computers

Article

Rendezvous Based Adaptive Path Construction for Mobile Sink in WSNs Using Fuzzy Logic

Omar Banimelhem * and **Fidaa Al-Quran**

Network Engineering and Security Department, Jordan University of Science and Technology, P.O. Box 3030, Irbid 22110, Jordan; fmalquran17@cit.just.edu.jo
* Correspondence: omelhem@just.edu.jo; Tel.: +962-2720-1000 (ext. 22465)

Abstract: In this paper, an adaptive path construction approach for Mobile Sink (MS) in wireless sensor networks (WSNs) for data gathering has been proposed. The path is constructed based on selecting Rendezvous Points (RPs) in the sensing field where the MS stops in order to collect the data. Compared with the most existing RP-based schemes, which rely on fixed RPs to construct the path where these points will stay fixed during the whole network lifetime, we propose an adaptive path construction where the locations of the RPs are dynamically updated using a Fuzzy Inference System (FIS). The proposed FIS, which is named Fuzzy_RPs, has three inputs and one output. The inputs are: the remaining energy of the sensor nodes, the transmission distance between the RPs and the sensor nodes, and the number of surrounding neighbors of each node. The output of FIS is a weight value for each sensor node generated based on the previous three parameters and, thus, each RP is updated to its new location accordingly. Simulation results have shown that the proposed approach extends the network lifetime compared with another existing approach that uses fixed RPs. For example, in terms of using the first dead node as a metric for the network lifetime, when the number of deployed sensor nodes changes from 150 to 300, an improvement that ranges from 48.3% to 83.76% has been achieved compared with another related approach that uses fixed RPs.

Keywords: WSN; mobile sinks; rendezvous points; fuzzy inference system; IoT; network lifetime

check for updates

Citation: Banimelhem, O.; Al-Quran, F. Rendezvous Based Adaptive Path Construction for Mobile Sink in WSNs Using Fuzzy Logic. *Computers* **2023**, *12*, 66. https://doi.org/10.3390/computers12030066

Academic Editor: Paolo Bellavista

Received: 21 December 2022
Revised: 22 January 2023
Accepted: 15 March 2023
Published: 20 March 2023

1. Introduction

With the continuous advances in information technology accompanied by the introduction of Internet of Things (IoT) applications, Wireless Sensor Networks (WSNs), which are the core of the IoT-based systems, are now a main player in our daily lives [1]. In WSNs, the data that is generated by the sensor nodes is usually collected by a central node called the sink or the base station (BS). Traditional WSNs with a fixed BS pose a primary limitation, which is called an energy hole problem, and which results as a consequence of depleting the energy of the nodes close to the sink rapidly because of the overwhelming traffic transmitted by other sensor nodes far from the sink. This could result in affecting the WSN efficiency, such as partitioning the network into unconnected areas. Indeed, partitioning the network is considered a crucial issue, especially in large-scale networks.

Recently, the widespread use of mobile robots has opened the door to leverage mobile sinks in WSNs for performing many tasks, such as employing them as data collectors in WSNs [2]. In fact, using mobile sinks (MS) can significantly increase the balance between the nodes and thus extend the network lifetime. Furthermore, it can increase the coverage of the network by reaching isolated uncovered areas. However, in any mobile sink-based approach, the balance between power consumption and the delay of data collection is still a real challenge and an active research area [3]. Different schemes of moving sinks have been proposed by the research community. These schemes can be divided into three categories: random, controlled, and predefined movement strategies [4]. In random movement methods, the MS follows a random path in its movement to collect the data. The main limitation of using random movement methods is the buffer overflow problem. Moreover, uncontrolled movements extend the overhead for each node until finding the location of the new position of the mobile sink and, thus, increase the ratio of dropped packets. In controlled schemes, the speed and direction of the next MS destination are determined according to the network situation. For example, the areas where the sensor nodes have urgent information will be given priority over the other locations. In a predefined movement, the MS moves according to a known fixed or dynamic path, which is generated dynamically as a function of network parameters such as the energy of the sensor nodes and their locations. To reduce the energy consumption in predefined movement methods, MS visits each location near sensor nodes and therefore yields more energy balance. However, visiting each sensor node will pose delay limitations, such as a long path length as well as increasing the delay of data delivery.

To address this issue, Rendezvous Points (RPs) schemes have been proposed [5,6]. In RP-based schemes, specific locations are chosen in order to reduce data gathering delays and balance energy consumption amongst nodes. The challenging task in such schemes is how to select the suitable RPs that satisfy both the energy consumption and the delay requirements. Indeed, RPs influence the path that an MS will follow to gather the data from the sensor nodes. Typically, the path of the MS is established by applying the traveling salesman problem (TSP) on the chosen RPs. Hence, the position and the number of RPs play a major role in constructing a path, which balances energy consumption and delay requirements. Figure 1 shows an example of a WSN where the RP-based model is used for data collection by the MS. However, selecting fixed RPs during the whole network operation results in an imbalance of energy consumption among the nodes that are associated with the same RP and thus reduce the network lifetime. Therefore, updating the locations of the RPs dynamically during the network operation will improve the performance of the network efficiently. In this paper, an adaptive path construction approach is proposed, where the locations of the RPs are dynamically updated using a Fuzzy Inference System (FIS). Three inputs are used to determine the updated RPs' locations. These inputs are: the remaining energy of the sensor nodes, the transmission distance between the RPs and the sensor nodes, and the number of surrounding neighbors of each node.

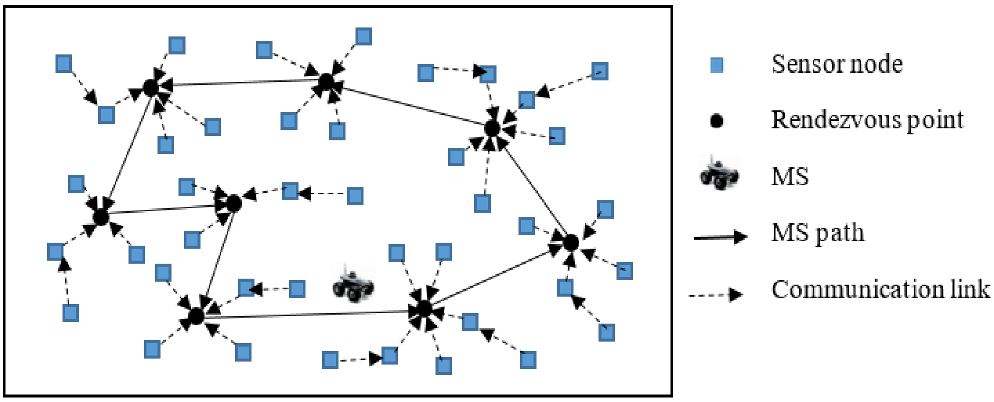

Figure 1. An example of a WSN where RP-based model is used for data collection by MS.

The remainder of the paper is structured as follows. The related work is presented in Section 2. The system model of the proposed approach is presented in Section 3. The proposed approach is discussed in Section 4. In Section 5, we present and discuss the simulation results, and the paper is concluded in Section 6.

2. Related Work

Several studies on random or controlled-based schemes for MSs have been conducted in WSNs ([7,8] are examples on random movement and [9,10] are examples of controlled-based movements). However, in this section, we address RP-based solutions that present a tradeoff between the random and controlled-based schemes in terms of the buffer flow problem in random movement schemes and the long delay problem in controlled-based schemes.

Park et al. [11] presented an approach in which the mobile sink moves along a fixed path and stops at a number of locations for collecting the data. The number of stop points over the path is selected using the Tabu search algorithm where the objective is minimizing the number of hop count from the sensor nodes to the mobile sink. Two algorithms called reduced k-means (RkM) and delay bound RkM (DBRkM) were proposed by Kaswan et al. [12] for generating a set of RPs that will be visited by the MS. The MS will then move over a fixed path that connects the selected RPs to gather the data from the sensor nodes. Banimelhem et al. [13] proposed an algorithm to generate a fixed path for the MS using principal component analysis (PCA), where the data can be gathered using either direct or multi-hop data transmission modes. A rendezvous-based routing protocol (RRP) was proposed by Sharma et al. [14] to address the need for energy efficiency and lower end-to-end latency. In the RRP, a rendezvous region is created in the middle of the network where the nodes, called backbone nodes, in this region form a tree, and where the other nodes communicate with the rendezvous region. Gupta et al. [6] proposed a routing method in WSN that depends on RPs and multiple MSs. At the beginning, the sensor nodes are distributed into a set of clusters using mean shift clustering (MSC). A cluster head (CH) for each cluster is then selected using the Bald Eagle Search (BES) algorithm. After that, the authors used the hybrid seagull optimization and salp swarm (SOSS) algorithm in order to find the RPs and the travelling route of each mobile sink in the network.

Vajdi et al. [15] proposed an approach that chooses a group of RPs outside the predetermined trajectory such that the defined path can accomplish the goals of minimizing sensor node energy consumption and decreasing network average data delivery time. Raj et al. [16] proposed an approach that builds a reliable and smart route for the mobile sink utilizing game theory and improves ACO-based MS route selection and the Data Gathering (GTAC-DG) algorithm. A set of rendezvous points (RPs) is selected to construct the path for the MS using an ACO-based algorithm. The GTAC-DG algorithm is used to create a path for the MS. Boyineni et al. [17] proposed an approach called the ant colony optimization (ACO)-based mechanism (ACO-RMS) for selecting the RPs and scheduling the mobile sink in the event-driven WSNs. The load of each sensor in the ACO-RMS approach is initially identified using a spanning tree. Different factors, such as distance, remaining energy, and total packets generated by the sensor nodes at a particular time, are used for the RPs' selection and path use. Donta et al. [18] proposed an extended ant colony optimization (ACO)-based MS path construction for event-driven WSNs, where the ACO algorithm selects the best set of RPs and the path that the MS will travel between these RPs.

Ghaleb et al. [19] proposed an approach where RPs are used to collect the data using data compression techniques from nearby sources and then send the data to the mobile sink when it travels over the path connecting these RPs. Furthermore, the authors proposed an algorithm called a minimal constrained rendezvous point (MCRP), which ensures that the collected data are relayed to the RPs taking into consideration three constraints: RPs' locations, bounded relay hop, and number of nearby sources.

3. System Model

In this paper, we propose an adaptive data collection scheme for homogeneous WSNs, where a set of sensor nodes S are deployed randomly in the sensing area. A single MS will move in the network area to collect the sensed data. We assume that the MS moves at a constant speed and it stops at RPs to collect the data. The stopping duration at each RP is assumed to be fixed and enough to collect the data from all corresponding sensors. For energy consumption, the energy model proposed in [20] is assumed. Using this model, the energy consumed (E_T) to transmit a k_b-bit packet between two sensor nodes s_1 at location (x_1, y_1) and sensor node s_2 at location at (x_2, y_2) is calculated as:

$$E_T = \left\{ \begin{array}{ll} k_b E_{elec} + k_b \varepsilon_{fs} d^2, & d < d_0 \\ k_b E_{elec} + k_b \varepsilon_{amp} d^4, & d \geq d_0 \end{array} \right\} \tag{1}$$

and the energy consumed by sensor node s_2 to receive the k_b-bit packet (E_R) is calculated as:

$$E_R = k_b\, E_{elec} \tag{2}$$

where E_{elec} is the electronics energy; ε_{fs} and ε_{amp} denote the amplifier energy that depends on the required receiver sensitivity and the receiver noise figure, respectively; and d is the distance between sensor nodes s_1 and s_2 and is given as:

$$d = \sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2} \tag{3}$$

The notations that are used in this paper are shown in Table 1.

Table 1. Notations used.

Term	Definition
S	Set of sensor nodes
s_i	Sensor node i
S_j	Set of sensor nodes associated with RP j
P	Set of potential RPs
C	Set of RPs that are used to build the MS path
$S_{RP}(i)$	Number of sensor nodes associated with RP i
$H_{RP}(i)$	Average hop distance between RP i and the sensor nodes associated with it
(X_{RPi}, Y_{RPi})	Location of RP_i
E_i	Remaining energy of sensor node i
$d_{i \to RPi}$	The distance between sensor node i and its corresponding RP
NBi	The 1-hop neighbors of sensor node i
PL	Path length
v	MS speed

4. Proposed Approach

In this section, we discuss the proposed approach for building a dynamic path for the mobile sink to collect the data from the sensor nodes in the network. The path is built based on selecting a set of RPs that will be used as stopping points where the MS will gather the data from the sensor nodes. First, we introduce the algorithm that determines the initial locations of the RPs, and we then discuss the algorithm that is used to update the RPs' locations in each round of data collection.

4.1. Initial Locations of the RPs

The initial locations of the RPs are obtained as given in Algorithm 1. First, as in [12], a set of P RPs is obtained by clustering the sensor nodes into $|P|$ clusters using k-means algorithm [21]. These $|P|$ points represent the set of candidate RPs for building the MS path. Each RP i, ($1 \le i \le |P|$) is then assigned a priority value $R(i)$ using Equation (4):

$$R(i) = \frac{S_{RP}(i)}{H_{RP}(i)} \tag{4}$$

where $S_{RP}(i)$ is the number of sensor nodes associated with RP i and $H_{RP}(i)$ is the average hop distance between RP i and the sensor nodes associated with it. Equation (4) gives high priority for the RP, which has more sensor nodes attached to it, and the average distance of these nodes to that RP is small compared to the other potential RPs. Assume PL is the length, in meters, of the path that connects all required RPs to collect the data. Assume the time that the MS needs to receive the data from the sensor nodes when it stops at each RP is T_{data}, then the total time for collecting the data from all sensor nodes T_{total} in each round is given as:

$$T_{total} = c \times T_{data} + \frac{PL}{v} \tag{5}$$

where c is the number of RPs obtained using Algorithm 1 ($c \leq |P|$) and v is the MS speed. Some WSN applications require that the sensed data should be collected within a specific delay limit. Therefore, depending on the WSN application (for example, a real-time application), the path length should not exceed a threshold value ($PL_{threshold}$). Based on that, in Algorithm 1, after determining the candidate set P as RPs using k-means algorithm, only c RPs of this set will be selected to build the MS path.

Algorithm 1: Finding the initial locations of the RPs

```
INPUT: S, PL_threshold
OUTPUT: C, path for MS
Begin RPs INITIAL LOCATIONS
P = k-means(S); // Cluster the S sensor nodes into P clusters using k-means algorithm [21]
for i = 1 to P do
      Calculate the priority value of RP i using Equation (4)
End for
Sort the set P of RPs based on their priority values in descending order
C = { }; /* Set C contains the RPs that will be used to construct the MS path*/
RPx = remove RP from P
C = C ∪ {RPx}
c = 1
While True do
RPy = remove RP from P
    C = C ∪ {RPy}
c = c + 1
   PL = TSP(C);/* Call traveling sales person algorithm to obtain the path between the RPs in C */
      If PL ≥ PL_threshold then break
End While
If (size(P) > 0) then /* if some RPs in P are not used to construct the path */
Redistribute the nodes attached to the RPs in P to the RPs in C
until the path length is equal or larger than the specified
threshold.
End if
End
```

Once the initial RPs are selected and the initial path for data collection is constructed, the MS determines for each sensor a corresponding RP where each sensor node will be assigned to the closest RP. The MS then broadcasts a rendezvous information packet (RIP) to the whole network to inform each node about its RP. Therefore, when each sensor node receives the RIP packet, it obtains its destination RP. Once all sensor nodes know the destination RPs, the MS starts to accomplish data gathering by going through all RPs. When the MS becomes close to each destination RP, it sends a polling message with the ID of the corresponding RP. In this case, the corresponding sensor nodes prepare and transmit their data to the MS when it reaches the corresponding RP. If a node is within the MS communication range, it sends its data directly. Otherwise, it forwards the data to the nearest node of the corresponding RP. This procedure is repeated until all RPs are visited.

If the RPs are kept fixed during the entire operation of the network, it could pose energy holes around the fixed RPs location and, thus, the MS will no longer be able to receive data from sensor nodes with two or more hops around the RPs. Therefore, in this paper, a dynamic approach for updating the RPs locations every predefined number of rounds is proposed. For this purpose, we assume the MS keeps track of the following parameters during each round:

1. The current locations of the RPs.

2. The ID of the corresponding RP for each sensor.
3. The distance between each sensor and the corresponding RP.
4. The energy level of each sensor.
5. The number of surrounding sensor nodes for each sensor node.

Based on this information, the locations of the RPs are appropriately updated to extend the network lifetime and reduce the overall energy consumption as well. To perform the dynamic data collection for each new updated RPs' selection, the MS updates the ID of the corresponding RP of each sensor and broadcasts new RIP packets to inform the sensors with the new ID of the destination RP.

4.2. Updating the RPs Locations

In the proposed approach, the locations of the RPs are updated dynamically using a Fuzzy Inference System (FIS) [22]. An example of using fuzzy logic to control the MS movement toward cluster heads for improving the LEACH protocol [23] was proposed in [24]. In this paper, the proposed algorithm is called Fuzzy_RPs. In each round, the location of each RP is updated based on three factors:

1. The energy of each sensor around the RP. The energy level of each sensor influences how much the RP should be moved close to the sensor node. To balance the energy consumption and extend the network lifetime, the sensor with a low energy level has more impact to change the RP location and bring the RP close to it.
2. The distance between the sensor node and the corresponding RP. As the distance of transmission influences the amount of energy consumption, the location of RP should be updated to balance the distance between all sensor nodes and their corresponding RP. This factor has a significant impact to mitigate the overall energy consumption during the network lifetime.
3. The number of sensor nodes around each node. The sensor node within the dense region has more impact compared to the sensor node within the sparse region to influence the change of the RP location. The idea behind this factor is to attract the MS towards the dense region in order to be close to as many nodes as possible and therefore reduce the energy that will be used for transmission by the sensor nodes. This factor breaks the tie when two or more nodes have the same distance to their corresponding RP. For example, when two sensors have the same energy level and the same distance from the current corresponding RP, the sensor node with a high number of surrounding sensor nodes will attract the MS to update the location of the current corresponding RP towards it more closely compared with the sensor node in the sparse region.

The proposed FIS has three Inputs, which represent the three factors discussed above. For each sensor node i, the inputs of the FIS will be:

1. E_i: the remaining energy of sensor node i.
2. $d_{i \to RPi}$: the distance between sensor node i and its corresponding RP.
3. NBi: the 1-hop neighbors of sensor node i.

The output of the FIS determines a weight value wj between 0 and 1. The weight value increases if the distance is high or the energy of the node is low. The output weight value influences the change of the RP location. The sensor node with a higher weight value has more impact to change the location of the RP toward its location. Figure 2 shows the block diagram of the proposed Fuzzy_RPs. It consists of the basic four stages of FIS: fuzzification, evaluation of the fuzzy rules, aggregation, and deffuzzyfication of the output fuzzy sets where the Mamdani model is used for this purpose.

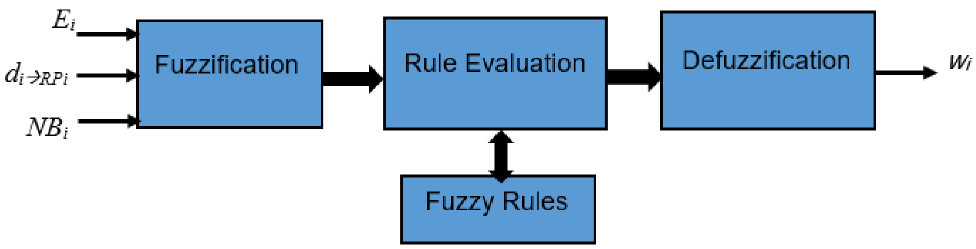

Figure 2. Block diagram of Fuzzy_RPs FIS.

The membership functions of the inputs and output are shown in Figure 3. The fuzzy rules are presented in Table 2, where AND operator is used to combine the three inputs. The shape of the membership functions of the fuzzy sets for the inputs and the output as well as the fuzzy rules are considered after running a set of simulations. In each run, the range of each fuzzy set and the fuzzy rules were evaluated, and the system was tuned until we got the membership functions shown in Figure 3 and the fuzzy rules given in Table 2.

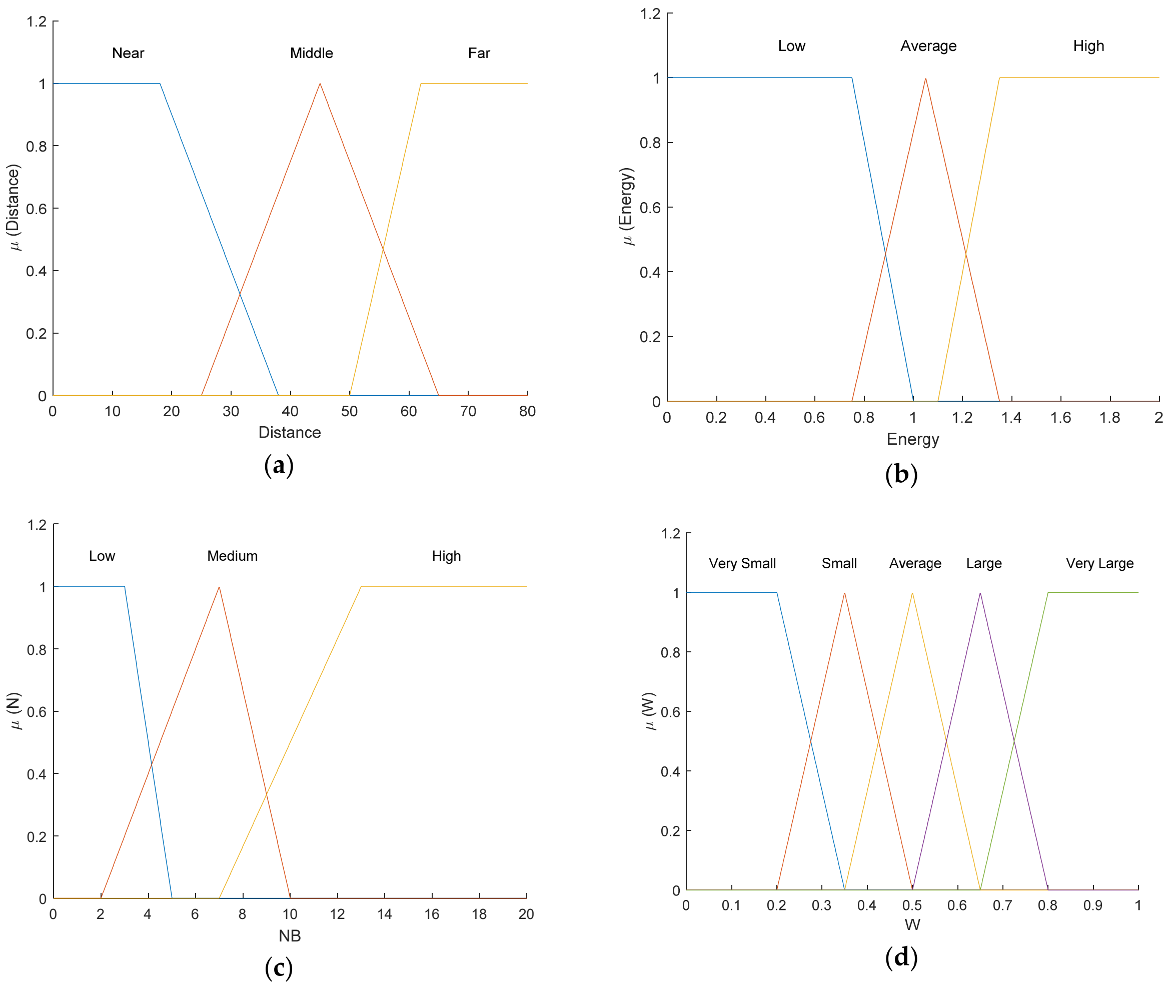

Figure 3. Membership functions of the inputs and output. (**a**) Membership functions of input Distance. (**b**) Membership functions of input Energy. (**c**) Membership functions of input (*NB*). (**d**) Membership functions gof output w.

Table 2. FIS Rules.

Rule Number	Inputs			Output w
	Distance	Energy	Number of Neighbors	
1	Near	Low	Low	Average
2	Near	Low	Average	Large
3	Near	Low	High	Very Large
4	Middle	Low	Low	Average
5	Middle	Low	Average	Large
6	Middle	Low	High	Very Large
7	Far	Low	Low	Large
8	Far	Low	Average	Very Large
9	Far	Low	High	Very Large
10	Near	Average	Low	Small
11	Near	Average	Average	Average
12	Near	Average	High	Large
13	Middle	Average	Low	Small
14	Middle	Average	Average	Average
15	Middle	Average	High	Very Large
16	Far	Average	Low	Average
17	Far	Average	Average	Large
18	Far	Average	High	Large
19	Near	High	Low	Very Small
20	Near	High	Average	Small
21	Near	High	High	Average
22	Middle	High	Low	Very Small
23	Middle	High	Average	Average
24	Middle	High	High	Small
25	Far	High	Low	Average
26	Far	High	Average	Large
27	Far	High	High	Large

Assume the current position of RP i is (X_{RPi}, Y_{RPi}), and the current position of sensor node j that is associated with RP i is (X_j, Y_j). Sensor node j will then attract RP i to its location and calculate its new position ($X_{RPi\to j}$, $Y_{RPi\to j}$) after calculating its weight value w_j using the FIS as follows:

$$X_{RPi\to j} = w_j \times (X_j - X_{RPi}) + X_j \tag{6}$$

$$Y_{RPi\to j} = w_j \times (Y_j - Y_{RPi}) + Y_j \tag{7}$$

The actual new updated position of RP i is based on all sensor nodes that are associated with RP i, and it is calculated as:

$$X_{Rpi}^{New} = \frac{\sum_{j=1}^{|S_j|} X_{RPi\to j}}{|S_j|} \tag{8}$$

$$Y_{Rpi}^{New} = \frac{\sum_{j=1}^{|S_j|} X_{RPi\to j}}{|S_j|} \tag{9}$$

where $|Sj|$ represents the total number of sensor nodes associated with RP i. After the locations of the RPs are updated, the MS constructs the adaptive path that passes through the new RPs set using Traveling Sales Man Problem Algorithm. Based on the new set of RPs, the mobile sink updates the corresponding RP of each sensor based on the nearest distance between each sensor with the new corresponding RP. When the mobile sink is about to reach an RP, it broadcasts a new polling message to the corresponding sensors with the ID of the new respective RP. Therefore, the sensors based on the polling message prepare the data and transmit it to the MS when it stops at the RP. Once the MS finishes the collection of the data from all corresponding sensor nodes, it moves to the next RP and repeats the process until it passes the whole set of RPs. Algorithm 2 presents the steps for updating the RPs' locations using FIS.

Algorithm 2: Fuzzy-based Adaptive Path Selection

INPUT: *S, C, U*
/ U is the period value to update the path */*
/ C is the set of RPs that are used to build the MS path */*
/ S is the set of sensor nodes */*
OUTPUT: *updated path for MS*
Begin *FUZZY_RPs*
Round = 1; / current round of collection data */*
while *there is still active nodes* **AND** *(mod(Round, U) = 1) OR Round = 1)* **do**
for *j = 1 to sizeof(S)* **do**
Determine the Nearest Corresponding RP of sensor s_j
Discover number of the One-Hop nodes of sensor s_j
End for
for *i = 1 to sizeof(C)* **do**
$RP_i = Ci$
for *each sensor* s_j *associated with* RP_i **do**
w_j *=FIS [energy(*s_j*), dist(*s_j *to RPi), Num of one Hop nodes (*s_j*)*
Calculate $XRP_{i \to j}$ *using Equation (6)*
Calculate $YRP_{i \to j}$ *using Equation (7)*
End for
New X of RPi = Calculate New X_{RPi} *using Equation. (8)*
New Y of RPi Calculate New Y_{RPi} *using Equation (9)*
End for
Round = Round + 1
End while
Use TSP algorithm [25] to construct the path that passes through the RPs in C
End

5. Performance Evaluation

In this section, the performance of the proposed approach is evaluated and compared with the DBRkM approach [12]. The parameters that were used in the simulation experiments are shown in Table 3. Each experiment with the same configuration was repeated for 10 runs where in each run the same number of sensor nodes is deployed randomly in the network. The average value is then considered for the experiment. The performance metrics that were used in the simulation are: the number of active sensor nodes, total energy consumption, the standard deviation of the remaining energy, and the path length during the network lifetime.

Figures 4 and 5 show the area of the sensing field and the network conditions using the DBRkM approach and the proposed FUZZY_RPs approach, respectively. As shown in Figure 4, the sensor nodes that are associated with RPs 3 and 4 suffer from high hop count and large distance. On the contrary, Figure 5 shows that the locations of RP 3 and 4 are updated at round 175 to balance the distance between the sensor nodes towards the RPs and to reduce the average hop count. These two figures are presented to show that the RPs are built and changed dynamically in the FUZZY_RPs approach (Figure 5) while the RPs' locations are fixed in the DBRkM approach.

Table 3. Parameters used in the simulation.

Parameter	Value
Target Area	220×220 m^2
Number of sensor nodes	150–300
Initial Energy of sensor nodes	2 Joule
Communication Range (Rc)	40 m
Packet Size (K_b)	4000 bits
Speed of mobile sink (v)	2 m/s
E_{elect}	50 nJ/bit
Mp	0.0013 pJ/bit/m^4

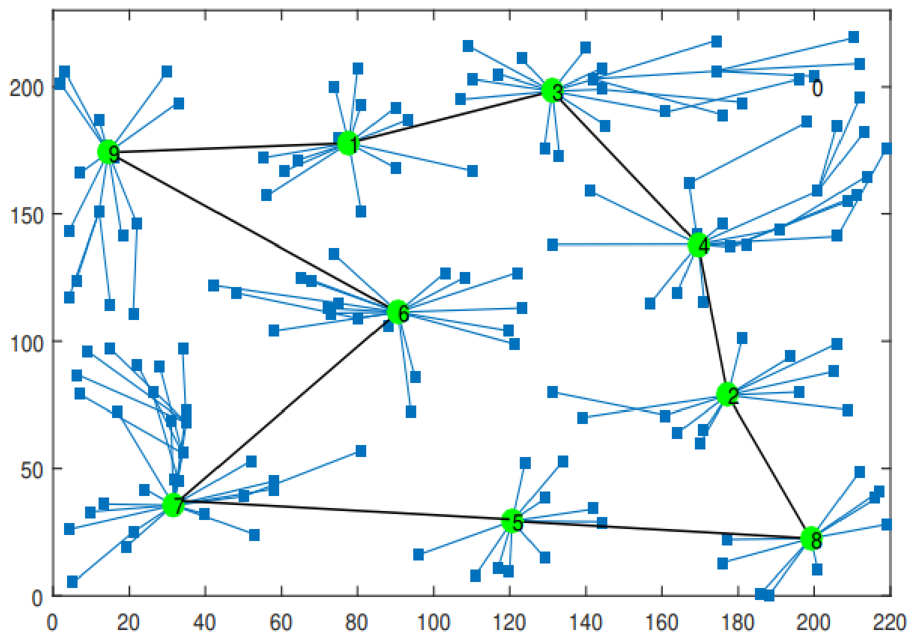

Figure 4. DBRkM Run-Time Simulation (average hop count = 1.24, tour length = 1034 m fixed at all rounds.

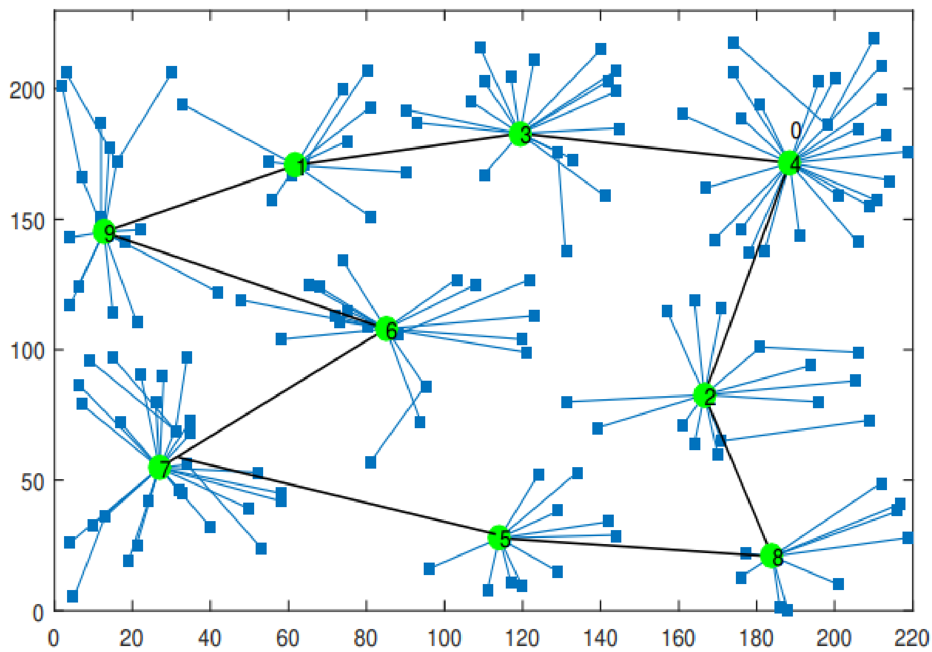

Figure 5. Fuzzy_RPs Run-Time Simulation at round 175 (average hop count = 1.12, tour length = 1016 m).

5.1. Network Lifetime

In this section, we compare the performance of the proposed approach with the DBRkM approach in terms of network lifetime. Three metrics are used for this purpose: the number of alive nodes, total remaining energy, and standard deviation of the remaining energy. Figure 6 shows the number of active nodes per round in both algorithms. The figure shows a significant improvement of the proposed Fuzzy_RPs approach compared to the DBRkM approach. This improvement is achieved by updating the locations of the RPs based on the energy of the nodes and therefore avoiding the energy holes around the RP.

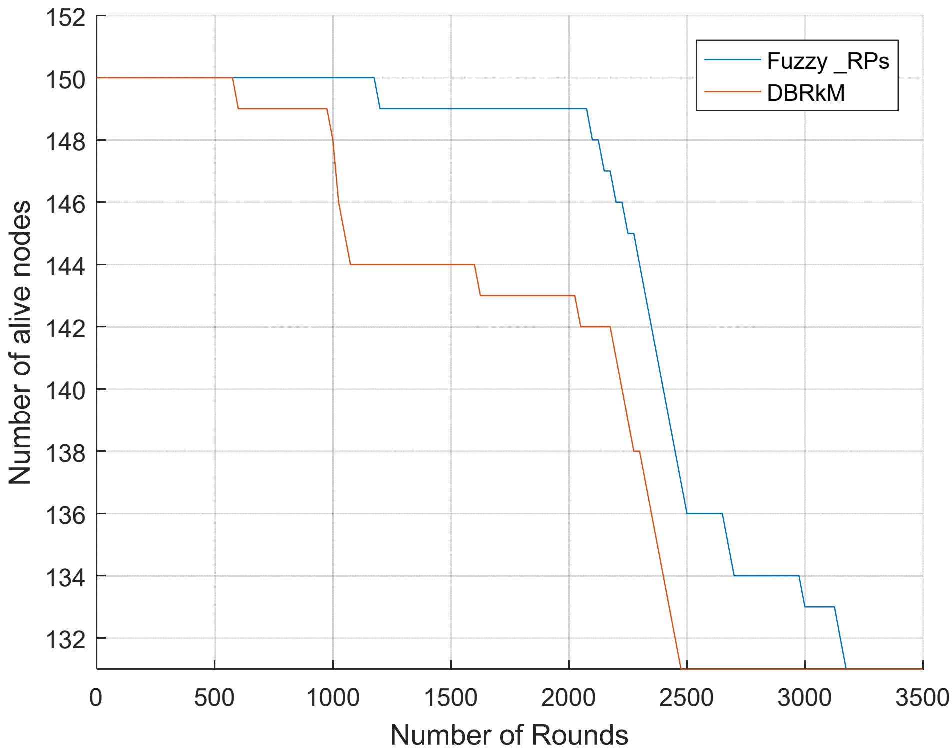

Figure 6. Number of alive nodes vs. rounds.

Figure 7 shows the overall amount of energy consumed throughout the first 3000 rounds. As can be seen in the figure, the proposed approach consumes less energy than the DBRkM approach. This improvement is achieved by keeping the RPs' points updated with the fewest hops and the shortest transmission distance possible. Figure 8 shows the standard deviation of the total remaining energy for live nodes per round for the proposed approach and the DBRkM approach. As shown in Figure 8 the standard derivation in the case of DBRkM approach increases compared to the Fuzzy_RPs approach. This increase comes from the fact that the RPs in DBRkM approach are fixed and therefore the nodes that are far away from these RPs will drain their energy fast compared to the nodes close to the RPs. On the contrary, the Fuzzy_RPs approach always achieves a balance in energy consumption by changing the RPs' locations in each round.

Figure 9 shows the network lifetime when the number of sensor nodes is changed from 150 through to 300. The network lifetime in this figure represents the number of rounds until the first node in the network dies. As shown in the figure, the proposed Fuzzy_RPs approach always outperforms the DBRkM approach. Table 4 shows the percentage of improvement that has been achieved by using the Fuzzy_RPs approach compared to DBEkM approach.

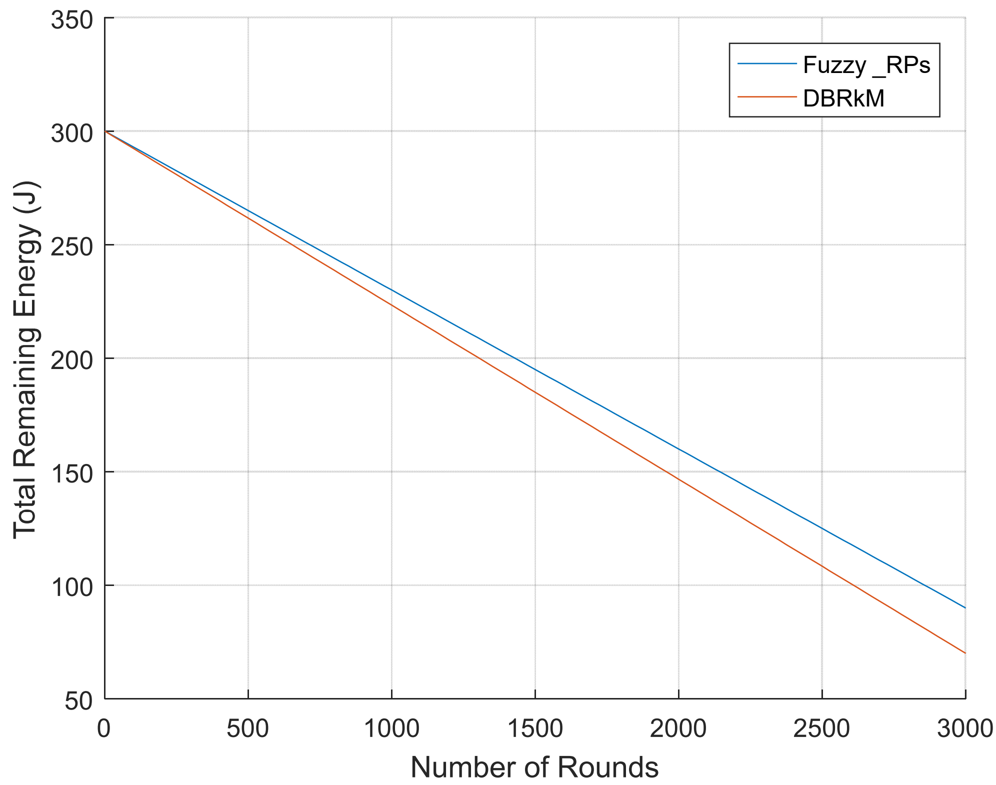

Figure 7. Total remaining energy vs. rounds.

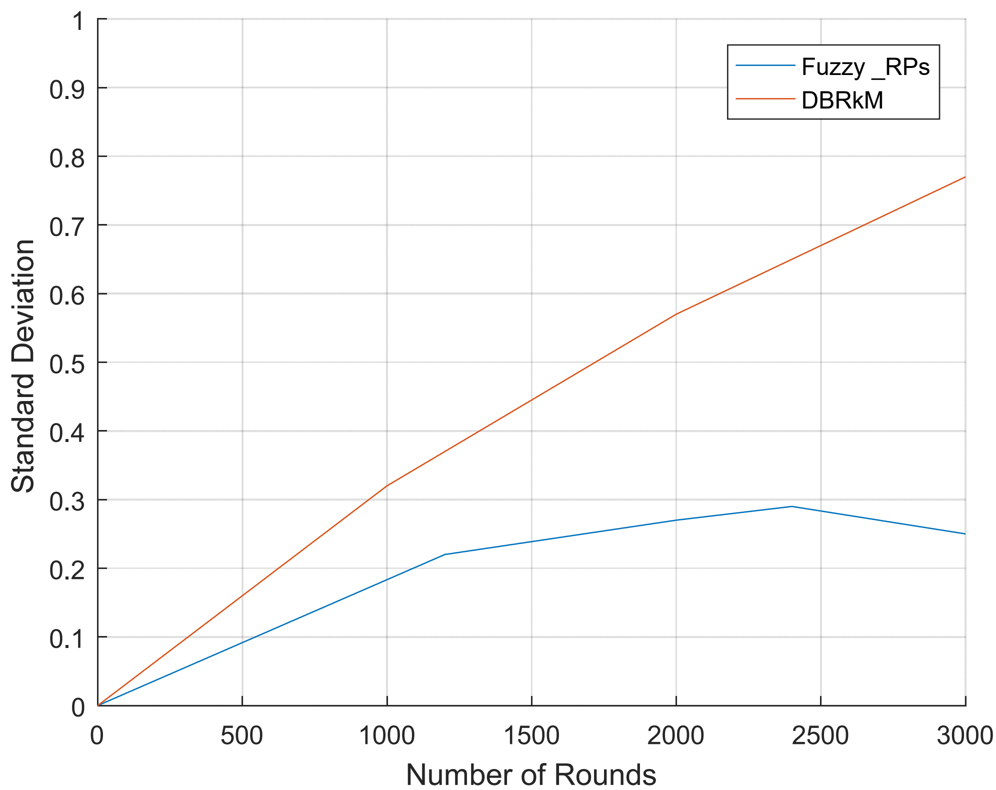

Figure 8. Standard deviation of the remaining energy vs. rounds.

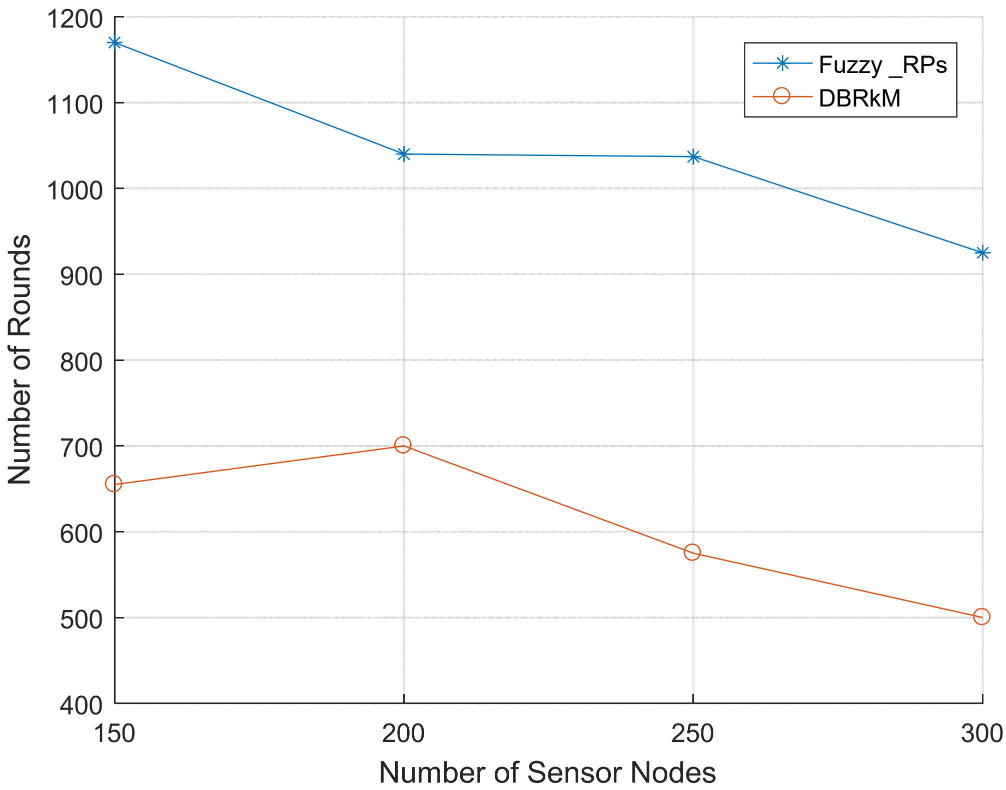

Figure 9. Network lifetime vs. number of sensor nodes.

Table 4. Network lifetime improvement with different number of sensor nodes.

Number of Sensor Nodes	Number of Rounds until First Node Dies		Improvement (%)
	Fuzzzy_RPs Approach	DBRkM Approach	
150	1172	655	78.93
200	1044	704	48.30
250	1037	575	80.35
300	928	505	83.76

5.2. Path Length during the Network Life Time

We assume that the MS energy is much more than the sensor node energy. The cost of energy dissipated as a result of the MS movement is represented by the length of the path that the MS will use to collect the data. Figure 10 shows the path length in meters in the DBRkM and FUZZY_RPs approaches. In this experiment, the number of sensor nodes was 150. As shown in the figure, the path length in the DBRkM approach is constant while the path construction is dynamic during the network lifetime in the FUZZY_RPs approach. As shown in the figure, the path length exceeds the DBRkM approach only in the last stage of the network, almost when the remaining energy in the network is low and most of the nodes are dead. As shown in Table 4, for the case when the number of sensor nodes is 150, the first dead node in the FUZZY_RPs approach occurs in round 1172 and for the DBRkM approach in round 655. However, during the lifetime of the network, the path length in the FUZZY_RPs approach is less than the path length in the DBRkM approach, especially in the first 2700 rounds.

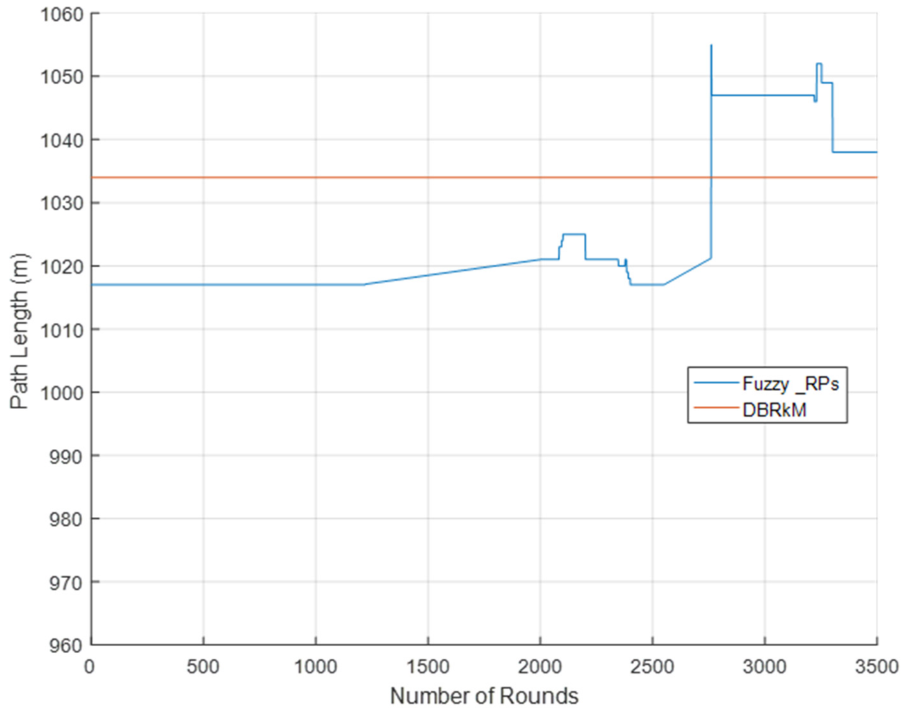

Figure 10. Path Length.

6. Conclusions

In this paper, we proposed an adaptive path construction for the mobile sink using FIS called FUZZY_RPs. After determining the required set of RPs using the k-means algorithm in the first round, the locations of these RPs are then adapted dynamically in the subsequent rounds using the FIS, which calculates the new location of each RP based on a weight value generated by each sensor node in the network. This weight value generated by a certain node determines how long the current location of the corresponding RP should be moved towards that node. The fuzzy-based calculation of the weight value for each sensor node depends on three factors: the remaining energy of the sensor node, the distance between the sensor node and its corresponding RP, and the 1-hop neighbors of the sensor node. Simulation results showed that the proposed approach outperforms the DBLkM approach, which uses fixed RPs over the whole network lifetime. For example, in terms of the network lifetime, the proposed FUZZY_RPs approach achieved an 83.76% improvement when the number of sensor nodes is 300. Future work can look into the adaptive path design for the MS by updating the locations of the RPs in environments with obstacles.

Author Contributions: Implementation, software, validation, writing—original draft preparation, data collection, and analysis, F.A.-Q.; supervision, writing—review and editing, project administration, O.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Jordan University of Science and Technology grant number 20200241.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author.

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

References

1. Abdur, R.M.; Arafatur, R.M.; Rahman, M.M.; Asyhari, A.; Bhuiyane, M.; Ramasamy, D. Evolution of IoT-enabled connectivity and applications in automotive industry: A review. *Veh. Commun.* **2021**, *27*, 100285. [CrossRef]
2. Agarwal, V.; Tapaswi, S.; Chanak, P. A Survey on Path Planning Techniques for Mobile Sink in IoT-Enabled Wireless Sensor Networks. *Wirel. Pers. Commun.* **2021**, *119*, 211–238. [CrossRef]
3. Akbar, M.; Javaid, N.; Abdul, W.; Ghouzali, S.; Khan, A.; Niaz, I.A.; Ilahi, M. Balanced Transmissions Based Trajectories of Mobile Sink in Homogeneous Wireless Sensor Networks. *J. Sens.* **2017**, *2017*, 4281597. Available online: https://www.hindawi.com/journals/js/2017/4281597/ (accessed on 1 September 2022). [CrossRef]
4. Temene, N.; Sergiou, C.; Georgiou, C.; Vassiliou, V. A survey on mobility in wireless sensor networks. *Ad Hoc Netw.* **2022**, *125*, 102726. [CrossRef]
5. Suh, B.; Berber, S. Rendezvous points and routing path-selection strategies for wireless sensor networks with mobile sink. *Electron. Lett.* **2016**, *52*, 167–169. [CrossRef]
6. Gupta, P.; Tripathi, S.; Singh, S. Energy efficient rendezvous points based routing technique using multiple mobile sink in heterogeneous wireless sensor networks. *Wirel. Netw.* **2021**, *27*, 3733–3746. [CrossRef]
7. Irish, A.E.; Terence, S.; Immaculate, J. Efficient data collection using dynamic mobile sink in wireless sensor network. In *Wireless Communication Networks and Internet of Things, Lecture Notes in Electrical Engineering*; Springer: Singapore, 2019; pp. 141–149. [CrossRef]
8. Wang, J.; Gao, Y.; Yin, X.; Li, F.; Kim, H. An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 9472075. [CrossRef]
9. Liang, W.; Luo, J.; Xu, X. Prolonging network lifetime via a controlled mobile sink in wireless sensor networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), Miami, FL, USA, 6–10 December 2010; pp. 1–6. [CrossRef]
10. Farzinvash, L.; Najjar-Ghabel, S.; Javadzadeh, T. A distributed and energy-efficient approach for collecting emergency data in wireless sensor networks with mobile sinks. *AEU Int. J. Electron. Commun.* **2019**, *108*, 79–86. [CrossRef]
11. Park, J.; Moon, K.; Yoo, S.; Lee, S. Optimal stop points for data gathering in sensor networks with mobile sinks. *Wirel. Sens. Netw.* **2012**, *4*, 8–17. [CrossRef]
12. Kaswan, A.; Nitesh, K.; Jana, P.K. Energy efficient path selection for mobile sink and data gathering in wireless sensor networks. *AEU-Int. J. Electron. Commun.* **2017**, *73*, 110–118. [CrossRef]
13. Banimelhem, O.; Taqieddin, E.; Shatnawi, I. An Efficient Path Generation Algorithm Using Principle Component Analysis for Mobile Sinks in Wireless Sensor Networks. *J. Sens. Actuator Netw.* **2021**, *10*, 69. [CrossRef]
14. Sharma, S.; Puthal, D.; Jena, S.K.; Zomaya, A.Y.; Ranjan, R. Rendezvous based routing protocol for wireless sensor networks with mobile sink. *J. Supercomput.* **2017**, *73*, 1168–1188. [CrossRef]
15. Vajdi, A.; Zhang, G.; Zhou, J.; Wei, T.; Wang, Y.; Wang, T. A New Path-Constrained Rendezvous Planning Approach for Large-Scale Event-Driven Wireless Sensor Networks. *Sensors* **2018**, *18*, 1434. [CrossRef] [PubMed]
16. Raj, P.V.P.; Khedr, A.M.; AL Aghbari, A. Data gathering via mobile sink in WSNs using game theory and enhanced ant colony optimization. *Wirel. Netw.* **2020**, *26*, 2983–2998. [CrossRef]
17. Boyineni, S.; Kavitha, K.; Sreenivasulu, M. Mobile sink-based data collection in event-driven wireless sensor networks using a modified ant colony optimization. *Phys. Commun.* **2022**, *52*, 101600. [CrossRef]
18. Donta, P.K.; Amgoth, T.; Annavarapu, C.S.R. An extended ACO-based mobile sink path determination in wireless sensor networks. *J. Ambient Intell. Hum. Comput.* **2021**, *12*, 8991–9006. [CrossRef]
19. Ghaleb, M.; Subramaniam, S.; Ghaleb, S.M. An Adaptive Data Gathering Algorithm for Minimum Travel Route Planning in WSNs Based on Rendezvous Points. *Symmetry* **2019**, *11*, 1326. [CrossRef]
20. Sun, G.; Liu, Y.; Zhang, J.; Wang, A.; Zhou, X. Node Selection Optimization for Collaborative Beamforming in Wireless Sensor Networks. *Ad Hoc Netw.* **2016**, *37*, 389–403. [CrossRef]
21. Hartigan, J.A.; Wong, M.A. Algorithm as 136: A k-means clustering algorithm. *Appl. Stat.* **1979**, *28*, 100–108. [CrossRef]
22. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
23. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless micro sensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7 January 2000; pp. 1–10. [CrossRef]
24. Banimelhem, O.; Abu-hantash, A. Fuzzy logic-based clustering approach with mobile sink for WSNs. In Proceedings of the 13th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 27–28 December 2017. [CrossRef]
25. Johnson, D.S.; McGeoch, L.A. Experimental analysis of heuristics for the STSP. In *The Traveling Salesman Problem and Its Variations*; Springer: Boston, MA, USA, 2007; pp. 369–443. [CrossRef]