sensors

MDPI

Article

An Energy-Efficient Evolutionary Clustering Technique for Disaster Management in IoT Networks

Morteza Biabani [1,*], Hossein Fotouhi [2] and Nasser Yazdani [1]

[1] School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14395-515, Iran; yazdani@ut.ac.ir

[2] School of Innovation, Design and Engineering, Mälardalen University, 721 23 Västerås, Sweden; hossein.fotouhi@mdh.se

* Correspondence: mbiabani@ut.ac.ir

Received: 15 March 2020; Accepted: 30 April 2020; Published: 6 May 2020

Abstract: Wireless Sensor Networks (WSNs) are key elements of Internet of Things (IoT) networks which provide sensing and wireless connectivity. Disaster management in smart cities is classified as a safety-critical application. Thus, it is important to ensure system availability by increasing the lifetime of WSNs. Clustering is one of the routing techniques that benefits energy efficiency in WSNs. This paper provides an evolutionary clustering and routing method which is capable of managing the energy consumption of nodes while considering the characteristics of a disaster area. The proposed method consists of two phases. First, we present a model with improved hybrid Particle Swarm Optimization (PSO) and Harmony Search Algorithm (HSA) for cluster head (CH) selection. Second, we design a PSO-based multi-hop routing system with enhanced tree encoding and a modified data packet format. The simulation results for disaster scenarios prove the efficiency of the proposed method in comparison with the state-of-the-art approaches in terms of the overall residual energy, number of live nodes, network coverage, and the packet delivery ratio.

Keywords: IoT networks; energy-efficient clustering and routing; disaster management; non-uniform distribution of events; evolutionary algorithms; tree encoding; forest fire

1. Introduction

The Internet of Things (IoT) includes a large number of distributed nodes which are capable of sensing and interacting remotely [1,2]. Wireless Sensor Networks (WSNs) are classified as the major players of IoT networks and provide a wireless connection between constrained devices. In IoT-based WSNs, nodes transmit collected data (such as environmental information) to a central destination known as a sink node through a wireless channel [3,4]. Due to the recent advancements in hardware design, sensor devices are tiny and have a low cost, enabling their deployment in harsh environments such as nuclear power plants and forests [2]. However, sensor devices have very limited battery power, requiring the design of energy efficient algorithms [2,5]. Limitations in energy resources have a direct impact on the application performance, as sensors with depleted batteries can cause network disconnections and consequently packet losses [6–9]. In a multi-hop sensor network, it is crucial to devise algorithms and mechanisms in such a way to select the optimal route for data transfer to reduce energy consumption [2,5,10,11].

There are several IoT applications that are classified based on their requirements and specifications. Safety-critical IoT applications have very strict requirements in terms of reliability and timeliness. This class of applications is defined as a set of behaviors and actions which are taken to prevent, control, and overcome disasters (e.g., fires and earthquakes) [2,6,9,12]. We define disaster management as techniques that deal with devising algorithms and protocols that provide an enhanced quality of service in safety-critical applications. Forest fire detection is an example of a safety-critical application; an unexpected fire can cause a great deal of environmental and ecological damage while threatening human life [13,14]. Regardless of the cause of the crisis, disasters such as earthquakes and fires create an emergency position and have irreparable effects on human society. Recently, due to the escalation of disasters, researchers have focused on the design of warning and remote control systems [15]. IoT protocols and algorithms provide more energy-efficient, portable, and scalable solutions to various disaster problems [2]. This technology provides an open opportunity for the development of disaster-resilient smart environments such as smart community solutions. In this paper, the proposed method uses IoT networks to remotely monitor forest fires. If a disaster occurs, an early warning system is supposed to perform disaster reporting, connecting to a central control unit that takes preventive actions. Due to the weather situation, forest fires may undergo sudden changes that make the monitoring process more complex as the location and spread of fire varies over time [13]. Furthermore, due to the non-uniform distribution of events and the uncontrolled behavior of fire, sensor nodes are unlikely to have a uniform energy consumption. There are some works in the literature focusing on clustering and routing approaches for forest fire management [13,16–20]. Most of these works are focused on rapid fire detection and have neglected the energy efficiency of the network as one of the key elements [17,19]. To the best of our knowledge, existing research works into WSNs in disaster management cover a uniform distribution of events [18,19,21]. These methods cannot be practically efficient in disaster management with a non-uniform distribution of events, which is more common in real experiments. Thus, disaster management using IoT-based WSNs requires an appropriate routing to be created from the cluster heads (CHs) to the sink in the disaster area.

Clustering is one of the techniques which is widely used for hierarchical routing [22–26]. The purpose of clustering is to search for and find optimal CH nodes. Due to the complexity of searching for optimal nodes as CHs, clustering in WSNs is time-consuming. In fact, clustering is known as an NP-hard problem [22,23,27]. However, previous studies show that during cluster head selection with traditional algorithms, it is necessary to employ evolutionary algorithms to increase network efficiency [6,27–30].

In this paper, the energy-efficient cluster head selection and routing (ECHSR) method based on evolutionary algorithms such as Particle Swarm Optimization (PSO) and the Harmony Search Algorithm (HSA) are proposed for disaster management. Although PSO has advantages such as being highly dynamic, featuring rapid convergence, and escaping local optimal solutions [23,27,29], in the face of high-dimensional optimization, it can be very difficult to search for all the possible solutions in every part of the search space [29,31,32]. Having the advantage of a high-search ability, the HSA is capable of solving this problem. Moreover, PSO is confined to exploration and HSA to exploitation, and it is wise to balance exploration and exploitation. Therefore, a hybrid PSO–HSA method will increase energy efficiency more than using these techniques separately [29]. Some studies indicate that meta-heuristic algorithms such as HSA and PSO cannot lead to maximum performance as it is necessary to overcome premature convergence to provide the optimal balance between exploration and exploitation [31,33,34]. Therefore, the modified hybrid PSO–HSA method can be a suitable choice for clustering in forest fires.

This study is conducted in two main phases: (i) devising an improved hybrid PSO–HSA-based CH selection method, and (ii) developing PSO-based routing with modified tree encoding. In the formulation of the fitness function of the clustering phase, new criteria entitled “energy efficiency”, “network closeness” and “network coverage” are considered. In addition to solving fitness function formulas, we use the adaptive weighted sum (AWS) method, which is based on Pareto optimization[28,35–38]. Furthermore, for the routing phase, we use the PSO to select relay nodes (RNs) and send packets from CHs to the sink node. However, the PSO may increase the number of invalid paths, which is not suitable for this application [7,27,39]. Thus, in order to reduce the invalid paths for our application, we provide an optimal encoding tree based on a new type of packet format. The simulation results demonstrate that, compared to the state-of-the-art, the proposed method increases the network lifetime by expanding the range of network coverage. The contributions of this paper are summarized as follows:

- Providing an energy-efficient method for clustering and routing in WSNs while considering the characteristics of disaster areas (with a non-uniform distribution of events).
- Proposing an improved hybrid PSO–HSA-based CH selection method with a novel fitness function including energy-efficiency, cluster closeness, and network coverage criteria.
- Utilizing the PSO to select relay nodes based on multi-hop routing with a fitness function including energy-efficiency and communication link quality criteria.
- Using the adaptive weighted sum (AWS) method as a mathematical model to solve the fitness function based on the multi-objective optimization problem (MOP) so that a lower optimization cost can be obtained.
- Developing a new type of packet format which is adapted to use in disaster applications; furthermore, modifying the PSO encoding priority-based routing to build an optimal routing tree with non-uniform events.
- Overcoming the state-of-the-art approaches in terms of their average energy consumption, the number of live nodes, network coverage, and packet delivery ratio with various sink locations in disaster scenarios.

The rest of the paper is organized as follows. A review of the related works is given in Section 2, which is then followed by our proposed method in Section 3. The simulation results and evaluations are described in Section 4. Finally, Section 5 concludes the work and explains the possible future works.

2. Related Works

In [22], the authors introduced a low-energy adaptive clustering hierarchy (LEACH), which is a popular distributed heuristic method. Clustering in this method is dynamic, random, probabilistic and periodic. LEACH determines a threshold for CH selection. It produces a random number between [0, 1] to select a CH in that period. One of the disadvantages of the LEACH protocol is the random selection of CHs, because all nodes can be eligible to be a CH. Another hierarchical heuristic method, named the threshold-sensitive energy efficient sensor network protocol (TEEN), was proposed to improve energy consumption using the two-layer structure for clustering [40]. In this method, two thresholds are applied: (i) the hard threshold (HT) and (ii) the soft threshold (ST). Each CH sends its threshold values to its members. Both thresholds are defined in order to reduce the number of transfers over the routing periods and thus to reduce energy consumption in the process. In [40], CH selection is based on probability, and the distribution of CHs will be non-uniform. The TEEN algorithm has improved clustering stability and energy efficiency compared to LEACH. Its main drawback is that nodes will never establish communication if the thresholds are not reached.

To overcome the limitations of heuristic algorithms, such as inefficiently exploring the search space, meta-heuristic algorithms have been proposed for optimal CH selection [27,29]. Access to every portion of the entire search space over each period is considered to be one of the most important criteria for increasing efficiency in these algorithms, which leads to optimal or near-optimal solutions. Meanwhile, evolutionary algorithms (EAs) such as PSO and HSA are some of the most famous population-based algorithms, and their fitness function determines the quality of the solutions [32].

The authors in [23] proposed an evolutionary PSO for energy-aware CH selection by combining criteria such as the Euclidean distance between the nodes and CHs and the remaining energy from particle energy to node energy, allowing an evaluation of the issues related to the degree of the nodes and the number of CH hops required to reach particle sinks. An analysis of the results proved that the inclusion of energy criteria in the fitness function plays a crucial role in energy efficiency. In [23], CH selection was based on probabilities, and the CH distribution was uniform. This method also focused more on the exploration stage than exploitation. In [41], the authors presented an evolutionary HSA inspired by music for CH selection. Musicians try to select better notes in order that their music can be heard better, which forms the concept of presenting the harmony fitness function. In other words, the best set of CHs is selected. The authors focused on the two criteria of energy (HE) and distance (HD) for harmony evaluation. The fitness function of the Hoang schema is as follows:

$$\mathrm{f}_{HS} = \alpha_1 \times \mathrm{HD} + \alpha_2 \times \mathrm{HE} \tag{1}$$

HD and HE are calculated according to Equations (2) and (3):

$$\mathrm{HD} = \max_{\mathrm{k}} \left\{ \sum_{\forall \mathrm{n_i} \in \mathrm{CM_k}} \frac{\mathrm{d}\left(\mathrm{n_i}, \mathrm{CH_k}\right)}{\parallel \mathrm{CM} \parallel} \right\} \tag{2}$$

$$\mathrm{HE} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{E}\left(\mathrm{n_i}\right)}{\sum_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{E}\left(\mathrm{CH_j}\right)} \tag{3}$$

It is noteworthy that α is a weight factor, and its value lies between 0 and 1. The furthest Euclidean distance of each node involved in a cluster to its CH (CH_k) is evaluated according to HD, in which the number of the members in each cluster is presented by $\parallel$ CM $\parallel$. HE is also evaluated by dividing the sum of current energy in all network nodes $E(n_i)$ by the sum of the current stage of CH energy $E(CH_j)$. In [41], CH selection is random and CH distribution is non-uniform. Additionally, HSA is focused more on the exploitation stage than exploration. In [27], the authors introduced a two-tier PSO protocol for clustering and routing in WSNs entitled TPSO-CR. The three measures used in the fitness function of the clustering were network coverage, the remaining energy of nodes, and the link communication quality. Moreover, the two criteria utilized in the fitness function of the routing were energy-efficiency and link communication quality, which are described in the following. In this paper, the energy criteria are used in an attempt to reduce the number of the sender nodes; by doing so, the residual energy will be saved optimally. Furthermore, tree encoding is facilitated by applying PSO to construct optimal multi-hop routing and determine the Relay Nodes (RNs). Thus, Equation (4) is defined in a more energy-saving state (ER1) as follows:

$$\mathrm{ER}_1 = \frac{||\mathrm{RNs}||}{||\mathrm{actCH}||} \tag{4}$$

ER1 divides the number of RNs by the number of active CHs in the corresponding period $\parallel$ actCH $\parallel$. In fact, Equation (4) reduces the number of active nodes. In addition, the RN with the highest level of energy should be offered as the sender node. ER2 adheres to that concept, as per Equation (5):

$$\mathrm{ER}_2 = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{E}\left(\mathrm{node_i}\right)}{\sum_{\mathrm{j}=1}^{\mathrm{RN}} \mathrm{E}\left(\mathrm{RN_j}\right)} \tag{5}$$

Similarly, ER2 divides the sum of the energy of all nodes by the sum of the RN energy and moderates the energy consumption. Next, maximizing link quality (LiQ) between RNs in the multi-hop routing tree leads to the maximization of the packet delivery ratio (PDR). Hence, the worst communication link quality must be avoided in all sub-branches of the routing tree. Equation (6) is calculated according to this objective as follows [27]:

$$\mathrm{LiQ} = Max_{rn} \sum_{\forall \mathrm{rn}_x \in \mathrm{RN}} \frac{\mathrm{RSSI}\left(\mathrm{rn_x}, \mathrm{rn_{x+1}}\right)}{\mathrm{minRSSI}} \tag{6}$$

The received signal strength indicator (RSSI) is used in this paper to evaluate the link quality between the cluster nodes. In [27], the link quality is the ratio of the RSSI between the CMs and their CHs to minRSSI. In Equation (6), it can be observed that we prevent the selection of the worst link quality (minRSSI) as far as possible between two consecutive sender nodes RSSI $\left(\mathrm{rn_x}, \mathrm{rn_{x+1}}\right)$. Hence, In order to maximize the quality of the clusters in terms of LiQ, we must minimize the worst quality of the clusters. So, the fitness function of routing is formulated as follows:

$$\min \mathrm{F}_{\mathrm{routing}} = \beta_1 \times \mathrm{ER}_1 + \beta_2 \times \mathrm{ER}_2 + \beta_3 \times \mathrm{LiQ} \tag{7}$$

where $\beta_j \in [0, 1]$. Furthermore, the authors in [29] produced a hybrid method based on PSO and HS algorithms for energy-efficient CH selection. The proposed meta-heuristic algorithm takes advantage of both algorithms, covering a global optimal in the search space (exploration) and going beyond the local optimal (exploitation). Furthermore, the authors were inspired by Equation (1) [41] as regards the fitness function, with the aim of energy-efficiently minimizing the average distance of the normal nodes to the CHs. In PSO–HSA [29], the particles are allowed to move from one region to another area of the search space based on the PSO algorithm. Since PSO encounters optimization limitations in high-dimensional scales, it is difficult to find any possible solution in the entire search space. Therefore, HSA helps PSO because of its high searching capacity. In [42], the authors used the cuckoo algorithm for clustering and employed an improved HSA for multi-hop routing in WSNs. They suggested that uniform energy consumption in the clustering and routing of large-scale wireless sensor networks is one of the main concerns in this area, because, due to traffic load, nodes near the sink suffer from high energy consumption. This paper also proposes criteria such as the energy, node degree, intra distance of clusters and the coverage ratio for the fitness function.

The performances of different clustering protocols in WSNs are summarized in Table 1, from which the following can be concluded: (1) the Hybrid PSO-HSA [29] is more energy-efficient than the cuckoo [42], PSO [23], and HS [41] algorithms; (2) the cuckoo algorithm, in comparison with PSO and HSA, has poor cluster stability, and all of the algorithms suffer from an imbalance of exploration and exploitation; (3) none of the related works into WSNs consider the non-uniform distribution of events, and this is not suitable for disaster application.

Table 1. Comparing different clustering protocols in Wireless Sensor Networks (WSNs) with our proposed method. LEACH: low-energy adaptive clustering hierarchy; TEEN: threshold-sensitive energy efficient sensor network protocol; PSO: Particle Swarm Optimization; HSA: Harmony Search Algorithm; CH: cluster head. TPSO-CR: Two-tier PSO protocol for Clustering and Routing. iCSHS: Integrated Clustering for Cuckoo Search and Harmony Search.

Reference	Clustering Schema	Converge	Energy-Efficient	Cluster Stability	Event Awareness	Routing	Clustering Objective
LEACH [22]	Random Distributed	No	Low	Low	No	single-hop	Changing CH based Prob./Random
TEEN [40]	Probability Centralized	No	Medium	Low	No	single-hop	Changing CH based Threshold
[23]	PSO-based Centralized	Low balanced	Moderate	High	No	single-hop	Energy Efficiency + Min Distance + Node Hop count
[41]	HSA-based Centralized	Low balanced	Medium	Low	No	single-hop	Energy Efficiency + Min Distance
TPSO-CR [27]	PSO-based Centralized	Moderate balanced	Moderate	High	No	multi-hop	Energy Efficiency + link Quality + Coverage
PSO-HSA [29]	PSO and HSA Centralized	Moderate balanced	High	High	No	single-hop	Energy Efficiency + Min Distance
iCSHS [42]	Cuckoo-based Distributed	Moderate balanced	Moderate	Low	No	multi-hop	Energy Efficiency + Node degree + Intra Distance + Coverage
This Paper	PSO and HSA Centralized	High balanced	very High	High	Yes	multi-hop	Energy Efficiency + Cluster Closeness + Coverage

3. Proposed Method

In this section, first, the network model is explained; then, by determining the AWS method for fitness functions, two main phases of our method are proposed.

3.1. Network Model

In our protocol, the nodes are scattered homogeneously and randomly with a uniform distribution in a rectangular field (100 m × 200 m). Figure 1 indicates the overall network model, including a description of the uniform random distribution of the nodes in a disaster application such as a forest fire.

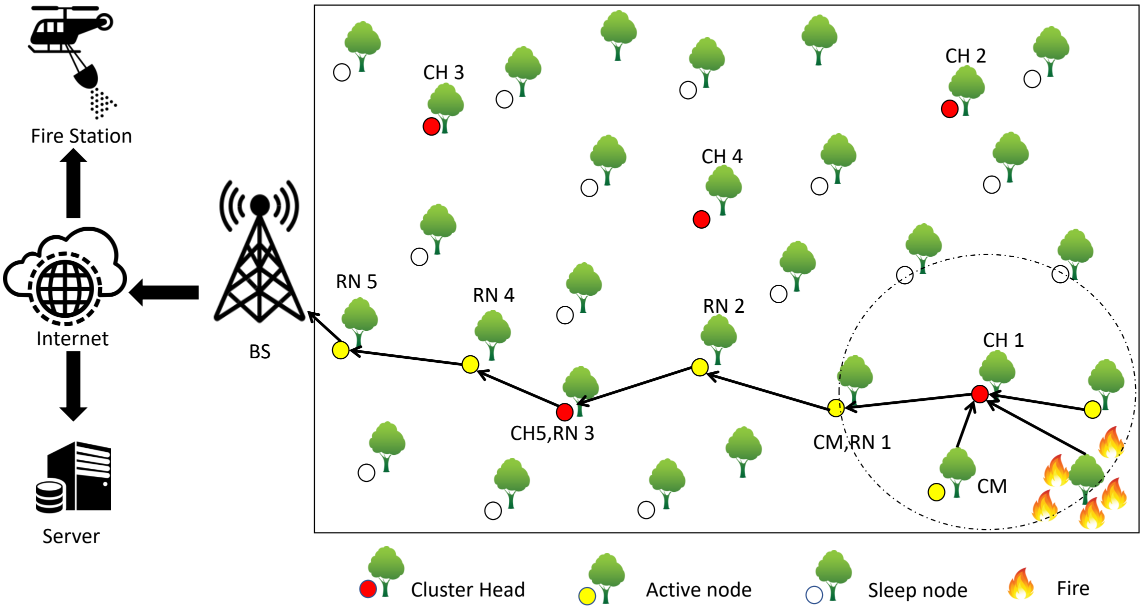


Figure 1. The network topology of a typical WSN network for disaster management.

Our method is configured based on the following assumptions, and the location of the sink is considered to be variable according to the evaluation of the algorithm's efficiency in the target application.

- The nodes are fixed (stationary) in the network.
- Nodes are not location-aware, but the sink is aware of its position.
- All nodes have the same level of energy (E0) initially, and the network is homogeneous [7].
- The minimum RSSI with the value of −97 db is known as the worst received signal [27].
- The number of CHs (k) is considered to be 5% of all nodes (N) [27,29,30].
- Nodes always send their data based on Time Division Multiple Access (TDMA), which is enforceable on IEEE 802.15.4/ZigBee standard. Additionally, we use the Timeout MAC (TMAC) protocol because it is compatible with different sleep schedules in WSNs and provides energy efficiency [27,43].
- The sink is responsible for clustering and routing. Therefore, the proposed method is centralized.
- The energy consumption of the nodes is estimated based on their distance from the sink or CH (by parameter d) [22].

In [22], the authors evaluate the energy consumption model based on the distance between the sender and receiver (d). In this model, the energy dedicated to sending messages is calculated by the length of the messages (L) and the distance between sender and receiver (d) as follows [22]:

$$\mathrm{E_{Tx}}\,(\mathrm{l},\mathrm{d}) = \begin{cases} L.\mathrm{E_{elec}} + L\varepsilon_{\mathrm{fs}}.\mathrm{d}^2 \ \ ,if & d \leq \mathrm{d}_0 \\ L.\mathrm{E_{elec}} + L\varepsilon_{\mathrm{amp}}.\mathrm{d}^4, if & d > \mathrm{d}_0 \end{cases} \tag{8}$$

In (8), the energy consumption to send one bit of data is calculated by $\mathrm{E_{elec}}$. Additionally, sending factors of $\varepsilon_{\mathrm{fs}}$ and $\varepsilon_{\mathrm{amp}}$ are valued in proportion to the booster model of the sender-radar. In [22], the consumed and dedicated energy of length L (in bits) will be calculated according to Equation (9) to receive each message.

$$\mathrm{E_{Rx}} = \mathrm{L.E_{elec}} \tag{9}$$

It is noteworthy that the initial distance d0 is calculated as follows in Equation (10):

$$d_0 = \sqrt{\frac{\varepsilon_{\mathrm{fs}}}{\varepsilon_{\mathrm{mp}}}} \tag{10}$$

3.2. AWS Method for LP Formula

One of the most significant challenges in disaster management is decision-making in harsh environments [44]. In these cases, the decision-maker faces different criteria influenced by the internal and external environment. Hence, multi-criteria decision making (MCDM) schemas are considered to be among the best tools to make decisions [36,39]. In MCDM, instead of using an optimality criterion, several metric criteria may be used. Among the methods to solve these problems is the adaptive weighted sum (AWS) method [38]. As the simplest method of MCDM analysis, the AWS method can be used only when all criteria are stated precisely in terms of the same domain. Moreover, it permits a multi-objective optimization problem (MOP) to be converted into a single objective mathematical optimization problem (SOMP) format. The MOP method is formulated as follows [28,37,38]:

$$\mathrm{MOP} : \min_{\mathrm{x} \in \Omega} \mathrm{F}\,(\mathrm{x}) = \min_{\mathrm{x} \in \Omega}\, (\mathrm{f}_1\,(\mathrm{x})\,, \mathrm{f}_2\,(\mathrm{x})\,,\ \ldots, \mathrm{f_n}\,(\mathrm{x})) \tag{11}$$

subject to $F(x) : \Omega \to R^n$. Multiplying the sum of the fitness functions in the weighted coefficients, the AWS method transforms Equation (11) to the SOMP as per Equation (12) [38]:

$$\min \sum_{j=1}^{N} w_j.f_j\,(x) \tag{12}$$

subject to $j \in \Omega$, where $w_j \geq 0, (\forall j = 1, 2, \ldots, N), and \sum_{j=1}^{N} w_j = 1$.

In this paper, we use the above equation to determine the fitness functions. Equation (12) is also called Pareto optimization. Typically, this kind of optimization gives a unique solution (if $w_j\ > 0$, $\forall j = 1, 2,\ \ldots,\ N$) [38]. As the proposed fitness functions are composed of different criteria and values, it is also essential to normalize the criteria with respect to the Pareto optimization and match them with the weighted coefficients [38].

3.2.1. Normalizing the LP Formula

In order to obtain a similar amplitude value of the fitness functions, each criterion of these functions is normalized between [0, 1] [36,37]; all criteria of the fitness function should be in one domain. Normalizing the interval between 0 and 1 requires the Utopia and Nadir points [38]. The Utopia point, which is the lower bound ($\mathrm{S^U}$) of the function, is modeled as follows:

$$\mathrm{s}_{\mathrm{y}}^{*} = \mathrm{fo}_{\mathrm{y}}\left(\mathrm{x}^{[\mathrm{y}]}\right) \tag{13}$$

where $\mathrm{s}_{\mathrm{y}}^{*} = \mathrm{argmin}_{\mathrm{x}}\left\{\mathrm{fo}_{\mathrm{y}}(\mathrm{x})\ : \mathrm{x} \in \Omega\right\}$ and subject to $s_y^* \in R, S^* \in R^Y$. The Utopia point is $\mathrm{S}^{\mathrm{U}} = \mathrm{S}^{*}$. The Nadir point, which is the upper bound (S^{N}) of the function, is modeled as follows [38]:

$$s_y^N = \max_{1 \le j \le Y}\left(fo_y\left(x^{[j]}\right)\right), \quad \forall y = 1, 2, \ldots, Y. \tag{14}$$

If considering the difference in values at points, then $\theta_{\mathrm{y}} = \frac{1}{\mathrm{s}_{\mathrm{y}}^{\mathrm{N}} - \mathrm{s}_{\mathrm{y}}^{\mathrm{U}}}$. Finally, the normalization equation is calculated as follows [38]:

$$0 \le \frac{\mathrm{fo}_{\mathrm{y}}(\mathrm{x}) - \mathrm{s}_{\mathrm{y}}^{\mathrm{N}}}{\mathrm{s}_{\mathrm{y}}^{\mathrm{N}} - \mathrm{s}_{\mathrm{y}}^{\mathrm{U}}} \le 1 \tag{15}$$

According to the above equation, it is possible to calculate values at each round of the simulation for each criterion.

3.3. Cluster Head Selection Phase

In the clustering phase, our evolutionary method is described through the contribution of the hybrid PSO–HSA [29] with the difference that the fitness function is modified and improved according to the disaster management conditions. The improved fitness function and stages of clustering will be explained below.

3.3.1. Improved Fitness Function for CH Selection

In order to increase the network lifetime, a set of optimal CHs should be selected; if the number of CHs were too low, nodes would have to increase their communication range to send the information, and so they would consume more energy. However, if the number of CHs were too high, the network would be a single hop, again leading to increased energy consumption. Therefore, it is necessary to determine the optimal criteria of the fitness function based on the application [26]. In this paper, the fitness function in WSN is a multi-objective function that needs to be minimized to reach optimal or near-optimal solutions. Additionally, three criteria have been used to select CHs. First, the energy-efficiency criterion (EE) is included, inspired by Equation (3) of the Hoang method [41], as mentioned above; then, two other criteria are added to the fitness function, which are called cluster closeness (CC) and network coverage (NC).

3.3.2. Cluster Closeness (CC)

To maximize cluster quality, it is necessary to make highly compact clusters to reduce energy consumption [33]. Furthermore, a high density of nodes is a good option for selecting a candidate CH from a region [33] because the energy consumption and internal communication cost of clusters is therefore minimized [14,27,29]. Thus, we attempted to decrease the mean distance of the CH to its neighbors for rapid data transmission. If the density of an arbitrary node x (CH candidate) is the number of single-hop neighbors $(nh(x))$ and the sum of the Euclidean distance of node x from single-hop neighbors $(h(x))$ is $dist(x,\ h(x))$, then the cluster closeness of an arbitrary node x is calculated as follows:

$$\text{Cluster Closeness } (\mathrm{CC}_x) = \frac{\sum_{\mathrm{y} \in \mathrm{h}(\mathrm{x})} \mathrm{dist}(\mathrm{x}, \mathrm{y})}{\mathrm{nh}(\mathrm{x})} \tag{16}$$

A node will be chosen as CH which has a lower value than the other candidate CHs because increasing the $nh(x)$value according to [33] is effective in selecting CHs. Similarly, as in Equation (2), the mean Euclidean distance of the CMs with the CHs is minimized.

3.3.3. Network Coverage (NC)

One of the important factors in overcoming the challenges of WSNs is network coverage [15,27]. If the aim is to reach higher scalability, then the network coverage should be increased [27]. Thus, finding a CH for the nodes that have no CHs (non-CHs) can effectively cover the network. In this paper, the network coverage is evaluated as follows:

$$\text{Network Coverage (NC)} = \frac{(\mathrm{N}^* - \mathrm{K}) - \sum_{\mathrm{k}=1}^{\mathrm{K}} ||\mathrm{CM}_k||}{\sum_{\mathrm{k}=1}^{\mathrm{K}} ||\mathrm{CM}_k||} \tag{17}$$

where $||\mathrm{CM}_k||$ represents the kth total number of CMs in the cluster, K is the number of CHs, and N* is the total number of nodes (N) minus the number of nodes annihilated in each round. Minimizing the number of non-CHs is perfromed through NC [15,27]; thus, by minimizing the *EE*, *CC*, and *NC* parameters, the best set of CHs is selected. A novel fitness function of the clustering phase is formulated as shown in Equation (18):

$$\min \mathrm{F}_{\text{clustering}} = \alpha_1 \times \mathrm{EE} + \alpha_2 \times \mathrm{CC} + \alpha_3 \times \mathrm{NC} \tag{18}$$

subject to EE>E (threshold), CC<T (maximum), $\sum_{i=1}^{3} \alpha_i = 1$ and $\alpha_i \in [0,1]$.

Here, nodes die if their energy is less then E (threshold), and T (maximum) is the maximum range of node transmission.

3.3.4. Clustering Steps

After the initialization of parameters, which is presented in the simulation setup (Section 4), the clustering phase consists of the following steps.

- Setting the Location and Velocity of the Particles

  The location and velocity of the particles are placed randomly in the following intervals, respectively: $[x_{min},\ x_{max}]$ and $[v_{min},\ v_{max}]$.The particle swarm size and harmony memory size (HMS) are initialized under the title of particle harmony memory (PHM). The initial PHM matrix [29], as shown in Equation (19), includes the number of random solutions generated to reach the optimal or near-optimal solution. Each row in the $HMS \times K$ matrix is a random solution, and the fitness function for each vector is calculated by Equation (18).

$$\begin{bmatrix} \mathrm{I}_1^1 & \mathrm{I}_2^1 & \cdots & \mathrm{I}_k^1 \\ \mathrm{I}_1^2 & \mathrm{I}_2^2 & \cdots & \mathrm{I}_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{I}_1^{HMS} & \mathrm{I}_2^{HMS} & \cdots & \mathrm{I}_k^{HMS} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{\mathbf{clustering\_1}} \\ \mathbf{f}_{\mathbf{clustering\_2}} \\ \vdots \\ \mathbf{f}_{\mathbf{clustering\_}n} \end{bmatrix} = \begin{bmatrix} \mathrm{F}^1 \\ \mathrm{F}^2 \\ \vdots \\ \mathrm{F}^n \end{bmatrix} \tag{19}$$

- Improving Harmony Vectors for PHM and Updating the Hybrid Matrix

  After PHM is defined, new harmony vectors $[I'_1,\ I'_2, \ldots,\ I'_k]$ should be produced to improve the harmony vectors of the PHM. Each component of the new harmony vectors is generated by harmony memory considering the rate (HMCR), and eventually an optimal candidate vector of PHM is modified by the pitch adjusting rate (PAR). In this step, the state-of-the-art method [29] uses the simple structure of HSA, but attention must be paid to variations in different stages of the search process. Additionally, the study in [34] demonstrated that HSA suffers from a slow convergence speed to achieve a global optimum; however, in [31], the dynamic version of HSA exhibited fast convergence at the start of the algorithm, converging to the global optimum in the near-final stages of the search process. Therefore, in this paper, each harmony vector (HV) of the PHM is obtained dynamically. The procedure of improving the harmony vectors for PHM is described in Algorithm 1 [31].

Algorithm 1: Pseudo code of Improving Harmony Vectors for PHM

```
Input  : PHM + k+ a new harmony vector $[I'_1, I'_2, \dots, I'_k]$
Output : Improved PHM
initialize parameters
for $x$=1:$K$ do
    /* update 1 (HMCR) based on Equation (23)                              */
    if $randn() <= HMCR$ then
        $\mathrm{I}'_{\mathrm{x}} = \mathrm{HV}(\mathrm{best}, \mathrm{k}) + \mathrm{Nd1} \times (\mathrm{HV}(\mathrm{ri1}, \mathrm{x}) - \mathrm{HV}(\mathrm{ri2}, \mathrm{x}))$      // method 1
        /* update 2 (PAR) based on Equation (24)                           */
        if $randn() <= PAR$ then
            $\mathrm{I}^{\mathrm{n}}_{\mathrm{x}} = \mathrm{HV}(\mathrm{best}, \mathrm{k})$
        end
    else
        $\mathrm{I}'_{\mathrm{x}} = \mathrm{HV}(\mathrm{ri3}, \mathrm{x}) + \mathrm{Nd2} \times (\mathrm{HV}(\mathrm{ri4}, \mathrm{x}) - \mathrm{HV}(\mathrm{ri5}, \mathrm{x}))$      // method 2
    end
end
```

where Nd1 and Nd2 are normal distribution functions with ($\mu = 0, \sigma = 1$), ri1 and ri2 are random integers between [1, HMS], ri3, ri4 and ri5 are random mutually exclusive integers between [1, HMS], the function $randn()$ is a uniform random number in [0, 1], and parameter k is generated uniformly by a discreet random number in [1, K] [31]. After method 1 in Algorithm 1, if $randn()$ $\leq$ PAR, the harmony vector (HV) should be tuned or pitched (a term in music) according to the PAR. $\mathrm{I}^{\mathrm{n}}_{\mathrm{x}}$ is defined as the closest node with more energy than current CH is the cluster. It is necessary to clarify that parameter best in update 2 as the index of the best HV inspired by the GHS method [45]. Next, updating the PHM is the last step in the HS algorithm [29]. To do this, if the fitness function of the new harmony vector ($\mathrm{I}^{\mathrm{n}}_{\mathrm{x}}$) is better than the value of the fitness function of the worst harmony vector in PHM ($f(I^n_x) > f(HV^{worst}_x)$), the new harmony is included in the matrix and the current worst harmony of the PHM is excluded. Thus, the PHM is updated with this condition, and this step is stopped If the iteration of HSA is bigger than the maximum iteration.

- Updating the Location and Velocity of the Particles

In this step, to reach the optimal location of the CHs, the PSO algorithm updates the location and velocity of the particles [2]. The best particle location (p_b) is determined by the minimum cost of the fitness function in a swarm particle [29]. The g_b, as the global best location, is determined by selecting the best solution of p_bs. In this paper, the nodes that harmonize with g_bs are taken, and the closest nodes to these harmonies form CHs. The velocity (V) and location (X) of the particles can be updated as follows [23,29]:

$$\mathrm{V_{xy}}\left(\mathrm{r}+1\right) = \omega \mathrm{V_{xy}}\left(\mathrm{r}\right) + \mathrm{r_a c_a}\left(\mathrm{p\_b_{xy}} - \mathrm{X_{xy}}\left(\mathrm{r}\right)\right) + \mathrm{r_b c_b}\left(\mathrm{g\_b_{xy}} - \mathrm{X_{xy}}\left(\mathrm{r}\right)\right) \tag{20}$$

$$\mathrm{X_{xy}}\left(\mathrm{r}+1\right) = \mathrm{X_{xy}}\left(\mathrm{r}\right) + \mathrm{V_{xy}}\left(\mathrm{r}+1\right) \tag{21}$$

where the index xy denotes the xth particles and dimension, c_a and c_b are acceleration constants, r_a and r_b are the random uniform amount in [0, 1], and factor ω is the inertia weight factor. Selecting a suitable value of ω creates a balance between exploration and exploitation [46]. In [29], the linear reduction of factor ω has a positive effect on the efficiency of the application of WSNs, while in [46], the authors proposed a nonlinear time-varying inertia weight formula that led to a faster convergence than the linear inertia factor used in [29,45]. Therefore, to enhance efficacy, we use the following high-order (∝) nonlinear formula:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min})\left(\frac{\text{iter}}{\text{r}_{\max}}\right)^{\propto} \tag{22}$$

where iter is the current of iteration and $\text{r}_{\max}$ is the total number of iterations. Equation (22) [46] decreases nonlinearly from $\omega_{\max} = 0.9$ to $\omega_{\min} = 0.4$ and finds a suitable value at each round. The obtained results of [46] show that excellent performance is achieved when $\propto \ = 1 / \pi^2$ and the two constants c_a and c_b—as cognitive and social parameters—have the value 2. The location and velocity of the particles are calculated and updated at each round of the proposed method and for each row in PHM. As mentioned above, the closest nodes to *g_b* will be the optimal CHs.

- Going Back to the First Step or Stopping the Algorithm

  When all the rounds of the clustering phase come to an end, the algorithm stops and a total number of *k* optimal CHs are determined for the routing phase. Otherwise, the algorithm goes back to the second step and repeats until it finds optimal or near-optimal solutions. However, before the next round starts, the values of HMCR and PAR in Algorithm 1 are required to be updated as follows [31]:

$$Update1: \text{HMCR}\,(\text{Iter}+1) = \text{HMCR.min} + \left(\frac{\text{Iter}}{\text{r}_{\max}}\right) \times (\text{HMCR.max} - \text{HMCR.min}) \tag{23}$$

$$Update2: \text{PAR}\,(\text{Iter}+1) = \text{PAR.max} - \left(\frac{\text{Iter}}{\text{r}_{\max}}\right) \times (\text{PAR.max} - \text{PAR.min}) \tag{24}$$

  These dynamic updates in the search process enhance the exploration capacity of the first stages and lead to the improvement of the local search in the last stages [31] because the variation of the HMCR value affects the diversity and convergence of PHM, with similar results for the PAR value in terms of the local and global optimum [34,41,42]. Hence, the initial values of Equations (23) and (24) are adjusted according to [31].

3.4. Routing Phase

In the routing phase, when the clustering phase is done optimally and a total number of k CHs are determined, the algorithm enters the routing phase to transmit the collected data. The routing phase in the proposed method is multi-hop routing, so it is necessary to specify the RNs in the sink path. The CMs send the information to their CH based on the Time Division Medium Access (TDMA) schedule, and the CHs pass the route to the sink based on the RNs [27]. To achieve this, only PSO is implemented, and the fitness function starts to construct the routing tree [39]. The transfer route leads to the formation of a routing tree [27,39]. If the tree is optimized more, this will have an important effect on the proper and quick transfer of the information.

3.4.1. Enhanced Tree Encoding

For the fitness function in this section (minF_routing), we use Equation (7), mentioned above [18], to determine the relay nodes (RNs). Next, the algorithms update the particle locations and velocities in the PSO. The RNs are responsible for transferring the gathered data to the sink. Considering the sensitivity of the points involved in a disaster area compared to other points, the information of these points needs to be transferred quickly; in order to build the routing tree, the PSO is performed again via the Base Station (BS) [39].

It should be noted that, by updating the location and velocity, the PSO increases the number of invalid paths [27]. Consequently, due to application, the number of invalid paths must be reduced, and valid paths must be detected quickly. To achieve this goal, the priority encoding/decoding approach of PSO has been successfully tested [27,39]. However, it is necessary to develop the tree encoding appropriate to the disaster areas. Thus, the new packet format with respect to the fire flag is presented in Figure 2.

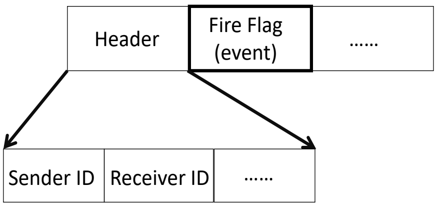


Figure 2. The new data packet format.

In Figure 3, a fire flag has been set up because the nodes are somehow aware of each other's status and make a more accurate decision regarding whether to send data to each other on the basis of events. In disaster areas, the fire flag can take two Boolean values of T as a non-uniform event (the disaster state) and F as a uniform event (the normal state). In order to validate paths from the CHs to the sink, nodes (particles) are mapped onto a random-priority number between $[-1, 1]$. The path construction procedure is shown for each CH in the following the Unified Modeling Language (UML) sequence diagram:

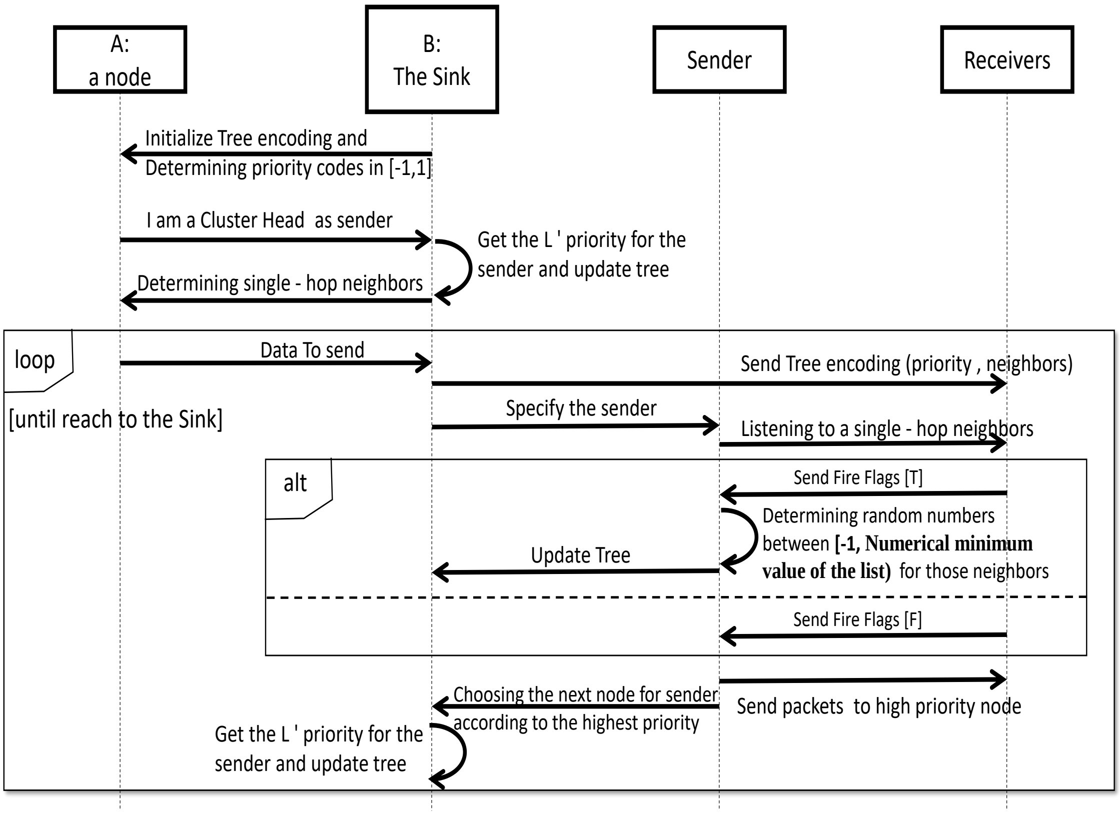

A: a node
B: The Sink
Sender
Receivers
Initialize Tree encoding and Determining priority codes in [-1,1]
I am a Cluster Head as sender
Get the L ' priority for the sender and update tree
Determining single - hop neighbors
loop
[until reach to the Sink]
Data To send
Send Tree encoding (priority , neighbors)
Specify the sender
Listening to a single - hop neighbors
alt
Send Fire Flags [T]
Determining random numbers between [-1, Numerical minimum value of the list) for those neighbors
Update Tree
Send Fire Flags [F]
Choosing the next node for sender according to the highest priority
Send packets to high priority node
Get the L ' priority for the sender and update tree

Figure 3. UML sequence diagram of tree encoding for each CH.

For clarity, the process of the construction routing tree with a simple example is shown below.

As shown in Figure 4, we have 15 nodes (with the sink and two CHs, 3 and 6) which are mapped onto the particles between [0, 14]. The sink has a tag of 0 with the highest priority H > 1, and nodes 5, 13 and 14 are in the fire range. The goal of all paths will be to reach the sink. For each CH, a list of 15 items is created based on the ID of the particles and their random priority. Next, each node, as the receiver identified by the sender, listens to its single-hop neighbors and chooses the next hop on the basis of the highest random priority generated between them.

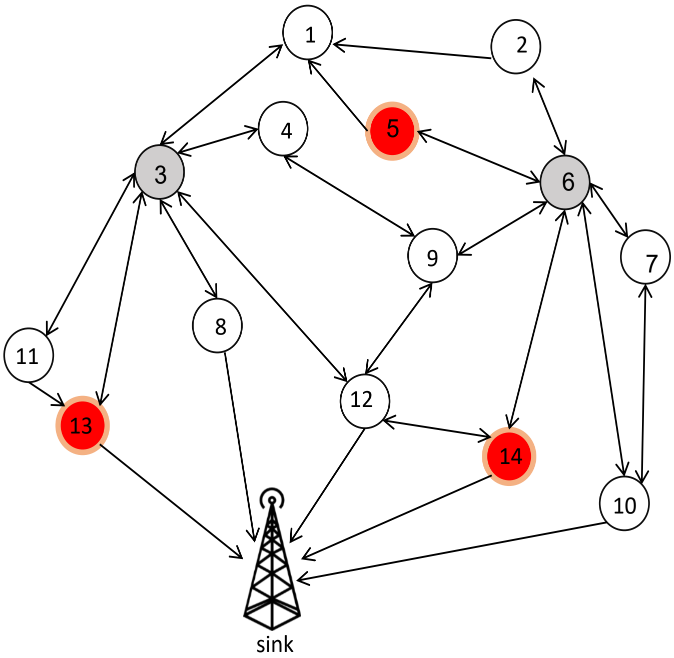


Figure 4. A random deployment of 15 nodes with two cluster heads of CH 3 and CH 6.

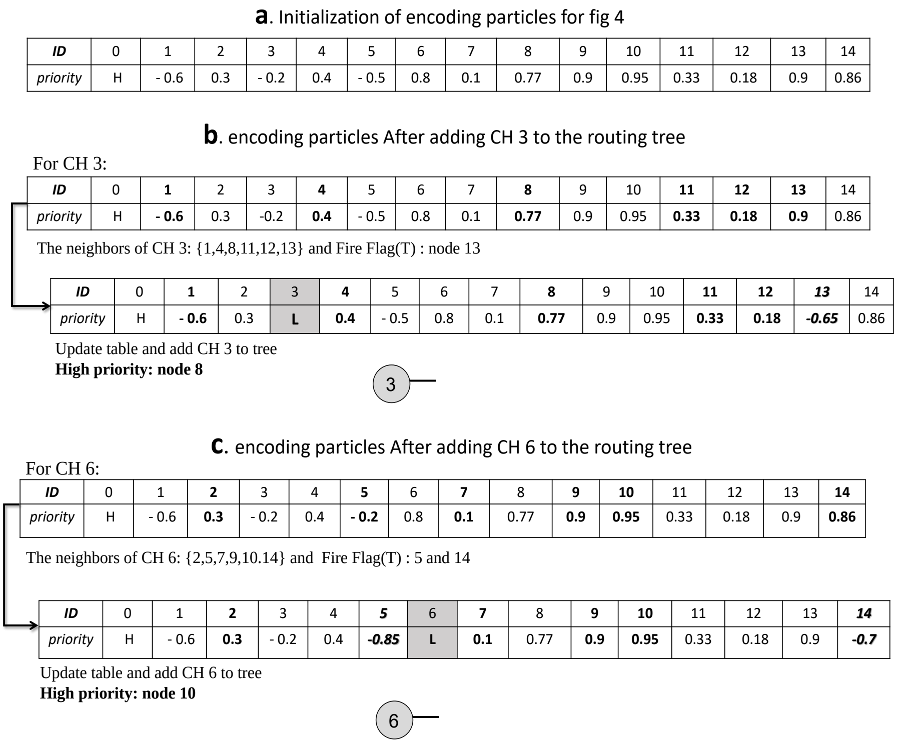
In the following, if the sender node takes a flag value of F when listening to its single-hop neighbors, then the next-hop is chosen based on the highest priorities of the single-hop neighbors. Otherwise, if the sender node takes the flag value of T, then the sender generates random priorities between $[-1, \textit{Numerical minimum value of the list}]$ for the single-hop neighbors (updating encoding particles). In this way, the packets try to pass through more reliable paths. Next, for the next hop chosen, the penalty L ($L < -1$) is set so that it does not participate in the selection process again. The path formation based on modified tree encoding for each CH (3 and 6) is illustrated in Figure 5.

As observed in Figure 5, the value of the fire flag is returned to T for CH 6 from nodes 5 and 14, and also for CH 3 from node 13. Thus, random priorities between $[-1, \textit{Numerical minimum value of the list}]$ are generated. Finally, choosing a single-hop neighbor based on the highest priority will construct the routing tree as shown in Figure 6.

According to Figures 5 and 6, paths $3 \rightarrow 8 \rightarrow 0$ and $6 \rightarrow 10 \rightarrow 0$ are generated, and then the routing tree and optimal CHs are broadcast to all nodes by the sink. After this phase, the sleep scheduling of nodes is considered. Additionally, if CHs, CMs, and RNs are unable to send any data, the sink will go to the sleep state and save energy.

3.5. Overall Proposed Algorithm

The main pseudo-codes of the proposed algorithm are formulated in Algorithm 2.



a. Initialization of encoding particles for fig 4

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
priority	H	- 0.6	0.3	- 0.2	0.4	- 0.5	0.8	0.1	0.77	0.9	0.95	0.33	0.18	0.9	0.86

b. encoding particles After adding CH 3 to the routing tree

For CH 3:

ID	0	**1**	2	3	**4**	5	6	7	**8**	9	10	**11**	**12**	**13**	14
priority	H	**- 0.6**	0.3	-0.2	**0.4**	- 0.5	0.8	0.1	**0.77**	0.9	0.95	**0.33**	**0.18**	**0.9**	0.86

The neighbors of CH 3: {1,4,8,11,12,13} and Fire Flag(T) : node 13

ID	0	**1**	2	3	**4**	5	6	7	**8**	9	10	**11**	**12**	***13***	14
priority	H	**- 0.6**	0.3	**L**	**0.4**	- 0.5	0.8	0.1	**0.77**	0.9	0.95	**0.33**	**0.18**	***-0.65***	0.86

Update table and add CH 3 to tree
High priority: node 8

c. encoding particles After adding CH 6 to the routing tree

For CH 6:

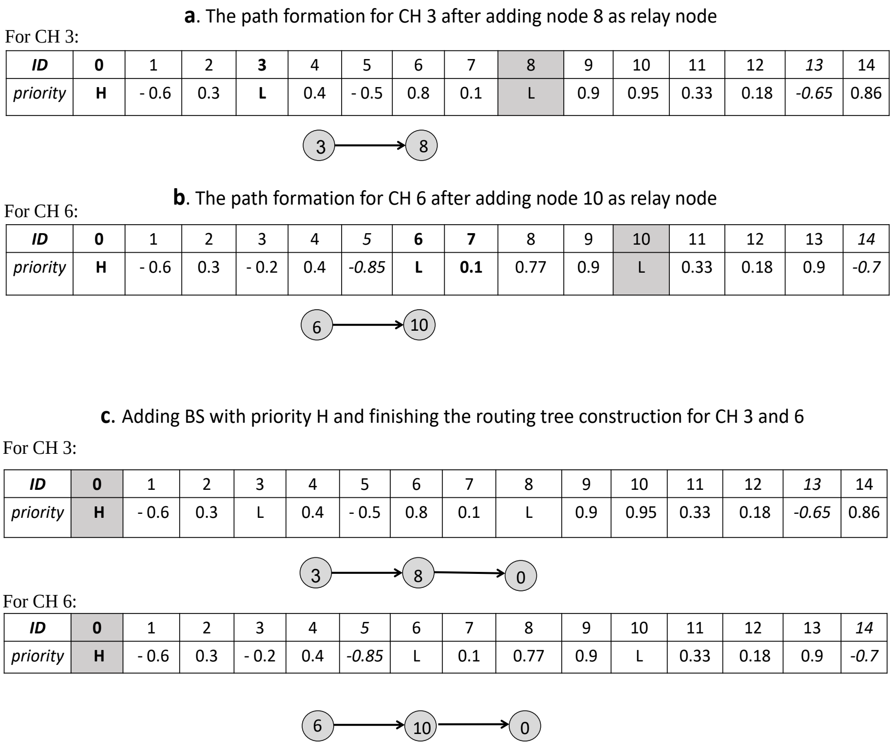
ID	0	1	**2**	3	4	**5**	6	**7**	8	**9**	**10**	11	12	13	**14**
priority	H	- 0.6	**0.3**	- 0.2	0.4	**- 0.2**	0.8	**0.1**	0.77	**0.9**	**0.95**	0.33	0.18	0.9	**0.86**

The neighbors of CH 6: {2,5,7,9,10.14} and Fire Flag(T) : 5 and 14

ID	0	1	**2**	3	4	***5***	6	**7**	8	**9**	**10**	11	12	13	***14***
priority	H	- 0.6	**0.3**	- 0.2	0.4	***-0.85***	L	**0.1**	0.77	**0.9**	**0.95**	0.33	0.18	0.9	***-0.7***

Update table and add CH 6 to tree
High priority: node 10

Figure 5. A simple example of modified tree encoding: (**a**) initialization of encoding particles for Figure 4 based on a random-priority number between [−1, 1], (**b**) encoding particles after adding CH 3 to the routing tree, and (**c**) encoding particles after adding CH 6 to the routing tree.



a. The path formation for CH 3 after adding node 8 as relay node

For CH 3:

ID	**0**	1	2	**3**	4	5	6	7	8	9	10	11	12	*13*	14
priority	**H**	- 0.6	0.3	**L**	0.4	- 0.5	0.8	0.1	L	0.9	0.95	0.33	0.18	*-0.65*	0.86

b. The path formation for CH 6 after adding node 10 as relay node

For CH 6:

ID	**0**	1	2	3	4	*5*	**6**	**7**	8	9	10	11	12	13	*14*
priority	**H**	- 0.6	0.3	- 0.2	0.4	*-0.85*	**L**	**0.1**	0.77	0.9	L	0.33	0.18	0.9	*-0.7*

c. Adding BS with priority H and finishing the routing tree construction for CH 3 and 6

For CH 3:

ID	**0**	1	2	3	4	5	6	7	8	9	10	11	12	*13*	14
priority	**H**	- 0.6	0.3	L	0.4	- 0.5	0.8	0.1	L	0.9	0.95	0.33	0.18	*-0.65*	0.86

For CH 6:

ID	**0**	1	2	3	4	*5*	6	7	8	9	10	11	12	13	*14*
priority	**H**	- 0.6	0.3	- 0.2	0.4	*-0.85*	L	0.1	0.77	0.9	L	0.33	0.18	0.9	*-0.7*

Figure 6. According to Figure 5, we continue the process of modified tree encoding: (**a**) the path formation for CH 3 after adding node 8 as a relay node (RN); (**b**) the path formation for CH 6 after adding node 10 as an RN; (**c**) adding BS with priority H and finishing the routing tree construction for CHs 3 and 6.

Algorithm 2: Overall proposed pseudo-code (ECHSR)

```
Input   :Max Round and initialize parameters
Output:k optimal CH + optimal routing tree
Round=1
bootstrapping (initialize parameters, set timers, broadcasting information)
while Max Round is not attained do
    Call Algorithm 3 as CH selection()      // The optimal combination of PSO and HSA
    if CHs recives all data from CMs based on TDMA then
        Call Algorithm 4 as Routing()                  // PSO based tree encoding
        broadcast optimal configuration (CHs + routing tree)
        set timers and send packets
    else
        if CHs, CMs and RNs do not have data to send then
            Go to Sleep mode and Save its Energy
        end
    end
    Max Round=Round+1;
end
```

Algorithm 3: The pseudo-code of CH-selection()

```
Result: k closest nodes to $g\_b$ will be the optimal CHs
Input   :network information and (PAR,HMCR,HMS,c1,c2,k,MaxIter)
Output:K cluster head
Initialize particles (random position and velocity)
Initialize HS parameters(par,hmcr,hms,c1,c2,maxiter)
Building PHM matrix based on Equation (19) (HMS and k)
Calculate $p\_best$ and $g\_best$ for each particles
while target fitness of clustering is not attained do
    foreach row of PHM do
        Calculate fitness function based Equation (18)
        if fitness(p) better than fitness(p_best) then
            $p\_best=p$
        else
            $p\_best=p\_best$.set best of $p\_bests$ as $g\_bests$
        end
    end
    while maximum iteration (MaxIter) is not attained do
        $I_x^n$=[$I'_1, I'_2, \ldots, I'_k$] as the New Harmony
        Call Algorithm 1 on PHM                // Improving Harmony Vectors for PHM
        if $f(I_x^n) > f(HV_x^{worst})$ then
            $I_x^n$ is included in matrix and $HV_x^{worst}$ of the PHM is excluded
        end
        Update dynamically of PAR and HMCR in Algorithm 1 based on Equations (23) and (24)
    end
    foreach row of PHM do
        update velocity based on Equation (20)
        update position based on Equation (21)
    end
end
```

Algorithm 4: The pseudo-code of Routing()

```
Result: A routing tree is built from the encoded particle for each CH
Input  : K cluster head and network information
Output: routing tree ending in sink
Initialize particles (random position and velocity) based on priority encoding
foreach CH as the source route do
    build a path from priority vector encoding/decoding    // as illustrated in UML 3
    if a valid path (reach to sink) then
        return fitness value calculated by Equation (7)
    else
        return penalty value (=0) as it fitness;
    end
end
Calculate p_best and g_best for each particles based on neighborhood topology
while target fitness of routing is not attained do
    Update velocity and positions of each particles.
    Evaluate fitness for each particle based on Equation (7)
    Update p_best for each particles (if current fitness better than p_best)
    Update g_best for each particle (choose best fitness value among all the neighbors)
end
```

4. Simulation Results and Evaluations

In this section, our proposed method (the so-called ECHSR) is compared with the state-of-the-art works on PSO–HSA [29] and TPSO-CR [27]. Furthermore, we implement the classical LEACH algorithm [22] as an example to show why we must transition to evolutionary algorithms. However, we performed the simulation in terms of the overall residual energy, the number of live nodes, network coverage, and the packet delivery ratio (PDR) with four different sink locations (the coordinates (50, 150), (100, 100), (0, 100) and (50, 200)) and also repeated each simulation 10 times with forest fire scenarios (Figure 7). Moreover, to evaluate efficiency, we examined our protocol with a uniform distribution of events using the metrics of the average residual energy and PDR. To begin the comparison, the simulation setup is explained, and then the simulation results are presented and analyzed.

4.1. Simulation Setup

MATLAB 2017a was used for the simulation. The system on which MATLAB was installed and the simulation processes were conducted had the following properties: CPU: Inter Core i7, memory: 8 GB RAM and operating system: Windows 10 Enterprise-64bit. The parameters and their values are presented in Table 2. For the AWS method, we assign the same weights to the parameters in the simulation [27,38], $\alpha_1 = \alpha_2 = \alpha_3 = 0.33$.

To bring the results closer to the real-world situation and make the theories more practical, it is necessary that the simulation environment for scientific research does not deviate greatly from reality. Many works, such as [13,20], have been conducted on forest fire modeling and have investigated how they burn. Given the simulation's capabilities, the computational complexity, and the difficulty of predicting the direction of fire, we decided to consider the speed and direction of the wind as being constant. By examining the previous works, we concluded that the fire burning model could be as shown in Figure 7; for instance, the NetLogo group implements forest fire models and confirms that fire spread can be random [47].

Table 2. Parameter settings for simulation.

N	AN	MR
N	Number of nodes	100
$m \times m$	Sensor field region	100 m × 200 m
E0	Initial energy of a nodes	0.5 (j)
K	Number of CHs	5
DP	Data packet length	4096 (bits)
DA	Energy data aggregation	5 (nJ)
[Xmin, Xmax]	Particle position	[0, 200]
[Vmin, Vmax]	Particle velocity	[0, 200] (m/s)
[c1min, c1max] and [c2min, c2max]	Acceleration constants	2
Eamp	Multipath fading transmitter amplifier energy	130 (pJ/bit/m2)
Eelec	Transmission and receiving energy	50 (nJ/bit)
Efs	Free space transmitter amplifier energy	10 (pJ/bit/m2)
Max Round	Maximum number of rounds	40 rounds
Sw	Swarm size	15
Iter	Number of iterations	5
HMCRmin and HMCRmax	Harmonic memory considering rate	0.8 and 0.9
PARmin and PARmax	Pitch adjustment rate	0.1 and 0.9
$\omega_{\max}$	Maximum values of inertia weight	0.9
$\omega_{\min}$	Minimum values of inertia weight	0.4
Base Station (BS)	Various sink locations	(0, 100), (100, 100), (50, 150), (50, 200)

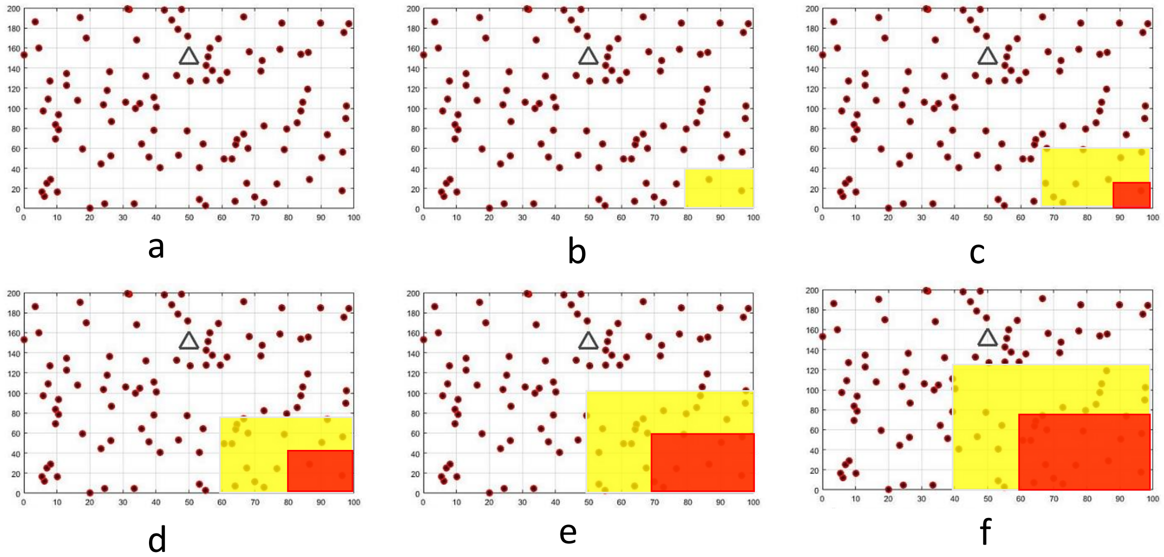


Figure 7. Simulation scenarios for forest fire in this work: (**a**) The normal state of the environment; (**b**) Fire smoke changes the state of the environment; (**c**) The fire starts in the forest; (**d**–**f**)— illustrates the spread of fire until 40 round of simulation.

In this work, a forest fire spreads from one direction and expands to the center [47]. To show the validity of the results, in addition to the scenario shown in Figure 7, we performed another simulation (without fire) in Section 4.3. In our random model, the yellow area indicates the non-uniformity of the events and red areas show the burned forest and annihilated nodes. As shown in Figure 7, the forest fire scenario includes different processes: (a) from round 1 to round 6, the simulation shows the normal state of the environment; (b) from round 6 to round 12, there is a non-uniform event such as fire smoke, which is indicated in yellow; (c) from round 12 to round 18, forest fires occur, which are shown in red. Similarly, for the rest of the scenario—(d), (e) and (f)—Figure 7 shows the growth of the forest fire proceeding to rounds 24, 30 and 40 of the simulation, leading to more dead nodes.

4.2. Simulation Results

In this section, the criteria for the comparison and analysis of the results are the overall residual energy, active nodes, network coverage and PDR ratio.

4.2.1. Overall Residual Energy

Considering the limitation of energy nodes, one of the goals of this paper was to present an energy-efficient algorithm. The decrease in the overall residual energy for all nodes leads to an increase in the network lifetime. This shows how the energy consumption rate in all nodes is distributed uniformly and optimally in different rounds. The residual energy is calculated as follows:

$$\sum_{n=1}^{N} E_{n,X} = E_{overall} \tag{25}$$

where $E_{n,X}$ is the overall energy consumption by a sensor node (n) after X rounds of data collection from the area-of-interest of the network, and $E_{overall}$ is determined as overall energy consumption by N number of total nodes after X number of simulation rounds of data collection in the network [42]. The mean of E_{total} until round X is defined as the average residual energy in this paper. According to Figures 8 and 9, although PSO–HSA [29], which is highly energy-efficient, is one of the most novel algorithms, this method could not compete with our method in forest fire scenarios (Figure 7).

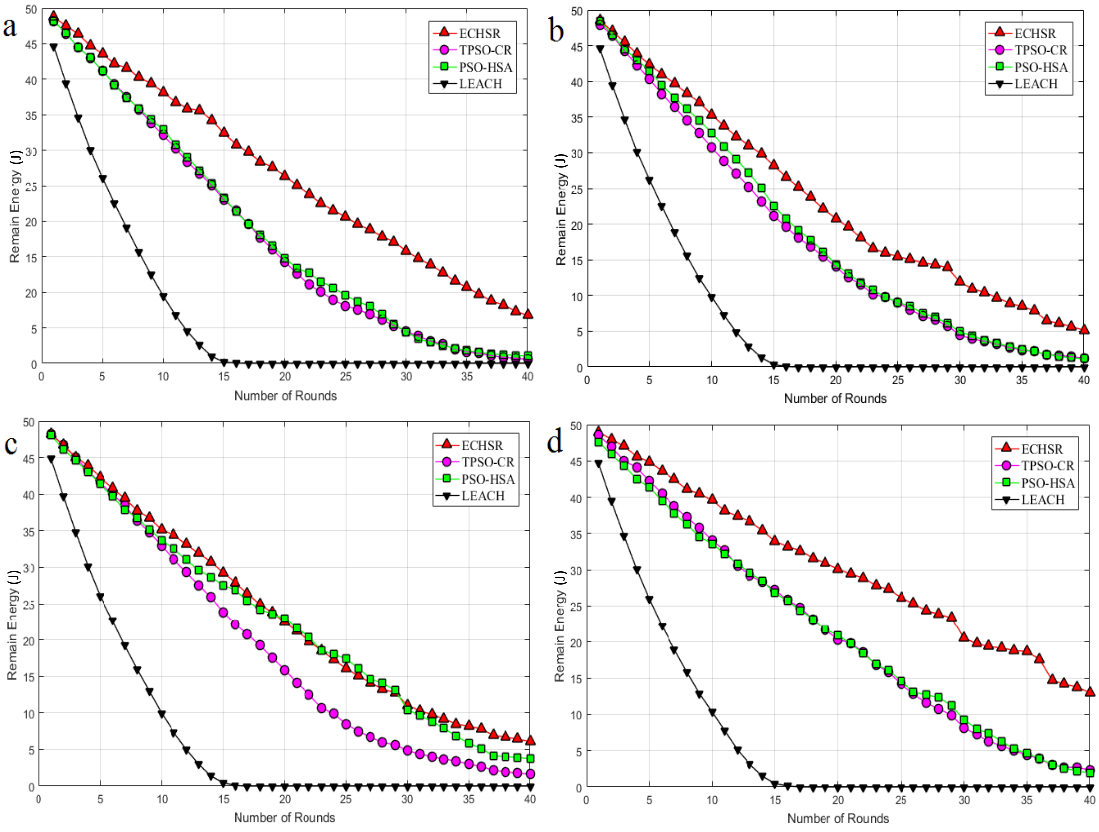


Figure 8. The residual energy for sink locations (**a**) (100, 100), (**b**) (0, 100), (**c**) (50, 150), and (**d**) (50, 200).

As shown in Figure 8, because of the random selection of CHs, the residual energy of LEACH drops to zero in the early rounds (around 15 rounds). Until 40 rounds of simulation have passed, it can also be seen in Figure 9 that PSO–HSA has a better average residual energy than TPSO-CR because the efficient searching of HS complements the dynamic capacity of PSO and also improves the balance between exploration and exploitation. In other words, the proposed method has better energy dissipation distributions than other methods. Additionally, the proposed method is improved in terms of the overall residual energy by about 1.04–1.42 times compared to PSO–HSA, by about 1.24–1.46 times compared to TPSO-CR and by about 1.70–1.77 times compared to LEACH for the different sink locations according to Figure 8. The reasons for this improvement are the optimum improvement of the structure of PSO–HSA and the attention paid to non-uniform events in the routing phase.

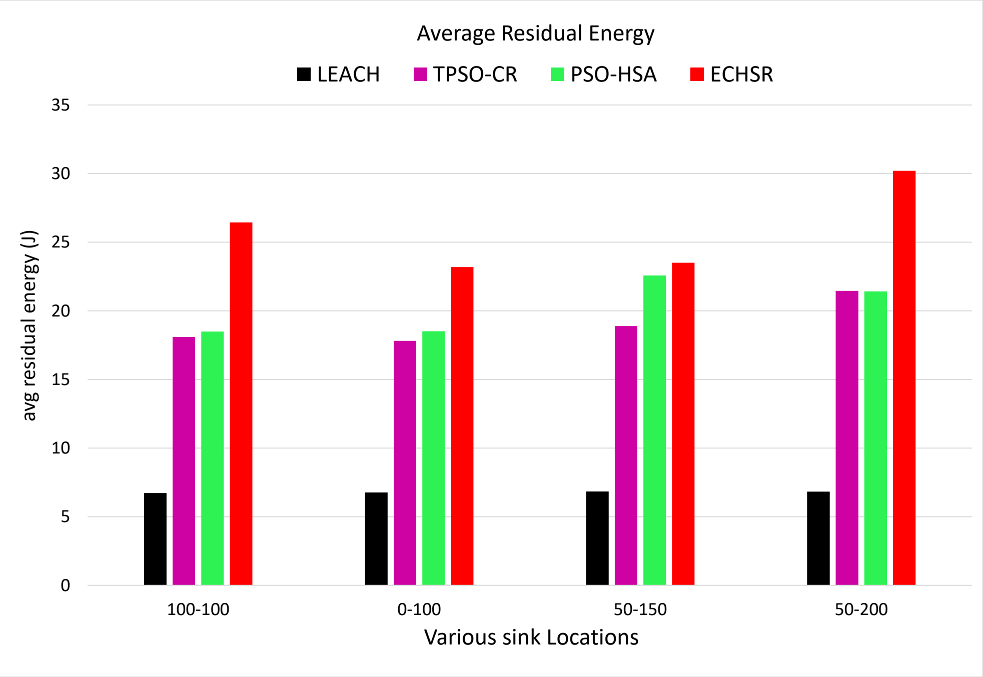


Figure 9. The average residual energy along with various sink locations.

4.2.2. Active Nodes

Another significant factor that directly affects the network lifetime is active nodes. Due to non-uniform events, the node energy involved in the disaster environments reaches zero after a while. The greater number of live nodes—or the reduction of the number of dead nodes—demonstrates that a better division of labor has taken place in terms of energy, and the CHs have optimally transferred the gathered data to the sink. In other words, the number of nodes increases the network lifetime that must be achieved to ensure network coverage. As shown in Figure 10, the proposed method in terms of active nodes outperforms PSO–HSA by about 16% to 36% and TPSO-CR by about 19% to 39% for the different sink locations. It can also be seen in Figure 10 that the LEACH algorithm also suffers as a result of the randomly chosen CHs in this application.

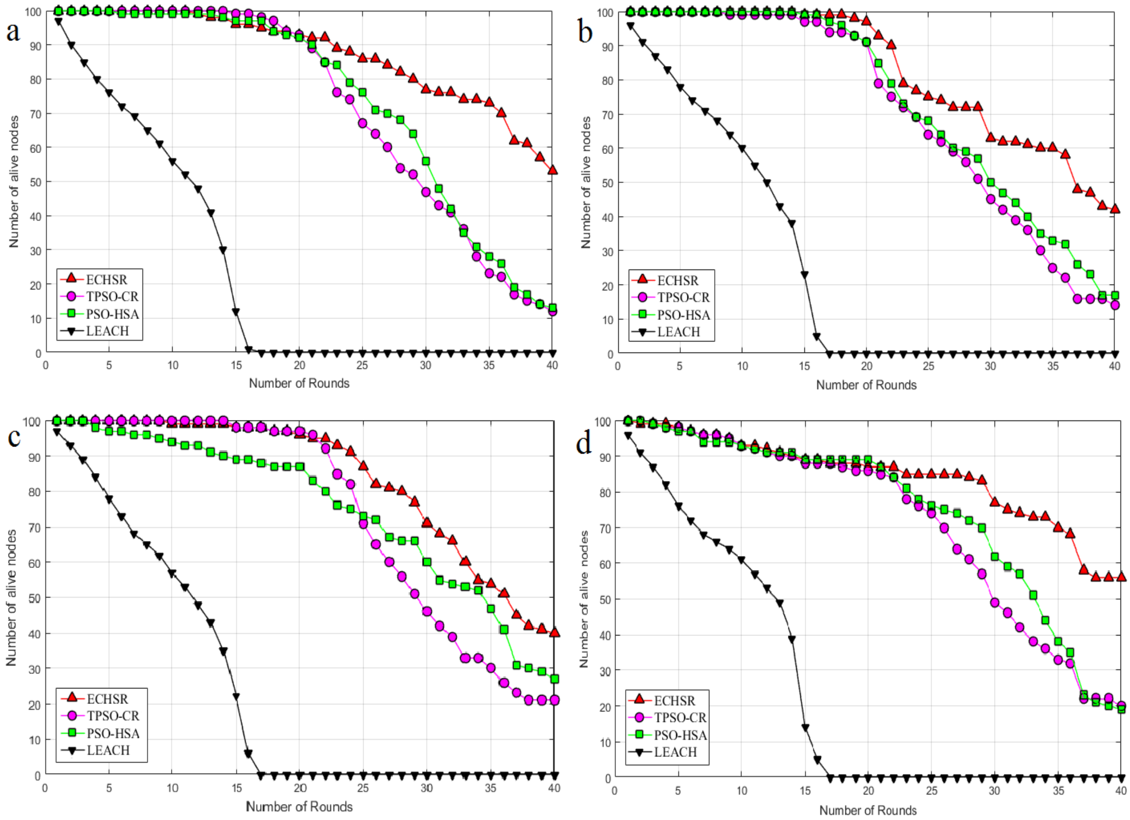


Figure 10. The number of live nodes for sink locations (**a**) (100, 100), (**b**) (0, 100), (**c**) (50, 150), and (**d**) (50, 200).

4.2.3. Network Coverage

One of the most important factors of the quality of service (QoS) of WSNs is network coverage. The selection of the best set of CHs requires maximum coverage. The maximum area coverage detected by WSN in forest fire application is necessary, because these areas might not be appropriate for the nodes due to their uniform random distribution. In other words, as mentioned before, the goal of this measure is to reduce the number of non-CH nodes. In order to evaluate the network coverage properly, an environment of 100 × 200 is graded in squares of 10 × 10 (dividing the area into smaller rectangles), as shown in Figure 11. Thus, the maximum number of graded points is 231, and the number of grids is 200. The goal of this division is to minimize the number of uncovered grids.

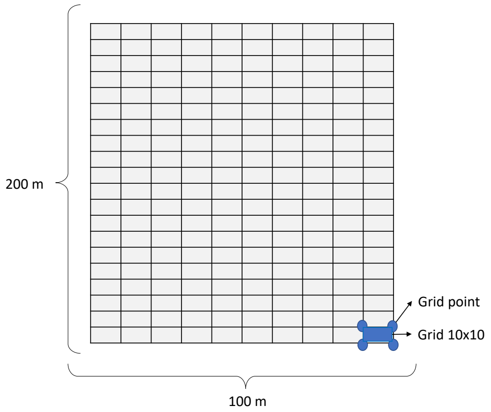


Figure 11. Dividing the area into smaller rectangles for coverage.

As can be seen in Figure 12, inspired by Figure 11, the proposed method outperforms PSO–HSA by about 1.08–1.32 times and TPSO-CR by about 1.18–1.34 times for the different sink locations. It is observed that the LEACH algorithm in round 16 has a network coverage of zero, which is much lower than the other methods. In general, the numerical results in Figure 12 demonstrate that the proposed method can cover the number of non-CHs more efficiently than PSO–HSA and TPSO-CR.

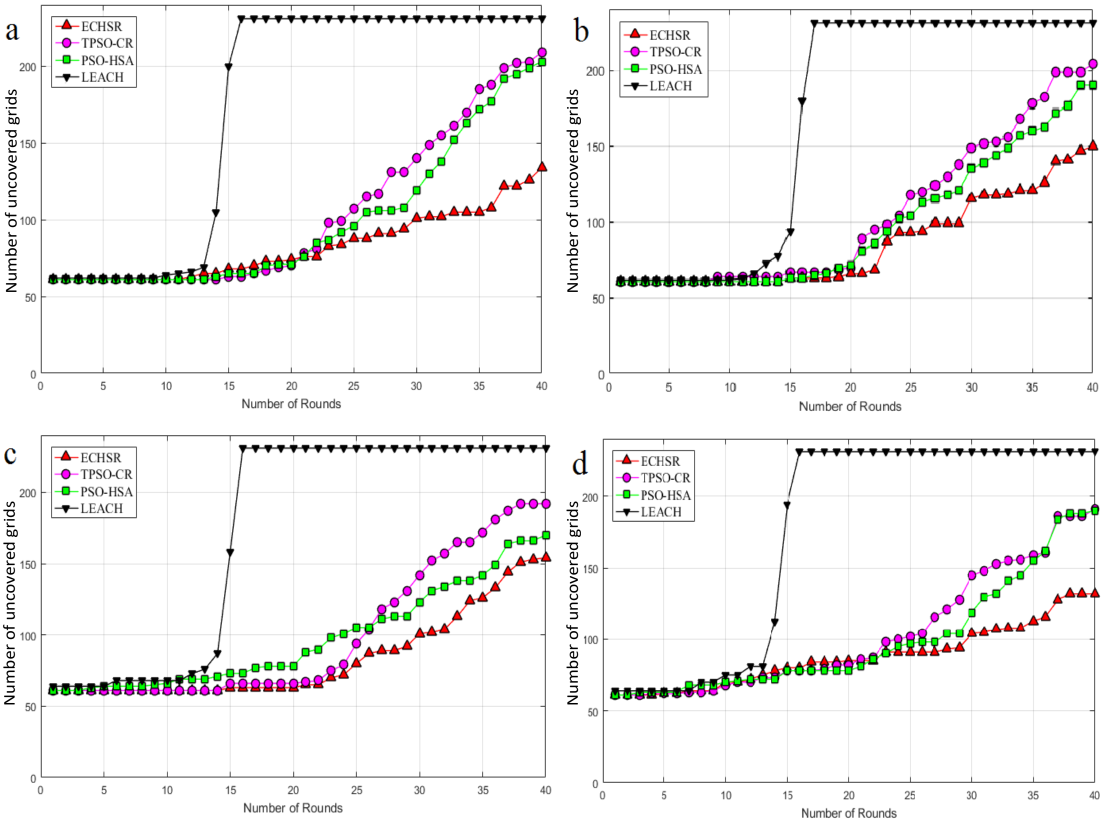


Figure 12. Network coverage for sink locations (**a**) (100, 100), (**b**) (0, 100), (**c**) (50, 150), and (**d**) (50, 200).

4.2.4. Packet Delivery Ratio

In the network domain, the ratio of data packets that the destination (sink) receives to the total packets generated by sources (nodes) is called the packet delivery ratio (PDR). The improvement of this factor leads to an increase in the number of active nodes. Furthermore, the greater the number of simulation rounds, the more PDR will be reduced. The PDR for each round of simulation is calculated with Equation (26):

$$\text{PDR} = \frac{\text{Number of packets received by the sink}}{\text{Total packets generated by nodes}} \tag{26}$$

As shown in Figure 13, considering that the worst link quality is removed by the fitness function (Equation (8)) and also that its routing tree is based on modified priority encoding, the proposed method has a higher average packet delivery ratio than other methods in 40 rounds of simulation. Furthermore, the numerical result demonstrates that PDR is increased in our method by about 1.07 to 1.38 times compared to PSO–HSA and by about 1.27 to 1.39 times compared to TPSO-CR for the different sink locations. It is clearly exhibited that LEACH has the lowest average PDR.

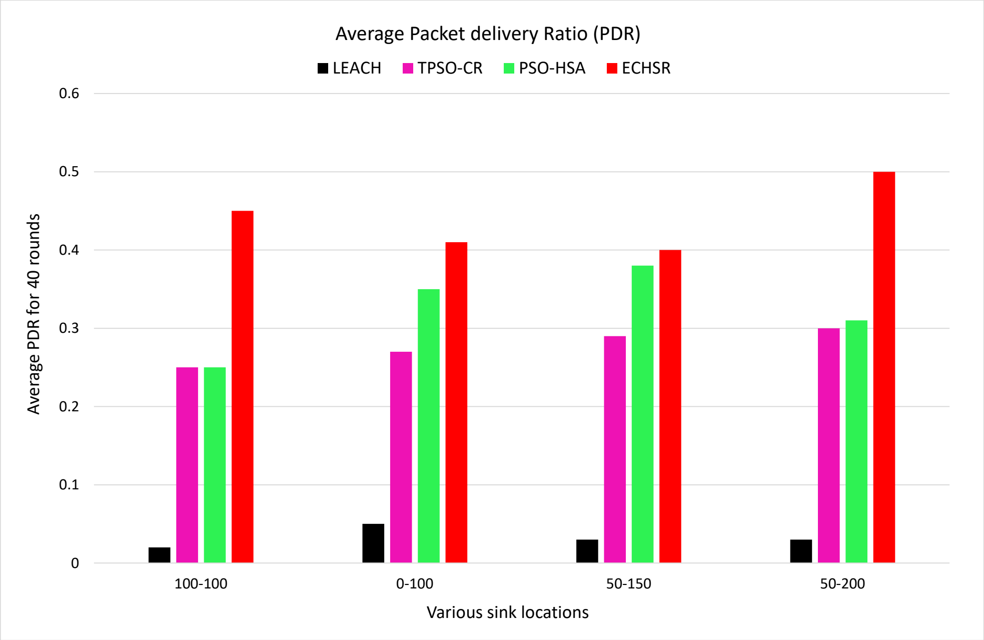


Figure 13. Packet delivery ratio (PDR) for various sink locations in forest fire scenarios and 40 rounds of the simulation.

4.3. Uniform Distribution of Events

We also tested the proposed method without a forest fire scenario (stable condition) in terms of the metrics of the average residual energy and packet delivery ratio (PDR) for various sink locations. In other words, across the entire network, events are uniformly distributed in each round of the simulations. According to the obtained results for the uniform distribution of events, the last node dies in the proposed method at about 1850 to 1900 rounds, which is superior to the state-of-the-art method [29], where this occurs at about 150 to 200 rounds. Up to 1900 rounds of the simulation, as shown in Figure 14, our protocol experiences the maximum average residual energy and consequently enhances the network lifetime compared to other methods.

Unlike the forest fire scenarios (Figure 7), with the changing position of the sink, there is no obvious difference in the efficiency of the algorithms. However, in order to evaluate the quality of clustering links and improvements in the packet transmission, we performed the average packet delivery ratio (PDR) analysis in Figure 15 based on various sink positions. Figure 15 clearly shows that the proposed method has the highest average PDR compared to other methods.

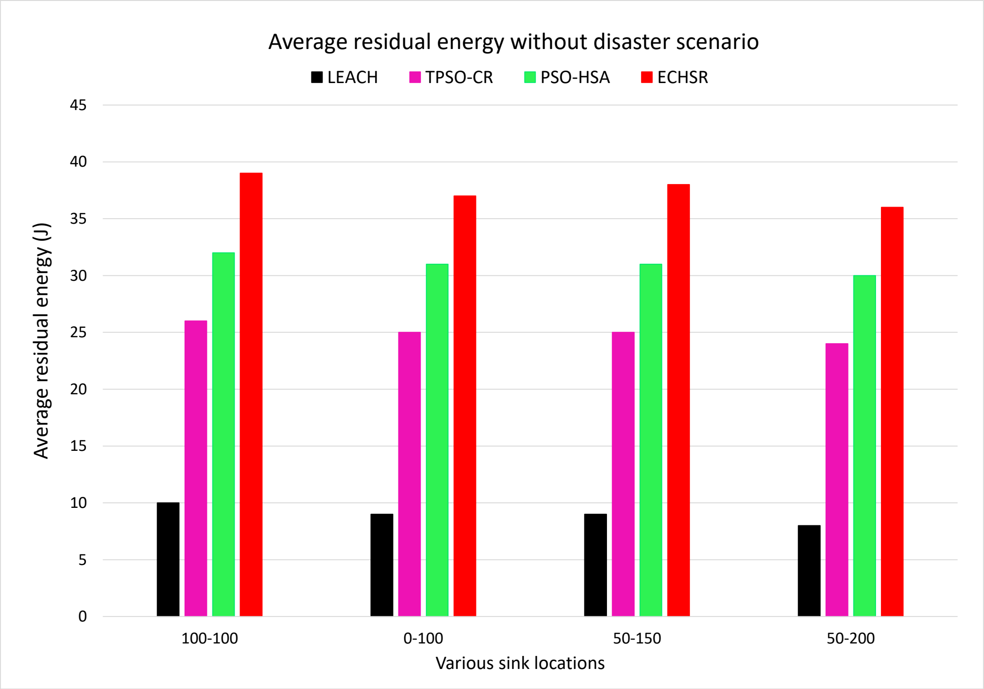


Figure 14. The average residual energy without forest fire scenarios for various sink locations.

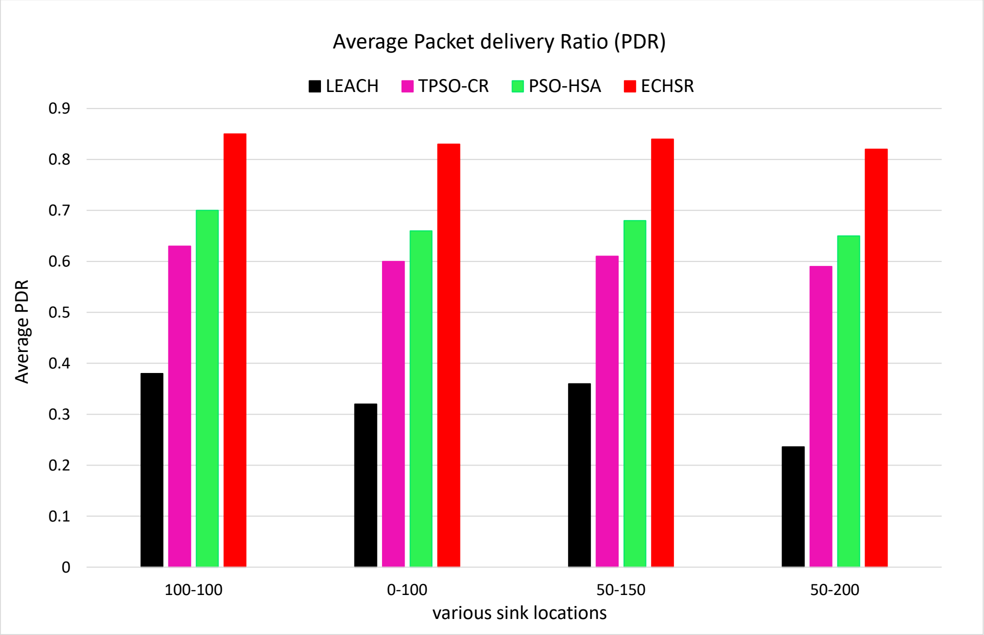


Figure 15. PDR without forest fire scenarios for various sink locations.

5. Conclusions and Future Work

In this work, energy-efficient cluster head selection and routing methods based on evolutionary algorithms were proposed for safety-critical WSN applications, focusing on forest fire. We have targeted energy efficiency as a major issue in such IoT networks. To overcome this challenge, an improved hybrid PSO–HSA-based CH selection technique with a novel fitness function including energy-efficiency, cluster closeness, and network coverage criteria were proposed. We employed PSO to select RNs with a fitness function including energy-efficiency and communication link quality, which resulted in the development of a modified tree encoding method based on a new data packet format. We validated the efficiency of the proposed method for a forest fire scenario compared with the related works, considering different sink node placements. By considering different performance metrics, such as the overall residual energy, the number of live nodes, network coverage, and the packet delivery ratio in simulation evaluations, we could make the following conclusions: (1) the proposed algorithm in a disaster area is much better than the state-of-the-art methods [27,29]; (2) the placement of the sink node affects the performance of the algorithm; (3) since the variation of HMCR and PAR parameters affect the fast convergence of the algorithms, it appears that these parameters may change dynamically; (4) probabilistic algorithms such as LEACH [22] cannot perform properly in a disaster application; and (5) in scenarios where events occur uniformly, our algorithm improves the network lifetime compared with other methods.

In future work, we are planning to design efficient MAC protocols based on this application to prevent long delays and extend simulation scenarios, especially for heterogeneous networks.

Author Contributions: Conceptualization, Software, Methodology and Writing—original draft: M.B.; Validation: M.B., H.F. and N.Y.; Visualization: M.B. and H.F.; Supervision: H.F. and N.Y.; Writing—review and editing: H.F. and N.Y.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Swedish Foundation for Strategic Research via the FiC project, and by the Swedish Research Council (Vetenskapsrådet) through the MobiFog starting grant, and by the Swedish Knowledge Foundation (KKS) through the FlexiHealth Prospekt, and the EU Celtic_Next/Vinnova project, Health5G (Future eHealth powered by 5G).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dhanvijay, M.M.; Patil, S.C. Internet of Things: A survey of enabling technologies in healthcare and its applications. *Comput. Netw.* **2019**, *153*, 113–131. [CrossRef]
2. Qiu, T.; Li, B.; Zhou, X.; Song, H.; Lee, I.; Lloret, J. A Novel Shortcut Addition Algorithm with Particle Swarm for Multi-sink Internet of Things. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3566–3577. [CrossRef]
3. Cai, S.; Zhu, Y.; Wang, T.; Xu, G.; Liu, A.; Liu, X. Data collection in underwater sensor networks based on mobile edge computing. *IEEE Access* **2019**, *7*, 65357–65367. [CrossRef]
4. Hao, Y.; Chen, M.; Hu, L.; Hossain, M.S.; Ghoneim, A. Energy efficient task caching and offloading for mobile edge computing. *IEEE Access* **2018**, *6*, 11365–11373. [CrossRef]
5. Akkaya, K.; Younis, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **2005**, *3*, 325–349. [CrossRef]
6. Hussain, I.; Ullah, M.; Ullah, I.; Bibi, A.; Naeem, M.; Singh, M. Optimizing Energy Consumption in the Home Energy Management System via a Bio-Inspired Dragonfly Algorithm and the Genetic Algorithm. *Electronics* **2020**, *9*, 406. [CrossRef]
7. Khan, I.; Singh, D. Energy-balance node-selection algorithm for heterogeneous wireless sensor networks. *ETRI J.* **2018**, *40*, 604–612. [CrossRef]
8. Rawat, P.; Singh, K.D.; Chaouchi, H.; Bonnin, J.M. Wireless sensor networks: a survey on recent developments and potential synergies. *J. Supercomput.* **2014**, *68*, 1–48. [CrossRef]
9. Araghizadeh, M.A.; Teymoori, P.; Yazdani, N.; Safari, S. An efficient medium access control protocol for WSN-UAV. *Ad Hoc Netw.* **2016**, *52*, 146–159. [CrossRef]
10. Johnson, D.B.; Maltz, D.A. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*; Springer: Boston, MA, USA, 1996; pp. 153–181.
11. Shen, H.; Bai, G. Routing in wireless multimedia sensor networks: A survey and challenges ahead. *J. Netw. Comput. Appl.* **2016**, *71*, 30–49. [CrossRef]
12. Jha, R.; Singh, A.; Tewari, A.; Shrivastava, P. Performance analysis of disaster management using WSN technology. *Procedia Comput. Sci.* **2015**, *49*, 162–169. [CrossRef]
13. Hefeeda, M.; Bagheri, M. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc Sens. Wirel. Netw.* **2009**, *7*, 169–224.
14. Avesani, P.; Perini, A.; Ricci, F. Interactive case-based planning for forest fire management. *Appl. Intell.* **2000**, *13*, 41–57. [CrossRef]

15. Priyadarsini, P.U.; Sriramya, P. Disaster management using evidence-based interactive trust management system for wireless sensor networks by Internet of Things. *Comput. Electr. Eng.* **2019**, *75*, 164–174. [CrossRef]
16. Yu, L.; Wang, N.; Meng, X. Real-time forest fire detection with wireless sensor networks. In Proceedings of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 26 September 2005; Volume 2, pp. 1214–1217.
17. Molina-Pico, A.; Cuesta-Frau, D.; Araujo, A.; Alejandre, J.; Rozas, A. Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network. *J. Sens.* **2016**, *2016*, 8325845 . [CrossRef]
18. Yuan, C.; Zhang, Y.; Liu, Z. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Can. J. For. Res.* **2015**, *45*, 783–792. [CrossRef]
19. Aslan, Y.E.; Korpeoglu, I.; Ulusoy, Ö. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Comput. Environ. Urban Syst.* **2012**, *36*, 614–625. [CrossRef]
20. Karafyllidis, I. Design of a dedicated parallel processor for the prediction of forest fire spreading using cellular automata and genetic algorithms. *Eng. Appl. Artif. Intell.* **2004**, *17*, 19–36. [CrossRef]
21. Alkhatib, A.A. A review on forest fire detection techniques. *IInt. J. Distrib. Sens. Netw.* **2014**, *10*, 597368. [CrossRef]
22. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7 January 2000; p. 10.
23. Singh, B.; Lobiyal, D.K. A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks. *Hum. Cent. Comput. Inf. Sci.* **2012**, *2*, 13. [CrossRef]
24. Zungeru, A.M.; Ang, L.M.; Seng, K.P. Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison. *J. Netw. Comput. Appl.* **2012**, *35*, 1508–1536. [CrossRef]
25. Krishnan, M.; Yun, S.; Jung, Y.M. Dynamic clustering approach with ACO-based mobile sink for data collection in WSNs. *Wirel. Netw.* **2019**, *25*, 4859–4871. [CrossRef]
26. Han, T.; Zhang, L.; Pirbhulal, S.; Wu, W.; de Albuquerque, V.H.C. A novel cluster head selection technique for edge-computing based IoMT systems. *Comput. Netw.* **2019**, *158*, 114–122. [CrossRef]
27. Elhabyan, R.S.; Yagoub, M.C. Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network. *J. Netw. Comput. Appl.* **2015**, *52*, 116–128. [CrossRef]
28. Rahimi, S.; Abdollahpouri, A.; Moradi, P. A multi-objective particle swarm optimization algorithm for community detection in complex networks. *Swarm Evol. Comput.* **2018**, *39*, 297–309. [CrossRef]
29. Shankar, T.; Shanmugavel, S.; Rajesh, A. Hybrid HSA and PSO algorithm for energy efficient cluster head selection in wireless sensor networks. *Swarm Evol. Comput.* **2016**, *30*, 1–10. [CrossRef]
30. Guleria, K.; Verma, A.K. Comprehensive review for energy efficient hierarchical routing protocols on wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 1159–1183. [CrossRef]
31. Wang, L.; Hu, H.; Liu, R.; Zhou, X. An improved differential harmony search algorithm for function optimization problems. *Soft Comput.* **2019**, *23*, 4827–4852. [CrossRef]
32. Gumaida, B.F.; Luo, J. A hybrid particle swarm optimization with a variable neighborhood search for the localization enhancement in wireless sensor networks. *Appl. Intell.* **2019**, *49*, 3539–3557. [CrossRef]
33. Bongale, A.M.; Nirmala, C.; Bongale, A.M. Hybrid Cluster Head Election for WSN Based on Firefly and Harmony Search Algorithms. *Wirel. Pers. Commun.* **2019**, *106*, 275–306. [CrossRef]
34. Sarkhel, R.; Das, N.; Saha, A.K.; Nasipuri, M. An improved harmony search algorithm embedded with a novel piecewise opposition based learning algorithm. *Eng. Appl. Artif. Intell.* **2018**, *67*, 317–330. [CrossRef]
35. Khalesian, M.; Delavar, M.R. Wireless sensors deployment optimization using a constrained Pareto-based multi-objective evolutionary approach. *Eng. Appl. Artif. Intell.* **2016**, *53*, 126–139. [CrossRef]
36. Handl, J.; Knowles, J. An evolutionary approach to multiobjective clustering. *IEEE Trans. Evol. Comput.* **2007**, *11*, 56–76. [CrossRef]
37. Grodzevich, O.; Romanko, O. Normalization and Other Topics in Multi-Objective Optimization 2006. Available online: http://www.maths-in-industry.org/miis/233/1/fmipw1-6.pdf (accessed on 2 May 2020)
38. Zhang, H.; Wang, R.; Liu, H.; Luo, H.; Liu, Y. MCDMSR: multicriteria decision making selection/replacement based on agility strategy for real optimization problems. *Appl. Intell.* **2019**, *49*, 2918–2941. [CrossRef]
39. Mohemmed, A.W.; Sahoo, N.C.; Geok, T.K. Solving shortest path problem using particle swarm optimization. *Appl. Soft Comput.* **2008**, *8*, 1643–1653. [CrossRef]

40. Manjeshwar, A.; Agrawal, D.P. TEEN: ARouting Protocol for Enhanced Efficiency in Wireless Sensor Networks. Available online: http://www.nhu.edu.tw/~cmwu/Lab/TEEN.pdf (accessed on 2 May 2020)
41. Hoang, D.; Yadav, P.; Kumar, R.; Panda, S. A robust harmony search algorithm based clustering protocol for wireless sensor networks. In Proceedings of the 2010 IEEE international conference on communications workshops, Capetown, South Africa, 23–27 May 2010; pp. 1–5.
42. Gupta, G.P.; Jha, S. Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques. *Eng. Appl. Artif. Intell.* **2018**, *68*, 101–109. [CrossRef]
43. Khatarkar, S.; Kamble, R. Wireless sensor network MAC protocol: SMAC & TMAC. *Indian J. Comput. Sci. Eng.* **2013**, *4*, 304–310.
44. Armano, G.; Farmani, M.R. Multiobjective clustering analysis using particle swarm optimization. *Expert Syst. Appl.* **2016**, *55*, 184–193. [CrossRef]
45. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 3, pp. 1945–1950.
46. Yang, C.; Gao, W.; Liu, N.; Song, C. Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight. *Appl. Soft Comput.* **2015**, *29*, 386–394. [CrossRef]
47. Wilensky, U. Netlogo. (Center for Connected Learning and Computer-Based Modeling.) 1999. Available online: http://ccl.northwestern.edu/netlogo/ (accessed on 2 May 2020)