

Article

# Energy-Efficient Deadline-Aware Data-Gathering Scheme Using Multiple Mobile Data Collectors

Rumpa Dasgupta and Seokhoon Yoon \*

Department of Electrical and Computer Engineering, University of Ulsan, Ulsan 680-749, Korea; rumpa.dasgupta3949@gmail.com

\* Correspondence: seokhoonyoon@ulsan.ac.kr; Tel.: +82-52-259-1403

Academic Editor: Yuh-Shyan Chen

Received: 21 January 2017; Accepted: 26 March 2017; Published: 1 April 2017

**Abstract:** In wireless sensor networks, the data collected by sensors are usually forwarded to the sink through multi-hop forwarding. However, multi-hop forwarding can be inefficient due to the energy hole problem and high communications overhead. Moreover, when the monitored area is large and the number of sensors is small, sensors cannot send the data via multi-hop forwarding due to the lack of network connectivity. In order to address those problems of multi-hop forwarding, in this paper, we consider a data collection scheme that uses mobile data collectors (MDCs), which visit sensors and collect data from them. Due to the recent breakthroughs in wireless power transfer technology, MDCs can also be used to recharge the sensors to keep them from draining their energy. In MDC-based data-gathering schemes, a big challenge is how to find the MDCs' traveling paths in a balanced way, such that their energy consumption is minimized and the packet-delay constraint is satisfied. Therefore, in this paper, we aim at finding the MDCs' paths, taking energy efficiency and delay constraints into account. We first define an optimization problem, named the delay-constrained energy minimization (DCEM) problem, to find the paths for MDCs. An integer linear programming problem is formulated to find the optimal solution. We also propose a two-phase path-selection algorithm to efficiently solve the DCEM problem. Simulations are performed to compare the performance of the proposed algorithms with two heuristics algorithms for the vehicle routing problem under various scenarios. The simulation results show that the proposed algorithms can outperform existing algorithms in terms of energy efficiency and packet delay.

**Keywords:** multiple mobile data collectors; data-gathering; energy constraint; delay constraint; wireless power transfer

---

## 1. Introduction

Wireless sensor networks (WSNs) comprise sensor nodes that are deployed to sense physical or environmental conditions for a diverse range of applications, including environmental monitoring, industrial sensing, battlefield surveillance, critical response, and so forth [1]. Traditionally, the sensing data are forwarded to a sink via multiple hops. However, it is known that multi-hop forwarding leads to the energy-hole problem, where sensors near the sink consume much more energy, since they forward more data packets than sensors farther from the sink [2]. Once these sensors run out of battery power, other sensors can no longer reach the sink, and the network becomes disconnected and decreases overall network lifetime. Moreover, multi-hop forwarding may incur high communications overhead due to frequent information exchange among sensors, and data transmission is affected by interference and collisions [3]. As a result, packet loss increases along with increases in the number of hops.

To avoid these problems, the mobile data-gathering scheme has been explored in recent years [4–16]. In such schemes, mobile data collectors (MDCs) (e.g., mobile sensors, robotic vehicles)

roam in the network, collect data from sensors via short-range communications and bring the data to the sink. By using MDCs for data collection, reliability in data delivery can be achieved, and energy consumption in transmitting the data can be reduced significantly. However, MDC-based data-gathering schemes can only slow down energy depletion of sensors, rather than prevent it.

In this paper, we consider using wireless power transfer technology to keep the WSN operational without the risk of energy depletion. The wireless power transfer technology is mainly classified into two categories: far-field [17–22] and near-field [23–25]. In the far-field or radiative technique, power is transferred by beams of electromagnetic radiation. Recently, this technique has been used in stationary wireless sensors networks in order to transfer power to sensor nodes [19–22]. In the near-field technique, power is transferred by magnetic fields or by electric fields. For example, in [24], Kurs et al. showed that wireless power transfer is feasible by using a technique called magnetic resonant coupling, i.e., electric power can be transferred from one storage device to another device without wires. The near-field technique has high power transfer efficiency at a shorter distance. In this work, the near-field technique is considered, since we can exploit the mobility of MDCs, which can approach the sensors and achieve high energy transfer efficiency, i.e., MDCs can recharge sensors to a certain energy level when they visit them periodically, such that the WSN remains operational without the risk of energy depletion.

In MDC-based data-gathering schemes, minimizing movement energy is critical, since MDCs consume most of their energy for movements. Balancing energy consumption is also important in the case of multiple MDCs. For example, some MDCs may travel a longer distance, compared to others, which results in unbalanced energy consumption and being unable to return to the sink. Meanwhile, guaranteeing packet delivery delays is also crucial for achieving acceptable quality of service in some applications. For example, the detection of critical events (e.g., intruder detection, forest fire detection [26]) should be reported to the sink within a predetermined time interval.

There have been many studies on data gathering in WSNs using MDCs [4–16]. In addition, some studies focused on energy replenishment in WSNs using MDCs [25,27–29]. However, most of the previous studies did not consider important factors, such as the MDCs' energy consumption for movement, their limited battery capacity, their travel time and the recharging time of the sensors. Moreover, none of them considered minimizing total energy, in cases of multiple MDCs, while meeting packet delivery delay requirements.

Therefore, in this work, we focus on finding the MDCs' paths that minimize total energy consumption while satisfying delay and energy constraints. We first formulate an integer linear programming (ILP) optimization problem. The objective of ILP formulation is to find the paths for MDCs that minimize the total energy consumption while guaranteeing that packets should reach the sink within a given deadline, and in each period, the energy consumption of each MDC (including energy for wireless charging, moving and data reception) cannot exceed its energy capacity limit; and every sensor should have residual energy above a minimum threshold at all times.

We note that finding the optimal paths for MDCs using the ILP is computationally expensive, particularly when the number of MDCs or the number of sensors is large. Therefore, in this paper, we propose efficient path selection algorithms that have two phases: clustering and group membership adjustment. We propose two clustering algorithms, angle-based clustering (ABC) and angle-based clustering with path-length ratio (ABC-PR), for partitioning the sensors into multiple groups. Then, two group-membership adjustment algorithms, nearest node assignment (NNA) and node assignment with maximal path-length decrease (MPD), are proposed to minimize the energy consumption of the MDCs and meet constraints, including the packet delay and residual energy of the sensors.

In the proposed clustering algorithms, nodes are partitioned into different groups in such a way that the travel distance of each MDC is close to that of the other MDCs. Then, in the group membership-adjustment algorithms, the group that has the maximum travel distance is selected. From this group, one node is reassigned to its clockwise adjacent group. Specifically, when finding a node for reassignment, NNA selects the node that is the nearest to its clockwise adjacent group, and MPD selects the node whose membership change reduces the path length the most.

Simulations and analysis are conducted to evaluate the performance of the proposed algorithms in various scenarios. The performance of the proposed algorithms is compared with two well-known heuristic algorithms for the vehicle routing problem (VRP) and VRP variants. Simulation results show that the proposed algorithms outperform existing approaches in terms of energy consumption and delay. Simulation results also indicate that MPD with ABC-PR shows the best performance, compared to others, and achieves a near-optimal solution.

The rest of the paper is organized as follows. In Section 2, we present an overview of existing studies related to our work. Section 3 introduces general assumptions and the network model considered in this study. In Section 4, the problem definition and problem formulation are presented. In Section 5, we describe the proposed two-phase path-selection algorithm. Section 6 presents the setup for simulation and performance analysis. Finally, we conclude the paper in Section 7.

## 2. Related Work

In this section, we present some existing studies that considered mobile data collectors for data collection and recharging sensor nodes in WSNs and then compare those studies with our work.

Many recent studies have been conducted for data collection in WSNs using a single MDC with different objectives [4–12]. For instance, three studies [6–8] focused on the minimization of packet delay with meeting a given deadline. Specifically, Xing et al. [6] proposed a rendezvous-based data collection scheme in which a group of nodes acts as rendezvous points to buffer and collect data received from source nodes and deliver them to a mobile base station within a given time period. In [7], Kansal et al. described a problem that assumes an MDC must traverse a given path within a predetermined time interval, with the objective of controlling the speed of the MDC in order to maximize data collection. A heuristic algorithm was proposed by Sugihara and Gupta [8], which considers both timing constraints and sensor locations to reduce packet delay in a data mule scheduling problem.

Somasundara et al. [9] and Gu et al. [10] studied the scheduling problem of MDC to avoid data loss at sensors due to buffer overflow. For instance, in [10], Gu et al. first partitioned the sensor nodes into multiple groups, depending on their locations and data-generation rates. Then, in each group, the MDC is scheduled to visit stationary sensors at a sufficient frequency to minimize buffer overflow.

In addition, Wang et al. [11] focused on the problem of maximizing network lifetime. They considered a mobile sink for visiting some sensor nodes in the network and formulated a linear programming optimization problem to determine the optimal path for the mobile sink in order to maximize network lifetime.

However, those works use a single MDC to collect data from the entire area. The MDC will take a long time to finish one tour when the size of the area is large. As a result, packet delay increases. Thus, only one MDC may not be useful for delay-sensitive applications. To minimize packet delay, this work uses multiple MDCs to collect data from sensors.

There are a number of studies that use multiple MDCs for data collection in WSNs [13–16]. For example, Keung et al. [14] and Aslanyan et al. [15] considered multiple MDCs for data collection from source nodes. They assumed that MDCs have random mobility patterns. Because the MDCs have random mobility, those schemes cannot estimate packet delay or the travel time of MDCs. Jea et al. also proposed a multiple MDC-based data collection scheme [16]. The authors considered MDCs that follow parallel straight lines for their movement and that collect data from sensors via multiple hops. This scheme is suitable for uniformly-distributed sensor networks. However, that type of MDC movement is not practical, because different obstacles may block the paths of the MDCs. In addition, all of the listed studies considered MDCs only for data collection.

In contrast, our work considers multiple MDCs that not only collect data from the sensors, but also replenish the energy in the sensors to keep the WSN operational without the risk of energy depletion. Moreover, we focus on finding MDCs' paths that minimize total energy consumption, with delay and energy constraints for each MDC.

There have been some studies that use MDCs to recharge sensor nodes' batteries wirelessly via wireless power transfer [25,27]. In those works, the collected data are forwarded to the sink through multiple hops. For example, in [25], Xie et al. considered a mobile charging vehicle that periodically recharges the sensor nodes' batteries to keep the energy level of each sensor above a minimum threshold. Peng et al. [27] developed a proof-of-concept prototype, and experiments were performed on the prototype to estimate its energy replenishment performance in small-scale networks. Although sensors' energy was replenished in those works, a significant amount of sensor energy was still wasted due to the multi-hop data transmissions. In our work, MDCs are used for energy replenishment, as well as data collection when visiting sensors, bringing the collected data to the sink within a predetermined time interval.

There are few works that consider MDCs for both collection and energy replenishment in WSNs [28,29]. For example, Pan et al. [28] studied an energy replenishment and data collection problem considering wireless charging, flow routing and data transmission constraints to determine the specific sensors, which need to be recharged using a wireless charging vehicle, so that network lifetime is maximized. However, those works did not consider delay requirements, which is important for achieving an acceptable quality of service. In addition, travel time of an MDC, the time consumed for recharging sensors and the MDC's moving energy consumption were neglected in those works.

In contrast, we consider every possible energy consumption of the MDCs, which includes MDCs' movement energy, energy consumption for receiving data from sensors and recharging the sensors. Moreover, travel time and sojourn time of the MDCs at each sensor node are considered in our work.

There exist some similarities between our work and the vehicle routing problem (VRP) [30] with its variants. The VRP is defined as finding routes for vehicles to deliver goods between a depot and customers to minimize the vehicles' total travel cost. Among different variants of VRP, our work is related to VRP with time windows (VRPTW) [31]. In VRPTW, vehicles are scheduled to visit all customers within a pre-defined time window. On the other hand, in this study, we consider both energy and timing constraints for each MDC. We also compare the performance of our proposed algorithms with two heuristic algorithms for VRP.

### 3. Network Model

In this section, we present the network model of the proposed scheme in detail. Figure 1 illustrates the network model of our proposed scheme. The considered network consists of a number of stationary sensors and a number of mobile data collectors. Sensors are deployed randomly in a circular area around the sink, with radius  $R$ , and MDCs are used to collect data from sensors and to deliver the data to the sink. In this paper, we denote  $N$  and  $K$  as the total numbers of sensors and MDCs, respectively. It is assumed that the sensor positions are known a priori. Each sensor periodically generates data and saves them in its buffer until an MDC visits it and collects the data. Besides, we assume that the generated data over the WSN should periodically be delivered to the sink within time interval  $D$ . Thus, each MDC needs to finish its tour within the deadline,  $D$ . Sensors are provided with a limited battery and consume energy for operation. Thus, the WSN can only remain operational for a limited amount of time. To keep the sensors from completely draining their energy and to keep the WSN alive, MDCs periodically recharge the visited sensors to a certain energy level. Each sensor node (except the sink) is visited by only one MDC. Each MDC starts its journey at the sink, and its speed is  $v$ . When it arrives at a sensor node, say sensor  $i$ , it will spend time  $t_i$  to charge the sensor node's battery wirelessly via wireless power transfer [24] and collects data from this sensor. After  $t_i$ , the MDC leaves sensor  $i$  and travels to the next sensor on its path. Moreover, let  $Q$  denote the battery capacity of each MDC, which is fully charged at the beginning of the tour. When the MDC finishes visiting sensors, it brings the collected data back to the sink. After that, the MDC battery will be replaced [25] to get it ready for the next tour with a fully-charged battery.

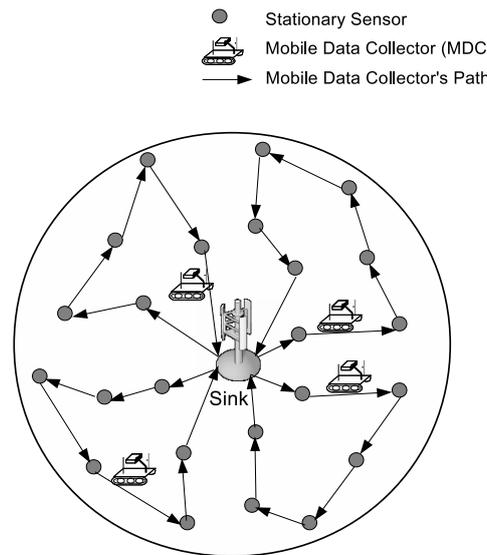


Figure 1. Network Model.

In this work, the energy model is the one referred to by Liu et al. [32]. Let  $e_S$  denote the energy consumption for sensing one bit of data. Furthermore,  $e_T$  and  $e_R$  denote the energy usage per bit of transmission and reception electronics, respectively. The energy consumed by a transmission amplifier for transmitting one bit is  $\rho l^\gamma$ , where  $\gamma$  is the signal decline factor, and  $l$  is the distance between transmitting and receiving nodes. Transmission energy  $e_{tx}$  and reception energy  $e_{rx}$  for one bit of data can be calculated as:

$$\begin{aligned} e_{tx} &= e_T + \rho l^\gamma \\ e_{rx} &= e_R \end{aligned} \quad (1)$$

#### 4. Problem Definition and Formulation

From the given description of the considered network and corresponding model, in this section, we define and formulate our problem.

##### 4.1. Problem Definition

This work aims at finding paths for MDCs to minimize the total energy consumption while satisfying delay and energy constraints. The problem of finding paths can be formally defined as follows.

**Definition 1.** *Delay-constrained energy minimization (DCEM): Find the MDCs' paths that minimize the total energy consumption of all MDCs with the delay constraints that packets should reach the sink within a given deadline,  $D$ , the energy consumption of each MDC cannot exceed its energy capacity,  $Q$ , in each period, and every sensor should have at least the minimum residual energy,  $e_{min}$ , all of the time.*

Note that DCEM is a more generalized problem than VRP [30]. Since the VRP is NP-hard [30], DCEM is also NP-hard.

##### 4.2. Problem Formulation

Now, we present an integer linear programming (ILP) formulation for DCEM. Definitions for symbols used in the ILP formulation are presented in Table 1.

**Table 1.** List of symbols. MDC, mobile data collector.

Symbol	Definition
$N$	Number of nodes
$K$	Number of MDCs
$V$	Set of nodes, $V = \{1, \dots, N\}$
$M$	Set of MDCs, $M = \{1, \dots, K\}$
$t_i$	Sojourn time of MDC at sensor $i$ , (s)
$E_i^{ch}$	Energy transfer rate from MDC to sensor $i$ , (J/s)
$E_Q^k$	Energy consumption of MDC $k$ to charge visited sensors, (J)
$E_m$	Energy consumption of MDC to travel one meter, (J/m)
$c_{ij}$	Travel distance from node $i$ to node $j$ , (m)
$E_{Mv}^k$	Energy consumption of MDC $k$ for moving, (J)
$e_{rx}$	Energy consumption to receive one bit of data, (J/bit)
$D$	Deadline of packet delay, (s)
$r$	Data generation rate of a sensor, (bits/s)
$E_{rx}^k$	Energy consumption of MDC $k$ to receive data from visited sensors, (J)
$Q$	Energy capacity of an MDC, (J)
$v$	Speed of MDCs, (s)
$\eta$	Energy transfer efficiency
$e_{min}$	Minimum residual energy of sensors, (J)
$t_{ij}$	Travel time from node $i$ to node $j$ , (s)
$T_k$	Time for one round journey of MDC $k$ , (s)
$e_s$	Energy consumption to generate one bit of data, (J/bit)
$e_{tx}$	Energy consumption to transmit one bit of data, (J/bit)
$u_i^k$	Number of nodes visited by MDC $k$ up to node $i$

Let us assume that a network consists of a set of nodes,  $V = \{1, \dots, N\}$ , where the element 1 represents the sink, and a set of arcs,  $E = \{(i, j) : i, j \in V, i \neq j\}$ . Furthermore, let us define  $M$  as a set of MDCs where  $M = \{1, \dots, K\}$ .

Let  $c_{ij}$  denote the travel distance (Euclidean distance) from node  $i$  to node  $j$ . We also define  $x_{ij}^k$  as a binary decision variable that becomes one if  $arc(i, j)$  is on the paths of the MDC  $k$ ,  $k \in M$ , but becomes zero otherwise. That is,

$$x_{ij}^k = \begin{cases} 1 & \text{if } arc(i, j) \text{ is on path of MDC } k. \quad i, j \in V, k \in M \\ 0 & \text{o/w} \end{cases} \quad (2)$$

Furthermore,  $u_i^k$  is defined as the number of nodes visited by MDC  $k$  up to node  $i$ . We also define  $t_{ij}$  as the travel time between node  $i$  and node  $j$ . Recall that  $t_i$  is the sojourn time of the MDC at node  $i$  for charging and collecting data. Note that  $t_1$  represents the sojourn time of the MDC at the sink. Each MDC spends a small amount of time,  $t_1$ , at the sink to replace its own battery.  $T_k$  refers to the time for one round journey of MDC  $k$  or the time for one period. Therefore,  $T_k$  can be calculated as follows:

$$T_k = \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1, \quad k = 1, \dots, K \quad (3)$$

Let  $E_m$  denote energy consumption by the MDC to travel one meter, and  $r$  is the data-generation rate of a sensor (in bits per s). Furthermore,  $E_i^{ch}$  denotes the energy transfer rate from the MDC to sensor  $i$ . Recall that  $e_{rx}$  is the energy consumption to receive one bit of data. We define  $E_Q^k$ ,  $E_{Mv}^k$  and  $E_{rx}^k$ , respectively, as energy consumed by MDC  $k$  for wireless charging of visited sensors, its movement

energy and the energy consumption to receive data from visited sensors during its one-round journey. Therefore, in each period, the total energy consumption of MDC  $k$  ( $k \in M$ ) can be calculated as follows:

$$E_Q^k + E_{Mv}^k + E_{rx}^k = E_i^{ch} t_i \sum_{i=1}^N \sum_{j=2}^N x_{ij}^k + E_m \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k c_{ij} + r e_{rx} D \sum_{i=1}^N \sum_{j=2}^N x_{ij}^k \quad (4)$$

Each sensor consumes energy for generating data and transmitting buffered data to the MDC. Recall that  $e_{tx}$  and  $e_s$  represent the consumed energy for transmitting and generating one bit of data, respectively. Furthermore, let  $\eta$  denote the energy transfer efficiency between the MDC and sensors. Then, after generating the data, transmitting them to the MDC and energy replenishment, the residual energy at sensor  $i$  by the end of a period is:

$$E_i^{ch} \eta t_i - r T_k e_s - r T_k e_{tx}, \quad i = 2, \dots, N \quad (5)$$

We replace  $T_k$  from Equation (3); then, Equation (5) can be rewritten as follows:

$$E_i^{ch} \eta t_i - r \left( \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1 \right) e_s - r \left( \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1 \right) e_{tx}, \quad i = 2, \dots, N \quad (6)$$

Now, the ILP problem can be formulated as follows:

$$\text{minimize} \quad E_i^{ch} t_i \sum_{k=1}^K \sum_{i=1}^N \sum_{j=2}^N x_{ij}^k + E_m \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k c_{ij} + r e_{rx} D \sum_{k=1}^K \sum_{i=1}^N \sum_{j=2}^N x_{ij}^k \quad (7)$$

$$\text{subject to} \quad \sum_{i=1}^N \sum_{k=1}^K x_{ij}^k = 1, \quad j \in V \setminus \{1\} \quad (8)$$

$$\sum_{j=1}^N \sum_{k=1}^K x_{ij}^k = 1, \quad i \in V \setminus \{1\} \quad (9)$$

$$\sum_{i=1}^N x_{ip}^k - \sum_{j=1}^N x_{pj}^k = 0, \quad k \in M, \quad p \in V \setminus \{1\} \quad (10)$$

$$\sum_{j=2}^N x_{1j}^k = 1, \quad k \in M \quad (11)$$

$$\sum_{i=2}^N x_{i1}^k = 1, \quad k \in M \quad (12)$$

$$\sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1 \leq D, \quad k \in M \quad (13)$$

$$E_i^{ch} t_i \sum_{i=1}^N \sum_{j=2}^N x_{ij}^k + E_m \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k c_{ij} + r e_{rx} D \sum_{i=1}^N \sum_{j=2}^N x_{ij}^k \leq Q, \quad k \in M \quad (14)$$

$$E_i^{ch} \eta t_i - r \left( \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1 \right) e_s - r \left( \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1 \right) e_{tx} \geq e_{min}, \quad i \in V \setminus \{1\} \quad (15)$$

$$u_1^k = 1, \quad k \in M \quad (16)$$

$$u_i^k - u_j^k + 1 \leq (N - 1)(1 - x_{ij}^k), \quad k \in M, \quad i, j \in V \setminus \{1\} \quad (17)$$

The objective function for the formulation is represented by Equation (7). Recall that every MDC consumes energy for wireless charging of visited sensors, its movement and receiving data from visited sensors. In Equation (7), the first, second and third term represent the energy consumption by all MDCs for wireless charging, moving and data reception, respectively. Therefore, Equation (7) presents the total energy consumption of all MDCs. The objective is to minimize the total energy consumption by finding optimal paths of MDCs.

Equations (8)–(17) represent constraints on the paths of the MDCs. More specifically, Constraints (8) and (9) ensure that every node except the sink is visited exactly once by only one MDC. For example, if MDC  $k$  arrives at node  $j$  ( $j \in V \setminus \{1\}$ ) from node  $i$ , then  $x_{ij}^k = 1$ . Since  $\sum_{i=1}^N \sum_{k=1}^K x_{ij}^k = 1$ , other MDCs from any other node cannot visit node  $j$ . Furthermore, when MDC  $k$  leaves node  $j$ , it only goes to one node among the  $N$  nodes to satisfy Constraint (9).

Constraint (10) is the route continuity constraint, which ensures that once an MDC arrives at a node, it must also leave that node. Specifically, if MDC  $k$  arrives at a node, say node  $p$  from node  $i$ , then  $x_{ip}^k = 1$ . Constraint (10) is satisfied whenever  $x_{pj}^k = 1$ , i.e., MDC  $k$  leaves node  $p$  and travels to another node  $j$ . Constraint (11) ensures that MDC  $k$  starts its tour from the sink to any other node, say node  $j$  (i.e.,  $\sum_{j=2}^N x_{1j}^k = 1$ ), and Constraint (12) ensures that MDC  $k$  returns to the sink from exactly one node among the  $N$  nodes (i.e.,  $\sum_{i=2}^N x_{i1}^k = 1$ ). Recall that 1 represents the sink.

Constraint (13) is the delay constraint for each MDC. It ensures that time for one round journey of MDC  $k$  ( $k \in M$ ), which includes the travel and charge time of the MDC, cannot exceed the given deadline,  $D$ . The time MDC  $k$  needs to finish a one-round journey is estimated in Equation (3). Constraint (14) limits the maximum energy consumption of each MDC. More specifically, it ensures that energy consumption by every MDC, which includes wireless charging, moving and data reception, cannot exceed its energy capacity,  $Q$ . The total energy consumption of MDC  $k$  ( $k \in M$ ) is calculated in Equation (4). Constraint (15) ensures that at the end of the current period, every sensor has the minimum residual energy,  $e_{min}$ , after being recharged by the MDC and consuming energy for generating and transmitting data. The residual energy of sensor  $i$  ( $i \in V \setminus \{1\}$ ) at the end of a period is presented in Equation (5).

Constraints (16) and (17) eliminate the sub-tours in the paths of the MDCs in a way similar to Miller–Tucker–Zemlin (MTZ) sub-tour elimination of the traveling salesman problem (TSP) [33]. MTZ sub-tour elimination constraints eliminate a collection of the disjoint sub-tour that together satisfies Constraints (8)–(15). Recall that  $u_i^k$  is the number of nodes visited by MDC  $k$  up to node  $i$ . Constraint (16) sets the value of  $u_i^k$  for the sink ( $i = 1$ ) to be one, i.e.,  $u_1^k = 1$ , and Constraint (17) ensures that  $u_j^k \geq u_i^k + 1$ , when  $x_{ij}^k = 1$ . Thus, Constraints (16) and (17) together prohibit the formation of any sub-tour within nodes in  $V \setminus \{1\}$ .

#### 4.3. Alternative Objectives

In this work, we focus on minimizing the total energy consumption. Depending on the requirements, alternative objectives can be considered. For example, the travel distance can be a main concern in some cases. Then, the problem can be defined as minimizing the total distance traveled by all MDCs and satisfying the delay and energy constraints. That is,

$$\text{minimize} \quad \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^K c_{ij} x_{ij}^k \quad (18)$$

Meanwhile, in some applications, mainly packet delay needs to be considered. In such a case, the objective can be minimizing the maximum tour time of the MDCs. Since data delivery relies on the physical movement of the MDCs, packet delay can be reduced by minimizing their tour time. Let  $\tau$  denote the maximum time from among the MDCs to finish one round journey. Thus, the problem

can also be expressed as follows: minimize the maximum tour time of MDCs,  $\tau$ , while satisfying the delay and energy constraints. That is,

$$\text{minimize } \tau \quad (19)$$

$$\text{subject to } \sum_{i=1}^N \sum_{j=1}^N x_{ij}^k (t_{ij} + t_i) - t_1 \leq \tau, \quad k \in M \quad (20)$$

$$\tau \leq D \quad (21)$$

In this case, delay Constraint (13) is replaced with Constraints (20) and (21). More specifically, Constraint (20) ensures that the travel time and charging time of every MDC cannot exceed the maximum tour time,  $\tau$ . Constraint (21) ensures that the maximum time an MDC needs to finish a one-round journey cannot exceed the given deadline,  $D$ .

## 5. Path Selection Algorithm with Delay and Energy Constraints

To find the optimal solution for the DCEM problem, the proposed ILP formulation can be solved using available optimization tools. However, it requires highly expensive computation and a long execution time when the number of MDCs or the number of sensors is large. Therefore, in this work, we propose efficient two-phase path-selection algorithms with delay and energy constraints. In Phase 1, clustering is performed to partition the sensors into several groups, and in Phase 2, the membership of each group is adjusted in order to meet the constraints and minimize the travel distances of MDCs. This paper proposes two clustering algorithms, angle-based clustering (ABC) and angle-based clustering with path-length ratio (ABC-PR). Then, two group membership-adjustment algorithms, nearest node assignment (NNA) and node assignment with maximal path-length decrease (MPD), are proposed. We first present the clustering algorithms. Then, two group membership-adjustment algorithms are described.

### 5.1. Clustering Algorithms

In our scheme, the sensor nodes are partitioned into  $K$  groups, where  $K$  is the number of MDCs used in the network. The generated data should be delivered to the sink within a given deadline,  $D$ , and the energy consumption of each MDC cannot exceed its energy capacity,  $Q$ . Thus, each MDC needs to finish its tour within delay constraint  $D$  and energy constraint  $Q$ . In order to satisfy the deadlines, it is desirable for MDCs to have a tour where the length is close to other MDCs, so that each MDC can meet the deadlines. Therefore, we propose the ABC and ABC-PR algorithms. The objective of ABC and ABC-PR is to divide the sensor nodes into  $K$  groups such that the travel distance of each MDC is close to the distance of the other MDCs.

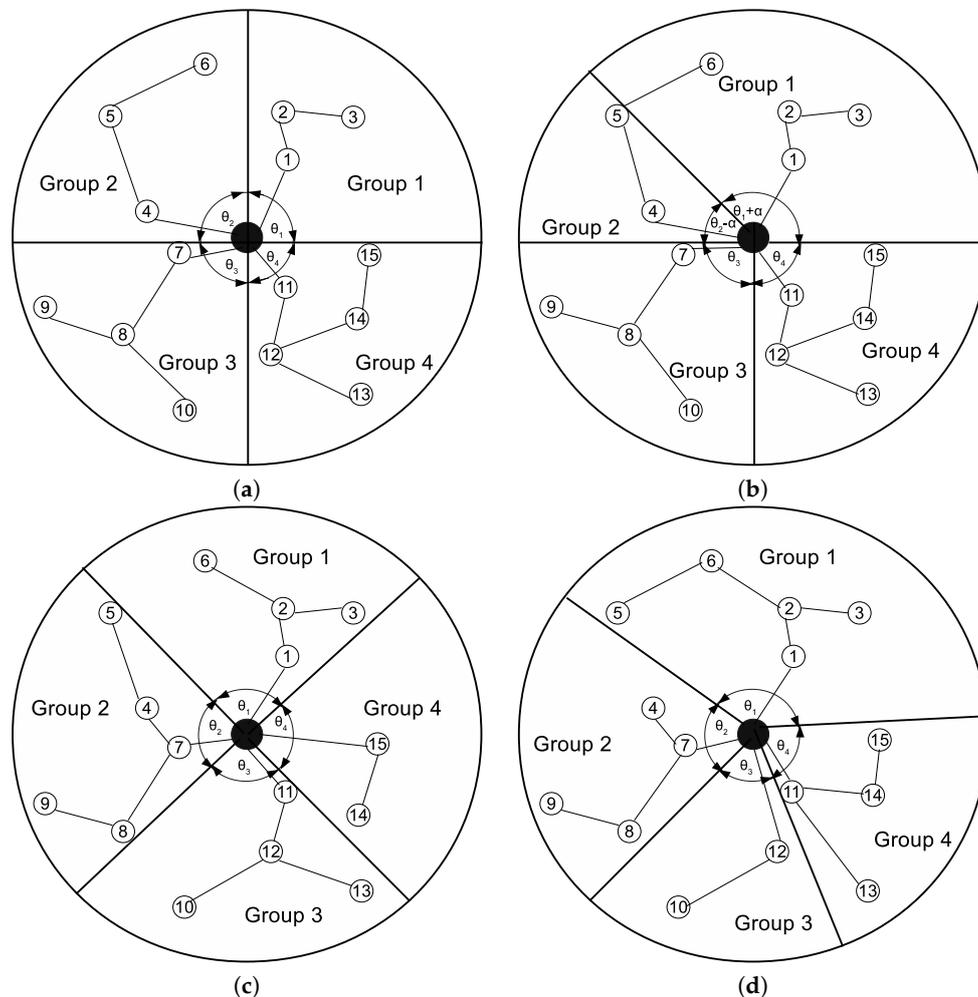
#### 5.1.1. Angle-Based Clustering

The main idea of the ABC algorithm is simple. ABC first divides the entire circular area into  $K$  sectors with equal angles, and the nodes located in the same sector belong to the same group. Then, the group (sector) angle is increased or decreased to transfer some nodes from one group to another. This process is repeated between two adjacent groups in a cyclic pattern in order to balance the travel distances of the MDCs with each other.

Figure 2 illustrates node grouping based on ABC. In Figure 2, the sink is located at the center of the area. Let  $\theta_i$  denote the angle of group  $i$ , ( $i = 1, \dots, K$ ), and assume there are four MDCs, i.e.,  $K = 4$ . Therefore, in the example in Figure 2a, the area is divided into four sectors such that  $\theta_1 = \theta_2 = \theta_3 = \theta_4$ . The nodes located in the same sector are considered as the same group members, and the sink is included in every group. In each group, a minimum spanning tree (MST) is built over the nodes, where each tree is rooted at the sink, as shown in Figure 2a. Then, the length for the MST of each group is estimated. Let us define  $L_i$  and  $L_{avg}$  as the length of the MST of group  $i$  ( $i = 1, \dots, K$ ) and the average length of the MSTs in all groups, respectively. Furthermore,  $\zeta$  represents a small positive constant.

If the difference between  $L_{avg}$  and  $L_i$  ( $i = 1, \dots, K$ ) is less than or equal to  $\zeta$ , we obtain the nodes for  $K$  groups. Otherwise, the MST length of the two groups is compared cyclically. Specifically, ABC compares the MST length of group  $i$  with  $i + 1$ , group  $i + 1$  with  $i + 2$  and group  $K$  with  $i$ . Let  $\alpha$  denote a rotation angle, and  $\alpha < \frac{\pi}{K}$ . In every comparison, if  $L_i > L_{i+1}$ ,  $\theta_i$  is decreased by  $\alpha$  and  $\theta_{i+1}$  is increased by  $\alpha$ . Otherwise, it is vice versa.

In the example in Figure 2a,  $L_1 < L_2$ . Therefore,  $\theta_1$  is increased by rotation angle  $\alpha$ , and  $\theta_2$  is decreased by  $\alpha$ , as shown in Figure 2b. As a result, Node 6 becomes a member of Group 1 instead of Group 2. The MST lengths for both groups are newly estimated with the addition of Node 6 in Group 1 and the exclusion of Node 6 from Group 2. After one iteration, nodes in the four groups are shown in Figure 2c. The new  $L_{avg}$  is estimated, and the difference between  $L_{avg}$  and  $L_i$  ( $i = 1, \dots, K$ ) is examined. If the difference is greater than  $\zeta$ , the next iteration begins. Let  $\beta$  denote a positive constant, and  $\beta < 1$ . For the next iteration,  $\alpha$  is set to  $\beta\alpha$ , and the same process is applied. This process is repeated until the difference between  $L_{avg}$  and  $L_i$  is either less than or equal to  $\zeta$  or  $\alpha$  is less than or equal to the threshold value,  $\varepsilon$ , where  $\varepsilon \leq 1$ . Figure 2d shows the final node groups obtained from ABC. The clustering algorithm is presented in Algorithm 1 (see Appendix A for the detailed clustering algorithm). In Algorithm 1, when ABC is used,  $\omega$  is obtained from Algorithm 2.



**Figure 2.** An example of angle-based clustering: (a) initial partitioning of the area and the minimum spanning tree (MST) of the sensors; (b)  $\alpha$  rotation between Group 1 and Group 2; (c) nodes in the four groups after the first iteration; and (d) the final node groups.

**Algorithm 1** Clustering algorithm.**Input:**  $V, M, \alpha, \zeta, \beta$ 

$V$ : Set of sensors  $\triangleright V = \{1, 2, \dots, N\}$   
 $M$ : Set of groups or MDCs  $\triangleright M = \{1, 2, \dots, K\}$   
 $\alpha$ : Rotation angle  $\triangleright \alpha < \frac{\pi}{K}$   
 $\zeta$ : Small positive constant  $\triangleright \zeta \geq 1$   
 $\beta$ : Positive constant  $\triangleright 0 < \beta < 1$   
**Output:**  $fin\Omega_K = \{N_1, N_2, \dots, N_K\}$   $\triangleright$  Set of nodes for  $K$  groups

```

1: Initialization;
2:  $N_i$  = Set of nodes of group  $i, i \in M$ ;
3: Divide the area into  $K$  sectors where  $\theta_1 = \theta_2 = \dots = \theta_K = \frac{2\pi}{K}$  and  $cur\Omega_K = \{N_1, N_2, \dots, N_K\}$ ;
4: For  $i = 1, \dots, K$ , calculate minimum spanning tree (MST) length  $L_i$  and average length of all
   MSTs  $L_{avg}$ ;
5: if  $|L_i - L_{avg}| > \zeta$  then  $\triangleright$  for  $i \in M$ 
6:   for each  $(G_i \% K)$  group and  $(G_{i+1} \% K)$  group do
7:     use ABC or ABC-PR to update  $\omega$ ;
8:     update  $\theta_i$  and  $\theta_{i+1}$  to  $\theta_i - \omega$  and  $\theta_{i+1} + \omega$ , respectively;
9:     update  $L_i$  and  $L_{i+1}$  using  $N_i, N_{i+1}, \theta_i, \theta_{i+1}$ ;
10:  end for
11: end if
12: Calculate new  $L_{avg}$ ;
13: if  $|L_i - L_{avg}| > \zeta$  then  $\triangleright$  for  $i \in M$ 
14:   update  $\alpha$  to  $\beta\alpha$ ;
15:   goto 6;
16: else
17:    $fin\Omega_K = cur\Omega_K$ ;
18:   exit;
19: end if

```

**Algorithm 2** Angle-based clustering (ABC).

```

1: function ABC( $L_i, L_{i+1}, \alpha$ )
2:   if  $L_i > L_{i+1}$  then
3:      $\omega = \alpha$ ;
4:   else if  $L_i < L_{i+1}$  then
5:      $\omega = -\alpha$ ;
6:   else
7:      $\omega = 0$ ;
8:   end if
9:   return  $\omega$ 
10: end function

```

## 5.1.2. Angle-Based Clustering with Path-Length Ratio

In ABC,  $\theta_i$  is increased or decreased by rotation angle  $\alpha$ , which remains the same for one iteration. For the second iteration,  $\alpha$  is set to  $\beta\alpha$ , and so on. Note that in ABC, the value of the rotation angle does not reflect the path lengths of the groups. As a result, when the ABC algorithm is used, travel distance for MDCs remains unbalanced, and an MDC may travel a longer distance, compared to the others. In order to address the problem of ABC, ABC-PR first estimates the path-length ratio to determine how much angular rotation is needed to balance the travel distances of the MDCs between two groups. After that, actual rotation angle  $\alpha'$  is calculated based on the path-length ratio and rotation angle  $\alpha$ . More specifically, in the MST length comparison between groups  $i$  and  $i + 1$ , actual rotation angle  $\alpha'$  is set to  $\frac{\max(L_i, L_{i+1})}{L_i + L_{i+1}} \alpha$ . When  $L_i > L_{i+1}$ , ABC-PR decreases  $\alpha'$  from  $\theta_i$ , and increases  $\alpha'$  to  $\theta_{i+1}$ . On the other hand, if  $L_i < L_{i+1}$ ,  $\theta_i$  and  $\theta_{i+1}$  are increased and decreased by  $\alpha'$ , respectively. The estimation of  $\alpha'$  in ABC-PR is presented in Algorithm 3.

Now, we estimate the time complexity of the clustering algorithm for one iteration. In the clustering algorithms, the MST lengths of node groups are estimated. The maximum number of nodes in each group is  $N$ , and the time complexity for finding MST is  $O(N^2)$ . The number of calculations for the rest is constant. Therefore, the time complexity of the clustering algorithms can be expressed as  $O(N^2)$ .

---

**Algorithm 3** Angle-based clustering with path-length ratio (ABC-PR).

---

```

1: function ABCPR( $L_i, L_{i+1}, \alpha$ )
2:    $\alpha' = \frac{\max(L_i, L_{i+1})}{L_i + L_{i+1}} \alpha$  ▷ Calculate  $\alpha'$ 
3:   if  $L_i > L_{i+1}$  then
4:      $\omega = \alpha$ ;
5:   else if  $L_i < L_{i+1}$  then
6:      $\omega = -\alpha'$ ;
7:   else
8:      $\omega = 0$ ;
9:   end if
10:  return  $\omega$ 
11: end function

```

---

## 5.2. Group Membership-Adjustment Algorithms

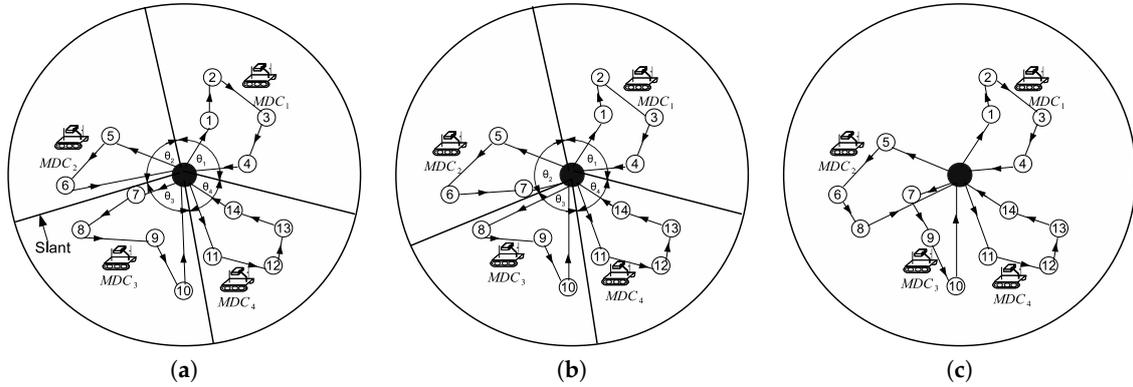
We obtained  $K$  sensor groups from the clustering algorithms, and each MDC is assigned to a group for collecting data and charging the sensor nodes' batteries. In order to meet the constraints of each MDC and minimize the total travel distance of the MDCs, we propose two group membership-adjustment algorithms: nearest node assignment and node assignment with maximal path-length decrease.

### 5.2.1. Nearest Node Assignment

In the NNA algorithm, an MDC's path for a group is first determined by the twice around the tree heuristic algorithm of TSP [34]. The borderline between a group and its clockwise adjacent group is defined as the slant. NNA selects the group that has the maximum travel distance among all of the groups. From this group, the node that is nearest to the slant is assigned to its clockwise adjacent group in order to reduce the travel distance of the group with maximum travel distance. The process is repeated to achieve the minimum travel distance, while each MDC satisfies delay and energy constraints. More specifically, let us define  $b$  as the number of rounds, and each round consists of  $S$  steps. We also define  $G_i$  as a group with the maximum travel distance of MDC, and  $G_{i-1}$  is its clockwise adjacent group. Let  $MDC_i$  and  $MDC_{i-1}$  denote the MDCs for  $G_i$  and  $G_{i-1}$ , respectively. Furthermore, let us assume that  $N_i$  and  $N_{i-1}$  are sets of nodes in  $G_i$  and  $G_{i-1}$ , respectively. Let  $h$  denote a node in  $N_i$  that is nearest to the slant. Therefore,  $h$  is re-assigned from  $N_i$  to  $N_{i-1}$ , and find the new path for both MDCs. This process is repeated  $S$  times. Among the  $S$  steps, NNA select the paths for MDCs that lead to minimizing the total travel distance while meeting delay and energy constraints for each MDC. Then, the next round begins, and the process is repeated until no further decrease in the total travel distance can be achieved.

Figure 3 illustrates the node assignment process of the proposed algorithm. In the example in Figure 3a, nodes are partitioned into four groups using the clustering algorithm. In each group, an MDC is employed to collect data and recharge sensor nodes' batteries. The path of each MDC is obtained by using the twice around the tree heuristic algorithm of TSP, and the path length is estimated. As shown in an example in Figure 3a,  $MDC_3$  has the maximum travel distance and node set of Group 3,  $N_3 = \{7, 8, 9, 10\}$ . From  $N_3$ , Node 7 is chosen for reassignment from Group 3 to Group 2, since Node 7 is nearest to the slant. Therefore, Node 7 is removed from  $N_3$  and added to  $N_2$ , i.e.,  $N_2 = \{5, 6, 7\}$  and  $N_3 = \{8, 9, 10\}$ . New paths of  $MDC_2$  and  $MDC_3$  are formed with the current nodes of each group, as shown in Figure 3b. The group membership-adjustment algorithm is presented in Algorithm 4

(see Appendix B for the detailed group membership-adjustment algorithm). In Algorithm 4, when NNA is used,  $h$  is obtained from Algorithm 5.



**Figure 3.** An example of group membership-adjustment algorithms: (a) initial route of MDCs, (b) nearest node assignment and (c) node assignment with maximal path-length decrease.

---

**Algorithm 4** Group membership-adjustment algorithm.

---

**Input:**  $M, fin\Omega_K, D, Q, P$

$M$ : Set of groups or MDCs

▷  $M = \{1, 2, \dots, K\}$

$fin\Omega_K$ : Set of nodes for  $K$  groups

▷  $fin\Omega_K = \{N_1, N_2, \dots, N_K\}$

$D$ : Deadline for delay

$Q$ : Energy capacity of MDC

$P$ : Minimum residual energy of sensor

**Output:**  $R = \{R_1, R_2, \dots, R_K\}$

▷ Set of paths for  $K$  MDCs

```

1: Initialization;
2: for each group do
3:   use TSP heuristics to find path of  $MDC_i, R_i$ ;
4:   calculate  $d_i, T_i, E_i, P_i$ ;      ▷  $d_i$ : travel distance of  $MDC_i, T_i$ : travel time of  $MDC_i, E_i$ : energy
consumption of  $MDC_i, P_i$ : residual energy of sensor
5: end for
6: while  $Z_{temp} \geq Z_{cur}$  do
7:   update  $Z_{temp}$  to  $Z_{cur}$  and update  $R$  to  $R_{cur}$ ;
8:   increment  $b$ ;
9:   while  $s \leq S$  do
10:    find group  $i$ , which has the maximum travel distance of MDC;
11:    use NNA or MPD to find node  $h$ , which will be re-assigned;
12:    remove  $h$  from  $N_i$  and add  $h$  to  $N_{i-1}$ ;
13:    find new  $R_i, R_{i-1}$ ;
14:    calculate new  $d_i, T_i, E_i$  and  $P_i$ ;
15:    calculate new  $d_{i-1}, T_{i-1}, E_{i-1}$  and  $P_{i-1}$ ;
16:    calculate total travel distance  $Z$ ;
17:   end while
18:    $R_{cur}$  = set of MDCs' paths with  $min(Z)$  and  $\{T_i \leq D \ \&\& \ E_i \leq Q \ \&\& \ P_i \geq P\}$ ;
19:   update  $Z_{cur}$  to  $min(Z)$ ;
20: end while
21: return  $R$  as the output.

```

---

**Algorithm 5** Nearest node assignment (NNA).

---

```

1: function NNA( $N_i, slant$ )
2:    $h =$  node in  $N_i$  nearest to slant;
3:   return  $h$ 
4: end function

```

---

**Algorithm 6** Node assignment with maximal path-length decrease (MPD).

---

```

1: function MPD( $N_i, N_{i-1}$ )
2:   for each  $h \in N_i$  do
3:      $tempN_i = \emptyset; tempN_{i-1} = \emptyset;$ 
4:      $tempN_i = N_i \setminus \{h\}; tempN_{i-1} = N_{i-1} \cup \{h\};$   $\triangleright$  examines every node membership change
5:     from  $N_i$  to  $N_{i-1}$ 
6:        $R_i = tspHeuristic(tempN_i);$   $\triangleright$  find  $R_i$  using  $tempN_i$ 
7:        $R_{i-1} = tspHeuristic(tempN_{i-1});$   $\triangleright$  find  $R_{i-1}$  using  $tempN_{i-1}$ 
8:       Calculate  $Y = d_i + d_{i-1};$   $\triangleright Y$ : travel distance of  $MDC_i$  and  $MDC_{i-1}$ 
9:     end for
10:     $h =$  node in  $N_i$  whose membership change gives  $min(Y);$ 
11:  return  $h$ 
12: end function

```

---

## 5.2.2. Node Assignment with Maximal Path-Length Decrease

In NNA, from all nodes in  $N_i$ , the node that is nearest to the slant is chosen for reassignment to the new group. However, such a simple strategy may not result in the greatest travel distance reduction. Moreover, there are cases where the newly assigned node is far away from the current nodes of  $G_{i-1}$ . In such a case, the path length of  $MDC_{i-1}$  may increase significantly. In order to address the problem of NNA algorithm, MPD first evaluates the path lengths of  $MDC_i$  and  $MDC_{i-1}$  by considering every possible node membership change from  $N_i$  to  $N_{i-1}$ . Then, the node is selected from  $N_i$ , whose membership change reduces the path length of both MDCs the most.

Let us define  $d_i$  and  $d_{i-1}$  as the travel distances of  $MDC_i$  and  $MDC_{i-1}$ , respectively. Furthermore, let  $Y = d_i + d_{i-1}$ . In MPD, each node is assigned from  $N_i$  to  $N_{i-1}$  sequentially, and  $Y$  is estimated in every case. Let  $h$  denote a node in  $N_i$  whose membership change gives the minimum  $Y$ . Therefore,  $h$  is actually reassigned to  $N_{i-1}$ , and the other nodes remain in  $N_i$ . As shown in the example in Figure 3c, Node 8's membership changes from Group 3 to Group 2 since the minimum  $Y$  can be achieved. The rest of the steps are similar to NNA. In Algorithm 4, when MPD is used,  $h$  is obtained from Algorithm 6.

Now, we compute the time complexity of the group membership-adjustment algorithm for one round. In the group membership-adjustment algorithm, the TSP heuristic algorithm is applied to find the MDCs' paths. The time complexity for finding the paths for MDCs is  $O(N^2 \log_2 N)$ . Therefore, the time complexity of the group membership-adjustment algorithm when the NNA algorithm is used can be expressed as  $O(N^2 \log_2 N)$ . Since MPD examines every assignment change from one group to another, the time complexity of the group membership-adjustment algorithm in the case of MPD is  $O(N^3 \log_2 N)$ .

**6. Performance Study**

In this section, we analyze the performance of the proposed algorithms. This paper proposes two group membership-adjustment algorithms, NNA and MPD. Furthermore, two clustering algorithms (ABC and ABC-PR) are proposed. Therefore, four combinations of the proposed algorithms are evaluated in the simulations:

- NNA with ABC
- NNA with ABC-PR
- MPD with ABC
- MPD with ABC-PR

We compare the performance of the proposed algorithms with two heuristic algorithms for VRP and VRPTW: vertex relocation (VR) [35] and generalized assignment (GA) [36]. We first present the simulation parameters and performance metrics and then discuss the performance results under various scenarios.

### 6.1. Simulation Setup

Sensor nodes are randomly deployed in a circular area with a radius of 500 m, where the sink is placed at the center of the region. Each sensor generates a data packet every two seconds and stores the packet in its buffer until an MDC collects it. When an MDC arrives at a sensor node, it will spend two seconds charging the sensor node's battery and then collects data from this sensor. The energy transfer rate from MDC to sensor is 5 J/s [25]. Based on the study in [24], energy transfer efficiency  $\eta$  and the distance between MDC and sensors to be charged are assumed to be 40% and 2 m, respectively. The energy capacity of the MDC is  $Q = 50$  KJ [28]. In addition, we adopt the following parameter values for the energy consumption model:  $\gamma = 2$ ,  $\rho = 10(pJ/bit/m^2)$  and  $e_T = e_R = e_S = 50$  nJ/bit [32]. We assume that an MDC consumes 8.27 joules to travel one meter (or 0.21 J/inch) [37]. Major simulation parameters are summarized in Table 2.

**Table 2.** Simulation parameters. Bold numbers represent default values.

Parameter	Value
Radius of circular area ( $R$ )	{400 <b>500</b> 600 700 800} m
Number of MDCs ( $K$ )	{2 3 <b>4</b> 5 6 7}
Number of sensors ( $N$ )	{20 40 <b>60</b> 80 100 120}
Data generation rate of sensor ( $r$ )	0.5 packet/s
Packet size	64 bytes or 512 bits
Speed of MDCs ( $v$ )	5 m/s
Number of steps ( $S$ )	20
Energy transfer efficiency ( $\eta$ )	40%
Sojourn time of MDC or charging time of sensor $i$ ( $t_i$ )	2 s
Deadline of packet delay ( $D$ )	800 s
Energy capacity of MDC ( $Q$ )	50 KJ
Energy transfer rate from MDC to sensor $i$ ( $E_i^{ch}$ )	5 J/s
Energy consumption of MDC to travel one meter ( $E_m$ )	8.27 J/m
Energy consumption for generating one bit of data ( $e_S$ )	50 nJ/bit
Energy usage per bit of transmission electronics ( $e_T$ )	50 nJ/bit
Energy usage per bit of reception electronics ( $e_R$ )	50 nJ/bit

In order to obtain fair results, we performed the simulations with five different seed values and took the average value. The GNU Linear Programming Kit [38] is used to solve the ILP problem. The simulator has been built using the Java language. Three performance metrics are used to evaluate the performance of algorithms:

- total energy consumption: total energy consumed by all MDCs to finish one tour
- maximum energy consumption: the maximum energy consumed among MDCs to finish one tour
- maximum travel time: the maximum travel time among MDCs to finish one tour

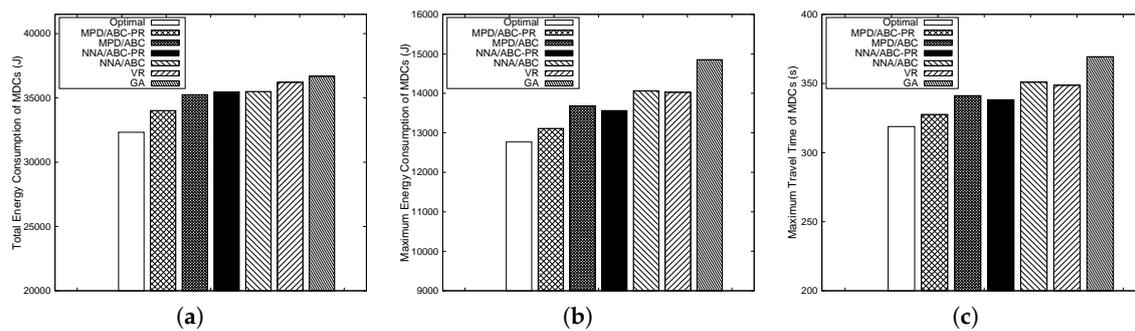
### 6.2. Performance Analysis in Small-Scale Scenarios

We first compare the performance of the proposed algorithms with the optimal result obtained by solving the ILP formulation and VR [35] and GA [36] algorithms. The execution time to achieve the optimal solution increases exponentially with the number of MDCs. Thus, we consider small-scale network scenarios when comparing the performance of the proposed algorithms with the optimal solution. In this subsection, we consider 15 sensor nodes with three and four MDCs to obtain the optimal solution from the ILP formulation. Other parameters remain unchanged. Figures 4 and 5

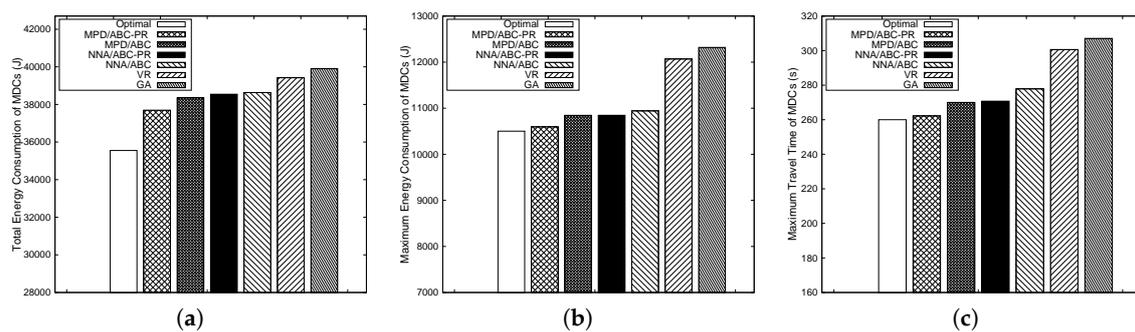
illustrate the performance of the considered schemes. The deadline for packet delay is set to 350 s and 280 s when the number of MDCs is three and four, respectively. Since MDCs collect data and deliver the collected data to the sink, each MDC needs to finish its tour within the given deadline to satisfy the delay requirement.

Figures 4a and 5a show the total energy consumed by all MDCs under different schemes when the number of MDCs is three and four, respectively. The maximum energy consumption and the maximum travel time of MDCs are also presented in Figures 4 and 5.

As shown in Figures 4 and 5, the solution from the ILP always exhibits the lowest total energy and the lowest travel time of MDCs, since it can find the optimal paths for all MDCs. Among the proposed algorithms, MPD performs better than NNA. In particular, MPD/ABC-PR shows the best performance, except for the optimal solution. The reason is that ABC-PR estimates the path-length ratio to determine how much angular rotation is needed between two groups to balance the travel distance of the MDCs and then calculates the actual rotation angle from the path-length ratio and rotation angle. Therefore, the differences in the travel distances of MDCs tend to be small, which results in a more balanced energy consumption. As seen in Figures 4b and 5b, the gap between maximum energy values with MPD/ABC-PR and the optimal solution is quite small. The maximum energy consumed by MDCs with MPD/ABC-PR is only 3% and 1% higher than the optimal solution when the number of MDCs is three and four, respectively.



**Figure 4.** Performance of different algorithms for 15 nodes and three MDCs. (a) Total energy consumption of MDCs; (b) Maximum energy consumption of MDCs; (c) Maximum traveling time of MDCs.



**Figure 5.** Performance of different algorithms for 15 nodes and four MDCs. (a) Total energy consumption of MDCs; (b) Maximum energy consumption of MDCs; (c) Maximum traveling time of MDCs.

For the group membership-adjustment algorithms, the group that has the maximum travel distance is chosen to reduce travel distance, and from this group, a node is reassigned to a neighboring group. MPD selects the node, such that the maximum travel distance reduction can be achieved. On the other hand, NNA selects the node that is nearest to the slant. In some cases, the newly assigned node is far from the current nodes in the group, and the travel distance of the MDCs can increase. Therefore,

in Figures 4a and 5a, the total energy consumption in NNA is higher than MPD. For example, when the number of MDCs is three, under ABC-PR, the total energy consumption of MPD and NNA is about 5% and 9% higher than the optimal solution, respectively.

In addition, as shown in Figures 4 and 5, GA shows the highest maximum energy and travel time among the algorithms. Moreover, under GA, the total energy consumed by all MDCs is much higher than the others. Under GA, the difference between the longest and shortest paths for MDCs is high, which results in unbalanced energy consumption. Due to unbalanced paths of the MDCs, each MDC cannot satisfy the delay requirement, as shown in Figures 4c and 5c. VR also shows high energy consumption, and the travel time of MDCs compared to all variants of the proposed algorithms when the number of MDCs is four and does not satisfy the delay requirement either. In both cases, all variants of our proposed algorithms can meet the delay requirement for each MDC. For example, when the number of MDCs is four, under ABC-PR, the maximum travel time of MPD and NNA is about 38 s (13%) and 30 s (10%) lower than VR, respectively.

### 6.3. Performance Analysis by Changing Network Parameters

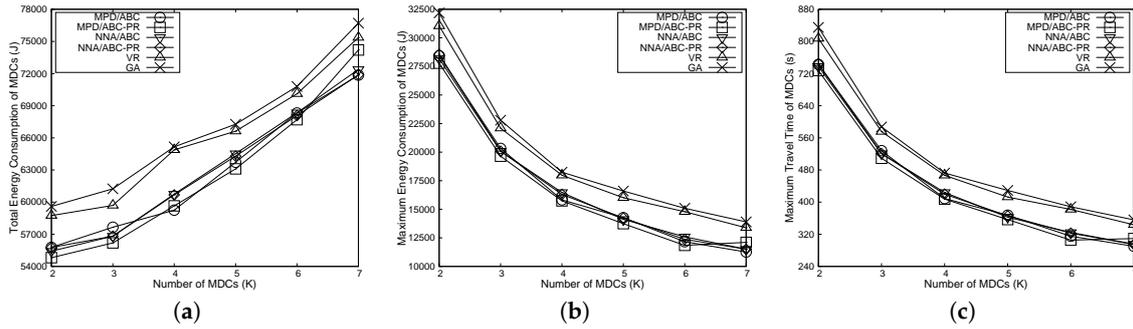
In this subsection, we compare the performance of the proposed algorithms, VR and GA, under different numbers of MDCs, sensors and different area radii. Note that we do not evaluate the performance of the ILP solution, since it takes a large amount of time to find the solution of the ILP formulation, particularly when the number of MDCs is large. Furthermore, when we collected the results by changing network parameters, in all simulations, the proposed algorithms terminated within three rounds. Since each round consists of 20 iterations (or steps), the algorithms performed 60 iterations to find the solution for the DCEM problem.

#### 6.3.1. Effects of the Number of MDCs

We compare the network performance of the considered algorithms for different numbers of MDCs, which change from 2 to 7. As seen in Figure 6a, the total energy consumption of MDCs is higher when there are more MDCs. As shown in Figure 6a, four variants of our proposed algorithms always exhibit a lower total energy than VR and GA over different numbers of MDCs. For instance, when the number of MDCs is four, MPD/ABC-PR uses 10% and 8% less energy than GA and VR, respectively. This is because, in the proposed algorithms, nodes are partitioned into multiple groups using clustering in such a way that the travel distance of each MDC is close to the other MDCs. Moreover, in the group membership-adjustment algorithms, path length is further reduced to achieve the minimum total travel distance of MDCs while satisfying delay and energy constraints. Due to two-phase path-length balancing and minimization, MPD/ABC-PR can achieve lower energy than the others.

Figure 6b compares the maximum energy consumed among the MDCs under the algorithms. As shown in Figure 6b, the maximum energy consumed by MDCs tends to decrease when the number of MDCs increases. In addition, all variants of the proposed algorithms have lower maximum energy than VR and GA. In particular, MPD/ABC-PR consistently shows the lowest maximum energy, compared to the others. For example, when the number of MDCs is five, the maximum energy consumed among MDCs to finish one tour is about 13,700 joules under MPD/ABC-PR, compared to 16,013 joules and 16,600 joules in the cases of VR and GA, respectively. The reason is that in VR and GA, the paths for MDCs are unbalanced, and an MDC travels a long path, compared to others, which results in unbalanced energy consumption.

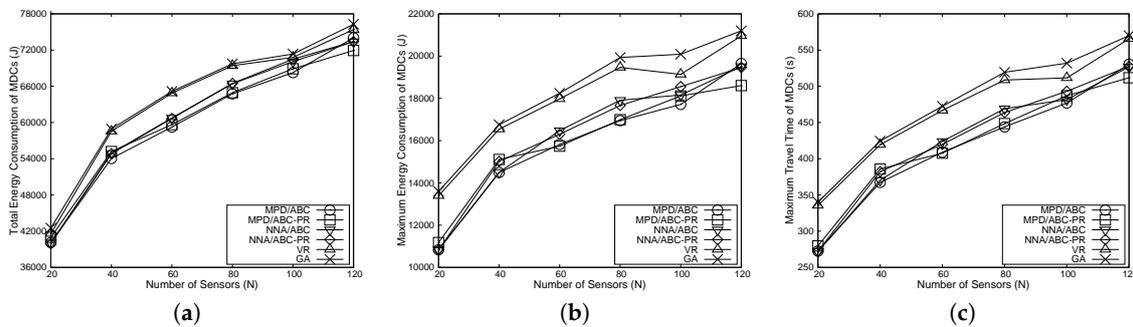
Figure 6c shows the maximum travel time of MDCs in the considered schemes. As shown in the Figure 6c, when the number of MDCs increases, the maximum travel time of MDCs decreases, since the distance traveled by each MDC decreases. Moreover, the two VRP heuristic algorithms always exhibit a longer maximum travel time than the others and do not satisfy the deadline for packet delay when the number of MDCs is two. In contrast, four variants of our proposed algorithms can meet the delay requirements in all cases.



**Figure 6.** Effects of the number of MDCs: (a) effects on total energy consumption of MDCs; (b) effects on maximum energy consumption of MDCs and (c) effects on maximum travel time of MDCs.

6.3.2. Effects of the Number of Sensors

The effects of the number of sensors on network performance are examined for all variants of the proposed algorithm and the VR and GA algorithms. The number of sensors is set to {20, 40, 60, 80, 100, 120}, while other parameters remain unchanged. Figure 7a shows that the energy consumption of algorithms consistently increases as the number of sensors grows, since each MDC needs to visit more sensors. Moreover, as in the previous subsection, four variants of the proposed algorithms always show a lower total energy consumption than VR and GA over the variations in the number of sensors. In addition, we can see from Figure 7a that the gap between energy values of the two heuristic algorithms and our proposed algorithms is greater when the number of sensors varies between 40 and 80. For example, when the number of sensors is 60, the total energy consumption under MPD/ABC-PR is 59,653.26 joules, which is much lower than the consumed energy of GA (65,162.88 joules) and VR (64,875.77 joules).



**Figure 7.** Effects of the number of sensors: (a) effects on total energy consumption of MDCs; (b) effects on maximum energy consumption of MDCs and (c) effects on maximum travel time of MDCs.

In terms of the maximum energy consumption of MDCs, Figure 7b illustrates that the maximum energy consumed by MDCs in the considered algorithms tends to increase, and MPD/ABC-PR can achieve the lowest maximum energy consumption among the considered algorithms. This result is due to the following reasons. In ABC-PR, we balance the travel distance of MDCs as much as possible. After that, in MPD, the group having the maximum travel distance is chosen to further reduce path length. This results in the lowest energy consumption, since the MDCs consume most of their energy for movement.

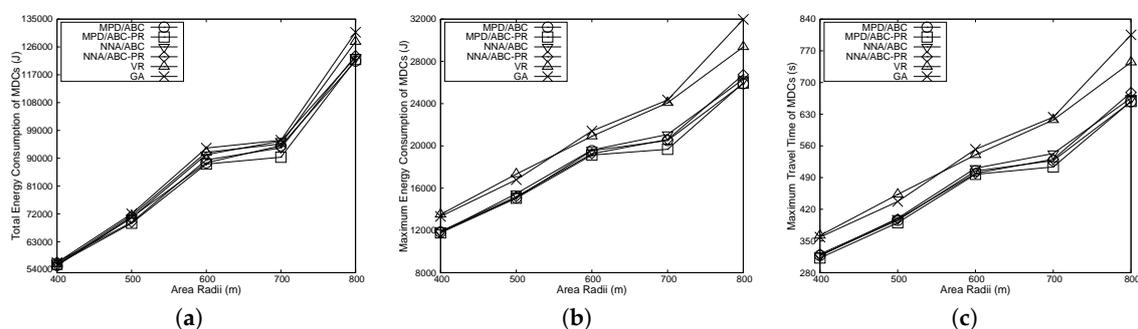
As seen in Figure 7c, four variants of our algorithms always exhibit a lower maximum travel time than the other two algorithms. Furthermore, as shown in Figure 7c, when the same clustering algorithm is used, MPD has a lower maximum travel time than NNA. The reason is that, under NNA, the node that is closest to the slant is chosen for reassignment. This assignment change may not result in the lowest travel distance for MDC. In contrast, MPD first examines every assignment change

and then selects the node for the actual assignment that reduces the path length by the maximum amount. Thus, MPD shows better performance compared to NNA. For example, when the number of sensors is 80, under ABC-PR, the maximum travel time of MPD and NNA is about 14% and 11% lower than GA, respectively.

### 6.3.3. Effects of Area Radii

In this subsection, we analyze the effects of area radii on the network performance. Radii of the circular area are varied between 400 m and 800 m. In this case, we consider the number of sensors and the number of MDCs to be 80 and five, respectively. Other parameters are kept constant. Figure 8a shows that total energy consumption rises with an increased radius. Moreover, the energy consumption of the proposed algorithm is similar to others. This result is due to the following reasons. When the same number of sensors is considered with an increased area, MPD and NNA may select similar nodes for reassignment, which results in similar energy consumption under all of the proposed algorithms.

Figure 8b,c show the variation in maximum energy and maximum travel time of the algorithms over different values for the radius. As can be seen from Figure 8b,c, the gap between the results of the two heuristic algorithms and our proposed algorithms tend to increase as area radii increase. For example, when area radius is set to 700 m, the maximum energy consumption of MPD/ABC-PR is about 4660 joules and 4400 joules lower than GA and VR, respectively. This is due to the following reasons. When the radius increases and the number of nodes is unchanged, distance among the nodes also increases, since nodes are deployed randomly in a circular area. GA and VR both only focus on minimizing total travel distance of the MDCs while finding paths for them. In contrast, our proposed algorithms find the MDCs' paths in a balanced way with the minimum total travel distance. This leads to the increased gap between the heuristic algorithms and our proposed algorithms.



**Figure 8.** Effects of area radii: (a) effects on total energy consumption of MDCs; (b) effects on maximum energy consumption of MDCs and (c) effects on maximum travel time of MDCs.

## 7. Conclusions

This paper considered a data collection scheme in which we use multiple MDCs to collect data from the sensors. In addition, MDCs also recharge the visited sensors to keep the WSN operational without the risk of energy depletion. In this paper, an optimization problem called DCEM was defined that finds the paths for MDCs in order to minimize the energy consumption considering delay and energy constraints.

We proposed the ILP optimization formulation to find the optimal solution for the DCEM problem. Furthermore, a two-phase path-selection algorithm was proposed to efficiently solve the DCEM problem. In the path-selection algorithm, clustering was performed in Phase 1 to partition the sensors into multiple groups, and in Phase 2, the membership of each group was adjusted to meet the constraints and minimize the energy consumption of MDCs. Simulations were conducted to evaluate the performance of the proposed algorithms in various scenarios. The performance of the proposed

algorithms was compared with two heuristic algorithms for VRP. Simulation results showed that the proposed algorithms can achieve lower energy consumption while guaranteeing data delivery.

**Acknowledgments:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A1003732) and the Ministry of Education (NRF-2016R1D1A3B03934617).

**Author Contributions:** Rumpa Dasgupta and Seokhoon Yoon developed the algorithms, conducted the experimental simulations and performed the analysis of this work. Rumpa Dasgupta and Seokhoon Yoon wrote and approved the final manuscript.

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

## Appendix A. Clustering Algorithm.

---

**Input:**  $V, M, \alpha, \zeta, \beta, \varepsilon$

$V$ : Set of sensors ▷  $V = \{1, 2, \dots, N\}$

$M$ : Set of groups or MDCs ▷  $M = \{1, 2, \dots, K\}$

$\alpha$ : Rotation angle ▷  $\alpha < \frac{\pi}{K}$

$\zeta$ : Small positive constant ▷  $\zeta \geq 1$

$\beta$ : Positive constant ▷  $0 < \beta < 1$

$\varepsilon$ : Threshold value ▷  $\varepsilon \leq 1$

**Output:**  $fin\Omega_K = \{N_1, N_2, \dots, N_K\}$  ▷ Set of nodes for  $K$  groups

- 1:  $fin\Omega_K = \{\emptyset\}; cur\Omega_K = \{\emptyset\};$
- 2:  $N_i =$  Set of nodes of group  $i, i \in M;$
- 3: Divide the area into  $K$  sectors where  $\theta_1 = \theta_2 = \dots = \theta_K = \frac{2\pi}{K}$  and  $cur\Omega_K = \{N_1, N_2, \dots, N_K\};$
- 4: For  $i = 1, \dots, K$ , calculate minimum spanning tree (MST) length  $L_i$  and average length of all MSTs  $L_{avg};$
- 5: **if**  $|L_i - L_{avg}| > \zeta$  **then** ▷ for  $i \in M$
- 6:     **for each**  $(G_i \% K)$  group and  $(G_{i+1} \% K)$  group **do**
- 7:         **if** ABC to be used **then**
- 8:              $\omega = ABC(L_i, L_{i+1}, \alpha);$
- 9:              $\theta_i = \theta_i - \omega; \theta_{i+1} = \theta_{i+1} + \omega;$  ▷ New  $\theta_i$  and  $\theta_{i+1}$  based on  $\alpha$
- 10:         **else if** ABC-PR to be used **then**
- 11:              $\omega = ABCPR(L_i, L_{i+1}, \alpha);$
- 12:              $\theta_i = \theta_i - \omega; \theta_{i+1} = \theta_{i+1} + \omega;$  ▷ New  $\theta_i$  and  $\theta_{i+1}$  based on  $\alpha'$
- 13:         **end if**
- 14:          $L_i = calcDist(N_i, N_{i+1}, \theta_i, \theta_{i+1});$
- 15:          $L_{i+1} = calcDist(N_i, N_{i+1}, \theta_i, \theta_{i+1});$
- 16:     **end for**
- 17: **end if**
- 18: Calculate new  $L_{avg};$
- 19: **if**  $|L_i - L_{avg}| > \zeta$  **then** ▷ for  $i \in M$
- 20:     **if**  $\alpha > \varepsilon$  **then**
- 21:          $\alpha = \beta\alpha;$
- 22:         goto 6;
- 23:     **else**
- 24:          $fin\Omega_K = cur\Omega_K;$
- 25:         exit;
- 26:     **end if**
- 27: **else**
- 28:      $fin\Omega_K = cur\Omega_K;$
- 29:     exit;
- 30: **end if**

---

## Appendix B. Group Membership-Adjustment Algorithm.

---

**Input:**  $M, fin\Omega_K, D, Q, P, S$

$M$ : Set of groups or MDCs

▷  $M = \{1, 2, \dots, K\}$

$fin\Omega_K$ : Set of nodes for  $K$  groups

▷  $fin\Omega_K = \{N_1, N_2, \dots, N_K\}$

$D$ : Deadline for delay

$Q$ : Energy capacity of MDC

$P$ : Minimum residual energy of sensor

$S$ : Number of steps

**Output:**  $R = \{R_1, R_2, \dots, R_K\}$

▷ Set of paths for  $K$  MDCs

```

1:  $R = \{\emptyset\}; R[b] = \{\emptyset\}; R_{cur} = \{\emptyset\}; Z_{cur} = 0; Z_{temp} = 0; b = 0;$ 
2:  $N_i =$  Set of nodes of group  $i, i \in M;$ 
3:  $R_i =$  Path of  $MDC_i, i \in M;$ 
4: for each  $i \in M$  do
5:    $R_i = tspHeuristic(N_i);$            ▷ using twice around the tree heuristic, find  $R_i$  for  $MDC_i$ 
6:   Calculate  $d_i, T_i, E_i, P_i;$        ▷  $d_i$ : travel distance of  $MDC_i, T_i$ : travel time of  $MDC_i, E_i$ : energy
                                       consumption of  $MDC_i, P_i$ : residual energy of sensor
7: end for
8: while  $Z_{temp} \geq Z_{cur}$  do
9:    $Z_{temp} = Z_{cur}, R = R_{cur};$ 
10:   $b = b + 1$                                ▷  $b$ = no. of rounds
11:  while  $s \leq S$  do                       ▷  $S$ = no. of steps
12:     $G_i = \arg \max d_i;$            ▷ find group  $i$  from  $K$  groups with maximum travel distance of MDC
13:     $slant =$  borderline between  $G_i$  and  $G_{i-1}$    ▷  $G_{i-1}$ = clockwise adjacent group of  $G_i$ 
14:    if NNA to be used then
15:       $h = NNA(N_i, slant);$ 
16:    else if MPD to be used then
17:       $h = MPD(N_i, N_{i-1});$ 
18:    end if
19:     $N_i = N_i \setminus \{h\};$            ▷ remove  $h$  from  $N_i$ 
20:     $R_i = tspHeuristic(N_i);$        ▷ find new  $R_i$  excluding  $h$ 
21:    Calculate new  $d_i, T_i, E_i$  and  $P_i;$ 
22:     $N_{i-1} = N_{i-1} \cup \{h\};$      ▷ add  $h$  to  $N_{i-1}$ 
23:     $R_{i-1} = tspHeuristic(N_{i-1});$    ▷ find new  $R_{i-1}$  including  $h$  from  $N_i$ 
24:    Calculate new  $d_{i-1}, T_{i-1}, E_{i-1}$  and  $P_{i-1}.$ 
25:    Calculate  $Z = \sum_{i=1}^K d_i;$        ▷  $Z$ : total travel distance of  $K$  MDCs
26:  end while
27:   $R[b] =$  set of MDCs' paths with  $min(Z)$  and  $\{T_i \leq D \ \&\& \ E_i \leq Q \ \&\& \ P_i \geq P\};$    ▷ from  $S$  steps,
                                       path set for MDCs that leads to minimum  $Z$  and satisfy constraints
28:   $Z_{cur} = min(Z), R_{cur} = R[b]$ 
29:  if  $b = 1$  then
30:     $Z_{temp} = Z_{cur};$ 
31:  end if
32: end while
33: Return  $R$  as the output.

```

---

## References

1. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330.
2. Wu, X.; Chen, G.; Society, I.C.; Das, S.K. Avoiding energy holes in wireless sensor networks with nonuniform node distribution. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 710–720.
3. Mahmood, M.A.; Seah, W.K.; Welch, I. Reliability in wireless sensor networks: A survey and challenges ahead. *Comput. Netw.* **2015**, *79*, 166–187.

4. Ma, M.; Yang, Y.; Zhao, M. Tour Planning for Mobile Data-Gathering Mechanisms in Wireless Sensor Networks. *IEEE Trans. Veh. Technol.* **2013**, *62*, 1472–1483.
5. Kumar, A.; Sivalingam, K.M.; Kumar, A. On reducing delay in mobile data collection based wireless sensor networks. *Wirel. Netw.* **2013**, *19*, 285–299.
6. Xing, G.; Li, M.; Wang, T.; Jia, W.; Huang, J. Efficient Rendezvous Algorithms for Mobility-Enabled Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2012**, *11*, 47–60.
7. Kansal, A.; Somasundara, A.A.; Jea, D.D.; Srivastava, M.B.; Estrin, D. Intelligent fluid infrastructure for embedded networks. In Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services, MobiSys '04, Boston, MA, USA, 6–9 June 2004; pp. 111–124.
8. Sugihara, R.; Member, S.; Gupta, R.K. Optimal speed control of mobile node for data collection in sensor networks. *IEEE Trans. Mob. Comput.* **2010**, *9*, 127–139.
9. Somasundara, A.A.; Ramamoorthy, A.; Srivastava, M.B. Mobile Element Scheduling with Dynamic Deadlines. *IEEE Trans. Mob. Comput.* **2007**, *6*, 395–410.
10. Gu, Y.; Bozdog, D.; Ekici, E.; Özgüner, F.; Lee, C.G. Partitioning based mobile element scheduling in wireless sensor networks. In Proceedings of the Second Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks, SECON, Santa Clara, CA, USA, 26–29 September 2005; pp. 386–395.
11. Wang, Z.M.; Basagni, S.; Melachrinoudis, E.; Petrioli, C. Exploiting sink mobility for maximizing sensor networks lifetime. In Proceedings of the 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 6–9 January 2014.
12. Ghaleb, M.; Subramaniam, S.; Othman, M.; Zukarnain, Z. Predetermined path of mobile data gathering in wireless sensor networks based on network layout. *EURASIP J. Wirel. Commun. Netw.* **2014**, doi:10.1186/1687-1499-2014-51.
13. Van Le, D.; Oh, H.; Yoon, S. HiCoDG: A Hierarchical Data-Gathering Scheme Using Cooperative Multiple Mobile Elements. *Sensors* **2014**, *14*, 24278–24304.
14. Keung, G.; Bo, L.; Zhang, Q. The Intrusion Detection in Mobile Sensor Network. *IEEE/ACM Trans. Netw.* **2012**, *20*, 1152–1161.
15. Aslanyan, H.; Leone, P.; Rolim, J. Data Propagation with Guaranteed Delivery for Mobile Networks. In *Experimental Algorithms*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 386–397.
16. Jea, D.; Somasundara, A.; Srivastava, M., Multiple controlled mobile elements (data mules) for data collection in sensor networks. In Proceedings of the First IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2005, Marina del Rey, CA, USA, 30 June–1 July 2005; pp. 244–257.
17. Liu, L.; Zhang, R.; Chua, K.C. Wireless information transfer with opportunistic energy harvesting. In Proceedings of the 2012 IEEE International Symposium on Information Theory, Cambridge, MA, USA, 1–6 July 2012; pp. 950–954.
18. Jabbar, H.; Song, Y.S.; Jeong, T.T. RF Energy Harvesting System and Circuits for Charging of Mobile Devices. *IEEE Trans. Consum. Electron.* **2010**, *56*, 247–253.
19. Mekikis, P.V.; Antonopoulos, A.; Kartsakli, E.; Lalos, A.S.; Alonso, L.; Verikoukis, C. Information Exchange in Randomly Deployed Dense WSNs With Wireless Energy Harvesting Capabilities. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 3008–3018.
20. Mekikis, P.V.; Lalos, A.S.; Antonopoulos, A.; Alonso, L.; Verikoukis, C. Wireless Energy Harvesting in Two-Way Network Coded Cooperative Communications: A Stochastic Approach for Large Scale Networks. *IEEE Commun. Lett.* **2014**, *18*, 1011–1014.
21. Mekikis, P.V.; Kartsakli, E.; Antonopoulos, A.; Alonso, L.; Verikoukis, C. Connectivity analysis in wireless-powered sensor networks with battery-less devices. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016.
22. Parzysz, F.; Ntontin, K.; Kalampoukas, K.; Verikoukis, C. Dedicated RF power transfer for wirelessly-powered wearable medical sensors. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016.
23. Garnica, J.; Chinga, R.A.; Lin, J. Wireless Power Transmission: From Far Field to Near Field. *Proc. IEEE* **2013**, *101*, 1321–1331.
24. Kurs, A.; Karalis, A.; Moffatt, R.; Joannopoulos, J.D.; Fisher, P.; Soljacic, M. Wireless Power Transfer via Strongly Coupled Magnetic Resonances. *Science* **2007**, *317*, 83–86.

25. Xie, L.; Shi, Y.; Hou, Y.T.; Sherali, H.D. Making Sensor Networks Immortal: An Energy-Renewal Approach With Wireless Power Transfer. *IEEE/ACM Trans. Netw.* **2012**, *20*, 1748–1761.
26. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless Sensor Networks: A Survey. *Comput. Netw.* **2002**, *38*, 393–422.
27. Peng, Y.; Li, Z.; Zhang, W.; Qiao, D. Prolonging sensor network lifetime through wireless charging. In Proceedings of the 2010 IEEE 31st Real-Time Systems Symposium (RTSS), San Diego, CA, USA, 30 November–3 December 2010; pp. 129–139.
28. Pan, M.; Li, H.; Pang, Y.; Yu, R.; Lu, Z.; Li, W. Optimal energy replenishment and data collection in wireless rechargeable sensor networks. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 125–130.
29. Pang, Y.; Zhang, Y.; Gu, Y.; Pan, M.; Han, Z.; Li, P. Efficient data collection for wireless rechargeable sensor clusters in harsh terrains Using UAVs. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 234–239.
30. Toth, P.; Vigo, D. (Eds.) *The Vehicle Routing Problem*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001.
31. Solomon, M.M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Oper. Res.* **1987**, *35*, 254–265.
32. Liu, H.; Chu, X.; Leung, Y.W.; Du, R. Minimum-Cost Sensor Placement for Required Lifetime in Wireless Sensor-Target Surveillance Networks. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 1783–1796.
33. Pataki, G. Teaching Integer Programming Formulations Using the Traveling Salesman Problem. *SIAM Rev.* **2003**, *45*, 116–123.
34. Laporte, G. The traveling salesman problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **1992**, *59*, 231–247.
35. Kindervater, G.; Savelsbergh, M. Vehicle Routing: Handling Edge Exchanges. In *Local Search in Combinatorial Optimization*; John Wiley & Sons, Inc.: New York, NY, USA, 1997; pp. 337–360.
36. Fisher, M.L.; Jaikumar, R. A generalized assignment heuristic for vehicle routing. *Networks* **1981**, *11*, 109–124.
37. Wang, Y.C.; Peng, W.C.; Tseng, Y.C. Energy-Balanced Dispatch of Mobile Sensors in a Hybrid Wireless Sensor Network. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *21*, 1836–1850.
38. GLPK (GNU Linear Programming Kit). Available online: <https://www.gnu.org/software/glpk/> (accessed on 11 March 2016).



© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).