IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 17, 2021, accepted January 30, 2021, date of publication February 3, 2021, date of current version February 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056701

Energy-Efficient and Fast Data Collection in UAV-Aided Wireless Sensor Networks for Hilly Terrains

REZOAN AHMED NAZIB AND SANGMAN MOH, (Member, IEEE)
[1]Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea

Corresponding author: Sangman Moh (smmoh@chosun.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MIST) under Grant 2019R1F1A1060501.

ABSTRACT Energy-constrained sensor nodes are often deployed in remote, hilly, and hard-to-reach areas for civilian and military purposes. In such wireless sensor networks (WSNs), an unmanned aerial vehicle (UAV) can be used to collect data from the sensor nodes. Low-altitude UAVs can be utilized to reduce the energy consumption of WSNs by optimizing the data collection position. In this study, we designed an energy-efficient and fast data collection (EFDC) scheme in UAV-aided WSNs for hilly areas with the help of a UAV as a data mule. First, we proposed a central bias hybrid energy-efficient distributed clustering algorithm for grouping the sensors. Then, we applied a modified tabu search algorithm to optimize the UAV position for collecting data from a cluster. To achieve fast data collection, we developed the traveling salesman problem with the derived data collection positions and solved it by applying a modified genetic algorithm. Based on our simulation results, the proposed EFDC scheme outperforms the conventional ones in terms of energy consumption, scalability, control overhead, delay, and load balancing.

INDEX TERMS Clustering, data collection, genetic algorithm, HEED, unmanned aerial vehicle, drone, wireless sensor network.

I. INTRODUCTION

Wireless sensor networks (WSNs) are one of the most investigated research topics in the last two decades. They are used in home and industry automation, forest monitoring, scientific experiments, environmental observation, border patrolling, machine and structure health monitoring, security enhancement, plant monitoring, underwater world observation, air pollution examination, water quality monitoring, natural disaster prevention, and landslide detection. Because sensor nodes are mostly cheap and battery-powered, they are highly energy-constrained [1]. Consequently, many studies have been devoted to minimizing their energy consumption by using various techniques such as clustering [2], efficient routing [3], optimizing the medium access control (MAC) [4], and optimal sink placing [5].

WSNs are often deployed in hard-to-reach areas. Data collection from such areas can be challenging due to the absence of any network communication center (NCC). The major disadvantage of NCC or any other infrastructure-based solutions are to build and maintain the infrastructures in such irregular and inaccessible terrains [6]. Ground robot-based mobile sink solution can be used to collect WSN data from such region.

The mobile sink-based solutions have triggered the investigation of the performance of unmanned aerial vehicles (UAVs) as a data mule. UAVs can easily fly toward a guided direction owing to their three-dimensional (3D) movement capability [7]. Compared to ground robots, UAVs can travel a greater distance within a shorter period of time [8]. Using UAVs for data collection in WSNs opens a new horizon of energy-efficient data collection from remote and inaccessible terrains [9]. In the UAV-aided WSNs (UWSNs), interconnectivity is not as important as in the conventional paradigms. A line of sight (LOS) communication can also be obtained by using the position optimization capability of the UAV.

Multi-rotor and fixed-wing UAVs are the two popular kinds of UAVs. Multi-rotor UAVs usually require a lesser bending angle for changing the moving direction compared to the fixed-wing UAVs. This type of UAVs can also float steadily

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

in the air. Thus, multi-rotor UAVs can be used to fine-tune the data collection position from a group of sensors.

The deployment requirements of sensors do not confirm a uniform distribution throughout the region of interest (ROI). Therefore, sensor grouping by segmenting the geographic location is not a good strategy. A better approach is to use a distributed clustering technique and allow the sensor nodes to decide the group themselves. In an infrastructure-less environment, the UAV does not get any prior information on the location of the CHs. In such cases, if the clustering algorithm runs more than once, the UAV will have to discover the CH's location in every round. Furthermore, if a CH fails before transmitting data to the UAV, then the data of cluster members (CMs) and CHs will be lost. Thus, hierarchical data collection is not a suitable option for infrastructure-less UWSNs.

The deployment of WSNs in the hilly or unreachable areas include vital reasons like border monitoring [10], landslide prediction [11], avalanche prediction [12], and important scientific data collection. In such application scenarios, remote monitoring is important as all of the scenarios raise a threat against human life or national security. However, the traditional WSN data collection approach is a tough task in hilly areas such as building, and maintaining static infrastructure is time-consuming and needs more efforts. On the other hand, traditional WSNs consume more energy due to relatively long-distance transmission, which is the main reason behind the decreased lifetime of sensor nodes. A mobile sink can reduce the energy consumption by lessening the transmitter-receiver distance but ground-based mobile sink is impossible to deploy in the hilly areas [13]. A UAV-based WSN data collection can be the solution of the given scenario. However, in the existing UAV-aided WSN research works, 3D maneuvering and steady hovering capability are not exploited yet. Our proposed EFDC exploits the quadcopters' 3D movement, steady hovering, and low-altitude flying capability to enable energy-efficient data collection from WSNs in an infrastructure-less scenario.

In this study, an energy-efficient and fast data collection (EFDC) scheme is proposed for UWSNs deployed in hilly terrains. Fig. 1 shows the graphical representation of the EFDC operation, where a multi-rotor UAV is deployed to collect WSN data from hilly terrain. The figure depicts the direct data collection mechanism of the EFDC scheme from the sensor nodes to the UAV. The suboptimal positions for data collection along with the shortest trajectory of the UAV are also shown in the figure to illustrate the main contribution of EFDC framework.

The novelty of the proposed EFDC framework lies in the heart of its infrastructure-less design. EFDC can operate without any prior information about the WSN topology, so it does not need the presence of any NCC. In the traditional schemes, data is transmitted from CMs to their CH and from CHs to the sink. In EFDC, the UAV collects the sensor's data directly from the nodes of a cluster. Using direct transmission from the sensor nodes to the UAV reduces the transmission count. As a result, the workloads and energy consumptions among the CH and CMs are also balanced. The clustering algorithm in EFDC also takes place only once. As a result, the exchange of control packets reduces significantly. In EFDC, the UAV acts as the searching agent. In such a design, the UAV changes its position physically to examine the received signal strength indicator (RSSI) value from the sensor nodes. The UAV acting as a search agent is a well-known approach used for UAV networks [14]. EFDC exploits 3D positioning capability of the multi-rotor quadcopter and reduces the transmission distance in a cluster by applying the tabu search mechanism. Reducing the energy consumption of sensor nodes by exploiting 3D positioning capability is also a novel idea in our proposed EDC. Though EFDC is specially designed for hilly/mountainous terrain, this data collection architecture will have the same performance efficiency as in the flat and urban terrain with the added facility of infrastructure-less operation.

The contributions of this study can be summarized as follows:

- We propose a center-biased hybrid energy-efficient distributed (CBHEED) clustering algorithm, in which the CHs are selected based on the central bias of their geolocation. The central bias of a node is calculated by forming a polygon with the help of the monotone chain convex hull algorithm. The proposed CBHEED is a distributed clustering algorithm, which is especially applicable for infrastructure-less area. The position of the elected CHs serves as the initial position for the data collection position searching mechanism.
- We formulate an optimization problem for fine tuning the data collection position in a cluster and propose a modified tabu search algorithm to find the sub-optimal solution. The optimization problem focuses on maximizing the RSSI value among all the cluster members as well as balancing the UAV-sensor distance in a cluster. We modify the tabu search algorithm in order to find out the sub optimal position for data collection within the minimum number of iterations by searching the least number of spaces.
- Based on the derived data collection positions from the aforementioned tabu search mechanism, we apply a modified genetic algorithm (GA) to determine the optimized trajectory to minimize the UAV travel time. The applied GA algorithm ensures the avoidance of premature convergences. Finding out the UAV trajectory enables the UAV to collect sensor data within the minimum amount of time.
- According to our evaluation, the proposed EFDC scheme outperforms the conventional schemes in terms of energy consumption, scalability, control overhead, delay, and load balancing.

The remainder of the article is organized as follows. In Section II, the related works are reviewed and discussed. The limitations of the existing studies and the motivations behind the research are also provided in Section II. In Section III, the system model of the EFDC scheme is introduced. In Section IV, we describe the working procedure of the EFDC scheme. Then, we elaborate and discuss the CBHEED clustering algorithm, initialization phase of the

EFDC scheme, CH finding algorithm, modified tabu search algorithm, and outline of the modified GA representing the data collection phase in section IV. In Section V, the performance of the proposed scheme is evaluated and compared with the conventional schemes. Finally, the conclusion is given in Section VI.

II. RELATED WORKS

Ali *et al.* [15] derived that an optimized constant speed of a UAV in a fixed altitude can reduce the data drop rate of WSN nodes. To formulate their design theoretically, the authors used a tri-rotor UAV. However, the proposed model is only applicable to linearly deployed wireless sensor nodes with minimum width. The forward and backward movement depiction of the UAV can only cover sensor nodes that are inside the radio range of the UAV. Though the UAV–CH distance in this scenario will be lower than the CH–sink distance, the CH transmission count will remain the same, resulting in an unbalanced network. Liu and Zhu [16] proposed three different transmission modes, namely waiting mode, sensor node–sink conventional transmission mode, and sensor node–UAV transmission, to increase the energy performance of the WSN. They utilized dynamic programming to obtain an optimal transmission policy recursive random search algorithm to optimize the trajectory of the UAV. In addition, they assumed a static infrastructure while trying to optimize the energy consumption by utilizing the UAV. However, this data collection scheme is not suitable for infrastructure-less scenarios whereas our proposed EFDC is specially designed to be suitable for infrastructure-less scenarios.

Ebrahimi *et al.* [17] formulated a joint optimization problem by considering node clustering and UAV trajectory optimization for dense and large networks. The authors attempted to reduce the energy consumption by using a compressive data gathering method to aggregate the sensed data, thus reducing the number of required transmissions. According to their proposed solution, a forwarding tree was constructed from the CMs to the CHs and the data were aggregated in each level of the tree. However, by forming the tree, the compressed data need to be retransmitted before reaching the CH. Based on the sampling data, the performance of the proposed system varied greatly. However, the data aggregation scheme will cause some extra energy consumption for WSNs but, in EFDC, the data aggregation duty is given to UAV. Zhan *et al.* [18] also considered a joint optimization problem to optimize the energy consumption of a network. The authors considered a wake-up schedule for the sensor nodes and the UAV trajectory to define the joint optimization problem. A block-fading channel was assumed to design the ground–UAV communication. Though the wake-up strategy in WSNs can save energy but this architecture is infrastructure-dependent unlike EFDC. An interesting read for the researchers working on optimal UAV trajectory with multi-parameter optimization is given in [19]. A UAV trajectory is proposed in [20] for WSN-based pesticide controlling. This research focuses on energy depletion through power consumption, harvesting solar power, energy storage limitation, and various QoS parameters. However, this work is best suited for cellular communication architectures. A systematic review on timely UAV-aided WSN research works is summarized in [21].

Say *et al.* [22] proposed a new priority-based MAC protocol to reduce the number of redundant transmissions. The priority-based frame selection process takes the mobility characteristics of the UAV into consideration. Based on the aforementioned MAC protocol, they also proposed a routing protocol to minimize the routing distance between the UAV and the sensor node. However, the fixed-wing UAV used in the design is not suitable for accurate positioning and generally needs a higher altitude compared to the rotary-wing UAV. Another MAC protocol for UWSNs was proposed in [23], considering fast and energy-efficient data gathering for critical situations. A survey on MAC protocols for UWSNs was proposed in [24]. However, our research goal does not include optimizing the medium access usage.

Ho *et al.* [25] applied particle swarm optimization to obtain the optimal WSN topology and UAV trajectory for reducing energy consumption. The proposed model was compared with a low-energy adaptive clustering hierarchy (LEACH) protocol to evaluate its performance. Though the framework considered a relatively flat terrain to model the radio communication, the radio model used in the literature can also be useful to design propagation models in other environments. Different from the other proposed models, the UAV is also utilized in this architecture to select the CH from the ground sensor nodes.

In [26], a test-bed experiment was conducted at the Fundulea National Research Institute under the Romanian project MUWI. This data collection framework assumed that the sensor transmits its sensed data to the nearest base station first. The base stations were considered as the waypoint for the UAV to collect the sensed data. A heavily infrastructure-dependent mechanism was shown, though the architecture utilized the UAV to minimize the data collection energy consumption. You and Zhang [27] considered affecting fading power of the propagated signal to model the UAV–WSN communication channel. An obstacle-aware 3D trajectory model was derived for the UAV's mobility. The proposed model successfully maximized the least data collection rate by calculating an effective outage probability. In [28], the authors proposed a new K-means++ based WSN clustering approach. This architecture assumed uneven and random deployment of sensor nodes in the field of interest. Based on the remaining energy and storage capacity, the CH was selected from the cluster with the help of fuzzy logic.

Chen *et al.* [9] proposed a data gathering mechanism for UWSNs, where the target area was also divided into clusters. The CH was determined based on the information value and the residual power in the sensor nodes. The direct future prediction was used to design the optimal trajectory of the data collection scheme. Pang *et al.* [29] also investigated the

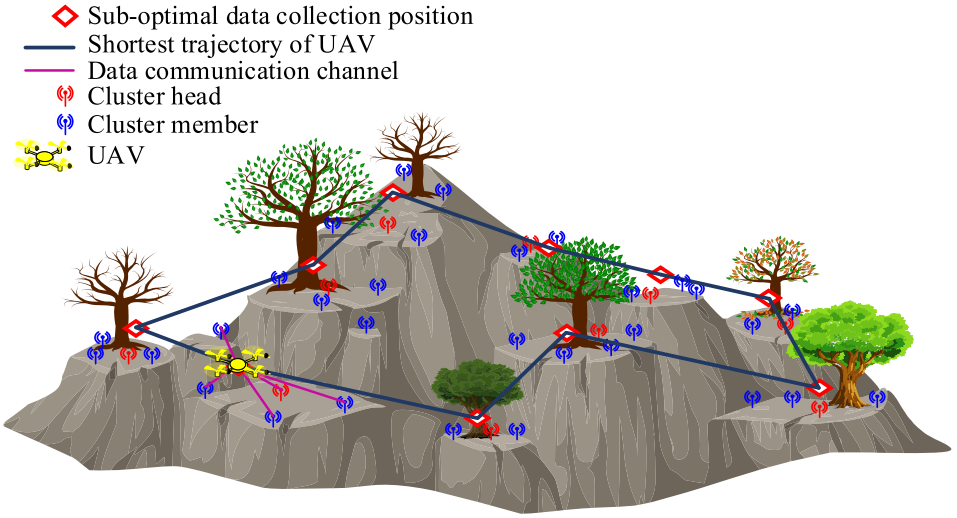

FIGURE 1. Graphical representation of data collection in a UWSN.

problem of data collection from harsh terrain. Besides WSN data collection, their architecture also considered recharging the sensor nodes while gathering data from them.

The optimal WSN CH selection technique is observed in many studies [30]. Some other studies investigated the optimal trajectory problem for collecting WSN data [31], [32], while some studies were performed to localize the sensor nodes [33], [34].

Some interesting research works for the cellular network have also been published [35]–[39], which are based on UAV-aided communication. To enable wireless power transmission and data communication at the same time, a UAV-aided non-orthogonal multiple access solution is presented in [35]. Besides data transmission and wireless power transfer, the proposal also ensures data security. However, the work tries to optimize the energy efficiency of users in the energy harvesting point of view, which is in contrast to the decreased transmission distance done in our proposed EFDC. A comprehensive review paper focused on 5G and beyond is given in [36]. The article presents UAV-aided 5G research studies briefly with proper recommendations. In [40], the authors address the problem of sink deployment, where the UAVs act as the sink. UAV deployment problem in other networking paradigm can also be facilitated from the research. A wireless powering system for sensor nodes and data collection with the help of UAV is proposed in [41]. Such deployment will elongate the lifetime of WSNs as their energy will be refilled as per need basis, and long transmission will be avoided with the help of UAVs. However, none of the aforementioned researches in this paragraph focuses on the optimal placing of a UAV as a sink node for the hilly terrains.

In [42], Trasviña-Moreno *et al.* did a test-bed experiment for an ocean infrastructure monitoring system. Where the sensors are installed inside buoys and UAV searches and collects the data based on the previous location. However, they did not apply any optimization process to reduce the energy consumption of the sensor nodes. Besides, the protocol is not also suitable for a large number of nodes. In [43], Dragana *et al.* proposed a surveillance system, combining WSN and UAV. They proposed a new stochastic channel modeling scheme for UAV-WSN communication. However, they did not consider UAV position optimization, which is the main contribution in our proposed EFDC. In [44], Bacco *et al.* used WSN and UAV to establish a monitoring system for the ancient buildings. However, the focus is given on 3D construction of the structure, and no optimization is done from the networking or data communication perspective. A UAV-based WSN border surveillance system is proposed in [45], but this architecture is also dependent on static infrastructure and does not consider the energy issue of sensor nodes. A test-bed of peat fire detection technique is given in [46] with the help of a WSN and fixed-wing UAV. However, the fixed-wing UAV is not applicable for position optimization. Besides, this technique is heavily dependent on the BS.

Even though data collection for UWSNs is more suitable for remote areas, the availability of infrastructure makes the scenario incompatible. Some existing techniques assume that the control center has prior knowledge of WSN topology. This can be a bottleneck in terms of random deployment of sensor nodes in harsh environments. Random deployment is specifically used in most of the studies, which can be matched with the real-life unequal distribution of sensor nodes. It is observed from the above discussion that most of the existing architectures have prior knowledge about the WSN topology. In our EFDC scheme, however, the UAV does not need any prior information from the infrastructure, which is different from the previous studies. This scenario decreases the utilization of UAVs. Collecting data only from CHs also reduces the utilization of the UAV while creating extra burden for CHs. Even though some of the studies show trajectory optimization

techniques, the altitude optimization technique is not shown in the investigated literature. The altitude optimization technique can lead to reduction in the distance between the nodes and the UAV, resulting in an efficient energy consumption.

The notations used in the study are summarized in Table 1.

III. SYSTEM MODEL

In this section, assumptions, communication model, and UAV mobility model are presented. The assumptions are listed separately for the application area, WSN, UAV, and MAC protocol. In the communication model, the descriptions of the application area, communication phases, corresponding jobs, and applied algorithms are mentioned. At the end of this section, the default mobility model of the UAV is explained.

A. ASSUMPTIONS

The limitations and assumptions of the study are categorized for the application area, WSN, UAV, and MAC protocol separately. While making the assumptions, we carefully considered the standard assumptions in related studies and the feasibility of implementation. The assumptions are given below:

1) ASSUMPTIONS FOR APPLICATION AREA

Hilly/Mountainous Terrain: It is assumed that the ROI is not flat and some natural obstacles are present in the environment such as trees, rocks, and uneven ground. These obstacles can cause scattering, diffraction, and reflection in the ground-to-ground transmission such as sensor-to-sensor communications [47]. However, air-to-ground or ground-to-air transmissions are not affected with the obstacles [48]. In case of EFDC, the UAV is assumed to collect data from the sensor nodes. The communication between sensors and UAV is obstacle-free according to this assumption of the application area. However, apart from mentioned application area, EFDC is also operable in flat or urban area.

Absence of Static Infrastructure: It is assumed that the WSN is deployed in a remote area, and any static infrastructure such as a static sink or any network communication point is absent. For this given scenario, the sensor nodes are unable to communicate with the NCC or outer world. Usually, a node with consistent power supply works as a sink node. Building such an end point will require power supply with cable connection and regular supervision. In such scenario, building an NCC is not an option for consideration. In general, deploying a WSN for monitoring purpose is not a permanent setup. Hence, building and maintaining any static infrastructure will introduce an additional overhead. For such scenarios, UAV is a more suitable option as an aerial mobile data collector.

2) ASSUMPTIONS FOR WSN

Location Awareness: The sensor nodes are location aware. They are equipped with a global positioning system (GPS). By utilizing the GPS module, a sensor node can query about its latitude, longitude, and altitude values [49]. The GPS module is a practical assumption from the viewpoint of cost and energy consumption. The sensor nodes are static, thus reading

TABLE 1. Notations Used in This Study.

Notation	Definition
K	Geo-position of sensor nodes
N	Set of sensor nodes
ξ_{UAV}^{m}	Transmission power required to transmit data from node m to the UAV
C	Set of clusters
l_m	Bit number that a CM wants to transmit
φ_{o_i,o_j}	Indicates whether the UAV is taking the path from position o_i to position o_j
$elec$	Energy consumption of a sensor node for transmitting one bit
ψ_{fs}	Transmitter amplifier model in a free space environment
ψ_{mp}	Transmitter amplifier model serving a multipath model
l	Number of bits in a specific transmission
δ_{th}	Threshold distance for data transmission
E_{elec}	Circuitry energy consumption for transmitting one bit of data
$E_{Tx-elec(l)}$	Circuitry energy consumption for transmitting l bit of data
δ	distance between two nodes
$E_{Tx-amp(l,\delta)}$	Energy consumption of the amplifier of a node to transmit l bit of data to distance δ
$E_{Rx}(l)$	Energy consumption for receiving l bit of data
$E_{Rx-elec}$	Energy consumption for receiving one bit of data
$E_{\forall}$	Total energy consumption due to data transmission and reception for a given number of rounds
F_{ngr}	Neighboring nodes ID
N_{poly}	Number of nodes in the polygon
CH_{prob}	Probability of a node to be a CH
ρ	Transmission range of node i
ω	Normalizing factor
τ_{min}	Least probability of being a CH
$\delta_{a,b}$	Geographical distance between position a and position b
S	Starting and exiting positions of the UAV in the ROI
T_{CH}	List of final CH and tentative CH
$K^{F_{ngr}}$	List of geolocations of neighboring nodes
DA_{UAV}, LA_{UAV}	Default and least altitude of the UAV
S_Y	Y-axis displacement of the S-path
$S_{Y_{Sensors}}$	Y-axis displacement based on the transmission range of the sensor nodes
$S_{Y_{UAV}}$	Y-axis displacement based on the transmission range of the UAV
ξ_{SENr}	Transmission radius of the sensor
ξ_{UAVr}	Transmission radius of the UAV
b_t	Beaconing time
$UAV_R^{S_Y}$	Effective ground transmission range
DV_{UAV}	Default speed of the UAV
$D_{\forall}$	Total delay for data collection for a given number of rounds
$K_{hull}^{F_{ngr}}$	Set of geolocations of the polygon
γ	Bit count of the largest number
τ_{min}	Lowest probability value of being a CH
N_{iter}	Number of iterations of the clustering algorithm
CH_{locs}	List of CHs' geolocations
C_{max}, C_{min}	Maximum and minimum co-ordinates of a cluster
C_{range}	Range of a cluster
Γ	Fraction co-efficient of co-ordinates ranges
UAV^h, UAV^l	UAV searching positions by adding and subtracting the step sizes respectively
P_t	Transmitted power of a sensor
P_r	Received power of a sensor
δ	Distance
γ_{RSSI}	Threshold limit of RSSI
$P_{loss}(\delta)$	Path loss at distance δ
E_i	Energy consumption of node i
η	Path loss exponent
σ_E	Standard deviation of the energy consumption
σ_{EI}	Standard deviation of energy consumption in the initial position
μ_{CE}	Mean value of the energy consumption of all nodes in a cluster
γ_{RSSI}	RSSI threshold limit
$RSSI_{init}^{C}$	List of RSSI values in a cluster at the initial position
$RSSI_{SI}$	Sum of RSSI values in the initial position
i_{sol}	Initial position for searching
R_n	Number of rounds
O	Set of data collection position
£	Transmission delay from a sensor node to the UAV
S	Starting and final positions of the UAV in the ROI

the positional information once is sufficient. This mechanism does not cause a notable amount of sensor's energy loss.

Static Nodes: The nodes are static. Once deployed, the sensor nodes do not change their positional values anymore. This is a common assumption practiced in the literatures. This assumption is also the major differentiating criteria between mobile ad hoc networks and wireless sensor networks. Node's mobility is also related with location reading criteria. With introducing mobility, the sensor nodes will have to take reading after regular time interval.

Homogeneous Nodes: The sensor nodes assumed in the experiment are homogeneous, which means that they have equal computational power and energy and the same radio communication module with the same transmission range. This assumption is also one of the popular assumptions practiced in [50]. However, to enable EFDC for heterogenous nodes, only the CH selection algorithm should be adapted. However, that is another research issue.

Node Characterization: The deployed sensor nodes can be categorized as CMs or CHs. The role of a sensor node is selected through the clustering algorithm, and no predetermined role is assumed. To design EFDC, we first run a clustering algorithm named CBHEED. There are two goals that the algorithm serves. One is to group the nodes and form the underlying WSN topology, and another is to assign the role of CH or CM to the sensors.

Node Deployment: The nodes are deployed in a completely random manner over the ROI. This assumption is a regular assumption. The performance with planned deployment is better than that with random deployment. As a result, an architecture tested with random deployment will result better than that with other planned deployment [51].

Energy Constraints: Sensors are battery powered and the batteries are not rechargeable. Available nodes in the market are mostly battery-powered. Thus, this is a valid assumption. However, there are studies [52] focused on energy harvesting based on solar power or other energy sources. Nevertheless, energy harvesting in WSNs is a current research issue and the cost will also increase with the added facility.

Adaptive Transmission Power Control: The sensor nodes have the ability to control the transmission power. Transmission power is re-tuned after the data collection position is obtained [53]. Based on this assumption, the sensor nodes will transmit with the required power only. This will reduce the transmission power cost and elongate the lifetime of the WSN. The sensor nodes need to re-tune the transmission power for a single time only when the data collection position is determined.

3) ASSUMPTIONS FOR UAV

Non-Constrained Energy: The UAV has enough energy to complete a single discovery or data collection round. Once the UAV comes back to the launching station, it can be recharged for the next round of operation [19]. This is a valid assumption as the capacity of the UAV energy source (battery) can be changed based on the mission [54]. There are also researches going on for solar-powered UAVs [55].

UAV Type: We utilized a quadcopter instead of a fixed-wing UAV. A quadcopter can easily change its position with the least amount of bending angle and can stay in a stationary position for an arbitrary amount of time [56], [57].

Buffer Capacity of UAV: It is assumed that the UAV is equipped with sufficient memory that can receive and carry all the sensed data from the sensor; thus, buffer overflow is not possible for the UAV. The UAV stores the data for a limited time only. When the UAV goes back to the control center, it offloads all the data and free up its memory. Thus, memory overflow in the later data collection round is not possible, either.

Collision and Obstacle Free Movement: It is assumed that the UAV does not face any obstacle on its way of movement. UAVs are equipped with multiple sensors, which ensures avoidance of any obstacle on its preplanned path [57].

UAV–WSN COMMUNICATION model: The communication model between the WSN and UAV is considered as LOS communication. EFDC assumes a low altitude UAV for data communication with densely deployed sensor nodes. The sensor nodes are further group into cluster and the UAV fine-tunes the data collection positions. As a result, even in case of the presence of any natural obstacle, the UAV will fin-tune the data collection position for higher RSSI, and a LOS communication channel will be established.

RSSI Calculation Ability: The UAV can calculate the RSSI power of the signal received from the ground sensor nodes. It's a trivial assumption for a wireless communication enabled devices. Available sensor nodes are also equipped with this facility. Thus, the RSSI calculation ability for the UAV is a legitimate assumption [58].

Least Flight Height: We assumed that the UAV could detect the least-flying height through an embedded sensor such as a sonar sensor or LDR sensor. This assumption is also related to the obstacle-free movement capability of the UAV. The ground is just another obstacle, where the UAV cannot go beyond that [57]. While the tabu search algorithm works, the LA_{UAV} is determined by this sensing ability.

4) MAC PROTOCOL

The EFDC scheme uses carrier-sense multiple access (CSMA) for hello packet transmissions. For data packet transmission, it uses a time-division multiple access (TDMA) protocol. We kept the MAC operation similar to the MAC protocol proposed for UWSNs in [23]; however, different from that in [23], we did not consider any priority for the nodes.

B. COMMUNICATION MODEL

Sensor nodes are deployed randomly throughout the ROI. The assumed ROI is a remote region, in which no infrastructure is available for the sensor nodes to transmit their data directly to a sink node or to any NCC. EFDC will be equally efficient even in the urban area with the added

facility of infrastructure-less design. Each sensor node in set $N = 1, 2, 3, \ldots, |N|\}$ has coordinates $K = k_1, k_2, k_3, \ldots, k_{|N|}$, where, $K_{|N|} \in \mathrm{R}^{3\times 1}$. However, for the flat terrain, two-dimensional consideration is enough as the sensor nodes do not have a considerable amount of height. On the other hand, the sensor nodes deployed for monitoring structures are readily suited with our present design without any kind of modification as the sensors deployed for such purpose have a significant amount of differences in the height parameter. The EFDC communication mechanism is divided into three phases: initialization, discovery, and data collection.

In the initialization phase, the CBHEED clustering algorithm forms clusters and elects CHs. The positions of the CHs are used as initial positions for the tabu search algorithm for data collection. Different from the conventional approaches, neither NCC nor UAV has any prior knowledge about the CHs' positions. The CHs' positions work as the pre-optimized position for UAV's data collection. As a result, the runtime of the tabu search algorithm significantly decreases. In EFDC, the extra workload of CHs is reduced by sending the sensed data directly to UAV. On the other hand, in the conventional UWSN data collection scenarios, the UAV collects data from CHs in WSNs [28], [38]–[41], [45]–[47], [55]. Because of the direct communication paradigm, the energy consumption in EFDC is well balanced and the re-election of CHs are redundant. A one-time clustering technique also reduces the number of exchanged control packets.

In the discovery phase, the UAV determines the position of CHs and obtains sub-optimal data collection positions with the help of the tabu search algorithm. At first, the UAV locates the CHs with the help of hello packets. The UAV follows the S-path mobility model [59] while discovering the positions of CHs. It keeps track of its distance with all discovered CH positions. When a UAV reaches the least distance from its path to a CH location, it visits the CH's position physically. The modified S-path mobility model is described in detail in the following subsection. After that, the UAV runs our proposed modified tabu search algorithm to find out a sub-optimal position to collect data from the cluster. The optimization is done to reduce the energy consumption and to ensure a balanced energy consumption of the sensors in the cluster.

In the data collection phase, the UAV follows the derived trajectory and collects data from the clusters. At first, the UAV forms a traveling salesman problem (TSP) with the sub-optimal data collection positions obtained with the tabu search. The TSP is solved with the help of a modified GA [60]. This modified version of GA ensures the avoidance of pre-mature convergence and a better run-time complexity. Ultimately, the shortest trajectory ensures fast data collection, which is the second objective of the EFDC procedure.

C. UAV MOBILITY MODEL

This subsection describes the parameter of the S-path mobility model used in this study. In the discovery phase of the EFDC scheme and in other cases where the UAV does not know the CHs' locations, the UAV follows the S-path mobility model. In the proposed scheme, the UAV starts searching for the CHs from the initial position of the ROI:

$$UAV_{start} = \begin{bmatrix} 0 \\ 0 \\ DA_{UAV} \end{bmatrix}, \tag{1}$$

where the UAV's initial position is denoted as UAV_{start} and the default altitude of the UAV is DA_{UAV}. The final or exiting position of the UAV after completing the search can be denoted as:

$$UAV_{final} = \begin{bmatrix} RoI_x \\ RoI_y \\ DA_{UAV} \end{bmatrix}, \tag{2}$$

where UAV_{final} is the final position, and RoI_x and RoI_y are the maximum x-axis and y-axis values of the ROI, respectively. When the UAV reaches the boundary of the x-axis from the initial points, the UAV jumps an S_Y amount of space according to the y-axis. The value of S_Y can be calculated as

$$S_Y = \begin{cases} S_{Y_{Sensors}} * 2, & S_{Y_{Sensors}} \leq S_{Y_{UAV}} \\ S_{Y_{UAV}} * 2, & S_{Y_{UAV}} < S_{Y_{Sensors}} \end{cases}, \tag{3}$$

where $S_{Y_{Sensors}}$ and $S_{Y_{UAV}}$ are the y-axis displacements based on the transmission range of the sensor nodes and the UAV, respectively. $S_{Y_{Sensors}}$ can be derived based on the following formula:

$$S_{Y_{Sensors}} = \sqrt[2]{\xi^2_{SENr} - DA^2_{UAV}}, \tag{4}$$

where ξ_{SENr} is the transmission radius of the sensor. The value of $S_{Y_{UAV}}$ can be calculated by

$$S_{Y_{UAV}} = \sqrt[2]{\xi^2_{UAVr} - DA^2_{UAV}}, \tag{5}$$

where ξ_{UAVr} is the transmission radius of the UAV. The calculation of S_Y is illustrated in Fig. 2.

The hello packet broadcast interval is set in a way that every sensor will be inside the transmission range of the UAV for at least a single transmission. To accomplish this task, we set the broadcast time as follows:

$$b_{prev} < b_{next} < (b_{prev} + (\frac{UAV_R^{S_y/2}}{DV_{UAV}})), \tag{6}$$

where b_{prev} is the time when the last beacon was sent and b_{next} denotes the time when the next beacon will be sent. $UAV_R^{S_Y/2}$ is the effective ground transmission range after half of the S_Y distance according to the y-axis, and DV_{UAV} is the default speed of the UAV. $\frac{UAV_R^{S_Y/2}}{DV_{UAV}}$ indicates the time duration that a sensor will be inside the transmission range of the UAV, where the sensor node's distance is half of S_y from the parallel line of the S-path mobility. The distance $\frac{S_y}{2}$ is the furthest distance a sensor node can be, from the predetermined UAV's path. To ensure that all the sensor nodes will receive the beacon packet from the UAV, the inequality of (6) must be satisfied.

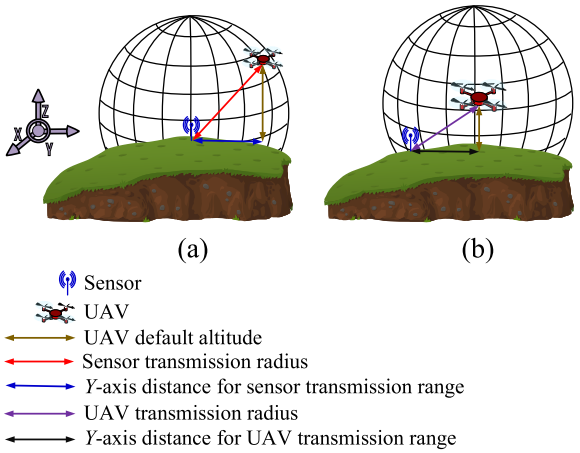

FIGURE 2. Calculation of S-path y-axis difference based on (a) sensor transmission range and (b) UAV transmission range.

TABLE 2. Working Procedure of EFDC Scheme.

Phase	Work to do	Algorithm
Initialization phase	Grouping the sensors into clusters, electing a CH with a better central bias	CBHEED clustering algorithm
Discovery phase	UAV scans the ROI and finds suboptimal data collection positions	Modified tabu search algorithm
Data collection phase	UAV collects sensors' data from the obtained suboptimal positions by following the optimized trajectory	Modified genetic algorithm

IV. ENERGY-EFFICIENT AND FAST DATA COLLECTION

In this section, the EFDC scheme is presented according to the phases presented in Table 2. The CBHEED algorithm is shown as a part of the initialization phase. In the discovery phase, following the S-path mobility model, the UAV discovers the CH positions and obtains the sub-optimal position for data collection from each cluster. The sub-optimal positions are obtained with the help of the modified tabu search algorithm. As a part of the data collection phase, the outline of the GA is given. The modified GA is used to obtain the shortest trajectory among the data collection positions.

A. CLUSTERING

The clustering algorithm is used in the EFDC scheme to group the underlying WSN in a distributed manner. The CH positions are used as the initial positions for the suboptimal position search algorithm for data collection. As there is no infrastructure available to assist the sensor nodes to form the cluster, the clustering technique has to be fully distributed for the application scenario. The HEED [61] algorithm is a well-known algorithm owing to its energy-efficient CH selection technique. We modified the original HEED algorithm and proposed the CBHEED algorithm, which fulfills the clustering need for our scenario.

In the EFDC scheme, the clustering process occurs only once. The UAV also searches the suboptimal positions for data collection only once based on the CH position. Hence, it is important for the clustering algorithm to determine the node whose geographical position is superior compared to other neighboring nodes. A node with a superior geographical position means that its cumulative distance and the fluctuations among the distances with the nodes in a cluster are also minimal. Similar to the original HEED clustering algorithm, the CBHEED algorithm also forms the cluster in three steps. Algorithm 1 presents the outline of the proposed CBHEED algorithm. The working procedure of Algorithm 1 is elaborated in the following subsections.

1) INITIALIZATION

In the initialization step, every node formulates its neighboring table by broadcasting an initial hello message. Along with the node's ID, this hello message also contains the node's geographical position. However, the RSSI level plays the most important role here for calculating the neighboring list. RSSI value must be within the threshold of the sensitivity level of a particular node. To form the neighbor list, the geo-location of a node is not considered. For example, in [62], it is mentioned that the sensitivity level of MICAz node is -94 dBm, which uses CC2420 RF transceivers. With the help of the RSSI value obtained from the hello messages, the receiving nodes formulate the F_{ngr} list, which contains the neighboring node IDs. Then, the sensor nodes calculate their central bias based on the monotone chain convex hull algorithm and the Paul Bourke's equation for centroid calculation [63].

The outline of the monotone convex hull algorithm is given as Algorithm 2. It should be mentioned that for calculating the polygon and the centroid, we took only the x and y coordinates of the neighbors. According to CBHEED, the nodes' RSSI value should be more than a minimum threshold value in order to be a neighbor. As a result, the nodes with lower RSSI value will be automatically excluded in the neighboring list formation process and do not participate in the subsequent calculation. Even though the hilly areas have differences in the z axis value, the UAV optimizes the data collection position in 3D space (Algorithm 4). Considering 3D space might be necessary if the CH is responsible for collecting data. In EFDC, however, the responsibility for data collection is not given to the CH but to the UAV. So, considering the 3D space while clustering is unnecessary in EFDC. The monotone chain convex hull [64], [65] algorithm is used to form the polygon by considering the neighboring nodes and it returns a sorted array of points of the polygon. The points are then fed into Paul Bourke's equation for calculating the exact centroid position of the polygon.

To determine the centroid, the area should be calculated first based on the derived coordinates from the convex hull algorithm. The area can be calculated using with the following equation:

$$A = \frac{1}{2} \sum_{i=0}^{N_{poly}-1} (X_i Y_{i+1} - X_{i+1} Y_i) , \tag{7}$$

where A denotes the area of the polygon, X is the sorted x-axis list of the sensor node's geolocation on the edge of

the polygon, Y contains the sorted y-axis list of the sensor node's geolocation on the edge of the polygon, and N_{poly} is the number of nodes in the polygon. The x and y coordinates of the centroid can be calculated based on the following, respectively.

$$Center_x = \frac{1}{6A} \sum_{i=0}^{N_{poly}-1} (X_i + X_{i+1})\,(X_i Y_{i+1} - X_{i+1} Y_i), \quad (8)$$

and

$$Center_y = \frac{1}{6A} \sum_{i=0}^{N_{poly}-1} (Y_i + Y_{i+1})\,(X_i Y_{i+1} - X_{i+1} Y_i), \quad (9)$$

where $Center_x$ and $Center_y$ represent the x and y coordinates of the polygon center. The probability of a node to become a CH can be assigned using the following equation:

$$CH_{prob} = \max\left((1 - \frac{\delta_{a,b}}{\rho} \times \omega), \tau_{min}\right), \quad (10)$$

where CH_{prob} is the probability of a node to become a CH, ρ indicates the transmission range of a node, ω is a normalizing factor on which the number of iterations of the clustering algorithm is dependent, and τ_{min} is the minimum value assigned in the nodes. When the central bias value becomes lower than a certain threshold value, τ_{min} is assigned as the CH_{prob} of a node. $\delta_{a,b}$ indicates the Euclidian distance between geographical positions a and b. Here, positions a means the position of the examining node, where $a = \{a_x, a_y\}$ and b corresponds to the polygon's center, where $b = \{b_x, b_y\}$.$\delta_{a,b}$ is calculated using the Euclidian distance formula as indicated below:

$$\delta_{a,b} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}. \quad (11)$$

The cost of a node is determined by the number of adjacent nodes. Similar to the original HEED algorithm, the node degree and cumulative distance of the neighbors are taken into consideration for calculating the cost. Fig. 3 illustrates an example of a centroid calculation after applying algorithm 2 and (11). Fig. 3(a) displays a greater distance, whereas Fig. 3(b) shows a lesser and better example of $\delta_{a,b}$.

The residual energy is not taken into consideration because the deployed nodes are homogeneous. Initially, the energy level inside all the sensors is the same. In the EFDC scheme, clustering is done only once. In contrast to the conventional clustering, the CH does not perform any extra work in the proposed scheme. The energy consumption is the same for all nodes. Depending on the CH_{prob} value, the nodes declare themselves as the tentative CH denoted as $CH_{tentative}$ and initially, all the sensor's CH_{final} flags, that is, $bool\_CH_{final}$ are set to false.

2) ITERATION

The second step of the clustering algorithm is called the iteration step as given in Algorithm 1. In this step, the nodes compare the cost of the neighboring nodes with their own

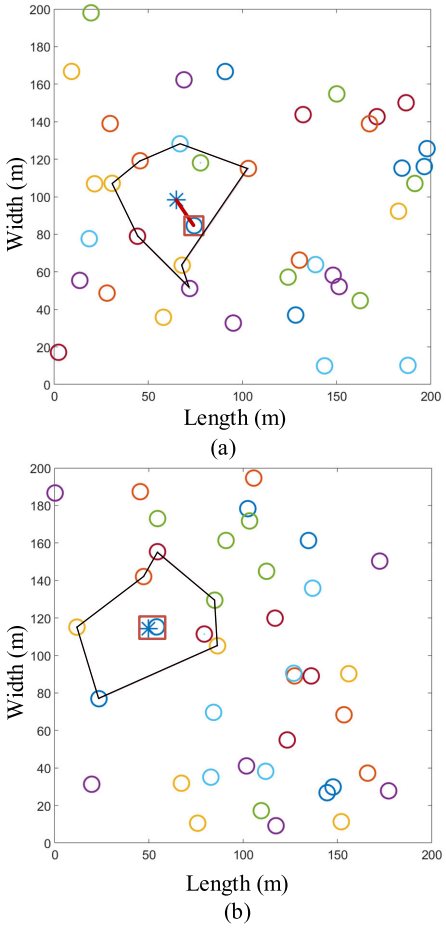

FIGURE 3. Bias examples: (a) bad centroid bias and (b) good centroid bias.

cost. The least cost node is selected as the temporary CH, expressed as $CH_{selected}$ from the T_{CH} list. The sensor selects its CH by receiving a final CH message from a CH_{final} or the sensor claims itself as a CH_{final} by its own. If a node finds itself having the least cost and the value of CH_{prob} is also 1, then the node sets its $bool\_CH_{final}$ value to true and broadcasts a CH final message to all neighbors. If the CH_{prob} is less than 1, then the node claims itself as the $CH_{tentative}$ and broadcasts a tentative CH hello message to its neighbors.

When the T_{CH} list of a node is empty and CH_{prob} value of the node is 1 then the CH_{final} flag of a sensor becomes true and the node broadcasts a CH final message. The T_{CH} gets updated every time a sensor node receives a new declaration of a node as a $CH_{tentative}$ or CH_{final}. The operation can be expressed by the following equation:

$$T_{CH} = \{CH\ sin\ step(i-1) \cup CHs\ in\ step(i)\}. \quad (12)$$

Apart from the above cases, a node might not have any CH in its vicinity and the CH_{prob} value might not be 1.

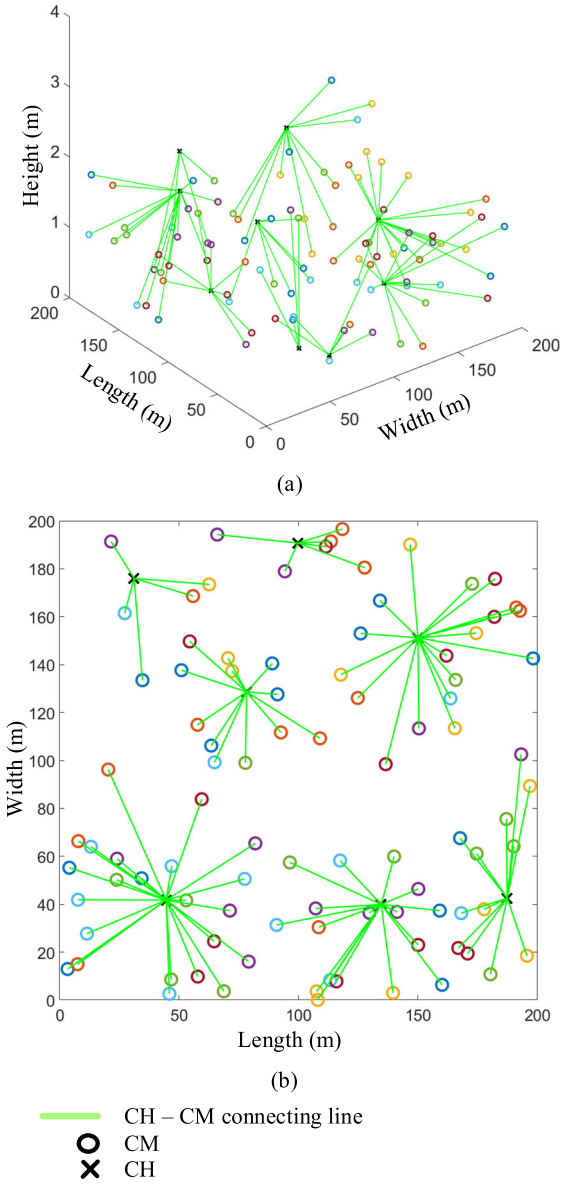

FIGURE 4. Illustration of clustering results in an example WSN: (a) 3D view of the clustering outcome and (b) top view of the clustering algorithm in 2D.

In such cases, the node declares itself as $CH_{tentative}$ based on a random value and broadcasts its status. In every iteration, the nodes increase the CH_{prob} value by multiplying with two. If the CH_{prob} value becomes greater than 1, then the corresponding node exits the iteration phase. In the iteration phase, either the sensor node selects a CH or it declares itself as the CH and quits the iteration round.

3) FINALIZATION

In the finalization step, the nodes check if their $bool\_CH_{final}$ is true or not. If it is not, then they find the least cost CH, namely, $CH_{selected}$ from the T_{CH} list and send a cluster join request to the least cost CH. Otherwise, the nodes broadcast hello packets by setting their CH_{status} to $CH_{tentative}$. Upon receiving the cluster join request, the CH_{final} adds the

Algorithm 1 CBHEED

Input: $\{K^{F_{ngr}} | K_1^{F_{ngr}}, K_2^{F_{ngr}}, K_3^{F_{ngr}}, \ldots, K_n^{F_{ngr}}$ contains the geolocations of sensor node's neighbors$\}$

Output: CH_{final}

Initialization

1. $F_{ngr} \leftarrow$ {Neighbor's list based on RSSI value}
2. Broadcast cost to all nodes $\in F_{ngr}$
3. Forms polygon using algorithm 2 (Monotone chain convex hull algorithm)
4. Calculates the area of the polygon (A) using (7)
5. Calculates $Center_x$ and $Center_y$ axis of the polygon using (8) and (9)
6. Calculates difference between node's geo-position and polygon centroid position ($\delta_{a,b}$) using (11)
7. Assigns CH_{prob} value using (10)
8. $bool\_CH_{fina} \leftarrow False$

Iteration

1. while ($True$)
2. if (empty (T_{CH})isnotequalto$True$)
3. $CH_{selected} \leftarrow$ min_cost(T_{CH})
4. if ($CH_{selected}.ID$ is equal to NID)
5. if (CH_{prob} is equal to 1)
6. broadcast_ch_info(NID, CH_{status}[[$space$]] $\leftarrow$ CH_{final}, $cost$)
7. $bool\_CH_{final}$[[$space$]] $\leftarrow$ $True$
8. else
9. broadcast_ch_info(NID, CH_{status}[[$space$]] $\leftarrow$ [[$space$]]$CH_{tentative}$, $cost$)
10. end if
11. end if
12. else if (CH_{prob} is equal to 1)
13. broadcast_ch_info(NID,$CH_{status} \leftarrow CH_{final}$, $cost$)
14. $bool\_CH_{final}$[[$space$]] $\leftarrow$ $True$
15. else if (Rand(0,1) is less than or equal to CH_{prob})
16. broadcast_ch_info(NID,$CH_{status} \leftarrow$ [[$space$]]$CH_{tentative}$, $cost$)
17. end if
18. $CH_{prev} \leftarrow CH_{prob}$
19. $CH_{prob} \leftarrow \min(CH_{prob} \times 2, 1)$
20. if (CH_{prev} is equal to 1)
21. break
22. end if
23. end while

Finalization

1. if ($bool\_CH_{final}$ is equal to $False$)
2. if (contain_Final_CH(T_{CH}) is equal to $True$)
3. $CH_{selected} \leftarrow$ min_cost(T_{CH})
4. cluster_join($CH_{selected}.ID$, NID)
5. else
6. broadcast_ch_info(NID, $CH_{status} \leftarrow$ [[$space$]] $CH_{tentative}$, $cost$)
7. end if
8. else
9. broadcast_ch_info(NID, $CH_{status} \leftarrow$ [[$space$]] CH_{final}, $cost$)
10. end if

requesting node information in the CM table. If the node itself is a CH_{final}, the node sends a CH final hello message to further inform the neighboring nodes about the final CH status. Fig. 4 displays a graphical representation of the outcome of the clustering algorithm. In the figure, the CHs and corresponding CM connections are shown.

4) POLYGON FORMULATION

In the EFDC scheme, the monotone chain convex hull algorithm proposed in [66] is used to form the polygon based on the neighboring nodes of an examining node. The algorithm extends the Graham scan [65] by sorting the selected data points. The algorithm is named as *monotone chain* because the algorithm computes the lower and upper hulls of a monotone chain of points. The pseudocode is given as Algorithm 2.

The algorithm first sorts the sensor nodes based on the geolocation values of their neighboring nodes. $K^{F_{ngr}}$ contains the values of the geolocations of the neighboring nodes. Two lists, namely, U_{list} and L_{list}, contain the points of the upper and lower hulls. For computing L_{list}, a subset ς of the sorted $K^{F_{ngr}}$ is taken with at least two nodes. All members of $K^{F_{ngr}}$ are iterated, and positions with the same directions are added. The added location first gets deleted and the next node's location gets inserted. Constructing U_{list} is achieved in the same manner as for L_{list}. A concatenation operation is done on U_{list} and L_{list} to produce the resulting $K_{hull}^{F_{ngr}}$, which contains the geolocation of the formed polygon.

5) RUNTIME COMPLEXITY OF THE CLUSTERING PROCESS

As Algorithm 2 is a subroutine of Algorithm 1 (CBHEED), the runtime complexity of the CBHEED algorithm depends on both of the algorithms. The runtime complexity for both of the algorithms are discussed as below:

a: RUNTIME COMPLEXITY FOR MONOTONE CHAIN CONVEX HULL ALGORITHM (ALGORITHM 2)

To calculate the central bias, we used the monotone chain convex hull algorithm. This algorithm needs to sort the coordinates of the sensors' geolocations, which is the most expensive process in terms of runtime complexity. By implementing the radix sort, the time complexity can be reduced to $O(\gamma n)$, where γ is the bit count of the largest number and n is the number of elements. For generating L_{list} and U_{list}, the algorithm takes $O(n)$ time. The function remove() takes $O(1)$ time and concat() needs $O(n)$ time to finish. In summary, the time complexity of Algorithm 2 is $O(n)$.

b: RUNTIME COMPLEXITY FOR CBHEED ALGORITHM (ALGORITHM 1)

The runtime of the CBHEED algorithm is similar to that of the original HEED algorithm [61] except for the changes due to the first parameter selection (that is, the central bias calculation). The central bias calculation is used to assign the CH_{prob} value of a sensor node. In addition to the cost of finding CH_{prob}, where the convex hull algorithm works, the remaining part of the initialization step takes $O(n)$ time.

In the worst case, a node will have CH_{prob} of τ_{min}. However, in every iteration, the CH_{prob} is doubled. The maximum number of iterations can be calculated using

$$2^{N_{iter}-1} \times \tau_{min} \geq 1 \tag{13}$$

and

$$N_{iter} \leq \left\lceil \log_2 \frac{1}{\tau_{min}} \right\rceil + 1, \tag{14}$$

where N_{iter} is the number of iterations in the iteration step of the clustering algorithm and τ_{min} is the minimum probability of being a CH. Thus, it is evident that the number of iteratios is constant and $N_{iter} \approx O(1)$. With the maximum number of n CHs, the runtime would be $O(1) \times runtimeofN_{iter}$. As N_{iter} is constant, the runtime of an iteration step is also $O(n)$. Inside the iteration step, the other computations take only constant time.

Algorithm 2 Monotone Chain Convex Hull Algorithm

Input: $\{K^{F_{ngr}} | K_1^{F_{ngr}}, K_2^{F_{ngr}}, K_3^{F_{ngr}}, \ldots, K_n^{F_{ngr}}$ contains the geolocations of sensor node's neighbors$\}$
Output: $\{K_{hull}^{F_{ngr}} \mid K_{hull^1}^{F_{ngr}}, K_{hull^2}^{F_{ngr}}, K_{hull^3}^{F_{ngr}}, \ldots, K_{hull^n}^{F_{ngr}}$ contains the geolocations of sensor nodes, which took part in forming the convex hull$\}$
1.sort (list $K^{F_{ngr}}$ according to the x-axis, in case of a tie using the y-axis)
// the U_{list} and L_{list} will hold the upper and lower hulls accordingly
2. $U_{list} \leftarrow \{\}$
3. $L_{list} \leftarrow \{\}$
4.For $i \leftarrow 1$ to length $(K^{F_{ngr}})$
5. while ($\varsigma[[space]] \subset [[space]]L_{list}$, where n($\varsigma$) is ≥ 2 and $K^{F_{ngr}}[i]$ does not make any counterclockwise turn with the sequence of the last 2 points of L_{list})
6. remove(L_{list} [last element])
7. append($L_{list}[K^{F_{ngr}}[i]]$)
8. end while
9.end for
10.for $i = 1$ to length $(K^{F_{ngr}})$
11. while ($\vartheta \subset U_{list}$, where n($\vartheta$) ≥ 2 and $K^{F_{ngr}}[i]$ does not make any counterclockwise turn with the sequence of the last 2 points of U_{list})
12. remove(U_{list} [last element])
13. append($U_{list}[K^{F_{ngr}}[i]]$)
14. end while
15.end for
16.remove(U_{list} [last_element])
17.remove(L_{list} [last_element])
18. $K_{hull}^{F_{ngr}} = \text{concat}(L_{list}, U_{list})$

In the finalization step, the time complexity is dependent on the number of final CHs found by the nodes. The cluster_join() function completes its operation within $O(1)$ time.

After the aforementioned analysis, it can be concluded that the complexity of the entire clustering technique is $O(n)$.

B. DISCOVERY OF DATA COLLECTION POSITION

In the EFDC scheme, the first round of the UAV is called the discovery phase. This phase has two main goals:

1. discovering the CH locations and
2. finding the suboptimal data collection positions.

In discovering the CH locations, the UAV follows the S-path mobility model and locates the CH locations. In the data collection position search algorithm, a suboptimal position is obtained for every cluster by the proposed modified tabu search algorithm.

1) DISCOVERING THE CH LOCATIONS

For the discovery phase, we assumed that the UAV follows an S-path using (1) and (2) as its initial and final positions, respectively. The working procedure of this phase is given as Algorithm 3. The list CH_{locs} holds all the discovered CH geolocations. The UAV broadcasts a CH finding beacon message to get a reply from the sensor nodes about their CH's position after every beaconing interval. In reply to the hello message, the sensor nodes send their corresponding CH positions back to the UAV. The reply message from the sensor nodes includes the CH's ID and geolocation. The UAV then adds the corresponding information to its buffer memory as discovered CHs.

Meanwhile, the UAV keeps searching for the minimal distance for a specific CH, based on the track of the mobility model. With the help of min_distance_to_a_CH() function, the UAV searches the relative distance with the CHs. After reaching a specific position where the distance of a corresponding CH is minimal, the UAV flies into the position that saves the UAV flight time. The UAV requests for the cluster information using a request cluster information hello message. Upon receiving the packet, the sensors reply the cluster information. This reply message contains the CMs' IDs and their geolocations. The UAV then finds the suboptimal data collection positions. The first parameter is the cluster information and the second parameter is the position of the corresponding CH. After finding the data collection position the UAV goes back to its previous position on the predefined S-path. The beaconing time interval is given in the description of S-path mobility model under section III-C. The pause time is set to according to (6).

Runtime complexity: The runtime complexity of Algorithm 3 depends on that of Algorithm 4 as Algorithm 4 is invoked inside Algorithm 3. Algorithm 4 runs at the predefined number of iterations denoted as $iter_{num}$. Except for calling Algorithm 4, all other components such as the append functions, value assigning task, UAV positioning task, networking task, and decisioning task have the runtime of $O(1)$. However, the outer loop keeps on running based on a continuous interval. Thus, the runtime complexity of Algorithm 3 is $O(n)$ and that of Algorithm 4 is constant.

Algorithm 3 Discovering the CH Locations

Output: $\{O_1, O_2, O_3, \ldots, O_{|C|}$ contains the suboptimal data gathering positions$\}$

Initialization:

1. $CH_{locs}, O \leftarrow \{\}$

Iteration:

2.while (*True*):
3. $CH_{tempPos} \leftarrow$
4. broadcast_CH_search_message()
5. if (receive (CH_{info}))
6. $CH_x \leftarrow$ discovered cluster head x-axis value
7. $CH_y \leftarrow$ discovered cluster head y-axis value
8. $CH_z \leftarrow$ discovered cluster head z-axis value
9. $CH_{tempPos} \leftarrow \{CH_x, CH_y, CH_z\}$
10. end if
11. append(CH_{locs}, $CH_{tempPos}$)
12. if (min_distance_to_a_CH())
13. UAV_acquires_CH_position ()
14. UAV_requests cluster_member_information ()
15. $CM_{info} \leftarrow$ UAV_receives_cluster_member_information ()
16. O_{single} = ALGORITHM 4 (CM_{info}, i_{sol})
17. append(O, O_{single})
18. UAV_acquires_previous_position();
19. end if
20. pause(); // following (6)
21.end while

2) SUBOPTIMAL POSITION SEARCHING ALGORITHM FOR DATA COLLECTION

To improve the quality of data collection position, we applied a modified tabu search algorithm, which returns a moderate solution with a smaller number of iterations. To apply this algorithm in our scenario, a selection mechanism for neighbor positions is necessary.

a: NEIGHBOR SELECTION MECHANISM

To select the neighboring positions for the UAV in 3D space, the UAV calculates the range of axis based on the upper and lower limits of the cluster boundary according to the sensor nodes' geolocations.

$$\begin{bmatrix} C_{maxX} \\ C_{maxY} \\ DA_{UAV} \end{bmatrix} - \begin{bmatrix} C_{minX} \\ C_{minY} \\ LA_{UAV} \end{bmatrix} = \begin{bmatrix} C_{rangeX} \\ C_{rangeY} \\ C_{rangeZ} \end{bmatrix} \tag{15}$$

where C_{rangeX}, C_{rangeY}, and C_{rangeZ} are the corresponding ranges of the x-, y-, and z-axis; C_{maxX} and C_{maxY} are the maximum values of the x-axis and y-axis of a cluster, respectively; DA_{UAV} is the default altitude of the UAV; C_{minX} and C_{minY} are the minimum x- and y-axis values of a cluster, respectively; and LA_{UAV} indicates the least possible altitude of the UAV. The transition step is calculated by taking a fraction of the

ranges. The steps for the corresponding axis are calculated based on the following equations:

$$\begin{bmatrix} C_{rangeX} \\ C_{rangeY} \\ C_{rangeZ} \end{bmatrix} \odot \begin{bmatrix} \Gamma_x \\ \Gamma_y \\ \Gamma_z \end{bmatrix} = \begin{bmatrix} Step_X \\ Step_Y \\ Step_Z \end{bmatrix} \tag{16}$$

where Γ_x, Γ_y, and Γ_z indicate the coefficient percentages of the range that should be taken as the step for the corresponding axis. The number of iterations depends on the values of Γ_x, Γ_y, and Γ_z. It can be observed that, with larger values of Γ, the UAV will require a lesser number of steps, but will produce relatively bad results. To keep a reasonable iteration number in finding a better position, we used Γ_x, Γ_y, and $\Gamma_z = 10\%$ of the entire range. Finding the optimal values of Γ_x, Γ_y, andΓ_z is another research issue, which is out of the scope of this present work.

The next position for iteration is calculated by adding and subtracting the step sizes calculated in (17) and (18) corresponding to their axes. By adding and subtracting the fractional value, the UAV will be able to explore the axes for both positive and negative direction. All possible combinations of the axes give 27 positions in total. However, in (19), we have constituted the exploring positions for the UAV, where the number of the positions are 6 in total. If we observe carefully, it can be seen that the other positions in the 27 positions (excluding the 6 position we are considering) are simply compound positions, which can be composed of the 6 simple positions. For example, we assume that the present location of the UAV is (UAV_X, UAV_y, UAV_z) and the best solution is hidden in a compound space ($UAV_X + Step_x$, $UAV_y - Step_y$, UAV_z). According to our approach, now the UAV may advance first towards the x axis and the new position will be ($UAV_X + Step_x$, UAV_y, UAV_z). Then, from the new position, the UAV may go to the negative direction of y axis, where the coordinate with respect to the first position of the UAV will be ($UAV_X + Step_x$, $UAV_y - Step_y$, UAV_z), which is the same as that of the compound position containing the best result as we assumed before. Besides, the number of searching spaces will also be limited. In the meantime, Algorithm 4 will also prevent the UAV to search in a repeated position.

$$\begin{bmatrix} UAV_x \\ UAV_y \\ UAV_z \end{bmatrix} + \begin{bmatrix} Step_X \\ Step_Y \\ Step_Z \end{bmatrix} = \begin{bmatrix} UAV_x^h \\ UAV_y^h \\ UAV_z^h \end{bmatrix}, \tag{17}$$

where UAV_x^h, UAV_y^h, and UAV_z^h are the three possible UAV searching positions from the previous positions UAV_x, UAV_y, and UAV_z. The other three possible positions can be derived by the following equation:

$$\begin{bmatrix} UAV_x \\ UAV_y \\ UAV_z \end{bmatrix} - \begin{bmatrix} Step_X \\ Step_Y \\ Step_Z \end{bmatrix} = \begin{bmatrix} UAV_x^l \\ UAV_y^l \\ UAV_z^l \end{bmatrix}, \tag{18}$$

where UAV_x^l, UAV_y^l, and UAV_z^l are the other three possible searching spaces for the UAV. All possible searching positions from a previous position matrix are obtained by concatenating the above two matrices, which can be expressed as

$$concat\left(\begin{bmatrix} UAV_x^h \\ UAV_y^h \\ UAV_z^h \end{bmatrix} \cdot \begin{bmatrix} UAV_x^l \\ UAV_y^l \\ UAV_z^l \end{bmatrix}\right) = \begin{bmatrix} UAV_x^h & UAV_x^l \\ UAV_y^h & UAV_y^l \\ UAV_z^h & UAV_z^l \end{bmatrix}. \tag{19}$$

Algorithm 4 utilizes the matrix derived from (19). By following a greedy process, the UAV selects the best position based on the evaluation of the objective function. We cleverly proposed the testing positions by keeping two objectives in mind. The first objective is to reduce the number of search spaces and the second is to include the best state from all possible states.

b: PROBLEM FORMULATION FOR FINDING SUB-OPTIMAL DATA COLLECTION POSITION

We formed the objective function based on the RSSI values of the sensors from the UAV. The RSSI value has been used as one of the key parameters in many studies [60], [61], [62]. The UAV changes its position physically and detects the RSSI values of the sensors. We considered the log-distance propagation model [70], which is an extension of the Friis free space model [71]. The simplest equation for calculating the RSSI value can be expressed as follows [72]:

$$P_r = P_t * \left(\frac{1}{\delta}\right)^{\eta}, \tag{20}$$

where P_r is the power received, P_t is power transmitted from the sender, δ denotes the distance, and η is the path loss exponent. The value of η differs from 1.6 to 6 [73]. In [74], the authors have done a test-bed experiment and found that, in near-ground communication, the value of η differs from 2.45 to 3.40 in an outdoor environment with obstacles. The authors in [74] used the CC2420 transceiver to conduct the experiment. In our case, however, we can safely assume that the value of η is 2 for UAV-to-sensor communication as the probability of LOS communication between UAV and sensor nodes are high. For sensor-to-sensor communication, the value of η is 2.45-3.40 for, as obtained in the experiment. In practice, the UAV will sense the RSSI value and it will execute Algorithm 4 based on the value. By taking the logarithm of both sides, we obtain [72]

$$10 \log P_r = 10 \log P_t - 10\eta \log \delta. \tag{21}$$

If we express P_r in dB as $RSSI$ and $10\eta \log \delta$ as the path loss, the equation can be rewritten as [75]

$$RSSI = P_t - P_{loss}(\delta) \text{ indBm}, \tag{22}$$

where P_{loss} denotes the path loss expressed in dBm. The log-distance path loss can be described as [70]

$$P_{loss}(\delta) = P_{loss}(\delta_o) + 10\eta \log\left(\frac{\delta}{\delta_o}\right), \tag{23}$$

where $P_{loss}(\delta)$ indicates the path loss at distance δ, and $P_{loss}(\delta_o)$ is the path loss at a reference distance δ_o. Replacing the value of $P_{loss}(\delta)$, we can rewrite (22) into

$$RSSI = P_t - \left(P_{loss}(\delta_o) + 10\eta \log\left(\frac{\delta}{\delta_o}\right)\right). \tag{24}$$

Usually, δ_o denotes one unit of distance. By updating the value of δ_o in (24), the following equation can be obtained:

$$RSSI = P_t - P_{loss}(\delta_o) - 10\eta \log(\delta). \tag{25}$$

The power perceived by a receiver from a reference distance can be expressed by

$$A = P_t - P_{loss}(\delta_o), \tag{26}$$

where A denotes the perceived power at a reference distance δ_o. Hence, the RSSI equation can be rewritten as

$$RSSI = A - 10\eta \log(\delta). \tag{27}$$

If the distance between the transmitter and the receiver increases, $P_{loss}(\delta)$ increases and the $RSSI$ value decreases. $RSSI$ is a function of the position constructed by the 3D position of the UAV. According to the energy model used in [76], the energy consumption of the sensor nodes for data transmission depends on the distance. Therefore, the objective function of the UAV searching procedure can be written in the following format:

$$f(x, y, z) = \max\left(\frac{1}{RSSI_{SI}} \sum_{i=1}^{|C_n|} RSSI_i + \left(1 - \frac{\sigma_E}{\sigma_{EI}}\right)\right), \tag{28}$$

where σ_E expresses the standard deviation of the energy consumption of the nodes in a cluster and $|C_n|$ denotes the number of sensor nodes in a cluster. $RSSI_{SI}$ and σ_{EI} means the sum of RSSI values and the value of standard deviation of energy consumption, respectively, in the initial position of the searching mechanism in a cluster.

The constraints of (28) are as follows

$$RSSI_i \leq \gamma_{RSSI}, \tag{29}$$

$$C_{minX} \leq UAV_x \leq C_{maxX}, \tag{30}$$

$$C_{minY} \leq UAV_y \leq C_{maxY}, \tag{31}$$

and

$$LA_{UAV} \leq z_{UAV} \leq DA_{UAV}. \tag{32}$$

Constraint (29) states that for a single node, the $RSSI$ value must not be less than the threshold limit γ_{RSSI}. This constraint also ensures that the value σ_E must not go higher than σ_{EI}. The γ_{RSSI} can be expressed by the following equation:

$$\gamma_{RSSI} = \min\left(RSSI_{init}^{C_n}\right), \tag{33}$$

where $RSSI_{init}^{C_n}$ is the list of initial RSSI values in cluster C_n. Constraint (30) states that the UAV cannot select a position with UAV_x, which is out of the cluster's x-axis boundary of C_{minX} and C_{maxX}. Constraint (31) indicates that the y-axis value UAV_y must be inside the boundary expressed by C_{minY} and C_{maxY}, and (32) requires that the z_{UAV} value must be within LA_{UAV} and DA_{UAV}. LA_{UAV} is the least possible altitude that a UAV can fly, which can be achieved based on internal height sensing capability.

The objective function for finding the suboptimal position for data collection is divided into two parts. The first portion of the function is formulated to find a place where the value of RSSI is the maximum in a cluster. The second part of the objective function states that the position should not only increase the cumulative RSSI value but also minimize the standard deviation σ_E of the energy consumption for all nodes in a cluster. We assume that the UAV can measure the RSSI value based on (27), and it can also estimate the energy consumption of the sensor nodes in a cluster. Maximizing the RSSI will reduce the energy consumption in a cluster, and minimizing σ_E will result in a more balanced energy consumption of the sensor nodes. RSSI values can be calculated as follows:

$$\sum_{i=1}^{|C_n|} RSSI_i = \sum_{i=1}^{|C_n|} (A - 10\eta \log(\delta)) \tag{34}$$

As presented before, δ represents the distance between the transmitter and the receiver. In the EFDC scheme, we measure the RSSI value of the sensor nodes from the UAV. Therefore, δ can be expressed in terms of the Euclidian distance between the UAV and the sensors nodes. Consequently, (34) can be written as

$$\sum_{i=1}^{|C_n|} \left(A - 10\eta \log \left(\frac{\sqrt{(x_i - UAV_x) + (y_i - UAV_y) + (z_i - UAV_z)}}{\delta_o}\right)\right), \tag{35}$$

and σ_E can be expressed as

$$\sigma_E = \sqrt{\frac{(E_i - \mu_{CE})^2}{|C_n|}}, \tag{36}$$

where E_i is the energy consumed by a sensor node, and μ_{CE} is the mean value of the energy consumption of all nodes in a cluster. μ_{CE} can be calculated based on the following formula:

$$\mu_{CE} = \frac{1}{|C_n|} \sum_{i}^{|C_n|} E_i. \tag{37}$$

C. MODIFIED TABU SEARCH ALGORITHM

The tabu search algorithm searches the selected neighboring coordinates as discussed above and chooses the best neighboring position greedily. The pseudocode of the modified tabu search algorithm is given as Algorithm 4.

The initial solution i_{sol} is selected as the initial position for the search mechanism, which is the CH position of the corresponding cluster. to_Visit_Neighbor is a queue that contains the calculated neighboring coordinates with the help of the calculate_neighboring_coordinates() function. This function selects the neighboring position based on the fractional value $Step_X$, $Step_Y$, and $Step_Z$ given as (16). The UAV iterates through all the neighboring positions and calculates the fitness function value by (28), except for the positions that the UAV has already visited. This technique is adopted from the core concept of the tabu search algorithm [77].

Algorithm 4 Suboptimal Position Search Algorithm for Data Collection

Input: CM_{info} = cluster information i_{sol} = selected CH position
Output: O_{pos} = best neighboring position
1. $O_{pos} \leftarrow [[space]]i_{sol}$
2.$count \leftarrow 0$
3. $loc_{visited} = \{\}$
4.while ($count$ is less than $iter_{num}$)
5. to_Visit_Neighbor$\leftarrow$ calculate_neighboring_co-ordinates (O_{pos})
6. $iter\_best_{pos} \leftarrow O_{pos}$
7. for $j \leftarrow$ ç1 till length(to_Visit_Neighbor)
8. $p_{sol} \leftarrow$ to_Visit_Neighbor(j)
9. if ($loc_{visited}$ does not contain p_{sol})
10. set_UAV_coordinates(p_sol);
11. UAV_broadcast _beacon_request(CM_{info});
12. UAV_receive_beacon_and send_acknowledges();
// Evaluates the value of the fitness function based on (28)
13. if (f (p_{sol}) is greater than or equal to f($iter\_best\_pos$))
14. $iter\_best_{pos} \leftarrow p_{sol}$
15. end if
16. append($loc_{visited}$, p_{sol})
17. end if
18. end for
19. if ($iter\_best_{pos}$ is equal to or less than O_{pos})
20. Break;
21. else
22. $O_{pos} \leftarrow iter\_best_{pos}$
23. end if
24. count $\leftarrow$count+1
25.end while

In every designated neighboring position, the UAV broadcasts a request for a beacon packet from specific cluster nodes. In reply, the sensor nodes send beacon signals from which the UAV calculates the RSSI strength for that specific position using (27). After receiving the beacon reply, the UAV unicasts an acknowledgement packet to the sensor node. The visited positions are recorded after every successful visit to the designated neighboring places and inserted into list $loc_{visited}$. After comparing with all the neighboring values, the UAV selects the best neighboring position as its next position. The best position is updated if any better solution is found; else, the loop terminates. The algorithm iterates until an exact number of iterations or the local optimum is found.

Runtime complexity: The number of iterations is fixed for the tabu search algorithm, and it may converge in a suboptimal position before reaching the maximum number of iterations. The inner loop in Algorithm 4 will be executed based on (19). The number of search operations is also limited to a constant number and it will be at most 6. Also, the UAV positioning task, value assigning task, appending operation, networking operation, and decision-making process take constant time. In summary, we can say the runtime complexity of the entire algorithm is at most the number of $iter_{num} \times 6$, where $iter_{num}$ is the maximum limit of iterations. As a result, the runtime complexity of this algorithm is constant (i.e., $O(1)$).

D. DATA COLLECTION

In the data collection phase, the UAV first computes the shortest trajectory by applying the modified GA [60]. The UAV follows the shortest trajectory for the rest of the data collection run. After computing the trajectory, the UAV goes to each derived data collection position and collects data from a specific cluster. In the discovery phase, Algorithm 3 produces a list containing the optimized positions for data collection. Based on this list, the UAV applies the modified GA from [60] and searches for the shortest trajectory for data collection. Fig. 5 shows the optimized trajectory calculated using the GA. Only top view of the trajectory is given in the figure to make it more comprehensible. The three-dimensional version of the optimal trajectory is cumbersome to understand. The modifications done in the different phases of GA are described further in the following subsection.

GA: The GA tries to find the best solution of a fitness function by implementing the metaphor ''survival of the fittest.'' The algorithm uses an evolutionary technique to discard the low fit values and tries to incorporate the best fit value inside the fixed size population.

The fitness function used to find the shortest trajectory can be given as:

$$g(O) = \delta_{(S,O_1)} + \delta_{(O_{|O|},S)} + \sum_{i=1}^{|O|} \delta_{O_i,O_{i+1}} \tag{38}$$

where, $\delta_{(S,O_1)}$ denotes the distance from the entering position S to the first data collection position O_1. $\delta_{(O_{|O|},S)}$ denotes the distance from the last data collecting position to the exiting point. $g(O)$ represents the entire distance that the UAV will travel to collect data in the ROI. In the first step of the GA, random solutions are being generated based on the suboptimal data collection positions. The metaphor chromosome is used to represent a solution. The major operations of the GE can be divided into crossover, mutation and selection. Modifications in all three stages are given below:

Crossover: In the crossover operation, extended partial mapped crossover (EPMX) policy is considered [60]. In this operation a pair of new chromosomes (CR) are created by crossing two parents CR. EPMX operation can be divided into five steps. At first, EPMX finds a crossover region by taking an arbitrary position. After that, the chromosomes are divided into the two parts namely, crossover region and match region. Then, the EPMX sorts and scan the match region to find the non-identical data collection position. the exchange policy is obtained from the non-identical corresponding positions.

Based on the exchange policy, data collection positions are changed in the crossover region and new chromosomes are

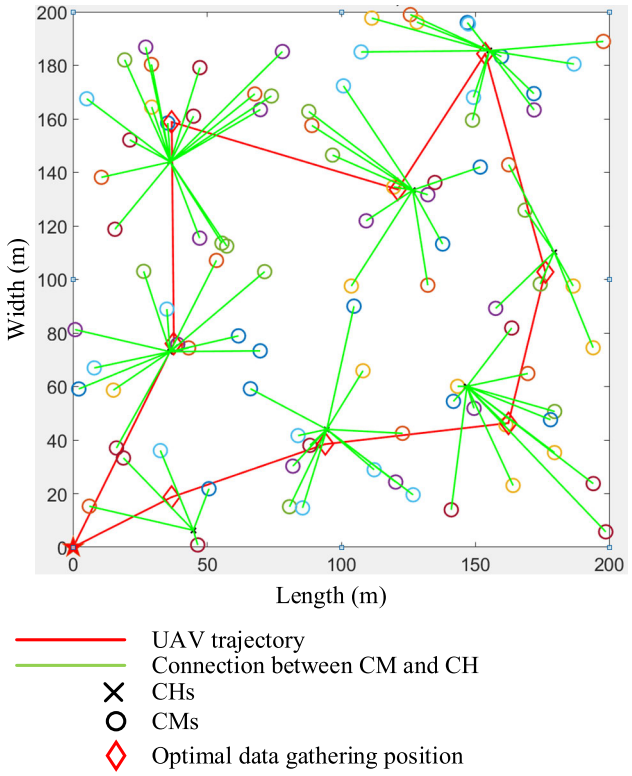

FIGURE 5. Illustration of the optimized trajectory of the UAV for data collection.

created. Unlike [60], our initial and final positions are not variable. A step by step example is given below:

Input: Taking two chromosomes for crossover operation

$$CR1 : \{o_s, o_5, o_3, o_2, o_4, o_1, o_8, o_9, o_7, o_{10}, o_6, o_s\}$$
$$CR2 : \{o_s, o_7, o_4, o_9, o_3, o_8, o_6, o_1, o_2, o_5, o_{10}, o_s\}$$

Step 1: Find a random crossover position

Crossover position: 6

Step 2: Divide each chromosome into match region and crossover region based on crossover position.

Match region:

$$CR1 : \{o_s, o_5, o_3, o_2, o_4, o_1\}$$
$$CR2 : \{o_s, o_7, o_4, o_9, o_3, o_8\}$$

Crossover region:

$$CR1 : \{o_8, o_9, o_7, o_{10}, o_6, o_s\}$$
$$CR2 : \{o_6, o_1, o_2, o_5, o_{10}, o_s\}$$

Step 3: Obtain the exchange policy

Matching operation:

$$CR1 : \{o_s, o_3, o_4, o_1, o_2, o_5\}$$
$$CR2 : \{o_s, o_3, o_4, o_7, o_8, o_9\}$$

Exchange policy: $1 \leftrightarrow 7, 2 \leftrightarrow 8, 5 \leftrightarrow 9$

Step 4: Apply the exchange policy into both of the chromosomes crossover region.

Exchange policy applied:

$$CR1' = \{o_s, o_5, o_3, o_2, o_4, o_1, |o_2, o_5, o_1, o_{10}, o_6, o_s\}$$
$$CR2' = \{o_s, o_7, o_4, o_9, o_3, o_8, |o_6, o_7, o_8, o_9, o_{10}, o_s\}$$

Step 5: Exchange crossover region and new chromosomes are created

Crossover region exchanged:

$$CR1'' = \{o_s, o_5, o_3, o_2, o_4, o_1, o_6, o_7, o_8, o_9, o_{10}, o_s\}$$
$$CR2' = \{o_s, o_7, o_4, o_9, o_3, o_8, o_2, o_5, o_1, o_{10}, o_6, o_s\}$$

Mutation: In this operation, the selected chromosomes are mutated by their own and new chromosomes are created. The newly created chromosomes or solutions are expected to perform better for the fitness function and prevent the premature convergence. The mutation operation adopted in EFDC can be divided into four steps. The first step is to generate a random position for mutation. Then, a random element is taken from the chromosome in the second step. In the third step, the randomly chosen element is inserted inside the randomly chosen position. Lastly, the previous element inside the randomly chosen position is taken and inserted in the location of randomly chosen element. A step by step example is given as follows:

Input: Taking one chromosome for mutation operation.

$$CR1 = \{o_s, o_7, o_4, o_9, o_3, o_8, o_2, o_5, o_1, o_{10}, o_6, o_s\}$$

Step 1: Select a random position.

Random position: 10

Step 2: Select a random element from the chromosome.

Radom data collection position: o_8

Step 3: o_8 is inserted into position 10 and o_{10} is stored

$$CR1 = \{o_s, o_7, o_4, o_9, o_3, o_8, o_2, o_5, o_1, o_8, o_6, o_s\}$$

Step 4: o_{10} is inserted in the previous position of o_8

$$CR1' = \{o_s, o_7, o_4, o_9, o_3, o_{10}, o_2, o_5, o_1, o_8, o_6, o_s\}$$

Selection: The third stage of GA is called selection, where some chromosomes are chosen from all the population for next round of evaluation. To ensure population diversity, a discrete roulette operator is used to select chromosomes as in [60]. In this mechanism, the percentage of selection probability is magnified thus, the chances of getting selected for poor performing chromosomes increases.

The stopping criterion for the GA is fixed as the "stall iteration limit." In this mechanism, if the GA procedure is unable to produce any better solution for a specific number of iterations, we stop the procedure and select the best chromosome that occurred so far.

Runtime Complexity: The TSP is a NP-complete problem, and the runtime complexity for finding the shortest trajectory based on the TSP problem is $O(n!)$, which is not a feasible option for real-life application. In order to minimize the time complexity of finding the shortest trajectory, a modified GA [60] is applied. The expected runtime of GA is $O(n \log n)$ and good solutions can be found in $O(\log n)$ [78]. Superior

TABLE 3. Simulation Parameters

Parameters	Estimated Value
Area	100×100–350×350 m^2
Number of sensor nodes	100
Initial energy	1 J
Data packet length	4 KB
Hello packet length	100 –150 B
Aggregation percentage	10%
UAV default flying altitude	50 m
Sensor's altitude	0 – 3 m
UAV default speed	20 m/s
Sensor mobility	Static
Carrier frequency	2.4 GHz
Antenna type	Omnidirectional
MAC protocol	CSMA, TDMA
Propagation mode	Log-distance path loss model
Path loss exponent (η) UAV↔sensor	2
Path-loss exponent (η) sensor↔sensor	2.45-3.40

runtime complexity and the chances of getting good solutions in less iterations make GA a favorable option to solve the TSP.

V. PERFORMANCE EVALUATION

A. SIMULATION ENVIRONMENT

The performance of the proposed EFDC scheme was evaluated via an extensive computer simulation using MATLAB. The parameters used are summarized in Table 3.

We compared our proposed scheme with two other data collection mechanisms, namely LEACH [79] with UAV and the original HEED with UAV. The compared mechanisms change their CHs in every round, representing the common approaches adopted for UWSN data collection. As assumed, our ROI is in a remote place where no static infrastructure is available. Thus, the UAV needs to determine the CH's location for the compared schemes first. As a result, we cannot apply any shortest path tour to optimize the data collection path in these schemes. We applied the S-path pattern for the mobility of the UAV for these two schemes as well. The S-pattern used in these schemes is the same as the mobility pattern we used in the discovery phase of our proposed scheme. Most of the data collection algorithms for UWSNs assume that they have prior knowledge about the CH positions with the help of static infrastructure. This is the main difference of our proposed scheme—the UAV cannot get any prior information about the topology because of the unreachability of the ROI. As a result, our research is not comparable with other studies in the field of UWSN data collection scheme, even though they are also dealing with the topic of WSN energy efficiency.

B. ENERGY CONSUMPTION MODEL

We utilized the simplest energy transmission model for calculating the WSN energy consumption. As shown in [76], the energy consumption of a WSN node mainly depends on the energy consumed for transmitting and receiving signals. The energy consumption for l bit data transmission to distance δ of a sensor node represented as $E_{Tx}(l, \delta)$ can be computed by the following equation:

$$E_{Tx}(l,\delta) = E_{Tx-elec(l)} + E_{Tx-amp(l,\delta)} = \begin{cases} l * E_{elec} + l * \psi_{fs} * \delta^2, & \delta < \delta_{th} \\ l * E_{elec} + l * \psi_{mp} * \delta^4, & \delta > \delta_{th} \end{cases}, \quad (39)$$

where E_{elec} represents the node's circuitry energy consumption for transmitting one bit data, $E_{Tx-elec(l)}$ the circuitry energy consumption for transmitting l bit data, and $E_{Tx-amp(l,\delta)}$ the energy consumption of the amplifier of a node to transmit l bit data to distance δ.ψ_{fs} and ψ_{mp} are environment dependent variables. ψ_{fs} serves as the transmitter amplifier model in the free space environment, whereas ψ_{mp} is for the multipath model. The use of ψ_{fs} or ψ_{mp} depends on the distance between the transmitter and the receiver. The threshold distance δ_{th} can be calculated using the following equation:

$$\delta_{th} = \sqrt{\frac{\psi_{fs}}{\psi_{mp}}}. \quad (40)$$

If the actual distance between the transmitter and the receiver is greater than δ_{th}, then the multipath energy consumption model is used; otherwise, the free space model is applied.

The energy consumption for receiving a message can be derived by the following equation:

$$E_{Rx}(l) = E_{Rx-elec} * l, \quad (41)$$

where the equation simply shows the energy consumed due to l bit data receiving, denoted by $E_{Rx}(l)$.$E_{Rx-elec}$ stands for the energy consumption for receiving one bit of data.

The energy consumption of the EFDC scheme is measured based on the data transmission and data receiving by the sensor nodes in three phases, namely initialization, discovery, and data collection. We calculated the energy consumption for data transmission based on (39) and (41). For the energy consumption analysis, the duration of the simulation depends on the completion of the number of rounds and it varies for the three compared schemes. It should be noted that we only considered the energy consumption of the deployed sensor

nodes. The UAV's energy consumption is not taken into consideration as it is rechargeable and can harvest energy through solar power. The energy consumption is obtained using the following formula:

$$E_{\forall} = \sum_{r_n=1}^{R_n} \sum_{j \in C} \sum_{i \in j} \left(\sum E_{Tx_i} + \sum E_{Rx_i} \right), \tag{42}$$

where $E_{\forall}$ is the total energy consumption of the sensor nodes due to data transmission and reception. E_{Tx_i} and E_{Rx_i} correspond to the energy consumption of a node i due to the transmission and reception of data, respectively. The definitions of E_{Tx_i} and E_{Rx_j} are given in (39) and (41), respectively. C is the set of clusters, j denotes a single cluster, and i denotes a single node from the cluster. R_n stands for the number of rounds and, with the help of r_n, we can iterate and compute the energy consumption for each round. The energy consumption is derived for a given number of rounds. The energy consumption analysis given in Figs. 6–9 and Fig. 11 is derived with the help of (42). EFDC tries to optimize the WSN energy by optimizing the transmission distance and reducing the number of transmissions. Distance optimization results in lower transmission power, and reducing the number of transmissions results in a less number of message receptions. Thus, to analyze the total energy consumption, we have taken these two key parameters in concern.

C. DELAY MODEL

The delay performance of the proposed EFDC scheme is derived based on the following formula:

$$D_{\forall} = \sum_{r_n=1}^{R_n} \left(\frac{g(O)}{DV_{UAV}} + \sum_{j \in C} \sum_{i \in j} \$ \right), \tag{43}$$

where $D_{\forall}$ denotes the total time required for a UAV to collect data from the WSN for a given number of rounds. DV_{UAV} is the default speed of the UAV. Data collection positions list O is computed with the help of Algorithm 3. From (38), we can derive $g(O)$, which denotes the total distance covered by a UAV by following the optimal trajectory. Fig. 5 gives a good visual explanation of the optimal trajectory. $\$$ is the transmission time taken by a single node to upload its sensed data to the UAV in a single round. C is the set of clusters, j denotes a single cluster, and i is a node of the cluster. To simplify the equation, we consider the delay for all transmissions to be equal and represent it as $\$$. The delay is computed for a given number of data collection rounds denoted as R_n, and r_n is a variable to iterate through the rounds. For simplicity, the UAV speed DV_{UAV} is kept constant. Thus, the acceleration, deceleration, wind effect, or any other parameters that might bring change into the UAV speed are not taken into consideration. The analysis given in Fig. 12 is done with the help of (43).

D. SIMULATION RESULTS AND DISCUSSION

In this subsection, the simulation results of our EFDC scheme are presented in performance graphs and comparatively discussed with the two conventional schemes, i.e., LEACH with UAV and HEED with UAV.

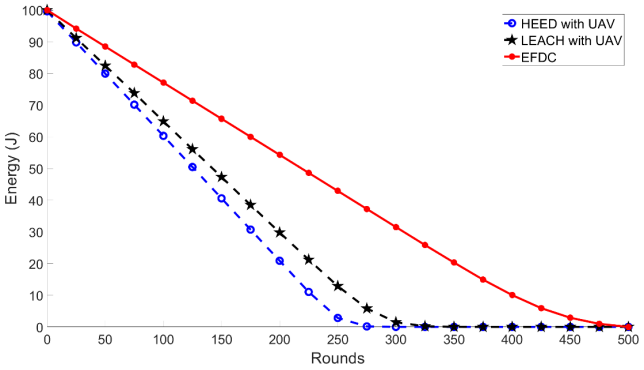

FIGURE 6. Energy performance with linear S-path approach.

Fig. 6 depicts the energy consumption of the EFDC scheme compared with those of the LEACH with UAV and HEED with UAV. The compared approaches follow the linear data collection approach, in which the UAV collects data from the shortest position according to its way of the S-path mobility model and does not visit the CH's position physically. The cumulative energy of the entire WSN is measured in joules and shown in the vertical axis, whereas the number of rounds is indicated in the horizontal axis.

From Fig. 6, it is evident that the energy consumption of our proposed mechanism is less than those of the LEACH with UAV and HEED with UAV. The lower energy consumption of the EFDC scheme is expected, because no distance optimization is performed in the compared approaches. According to our energy consumption model in (39), the transmission energy heavily depends on the distance between the transmitter and the receiver; thus, the total energy consumptions in the compared approaches are higher than that of our proposed approach. As no static infrastructure is taken into consideration in the EFDC approach, more energy is consumed for hello packet broadcasting in the other two approaches to determine the positions of the CHs in every round.

Fig. 7 displays the energy consumption comparison between the EFDC and the other two schemes. In this simulation, the UAV visits the CH's location to collect data from the clusters from its default altitude. Theoretically, the energy consumption should decrease as the distance between the CH and the UAV is reduced. However, our simulation result does not show a significant improvement for HEED with UAV, whereas the LEACH with UAV approach shows a slight improvement, and the WSN takes 50 rounds more to become completely dry compared to that in Fig. 7. The energy efficiency of our EFDC scheme does not only depend on the UAV visitation to the CH's position but also on other energy optimization factors such as direct data collection from the sensors and suboptimal position search.

Fig. 8 presents the comparison of dead nodes per round among the proposed EFDC and the compared schemes fol-

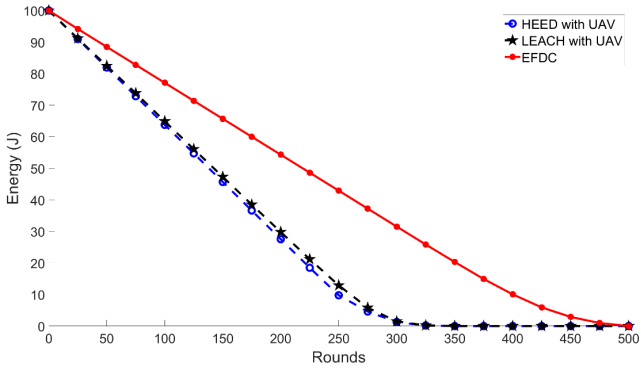

FIGURE 7. Energy performance with data collection approach from the CH position.

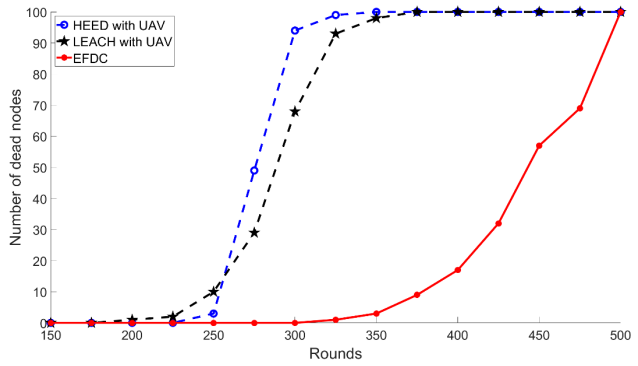

FIGURE 9. Number of dead nodes with data collection approach from the CH position.

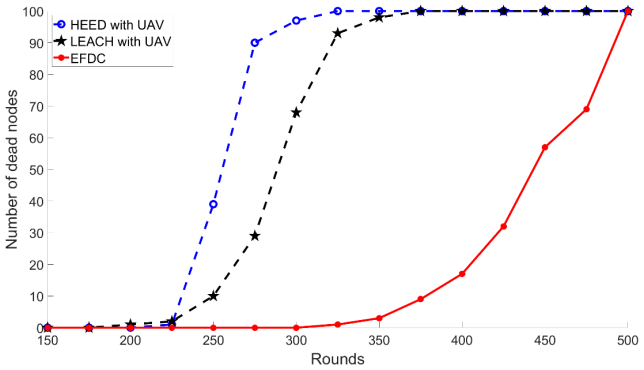

FIGURE 8. Number of dead nodes with linear S-path approach.

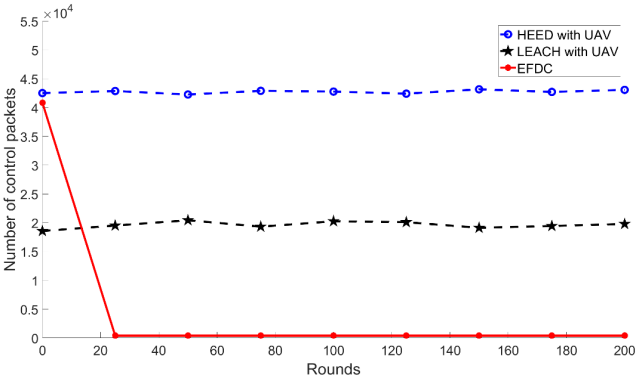

FIGURE 10. Number of control packets versus number of rounds.

lowing the S-path linear approach. The term dead node means that the node's specific energy becomes lower than the threshold value and the node becomes unable to transfer its sensed data to the other nodes or the UAV. The graph shows that the number of dead nodes in HEED with UAV is the highest, and EFDC shows the best result among the compared schemes. The number of dead nodes per round also indirectly indicates the lifetime of the WSN. In LEACH with UAV and HEED with UAV approaches, all nodes become dead in approximately 340 rounds whereas in EFDC, it took almost 500 rounds.

Fig. 9 shows a comparison of the EFDC with the two approaches in terms of the number of dead nodes, where the UAV visits the CH position to collect the sensed data from the clusters. This graph shows that even if the UAV visits the CH position with its default altitude and optimizes the distance between them, the dead node count for the proposed EFDC still shows a better result. This outcome also proves that our suboptimal positioning technique has a beneficial effect on the outcome of the dead node count per round performance metric, which cannot be achieved only by acquiring the CH's position for the UAV.

Fig. 10 depicts a comparison of the number of exchanged control packets among LEACH with UAV, HEED with UAV, and the proposed EFDC. We can observe that EFDC exchanges a relatively higher number of control packets in the first round compared to the subsequent rounds. As already mentioned, the clustering process takes place only once in EFDC. As a result, to form the cluster among the sensor nodes with the CBHEED clustering approach, the method consumes a relatively higher number of control packets. In the subsequent rounds, our approach does not reform the clusters; therefore, the number of exchanged control packets decreases dramatically. In the other two approaches, the CHs change in every round of data collection, so the sensor nodes need to exchange a suitable number of control packets to locate and initiate the data transmissions between the CHs and UAV. On the other hand, EFDC does not need to find the CH position in every round, which also contributes to the increasing number of the exchanged control packets.

Fig. 11 illustrates the scalability performance of the proposed EFDC. The scalability is measured among the three compared schemes by varying the area parameter. It should be noted that we took a square shape of ROI in consideration and the length and width were measured in meters. We assumed that the nodes are randomly deployed. For the two compared clustering techniques, the intra-cluster distance increases with the increment of the area. Therefore, the data transmission cost in terms of energy also increases. In the proposed EFDC, the suboptimal position search algorithm plays a major part behind the superior outcome. The tabu search finds a suitable place that optimizes the distance

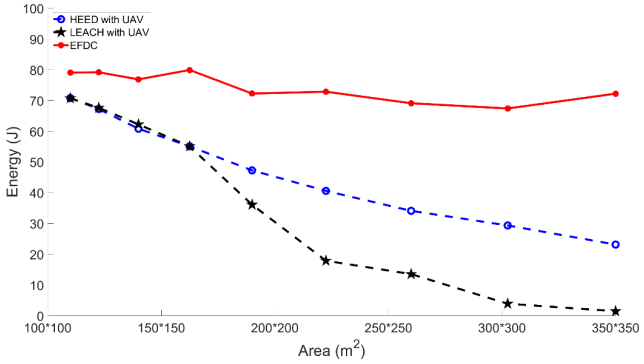

FIGURE 11. Energy consumption versus network area.

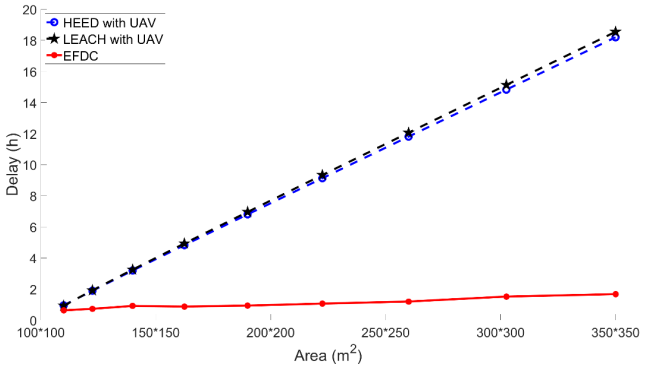

FIGURE 12. Data collection delay versus network area.

among all nodes, which also reduces the energy consumption of the WSN. With the increasing area of the ROI, the effectiveness and necessity of finding the data collection position also increases.

Fig. 12 displays the outcome of the delay analysis comparison between our proposal and the other two approaches. The delay performance of the compared approaches is calculated based on (43). The LEACH with UAV and HEED with UAV approaches do not know the position of the CH before they start for the data collection tour. As a result, both need to follow a search and collect mechanism. For implementing the scenario, we used an S-shaped UAV path from where the UAV simultaneously searches for the CHs and collects data from them. Consequently, the data collection time increases enormously with the increasing size of the ROI, whereas in EFDC, the UAV is able to collect all data collection positions in advance, and it calculates the shortest data collection trajectory based on the GA. The trajectory optimization algorithm shortens the data collection path; thus, our EFDC shows a better result. The graph also shows that in our method, the data collection time does not vary substantially with the size of the ROI, unlike those of the compared approaches, because in EFDC, the traveling distance depends on the distance of the calculated suboptimal position for data collection and not directly on the size of the ROI. In LEACH with UAV and HEED with UAV, the delay increases with increasing size of the ROI.

Fig. 13 is presented to analyze the energy depletion comparison between direct data collection and CH data collection. In order to show that the energy consumption between a CH and its CMs is balanced, only one cluster is assumed for simplicity in this simulation. In the direct data collection mechanism, all CMs along with their CH directly send their data to the UAV, whereas in the CH data collection mechanism, the CMs first send their data to the CH and the CH sends the data to the UAV. The data shown in fig. 13 were taken from one cluster consisting of nine CMs and one CH. The CH selection was done by our proposed CBHEED clustering technique. The horizontal axis shows the node ID and the vertical axis shows the remaining energy after data collection. The analysis was conducted by observing the energy depletion from the same cluster. The graph shows that even though both cases consume similar amounts of energy for sending data from the CMs to the CH or UAV, the CH consumes more energy in the CH data collection method. Thus, collecting data through the CH will consume more energy because of an imbalanced energy consumption, and the direct data collection approach is the better option for our given scenario.

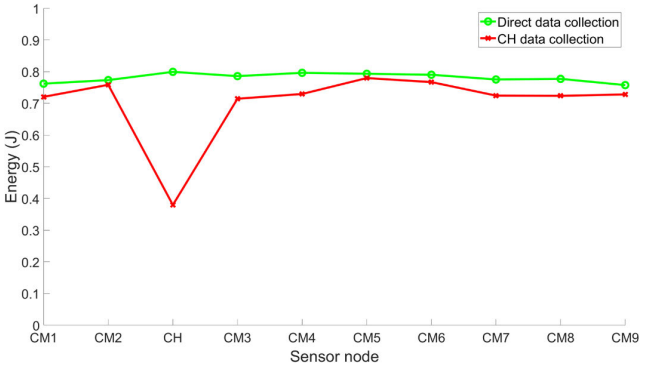

FIGURE 13. Energy consumption at different nodes using CM-UAV direct transmission and CM-CH-UAV transmission.

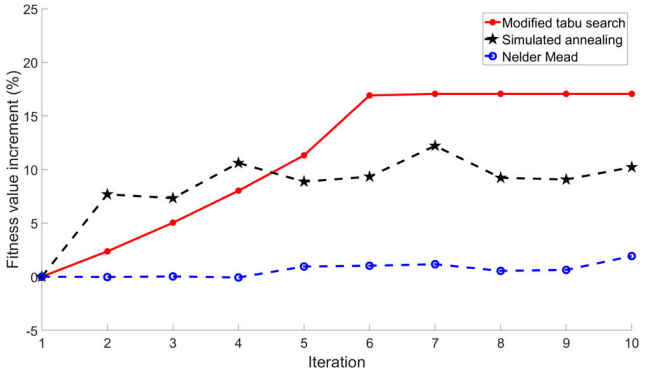

FIGURE 14. Convergence along with iterations.

Fig. 14 shows the convergence along with iterations for the three algorithms of the modified tabu search, simulated annealing [80], and Nelder Mead optimization [81]. In the simulation, data are taken five times for every iteration and, then, the percentage of the changed fitness value is recorded based on the initial fitness values for all the three algorithms. The graph shows the relative increment of fitness values,

in which it can be seen that the proposed tabu search algorithm does not bring any change after the sixth iteration. This is a desired phenomenon for implementing the tabu search as the goal is to achieve a moderately optimized data collection position with the minimal iteration count. Even though the energy consumption of UAV has not considered in designing EFDC, the higher the number of iterations is, the higher the energy depletion will be for the UAV as well as the sensor nodes. As for every iteration, the sensor nodes also need to broadcast a beacon packet.

VI. CONCLUSION

In this study, we proposed an EFDC scheme for UWSNs. This scheme is suitable for data collection in hilly or mountainous areas, where infrastructures are difficult to build and maintain. Energy-efficient data collection requires a suitable UAV position for data collection. To find an initial data collection position, we proposed the CBHEED clustering algorithm by modifying the HEED algorithm. The probability of being a CH of a sensor node depends on the central bias of its geolocation in a polygon formed by its neighboring nodes. The polygon formulation was performed by applying the monotone chain convex hull algorithm and the centroid of the polygon was derived by applying Paul Bourke's centroid finding calculation. The positions of the CHs were selected by the CBHEED algorithm, which tries to minimize the overall energy consumption of data collection within a cluster.

The second level of energy optimization was conducted by computing a suboptimal position for data collection by applying a modified tabu search algorithm. This algorithm tries to determine a better position that will consume less energy and improve load balancing in terms of energy consumption in a cluster simultaneously. The UAV-aided data collection approach is separated into discovery and data collection phases. In the discovery phase, the UAV searches the CH locations and optimizes the data collection position based on the modified tabu search algorithm. We applied a modified GA to optimize the trajectory of the data collection route based on the derived data collection positions. In the data collection phase, the sensed data are collected from each of the sensors to the UAV via a direct connection with a cluster. As a result, no extra workload is given on the CH such as collecting and aggregating data from the CMs. The altitude with the position is also optimized and thus, less energy is consumed compared with the conventional approaches. In EFDC, we ran the discovery phase for a single time only as the CH positions do not change. We compared the performance of the proposed EFDC with HEED with UAV and LEACH with UAV in terms of energy efficiency, dead node comparison, scalability, and load balancing.

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous referees for their helpful comments in improving the quality of this article.

REFERENCES

[1] O. Ogundile and A. Alfa, "A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks," *Sensors*, vol. 17, no. 5, p. 1084, May 2017.
[2] P. Nayak and B. Vathasavai, "Energy efficient clustering algorithm for multi-hop wireless sensor network using Type-2 fuzzy logic," *IEEE Sensors J.*, vol. 17, no. 14, pp. 4492–4499, Jul. 2017.
[3] M. Y. Arafat, M. A. Habib, and S. Moh, "Routing protocols for UAV-aided wireless sensor networks," *Appl. Sci.*, vol. 10, no. 12, p. 4077, Jun. 2020.
[4] Y. Liu, K. Ota, K. Zhang, M. Ma, N. Xiong, A. Liu, and J. Long, "QTSAC: An energy-efficient MAC protocol for delay minimization in wireless sensor networks," *IEEE Access*, vol. 6, pp. 8273–8291, Feb. 2018.
[5] D. R. Dandekar and P. R. Deshmukh, "Energy balancing multiple sink optimal deployment in multi-hop wireless sensor networks," in *Proc. 3rd IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2013, pp. 408–412.
[6] M. Navarro, T. W. Davis, Y. Liang, and X. Liang, "A study of long-term WSN deployment for environmental monitoring," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 2093–2097.
[7] S. Nagata, I. Miyagawa, and K. Murakami, "Flexible flight navigation for quadcopters based on geometric formation using motion sensing," in *Proc. Int. Workshop Adv. Image Technol. (IWAIT)*, Jan. 2018, pp. 1–4.
[8] D. R. McArthur, A. B. Chowdhury, and D. J. Cappelleri, "Autonomous control of the interacting-BoomCopter UAV for remote sensor mounting," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5219–5224.
[9] J. Chen, F. Yan, S. Mao, F. Shen, W. Xia, Y. Wu, and L. Shen, "Efficient data collection in large-scale UAV-aided wireless sensor networks," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019.
[10] M. Hammoudeh, F. Al-Fayez, H. Lloyd, R. Newman, B. Adebisi, A. Bounceur, and A. Abuarqoub, "A wireless sensor network border monitoring system: Deployment issues and routing protocols," *IEEE Sensors J.*, vol. 17, no. 8, pp. 2572–2582, Apr. 2017.
[11] M. S. Sumathi, G. S. Anitha, and N. H. Sridhar, "Efficient data handling of wireless sensor network for real time landslide monitoring system using fuzzy technique," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Apr. 2017.
[12] M. Eckerstorfer, Y. Bühler, R. Frauenfelder, and E. Malnes, "Remote sensing of snow avalanches: Recent advances, potential, and limitations," *Cold Regions Sci. Technol.*, vol. 121, pp. 126–140, Jan. 2016.
[13] M. I. Khan, W. N. Gansterer, and G. Haring, "Static vs. Mobile sink: The influence of basic parameters on energy efficiency in wireless sensor networks," *Comput. Commun.*, vol. 36, no. 9, pp. 965–978, May 2013.
[14] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct. 2019.
[15] Z. A. Ali, S. Masroor, and M. Aamir, "UAV based data gathering in wireless sensor networks," *Wireless Pers. Commun.*, vol. 106, no. 4, pp. 1801–1811, Jun. 2019.
[16] B. Liu and H. Zhu, "Energy-effective data gathering for UAV-aided wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2506, 2019.
[17] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "UAV-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1893–1905, Apr. 2019.
[18] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.
[19] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, Jun. 2019.
[20] J. Hu, T. Wang, J. Yang, Y. Lan, S. Lv, and Y. Zhang, "WSN-assisted UAV trajectory adjustment for pesticide drift control," *Sensors*, vol. 20, no. 19, p. 5473, Sep. 2020.
[21] D. Popescu, F. Stoican, G. Stamatescu, O. Chenaru, and L. Ichim, "A survey of collaborative UAV–WSN systems for efficient monitoring," *Sensors*, vol. 19, no. 21, p. 4690, Oct. 2019.
[22] S. Say, H. Inata, J. Liu, and S. Shimamoto, "Priority-based data gathering framework in UAV-assisted wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 14, pp. 5785–5794, Jul. 2016.
[23] S. Poudel and S. Moh, "Energy-efficient and fast MAC protocol in UAV-aided wireless sensor networks for time-critical applications," *Sensors*, vol. 20, no. 9, p. 2635, May 2020.

[24] S. Poudel and S. Moh, "Medium access control protocols for unmanned aerial vehicle-aided wireless sensor networks: A survey," *IEEE Access*, vol. 7, pp. 65728–65744, 2019.
[25] D.-T. Ho, E. I. Grøtli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of wireless sensor network and UAV data acquisition," *J. Intell. Robotic Syst.*, vol. 78, no. 1, pp. 159–179, Apr. 2015.
[26] D. Popescu, F. Stoican, L. Ichim, G. Stamatescu, and C. Dragana, "Collaborative UAV-WSN system for data acquisition and processing in agriculture," in *Proc. 10th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, Sep. 2019, pp. 519–524.
[27] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, Jun. 2019.
[28] J. Mi, X. Wen, C. Sun, Z. Lu, and W. Jing, "Energy-efficient and low package loss clustering in UAV-assisted WSN using K means++ and fuzzy logic," in *Proc. IEEE/CIC Int. Conf. Commun. Workshops China (ICCC Workshops)*, Aug. 2019, pp. 210–215.
[29] Y. Pang, Y. Zhang, Y. Gu, M. Pan, Z. Han, and P. Li, "Efficient data collection for wireless rechargeable sensor clusters in harsh terrains using UAVs," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 234–239.
[30] S. K. Haider, M. A. Jamshed, A. Jiang, H. Pervaiz, and Q. Ni, "UAV-assisted cluster-head selection mechanism for wireless sensor network applications," in *Proc. UK/ China Emerg. Technol. (UCET)*, Aug. 2019, pp. 1–2.
[31] F. Stoican, D. Popescu, and L. Ichim, "Trajectory design for effective and secure communication in UAV-WSN systems," in *Proc. IEEE Radio Antenna Days Indian Ocean (RADIO)*, Sep. 2019, pp. 1–2.
[32] J. Yang, X. Wang, Z. Li, P. Yang, X. Luo, K. Zhang, S. Zhang, and L. Chen, "Path planning of unmanned aerial vehicles for farmland information monitoring based on WSN," in *Proc. 12th World Congr. Intell. Control Autom. (WCICA)*, Jun. 2016, pp. 2834–2838.
[33] J. Grigulo and L. B. Becker, "Experimenting sensor nodes localization in WSN with UAV acting as mobile agent," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 808–815.
[34] D. S. Jasrotia and M. J. Nene, "Localisation using UAV in RFID and sensor network environment:Needs and challenges," in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*, Oct. 2019, pp. 274–279.
[35] W. Wang, J. Tang, N. Zhao, X. Liu, X. Y. Zhang, Y. Chen, and Y. Qian, "Joint precoding optimization for secure SWIPT in UAV-aided NOMA networks," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5028–5040, Aug. 2020.
[36] Q. Wu, J. Xu, Y. Zeng, D. W. K. Ng, and N. Al-Dhahir, "5G-and-beyond networks with UAVs: From communications to sensing and intelligence," Oct. 2020, *arXiv:2010.09317v1*. [Online]. Available: https://arxiv.org/abs/2010.09317
[37] M. M. Azari, G. Geraci, A. Garcia-Rodriguez, and S. Pollin, "UAV-to-UAV communications in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 6130–6144, Sep. 2020.
[38] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.
[39] Z. Xiao, P. Xia, and X.-G. Xia, "Enabling UAV cellular with millimeter-wave communication: Potentials and approaches," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 66–73, May 2016.
[40] M. Tao, X. Li, H. Yuan, and W. Wei, "UAV-aided trustworthy data collection in federated-WSN-enabled IoT applications," *Inf. Sci.*, vol. 532, pp. 155–169, Sep. 2020.
[41] T. D. P. Perera, S. Panic, D. N. K. Jayakody, P. Muthuchidambaranathan, and J. Li, "A WPT-enabled UAV-assisted condition monitoring scheme for wireless sensor networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 23, 2020, doi: 10.1109/TITS.2020.3018493.
[42] C. Trasviña-Moreno, R. Blasco, Á. Marco, R. Casas, and A. Trasviña-Castro, "Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring," *Sensors*, vol. 17, no. 3, p. 460, Feb. 2017.
[43] C. Dragana, V. Mihai, G. Stamatescu, and D. Popescu, "In-network stochastic consensus for WSN surveillance applications," in *Proc. 9th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2017, pp. 1–6.
[44] M. Bacco, P. Barsocchi, P. Cassara, D. Germanese, A. Gotta, G. R. Leone, D. Moroni, M. A. Pascali, and M. Tampucci, "Monitoring ancient buildings: Real deployment of an IoT system enhanced by UAVs and virtual reality," *IEEE Access*, vol. 8, pp. 50131–50148, 2020.
[45] S. Berrahal, J.-H. Kim, S. Rekhis, N. Boudriga, D. Wilkins, and J. Acevedo, "Border surveillance monitoring using quadcopter UAV-aided wireless sensor networks," *J. Commun. Softw. Syst.*, vol. 12, no. 1, pp. 67–82, Mar. 2016.
[46] R. Teguh, T. Honma, A. Usop, H. Shin, and H. Igarashi, "Detection and verification of potential peat fire using wireless sensor network and UAV," in *Proc. Int. Conf. Inf. Technol. Elect. Eng.*, Jun. 2012, pp. 6–10.
[47] T. O. Olasupo, "Propagation modeling of IoT devices for deployment in multi-level hilly urban environments," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 346–352.
[48] Z. Qiu, X. Chu, C. Calvo-Ramirez, C. Briso, and X. Yin, "Low altitude UAV air-to-ground channel measurement and modeling in semiurban environments," *Wireless Commun. Mobile Comput.*, vol. 2017, Nov. 2017, Art. no. 1587412.
[49] T. Hayes and F. H. Ali, "Location aware sensor routing protocol for mobile wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 6, no. 2, pp. 49–57, Apr. 2016.
[50] J. T. Thirukrishna, S. Karthik, and V. P. Arunachalam, "Revamp energy efficiency in homogeneous wireless sensor networks using optimized radio energy algorithm (OREA) and power-aware distance source routing protocol," *Future Gener. Comput. Syst.*, vol. 81, pp. 331–339, Apr. 2018.
[51] S. Hasegawa, R. Kitagawa, T. Ito, T. Nakajima, S.-J. Kim, Y. Shoji, and M. Hasegawa, "Performance evaluation of machine learning based channel selection algorithm implemented on IoT sensor devices and its application to wireless sensor network for building monitoring system," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 161–166.
[52] R. La Rosa, P. Livreri, C. Trigona, L. Di Donato, and G. Sorbello, "Strategies and techniques for powering wireless sensor nodes through energy harvesting and wireless power transfer," *Sensors*, vol. 19, no. 12, p. 2660, Jun. 2019.
[53] G. Liu, Z. Li, X. Zhou, and S. Li, "Transmission power control for wireless sensor networks," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, vol. 9258, Sep. 2007, pp. 2596–2599.
[54] L. Mariga, I. Silva Tiburcio, C. Martins, A. A. Prado, and C. Nascimento, Jr., cmartins, "Measuring battery discharge characteristics for accurate UAV endurance estimation," *Aeronaut. J.*, vol. 124, no. 1277, pp. 1099–1113, 2020.
[55] J. Wu, H. Wang, Y. Huang, Z. Su, and M. Zhang, "Energy management strategy for solar-powered UAV long-endurance target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 1878–1891, Aug. 2019.
[56] M. Khan, "Quadcopter flight dynamics," *Int. J. Sci. Technol. Res.*, vol. 3, no. 8, pp. 130–135, 2014.
[57] A. Joukhadar, M. AlChehabi, C. Stöger, and A. Müller, "Trajectory tracking control of a quadcopter UAV using nonlinear control," in *Mechanisms and Machine Science*, vol. 58. Amsterdam, The Netherlands: Springer, 2019, pp. 271–285.
[58] X. Ouyang, F. Zeng, D. Lv, T. Dong, and H. Wang, "Cooperative navigation of UAVs in GNSS-denied area with colored RSSI measurements," *IEEE Sensors J.*, vol. 21, no. 2, pp. 2194–2210, Jan. 2021.
[59] E. Christy, R. P. Astuti, B. Syihabuddin, B. Narottama, O. Rhesa, and F. Rachmawati, "Optimum UAV flying path for device-to-device communications in disaster area," in *Proc. Int. Conf. Signals Syst. (ICSigSys)*, May 2017, pp. 318–322.
[60] Z. Tao, "TSP problem solution based on improved genetic algorithm," in *Proc. 4th Int. Conf. Natural Comput.*, vol. 1, 2008, pp. 686–690.
[61] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct. 2004.
[62] S. H. Ahmed, S. H. Bouk, N. Javaid, and I. Sasase, "RF propagation analysis of MICAz Mote's antenna with ground effect," in *Proc. 15th Int. Multitopic Conf. (INMIC)*, Dec. 2012, pp. 270–274.
[63] P. Bourke. (1988). *Calculating the Area and Centroid of a Polygon*. [Online]. Available: http://www.seas.upenn.edu/~sys502/extra_materials/Polygon%20Area%20and%20Centroid.pdf
[64] G. Mei, J. C. Tipper, and N. Xu, "An algorithm for finding convex hulls of planar point sets," in *Proc. 2nd Int. Conf. Comput. Sci. Netw. Technol.*, Dec. 2012, pp. 888–891.
[65] D. C. S. Allison and M. T. Noga, "Some performance tests of convex hull algorithms," *BIT*, vol. 24, no. 1, pp. 2–13, Mar. 1984.
[66] A. Andrew and A. Am, "Another efficient algorithm for convex hulls in two dimensions," *Inf. Process. Lett.*, vol. 9, no. 5, pp. 216–219, 1979.

[67] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.
[68] A. E. Lagias, T. D. Lagkas, and J. Zhang, "New RSSI-based tracking for following mobile targets using the law of cosines," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 392–395, Jun. 2018.
[69] Z. Hao, N. Qu, X. Dang, and J. Hou, "RSS-based coverage deployment method under probability model in 3D-WSN," *IEEE Access*, vol. 7, pp. 183091–183104, 2019.
[70] M. Cheffena and M. Mohamed, "Empirical path loss models for wireless sensor network deployment in snowy environments," *IEEE Antennas Wireless Propag. Lett.*, vol. 16, pp. 2877–2880, Sep. 2017.
[71] F. Lassabe, P. Canalda, P. Chatonnay, F. Spies, and O. Baala, "A Friis-based calibrated model for WiFi terminals positioning," in *Proc. 6th IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, Jun. 2005, pp. 382–387.
[72] A. A. Hussein, T. A. Rahman, and C. Y. Leow, "Performance evaluation of localization accuracy for a log-normal shadow fading wireless sensor network under physical barrier attacks," *Sensors*, vol. 15, no. 12, pp. 30545–30570, 2015.
[73] J. Miranda, R. Abrishambaf, T. Gomes, P. Goncalves, J. Cabral, A. Tavares, and J. Monteiro, "Path loss exponent analysis in wireless sensor networks: Experimental evaluation," in *Proc. 11th IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2013, pp. 54–58.
[74] W. Tang, X. Ma, J. Wei, and Z. Wang, "Measurement and analysis of near-ground propagation models under different terrains for wireless sensor networks," *Sensors*, vol. 19, no. 8, p. 1901, Apr. 2019.
[75] A. Pratap Singh, D. Pratap Singh, and S. Kumar, "NRSSI: New proposed RSSI method for the distance measurement in WSNs," in *Proc. 1st Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Sep. 2015, pp. 296–300.
[76] P. Nayak and A. Devulapalli, "A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime," *IEEE Sensors J.*, vol. 16, no. 1, pp. 137–144, Jan. 2016.
[77] S. Pierre and F. Houeto, "Assigning cells to switches in cellular mobile networks using taboo search," *IEEE Trans. Syst., Man Cybern., B (Cybern.)*, vol. 32, no. 3, pp. 351–356, Jun. 2002.
[78] C. W. Tsai, S. P. Tseng, M. C. Chiang, C. S. Yang, and T. P. Hong, "A high-performance genetic algorithm: Using traveling salesman problem as a case," *Sci. World J.*, vol. 2014, May 2014, Art. no. 178621.
[79] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
[80] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Amsterdam, The Netherlands: Springer 1987.
[81] D. M. Olsson and L. S. Nelson, "The nelder-mead simplex procedure for function minimization," *Technometrics*, vol. 17, no. 1, pp. 45–51, Feb. 1975.

REZOAN AHMED NAZIB received the B.Sc. degree in computer science from BRAC University, Bangladesh, in 2017. He is currently pursuing the M.Sc. degree with the Mobile Computing Laboratory, Chosun University, South Korea. From 2017 to 2018, he was an Associate Consultant with SS Solution Ltd., Bangladesh. His current research interests include ad hoc networks and unmanned aerial networks with a focus on network architectures and protocols.

SANGMAN MOH (Member, IEEE) received the M.S. degree in computer science from Yonsei University, South Korea, in 1991, and the Ph.D. degree in computer engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2002. Since late 2002, he has been a Professor with the Department of Computer Engineering, Chosun University, South Korea. From 2006 to 2007, he was on leave at Cleveland State University, USA. Until 2002, he was with the Electronics and Telecommunications Research Institute (ETRI), South Korea, where he served as a Project Leader. His research interests include mobile computing and networking, ad hoc and sensor networks, cognitive radio networks, unmanned aerial vehicle networks, and parallel and distributed computing systems.

Dr. Moh is a member of ACM, IEICE, KIISE, IEIE, KIPS, KICS, KMMS, IEMEK, KISM, and KPEA.