

Article

# Efficient Wireless Charger Deployment for Wireless Rechargeable Sensor Networks

Jehn-Ruey Jiang \* and Ji-Hau Liao

Department of Computer Science and Information, National Central University, Taoyuan 32001, Taiwan;  
jigoxd@gmail.com

\* Correspondence: jrjiang@csie.ncu.edu.tw; Tel.: +886-3-422-7151

Academic Editor: Chang Wu Yu

Received: 11 June 2016; Accepted: 22 August 2016; Published: 31 August 2016

**Abstract:** A wireless rechargeable sensor network (WRSN) consists of sensor nodes that can harvest energy emitted from wireless chargers for refilling their batteries so that the WRSN can operate sustainably. This paper assumes wireless chargers are equipped with directional antennas, and are deployed on grid points of a fixed height to propose two heuristic algorithms solving the following wireless charger deployment optimization (WCDO) problem: how to deploy as few as possible chargers to make the WRSN sustainable. Both algorithms model the charging space of chargers as a cone and calculate charging efficiency according power regression expressions complying with the Friis transmission equation. The two algorithms are the greedy cone covering (GCC) algorithm and the adaptive cone covering (ACC) algorithm. The GCC (respectively, ACC) algorithm greedily (respectively, adaptively) generates candidate cones to cover as many as possible sensor nodes. Both algorithms then greedily select the fewest number of candidate cones, each of which corresponds to the deployment of a charger, to have approximate solutions to the WCDO problem. We perform experiments, conduct simulations and do analyses for the algorithms to compare them in terms of the time complexity, the number of chargers deployed, and the execution time.

**Keywords:** wireless rechargeable sensor network (WRSN); energy harvesting; charger deployment; sustainability; greedy algorithms; directional antennas; Friis transmission equation

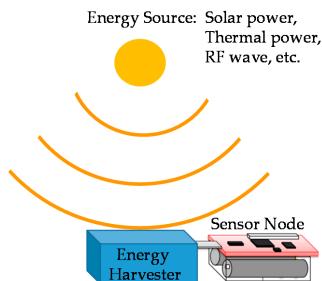
---

## 1. Introduction

A wireless rechargeable sensor network (WRSN) [1] consists of many rechargeable sensor nodes and one or more sink nodes. Sensor nodes are powered by batteries and can sense physical phenomena (e.g., temperature, humidity, atmospheric pressure, light intensity, and electromagnetic field strength) and forward the sensed data to the sink node(s) via multi-hop wireless communications. WRSNs are sustainable since they utilize energy harvesting technologies to replenish the batteries of sensor nodes. The replenishing is achieved by equipping every sensor node with an energy harvester to harvest energy from different sources [2,3], such as solar power, thermal power, radio frequency (RF) waves, or air flow, as shown in Figure 1. Many recent studies [4–13] focus on the WRSNs due to their sustainability. Some other studies [14–20] investigate battery-less or battery-free wireless sensor networks (WSNs) in which sensor nodes are powered by capacitors storing energy harvested through wireless power transmission (WPT), especially microwave power transmission (MPT). Without loss of generality, such WSNs are regarded as WRSNs in this paper.

Energy harvesting technologies are generally divided into two classes: (1) charger-based energy harvesting and (2) ambient energy harvesting. The former technology deploys specific devices, called chargers, to emit energy, and utilizes harvesters to capture the energy. The latter technology applies ambient energy harvesting devices to capture energy from the ambient environment. Typical devices include solar panels, thermoelectric generators, and ambient RF harvesters to capture energies from TV,

Wi-Fi, and 3G/4G stations. It is more difficult for the latter to control the amount of harvested energy due to the unstable nature of ambient energy sources (ESs). This is problematic if sensor nodes have to meet specific energy requirements. Therefore, this paper pays attention to WRSNs using charger-based energy harvesting technologies. It practically focuses on indoor WRSNs using the RF wireless energy harvesting technology.



**Figure 1.** A wireless rechargeable sensor network (WRSN) sensor node equipped with an energy harvester to harvest energy. RF: radio frequency.

The deployment of “RF wireless chargers” (or “chargers” for short) is challenging, since every harvester needs to be covered by at least one charger and the charging space of chargers is usually limited. For example, the effective charging distance is 3–5 m and the effective charging angle is 30°–45° between the Powercast TX91501-3W-ID charger (Powercast Corp., Pittsburgh, PA, USA) [21] and the Powercast P2110-EVAL-02 harvester (Powercast Corp.) [22]. It is also a cost-consuming task, as wireless chargers are expensive. For example, the Powercast TX91501-3W-ID charger currently costs about 1,000 US dollars. This motivated the authors to study the wireless charger deployment optimization (WCDO) problem defined below. Given a set of WRSN sensor nodes equipped with harvesters, the WCDO addresses how to deploy as few as possible chargers to cover all sensor nodes for satisfying their energy requirements to make the WRSN sustainable. To the best of our knowledge, no research has addressed the same problem up to now.

This paper proposes two algorithms for solving the WCDO problem. It is assumed that wireless chargers are equipped with directional antennas whose charging space is modeled as a cone and charging efficiency complies with the Friis free space transmission equation [23]. It is also assumed that chargers are deployed on grid points at a specific height and the sensor nodes equipped with wireless harvesters are distributed beneath the chargers on the floor or object surfaces. The proposed algorithms are the greedy cone covering (GCC) algorithm and the adaptive cone covering (ACC) algorithm. Both algorithms treat chargers as cones. The GCC (respectively, ACC) algorithm greedily (respectively, adaptively) generates candidate cones that can cover as many as possible sensor nodes. The two algorithms then greedily select the fewest number of candidate cones, each of which corresponds to the deployment of a charger, to derive approximate solutions to the WCDO problem. We perform extensive experiments to justify our assumptions. We also conduct comprehensive simulations and do analyses for the algorithms to compare them in terms of the time complexity, the number of chargers deployed, and the execution time.

The remainder of the paper is organized as follows: some related studies are reviewed in Section 2. Section 3 presents the modeling and the problem definition. In Section 4, the two proposed algorithms to solve the problem are described. Section 5 shows algorithm time complexity analyses and grid point separation analyses for deploying chargers. Experiments about the charging efficiency and results of algorithm simulations are described in Section 6. Finally, Section 7 concludes the paper.

## 2. Related Work

In this section, we review the research results [3–8,13–17] most related to our work. Kim et al. [3] investigated various ambient energy-harvesting technologies using different power sources, such

as solar power, thermal power, piezoelectric power, and wireless RF power, in the aspect of their applicability for self-sustainable wireless sensor platforms. They stated that although ambient RF energy has low power density values of  $0.2 \text{ nW/cm}^2$ – $1 \mu\text{W/cm}^2$  and low radio frequency-direct current (RF-DC) conversion values of 10%–30%, a larger amount of total available power can be harvested by utilizing high gain antennas. Some antennas designed by the authors were shown [3]. For example, a folded dual-band antenna at 915 MHz and 2.45 GHz was designed to harvest RF power from cellular and WiFi sources with power densities about  $1 \mu\text{W/cm}^2$ . Moreover, a folded dipole antenna was designed to harvest RF power from a two-way radio at the UHF band (464 and 468 MHz). They also implemented a prototype of an embedded microcontroller-enabled sensor platform successfully powered by an ambient UHF digital TV signal (512–566 MHz) from a broadcasting antenna that is 6.3 km away.

Fu et al. [4] proposed a moving and stopping plan for mobile chargers to guarantee that the time to charge all given stationary sensor nodes is minimized. An optimal solution using linear programming for the plan is proposed; however, the computational complexity of the solution is very high. Therefore, a heuristic solution discretizing the charging power on a two-dimension space was proposed to reduce the time complexity. The heuristic solution has been proved to have a bounded approximation ratio.

He et al. [5] proposed two types of energy provisioning problems for WRSNs: point provisioning and path provisioning. The former addresses how to use the least number of chargers to ensure that a static sensor node placed in any position of the network will receive a sufficient recharge rate to ensure WRSN sustainability. The latter addresses the same problem for mobile sensor nodes under the assumption that sensor nodes can harvest excess energy in power-rich areas and store the energy for later use in power-deficient areas. The authors utilized the triangular deployment of chargers to solve the two problems. They also proposed a modified Friis free space transmission equation adjusted for short distance transmission for evaluating the deployment. The original Friis equation and the modified version are shown below in Equations (1) and (2), respectively:

$$P_r = G_t G_r \left( \frac{\lambda}{4\pi D} \right)^2 P_t \quad (1)$$

$$P_r = \frac{G_t G_r \eta}{L_p} \left( \frac{\lambda}{4\pi(D + \beta)} \right)^2 P_t \quad (2)$$

In Equations (1) and (2),  $P_t$  and  $G_t$  denote the power and the antenna gain of the transmitter (i.e., charger), respectively, and  $P_r$  and  $G_r$  denote the power and the antenna gain of the receiver (i.e., harvester), respectively. Furthermore,  $\lambda$  is the wave length, and  $D$  is the distance between the antennas of the charger and the harvester.  $\eta$  is the rectifier efficiency,  $L_p$  is polarization loss, and  $\beta$  is a parameter to adjust the original Friis equation for short distance transmission.

Dai et al. [6] stated that if the constraints of sensor node speed and battery capacity are considered, the sustainable operation of sensor nodes may not be guaranteed. It proposed the concept of quality of energy provisioning (QoEP) to characterize the expected portion of time that a sensor node can sustain operation by taking into account node speed and battery capacity. The upper and lower bounds of QoEP in 1D and 2D sensor node mobility cases are derived.

Xu et al. [7] considered the scheduling problem of arranging multiple mobile chargers to collaboratively charge sensor nodes with the goal of minimizing the total traveling distance of all chargers. This can be formulated as finding multiple travelling salesperson problem (TSP) closed tours covering all sensor nodes such that the total length of the tours is minimized. The authors proposed a 2-approximation algorithm to solve the problem under the assumption that the energy consumption rates of all sensor nodes are fixed.

Liang et al. [8] considered a scheduling problem similar to that addresses by Xu et al. in [6]. The problem is how to use the minimum number of mobile chargers to charge sensor nodes so that none of sensor nodes runs out of energy, subject to the energy capacity of mobile chargers. The authors designed a 5-approximation algorithm to find multiple closed tours. Afterwards, by bounding the

total distance of each tour, the authors designed another approximation algorithm invoking the 5-approximation algorithm to provide approximate solutions to the problem.

Ye et al. [9] introduced a new metric, the charging utility, to measure the charging quality of the charger with the considerations of both the fairness of charging sensor nodes and the number of sensor nodes charged by mobile chargers traveling along closed tours. It is assumed that each mobile charger can only travel a limited length per tour, due to its limited energy capacity. The authors formulated an optimization problem of scheduling a mobile charger to charge sensor nodes, with an objective of maximizing the charging utility, subject to the total traveling distance of the mobile charger per tour and the deadline of each sensor node to be charged. An approximation algorithm with a guaranteed approximation ratio was proposed to solve the problem.

Maehara et al. [14] proposed a WPT scheme called multi-point carrier shift diversity (MPCSD), in which the center frequency of the carrier of each power transmitting point (or transmitter) is slightly shifted with a predefined amount to create artificial fading. Suppose there are  $N$  different transmitters, the carrier frequency  $f_n$  of the  $n$ th transmitter is defined in Equation (3), shown below:

$$f_n = f_c - \frac{B}{2} + \frac{(2n - 1)B}{2N} \quad (3)$$

In Equation (3),  $f_c$  is the center frequency,  $B$  is the channel bandwidth, and  $n = 1, \dots, N$ . The proposed scheme is shown to be capable of mitigating power attenuation caused by the path loss effect and the standing-wave effect created by multipath and interference between multiple power transmitters so that the coverage of energy supply field is improved. Maehara et al. [15] investigated the energy transmission coverage of the MPCSD scheme through theoretical discussions and practical experiments. It is shown that the MPCSD scheme can achieve 100% of coverage in an indoor environment for the sensor node duty factor of 1/100 (i.e., the sensor node is in the transmitting mode for the duration of 10 ms per second).

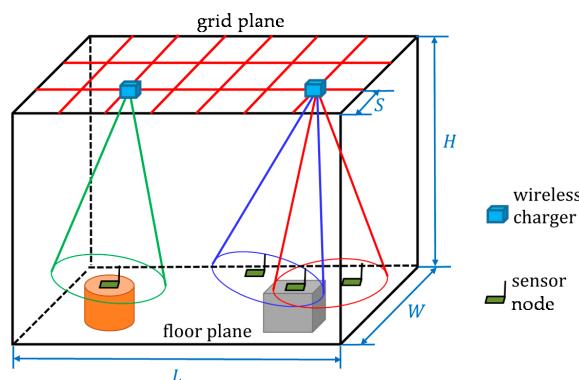
Imoto et al. [16] conducted experiments in an IEEE 802.11g-based wireless local area network (WLAN) in which an ES transmits microwave power to a sensor station (SS) using the carrier sense multiple access with collision avoidance (CSMA/CA) protocol to transmit data to an access point (AP). They derived the following two requirements for coexistence of MPT and the WLAN wireless data transmission (WDT). First, MPT and WDT should not proceed simultaneously. Therefore, the ES should share the timing information of MPT with the SS so that the SS attempts to transmit data frames only when the ES does not transmit microwave power. Second, the ES should not transmit microwave power to the SS when the AP broadcasts beacon frames so that the SS is not disassociated from the AP. This can also prevent data throughput degradation due to WLAN automatic data rate selection.

Yamashita et al. [17] assumed the same environment as assumed by Imoto et al. [16] and designed and implemented a scheduling scheme to solve the disassociation problem and the throughput degradation problem caused by the coexistence of MPT and WDT. In the scheduling scheme, the SS goes to the active mode to transmit data towards and to receive delivery traffic indication map (DTIM) beacons from the AP periodically. It goes to the power-saving (sleep) mode for most of the time to save energy. As long as the SS is in the sleep mode, the ES can transmit microwave power. On the basis of link budget analysis, the authors also clarified the maximum distance between the EC and the SS to conclude that the scheduling scheme can be applied to a closed space or room.

Yoshida et al. [18] designed an aerospace WSN that allow the coexistence of high-power MPT and WDT. The aerospace WSN is used in the aircraft application of the reusable vehicle test (RVT) or space capsule for health and safety monitoring inside the RVT. The WSN is based on the time-division operation and the frequency-division operation. The former operation uses a control function to schedule MPT and WDT so that MPT is not activated when WDT is activated. The latter operation applies different frequency bands for WPT and WDT to avoid the interference between them. Through a practical experiment lasting 120 min, it is shown that the aerospace WSN can achieve sustainable balance between harvested power and consumed power.

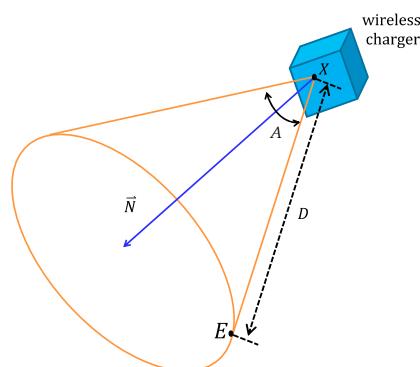
### 3. Modeling and Problem Definition

Sensor nodes in the WRSN are assumed to be distributed in a cuboid with the length  $L$ , width  $W$  and height  $H$ ; they have different energy requirements and can be located on the floor plane or object surfaces. The energy requirements and locations are summed to be known in advance and fixed. Wireless chargers equipped with directional antennas are assumed to be deployed at grid points on the grid plane or the deployment plane that is with height  $H$  and parallel to the floor plane. Moreover, the grid points are with the separation  $S$ ; that is, the nearest grid points are of the distance  $S$ . Please refer to Figure 2 for the illustration of the distribution of sensor nodes and the deployment of wireless chargers. Note that we assume no more than  $Q$ ,  $Q \geq 1$ , chargers can be deployed at a same grid point.



**Figure 2.** The schematic view of the WRSN deployment cuboid.

The effective charging space of wireless chargers is modeled as a cone, called a charger cone. Note that the two terms “charger” and “cone” are used interchangeably later in this paper. The modeling is based on the fact that direction antennas usually have specific beamwidth due to vertical and horizontal polarization. For example, the direction antenna of the Powercast TX91501-3W-ID charger has  $60^\circ$  beamwidth in both the horizontal direction and the vertical direction. As shown in Figure 3, every charger cone is characterized by an apex  $X$ , a normal vector  $\vec{N}$  whose direction is parallel to the symmetrical axis of the cone, an effective charging distance (i.e., the cone generatrix length)  $D$ , and an effective charging angle (i.e., the opening angle)  $A$ . When a sensor node is within the charger cone of a charger, we assume the sensor node can be charged effectively by the charger; otherwise, the sensor node cannot be charged effectively. Note that the point  $E$  is regarded as an extreme point within the charger cone, where the distance between  $E$  and  $X$  is  $D$  and the angle between the normal vector  $\vec{N}$  and the vector going from  $X$  to  $E$  is  $A/2$ .



**Figure 3.** A charger cone and its parameters.

The famous Friis transmission equation [23], shown in Equation (1), is used to model the energy received by the harvester. By the equation, we can see that the received energy  $P_r$  is inversely proportional to  $D^2$ , that is,  $P_r = \alpha D^{-2}$ , where  $\alpha$  is a constant and  $D$  is the distance between the charger and the harvester. This paper utilizes a more general empirical model  $P_r = \alpha D^{-\beta}$ , where  $\beta > 0$ , to estimate the energy received by the harvester. Later in Section 5, we will show how to apply the power regression analysis to derive appropriate  $\alpha$  and  $\beta$  values for the model to fit experimental data accurately.

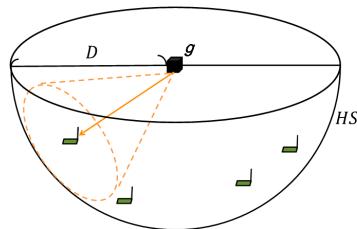
With all the assumptions mentioned above, the WCDO problem deals with how to deploy the minimum number of wireless chargers to cover all sensor nodes and fulfil their energy demands. To be more precise, given a set of sensor nodes distributed in a cuboid and a set of grid points, the WCDO problem addresses how to deploy as few as possible chargers at grid points to charge sensor nodes to make them sustainable under the assumption that the effective charging space of chargers is a cone. A special case of the WCDO problem in which every charger's charging space is a half sphere (instead of a cone) and every sensor node needs to be covered by only one charger can be solved by the reduction to the NP-hard set covering (SC) problem [24]. It is believed the WCDO problem is also NP-hard. Nevertheless, the NP-hardness of the WCDO problem has not yet proven.

#### 4. Proposed Algorithms

Two algorithms, namely the GCC algorithm and the ACC algorithm, are proposed to solve the WCDO problem efficiently. The algorithms have different time complexity and achieve different approximate solutions to the WCDO problem. The details of the two algorithms are described below.

For a WSRN with  $n$  sensor nodes and with chargers being deployed on some of  $m$  grid points, the set of sensor nodes is denoted by  $NS = \{s_1, s_2, \dots, s_n\}$  and the set of grid points is denoted by  $GS = \{g_1, g_2, \dots, g_m\}$ , where  $m = (\left\lfloor \frac{L}{S} \right\rfloor + 1) \times (\left\lfloor \frac{W}{S} \right\rfloor + 1)$ ,  $S$  is the separation of grid points, and  $L$ ,  $W$  and  $H$  are the length, width, and height of the sensor node deployment cuboid, respectively. Note that we assume each sensor node  $s_i$  in  $NS = \{s_1, s_2, \dots, s_n\}$  has energy requirement  $e_i$  in  $ES = \{e_1, e_2, \dots, e_n\}$ . There are two forms of energy requirements: coverage requirement and power requirement. For the coverage requirement,  $e_i$  is a positive integer  $1, 2, \dots$ , indicating sensor node  $s_i$  needs the coverage of  $e_i$  chargers to fulfill its energy demand. For the power requirement,  $e_i$  is of the unit mW, indicating  $s_i$  needs to capture  $e_i$  mW of power to make itself sustainable. The coverage requirement is actually about extreme cases. Specifically, a node's coverage requirement can be derived by calculating the ratio of the node's power requirement over the power harvested by the extreme point  $E$ , as indicated in Figure 3.

Both algorithms have two phases. The first one is the cone generation phase to generate candidate cones to be selected; the second one is the cone selection phase to select some candidate cones for deploying chargers. In the first phase, the algorithms generate cones by taking every grid point as the apex of cones. As shown in Figure 4, for every grid point, the algorithms first generate a half sphere ( $HS$ ) centered at the point and calculate the sensor set ( $SS$ ) of sensor nodes covered by  $HS$  (i.e., within  $HS$ ). Afterwards, the first algorithm generates candidate cones on a node-pair-by-node-pair basis (i.e., for every pair of nodes in  $SS$ ), while the second algorithm, on a node-by-node basis (i.e., for every node in  $SS$ ). For generating better candidate cones, the algorithms greedily adjust cones so that every single cone can cover as many as sensor nodes or can charge sensor nodes more efficiently.



**Figure 4.** The illustration of the lower half sphere  $HS$  centered at  $g$  with radius  $D$ .

In the second phase, both algorithms first unmark all sensor nodes. Note that a sensor node is unmarked if its energy requirement is not satisfied; otherwise, it will be marked. The algorithms then run iteration by iteration, and greedily select the qualified cone covering most unmarked sensor nodes, where a cone is said to be qualified if it takes a grid point  $g$  as its apex and  $g$  is totally taken by no more than  $Q$  selected cones as their apex. A selected cone corresponds to a deployed charger. For every newly selected cone, the algorithms marks every sensor node whose energy requirement is satisfied by the cone. And the algorithms continue until all sensor nodes are marked. In this way, the WCDO problem can be solved near-optimally.

The pseudo code of the GCC algorithm is shown in Figure 5. In the code,  $\vec{u}_x$  stands for a vector going from grid point  $g$  to sensor node  $s_{a_x}$ , and  $Cone(\vec{u}_x)$  stands for a cone which takes  $g$  as the apex, takes  $\vec{u}_x$  as the symmetrical axis (or the *direction*), and has an effective charging distance  $D$  and an effective charging angle  $A$ . Also note that  $|Cone(\vec{u}_x)|$  stands for the number of unmarked sensor nodes in  $NS$  covered by  $Cone(\vec{u}_x)$ . Moreover,  $\psi(\vec{u}_x, \vec{u}_y)$  stands for the angle between the two vectors  $\vec{u}_x$  and  $\vec{u}_y$ .

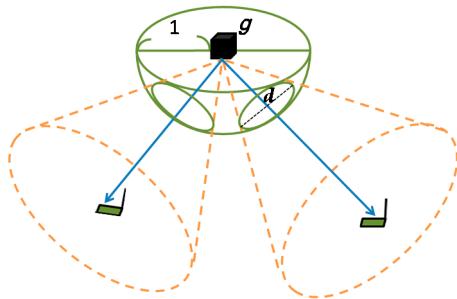
Algorithm: GCC (Greedy Cone Covering)	
Input: $NS = \{s_1, s_2, \dots, s_n\}, ES = \{e_1, e_2, \dots, e_n\}, GS = \{g_1, g_2, \dots, g_m\}$	
Output: $CS^*, US$	
1.	<b>unmark</b> all nodes in $NS$
2.	$CS^* \leftarrow CS \leftarrow \emptyset$ // $CS^*$ and $CS$ are cone sets
3.	<b>for</b> every grid point $g \in GS$ <b>do</b>
4.	$HS \leftarrow$ a lower half sphere centered at $g$ with radius $D$
5.	<b>if</b> ( $HS$ covers only node $s_{a_1}$ ) <b>then</b>
6.	<b>generate</b> $\vec{u}_1 \leftarrow \vec{gs}_{a_1}$
7.	<b>add</b> $Cone(\vec{u}_1)$ into $CS$
8.	<b>else if</b> ( $HS$ covers $k, k > 1$ , nodes $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ ) <b>then</b>
9.	<b>generate</b> $\vec{u}_1 \leftarrow \frac{\vec{gs}_{a_1}}{\ \vec{gs}_{a_1}\ }, \vec{u}_2 \leftarrow \frac{\vec{gs}_{a_2}}{\ \vec{gs}_{a_2}\ }, \dots, \vec{u}_k \leftarrow \frac{\vec{gs}_{a_k}}{\ \vec{gs}_{a_k}\ }$
10.	<b>for</b> $x \leftarrow 1$ to $k$ <b>do</b>
11.	<b>for</b> $y \leftarrow 1$ to $k$ <b>do</b>
12.	<b>if</b> ( $\psi(\vec{u}_x, \vec{u}_y) = A$ ) <b>then</b>
13.	<b>add</b> into $CS$ the cone indicated in Figure 7 (a)
14.	<b>else if</b> ( $\psi(\vec{u}_x, \vec{u}_y) > A$ ) <b>then</b>
15.	<b>add</b> into $CS$ the 2 cones indicated in Figure 7 (b)
16.	<b>else if</b> ( $\psi(\vec{u}_x, \vec{u}_y) < A \wedge x \neq y$ ) <b>then</b>
17.	<b>add</b> into $CS$ the 4 cones indicated in Figure 7 (c)
18.	<b>repeat</b>
19.	<b>select</b> qualified cone $c$ in $CS$ that covers the most unmarked nodes in $NS$
20.	$CS^* \leftarrow CS^* \cup \{c\}; CS \leftarrow CS - \{c\}$
21.	<b>for</b> every sensor node $s_i$ covered by $c$ <b>do</b>
22.	$e_i \leftarrow e_i - EP(c, s_i)$
23.	<b>if</b> ( $e_i \leq 0$ ) <b>then mark</b> $s_i$
24.	<b>until</b> ( (all nodes in $NS$ are marked) $\vee (CS = \emptyset)$ )
25.	$US \leftarrow$ the set of unmarked nodes in $NS$
26.	<b>return</b> $CS^*, US$

**Figure 5.** The greedy cone covering (GCC) algorithm.

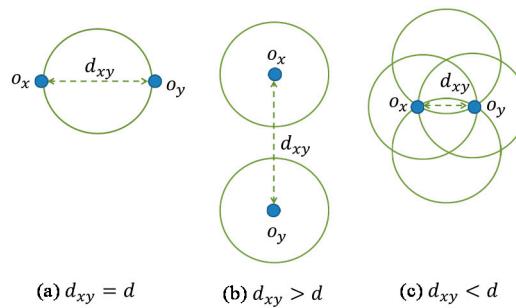
In the GCC algorithm, for every grid point  $g$ , a lower half sphere  $HS$  is obtained with  $g$  as the center and  $D$  as the radius. If  $HS$  covers only one sensor node  $s_{a_1}$ , then the algorithm generates one candidate cone taking the vector going from  $g$  to  $s_{a_1}$  as the direction. However, if  $HS$  covers  $k, k > 1$ , sensor nodes  $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ , then the algorithm generates  $k$  unit vectors  $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k$  going from  $g$  to  $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ . Afterwards, the GCC algorithm compares the opening angle  $A$  with the angle  $B$  (i.e.,

$\psi(\vec{u}_x, \vec{u}_y)$ ) between any two vectors  $\vec{u}_x$  and  $\vec{u}_y$  associated with the two distinct sensor nodes  $s_{a_x}$  and  $s_{a_y}$ . There are three cases for the comparison:  $B = A$ ,  $B > A$ , and  $B < A$ .

The GCC algorithm performs the aforementioned angle comparison by projecting every vector and every cone onto the surface of a unit sphere centered at grid point  $g$ . As shown in Figure 6, the projection of a vector is a point, while the projection of a cone is a circle of diameter  $d$ . Referring to Figure 7a–c, for any pair of distinct sensor nodes  $s_{a_x}$  and  $s_{a_y}$ , where  $1 \leq x, y \leq k$ , let  $o_x$  and  $o_y$  be the projection points of  $gs_{a_x}$  and  $gs_{a_y}$  on the unit sphere  $O$  centered at grid point  $g$ , and let  $d_{xy}$  be the Euclidean distance between the two points. There are three cases of the relationship between  $d_{xy}$  and  $d$ , which correspond to the three relationship cases of  $A$  and  $B$ . The three cases are elaborated below. Case (a)  $d_{xy} = d$ . This case corresponds to the case of  $B = A$ . For such a case, one candidate cone is generated. Note that  $o_x$  and  $o_y$  should be on the diameter of the projection circle of the cone. Case (b)  $d_{xy} > d$ . This case corresponds to the case of  $B > A$ . For such a case, two candidate cones taking  $gs_x$  and  $gs_y$  as directions are generated. Case (c)  $d_{xy} < d$ . This case corresponds to the case of  $B < A$ . For such a case, four candidate cones are generated. For two of the cones,  $o_x$  and  $o_y$  should be on the arcs of the projection circles (at the top and at the bottom) of the cones. For another cone,  $o_x$  should be on the projection circle (on the right), and the line  $\overline{o_x o_y}$  should go through the center of the circle. For yet another cone,  $o_y$  should be on the projection circle (on the left), and the line  $\overline{o_x o_y}$  should go through the center of the circle. Note that for Case (c), the GCC algorithm generates four candidate cones to cover sensor nodes  $s_{a_x}$  and  $s_{a_y}$ . In practice, it greedily adjusts the cones to the extreme limitation with the purpose that the cones may at the same time cover as many as possible other sensor nodes.



**Figure 6.** The projection of a cone (respectively, vector) onto a unit sphere surface is a circle (respectively, point).



**Figure 7.** Three cases for the GCC algorithm to generate candidate cones.

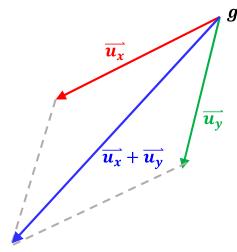
After generating candidate cones, the GCC algorithm greedily selects the qualified cone covering the most unmarked sensor nodes to deploy a charger. Note that if a sensor node's energy demand is met by the chargers already deployed, it is marked; otherwise, it is unmarked. In this way, chargers are deployed iteration by iteration until all sensor nodes are marked. Also note that we use function  $EP(c, s_i)$  to estimate the energy provision between the charger  $c$  and the sensor node  $s_i$ .

Corresponding to the coverage requirement,  $EP(w, s_i)$  returns 0 or 1, indicating the cone or the charger  $c$  can or cannot cover the sensor node  $s_i$ . Corresponding to the power requirement,  $EP(c, s_i)$  returns the power (in the unit of mW) received by  $s_i$  from the energy emitted by charger  $c$ . As will be shown in Section 5, the received power can be calculated by interpolating power regression expressions derived from practical experimental data.

The pseudo code of the ACC algorithm is shown in Figure 8. Like the GCC algorithm, the ACC algorithm first derives, for every grid point  $g$ , a lower half sphere  $HS$  centered at  $g$  with the radius  $D$ . If  $HS$  covers only one sensor node  $s_{a_1}$ , then the algorithm generates one candidate cone taking the vector going from  $g$  to  $s_{a_1}$  as the direction. If  $HS$  covers  $k, k > 1$ , sensor nodes  $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ , then the algorithm generates  $k$  unit vectors  $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k$  going from  $g$  to  $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ . The ACC algorithm constructs candidate cones on a node by node basis. In practice, the ACC algorithm initially constructs a candidate cone  $Cone(\vec{u}_x)$  to cover sensor node  $s_{a_x}$  for  $1 \leq x \leq k$ . Besides covering node  $s_{a_x}$ , the ACC algorithm adaptively and greedily adjusts  $\vec{u}_x$  so that  $Cone(\vec{u}_x)$  can cover as many as possible other sensor nodes. This is achieved by adjusting the direction of  $\vec{u}_x$  by composing  $\vec{u}_x$  and  $\vec{u}_y$  for every vector  $\vec{u}_y$ , where  $1 \leq y \leq k$  and  $x \neq y$  (Figure 9). The direction adjustment (Lines 12–14) proceeds on a node-by-node basis. Certainly, the direction adjustment should guarantee that the number of sensor nodes covered by  $Cone(\vec{u}_x)$  is increasing and that the sensor node  $s_{a_x}$  is still covered by  $Cone(\vec{u}_x)$  whatever the adjusted  $\vec{u}_x$  is.

Algorithm: ACC (Adaptive Cone Covering)	
Input:	$NS = \{s_1, s_2, \dots, s_n\}, ES = \{e_1, e_2, \dots, e_n\}, GS = \{g_1, g_2, \dots, g_m\}$
Output:	$CS^*, US$
1.	<b>unmark</b> all nodes in $NS$
2.	$CS^* \leftarrow CS \leftarrow \emptyset$ // $CS^*$ and $CS$ are cone sets
3.	<b>for</b> every grid point $g \in GS$ <b>do</b>
4.	$HS \leftarrow$ a lower half sphere centered at $g$ with radius $D$
5.	<b>if</b> ( $HS$ covers only node $s_{a_1}$ ) <b>then</b>
6.	<b>generate</b> $\vec{u}_1 \leftarrow \vec{gs}_{a_1}$
7.	<b>add</b> $Cone(\vec{u}_1)$ into $CS$
8.	<b>else if</b> ( $HS$ covers $k, k > 1$ , nodes $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ ) <b>then</b>
9.	<b>generate</b> $\vec{u}_1 \leftarrow \frac{\vec{gs}_{a_1}}{\ \vec{gs}_{a_1}\ }, \vec{u}_2 \leftarrow \frac{\vec{gs}_{a_2}}{\ \vec{gs}_{a_2}\ }, \dots, \vec{u}_k \leftarrow \frac{\vec{gs}_{a_k}}{\ \vec{gs}_{a_k}\ }$
10.	<b>for</b> $x \leftarrow 1$ to $k$ <b>do</b>
11.	<b>for</b> $y \leftarrow 1$ to $k$ <b>do</b>
12.	<b>if</b> (( $x \neq y$ ) $\wedge$ ( $s_{a_x} \in Cone(\vec{u}_x + \vec{u}_y)$ ) $\wedge$ ( $ Cone(\vec{u}_x + \vec{u}_y)  >  Cone(\vec{u}_x) $ )) <b>then</b>
13.	$\vec{u}_x \leftarrow \vec{u}_x + \vec{u}_y;$
14.	$\vec{u}_x \leftarrow \frac{\vec{u}_x}{\ \vec{u}_x\ }$
15.	<b>add</b> the $k$ cones $Cone(\vec{u}_1), \dots, Cone(\vec{u}_k)$ into $CS$
16.	<b>repeat</b>
17.	<b>select</b> a qualified cone $c$ in $CS$ that covers the most unmarked nodes in $NS$
18.	$CS^* \leftarrow CS^* \cup \{c\}; CS \leftarrow CS - \{c\}$
19.	<b>for</b> every sensor node $s_i$ covered by $c$ <b>do</b>
20.	$e_i \leftarrow e_i - EP(c, s_i)$
21.	<b>if</b> ( $e_i \leq 0$ ) <b>then mark</b> $s_i$
22.	<b>until</b> ((all nodes in $NS$ are marked) $\vee$ ( $CS = \emptyset$ ))
23.	$US \leftarrow$ the set of unmarked nodes in $NS$
24.	<b>return</b> $CS^*, US$

Figure 8. The adaptive cone covering (ACC) algorithm.



**Figure 9.** Illustration of candidate cone direction adjustment by composing two vectors.

In the GCC and the ACC algorithms,  $US$  stands for the set of unmarked sensor nodes with unsatisfied energy requirement. For most cases, the two algorithm stop and return  $CS^*$  and  $US$ , with  $US$  is empty. For such cases, the cones in  $CS^*$  can be used to deploy chargers to satisfy the energy requirements of all sensor nodes. On the contrary, when the algorithms terminate and return a non-empty set of  $US$ , they should re-run by setting  $NS = US$ , and the re-running should proceed until  $US$  is empty. The re-running is to satisfy the pending energy requirements of unmarked nodes. When  $US$  is empty, the requirements of all sensor nodes are satisfied and all the cones ever reported in  $CS^*$  by running and re-running the algorithms are the final selected cones for deploying chargers. However, the re-running algorithms will report an error if it finds that no available grid point exists, where an available grid point is a grid point that no more than  $Q$  selected cones take as their apex. For such an error condition, we may need to decrease  $S$ , the separation of grid points, to increase the total number of grid points for more possible locations for deploying chargers to satisfy sensor nodes' energy requirements.

## 5. Analysis

### 5.1. Algorithm Time Complexity Analysis

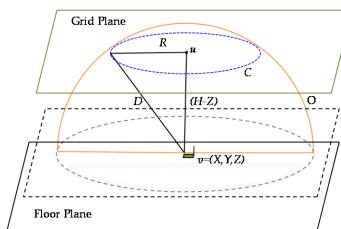
In this subsection, we analyze the time complexity of the two proposed algorithms. In the GCC algorithm, for every grid point  $g$ , a half sphere  $HS$  centered at  $g$  of radius  $D$  is formed (Lines 3–4). If  $S$  covers  $k$  sensor nodes, then candidate cones are generated on a node-pair-by-node-pair basis (Lines 9–17). There are up to  $4C_2^k$  candidate cones that can be generated, where  $k \leq n$ . Since there are  $m$  grid points, the number of candidate cones generated is then  $4mC_2^k \leq mn^2 = O(mn^2)$  and the time complexity to generate the candidate cones is also  $O(mn^2)$ . On each iteration in the repeat-until loop (Lines 18–24), the GCC algorithm first selects the cone  $c$  from  $CS$  that covers most unmarked nodes in  $NS$  (Line 19). Since there are at most  $mn^2$  cones in  $CS$ , and the algorithm should check if a cone can cover all nodes, the selection can be done in  $O(mn^3)$  time complexity. The GCC algorithm then checks if every unmarked node covered by cone  $c$  can be marked after updating its energy requirement (Lines 21–23). The checking takes  $O(n)$  time complexity. A cone can be selected only once in the repeat-until loop, so there are at most  $O(mn^2)$  iterations in the loop. In summary, there are  $O(mn^2)$  iterations in the repeat-until loop, and each iteration's time complexity is  $O(mn^3)$  (Line 19). Thus, the time complexity of the loop is  $O(m^2n^5)$ . Since the repeat-until loop has the highest time complexity in the algorithm, we thus have that the time complexity of the GCC algorithm is  $O(m^2n^5)$ .

The ACC algorithm forms a half sphere  $HS$  centered at  $g$  of radius  $D$  for every grid point  $g$  (Lines 3–4). If  $HS$  covers  $k$  sensor nodes, then  $k$  candidate cones will be generated,  $k \leq n$  (Line 8). Every candidate cone is tested for possible direction adjustment for  $k$  times (Lines 9–15), so there are  $k^2 = O(n^2)$  adjustment tests. Since there are  $m$  grid points, the ACC algorithm forms  $m$  half spheres. Let the half spheres cover  $k_1, k_2, \dots, k_m$  sensor nodes, respectively. Thus, the total number of candidate cones is  $\sum_{i=1}^m k_i \leq mn = O(mn)$  and the total number of adjustment tests is of the time complexity  $O(mn^3)$ . On each iteration in the repeat-until loop (Lines 16–22), the ACC algorithm first selects the cone  $c$  from  $CS$  that covers the most unmarked nodes in  $NS$  (Line 17). Since there are at most  $mn$  cones

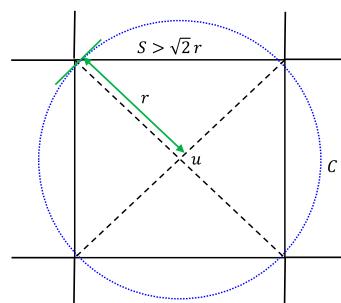
in  $CS$ , and the algorithm should check if a cone can cover all nodes, the selection (Line 17) can be done in  $O(mn^2)$  time complexity. The ACC algorithm then checks if every unmarked node covered by cone  $c$  can be marked after updating its energy requirement (Lines 19–21). The checking takes  $O(n)$  time complexity. A cone can be selected only once in the repeat-until loop, so there are at most  $O(mn)$  iterations in the loop. Thus, the time complexity of the repeat-until loop is  $O(m^2n^3)$ . Since the repeat-until loop has the highest time complexity in the ACC algorithm, the time complexity of the ACC algorithm is  $O(m^2n^3)$ .

## 5.2. Grid Point Separation Analysis

In this subsection, we analyze the relationship of  $S$ ,  $D$  and  $H$ , which are the grid point separation, the effective charging distance and high of the deployment plane, respectively. For an arbitrary sensor node located at position  $v = (X, Y, Z)$ , we can construct a sphere  $O$  with radius  $D$  (i.e., the charging distance). The intersection of  $O$  and the charger deployment plan (i.e., the grid plane)  $P$  is a circle  $C$  with radius  $R = \sqrt{D^2 - (H - Z)^2}$  centered at  $u$  (Figure 10). For any point  $u'$  within the area of  $C$ , we have  $\overline{vu'} \leq D$ . So, if there is at least one grid point  $g$  within  $C$ , the sensor node located at  $v$  can be covered by a charger deployed at  $g$ . When  $Z = 0$ ,  $v$  is on the floor plane and the circle  $C$  has the smallest radius  $r = \sqrt{D^2 - H^2}$ . As shown in Figure 11, if  $S > \sqrt{2} r$ , then there may be no grid point within the area of circle  $C$ . On the contrary, if  $S \leq \sqrt{2} r = \sqrt{2(D^2 - H^2)}$ , there is at least one grid point within the area of circle  $C$ . In summary,  $S \leq \sqrt{2(D^2 - H^2)}$  ensures that every sensor node is covered by at least one charger deployed at one of the grid points.



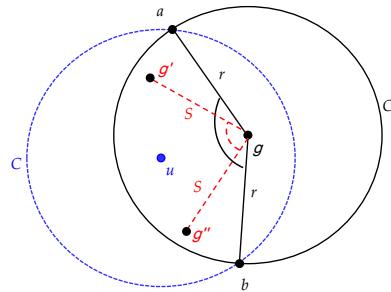
**Figure 10.** Illustration of the intersection of the grid plane with an upper half sphere of radius  $D$  (the charging distance) centered at a sensor node.



**Figure 11.** Illustration of no grid point within the area of circle  $C$  with radius  $r$ .

Below we show that if  $S \leq r = \sqrt{D^2 - H^2}$ , then there is at least two grid points within the area of circle  $C$  centered at  $u$  with radius  $r$ .  $S \leq r$  implies  $S \leq \sqrt{2} r$ , which in turns implies there is at least one grid point  $g$  within  $C$  by the fact just mentioned in the last paragraph. Let  $C'$  be the circle centered at  $g$  with radius  $r$  and let  $a$  and  $b$  be the intersection of  $C$  and  $C'$  (Figure 12). Since  $C$  and  $C'$  have the same radius, we have that the angel  $\angle agb$  associated with the minor arc  $\widehat{ab}$  should be larger than  $120^\circ$ ; otherwise,  $g$  will not be within  $C$ . For neighboring grid points  $g$ ,  $g'$  and  $g''$  that are apart from each other with separation  $S$ , their associated angle  $\angle g'gg''$  is  $90^\circ$ . We have that either  $g'$  or  $g''$  is within the sector associated with the minor arc  $\widehat{ab}$ . Therefore, there are at least two grid points within the area of

C. That is to say, there are at least two chargers covering a sensor node. In summary,  $S \leq \sqrt{(D^2 - H^2)}$  ensures that every sensor node is covered by at least two chargers deployed at two of the grid points.



**Figure 12.** Illustration of two circles both of the radius  $r$  intersecting at two points  $a$  and  $b$ .

For even smaller separation, we can use the “dies per wafer” equation in [25] to estimate approximately the number of grid points within circle  $C$  with radius  $r = \sqrt{D^2 - H^2}$ . The equation is shown below:

$$\text{Dies per wafer} = \frac{\pi \times (\text{Wafer Ridus})^2}{\text{Die area}} - \frac{2\pi \times (\text{Wafer Ridus})}{\sqrt{2 \times (\text{Die area})}} \quad (4)$$

If we take the grid square as the die area and the wafer radius as  $r$ , the radius of circle  $C$ , then we have the following equation for  $N_{GS}$ , the number of grid squares in circle  $C$ :

$$N_{GS} = \frac{\pi \times r^2}{S^2} - \frac{2\pi \times r}{S\sqrt{2}} \quad (5)$$

One grid square has four grid points. However, adjacent grid square share grid points. When we count the number of grid points, a grid square is associated with two non-redundant grid points. Therefore, we have the following equation for  $N_{GP}$ , the number of grid points in circle  $C$ :

$$N_{GP} = 2 \times N_{GS} = 2 \times \left( \frac{\pi \times r^2}{S^2} - \frac{2\pi \times r}{S\sqrt{2}} \right) \quad (6)$$

In summary, for grid separation  $S$  much smaller than  $r = \sqrt{D^2 - H^2}$ , each sensor node is covered by at least  $2 \times \left( \frac{\pi \times r^2}{S^2} - \frac{2\pi \times r}{S\sqrt{2}} \right)$  chargers approximately.

## 6. Experiments and Simulations

### 6.1. Experiments

In this subsection, we show results of extensive experiments in indoor environments for modeling the RF charging efficiency (i.e., the received energy or received power). The devices used in the experiments are Powercast TX91501-3W-ID wireless charger [21] and P2110-EVAL-02 power harvester [22], as shown in Figure 13. The former is a wireless charger with a patch directional antenna with 8 dBi gain and 60° beamwidth in both horizontal and vertical directions. It is a 915 MHz transmitter with 3 Watts equivalent isotropically radiated power (EIRP). The latter is a power harvester with a dipole omni-directional antenna with 1 dBi gain. It can harvest energy from RF waves, provide intermittent/pulsed power output up to 4.21 V, and connect to rechargeable batteries.

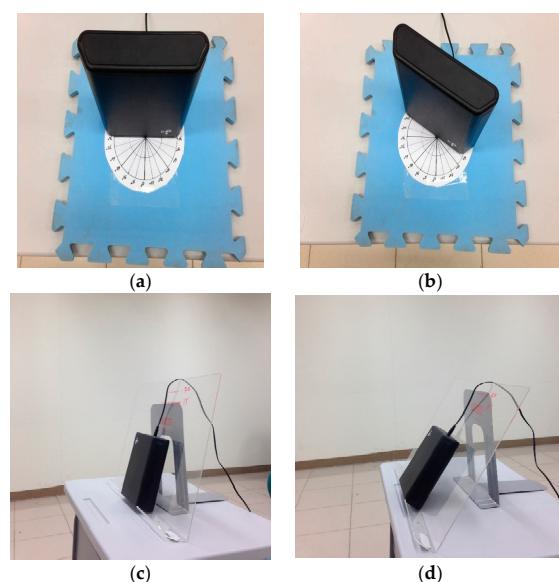


**Figure 13.** (a) Powercast TX91501-3W-ID wireless charger; and (b) P2110-EVAL-02 power harvester.

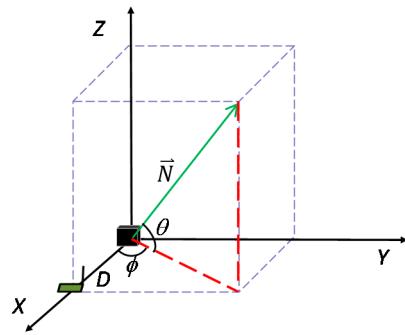
As shown in Figure 14, a wireless charger and a power harvester are set up. The distance  $D$  between the two devices is set to be 0.5 m, 1.0 m, . . . , or 4.5 m for experiments. As shown in Figure 15, the orientation of the charger's patch antenna is specified by two angles, the horizontal angle (azimuth)  $\phi$  and the vertical angle (elevation)  $\theta$ . Either angle is set to be  $0^\circ$ ,  $15^\circ$ , . . . , or  $90^\circ$ . Figure 16 illustrates the distance and the angle settings. In Figure 16, the charger is assumed to be at the origin and the harvester is assumed to be on the  $x$ -axis and be  $D$  apart from the charger. Moreover, the charger's antenna direction is indicated by vector  $N$  with the horizontal angle  $\Phi$  and the vertical angle  $\theta$  relative to the  $x$ -axis.



**Figure 14.** The experimental setting of a wireless charger and a power harvester.



**Figure 15.** Experimental settings for (a,b) different horizontal angles (azimuths); and (c,d) different vertical angles (elevations).



**Figure 16.** The illustration of distance  $D$ , horizontal angle  $\Phi$ , and vertical angle  $\theta$ .

By a multimeter, we measure the voltage ( $V$ ) and current ( $I$ ) of the output of the harvester 10 times for every distance  $D = 0.5$  m,  $1.0$  m, . . . , and  $4.5$  m, every horizontal angle  $\Phi = 0^\circ, 15^\circ, \dots, 90^\circ$ , and every vertical angle  $\theta = 0^\circ, 15^\circ, \dots, 90^\circ$ . We average the measured data and then apply the average data to derive the received power (i.e., received energy). Table 1 (respectively, Table 2) shows the experimental results of the received power in the unit of mW for different distances and horizontal (respectively, vertical) angles with the vertical (respectively, horizontal) angle is 0. Note that  $0.01$  V and  $0.01$  mA are the smallest measurable voltage and current, respectively. Therefore, the voltage values smaller than  $0.01$  V or the current value smaller than  $0.01$  mA are not measurable and such cases lead to “not available” (i.e., “-”) symbols shown in the Tables.

**Table 1.** Received power (mW) for different distances and horizontal angles.

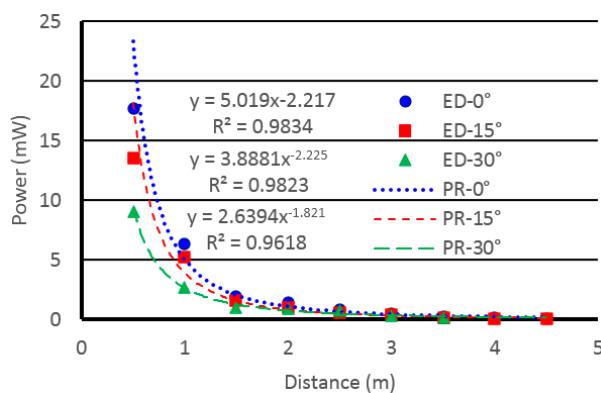
$D \setminus \Phi$	$0^\circ$	$15^\circ$	$30^\circ$	$45^\circ$	$60^\circ$	$75^\circ$	$90^\circ$
0.5 m	17.63	16.38	13.89	9.89	3.96	2.15	0.37
1.0 m	6.3	5.72	3.96	2.29	1.14	0.58	-
1.5 m	1.93	1.26	1	0.72	0.42	-	-
2.0 m	1.39	1.04	0.8	0.57	0.25	-	-
2.5 m	0.84	0.74	0.62	0.34	-	-	-
3.0 m	0.47	0.31	0.18	-	-	-	-
3.5 m	0.28	0.15	-	-	-	-	-
4.0 m	0.21	0.11	-	-	-	-	-
4.5 m	0.14	-	-	-	-	-	-

**Table 2.** Received power (mW) for different distances and vertical angles.

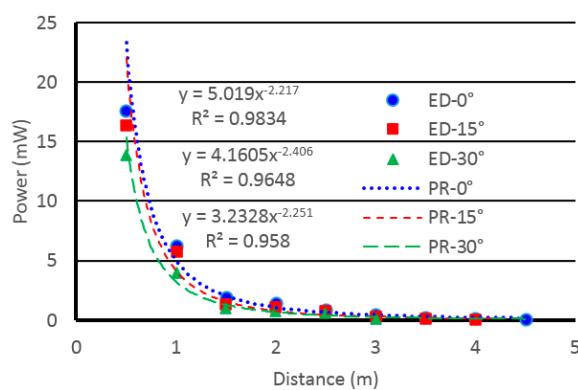
$D \setminus \theta$	$0^\circ$	$15^\circ$	$30^\circ$	$45^\circ$	$60^\circ$	$75^\circ$	$90^\circ$
0.5 m	17.63	13.52	9.07	5.55	2.78	0.92	0.26
1.0 m	6.3	5.23	2.65	1.26	0.79	0.41	-
1.5 m	1.93	1.52	1.02	0.56	0.29	-	-
2.0 m	1.39	0.95	0.87	0.34	0.18	-	-
2.5 m	0.84	0.61	0.76	0.28	-	-	-
3.0 m	0.47	0.42	0.36	0.15	-	-	-
3.5 m	0.28	0.21	0.19	-	-	-	-
4.0 m	0.21	0.14	-	-	-	-	-
4.5 m	0.14	0.12	-	-	-	-	-

By experimental data in Table 1, Table 2 and the specification [13] stating that the charger has a beamwidth of  $60^\circ$  in both horizontal and vertical directions, we model the charging space of a wireless charger as a cone with the opening angle of  $60^\circ$  and the generatrix length of  $D$ . We perform power regression on the data to obtain an exponential function taking  $D$  as the independent variable (i.e.,  $x$ ) and  $P$  as the dependent variable (i.e.,  $y$ ) for every vertical and horizontal angle of  $0^\circ, 15^\circ$

and  $30^\circ$ . For example, Figure 17 shows the functions derived by performing power regression analysis on experimental data of received energy (or power) for horizontal angle  $\Phi = 0^\circ, 15^\circ$  and  $30^\circ$ , and vertical angle  $\theta = 0^\circ$ . Moreover, Figure 18 shows the functions derived by performing power regression analysis on experimental data of received power for horizontal angle  $\Phi = 0^\circ$  and vertical angle  $\theta = 0^\circ, 15^\circ$  and  $30^\circ$ . We can see that the functions, namely  $y = 5.019x^{-2.217}$ ,  $y = 3.8881x^{-2.225}$ ,  $y = 2.6394x^{-1.821}$ ,  $y = 1.3522x^{-1.939}$ ,  $y = 0.7134x^{-2.01}$ , fit well with the experimental data, as they all have coefficient of determination (i.e.,  $R^2$ ) [26] values larger than 0.958, which is very close to the maximum value 1. They also comply with the Friis equation, since the function exponents are approximately -2. Therefore, we may well estimate the power received by the harvester with the functions. In practice, we interpolate the values of the functions to estimate the received power for given parameters, like distance  $D$ , horizontal angle  $\Phi$ , and vertical angle  $\theta$ .



**Figure 17.** The functions derived by performing power regression analysis on experimental data (ED) of the power received (PR) for horizontal angel  $\Phi = 0^\circ, 15^\circ$  and  $30^\circ$ , and vertical angle  $\theta = 0^\circ$ .



**Figure 18.** The functions derived by performing power regression analysis on ED of the PR for horizontal angel  $\Phi = 0^\circ$ , and vertical angle  $\theta = 0^\circ, 15^\circ$  and  $30^\circ$ .

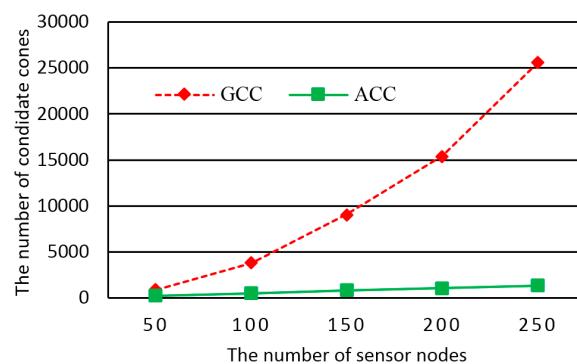
## 6.2. Simulations

This subsection shows the simulation results of the proposed GCC algorithm and ACC algorithm. The simulator is implemented in C++ language, and the simulation parameters are shown in Table 3. The sensors are distributed randomly in a  $20\text{ m} \times 15\text{ m} \times 2.3\text{ m}$  cuboid. The chargers can be deployed at grid points on the grid plane at the height of 2.3 m, and the grid point separation is 1.8 m. The charging distance is 3 m, and the charging cone opening angle is  $30^\circ$ . It is noted that 30 experiments were executed per simulation case for averaging the simulation results. The parameter settings are shown in Table 3.

We first perform simulations for three scenarios of coverage requirements: all sensor nodes require 1-, 2-, or 3-coverage of chargers. Figure 19 shows the comparisons of GCC and ACC algorithms in terms of the number of the candidate cones generated. The simulation results are all the same for the 1-, 2- and 3-coverage requirement scenarios, so only one diagram is shown in Figure 19. We can observe that GCC generates much more candidate cones than ACC. This is because GCC has  $O(mn^2)$  of candidate cones, while ACC has  $O(mn)$  of candidate cones, where  $n$  is the number of sensor nodes and  $m$  is the number of grid points.

**Table 3.** Simulation settings.

Item	Parameter
Sensor deployment cuboid	20 m ( $L$ ) $\times$ 15 m ( $W$ ) $\times$ 2.3 m ( $H$ )
Number of sensors	50, 100, 150, 200, 250
Effective charging distance	3 m ( $D$ )
Opening angle	30° ( $A$ )
Height of grid points	2.3 m ( $H$ )
Grid point separation	1.8 m ( $S$ )
Coverage requirement of sensor nodes	1-, 2-, or 3-coverage
Energy requirement of sensor nodes	0.18, 0.54, 0.9 mW
Times of simulation per case	30 times/case



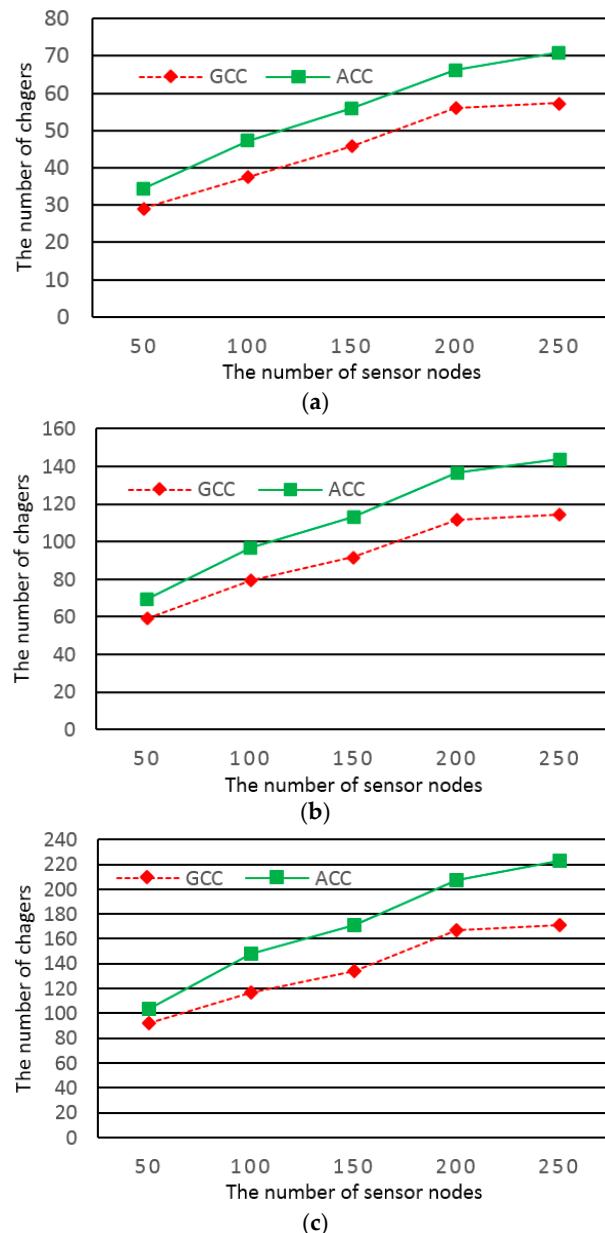
**Figure 19.** The comparisons of GCC and ACC in terms of the number of generated candidate cones.

Figure 20 shows the comparisons of GCC and ACC in terms of the number of the chargers deployed for the 1-, 2- and 3-coverage requirement scenarios. By Figure 20, we can observe that GCC outperforms ACC in terms of the number of chargers required. This is because GCC greedily adjusts charger antenna directions to the extreme for some cases to cover more sensor nodes by one candidate cone, and GCC generates more candidate cones to be selected. On the contrary, ACC adaptively adjusts antenna directions to cover sensor nodes with smaller horizontal angels and vertical angels, and ACC generates fewer candidate cones.

Figure 21 shows the comparisons of GCC and ACC in terms of execution time for scenarios of different coverage requirements. We can observe that ACC runs much faster than GCC, especially for the scenario of high energy consumption. This is because ACC has  $O(m^2n^3)$  time complexity, while GCC has  $O(m^2n^5)$  time complexity. However, the time complexity of the big  $O$  notation is just an upper bound. Some experimental cases may have sensor nodes whose coverage requirements are met very early. For such cases, the algorithm execution may not be too long. This accounts for why the execution times of algorithms do not grow according to the functions of  $m^2n^3$  or  $m^2n^5$ .

We now consider the form of power requirements, in which every sensor node present its power in the unit of mW for compensating its power consumption. For setting simulation parameters, we derive the sensor node power consumption from the specification of TI eZ430 (Texas Instruments Inc., Dallas, TX, USA) [27], which a common off-the-shelf sensor node device. The working voltage of eZ430

is 1.8 V and the working currents are 270  $\mu$ A, 0.1  $\mu$ A, 11.1 mA and 16.6 mA for processor processing, sleep mode, transmission mode, and reception mode, respectively. According to Reference [28], the sensor node can use the energy most efficiently when the ratios for the processor processing, sleep mode, transmission mode, and reception mode are 0.47%, 98.77%, 0.48% and 0.28%, respectively. According to the ratios, the current is about 0.1 mA for the eZ430-based sensor node to use the energy most efficiently, so the power consumption is thus about  $0.1 \text{ mA} \times 1.8 \text{ V} = 1.8 \text{ mW}$  for such ratios.

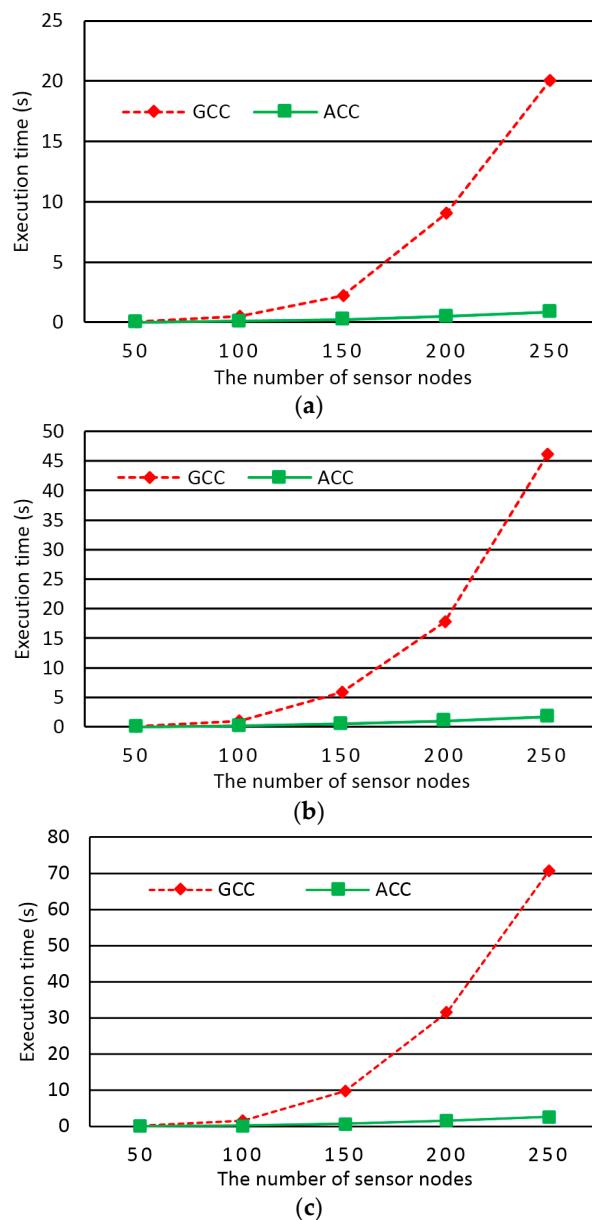


**Figure 20.** The comparisons of GCC and ACC in terms of the number of deployed chargers for: (a) 1-coverage; (b) 2-coverage; and (c) 3-coverage requirement.

We perform simulations for three scenarios of energy requirements, namely the low, medium, and high power requirement scenarios. For the low power requirement scenario, it is assumed 80% of sensor nodes are of 1.8 mW power requirement, 10%, 5.4 ( $=1.8 \times 3$ ) mW and 10%, 9.0 ( $=1.8 \times 5$ ) mW. For the medium energy requirement scenario, it is assumed 10% of sensor nodes are of 1.8 mW

power requirement, 80%, 5.4 mW and 10%, 9.0 mW. And for the high power requirement scenario, it is assumed 10% of sensor nodes are of 1.8 mW power requirement, 10%, 5.4 mW and 80%, 9.0 mW.

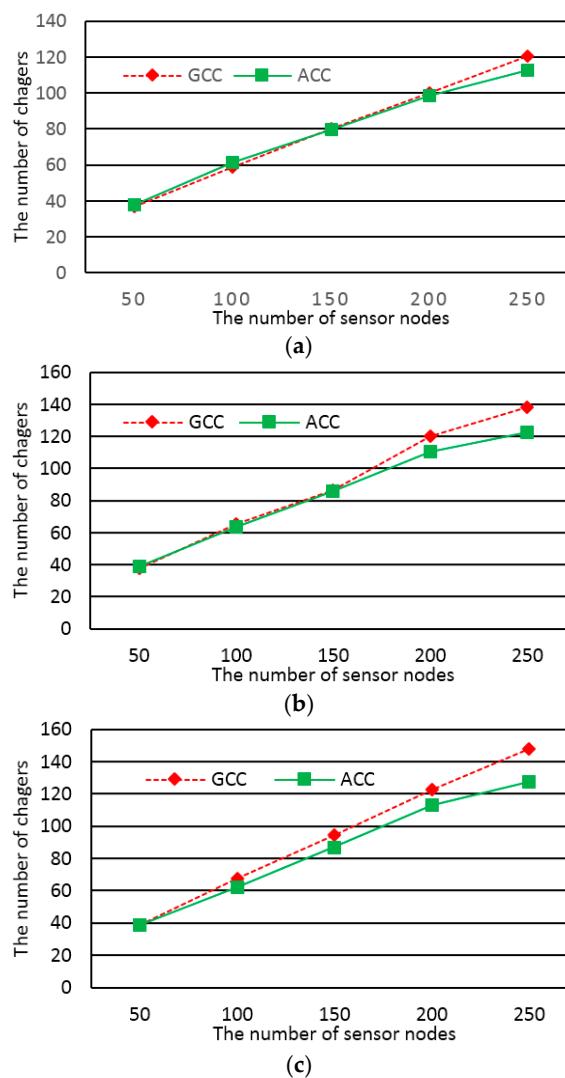
In the candidate cone generation phase, the GCC and the ACC algorithms generate the same number of candidate cones no matter what energy requirements (coverage requirements or power requirements) are considered. Please refer to Figure 19 for the comparisons of GCC and ACC in terms of the number of candidate cones.



**Figure 21.** The comparisons of GCC and ACC in terms of the execution time for: (a) 1-coverage; (b) 2-coverage; and (c) 3-coverage requirements.

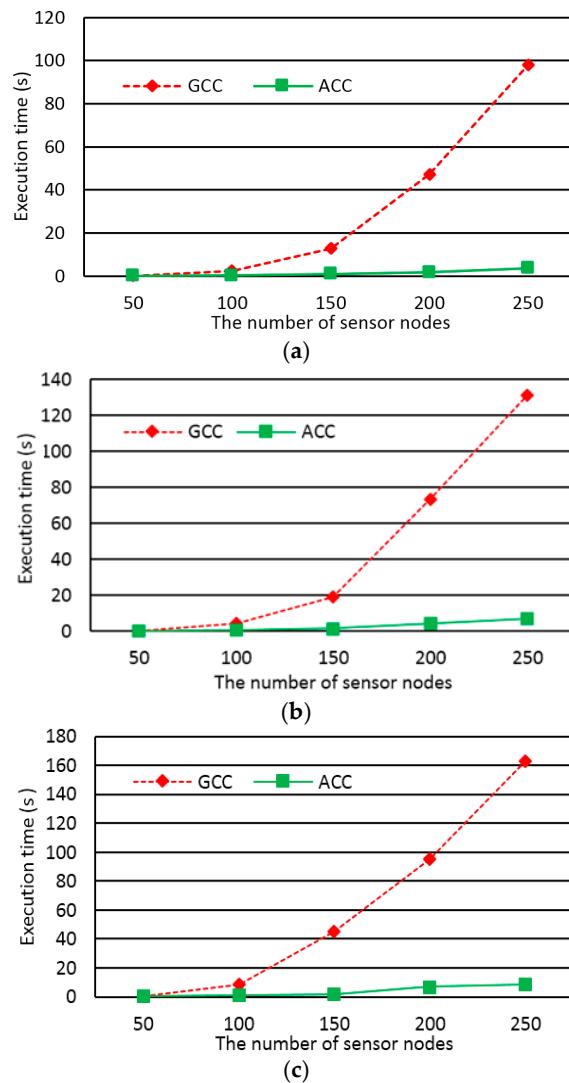
Figure 22 shows the comparisons of GCC and ACC in terms of the number of the chargers deployed for different power requirement scenarios. We can observe that ACC outperforms GCC for almost all scenarios. The superiority of ACC is most significant in the high power requirement scenario, while is least significant in the low power requirement scenario. This is because GCC greedily adjust directions of cones to the extreme for some sensor node-pair cases that nodes can cover by one cone. Consequently, some sensors are covered by chargers with larger horizontal and vertical angles.

Hence, those sensor nodes may need to be covered by more cones (i.e., chargers) to make their power requirements satisfied. On the contrary, ACC greedily and adaptively adjust the direction of every cone to cover as many as possible sensor nodes by vector composition. Consequently, most sensors are covered by chargers with smaller horizontal and vertical angles. Hence, sensor nodes may need to be covered by fewer cones (i.e., chargers) to make their power requirements satisfied. The explanation above also accounts for the reason why the superiority of ACC is more significant for scenarios of higher power requirements.



**Figure 22.** The comparisons of GCC and ACC in terms of the number of deployed chargers for: (a) low; (b) medium; and (c) high power requirement.

Figure 23 shows the comparisons of GCC and ACC in terms of execution time for scenarios of different power requirements. We can observe that ACC runs much faster than GCC, especially for the scenario of high energy consumption. This is because ACC has  $O(m^2n^3)$  time complexity, while GCC has  $O(m^2n^5)$  time complexity.



**Figure 23.** The comparisons of GCC and ACC in terms of the execution time for: (a) low; (b) medium; and (c) high power requirement.

## 7. Conclusions

In this paper, we study the WCDO problem concerning how to deploy as few as possible chargers to make a WRSN sustainable. We have proposed two heuristic algorithms, namely the GCC algorithm and the ACC algorithm, to provide approximate solutions to the problem. We consider two forms of energy requirements: coverage requirement and power requirement. For the coverage requirement scenarios, GCC outperforms ACC in terms of the number of chargers deployed. On the contrary, for the power requirement scenarios, ACC outperforms GCC in terms of the number of chargers deployed. However, for both forms of requirements, ACC has shorter execution time than GCC, as ACC has  $O(m^2n^3)$  time complexity and GCC has  $O(m^2n^5)$  time complexity, where  $n$  is the number of sensor nodes and  $m$  is the number of grid points.

In the proposed GCC and ACC algorithms, the chargers are restricted to be deployed on grid points of a grid, which may limit the application of the proposed algorithms. Moreover, the charger's charging space is modeled as a cone with a specific generatrix length and a specific opening angle, which may not be accurate enough for calculating the energy provision offered by the charger. We are now designing more efficient and flexible algorithms, such as meta-optimization algorithms, to solve the problem without the charger deployment restriction or the cone modeling of the charging space.

Such enhanced algorithms are suitable to WSNs in a closed space or room like space capsule in which sensor nodes have known and fixed positions and energy requirements. Redundant sensor nodes may be applied to such WSNs to prolong the network life time; however, the weight of those redundant nodes may be a burden on the space capsule. The WRSNs that rely on WPT to power sensor nodes are thus suitable for the environment, to which the enhancements of the GCC and ACC algorithms can be applied.

We have noticed a related solution [29] using Voronoi diagrams to help solve the directional coverage problem in WSNs whose sensor nodes are associated with rotatable directional sector-shaped sensing areas. However, the solution is for 2D environments and cannot be applied directly for solving the WCDO problem which considers 3D environments. We thus plan to study how to apply 3D Voronoi diagram algorithms to solve the WCDO problem in the future.

**Acknowledgments:** This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant Nos. 104-2221-E-008-017-, 105-2221-E-008-078- and 105-2218-E-008-008-.

**Author Contributions:** Ji-Hau Liao conceived, designed and performed the experiments; Jahn-Ruey Jiang and Ji-Hau Liao designed the algorithms, analyzed the data, and wrote the paper.

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

## References

1. 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. [[CrossRef](#)]
2. Tan, Y.K.; Panda, S.K. Review of Energy Harvesting Technologies for Sustainable Wireless Sensor Network. In *Sustainable Wireless Sensor Networks*; Seah, W., Tan, Y.K., Eds.; Tech Publishing: Rijeka, Croatia, 2010; pp. 15–43.
3. Kim, S.; Vyas, R.; Bito, J.; Niotaki, K.; Collado, A.; Georgiadis, A.; Tentzeris, M.M. Ambient RF energy-harvesting technologies for self-sustainable standalone wireless sensor platforms. *Proc. IEEE* **2014**, *102*, 1649–1666. [[CrossRef](#)]
4. Fu, L.; Cheng, P.; Gu, Y.; Chen, J.; He, T. Minimizing charging delay in wireless rechargeable sensor networks. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Turin, Italy, 14–19 April 2013; pp. 2922–2930.
5. He, S.; Chen, J.; Jiang, F.; Yau, D.; Xing, G.; Sun, Y. Energy provisioning in wireless rechargeable sensor networks. *IEEE Trans. Mob. Comput.* **2013**, *12*, 1931–1942. [[CrossRef](#)]
6. Dai, H.; Chen, G.; Wang, C.; Wang, S.; Wu, X.; Wu, F. Quality of energy provisioning for wireless power transfer. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 527–537. [[CrossRef](#)]
7. Xu, W.; Liang, W.; Lin, X.; Mao, G.; Ren, X. Towards perpetual sensor networks via deploying multiple mobile wireless chargers. In Proceedings of the 43rd IEEE International Conference on Parallel Processing (ICPP), Minneapolis, MN, USA, 9–12 September 2014; pp. 80–89.
8. Liang, W.; Xu, W.; Ren, X.; Jia, X.; Lin, X. Maintaining sensor networks perpetually via wireless recharging mobile vehicles. In Proceedings of the IEEE 39th conference on local computer networks (LCN), Edmonton, AB, Canada, 8–11 September 2014; pp. 270–278.
9. Ye, X.; Liang, W. Charging utility maximization in wireless rechargeable sensor networks. *Wirel. Netw.* **2016**. [[CrossRef](#)]
10. Zhao, M.; Li, J.; Yang, Y. A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 2689–2705. [[CrossRef](#)]
11. Han, G.; Qian, A.; Liu, L.; Jiang, J.; Zhu, C. Impacts of traveling paths on energy provisioning for industrial wireless rechargeable sensor networks. *Microprocess. Microsyst.* **2015**, *39*, 1271–1278. [[CrossRef](#)]
12. Lin, C.; Wu, G.; Obaidat, M.S.; Yu, C.W. Clustering and splitting charging algorithms for large scaled wireless rechargeable sensor networks. *J. Syst. Softw.* **2016**, *113*, 381–394. [[CrossRef](#)]
13. Lin, C.; Wang, Z.; Han, D.; Wu, Y.; Yu, C.W.; Wu, G. TADP: Enabling temporal and distal priority scheduling for on-demand charging architecture in wireless rechargeable sensor Networks. *J. Syst. Archit.* **2016**. [[CrossRef](#)]

14. Maehara, D.; Tran, G.K.; Sakaguchi, K.; Araki, K.; Furukawa, M. Experiments validating the effectiveness of multi-point wireless energy transmission with carrier shift diversity. *IEICE Trans. Commun.* **2014**, *97*, 1928–1937. [[CrossRef](#)]
15. Maehara, D.; Tran, G.K.; Sakaguchi, K.; Araki, K. Experimental study on battery-less sensor network activated by multi-point wireless energy transmission. 2016. arXiv:1601.00788. arXiv.org e-Print archive. Available online: <https://arxiv.org/abs/1601.00788> (accessed on 15 August 2016).
16. Imoto, N.; Yamashita, S.; Ichihara, T.; Yamamoto, K.; Nishio, T.; Morikura, M.; Shinohara, N. Experimental investigation of co-channel and adjacent channel operations of microwave power and IEEE 802.11g data transmissions. *IEICE Trans. Commun.* **2014**, *97*, 1835–1842. [[CrossRef](#)]
17. Yamashita, S.; Imoto, N.; Ichihara, T.; Yamamoto, K.; Nishio, T.; Morikura, M.; Shinohara, N. Implementation and feasibility study of co-channel operation system of microwave power transmissions to IEEE 802.11-based batteryless sensor. *IEICE Trans. Commun.* **2014**, *97*, 1843–1852. [[CrossRef](#)]
18. Yoshida, S.; Hasegawa, N.; Kawasaki, S. The aerospace wireless sensor network system compatible with microwave power transmission by time-and frequency-division operations. *Wirel. Power Transf.* **2015**, *2*, 3–14. [[CrossRef](#)]
19. Ichihara, T.; Mitani, T.; Shinohara, N. Study and development of an intermittent microwave power transmission system for a ZigBee device. In Proceedings of the IEEE Wireless Power Transfer Conference (WPTC), Jeju, Korea, 8–9 May 2014; pp. 40–43.
20. Correia, R.; De Carvalho, N.B.; Fukuday, G.; Miyaji, A.; Kawasaki, S. Backscatter wireless sensor network with WPT capabilities. In Proceedings of the IEEE MTT-S International Microwave Symposium, Phoenix, AZ, USA, 17–22 May 2015; pp. 1–4.
21. Powercast TX91501–3W-ID Wireless Charger User Manual. Available online: <http://www.powercastco.com/PDF/TX91501-manual.pdf> (accessed on 30 May 2016).
22. Powercast P2110-EVAL-02 Power Harvester User Manual. Available online: <http://www.powercastco.com/PDF/P2110-EVAL-02-manual.pdf> (accessed on 30 May 2016).
23. Shaw, J.A. Radiometry and the Friis transmission equation. *Am. J. Phys.* **2013**, *81*, 33–37. [[CrossRef](#)]
24. Yang, Q.; Nofsinger, A.; McPeek, J.; Phinney, J.; Knuesel, R. A complete solution to the set covering problem. In Proceedings of the International Conference on Scientific Computing (CSC), Las Vegas, NV, USA, 27–30 July 2015; p. 36.
25. Hennessy, J.L.; Patterson, D.A. *Computer Architecture: A Quantitative Approach*, 5th ed.; Elsevier: Amsterdam, The Netherlands, 2011.
26. Nagelkerke, N.J. A note on a general definition of the coefficient of determination. *Biometrika* **1991**, *78*, 691–692. [[CrossRef](#)]
27. TI eZ430-RF2500 Development Tool Guide (Rev. F). Available online: <http://www.ti.com/lit/pdf/slau227> (accessed on 30 May 2016).
28. Bougard, B.; Catthoor, F.; Daly, D.C.; Chandrakasan, A.; Dehaene, W. Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives. *Des. Autom. Test Eur.* **2005**, *1*, 196–201.
29. Sung, T.W.; Yang, C.S. Voronoi-based coverage improvement approach for wireless directional sensor networks. *J. Netw. Comput. Appl.* **2014**, *39*, 202–213. [[CrossRef](#)]



© 2016 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/>).