February 2021
Master’s Degree Thesis

Data Collection Scheme for Unmanned Aerial Vehicle-Aided Wireless Sensor Networks

Graduate School of Chosun University

Department of Computer Engineering

Rezoan Ahmed Nazib

Data Collection Scheme for Unmanned Aerial Vehicle-Aided Wireless Sensor Networks

무인 비행체 활용 무선 센서 네트워크를 위한 데이터 수집 기법

February 25, 2021

Graduate School of Chosun University

Department of Computer Engineering

Rezoan Ahmed Nazib

Data Collection Scheme for Unmanned Aerial Vehicle-Aided Wireless Sensor Networks

Advisor: Prof. Sangman Moh, PhD

A thesis submitted in partial fulfillment of the
requirements for a master's degree

October, 2020

Graduate School of Chosun University

Department of Computer Engineering

Rezoan Ahmed Nazib

나집 레조안 아흐메드
석사학위논문을 인준함

위원장 조선대학교 교수 정일용 (인)

위　원 조선대학교 교수 최우열 (인)

위　원 조선대학교 교수 모상만 (인)

2020년 11월

조선대학교 대학원

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Data Collection Scheme for Unmanned Aerial Vehicle-Aided Wireless Sensor Networks

Rezoan Ahmed Nazib
Advisor: Prof. Sangman Moh, Ph.D.
Department of Computer Engineering
Graduate School of Chosun University

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. We introduced a joint optimization problem based on the EFDC scheme and provided low-complexity solutions. 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 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.

요 약

무인 비행체 활용 무선 센서 네트워크를 위한 데이터 수집 기법

레조안 아흐메드 나집
지도교수: 모상만
컴퓨터공학과
조선대학교 대학원

에너지 제약 센서 노드는 민간 및 군사 목적으로 원격 지역, 구릉지, 접근하기 어려운 지역에 배치되는 경우가 많다. 이러한 무선 센서 네트워크(WSN)에서는 무인 비행체(UAV)를 사용하여 센서 노드로부터 데이터를 수집할 수 있다. 저고도 UAV 를 활용하면 데이터 수집 위치를 최적화해 WSN 의 에너지 소비량을 줄일 수 있다. 본 연구에서는 데이터 운반 장치로서 UAV 의 도움을 받아 구릉지에 대한 UAV 활용 WSN (UWSN) 시스템에서 에너지 효율적이고 빠른 데이터 수집(EFDC) 기법를 고안하였다. 제안한 EFDC 기법에서는 공동 최적화 문제에 대한 낮은 복잡도의 솔루션을 제공한다. 먼저 센서 그룹화를 위해 에너지 효율적인 하이브리드 분산 클러스터링 알고리즘을 제안하였다. 그런 다음 수정형 Tabu 검색 알고리즘을 적용하여 클러스터에서 데이터를 수집하는 UAV 위치를 최적화하였다. 결정된 데이터 수집 위치들을 대상으로 여행 세일즈맨 문제를 적용하고 유전자 알고리즘을 채택하여 데이터 수집 속도를 향상시켰다. 시뮬레이션 결과에 의한면, 제안한 EFDC 기법은 에너지 소비, 확장성, 제어 오버헤드, 지연 및 부하 분산 측면에서 기존 방식보다 우수한 성능을 갖는다.

I. INTRODUCTION

A. Overview

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 UAV-aided WSNs (UWSNs), interconnectivity is not as important as in the conventional paradigms. Higher line of sight (LOS) 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 direction changing compared to the fixed-wing UAVs. This type of UAVs can also float steadily in the air. Thus, multi-rotor UAVs can be used to fine-tune the data collection position from a group of sensors.

B. Research Objective

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.

In this study, an energy-efficient and fast data collection (EFDC) scheme is proposed for UWSNs deployed in hilly terrains. Figure. 1 shows the graphical representation of the EFDC operation, where a multi-rotor UAV is deployed to collect WSN data from a hilly terrain. The figure depicts the direct data collection mechanism of the EFDC scheme from the sensor nodes to the UAV.

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 [10]. 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.

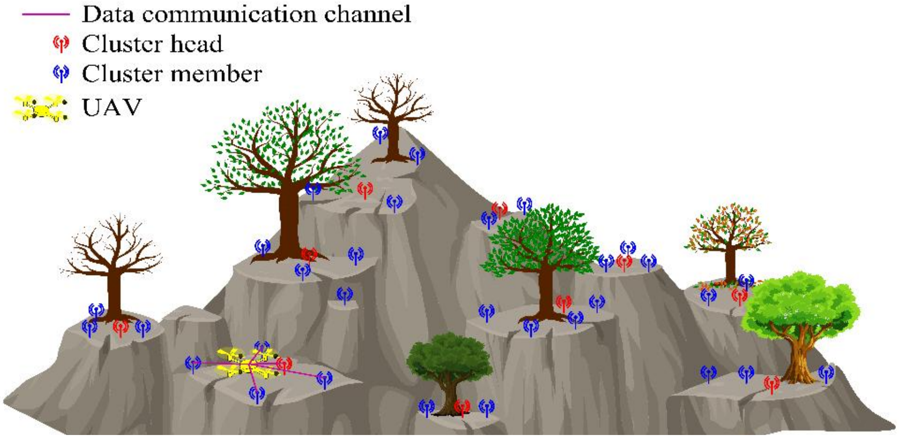


Figure 1. Graphical representation of data collection in a UWSN.

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.

C. Thesis Layout

Rest of the thesis is organized as follows:

In Chapter 2, the related works are reviewed and discussed. The limitations of the existing studies and the motivation behind our research are also provided in Chapter 2. In Chapter 3, the system model of the EFDC scheme are introduced. In Chapter 4, 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 Chapter 4. In Chapter 5, the performance of the proposed scheme is evaluated and compared with the conventional schemes. Finally, the conclusion is given in Chapter 6.

II. RELATED WORKS

Ali et al. [1] 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 [2] 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. [3] 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. [4] 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.

Say et al. [5] 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 [6], considering fast and energy-efficient data gathering for critical situations. A survey on MAC protocols for UWSNs was proposed in [7]. However, our research goal does not include optimizing the medium access usage.

Johansen et al. [8] 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 [9], 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 [10] 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 [11], 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. [12] proposed a data gathering mechanism for UWSNs, where the target area was also divided in clusters. The CH was determined based on the information value and the residual power in the sensor nodes. Direct future prediction was used to design the optimal trajectory of the data collection scheme. Pang et al. [13] also investigated the problem of data collection from a harsh terrain. Besides WSN data collection, their architecture also considered to recharge the sensor nodes while gathering data from them.

The optimal WSN CH selection technique is observed in many studies [14]. Some other studies investigated the optimal trajectory problem for collecting

WSN data [15], [16], while some studies were performed to localize the sensor nodes [17], [18].

Based on the aforementioned studies, even though UAVs are utilized as mules for data collection, the superior positioning capability of quadcopters is not utilized at all. One possible application for UAVs could be collecting data from unreachable or hard-to-reach areas, if construction and maintenance of static infrastructure in such areas are not feasible. Moreover, WSNs can be deployed for a limited observation period, which is also inefficient in terms of cost. Communication through cellular infrastructure is also not an energy-efficient solution because the sensor nodes might need to transmit at longer distances. The proposed EFDC scheme does not assume the presence of any static infrastructure, which makes it applicable for harsh or hilly terrains. The data collection position is redefined with a search mechanism to reduce the transmission power of the sensor nodes.

In [28], Carlos et al. did an 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 previous location. However, they did not apply any optimization process to reduce energy consumption of the sensor nodes. Besides, the protocol is not also suitable for a large number of nodes. In [29], 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 [30], 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 [31], 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 [32] 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.

Because data collection for UWSNs is more suitable for remote areas, the availability of infrastructure makes the scenario incompatible. Some proposed techniques assume that the control center has prior knowledge of the 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 for simulation purposes, which can be matched with the real-life unequal distribution of WSN nodes. It is observed from the above discussion that most of the proposed architectures have prior knowledge about the WSN topology. In our EFDC scheme, however, the UAV does not need any prior information from the infrastructures, which is different from the previous studies. This scenario decreases the utilization of UAVs. Collecting data only from the CH also reduces the utilization of the UAV while creating extra burden for the CH sensor nodes. Even though some of the studies showed trajectory optimization techniques, the altitude optimization technique is particularly missing from 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

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the assumptions, communication model, and problem formulation for the EFDC scheme are given. 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. The objective function is formulated by considering the two main objectives of the proposed scheme, namely energy efficient data collection and fast data collection. At the end of this section, the default mobility model of the UAV is explained. The symbols and notations used in the study are given in Table I.

Table 1. Symbols and notations

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
E_{Ag}	Energy consumption due to aggregation
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
F_{ngr}	Neighboring nodes ID
N_{poly}	Number of nodes in the polygon
CH_{prob}	Probability of a node to be a CH
ρ_i	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^{Fngr}	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
K_{hull}^{Fngr}	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
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
i_{sol}	Initial position for searching
R_n	Number of rounds
C	Set of clusters

O	Set of data collection position
£	Transmission delay
S	Starting and final positions of the UAV in the ROI

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 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 on the ground communication.

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.

B. 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.

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.

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.

Collision and obstacle free movement: It is assumed that the UAV does not face any obstacle on its way of movement.

UAV–WSN communication model: The communication model between the WSN and UAV is not considered as LOS communication. Depending on the randomly deployed natural obstacle's height, the RSSI value is derived based on the log-normal shadowing effect.

RSSI calculation ability: The UAV can calculate the RSSI power of the signal received from the ground sensor nodes.

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.

1. 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.

Static nodes: The nodes are static. Once deployed, the sensor nodes do not change their positional values anymore.

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.

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.

Node deployment: The nodes are deployed in a completely random manner over the ROI.

Energy constraints: Sensors are battery powered and the batteries are not rechargeable.

Adaptive transmission power control: The sensor nodes have the ability to control the transmission power. Transmission power is retuned after the data collection position is obtained [19].

2. 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 [20]; however, different from that in [20], we did not consider any priority for the nodes.

C. Communication Model

Sensor nodes are deployed randomly throughout the ROI. The assumed ROI is a remote region. Hence, no infrastructure is available for the sensor nodes to transmit their data directly to a sink node or to any NCC. Each sensor node in set $N = \{1, 2, 3, \dots, |N|\}$ has coordinates $K = \{k_1, k_2, k_3, \dots, k_{|N|}\}$, where, $K_{|N|} \in \mathbb{R}^{3\times1}$.

In a conventional UWSN data collection scenario, the UAV collects data from the CHs in the WSN [28], [38], [39], [40], [41], [45], [46], [47], [55]. In the EFDC scheme, every sensor node directly transmits its data to the UAV. The EFDC communication mechanism is divided into three phases: initialization, discovery, and data collection. Table II presents the algorithms used in the different phases of the proposed scheme along with their goals. In the initialization phase, the CBHEED clustering algorithm constructs clusters and elects CHs from the sensors. The positions of these CHs are used as initial positions for the suboptimal position search algorithm for data collection. Different from the conventional approaches, neither the NCC nor the UAV has any prior knowledge about the CHs' positions. In the discovery phase, the UAV determines the position of the CHs with the help of hello packets. All sensor nodes keep listening for hello packets broadcasted from the UAV. After receiving a hello packet, the sensor nodes send their corresponding CH positions. In the discovery phase, the UAV follows the S-path mobility model. The UAV keeps track of its distance to all CH positions. When a UAV reaches the least distance from its path to a CH location, it visits the CH's position physically. After reaching the CH's position, the UAV runs a modified tabu search algorithm to find out a suboptimal position to collect data from the cluster.

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 position	CBHEED clustering algorithm
Discovery phase	UAV scans the ROI and finds suboptimal data collection positions	Modified tabu search algorithm
Data collection phase	UAV collects sensor data from the calculated position by following optimized trajectory	Genetic algorithm

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 trajectory computed by a modified GA based on the derived data collection positions. We assumed that the deployed nodes are homogeneous, and in every round, the sensors have some data to send. In our EFDC scheme, extra workload of the CHs is reduced by sending the sensors data directly to the UAV. Because of this, the re-election of CH is redundant here. A one-time clustering technique reduces the number of exchanged control packets and decreases the CH energy consumption.

All the nodes store their data in their own buffer memory and transmit them directly to the UAV. Therefore, the chances of memory overflow are also minimized. Again, in the conventional methods, when the CHs are re-elected in every round, the UAV would need to re-initialize the discovery phase. Consequently, the energy consumption will increase along with the exchange of more control packets.

D. EFDC Objective

The objective of this study is to minimize the total transmission power of the sensor nodes and optimize the trajectory of the UAV. The total transmission power is calculated by multiplying single-bit data transmission cost (power) with the number of transmitted bits. Single-bit data transmission cost from a sensor node m of a specific cluster to the UAV can be denoted with ξ_{UAV}^{m}. The number of bits for node m can be denoted with l_m. So, multiplying ξ_{UAV}^{m} with l_m will reveal the energy cost for a single transmission for a single node. If we iterate the transmission cost for all the nodes for all the clusters, we will get the transmission cost for a single round transmission. Minimization of this transmission cost will constitute the first goal of EFDC, which is stated as "energy-efficient data collection". The mathematical expression of this objective is as follows:

$$min\left(\left(\sum_{u\in C}\sum_{m\in u} l_m * \xi_{UAV}^{m}\right)\right), \tag{1}$$

To achieve the goal of the energy-efficient data collection, we apply the tabu search algorithm for finding suboptimal positions of data collection. The algorithm optimizes the transmission distance in all the clusters, resulting in an energy efficient data collection.

EFDC tries to achieve fast data collection by optimizing the travelling distance of UAV for data collection. We can calculate the trajectory by finding out the data collection positions and adding the distances from one data collection position to other positions. The mathematical expression of the second objective of EFDC is as follows:

$$min\left(\sum \delta_{O_i,O_j}\, \varphi_{O_i,O_j}\right), \tag{2}$$

where φ_{O_i,O_j} indicates whether the UAV is taking the path from position O_i to position O_j and the value varies between 0 and 1; and δ_{O_i,O_j} refers to the distance between position O_i and position O_j.

We form the data collection position list O by appending all the designated positions elected from all clusters by tabu search. After this step, we have the data collection positions that will determine the value of φ_{O_i,O_j}. To ensure shortest trajectory, we apply a modified version of the well-known GA to the discovery procedure of the optimized trajectory by forming a TSP problem with O.

E. 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:

$$UAV_{start} = \begin{bmatrix} 0 \\ 0 \\ DA_{UAV} \end{bmatrix}, \tag{3}$$

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

$$UAV_{final} = \begin{bmatrix} RoI_x \\ RoI_y \\ DA_{UAV} \end{bmatrix}, \tag{4}$$

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

$$S_Y = \begin{cases} S_{Y_{Sensors}} * 2, & S_{Y_{Sensors}} \leq S_{Y_{UAV}} \\ S_{Y_{UAV}} * 2, & S_{Y_{UAV}} < S_{Y_{Sensors}} \end{cases}, \tag{5}$$

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:

$$S_{Y_{Sensors}} = \sqrt[2]{{\xi_{SENr}}^2 - {DA_{UAV}}^2}, \tag{6}$$

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

$$S_{Y_{UAV}} = \sqrt[2]{{\xi_{UAVr}}^2 - {DA_{UAV}}^2}, \tag{7}$$

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

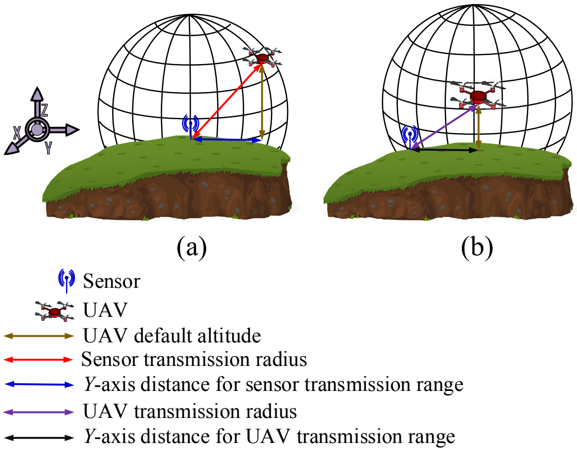

Z
X
Y
(a)
(b)
Sensor
UAV
UAV default altitude
Sensor transmission radius
Y-axis distance for sensor transmission range
UAV transmission radius
Y-axis distance for UAV transmission range

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

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:

$$b_t < b_{t+1} < b_t + \left(\frac{UAV_R^{S_Y}}{DV_{UAV}}\right), \tag{8}$$

where b_t is the time of the previous hello packet, b_{t+1} is the time of the next packet, $UAV_R^{S_Y}$ is the effective ground transmission range after S_Y distance in the *y*-axis, and DV_{UAV} is the default speed of the UAV.

IV. ENERGY-EFFICIENT AND FAST DATA COLLECTION

This section describes the EFDC procedure according to the phases presented in Table II. The CBHEED algorithm is shown as part of the initialization phase. Discovery of the data collection positions is given to represent the discovery phase. Discovering the CH position and deriving the suboptimal position for the data collection mechanism are also given. As part of the data collection phase, the outline of the GA is given.

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 [22] 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 to all nodes in the cluster is 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 nod is not considered. For example, in [23], 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 (line 1). 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 [24].

The outline of the monotone convex hull algorithm is given as Algorithm 2 (line 3). It should be mentioned that for calculating the polygon and the centroid, we took only the x and y coordinates of the neighbors. As, the nodes, having 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 [25], [26] algorithm is used to form the polygon by considering the neighboring nodes and it returns a sorted array of points of the polygon (line 3). The points are then fed into Paul Bourke's equation for calculating the exact centroid position of the polygon (line 5).

To determine the centroid, the area should be calculated first based on the derived coordinates from the convex hull algorithm (line 4). The area can be calculated using with the following equation:

$$A = \frac{1}{2}\sum_{i=0}^{N_{poly}-1}(X_i Y_{i+1} - X_{i+1} Y_i), \tag{9}$$

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 (10) and (11), respectively.

$$Center_x = \frac{1}{6A}\sum_{i=0}^{N_{poly}-1}(X_i + X_{i+1})(X_i Y_{i+1} - X_{i+1} Y_i), \tag{10}$$

and

$$Center_y = \frac{1}{6A}\sum_{i=0}^{N_{poly}-1}(Y_i + Y_{i+1})(X_i Y_{i+1} - X_{i+1} Y_i), \tag{11}$$

where $Center_x$ and $Center_y$ represent the x and y coordinates of the polygon center (line 5). The probability of a node to become a CH can be assigned using the following equation:

$$CH_{prob} = max\left((1 - \frac{\delta_{a,b}}{\rho} \times \omega), \tau_{min}\right), \tag{12}$$

where CH_{prob} is the probability of a node to become a CH (line 7), 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 = {ax, ay} and b corresponds to the polygon's center, where b = {bx, by} (line 6). $\delta_{a,b}$ is calculated using the Euclidian distance formula as indicated below:

$$\delta_{a,b} = \sqrt{(a_x - b_x)^2 + \left(a_y - b_y\right)^2}. \tag{13}$$

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 (13). 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 (line 8).

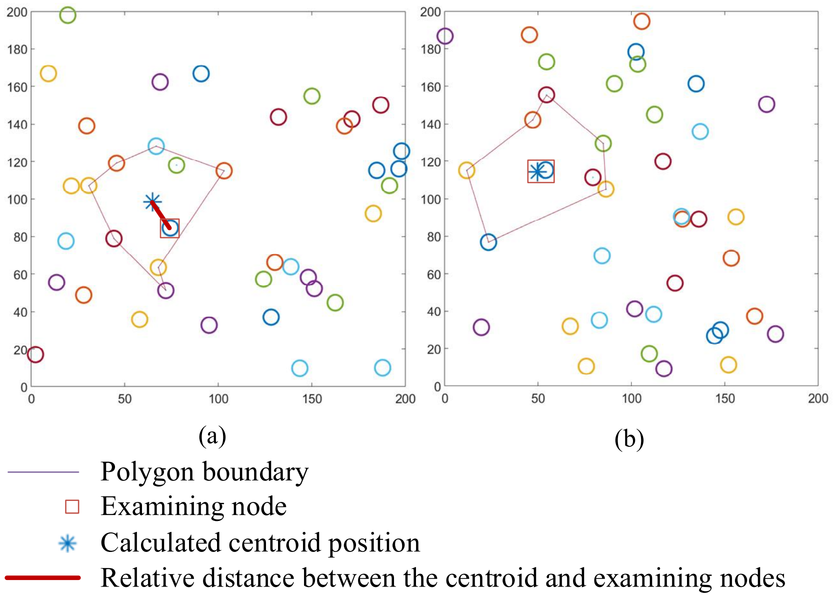


Figure 3. Bias examples: (a) bad centroid bias and (b) good centroid bias.

2. Iteration

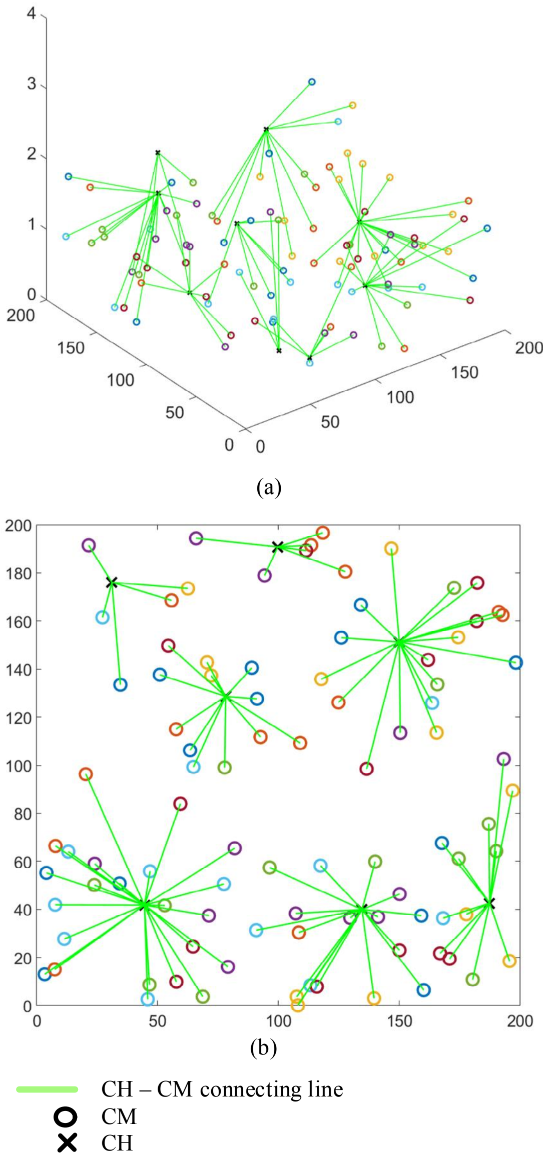
The second step of the clustering algorithm is called the iteration step as given in Algorithm 1. In this step, the nodes compare the cost of the neighboring nodes with their own cost. The least cost node is selected as the temporary CH, expressed as $CH_{selected}$ from the T_{CH} list. The sensor selects its CH by receiving a final CH message from a CH_{final} or the sensor claims itself as a CH_{final} by its own (line 3). 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 (lines 3–11). The CH_{final} flag of a sensor becomes true only when the CH_{prob} value of the node is 1. The T_{CH} gets updated every time a sensor node receives a new declaration of a node as a $CH_{tentative}$ or CH_{final} (lines 6, 9, 13, and 16). The operation can be expressed by the following equation:

$$T_{CH} = \{CHs\ in\ step\ (i-1) \cup CHs\ in\ step\ (i)\}. \tag{14}$$

Every node having a CH_{prob} of 1 declares itself as CH_{final} and broadcasts its status to its neighbors (lines 12–14). Apart from the above cases, a node might not have any CH in its vicinity. In such cases, the node declares itself as $CH_{tentative}$ based on a random value and broadcasts its status (lines 15–16). 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 (lines 19–23). 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 (lines 1–4). 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 requesting node information in the CM table (lines 5–8). 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 (line 9). Figure. 4 displays a graphical representation of the outcome of the clustering algorithm. In the figure, the CHs and corresponding CM connections are shown.



(a)

(b)

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

Algorithm 1: CBHEED

```
Input: {$K^{F_{ngr}}$ | $K_1^{F_{ngr}}$, $K_2^{F_{ngr}}$, $K_3^{F_{ngr}}$, …, $K_n^{F_{ngr}}$ contains the geolocations of sensor node's neighbors}
Output: $CH_{final}$
Initialization
$F_{ngr}$ ← { Neighbor's list based on RSSI value }
Broadcast cost to all nodes ∈ $F_{ngr}$
Forms polygon using algorithm 2 (Monotone chain convex hull algorithm)
Calculates the area of the polygon ($A$) using equation (9)
Calculates $Center_x$ and $Center_y$ axis of the polygon using (10) and (11)
Calculates difference between node's geo-position and polygon centroid position ($\delta_{a,b}$) using (13)
Assigns $CH_{prob}$ value using (12)
$bool\_CH_{final}$ ← $False$
Iteration
while ($True$)
        if (empty ($T_{CH}$) is not equal to $True$)
                $CH_{selected}$ ← min_cost($T_{CH}$)
                if ($CH_{selected}.ID$ is equal to $NID$)
                        if ($CH_{prob}$ is equal to 1)
```

```
                    broadcast_ch_info($NID$, $CH_{status} \leftarrow CH_{final}$, $cost$)
                    $bool\_CH_{final} \leftarrow True$
                else
                    broadcast_ch_info($NID$, $CH_{status} \leftarrow CH_{tentative}$, $cost$)
                end if
            end if
        else if ($CH_{prob}$ is equal to 1)
            broadcast_ch_info($NID$, $CH_{status} \leftarrow CH_{final}$, $cost$)
            $bool\_CH_{final} \leftarrow True$
        else if (Rand(0,1) is less than or equal to $CH_{prob}$)
            broadcast_ch_info($NID$, $CH_{status} \leftarrow CH_{tentative}$, $cost$)
        end if
        $CH_{prev} \leftarrow CH_{prob}$
        $CH_{prob} \leftarrow \min(CH_{prob} \times 2, 1)$
        if ($CH_{prev}$ is equal to 1)
            break
        end if
    end while
```

Finalization

```
if ($bool\_CH_{final}$ is equal to $False$)
    if (contain_Final_CH($T_{CH}$) is equal to $True$)
        $CH_{selected}$ ← min_cost($T_{CH}$)
        cluster_join($CH_{selected}$.ID, NID)
    else
        broadcast_ch_info(NID, $CH_{status}$ ← $CH_{tentative}$, cost)
    end if
else
    broadcast_ch_info(NID, $CH_{status}$ ← $CH_{final}$, cost)
end if
```

4. Polygon Formulation

In the EFDC scheme, the monotone chain convex hull algorithm proposed in [27] is used to form the polygon based on the neighboring nodes of an examining node. The algorithm extends the Graham scan [26] 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 (line 1). $K^{F_{ngr}}$ 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 (lines 2–4). 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 (lines 5–10). Constructing U_{list} is achieved in the same manner as for L_{list} (lines 12–16). 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.

Algorithm 2: Monotone chain convex hull algorithm

Input: $\{K^{F_{ngr}} \mid K_1^{F_{ngr}}, K_2^{F_{ngr}}, K_3^{F_{ngr}}, \ldots, K_n^{F_{ngr}}$ 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)
2. // the $U_{list}$ and $L_{list}$ will hold the upper and lower hulls accordingly
3. $U_{list} \leftarrow \{\}$
4. $L_{list} \leftarrow \{\}$
5. For $i \leftarrow 1$ to length ($K^{F_{ngr}}$)
6.         while ($\varsigma \subset L_{list}$, where n($\varsigma$) is $\geq 2$
                             and
           $K^{F_{ngr}}[i]$ does not make any counterclockwise turn with
           the sequence of the last 2 points of $L_{list}$)
7.         remove($L_{list}$ [last element])
```

```
        append($L_{list}$ [$K^{F_{ngr}}$ [$i$]])
    end while
end for
for $i$ = 1 to length ($K^{F_{ngr}}$)
    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 $U_{list}$)
        remove($U_{list}$ [last element])
        append($U_{list}$ [$K^{F_{ngr}}$ [$i$]])
    end while
end for
remove($U_{list}$ [last_element])
remove($L_{list}$ [last_element])
$K_{hull}^{F_{ngr}}$ = concat($L_{list}$, $U_{list}$)
```

5. Runtime Complexity of the Clustering Process

The runtime of the CBHEED algorithm is similar to that of the original HEED algorithm [22] except for the changes due to the first parameter selection, that is, the central bias calculation. 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(γn), 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. Other than the cost of finding CH_{prob}, where the convex hull algorithm works, the remaining part of the initialization step takes a maximum of $O(n)$ time.

In the worst case, in the iteration step, 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

$$2^{N_{iter}-1} \times \tau_{min} \geq 1, \tag{15}$$

and

$$N_{iter} \leq \left\lceil \log_2 \frac{1}{\tau_{min}} \right\rceil + 1, \tag{16}$$

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 iteration number is constant and $N_{iter} \approx O(1)$. With a maximum number of n CH, the runtime would be $O(1) \times$ runtime of N_{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 a constant amount of time.

In the finalization step, the time complexity is dependent on the number of final CH found by the nodes. The cluster_join() function completes its operation within $O(1)$ time complexity. After the aforementioned analysis, it can be concluded that the complexity of the entire clustering technique is $O(n)$.

B. Discovery of Data Collection Positon

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 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 (3) and (4) as it's 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 (line 1). The UAV broadcasts a CH finding hello message to get a reply from the sensor nodes about their CH's position (line 4). In reply to the hello message, the sensor nodes send their corresponding CH positions back to the UAV using the CSMA protocol. 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 (lines 5–10).

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 (line 12). After reaching a specific position where the distance of a corresponding CH is minimal, the UAV flies into the position (line 13) that saves the UAV flight time. The UAV requests for the cluster information (line 14) 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 (line 15). The UAV then finds the suboptimal data collection place (line 16).

Algorithm 3: Discovering the CH locations

```
Output: $\{O \mid O_1, O_2, O_3, \dots, O_{|C|}$ 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($CH_{locs}, CH_{tempPos}$)
```

```
    if (min_distance_to_a_CH())
            UAV_acquires_CH_position ()
            UAV_requests_cluster_member_information ()
            $CM_{info}$ ← 
            UAV_receives_cluster_member_information ()
            $O_{single}$ = UAV_data_collection_position_search
            $(CM_{info})$
            append($O$, $O_{single}$)
    end if
end while
```

2. Suboptimal Position Search 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.

$$\begin{bmatrix} C_{maxX} \\ C_{maxY} \\ DA_{UAV} \end{bmatrix} - \begin{bmatrix} C_{minX} \\ C_{minY} \\ LA_{UAV} \end{bmatrix} = \begin{bmatrix} C_{rangeX} \\ C_{rangeY} \\ C_{rangeZ} \end{bmatrix} \tag{17}$$

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:

$$\begin{bmatrix} C_{rangeX} \\ C_{rangeY} \\ C_{rangeZ} \end{bmatrix} \odot \begin{bmatrix} \Gamma_x \\ \Gamma_y \\ \Gamma_z \end{bmatrix} = \begin{bmatrix} Step_X \\ Step_Y \\ Step_Z \end{bmatrix} \tag{18}$$

where $\Gamma_x, \Gamma_y,$ 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 $\Gamma_x, \Gamma_y,$ 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 $\Gamma_x, \Gamma_y,$ and $\Gamma_z = 10\%$ of the entire range. Finding the optimal values of $\Gamma_x, \Gamma_y,$ and Γ_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 (19) and (20) corresponding to their axes. By adding and subtracting the fractional value, the UAV will be able to explore all possible places according to the axes line. Besides, the number of searching spaces will be limited. In the meantime, Algorithm 4 will also prevent the UAV to search in a repeated position.

$$\begin{bmatrix} UAV_x \\ UAV_y \\ UAV_z \end{bmatrix} + \begin{bmatrix} Step_X \\ Step_Y \\ Step_Z \end{bmatrix} = \begin{bmatrix} UAV_x^h \\ UAV_y^h \\ UAV_z^h \end{bmatrix}, \tag{19}$$

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:

$$\begin{bmatrix} UAV_x \\ UAV_y \\ UAV_z \end{bmatrix} - \begin{bmatrix} Step_X \\ Step_Y \\ Step_Z \end{bmatrix} = \begin{bmatrix} UAV_x^l \\ UAV_y^l \\ UAV_z^l \end{bmatrix}, \tag{20}$$

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

$$concat\left(\begin{bmatrix} UAV_x^h \\ UAV_y^h \\ UAV_z^h \end{bmatrix} . \begin{bmatrix} UAV_x^l \\ UAV_y^l \\ UAV_z^l \end{bmatrix}\right) = \begin{bmatrix} UAV_x^h & UAV_x^l \\ UAV_y^h & UAV_y^l \\ UAV_z^h & UAV_z^l \end{bmatrix}. \tag{21}$$

Algorithm 4 (line 5) utilizes the matrix derived from (21). 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) Objective Function

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, which is an extension of the Friis free space model [31]. The simplest equation for calculating the RSSI value can be expressed as follows [32]:

$$P_r = P_t * \left(\frac{1}{\delta}\right)^{\eta}, \tag{22}$$

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 [33]. In [34], 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 [34] used the CC2420 transceiver to conduct the experiment. In our case, however, we can safely assume that the value of η is 2 as the probability of LOS communication between UAV and sensor nodes are high. 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 [32]

$$10\log P_r = 10\log P_t - 10\eta\log\delta. \tag{23}$$

If we express P_r in dB as $RSSI$ and $10\eta\log\delta$ as the path loss, the equation can be rewritten as [35]

$$RSSI = P_t - P_{loss}(\delta) \text{ in dBm}, \tag{24}$$

where P_{loss} denotes the path loss expressed in dBm. The log-distance path loss can be described as [36]

$$P_{loss}(\delta) = P_{loss}(\delta_o) + 10\,\eta\log\left(\frac{\delta}{\delta_o}\right), \tag{25}$$

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 (23) into

$$RSSI = P_t - \left(P_{loss}(\delta_o) + 10\,\eta\log\left(\frac{\delta}{\delta_o}\right)\right). \tag{26}$$

Usually, δ_o denotes one unit of distance. By updating the value of δ_o in (25), the following equation can be obtained:

$$RSSI = P_t - P_{loss}(\delta_o) - 10\,\eta\log(\delta). \tag{27}$$

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

$$A = P_t - P_{loss}(\delta_o), \tag{28}$$

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

$$RSSI = A - 10\,\eta\log(\delta). \tag{29}$$

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 (41), 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:

$$f(x, y, z) = max \sum_{i=1}^{|C_n|} RSSI_i \text{ and } min\, \sigma_E, \tag{30}$$

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. 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 assumed that the UAV can measure the RSSI values and based on (40), it can also estimate the energy consumption of the sensors nodes in a cluster. Maximizing the RSSI is the main objective of the searching algorithm, whereas minimizing σ_E works as a tiebreaker, resulting in a better load balancing data collection mechanism. The first portion of the objective function can be written as

$$\textstyle\sum_{i=1}^{|C_n|} RSSI_i = \sum_{i=1}^{|C_n|}(A - 10\,\eta \log(\delta)). \tag{31}$$

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, (31) can be written as

$$\textstyle\sum_{i=1}^{|C_n|}\left(A - 10\,\eta \log\left(\frac{\sqrt{(x_i - UAV_x) + (y_i - UAV_y) + (z_i - UAV_z)}}{\delta_o}\right)\right), \tag{32}$$

and σ_E can be expressed as

$$\sigma_E = \sqrt{\frac{(E_i - \mu_{CE})^2}{|C_n|}}, \tag{33}$$

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:

$$\mu_{CE} = \frac{1}{|C_n|}\sum_{i}^{|C_n|} E_i, \tag{34}$$

where the constraints are

$$RSSI_i \leq \gamma_{RSSI}, \tag{35}$$

$$C_{minX} \leq UAV_x \leq C_{maxX}, \tag{36}$$

$$C_{minY} \leq UAV_y \leq C_{maxY}, \tag{37}$$

and

$$LA_{UAV} \leq z_{UAV} \leq DA_{UAV}. \tag{38}$$

Constraint (35) states that for a single node, the $RSSI$ value must not be less than the threshold limit γ_{RSSI}. The γ_{RSSI} can be expressed by the following equation:

$$\gamma_{RSSI} = \min\left(RSSI_{init}^{C_n}\right), \tag{39}$$

where $RSSI_{init}^{C_n}$ is the list of initial RSSI values in cluster C_n. Constraint (36) states that the UAV cannot select a position with UAV_x, which is out of the cluster's x-axis boundary, that is, C_{minX} and C_{maxX}. Constraint (37) indicates that the y-axis value UAV_y must be inside the boundary expressed by C_{minY} and C_{maxY}, and (38) requires that the z_{UAV} value must be within LA_{UAV} and DA_{UAV}.

3. 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.

Algorithm 4: Suboptimal position search algorithm for data collection

```
Input: $i_{sol}$ = selected CH position, $iter_{num}$ = max iteration number
Output: $O_{pos}$ = best neighboring position
$O_{pos} \leftarrow i_{sol}$
$count \leftarrow 0$
$loc_{visited} = \{\}$
while ($count$ is less than $iter_{num}$)
    to_Visit_Neighbor ← calculate_neighboring_coordinates($O_{pos}$)
    $iter\_best_{pos} \leftarrow O_{pos}$
    for $j$ ← 1 till length(to_Visit_Neighbor)
        $p_{sol}$← to_Visit_Neighbor($j$)
        if ($loc_{visited}$ does not contain $p_{sol}$)
            set_UAV_coordinates($p\_sol$);
            UAV_broadcast _beacon_request();
            UAV_receive_beacon();
            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 ←count+1
25. end while
```

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 (line 5). This function selects the neighboring position based on the fractional value $Step_X$, $Step_Y$, and $Step_Z$ given as (18). The UAV iterates through all the neighboring positions and calculates the fitness function value by (30), except for the positions that the UAV has already visited (line 9). This technique is adopted from the core concept of the tabu search algorithm [37].

In every designated neighboring position, the UAV broadcasts a request for a beacon packet from specific cluster nodes (line 11). In reply, the sensor nodes send beacon signals from which the UAV calculates the RSSI strength for that specific position using (32). The visited positions are recorded after every successful visit to the designated neighboring places and inserted into list $loc_{visited}$ (line 16). After comparing with all the neighboring values, the UAV selects the best neighboring position as its next position (lines 13–15). The best position is updated if any better solution is found (line 22); else, the loop terminates (lines 19–21). The algorithm iterates until an exact number of iterations or the local optimum is found.

4. Data Collection

In the data collection phase, the UAV first formulates the TSP problem based on the derived sub-optimal data collection positions, denoted as O. 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 shortest trajectory a modified GA [38] is applied. The expected runtime of GA is $O(n \log n)$ and good solutions can be found in $O(\log n)$ [39]. Superior runtime complexity and the chances of getting good solutions in lower iteration make GA a favorable option to solve the TSP. 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 [38] and searches for the shortest trajectory for data collection. Figure. 5 shows the optimized trajectory calculated using the GA. 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:

$$g(O) = \delta_{(S,O_1)} + \delta_{(O_{|O|},S)} + \sum_{i=1}^{|O|} \delta_{O_i,O_{i+1}} \tag{40}$$

where, $\delta_{(S,O_1)}$ denotes the distance from the entering position S to the first data collection position O1. $\delta_{(O_{|O|},S)}$ 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 [38]. 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 created. Unlike [38], 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 3: 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

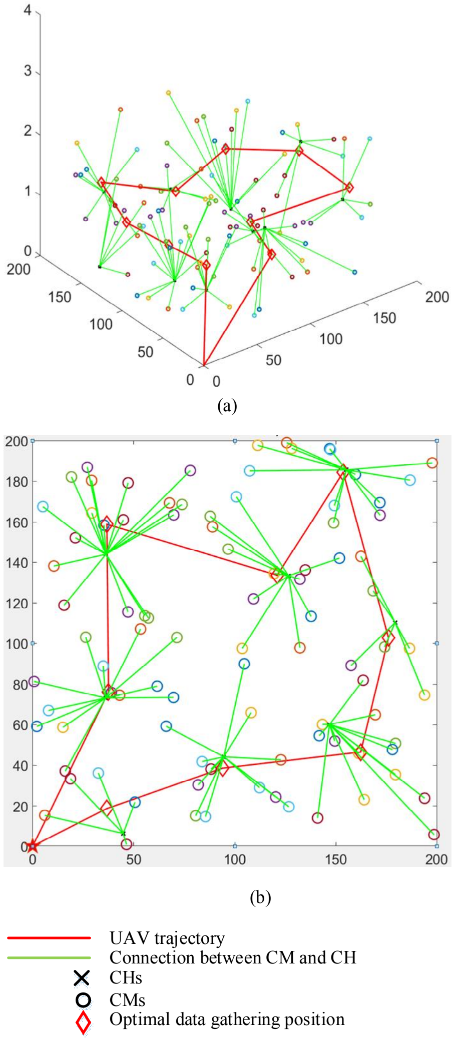
$CR1$={$o_s, o_7, o_4, o_9, o_3, o_8, o_2, o_5, o_1, o_8, o_6, o_s$}

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 [38]. 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.



(a)

(b)

Figure 5. Illustration of the optimized trajectory: (a) 3D view of the trajectory and (b) top view of the trajectory in 2D.

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 III.

Table 3. Simulation parameters

Parameters	**Estimated Value**
Area	200×200–500×500 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%
Packet generation	10/round
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
Path loss exponent (η) UAV↔sensor	2

Path-loss exponent (η)sensor↔sensor	2.45-3.40

We compared our proposed scheme with two other data collection mechanisms, namely LEACH [40] 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 [41], 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:

$$E_{Tx}(l,\delta) = E_{Tx-elec(l)} + E_{Tx-amp(l,\delta)}$$
$$= \begin{cases} l * E_{elec} + l * \psi_{fs} * \delta^2, & \delta < \delta_{th} \\ l * E_{elec} + l * \psi_{mp} * \delta^4, & \delta > \delta_{th} \end{cases}, \tag{41}$$

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:

$$\delta_{th} = \sqrt{\frac{\psi_{fs}}{\psi_{mp}}}. \tag{42}$$

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:

$$E_{Rx}(l) = E_{Rx-elec} * l, \tag{43}$$

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 (41) and (43). 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:

$$\sum_{r_n=1}^{R_n} \sum_{j \in C} \sum_{i \in j} \left(\sum E_{Tx_i} + \sum E_{Rx_i} + \sum E_{Ag} \right), \tag{44}$$

where E_{Tx_i} and E_{Rx_i} correspond to the energy consumption of node i due to the transmission and receiving of signals, respectively. E_{Ag} represents the consumed energy due to data aggregation-based computation. The definitions of E_{Tx_i} and E_{Rx_i} are given by (41) and (43), respectively. C is the list of clusters and R_n stands for the number of rounds taken into consideration for the calculation.

C. Delay Model

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

$$\sum_{r_n=1}^{R_n}\left(\frac{1}{DV_{UAV}}\left(\delta_{(S,O_1)}+\delta_{(O_{|O|},S)}+\sum_{i=1}^{|O|}\delta_{O_i,O_{i+1}}\right)+\sum_{j\in C}\sum_{i\in j}£\right), \tag{45}$$

where DV_{UAV} is the default speed of the UAV, $\delta_{O_i,O_{i+1}}$ represents the distance between position O and next position O_{i+1}, $\delta_{(S,O_1)}$ is the distance between the starting point S and the first data collection position O_1, and $\delta_{(O_{|O|},S)}$ means the distance between the final data collection position $O_{|O|}$ and the starting and exiting point S of the ROI. To simplify the equation, we considered the delay for all transmissions to be equal and expressed it as £.

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.

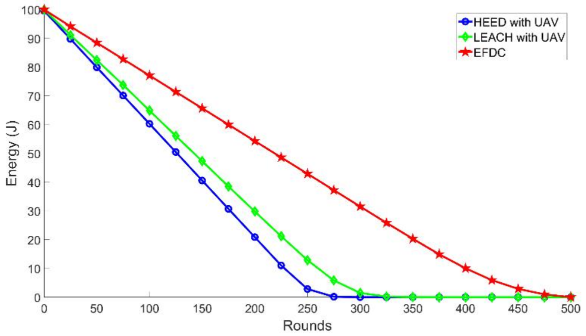


Figure 6. Energy performance with linear S-path approach.

Figure. 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 the figure, 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 (41), 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.

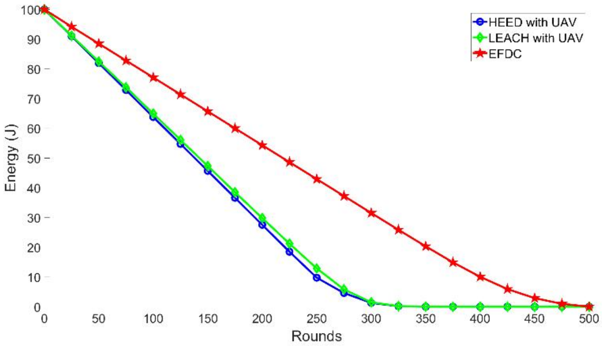


Figure 7. Energy performance with data collection approach from the CH position.

Figure. 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 Figure. 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.

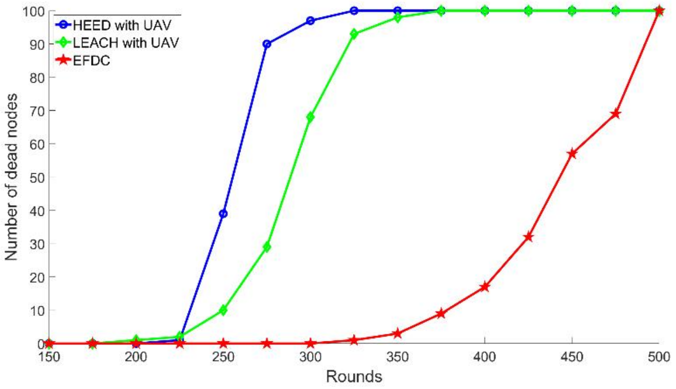


Figure 8. Number of dead nodes with linear S-path approach.

Figure. 8 presents the comparison of dead nodes per round among the proposed EFDC and the compared schemes following 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.

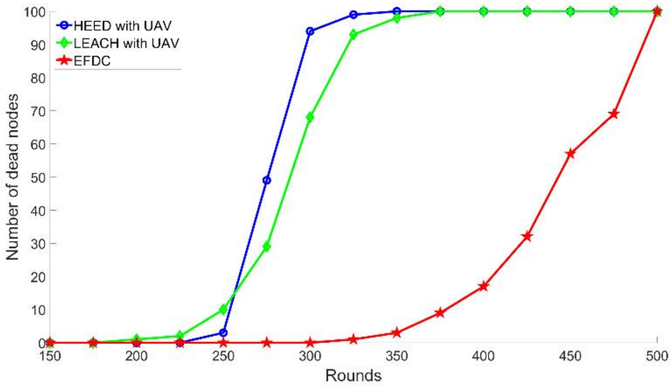


Figure 9. Number of dead nodes with data collection approach from the CH position.

Figure. 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.

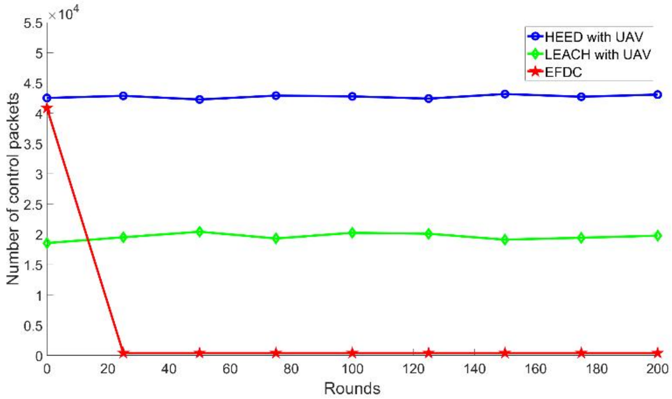


Figure 10. Number of control packets versus number of rounds.

Figure. 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.

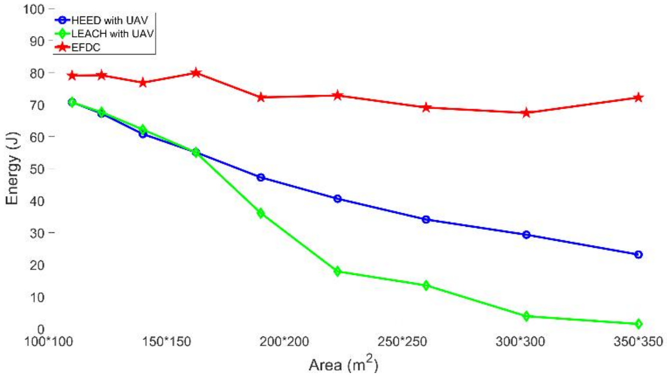


Figure 11. Energy consumption versus network area.

Figure. 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 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.

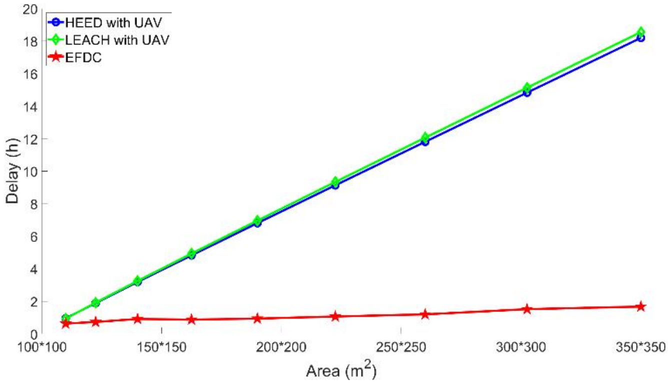


Figure 12. Data collection delay versus network area.

Figure. 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 (45). 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.

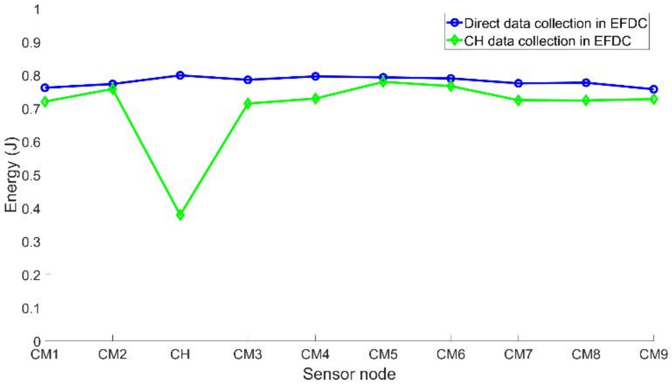


Figure 13. Energy consumption at different nodes using CM-UAV direct transmission and CM-CH-UAV transmission.

Figure. 13 is presented to analyze the energy depletion comparison between direct data collection and CH data collection. 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 figure. 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.

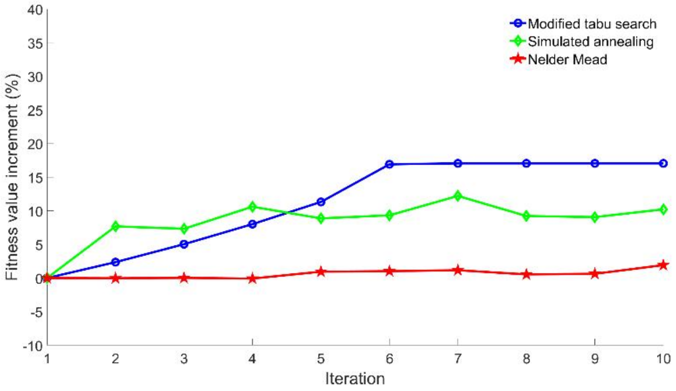


Figure 14. Convergence along with iterations.

Figure. 14 shows the convergence along with iterations for the three algorithms of the modified tabu search, simulated annealing [42], and Nelder Mead optimization [43]. 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 moderate 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.

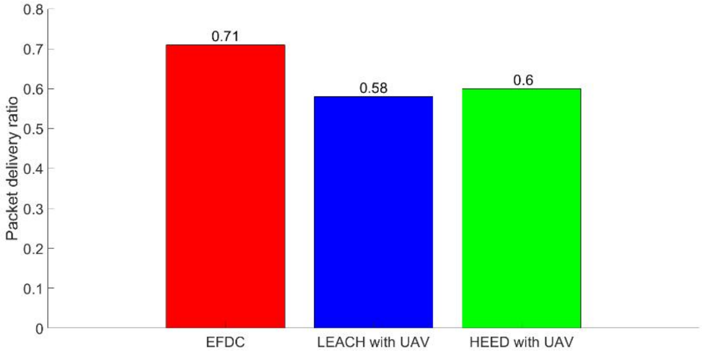


Figure 15 Comparison of packet delivery ratio

In figure 15 the comparison of normalized value of packet delivery ratio (PDR) among the investigating frameworks are given. From the figure it can be observed that the EFDC outperforms other two compared frameworks in terms of PDR also. The outcome of this experiment is reasonable due to the practical assumption of the η. We have assumed the value of η is between 2.45 to 3.40 and assigned randomly for all the edges WSN. Whereas, the value of η is assumed to be 2 for the UAV-sensor and sensor-UAV communication, ensures no packet loss due to weak signal. Another reason of packet drop is the adopted mobility model of the compared protocols. EFDC follows an optimized trajectory and gets a larger data transmission window. In case of the compared protocols, they follow a S-path mobility model, where the data transmission window is less and the packet drop increases.

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. We formulated a joint optimization problem in this regard and divided the problem into two parts. 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 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.

BIBLIOGRAPHY

[1] Z. A. Ali, S. Masroor, and M. Aamir, "UAV Based Data Gathering in Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 106, no. 4, pp. 1801–1811, 2019.

[2] B. Liu and H. Zhu, "Energy-effective data gathering for uav-aided wireless sensor networks," *Sensors (Switzerland)*, vol. 19, no. 11, pp. 1–12, 2019.

[3] 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, 2019.

[4] C. Zhan, Y. Zeng, and R. Zhang, "Energy-Efficient Data Collection in UAV Enabled Wireless Sensor Network," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 3, pp. 328–331, 2018.

[5] S. Say, H. Inata, J. Liu, and S. Shimamoto, "Priority-Based Data Gathering Framework in UAV-Assisted Wireless Sensor Networks," *IEEE Sens. J.*, vol. 16, no. 14, pp. 5785–5794, 2016.

[6] 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.

[7] S. Poudel and S. Moh, "Medium Access Control Protocols for Unmanned Aerial Vehicle-Aided Wireless Sensor Networks: A Survey," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 65728–65744, 2019.

[8] 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. Robot. Syst.*, vol. 78, no. 1, pp. 159–179, Apr. 2015.

[9] D. Popescu, F. Stoican, L. Ichim, G. Stamatescu, and C. Dragana, "Collaborative UAV-WSN system for data acquisition and processing in agriculture," in *Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*, 2019, vol. 1, pp. 519–524.

[10] C. You and R. Zhang, "3D Trajectory Optimization in Rician Fading for UAV-Enabled Data Harvesting," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 6, pp. 3192–3207, 2019.

[11] J. Mi, X. Wen, C. Sun, Z. Lu, and W. Jing, "Energy-efficient and low package loss clustering in UAV-assisted WSN using kmeans++ and fuzzy logic," in *2019 IEEE/CIC International Conference on Communications Workshops in China, ICCC Workshops 2019*, 2019, pp. 210–215.

[12] J. Chen *et al.*, "Efficient Data Collection in Large-Scale UAV-aided Wireless Sensor Networks," in *2019 11th International Conference on Wireless Communications and Signal Processing, WCSP 2019*, 2019.

[13] 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 *2014 IEEE Global Communications Conference, GLOBECOM 2014*, 2014, pp. 234–239.

[14] 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 *2019 UK/China Emerging Technologies, UCET 2019*, 2019.

[15] F. Stoican, D. Popescu, and L. Ichim, "Trajectory Design for Effective and Secure Communication in UAV-WSN Systems," in *2019 IEEE Radio and Antenna Days of the Indian Ocean, RADIO 2019*, 2019.

[16] J. Yang *et al.*, "Path planning of unmanned aerial vehicles for farmland information monitoring based on WSN," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2016, vol. 2016-September, pp. 2834–2838.

[17] J. Grigulo and L. B. Becker, "Experimenting Sensor Nodes Localization in WSN with UAV Acting as Mobile Agent," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2018, vol. 2018-September, pp. 808–815.

[18] D. S. Jasrotia and M. J. Nene, "Localisation using UAV in RFID and Sensor Network Environment:Needs and Challenges," 2020, pp. 274–279.

[19] G. Liu, Z. Li, X. Zhou, and S. Li, "Transmission Power Control for Wireless Sensor Networks," in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 2007, vol. 9258, pp. 2596–2599.

[20] 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.

[21] 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. Robot. Syst. Theory Appl.*, vol. 78, no. 1, pp. 159–179, 2015.

[22] O. Younis and S. Fahmy, “HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Trans. Mob. Comput.*, vol. 3, no. 4, pp. 366–379, 2004.

[23] S. H. Ahmed, S. H. Bouk, N. Javaid, and I. Sasase, “RF propagation analysis of MICAz mote’s antenna with ground effect,” *2012 15th Int. Multitopic Conf. INMIC 2012*, pp. 270–274, 2012.

[24] P. Bourke, “Calculating the area and centroid of a polygon. Available from: http://www.seas.upenn.edu/~sys502/extra_materials/Polygon%20Area%20and%20Centroid.pdf,” no. C, pp. 3–5, 1988.

[25] G. Mei, J. C. Tipper, and N. Xu, “An algorithm for finding convex hulls of planar point sets,” in *Proceedings of 2nd International Conference on Computer Science and Network Technology, ICCSNT 2012*, 2012, pp. 888–891.

[26] 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.

[27] A. ANDREW and A. AM, “ANOTHER EFFICIENT ALGORITHM FOR CONVEX HULLS IN TWO DIMENSIONS,” *ANOTHER Effic. ALGORITHM CONVEX HULLS TWO Dimens.*, 1979.

[28] 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.

[29] A. E. Lagias, T. D. Lagkas, and J. Zhang, “New RSSI-Based Tracking for Following Mobile Targets Using the Law of Cosines,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 3, pp. 392–395, Jun. 2018.

[30] 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.

[31] F. Lassabe, P. Canalda, P. Chatonnay, F. Spies, and O. Baala, “A Friis-based calibrated model for WiFi terminals positioning,” in *Proceedings*

- *6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005*, 2005, pp. 382–387.

[32] A. A. Hussein, T. A. Rahman, and C. Y. Leow, “Performance evaluation of localization accuracy for a log-normal shadow fadingwireless sensor network under physical barrier attacks,” *Sensors (Switzerland)*, vol. 15, no. 12, pp. 30545–30570, 2015.

[33] J. Miranda *et al.*, “Path loss exponent analysis in Wireless Sensor Networks: Experimental evaluation,” *IEEE Int. Conf. Ind. Informatics*, pp. 54–58, 2013.

[34] 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 (Switzerland)*, vol. 19, no. 8, 2019.

[35] Akhand Pratap Singh, Devesh Pratap Singh, and S. Kumar, “NRSSI: New proposed RSSI method for the distance measurement in WSNs,” in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, 2015, pp. 296–300.

[36] M. Cheffena and M. Mohamed, “Empirical Path Loss Models for Wireless Sensor Network Deployment in Snowy Environments,” *IEEE Antennas Wirel. Propag. Lett.*, vol. 16, pp. 2877–2880, Sep. 2017.

[37] S. Pierre and F. Houéto, “Assigning cells to switches in cellular mobile networks using taboo search,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 32, no. 3, pp. 351–356, Jun. 2002.

[38] T. Zhou, “TSP problem solution based on improved genetic algorithm,” *Proc. - 4th Int. Conf. Nat. Comput. ICNC 2008*, vol. 1, pp. 686–690, 2008.

[39] 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, 2014.

[40] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, pp. 660–670, 2002.

[41] P. Nayak and A. Devulapalli, “A Fuzzy Logic-Based Clustering Algorithm for WSN to Extend the Network Lifetime,” *IEEE Sens. J.*, vol. 16, no. 1, pp. 137–144, Jan. 2016.

[42] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Springer Netherlands, 1987.

[43] D. M. Olsson and L. S. Nelson, “The nelder-mead simplex procedure for function minimization,” *Technometrics*, vol. 17, no. 1, pp. 45–51, 1975.

ACKNOWLEDGEMENT

I would like to give my sincere gratitude to all the people that have helped me in the time of pursuing my Master's study and research.

First of all, I would like to express my immense respect and gratefulness to my advisor, Prof. Sangman Moh for giving me the opportunity to pursue my Master's degree at Chosun University. His constant guidance, support, and useful suggestions have encouraged and directed me through difficulties during my study and research works. His supervision and persistent guidelines have motivated me towards performing good research works. I would always be indebted to him for the useful lessons of professionalism, time-management, and impeccable mentorship.

Secondly, I wish to express my warm and sincere thanks to the thesis committee members, Prof. Il Yong Chung and Prof. Wooyeol Choi for their constructive comments and invigorating suggestions. All their insights regarding my research works have helped me in improving and extending it in different perspectives.

Thirdly, I am beholden for the chance to be a part of such a unique group of students, faculty, and staff in the Department of Computer Engineering, Chosun University. Also, I would like to show my sincere acknowledgment to Mobile Computing Lab for providing me such a wonderful opportunity and an atmosphere to grow me academically and otherwise. I am thankful to my lab-mates for their moral as well as academic support. Furthermore, I would like to heartily thank all my seniors and friends from Bangladesh at Chosun University for their affection and cooperation that made my life easier and cheerful in South Korea.

Last but not the least, I would like to thank my parents, family members, and friends for their constant support in difficult times. It would have been impossible for me to achieve anything without their encouragement and assistance. I would like to dedicate my work to them.