future internet

MDPI

Article

MCCM: An Approach for Connectivity and Coverage Maximization

Alia Ghaddar [1,*], Monah Bou Hatoum [2], Ghassan Fadlallah [3] and Hamid Mcheick [3]

[1] Department of Computer Science, International University of Beirut, Beirut 146404, Lebanon
[2] Department of Computer Science, Lebanese University, Rafic Hariri Educational Campus, Beirut 146404, Lebanon; monah1984@gmail.com
[3] Department of Computer Science, University of Quebec at Chicoutimi, Chicoutimi, QC G7H 2B1, Canada; Ghassan-mustapha.fadlallah1@uqac.ca (G.F.); hamid_mcheick@uqac.ca (H.M.)
* Correspondence: alia.ghaddar@liu.edu.lb

Received: 27 December 2019; Accepted: 18 January 2020; Published: 21 January 2020

Abstract: The internet of Things (IoT) has attracted significant attention in many applications in both academic and industrial areas. In IoT, each object can have the capabilities of sensing, identifying, networking and processing to communicate with ubiquitous objects and services. Often this paradigm (IoT) using Wireless Sensor Networks must cover large area of interest (AoI) with huge number of devices. As these devices might be battery powered and randomly deployed, their long-term availability and connectivity for area coverage is very important, in particular in harsh environments. Moreover, a poor distribution of devices may lead to coverage holes and degradation to the quality of service. In this paper, we propose an approach for self-organization and coverage maximization. We present a distributed algorithm for "Maintaining Connectivity and Coverage Maximization" called *MCCM*. The algorithm operates on different movable devices in homogeneous and heterogeneous distribution. It does not require high computational complexity. The main goal is to keep the movement of devices as minimal as possible to save energy. Another goal is to reduce the overlapping areas covered by different devices to increase the coverage while maintaining connectivity. Simulation results show that the proposed algorithm can achieve higher coverage and lower nodes' movement over existing algorithms in the state of the art.

Keywords: coverage maximization; connectivity; wireless sensor networks (WSNs); heterogeneous devices; Internet of Things (IoT)

1. Introduction

Growth in technology and computing power is continuously progressing and transforming the way societies connect to each other and to the physical world. The tendency to bring smartness to living and environments to make them more adaptive, autonomous and personalized to human needs, prepares the ground to smart worlds [1]. The Internet of Things (loT) is defined by the International Telecommunication Union (ITU) [2] and the European Research Cluster on the Internet of Things (IERC) [3] as a dynamic global network infrastructure with self-configuring capabilities based on standard and inter-operable communication protocols. Mostly a smart network of linked objects and machines, which communicate information via wireless technology and the cloud [4]. Wireless Sensor Network (WSN) plays an important role in conjunction with IoT. It consists of many distributed sensors for environmental monitoring tasks. It brings more control, precisions, early alerts, monitoring and tracking the environment. This is reflected in many projects such as Array of Things (AoT) [5], NEOM City [6,7], Google's Loon [8], etc. Google's Loon project aims to create an aerial wireless network formed by nodes such as satellites, balloons, drones, or fixed-wing aircraft. These flying nodes provide network access and rescue missions over areas in disaster scenarios.

Although IoT has great benefits, it has some major challenges such as: Connectivity and coverage, energy consumption, reliability and privacy. The initial deployment of devices in an area of interest might not always meet the mission needs (rescue or military). The basic requirement after deployment is to have sufficient coverage and connectivity [9]. In many cases, the mobility feature of devices is used to maximize the coverage [10]. Wireless sensor devices can sense, process, and in some applications move and organize themselves for coverage missions. These tasks drain devices' energy and become very challenging especially for power-constrained devices [11]. For this purpose, energy saving during coverage maximization should be considered, and it is necessary to cover all targets areas with a minimum number of devices.

In this paper, we are interested in maintaining connectivity and coverage in wireless distributed networks composed of heterogeneous mobile devices. Our work consists of three parts (1) a distributed algorithm for Coverage Maximization and Maintaining Connectivity (MCCM) between group of neighboring devices, (2) gaps bridging between disjoint groups, and (3) comparison and validation of the proposed distributed algorithm. The algorithm ensures that each device is connected to at least two other devices in the network. This setup increases the network resistance against failure as well as reliability.

The remaining part of the paper is organized as follows: In Section 2, we investigate related works on connectivity and coverage maximization. The motivation and contribution of the work are presented in Sections 3 and 4 respectively. The performance of the algorithm as well as a comparative study with related works are discussed in Section 5. Section 6 concludes the paper.

2. Related Works

Maintaining connectivity between devices and maximizing coverage is big challenge to reduce the cost of deployment, and to ensure the reliability of data. They are key issues in WSNs as they measure how effectively a network field is monitored by its sensor devices. Coverage is to guarantee that each area is monitored by at least one node, and connectivity is to guarantee that any device can communicate with each other [12]. Protocols related to area coverage are classified based on the sensor types (mobile, static) and the coverage requirement (1-coverage or *k*-coverage) [13]. Different algorithms [14–19] were proposed in the literature and have good impact on the coverage optimization. Protocols can be classified into either distributed or centralized. The type of sensors in the network can be homogeneous (The sensors have the same sensing range and technical properties) or heterogeneous. Studies such as CTC_k [16] and $DCTC_k$ [16] take into account energy efficiency for coverage and connectivity optimization. Their algorithms work for centralized and distributed coverage in heterogeneous networks and tackle the k-coverage problem. In $DCHGA$ [20] (Dynamic Clustering of Heterogeneous WSNs using Genetic Algorithm), authors present a genetic algorithm. The proposed work can operate on both homogeneous and heterogeneous environments. This algorithm requires a Central Station (CS) with advanced resources to control the devices distributed in the network. Another work ($HMRC$) [21] uses the harmony search algorithm to improve connectivity and coverage within an area of interest. The algorithm in [21] only works on networks where the devices are homogeneous and controlled by a Sink Node (SN) or Central Station (CS).

The authors in [22], presented an asynchronous algorithm ($AAEECC$) that is energy efficient and provides integrated coverage and connectivity. It is fully distributed and scalable. They also presented a fault-tolerance connectivity maintenance algorithm ($FTCM$) to increase the resistance against failure that might occur during the nodes scheduling. In [23], authors presented an algorithm based on Voronoi partition. The algorithm eliminates the isolated nodes and reduces the redundant coverage. Nodes move to their final best position to maintain network connectivity. A deployment technique ($ABSDC$) is suggested in [24] to enhance coverage, connectivity, and reliability. The idea is to use the angle between sensor nodes and their neighbors. The work showed higher coverage percentages and good connectivity compared to the Glowworm Swarm Optimization (GSO) approach [25]. This algorithm works on homogeneous environments only. Another algorithm (OPDCC) [26] addresses the $k-$coverage and $m-$connected problem in WSN. It estimates the minimum number of nodes (and their locations) required for the connectivity and coverage. The proposed algorithm requires a central station (CS) to do all the calculations.

Another novel algorithm for Coverage Maximization of Heterogeneous Wireless Network ($CMHN$) is proposed in [27]. The algorithm is centralized. Devices are heterogeneous and the sensing range of each device has a circular shape. The circumference of the circle is represented as an interval of an angle (0, 2Π). For any two connected devices D_i and D_j with sensing ranges r_i and r_j respectively, the distance d_{ij} between them is less than the sum of their ranges. A device moves (after random deployment) to a position that ensures a maximum coverage and connects to a closest device to maintain connectivity. The main goal is to keep the movement of devices as minimal as possible to save energy. Another goal is to reduce the overlapping areas covered by different devices and keep it to a small margin to maximize the coverage while maintaining connectivity. $CMHN$ ensures that every device is connected to at least $k = 2$ devices. In this algorithm, the notion of $m-$ratio is introduced. It adds flexibility in controlling the overall displacement in the network.

$$m(i,j) = m_{ij} = \left\{ \begin{matrix} m, & r_i < r_j \\ \frac{m+1}{2}, & r_i = r_j \\ 1, & r_i > r_j \end{matrix} \right\} \tag{1}$$

where $m \in [0,1]$ is a user-defined value. To determine the location to which a device will move, authors introduced two rules (called the "empty-arc" rules). A device D_i has 2π empty arc when there is no other device D_j such that $d_{ij} < r_i + r_j$. In other words, it has no intersection area with another device on its circumference. An empty arc has initially the value $2\pi r_i$. The value of the circumference starts to decrease when other devices start to intersect with the device D_i until it reaches zero. It is worthy to mention that the device D_i starts to fill this empty arc during the maximization process and any overlapping before the maximization process is ignored by the device D_i. If a device D_i has empty arc the minimum required overlapping between D_i and another device D_j, is defined by $\tau_{ij} = m_{ij} \cdot r_i + m_{ji} \cdot r_j$. D_i moves to a position that guarantees τ_{ij}. This is called "circle-circle" connection. Another rule is "circle-two circle" connection. Given two intersecting circles, a new coming device D_i, will be able to connect to them if it has an empty arc. D_i moves along the line through the two crossing points of the crossed circles. It chooses the closest point of intersection and connects to the closest circle that has free arc on that point [27]. Simulation results showed a significant improvement in the coverage gain.

3. Motivation

The examined algorithms in the related works require higher computational resources like $DCHGA$ [20], $HMRC$ [21], $VCACWSN$ [23], $OPDCC$ [26], $DCTC_k$ [16], and $AAEECC$ [22]. We will present this part in Section 5. Applying the aforementioned algorithms in distributed environments to run locally on each device, drains the battery energy quickly. Consequently, they might not behave well in networks where devices are energy-constrained. Although $ABSDC$ [24] presents an efficient algorithm and requires low energy consumption. However, authors adopted the triangular circular packing idea. This approach forces a fixed angle between the connected devices as well as a fixed distance between them. Every device should be $r * \sqrt{3}$ apart from its neighbor devices where r is the device's range. This restricted the algorithm to work on homogeneous environments only. Also, the overlapping between devices is constant and is the same between all devices which is not suitable for heterogeneous environments.

$CMHN$ [27] is designed to maintain connectivity and maximize the coverage. It minimizes energy consumption by reducing the number of nodes' displacement in the network. However, it is a centralized algorithm. In some scenarios, the devices need to take decisions locally, without waiting to be told what to do by a central station, such as in harsh environments, wide forests, monitoring huge amounts of cattle. In such situations, the adoption of distributed algorithm is more appropriate. By investigating the behavior of this algorithm, we noticed the formation of gaps/holes in some scattered distributions. It occurs when a device D_i moves to its final position, it tries to find the closest device D_j that can intersect with (i.e., a device that has enough free arc on its circular range for connection).

In this paper, our work is based on $CMHN$. We propose a distributed version called $MCCM$, for maintaining connectivity and maximizing coverage in disaster and harsh scenarios. The basic idea is that devices are deployed randomly and need to re-organize themselves. Every device takes the decision to move to a nearest location to maximize its coverage while maintaining connectivity with others. We aim to have fewer movements and energy drain. Another part we tackle in this work is that after the self-organizing process is done, disjoints groups of connected devices might appear in the network. We call them "islands" of devices. We bridge the islands to ensure more connectivity using alpha-shape technique to find the islands' boundaries and shortest path between adjacent boundary sets.

4. Distributed Maintaining Connectivity and Coverage Maximization (MCCM)

In this section, we present the two main parts of our work. The first is $MCCM$; a distributed algorithm which runs independently on each device D_i for coverage maximization. Then, the bridging part that connects the isolated islands using alpha-shape method. The approach consists of three steps as shown in Figure 1: Neighborhood discovery, coverage maximization and bridging process.

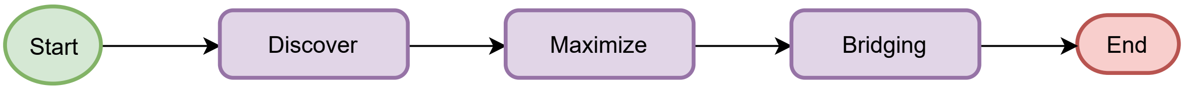


Figure 1. $MCCM$ process steps.

4.1. Neighborhood Discovery Process

This process builds the neighborhood list for each device D_i. Every device will know about the other devices connected directly or indirectly to it. We denote Ω the set of n heterogeneous devices in the network:

$$\Omega = \{D_i | i \in [1, n]\}$$

We denote D_i the $i-$th device in the network with center (x_i, y_i) and range r_i. The distance $dist\,(D_i, D_j)$ between two devices D_i and D_j is the Euclidean distance denoted d_{ij}. We say two devices D_i and D_j are “neighbors”, if their sensing ranges intersect.

D_i is directly connected to D_j when $d_{ij} < r_i + r_j$ and both devices have minimal overlapping τ_{ij} according to [27]. In this case, we denote $D_i \diamond D_j$.

We say D_i is indirectly connected to D_j when there is another device D_f, such that $D_i \diamond D_f$ and D_j is directly/indirectly connected to D_f. We denote this case $D_i \bullet D_j$.

An island (or group) of connected devices to D_i is denoted $\mathcal{N}_i$. It is the set of k devices that includes D_i and its neighbors as follows:

$$\mathcal{N}_i = \{D_j \in \Omega | D_j \diamond D_i \vee D_j \bullet D_i\}$$

where $|\mathcal{N}_i| = k$ and $k \leq n$.

Figure 2 shows how the discovery process occurs on a device $D_i \in \mathcal{N}_i$.

The main steps are as follows:

- Any device D_i waits for a certain amount of time t to emit a discovery signal. It is allowed to make c trials.
- Upon receiving a signal from a device D_j, both devices start the "Add to neighborhood list" process, only if they have empty arcs on their circumferences. If so D_j is added to $\mathcal{N}_i$. The devices in $\mathcal{N}_i$ update their list about the newly coming device D_j.
- When the emit counter exceeds c times, the device goes to sleep for a certain period of time n. Then, it tries again until the sleeping count exceeds a value s.
- The discovery process ends when the device does not have any more empty arcs or the sleeping counter reaches s.

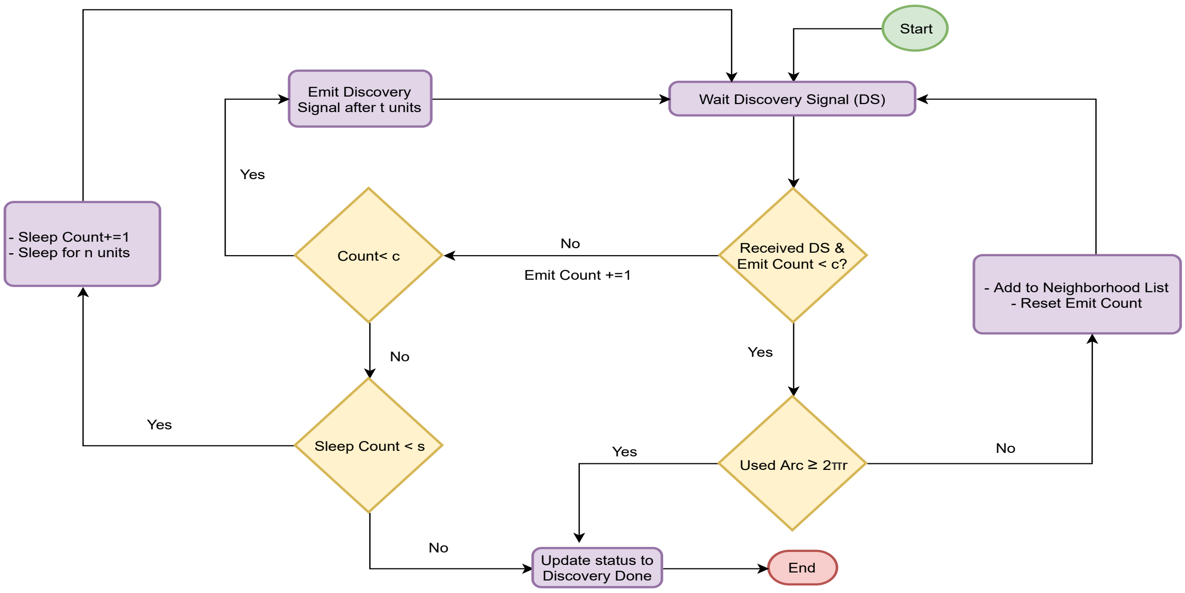

Start
Emit Discovery Signal after t units
Wait Discovery Signal (DS)
Yes
- Sleep Count+=1
- Sleep for n units
Count< c
No
Emit Count +=1
Received DS & Emit Count < c?
- Add to Neighborhood List
- Reset Emit Count
No
Yes
Yes
Sleep Count < s
Yes
Used Arc ≥ 2πr
No
No
Update status to Discovery Done
End

Figure 2. Neighborhood discovery process of device D_i.

4.2. Coverage Maximization

This process starts once the discovery process finishes. It occurs in each island $\mathcal{N}_i$, if it has more than one device (i.e., $\mid\mathcal{N}_i\mid>1$). Figure 3 shows how the maximization process works. The main idea is to find for each device D_i its location, to maximize the coverage.

The steps are as follows:

- A device $D_i \in \mathcal{N}_i$ calculates the mean coordinates $\mathcal{M}(x_m, y_m)$. It is the mean location of the devices in $\mathcal{N}_i$, where $x_m = \frac{1}{k}\sum_{l=1}^{k}(x_l)$, and $y_m = \frac{1}{k}\sum_{l=1}^{k}(y_l)$, such that $D_l(x_l, y_l) \in \mathcal{N}_i$.
- A device D_i moves to the location $\mathcal{M}(x_m, y_m)$ if it is the closest device to it, thus:

$$dist(D_i, \mathcal{M}) = min\{dist(D, \mathcal{M}), \forall D \in \mathcal{N}_i\}$$

  The device then updates its status to ready for the monitoring mission.
- Otherwise, D_i finds its next location by guessing the behavior of its adjacent devices located closer to $\mathcal{M}$. D_i orders its neighbors by their initial position with respect to the mean location $\mathcal{M}$. Then it chooses the neighbors that are closer to $\mathcal{M}$. It estimates their positions and moves according to the estimated position. Assume the sequence D_{i+1}, D_i, D_{i-1} and D_{i-2} is ordered from the farthest to the closest devices to $\mathcal{M}$. Device D_{i+1} is the farthest one. D_i calculates its final position in two ways: (1) either it relies on D_{i-1} estimated position, if the circle-to-circle method is to be applied, (2) or it relies on D_{i-1} and D_{i-2} positions if the circle-to-two circles method is to be used. D_{i+1} does the same. It relies on D_i and/or D_{i-1} estimated positions, and so. For instance, let ζ be the set of devices located between D_i and $\mathcal{M}(x_m, y_m)$.

$$\zeta = \{D \in \mathcal{N}_i | min(y_i, y_m) \leq y_D \leq max(y_i, y_m) \wedge min(x_i, x_m) \leq x_D \leq max(x_i, x_m)\}$$

D_i evaluates the possible location of each device $D \in \zeta$ by applying the empty-arc rules. Then it moves towards the closest device D_k and connects to it, where $D_k = f(D_i, \zeta)$:

$$f(D_i, \zeta) = \{D_k \in \zeta | (D_k \diamond \mathcal{M} \vee D_k \bullet \mathcal{M}) \wedge dist(D_i, D_k) = min\{dist(D_i, D)\}\}$$

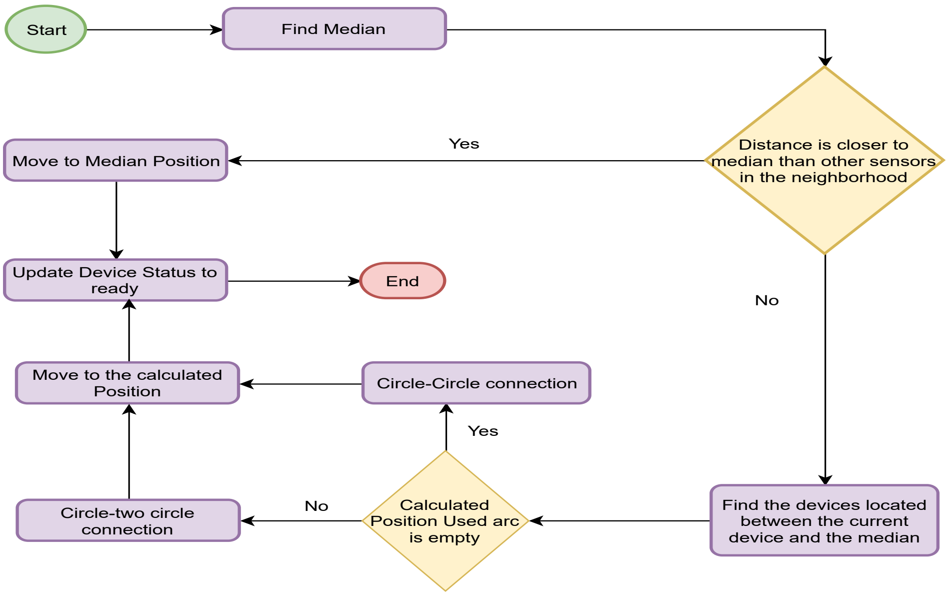

Start
Find Median
Distance is closer to median than other sensors in the neighborhood
Yes
Move to Median Position
Update Device Status to ready
End
No
Move to the calculated Position
Circle-Circle connection
Yes
Find the devices located between the current device and the median
Calculated Position Used arc is empty
No
Circle-two circle connection

Figure 3. Neighborhood maximization process of device D_i.

4.3. *Bridging*

In the earlier section, we introduced the maximization process which runs in a distributed manner. Every device knows about the devices that belong to the same neighborhood only. This might form a network with different disconnected neighborhoods (islands) or isolated devices. We mean by isolated device, the one that does not belong to any neighborhood, and we refer to it as an island with only one device. One of the real examples for such problems is losing the connectivity between the different troops in the battle fields. To provide connectivity between the different isolated islands. We need to build connections (bridges) between these islands. These bridges do not need to cover the whole area. It is enough to have them between these islands and the central station to maintain the flow of data. In this section, we provide two different type of bridging: A distributed and centralized bridging.

4.3.1. Centralized Bridging

At the end of the maximization process, the network might be composed of different islands in which the coverage is maximized. Thus, we have $\Omega = \cup_{i=1}^{t} \mathcal{N}_i$ where t is the number of islands in the network. For any two islands $\mathcal{N}_i \in \Omega$ and $\mathcal{N}_j \in \Omega$, we have $\mathcal{N}_i \cap \mathcal{N}_j = \phi$ where $i \neq j$ (Figure 4).

To bridge the islands, we use flying nodes or drones. This is useful in disaster locations that humans cannot reach. The goal is to find the closest points between two islands and bridge with the lowest number of drones. We use alpha shape to determine the boundaries of each island. The alpha-shape algorithm attempts to define the shape of a finite set of points in the plane. Formally, the boundary is a graph $G_i(\mathcal{V}_i, E_i)$ (Figure 4), where $\mathcal{V}_i$ is the list of devices on the boundary of $\mathcal{N}_i$ and E_i is the set of edges ($|E_i| = e$).

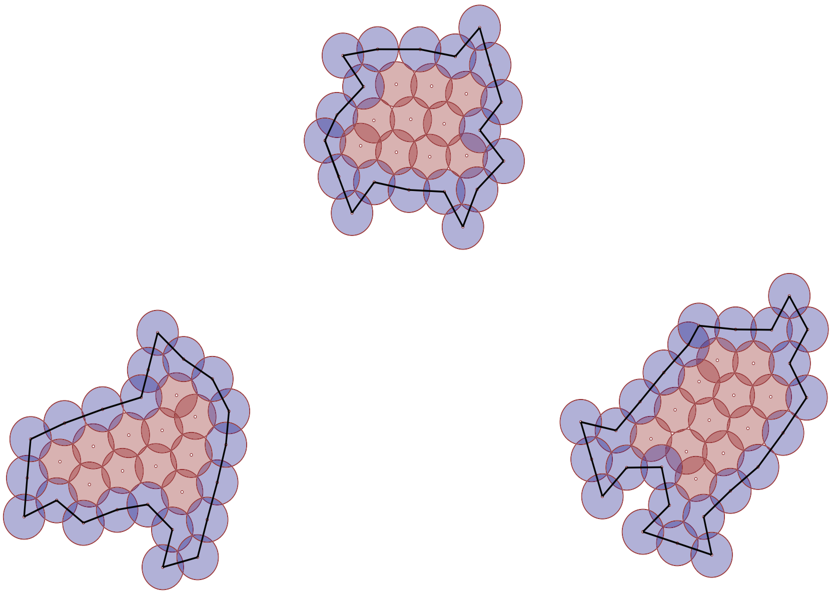


Figure 4. Example of the boundary obtained by alpha shape.

We denote $\alpha-$shape$(\mathcal{N}_i)$ the list of edges of an island $\mathcal{N}_i$ (Figure 4).

$$E_i = \alpha-\text{shape}(\mathcal{N}_i) = \{(D_u, D_v)|D_u \in \mathcal{N}_i, D_v \in \mathcal{N}_i\}$$

where (D_u, D_v) is an edge between two vertices D_u and D_v. The set of devices on the boundaries are denoted $\mathcal{V}_i$ as follows:

$$\mathcal{V}_i = \{D_q \in E_i | q \in [1, b] \wedge b \leq |\mathcal{N}_i|\}$$

We denote $\mathcal{N}_i^v$ the list of sub chains of E_i.

An island $\mathcal{N}_i$ may have one or more neighboring islands $\mathcal{N}_j$, called adjacent islands. Assume $\mathcal{N}_i$ has β adjacent islands. We denote $A(\mathcal{V}_i)$ the set of devices on the edges of the neighboring islands. In other words, it is a set of β adjacent boundary sets $\mathcal{V}_j$ to $\mathcal{V}_i$.

$$A(\mathcal{V}_i) = \cup_{j=1}^{\beta}\{\mathcal{V}_j\} = \{D_q \in V_j | V_j \neq V_i \wedge j \leq (t-1)\}$$

The bridging process is initiated by the central station (CS). Drones are sent to discover the isolated islands using discovery algorithms such as the Fast Neighbor Discovery Algorithm (FNDA) [28]. Drones calculate the boundary set for each neighborhood $\mathcal{N}_i$ using alpha-shape algorithm.

After determining the boundaries $\mathcal{E}_i$ of each island $\mathcal{N}_i$, this bridging process starts finding the minimal distance between any two adjacent boundary sets. Computing the minimum distance between two islands $\mathcal{N}_i$ and $\mathcal{N}_j$ (Figure 5) is a well-studied problem in computational geometry [29].

Let $d(G_i, G_j)$ be this distance:

$$d(G_i, G_j) = min\{d(\mathcal{N}_i^v, \mathcal{N}_j^v)\}$$

where $\mathcal{N}_i^v, \mathcal{N}_j^v$ are linearly separable sub chains of $\mathcal{N}_i$ and $\mathcal{N}_j$ respectively. The problem of determining $d(G_i, G_j)$ is decomposed into a collection of subproblems $d(\mathcal{N}_i^v, \mathcal{N}_j^v)$ [29].

When both G_i and G_j are convex, their separation can be determined in O(log n) time [30]. If exactly one of G_i or G_j is convex, then their separation can be computed in O(n) time [31]. When neither polygon is convex their separation can be found in O(n log(n)) time by traversing the contour between the polygons in their generalized Voronoi diagram [32]. According to [29], determining a closest pair among all pairs (D_i, D_j) where $D_i \in A(\mathcal{V}_j)$ and $D_j \in A(\mathcal{V}_i)$, runs in O(log(n)) time.

Let $\Gamma = \{\gamma_i(x_i, y_i, z_i, r_i)\}_{i=1}^{g}$ be a set of g drones (x_i, y_i, z_i) needed for bridging. The range of each drone is r_Γ. The central station (CS) calculates the required number of drones as well as the position of each drone on the shortest path between $\mathcal{N}_i$ and its adjacent island $\mathcal{N}_j$. The number of drones is equal to $d(G_i, G_j)/r_\Gamma$.

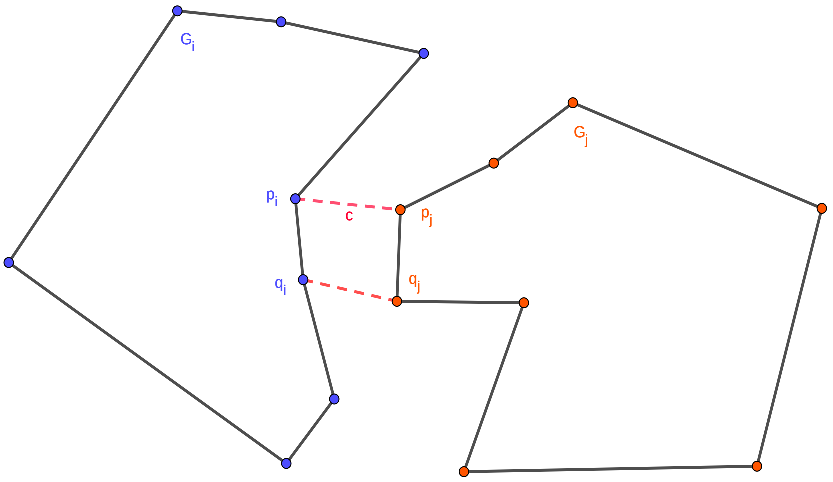


Figure 5. Distance between boundaries.

4.3.2. Distributed Bridging

A large enough number of drones are deployed in the area of interest. The steps are as follows:

- Drones start scanning the area of interest using coverage path planning algorithms to find any neighborhood or isolated devices.
- If a drone finds a neighborhood, the drone stack to the closest point between the neighborhood and the central station. The drone becomes part of this island and all the devices in the neighborhood knows about it. The chain of connected drones continues growing as straight line between the neighborhood and the central station. Drones either connect to a device at the edge of the island or to another drone. This connection is established using the τ formula.
- The process of discovery continues till all the links are established between the central station and the neighborhoods or isolated devices if any.

Both ways can be used to form a connected network between all the devices in the area of interest. However, the distributed approach requires greater deployed number of drones. We opt to use the centralized approach.

5. Validation, Simulation and Comparison with Existing Algorithms

We consider terrain where nodes can move without blockage in their ways and have enough energy to reach the final destination. Our goal is to understand how the coverage maximization algorithm behaves. We run different simulations to measure the performance of MCCM algorithm. The number of devices in each scenario varied between 25 and 150 devices ($N = \{25, 50, 75, 100, 125, 150\}$). They were distributed randomly in the area of interest. We tested the algorithm using homogeneous and heterogeneous distributions. We mean by homogeneous, the presence of sensors with the same characteristics and technical properties. In heterogeneous distributions, sensors are different in sensing ranges and characteristics. We choose the sensing range of the devices in the homogeneous distributions to be 5 m. It varies between 3 m and 8 m for heterogeneous distributions. For a better study, we followed the statistical approach by taking a sample of 10 runs for each deployment. We chose the area of interest to be a rectangular area of

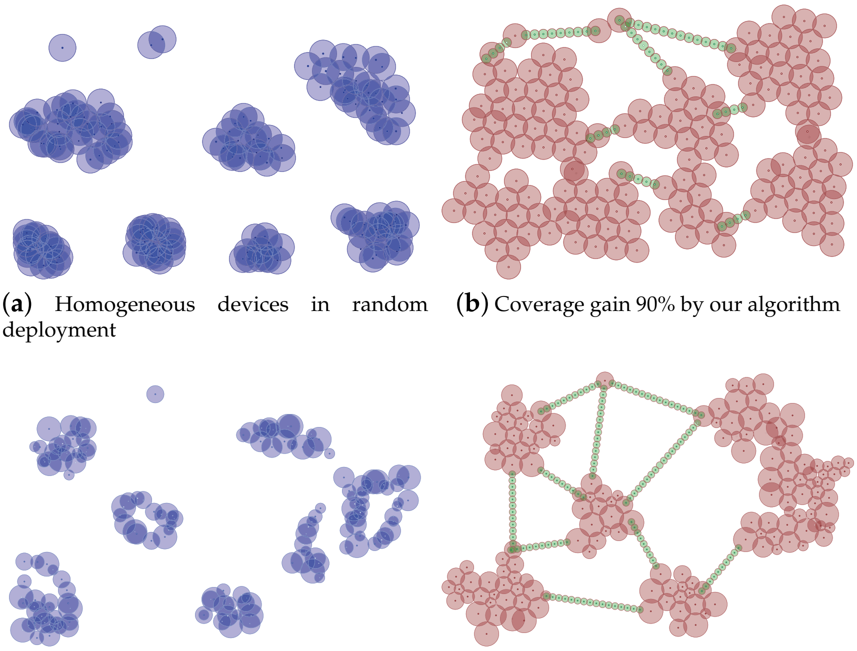
250 m × 250 m. The simulation parameters are presented in Table 1. The device characteristics are picked similarly to [33].

Table 1. Simulation parameters.

Parameter	Values
Area size	250 m × 250 m
Number of nodes	25, 50, 75, 100, 125, 150
Sensing range (meter)	3, 5, 8
Battery voltage	11.1 V
Horizontal Movement H_e	2.8 J/m
Energy Capacity C	4900 mAh
Battery Energy	195,804 J
Speed	2 m/s

Figure 6 shows the simulation results for 150 devices in both heterogeneous and homogeneous environments. The blue colored figures represent the deployment before applying our algorithm, the red colored figures represent the result after applying our algorithm and the green circles represent drones.

Figure 6a shows a random deployment of 150 devices in a homogeneous environment and Figure 6b shows the result after applying our algorithm. The coverage gain is 90%. The average device displacement is 18.21 m. Each device D_i has at least three devices directly connected to it. Figure 6c shows a random deployment of 150 devices in a heterogeneous environment, and Figure 6d shows the result after applying our algorithm. The coverage gain reached 87%. The average device displacement is 22.13 m. Each device D_i has at least four devices directly connected to it. In the next section we present a comparative study with other related works.



(**a**) Homogeneous devices in random deployment

(**b**) Coverage gain 90% by our algorithm

(**c**) Heterogeneous devices in random deployment

(**d**) Coverage gain of 87% by our algorithm

Figure 6. Coverage gain by MCCM: the blue (resp. red circles) represent the devices before (resp. after) running our algorithm; The green nodes are the drones that bridge the isolated islands.

5.1. Comparative Study

In this section, we present the key features of $MCCM$ and existing algorithms in the state of the art. We discuss the coverage gain, the displacement of nodes as well as the energy gain by the algorithms. Table 2 lists the following characteristics: The algorithm's architecture (Centralized or Distributed), the type of distribution (Homogeneous or Heterogeneous), the resource usage (low, high, medium) and the time complexity.

The time complexity of $MCCM$ is O(nlog(n)) similarly to $CMHN$ and $ABSDC$, where n is the number of devices. Other algorithms (such as $OPODCC$ and $HMRC$) require high computational resources. They are either centralized or suitable for homogeneous environments only. For instance, $DCHGA$ uses genetic code approach to calculate the final positions of the devices within the network. It requires high resources since the required memory allocation is $n \times p \times g$ where n is the number of devices, p is the population size, and g is the number of generations.

Table 2. Comparative study between our algorithms and existing related works.

Name	Architecture	Distribution	Resource	Complexity
CMHN [27]	Centralized	Heterogeneous	Low	$O(nlog(n))$
MCCM	Distributed	Heterogeneous	Low	$O(nlog(n))$
DCTC$_k$ [16]	Distributed	Heterogeneous	Low	$O(p \times q)$
DCHGA [20]	Centralized	Heterogeneous	High	$O(p \times g \times n)$
HMRC [21]	Centralized	Homogeneous	Medium	$O(n^2)$
AAEECC + FTCM [22]	Distributed	Homogeneous	Medium	$O(n^2)$
VCACWSN [23]	Centralized	Homogeneous	Medium	$O(n^2)$
ABSDC [24]	Centralized	Homogeneous	Low	$O(nlog(n))$
OPODCC [26]	Centralized	Homogeneous	High	$O(n^3)$

In the next section, we rely on the following indicators to measure the performance of our algorithm: (1) the average number of movements per device $mov(\Omega)$ related to distance, (2) the coverage gain $A_{gain}(\Omega)$, which is the probability that each point in the area is covered and (3) the energy depleted by the moving devices denoted E.

5.1.1. Coverage Gain and Displacement

Let D_i^f be the final position of a device D_i after applying the algorithm. To measure the average number of movements denoted $mov(\Omega)$, in the network Ω, we use Equation (2).

$$mov(\Omega) = \frac{1}{n}\sum_{i=1}^{n} dist(D_i, D_i^f) \tag{2}$$

As for the coverage gain, we adopt the evaluation method of Abo-Zahhad et al. in [10]. We treat the Area of Interest (AoI) as a matrix C of dimension $w \times h$. $C(i,j)$ presents the cell at position (i,j) in the matrix C where $i \leq w$ is a row in C, and $j \leq h$ is a column C. To exemplify, Figure 7 shows the distribution of two devices in the matrix D_A (the red circle) and D_B (the black circle). The point D is uncovered, which means there is no device covering this point. The point C is covered by two devices D_A and D_B. The point E is only covered by the device D_B.

For any point (x,y) in the matrix C, the probability $P(x,y,D_i)$ that the point is covered by a device D_i is given by Equation (3), where (x_i,y_i) are the coordinates of the device and r_i is its range.

$$P(x,y,D_i) = \begin{Bmatrix} 1, & \sqrt{(x-x_i)^2+(y-y_i)^2} < r_i \\ 0, & otherwise \end{Bmatrix} \tag{3}$$

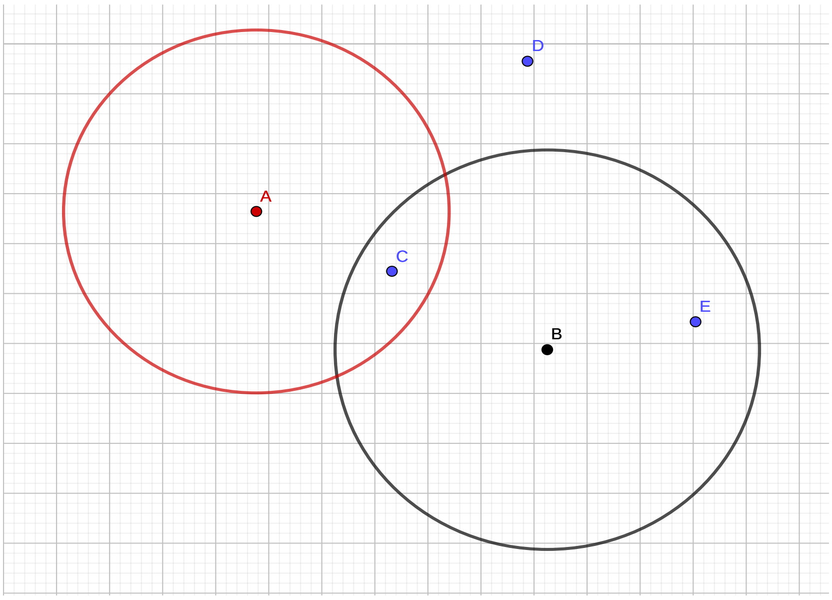


Figure 7. The matrix C of coverage.

For the whole network $\Omega = \{D_i\}_{i=1}^{n}$, we will use the above equation to find the probability that the point is covered. It is denoted $P(x, y, \Omega)$ presented in Equation (4).

$$P(x, y, \Omega) = 1 - \prod_{i=1}^{n}(1 - P(x, y, D_i)) \tag{4}$$

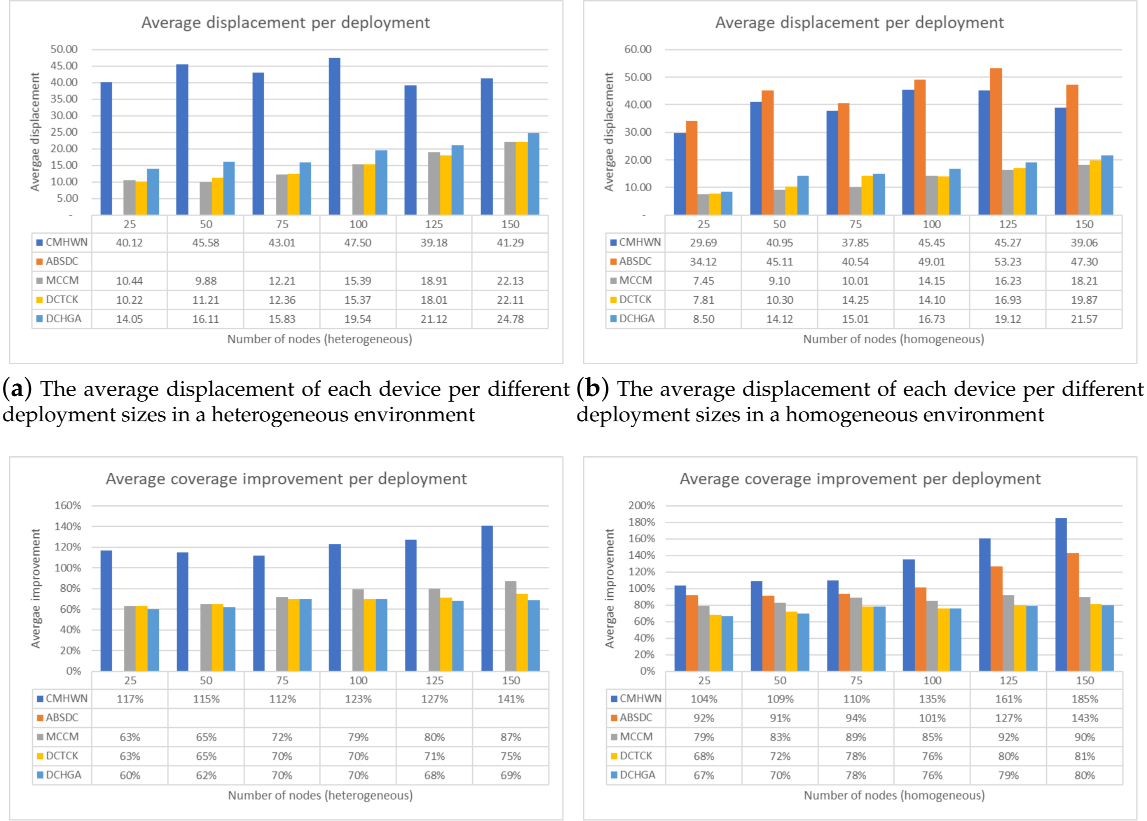
The coverage of the area of interest $(w \times h)$ in Equation (5), is the summation of all the aforementioned probabilities:

$$A_{cov}(\Omega) = \sum_{x=1}^{w}\sum_{y=1}^{h} P(x, y, \Omega) \tag{5}$$

We care about the coverage gained in the network by our algorithm. Therefore, we calculate the difference between the coverage obtained after applying our algorithm (denoted $A^{f}_{cov}(\Omega)$) and the coverage before applying the algorithm ($A_{cov}(\Omega)$) as shown in Equation (6).

$$A_{gain}(\Omega) = \frac{A^{f}_{cov}(\Omega) - A_{cov}(\Omega)}{A_{cov}(\Omega)} \tag{6}$$

We run 60 scenarios of random deployments with different number of devices. Figure 8a (resp. Figure 8b) clearly depicts that *MCCM* and *CMHN* are showing least nodes' displacement in heterogeneous (resp. homogeneous) environments. Figure 8c (resp. Figure 8d) represents the coverage gain in the heterogeneous (resp. homogeneous) environments. The centralized algorithms showed higher values in terms of coverage gain over distributed algorithms, as we expected. The reason is that by having the calculation done by a Central Station (CS), all the devices will move to their final optimized position. The centralized algorithm *CMHN* showed better results over ABSDC. This is due to the presence of the empty-arc rules that reduce the overlapping between devices and change it according to their ranges. In ABSDC, there are fixed angle and fixed distance between devices which makes the size of the overlapped area higher than *CMHN*. Both algorithms require low computational resources and they use simple mathematical formulas. *ABSDC* achieved 143% coverage gain while algorithm *CMHN* achieved 185% coverage gain. This means 42% more coverage gain over *ABSDC*.



	25	50	75	100	125	150
CMHWN	40.12	45.58	43.01	47.50	39.18	41.29
ABSDC						
MCCM	10.44	9.88	12.21	15.39	18.91	22.13
DCTCK	10.22	11.21	12.36	15.37	18.01	22.11
DCHGA	14.05	16.11	15.83	19.54	21.12	24.78

(**a**) The average displacement of each device per different deployment sizes in a heterogeneous environment

	25	50	75	100	125	150
CMHWN	29.69	40.95	37.85	45.45	45.27	39.06
ABSDC	34.12	45.11	40.54	49.01	53.23	47.30
MCCM	7.45	9.10	10.01	14.15	16.23	18.21
DCTCK	7.81	10.30	14.25	14.10	16.93	19.87
DCHGA	8.50	14.12	15.01	16.73	19.12	21.57

(**b**) The average displacement of each device per different deployment sizes in a homogeneous environment

	25	50	75	100	125	150
CMHWN	117%	115%	112%	123%	127%	141%
ABSDC						
MCCM	63%	65%	72%	79%	80%	87%
DCTCK	63%	65%	70%	70%	71%	75%
DCHGA	60%	62%	70%	70%	68%	69%

(**c**) The average coverage improvement of the network in a heterogeneous environment

	25	50	75	100	125	150
CMHWN	104%	109%	110%	135%	161%	185%
ABSDC	92%	91%	94%	101%	127%	143%
MCCM	79%	83%	89%	85%	92%	90%
DCTCK	68%	72%	78%	76%	80%	81%
DCHGA	67%	70%	78%	76%	79%	80%

(**d**) The average coverage improvement of the network in a homogeneous environment

Figure 8. Validation results of our algorithms against some existing algorithms.

MCCM showed better results compared to the distributed algorithms $DCTC_k$ and *DCHGA*. $DCTC_k$ achieved 75% coverage gain while *MCCM* achieved 87% coverage gain. This means that our algorithm increases the coverage gain by 12% in heterogeneous environment. As for the homogeneous environment scenarios, our algorithm achieves 9% gain over $DCTC_k$. The latter reached a gain of 81% when *MCCM* attained 90%. As a result, our algorithm performed better coverage gain in homogeneous environment. Similar results are obtained in the heterogeneous environment.

As for the displacement, Figure 8b shows the displacement made by each algorithm to maximize the network coverage. *CMHN* has better results than the other centralized algorithms. *MCCM* achieves lower average node's displacement and better coverage over the other distributed algorithms ($DCTC_k$ and *DCHGA*).

Table 3 summarizes the average overall performance of *MCCM* compared to the distributed algorithms $DCTC_k$ and *DCHGA* in homogeneous and heterogeneous environments. It increases the coverage gain up to 19% and reduces the nodes displacement by 35.55% compared to the algorithms in the table. In the next section, the results obtained in Figure 8 are interpreted from the energy saving perspective.

Table 3. MCCM performance over existing related works.

HOMOGENEOUS										
Number of nodes	**50**		**75**		**100**		**125**		**150**	
MCCM versus	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHG**	**DCTC$_k$**	**DCHGA**
Increase in coverage gain in MCCM	15%	19%	14%	14%	12%	12%	15%	16%	11%	13%
Reduction in Displacement	11.65%	35.55%	29.75%	33.31%	0.35%	15.42%	4.13%	15.12%	8.35%	15.58%
HETEROGENEOUS										
Number of nodes	**50**		**75**		**100**		**125**		**150**	
MCCM versus	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHGA**	**DCTC$_k$**	**DCHGA**
Increase in coverage gain in MCCM	0%	5%	3%	3%	13%	13%	13%	18%	16%	26%
Reduction in Displacement	11.86%	38.67%	1.21%	22.87%	0.13%	21.24%	5%	10.46%	0.09%	10.69%

5.1.2. Energy Consumption

As the energy consumption is proportional to the displacement, we adopt the formula in Equation (7). It measures the energy depleted by a moving device, where H_e is the energy required to move one meter (J/m) and d is the displacement (m).

$$E = H_e * d \tag{7}$$

$CMHN$ and $ABSDC$ are both centralized algorithms. However, $ABSDC$ operates only in homogeneous environments. For fair comparison we run both in scenarios composed of 150 homogeneous devices. The results were as follows: $ABSDC$ required an average displacement of 47.3 m to maximize the coverage. Consequently, the energy consumed by the devices in $ABSDC$ was: $E_{ABSDC} = 2.8 \times 47.3 = 132.44$ J. $CMHN$ required 39.06 m displacement to maximize the coverage. The energy consumed is $E_{CMHN} = 2.8 \times 39.06 = 109.368$ J. Hence, the centralized version of MCCM algorithm reduced the displacement and improved the coverage by saving 17.4% of energy in the network:

$$\frac{E_{(CMHN)} - E_{(ABSDC)}}{E_{(ABSDC)}} = \frac{109.368 - 132.44}{132.44} = -0.174$$

$MCCM$ and $DCTC_k$ are distributed algorithms and can operate on different network environments (homogeneous and heterogeneous). $DCTC_k$ requires higher computational resources than $MCCM$ (as shown in Table 2). The comparison of the two algorithms in a homogeneous environment of 150 devices depicted the following:

$DCTC_k$ required an average of 19.87 m displacement to maximize the coverage. The consumed energy is $E_{DCTC_k} = 2.8 \times 19.87 = 55.636$ J. Whereas the $MCCM$ required 18.21 m displacement to maximize the coverage, the energy is $E_{MCCM} = 2.8 \times 18.21 = 50.988$ J. We can notice that $MCCM$ algorithm reduced the nodes displacement by saving 8% of energy over $DCTC_k$ algorithm:

$$\frac{E_{(MCCM)} - E_{(DCTC_k)}}{E_{(DCTC_k)}} = \frac{50.988 - 55.636}{55.636} = -0.08$$

$MCCM$ behaves similarly in other scenarios. Displacements results in Table 3 showed an energy saving up to 35.5% in $MCCM$ compared to $DCTC_k$ and $DCHGA$.

6. Conclusions and Future Works

The collaboration between UAV and WSN into the Internet of things (IoT), is gaining a lot of attention, as a solution to detect emergency events especially in rural and harsh environments. It guarantees an autonomous network healing and quick actions during long-distance missions. Nonetheless, it contributes to a rural transformation by motivating people to live in rural area and facilitating their daily activities. The main issue in harsh and disaster environments is to maintain connectivity between all the objects (devices) and to maximize the coverage between them. Reducing the mobility of devices is also another issue. Many constraints have been considered and handled in this article such as the distributed environment, the heterogeneity and mobility of devices. The main goal is to keep the movement of devices as minimal as possible to save energy. Another goal is to reduce the overlapping areas covered by different devices and keep it to a small margin to maximize the coverage while maintaining connectivity. We proposed $MCCM$ a distributed algorithm for coverage maximization, in which each mobile device re-organizes itself to maximize the coverage in its neighborhood. Since the random deployment and the movement of devices, might form isolated groups or "islands", we bridge these groups using alpha-shape technique. Simulation results showed an increase in coverage gain and reduction in nodes' displacement and energy consumption, compared to other algorithms in both homogeneous and heterogeneous environments.

We noticed the following about $MCCM$: It may happens that a device D_i trying to maximize the coverage in an island $\mathcal{N}_i$, might meet another device D_j from another island $\mathcal{N}_j$ on his way. The two devices might overlap at their final positions and might not be located in their optimal position with respect to each other (as they belong to two different islands). The reason is that the maximization process applies only inside an island and not when two islands meet. This behavior can be beneficial in scenarios where overlapping gets less priority, while the presence of redundancy enhances network reliability. From the coverage optimization perspective, it is recommended to have the least overlapping area among devices while maximizing coverage. In the future works, we intend to tackle this matter by adapting the m-ratio and τ_{ij} constraints when combining islands together. We also aim to do more experimentation with industrial fields and concrete case studies.

Author Contributions: Conceptualization, H.M., M.B.H.; Funding acquisition, G.F.; Investigation, M.B.H., A.G., G.F.; Supervision, A.G., H.M.; Validation, M.B.H.; Writing-original draft, M.B.H.; Writing-review and editing, A.G., All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers, for their valuable comments which improved the clarity and quality of the paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

List of Acronyms

IoT	internet of Things
WSN	Wireless Sensor Network
MCCM	Coverage Maximization and Maintaining Connectivity
ITU	International Telecommunication Union
IERC	European Research Cluster on the Internet of Things
AoI	Area of Interest
AoT	Array of Things
CTC_k	connected target k-coverage
$DCTC_k$	distributed connected target k-coverage
DCHGA	Dynamic Clustering of Heterogeneous WSNs using Genetic Algorithm
CS	Central Station
SN	Sink Node
HMRC	Harmony Memory Considering Rate
AAEECC	Asynchronous Algorithm for providing Energy Efficient Coverage and Connectivity
FTCM	Fault Tolerant Connectivity Maintenance
ABSDC	Angles-based Sensors Deployment Algorithm to Develop the Coverages in WSN
GSO	Glowworm Swarm Optimization
OPDCC	Optimal Deployment for k-Coverage in m-Connected Wireless Networks
CMHN	Coverage Maximization for Heterogeneous Wireless Network
VCACWSN	Voronoi coverage Algorithm based on Connectivity for Wireless Sensor Networks
UAV	Unmanned Aerial Vehicle

List of Symbols

D_i	A device i in the network
r_i	Range of device D_i
$m(i,j)$	$m-$ratio between devices D_i and D_j
d_{ij}	Distance between devices D_i and D_j, denoted $dist\left(D_i, D_j\right)$
τ_{ij}	Minimum required overlapping between devices D_i and D_j
Ω	Set of devices in the network
$\mathcal{N}_i$	An island i in the network

$\mathcal{M}$	The mean location of the devices in an island
ζ	Set of devices located between Device D_i and $\mathcal{M}$
$G_i(\mathcal{V}_i, E_i)$	A graph of $\mathcal{V}_i$ vertices and E_i edges
$A(\mathcal{V}_i)$	A set of devices on the edges of the neighboring islands
$d(G_i, G_j)$	Minimum distance between two graphs G_i and G_j
Γ	Set of drones
D_i^f	Final position of Device D_i
$mov(\Omega)$	Average number of movements in the network
$P(x, y, D_i)$	Probability that point (x,y) is covered by device D_i
$A_{cov}(\Omega)$	Coverage obtained before applying our algorithm
$A_{cov}^f(\Omega)$	Coverage obtained after applying our algorithm
$A_{gain}(\Omega)$	Coverage gained in the network
E	Energy depleted by a moving device
H_e	The energy required to move one meter

References

1. Bica, I.; Chifor, B.C.; Arseni, S.; Matei, I. Multi-Layer IoT Security Framework for Ambient Intelligence Environments. *Sensors* **2019**, *19*, 4038. [CrossRef] [PubMed]
2. Jones, R. *The International Telecommunication Union*; IEEE: New York, NY, USA, 2019; p. 1248. [CrossRef]
3. Hennebert, C.; Baldini, G.; Peirce, T.; Botterman, M.; Talacchini, M.; Guimaraes Pereira, A.; Handte, M.; Rotondi, D.; Pohls, H. IoT Governance, Privacy and Security Issues. Available online: https://bhanu18.wordpress.com/2015/07/01/iot-governance-privacy-and-security-issues/(accessed on 21 January 2020).
4. Gassmann, O.; Böhm, J.; Palmié, M. *Smart Cities*; Emerald Publishing Ltd.: Bingley, UK, 2019; pp. 25–66. [CrossRef]
5. Vepachedu, S. Privacy and Internet of Things, Array of Things & Artificial Intelligence. Available online: https://www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf (accessed on 21 January 2020).
6. Aly, H. Royal Dream: City Branding and Saudi Arabia's NEOM. *Middle East Top. Argum.* **2019**, *12*. [CrossRef]
7. Farag, A. The story of NEOM City: Opportunities and challenges. In *New Cities and Community Extensions in Egypt and the Middle East*; Springer: Berlin, Germany, 2019; pp. 35–49. [CrossRef]
8. Devaul, W.; Devaul, R.; Teller, E.; Biffle, C.L.; Weaver, J. Ballon Network with Free-Space Optical Communication between Super-Node Balloons and RF Communication between Super-Node and Sub-Node Balloons. U.S. Patent No. 8,718,477, 24 March 2014.
9. Cuka, M.; Ozera, K.; Obukata, R.; Elmazi, D.; Oda, T.; Barolli, L. *Implementation of a GA-based Simulation System for Placement of IoT Devices: Evaluation for a WSAN Scenario*; Springer: Cham, Switzerland, 2018, pp. 34–42. [CrossRef]
10. Abo-Zahhad, M.; Ahmed, S.M.; Sabor, N.; Sasaki, S. Coverage maximization in mobile Wireless Sensor Networks utilizing immune node deployment algorithm. In Proceedings of the 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), Toronto, ON, Canada, 4–7 May 2014, pp. 1–6. [CrossRef]
11. Jaradat, Y.; Masoud, M.; Jannoud, I.; Zaidan, D. The Impact of Nodes Distribution on Energy Consumption in WSN. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 590–595. [CrossRef]
12. Tripathi, A.; Gupta, H.P.; Dutta, T.; Mishra, R.; Shukla, K.K.; Jit, S. Coverage and Connectivity in WSNs: A Survey, Research Issues and Challenges. *IEEE Access* **2018**, *6*, 26971–26992. [CrossRef]
13. Wang, X.; Xing, G.; Zhang, Y.; Lu, C.; Pless, R.; Gill, C. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2003; Volume 1, pp. 28–39. [CrossRef]
14. Yeasmin, N. k-Coverage Problems and Solutions in Wireless Sensor Networks: A Survey. *Int. J. Comput. Appl.* **2014**, *100*, 1–6. [CrossRef]
15. Zhu, C.; Zheng, C.; Shu, L.; Han, G. A survey on coverage and connectivity issues in wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 619–632. [CrossRef]

16. Yu, J.; Chen, Y.; Huang, B. On Connected Target k-Coverage in Heterogeneous Wireless Sensor Networks. In Proceedings of the 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI), Beijing, China, 17–18 October 2014; pp. 262–265. [CrossRef]
17. Gupta, N.; Kumar, N.; Jain, S. Coverage problem in wireless sensor networks: A survey. In Proceedings of the 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, India, 3–5 October 2016; pp. 1742–1749. [CrossRef]
18. Ahmed, A.; Awais, M.; Akram, T.; Kulac, S.; Alhussein, M.; Aurangzeb, K. Joint Placement and Device Association of UAV Base Stations in IoT Networks. *Sensors* **2019**, *19*, 2157. [CrossRef] [PubMed]
19. More, A.; Raisinghani, V. A Survey on Energy Efficient Coverage Protocols in Wireless Sensor Networks. *J. King Saud Univ. Comput. Inf. Sci.* **2016**, *29*. [CrossRef]
20. Elhoseny, M.; Hassanien, A.E. Expand Mobile WSN Coverage in Harsh Environments. In *Dynamic Wireless Sensor Networks*; Springer: Cham, Switzerland, 2019; pp. 29–52. [CrossRef]
21. Sharma, D.; Gupta, V. Improving coverage and connectivity using harmony search algorithm in wireless sensor network. In Proceedings of the 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Dehradun, India, 17–18 November 2017; pp. 1–7. [CrossRef]
22. Ansari, N.A.; Deshpande, U.A.; Mohammad, S.P. An asynchronous algorithm for providing energy efficient coverage and connectivity in Wireless Sensor Networks. In Proceedings of the 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bangalore, India, 6–9 November 2016; pp. 1–6. [CrossRef]
23. Qihua, W.; Ge, G.; Lijie, C.; Xufeng, X. Voronoi coverage algorithm based on connectivity for wireless sensor networks. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 7833–7837. [CrossRef]
24. Hasson, S.T.; Finjan, A.A.R. A suggested angles-based sensors deployment algorithm to develop the coverages in WSN. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 547–552. [CrossRef]
25. Kadavy, T.; Pluhacek, M.; Viktorin, A.; Senkerik, R. *Multi-Swarm Optimization Algorithm Based on Firefly and Particle Swarm Optimization Techniques*; Springer: Cham, Switzerland, 2018; pp. 405–416. [CrossRef]
26. Gupta, H.P.; Tyagi, P.K.; Singh, M.P. Regular Node Deployment for k -Coverage in m -Connected Wireless Networks. *IEEE Sens. J.* **2015**, *15*, 7126–7134. [CrossRef]
27. Mcheick, H.; Hatoum, M.S.B.; Ghaddar, A. CMHN: Coverage Maximization of Heterogeneous Wireless Network. In Proceedings of the 16th ACS/IEEE International Conference on Computer Systems and Applications AICCSA'19, Abu Dhabi, UAE, 3–7 November 2019.
28. Wei, L.; Sun, W.; Chen, H.; Yuan, P.; Yin, F.; Luo, Q.; Chen, Y.; Chen, L. A Fast Neighbor Discovery Algorithm in WSNs. *Sensors* **2018**, *18*, 3319. [CrossRef] [PubMed]
29. Amato, N.M. *Determining the Seperation of Simple Polygons*; World Scientific Publishing Company: Singapore, 1994; pp. 457–474.
30. Dobkin, D.P.; Kirkpatrick, D.G. Determining the separation of preprocessed polyhedra a unied approach. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 1990; Volume 443, pp. 400–413.
31. Chin, F.; Sampson, J.; Wang, C.A. A unifying approach for a class of problems in the computational geometry of polygons. *Vis. Comput.* **1985**, *1*, 124–132. [CrossRef]
32. Yap, C.K. An O(n log n) algorithm for the Voronoi diagram of a set of simple curve segments. *Discret. Comput. Geom.* **1987**, *2*, 365–393. [CrossRef]
33. Aleksandrov, D.; Penkov, I. Energy Consumption of Mini UAV Helicopters with Different Number of Rotors. In Proceedings of the 11th International Symposium Topical Problems in the Field of Electrical and Power Engineering, Parnu, Estonia, 16–21 January 2012.