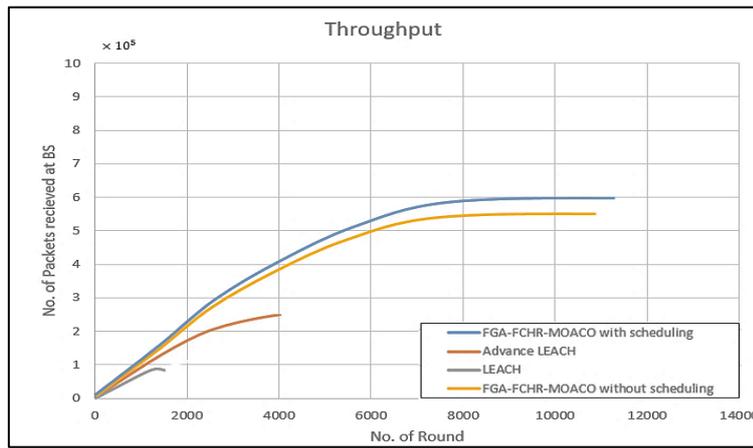
(b)



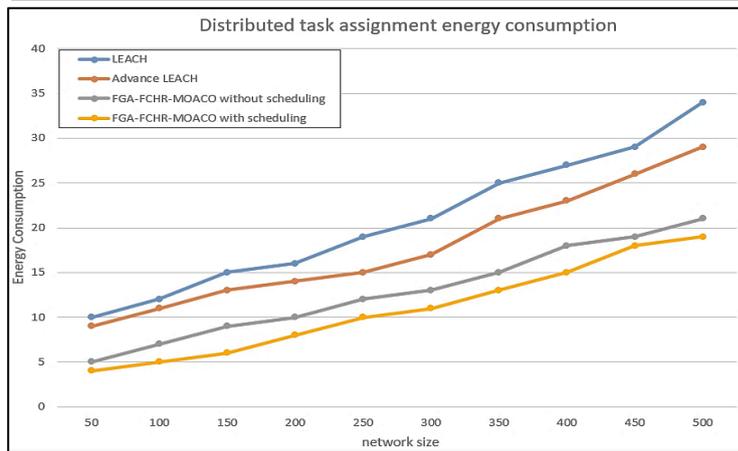
(c)

Figure 7-FGA-FCHR-MOEA/D-RP (a) Network lifetime (b) Throughput (c) Energy Consumption

(a)



(b)



(c)

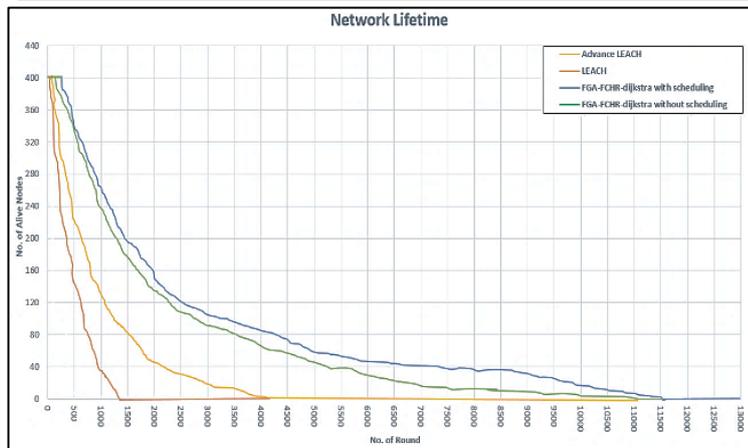
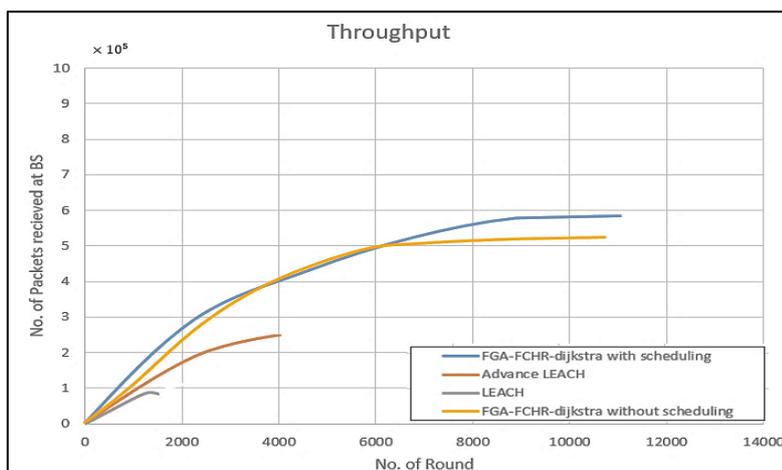
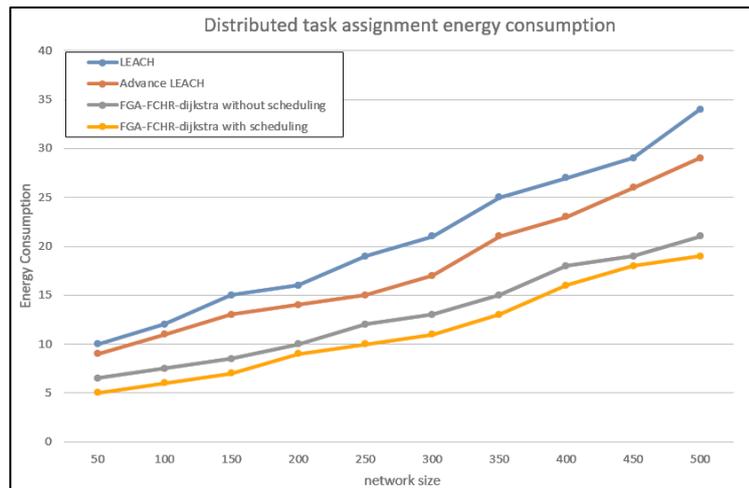


Figure 8-FGA-FCHR-MOACO-RP (a) Network lifetime (b) Throughput (c) Energy Consumption

(a)





(b)

Figure 9. FGA-FCHR-Dijkstra-RP (a) Network lifetime (b) Throughput (c) Energy Consumption

The simulation results reveal that the suggested protocols decrease the total energy consumption, extend the network lifetime, and increase the throughput. The results of the simulation show that the proposed protocols work well in all topologies tested. In the proposed protocols, the average energy consumption is less than that of LEACH, and A-LEACH protocols. It could be shown that the network's energy usage is saved; therefore, the network lifetime is extended and the throughput is increased. The suggested protocols thus make the network last longer with higher stability.

6. Conclusions

In this paper, the network life-time maximization was discussed by using machine learning approaches for WSN protocols. The proposed protocols which are based on ML algorithms depend on many parameters for selecting the CHs during CH rotation, scheduling the nodes, and selecting the path toward the Sink. These parameters are the sensor nodes' residual energy, distances between the CHs, and distances between CHs and sink. The ML algorithms were combined for the sake of minimizing the consumption of energy in addition to maximizing the network life-time. The performance analysis of the proposed protocols are compared with two existing protocols via various network scenarios. The experimental results confirm that the suggested protocols provide better performance compared with other existing protocols concerning network life-time, energy-consumption, and number of rounds until the first node dies. This work will be expanded in the future through the integration of congestion control and security problems when moving packets from CHs to the sink node.

References

- [1] Hussain, S., Matin, A.W. and Islam, O., April. Genetic algorithm for energy efficient clusters in wireless sensor networks. In Fourth International Conference on Information Technology (ITNG'07) (pp. 147-154). IEEE, 2007.
- [2] Peiravi, A., Mashhadi, H.R. and Hamed Javadi, S., "An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm". *International Journal of Communication Systems*, vol. 26, no. 1, pp.114-126, 2013.
- [3] Wang, J., Cao, J., Ji, S. and Park, J.H., "Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks". *The Journal of Supercomputing*, vol. 73, no. 7, pp.3277-3290, 2017.

- [4] Yigitel, M.A., Incel, O.D. and Ersoy, C., "QoS-aware MAC protocols for wireless sensor networks: A survey". *Computer Networks*, vol. 55, no. 8, pp.1982-2004, 2011.
- [5] Kuila, P. and Jana, P.K., "Heap and parameter-based load balanced clustering algorithms for wireless sensor networks". *International Journal of Communication Networks and Distributed Systems*, vol. 14, no. 4, pp.413-432, 2015.
- [6] Kuila, P. and Jana, P.K., "A novel differential evolution based clustering algorithm for wireless sensor networks". *Applied soft computing*, vol. 25, pp.414-425, 2014.
- [7] Gharaei, N., Bakar, K.A., Hashim, S.Z.M. and Pourasl, A.H., "Energy-Efficient Intra-Cluster Routing Algorithm to Enhance the Coverage Time of Wireless Sensor Networks". *IEEE Sensors Journal*, vol. 19, no. 1, pp.4501-4508, 2019.
- [8] Gharaei, N., Bakar, K.A., Hashim, S.Z.M. and Pourasl, A.H., "Energy-Efficient Intra-Cluster Routing Algorithm to Enhance the Coverage Time of Wireless Sensor Networks". *IEEE Sensors Journal*, vol. 19, no. 12, pp.4501-4508, 2019.
- [9] Guo, W. and Zhang, W., "A survey on intelligent routing protocols in wireless sensor networks". *Journal of Network and Computer Applications*, vol. 38, pp.185-201, 2014.
- [10] Pachlor, R. and Shrimankar, D., "LAR-CH: A cluster-head rotation approach for sensor networks". *IEEE Sensors Journal*, vol. 18, no. 23, pp.9821-9828, 2018.
- [11] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H., 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, 2000.
- [12] Lindsey, Stephanie, and Cauligi S. Raghavendra., "PEGASIS: Power-efficient gathering in sensor information systems. In Proceedings", *IEEE aerospace conference*, vol. 3, pp. 3-3, 2002.
- [13] Chakraborty, A., Mitra, S.K. and Naskar, M.K., "A Genetic algorithm inspired routing protocol for wireless sensor networks". *International Journal of Computer Intelligence-Theory and Practice*, no. 6, 2011.
- [14] Kim, J.M., Park, S.H., Han, Y.J. and Chung, T.M., February. CHEF: cluster head election mechanism using fuzzy logic in wireless sensor networks. In 2008 10th International Conference on Advanced Communication Technology (Vol. 1, pp. 654-659). IEEE, 2008.
- [15] Ma, Z., Li, G. and Gong, Q., "Improvement on leach-c protocol of wireless sensor network (leach-cc)". *International Journal of Future Generation Communication and Networking*, vol. 9, no. 2, pp.183-192, 2016.
- [16] Ran, G., Zhang, H. and Gong, S., "Improving on LEACH protocol of wireless sensor networks using fuzzy logic". *Journal of Information & Computational Science*, vol. 7, no. 3, pp.767-775, 2010.
- [17] Bari, A., Wazed, S., Jaekel, A. and Bandyopadhyay, S., "A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks". *Ad Hoc Networks*, vol. 7, no. 4, pp.665-676, 2009.
- [18] Gupta, S.K., Kuila, P. and Jana, P.K.,. GAR: An energy efficient GA-based routing for wireless sensor networks. In International conference on distributed computing and internet technology (pp. 267-277). Springer, Berlin, Heidelberg, 2013, February .
- [19] Gupta, S.K. and Jana, P.K., "Energy efficient clustering and routing algorithms for wireless sensor networks: GA based approach". *Wireless Personal Communications*, vol. 83, no. 3, pp.2403-2423, 2015.
- [20] Ruwaida M.Y., Hybrid Routing Algorithms for Wireless Sensor Networks using Soft Computing Techniques. Ph.D. Thesis, Department of computer science, the Informatics Institute for Postgraduate Studies at the Iraqi Commission for Computers and Informatics, Iraq, 2020.
- [21] Yas, R.M. and Hashem, S.H., "Using Fuzzy based clustering and Cluster-Head rotation for Energy efficient Routing Protocol in WSNs". *International Journal of Advanced Science and Technology*, vol. 29, no. 10s, 2020.
- [22] Yas, R.M. and Hashem, S.H., "Unequal clustering and scheduling in Wireless Sensor Network using Advance Genetic Algorithm". In *Journal of Physics: Conference Series* , vol. 1530, no. 1, p. 012076, 2020. IOP Publishing.