Wireless Charger Deployment Optimization for Wireless Rechargeable Sensor Networks

Ji-Hau Liao and Jehn-Ruey Jiang
Department of Computer Science and Information
National Central University
Jhongli, Taiwan

***Abstract*—In a wireless rechargeable sensor network (WRSN), sensor nodes can harvest energy from wireless chargers to refill their power supplies so that the WRSN can operate sustainably. This paper considers wireless chargers equipped with 3D-beamforming directional antennas, and assumes they can be deployed on grid points at a fixed height to propose two greedy algorithms solving the following critical problem: how to deploy as few as possible chargers to make the WRSN sustainable. The first algorithm is the node based greedy cone selecting (NB-GCS) algorithm trying to optimize the number of chargers based on node positions. The second algorithm is the pair based greedy cone selecting (PB-GCS) algorithm trying to optimize the number of chargers based on node pairs. We conduct simulation and analyze the time complexity of the NB-GCS and PB-GCS algorithms. As will be shown, the latter is better in terms of the number of chargers, while the former has lower time complexity.**

Keywords—wireless rechargeable sensor networks; charger deployment; sustainability; greedy algorithms; directional antennas

I. INTRODUCTION

A wireless rechargeable sensor network (WRSN) [1-3] consists of a lot of sensor nodes and few sink nodes, where sensor nodes are powered by batteries and can sense physical phenomena (e.g., temperature, humidity, and light intensity, etc.) and transmit the sensed data to sink nodes through multi-hop wireless communications. In a WRSN, sensor nodes utilize the energy harvesting technology [4-9], to convert harvested energy, such as solar, wind, radio frequency (RF) energies, into direct currents (DC) to replenish their power supplies so that the WRSN can operate sustainably.

Energy harvesting technologies are generally divided into two classes: (1) intensive energy harvesting and (2) non-intensive energy harvesting. The former uses a specific device, called the wireless charger, to perform the wireless charging process to transmit energy to power receivers attached to sensor nodes [4]. The latter mounts sensor nodes onto the energy harvesting device, such as solar panels, to independently harvest energy. The latter is more difficult to control as it is prone to environmental factors. Accordingly, this paper focuses on WRSNs using the intensive energy technology.

The chargers are expensive and their deployment is a time- and cost-consuming task. This motivates us to study the optimization problem of how to deploy as few as possible chargers in a WRSN to cover all sensor nodes to make the WRSN sustainable. This paper considers a WRSN with wireless chargers equipped with 3D-beamforming directional antennas, assumes chargers are deployed on grid points at a fixed height, and proposes two greedy algorithms solving the optimization problem. The charging space of a charger equipped with a directional antenna is modelled by a cone. The first algorithm is the node based greedy cone selecting (NB-GCS) algorithm trying to optimize the number of chargers based on node positions. The second algorithm is the pair based greedy cone selecting (PB-GCS) algorithm trying to optimize the number of chargers based on node pairs. We conduct simulation and analyze the time complexity of the NB-GCS and PB-GCS algorithms. As will be shown, the latter is better in terms of the number of chargers, while the former has lower time complexity.

The remainder of the paper is organized as follows. Section 2 presents the problem definition. In section 3, two algorithms to solve the problem are proposed and analyzed. The simulation results and their comparisons are described in section 4. And finally, section 5 concludes the paper.

II. PROBLEM DEFINITION

Sensor nodes in the WRSN are assumed to be deployed in a cuboid with the length L, width W and height H; they can be located on the ground or object surfaces. On the other hand, the wireless chargers equipped with 3D-beamforming directional antennas are assumed to be deployed at grid points on the grid with height H, where the side length of the grid is G and each grid point allows the deployment of several wireless chargers. All sensor nodes are homogeneous and all chargers are also homogeneous. Fig. 1 shows a scenario of the WRSN schematically.

The effective charging space of wireless chargers is assumed to be a cone, called a *charger cone*. As shown in Fig. 2, every charger cone is characterized by an apex o, a normal vector $\vec{N}$ whose direction is parallel to the symmetrical axis of the cone, an effective charging distance R, and an effective charging angle threshold θ (i.e., the acute angle between the cone lateral surface and the cone symmetrical axis). 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. The point X in Fig. 2 is an *extreme point* within the charger cone; it is on the inner side of the charger cone surface and its distance to the cone apex is R.

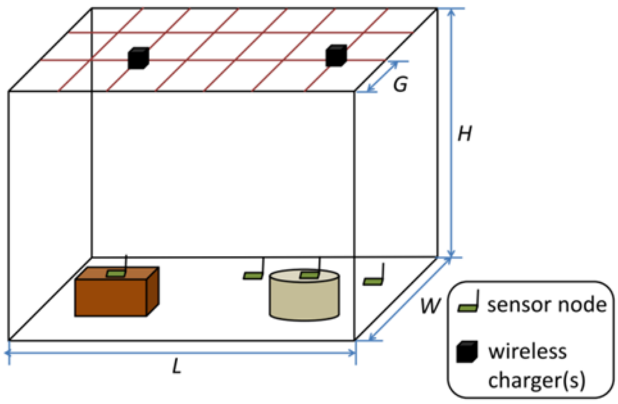

Figure 1. The schematic view of the WRSN

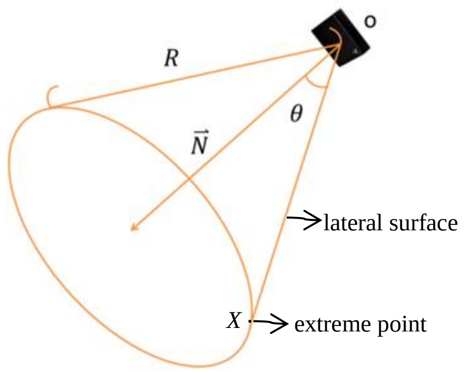

Figure 2. A charger cone and its parameters

With all the assumptions mentioned above, this paper tries to solve the *wireless charger deployment optimization (WCDO)* problem to deploy the minimum number of wireless chargers to cover all sensor nodes. In the next section, two heuristic algorithms that utilize the greedy concept are proposed to solve the problem effectively.

III. Two Greedy Algorithms to Solve the WCDO Problem

The WCDO problem can be solved by reducing it to the NP-hard set covering (SC) problem, which is to identify the smallest number of Q's subsets whose union is U, where U is a given universal set and Q is a collection of subsets of U. It is believed the WCDO problem is also NP-hard. Nevertheless, the NP-hardness of the WCDO problem has not yet proven. The reduction is done by transforming the set of all sensor nodes into U and transforming a set of sensors covered by a charger cone into a set in S.

Below in this subsection, we propose two greedy heuristic algorithms: the NB-GCS (Node Based Greedy Cone Selecting) algorithm and the PB-GCS (Pair Based Greedy Cone Selecting) algorithm to solve the WCDO problem near-optimally.

For a WSRN with n sensor nodes and with chargers being deployed on some of p grid points, the set of sensor nodes is denoted by $SN = \{s_1,\ s_2, \dots, s_n\}$ and the set of grid points is denoted by $GP = \{g_1,\ g_2, \dots,\ g_p\}$, where $p = (\left\lfloor\frac{L}{G}\right\rfloor + 1) \times (\left\lfloor\frac{W}{G}\right\rfloor + 1)$, G is *separation* of grid points (i.e., the distance between two nearby grid points), and L, W and H are the *length*, *width*, and *height* of the sensor node deployment cuboid, respectively.

The NB-GCS and PB-GCS algorithms first unmark all sensor nodes, and generate cones whose apexes are located at grid points. Specifically, the former algorithm generates cones on a node-by-node basis, while, the latter, a node-pair-by-node-pair basis. The algorithms then run iteration by iteration, and greedily select the cone covering the most unmark sensor nodes. They mark every node sensor that is properly covered at each iteration, and run until all sensor nodes are marked. Note that we assume each sensor node in $SN = \{s_1,\ s_2, \dots, s_n\}$ has the coverage requirement $CN = \{c_1,\ c_2, \dots, c_p\}$, where c_i indicates sensor node s_i needs to be covered by at least c_i chargers to meet the coverage requirement. Such a requirement can be obtained by estimating the worst case charging efficiency of a sensor node charged by a charger and by estimating the energy consumption of the sensor node. In this way, the WCPO problem can be solved near-optimally. Below, the pseudo codes of NB-GCS and PB-GCS are shown in Fig. 3 and Fig. 4, respectively.

In the pseudo codes, $\overrightarrow{u_x}$ stands for a vector going from grid point g to sensor node s_{a_x}, and $Cone(\overrightarrow{u_x})$ stands for a cone which takes g as the apex, takes $\overrightarrow{u_x}$ as the symmetrical axis (or the *direction*), and has an effective charging distance R and an effective charging angle threshold θ. Also note that $|Cone(\overrightarrow{u_x})|$ stands for the number of sensor nodes in SN covered by $Cone(\overrightarrow{u_x})$.

In the NB-GCS algorithm, for any grid point g, a sphere S is obtained with g as the center and R as the radius. If S covers k sensor nodes $s_{a_1}, s_{a_2}, \dots, s_{a_k}$, then k candidate cones will be generated to take vectors going from g to $s_{a_1}, s_{a_2}, \dots, s_{a_k}$ as directions, respectively. The basic idea of NB-GCS is to adjust the direction of the k candidate cones to cover more nodes in SN. This is achieved by moving the direction of $Cone(\overrightarrow{u_x})$ towards the direction of $Cone(\overrightarrow{u_y})$ through calculating $\overrightarrow{u_x} + \overrightarrow{u_y}$ for every sensor node s_{a_x} and s_{a_y}, where,$1 \le x$, $y \le k$ and $x \ne y$. The cone direction adjustment, namely $\overrightarrow{u_x} = \overrightarrow{u_x} + \overrightarrow{u_y}$, proceeds on a node-by-node basis. Certainly, the direction adjustment should guarantee that the number of sensor nodes covered by $Cone(\overrightarrow{u_x})$ is increasing and that the sensor node s_{a_x} is still covered by $Cone(\overrightarrow{u_x})$ with the adjusted $\overrightarrow{u_x}$.

Algorithm: NB-GCS (Node Based Greedy Cone Selecting)

Input: $SN = \{s_1,\ s_2, \dots, s_n\}, CN = \{c_1,\ c_2, \dots, c_n\}, GP = \{g_1,\ g_2, \dots, g_p\}, n, p, R, \theta$
Output: C^*, m

Step 1:
Set all nodes in *SN* as unmarked; set cone set *C* and C^* as empty
Step 2:
For each grid point $g \in GP$ do
  S = a sphere centered at *g* with radius *R*
  If (*S covers k (k*>*0)* nodes $s_{a_1}, s_{a_2}, \dots, s_{a_k}$) then
    Generates $\overrightarrow{u_1} = \frac{\overrightarrow{gs_{a_1}}}{\|\overrightarrow{gs_{a_1}}\|}, \overrightarrow{u_2} = \frac{\overrightarrow{gs_{a_2}}}{\|\overrightarrow{gs_{a_2}}\|}, \dots, \overrightarrow{u_k} = \frac{\overrightarrow{gs_{a_k}}}{\|\overrightarrow{gs_{a_k}}\|}$
    // $\overrightarrow{u_x}$ is a vector going from *g* to s_{a_x}, 1≤*n*≤*k*
    For *x* = 1, 2,…, *k* do
      For *y* = 1, 2,…, *k* do
        If $(\emptyset(\overrightarrow{u_x}, \overrightarrow{u_y}) \le \theta, x \ne y$, and $s_{a_x} \in Cone(\overrightarrow{u_x} + \overrightarrow{u_y})$, and $|\,Cone(\overrightarrow{u_x} + \overrightarrow{u_y})| > |\,Cone(\overrightarrow{u_x})|$) then
        $\overrightarrow{u_x} = \overrightarrow{u_x} + \overrightarrow{u_y}$, $\overrightarrow{u_x} = \frac{\overrightarrow{u_x}}{\|\overrightarrow{u_x}\|}$
    Add into *C* the *k* cones $Cone(\overrightarrow{u_1})$,…,$Cone(\overrightarrow{u_k})$
Step 3:
Repeat
  Select cone *w* in *C* that covers the most unmarked nodes in *SN*
  Set $C^* = C^* \cup \{w\}$ and *C*=*C*−{*w*}
  For every sensor node s_i covered by *w* do
    Increase by 1 the coverage degree d_i of s_i
    If ($d_i \ge c_i$) then mark s_i
Until (all nodes in *SN* are marked)
Step 4:
Return C^* and $m = |C^*|$

Figure 3. The NB-GCS algorithm

Algorithm: PB-GCS (Pair Based Greedy Cone Selecting)

Input: $SN = \{s_1,\ s_2, \dots, s_n\}, CN = \{c_1,\ c_2, \dots, c_n\}, GP = \{g_1,\ g_2, \dots, g_p\}, n, p,\ R, \theta$
Output: C^*, m

Step 1:
Set all nodes in *SN* as unmarked; set cone set *C* and C^* as empty
Step 2:
For each grid point $g \in GP$ do
  S = a sphere centered at *g* with radius *R*
  If (*S covers* 1 node s_{a_1}) then
    Generate $\overrightarrow{u_1} = \overrightarrow{gs_{a_1}}$ and add $Cone(\overrightarrow{u_1})$ into *C*
  Else If (*S covers k, k*>1, nodes $s_{a_1}, s_{a_2}, \dots, s_{a_k}$) then
    Generates $\overrightarrow{u_1} = \frac{\overrightarrow{gs_{a_1}}}{\|\overrightarrow{gs_{a_1}}\|}, \overrightarrow{u_2} = \frac{\overrightarrow{gs_{a_2}}}{\|\overrightarrow{gs_{a_2}}\|}, \dots, \overrightarrow{u_k} = \frac{\overrightarrow{gs_{a_k}}}{\|\overrightarrow{gs_{a_k}}\|}$
    // $\overrightarrow{u_x}$ is a vector going from *g* to s_{a_x}, 1≤*n*≤*k*
    For *x* = 1, 2,…, *k* do
      For *y* = 1, 2,…, *k* do
        If $(\emptyset(\overrightarrow{u_x}, \overrightarrow{u_y}) = \theta, x \ne y)$ then
          Add into *C* the cone indicated by Fig. 6 (i)
        Else If $(\emptyset(\overrightarrow{u_x}, \overrightarrow{u_y}) > \theta, x \ne y)$ then
          Add into *C* the 2 cones indicated by Fig. 6 (ii)
        Else If $(\emptyset(\overrightarrow{u_x}, \overrightarrow{u_y}) < \theta, x \ne y)$ then
          Add into *C* the 4 cones indicated by Fig. 6 (iii)
Step 3:
Repeat
  Select cone *w* in *C* that covers the most unmarked nodes in *SN*
  Set $C^* = C^* \cup \{w\}$ and *C*=*C*−{*w*}
  For every sensor node s_i covered by *w* do
    Increase by 1 the coverage degree d_i of s_i
    If ($d_i \ge c_i$) then mark s_i
Until (all nodes in *SN* are marked)
Step 4:
Return C^* and $m = |C^*|$

Figure 4. The PB-GCS algorithm

Below we analyze the NB-GCS algorithm. In step 1, some initialization tasks are done in O(*n*) time. In step 2, for any grid point *g*, a sphere *S* centered at *g* of radius *R* is formed. If *S* covers *k* sensor nodes, then *k* candidate cones will be generated, *k*≤*n*. Every candidate cone is tested for possible direction adjustment for *k* times, so there are k^2=O(n^2) adjustment tests. Since there are *p* grid points, NB-GCS forms *p* spheres which respectively cover$k_1, k_2, \dots, k_p$ sensor nodes. Thus, the total number of candidate cones is $\sum_{i=1}^{p} k_i \le pn$ =O(*pn*) and the total number of adjustment tests is O(pn^2). Step 3 is a repeat-until loop. At each iteration of the loop, NB-GCS first selects the cone *w* from *C* that covers the most unmarked nodes in SN. Since there are at most *pn* cones and each cone can cover at most *n* nodes, the selection can be done in O(pn^2) time. NB-GCS then checks if every unmarked node covered by *w* can be marked by adjusting and checking the coverage degree of the node. The checking takes O(*n*) time. A cone can be selected only once in the repeat-until loop, so there are at most *pn* iterations in the loop. Thus, the time complexity of step 3 is O(p^2n^3). Since step 3 has the highest time complexity among all steps, we have that the time complexity of the NB-GCS algorithm is O(p^2n^3).

In the PB-GCS algorithm, for any grid point *g*, a sphere *S* centered at *g* of radius *R* is formed. If *S* covers *k* sensor nodes, then at most $4\,\mathrm{C}_2^k$ candidate cones are generated. This is because PB-GCS runs on the basis of node pairs of *k* nodes covered by the sphere. In practice, PB-GCS tests for every pair of two nodes if the angle between the vectors associated with the two nodes is within (i.e., less than) the angle threshold *θ*. There are three cases for the test. (Case 1) If the angle is equal to the threshold *θ*, 1 candidate cone is generated. (Case 2) If it is larger than to *θ*, 2 candidate cones are generated. (Case 3) If it is less than *θ*, 4 candidate cones are generated.

To be more precise, three cases exist for any pair of distinct sensor nodes s_x and s_y in terms of the angle between $\overrightarrow{gs_x}$ and $\overrightarrow{gs_y}$. To simplify the calculation, PB-GCS tests the three cases by projecting cones and vectors associated with sensor nodes onto unit sphere surfaces. Referring to Fig. 5, the projection of a vector (resp., cone) onto the surface of a unit sphere centered at *g* is a point (resp., circle of radius *r*). Let d_{xy} be the Euclidean distance between the two projection points of $\overrightarrow{gs_x}$ and $\overrightarrow{gs_y}$ onto the unit sphere. Referring to Fig. 6, there are three cases of the relationship between d_{xy} and *r*, which correspond to the three cases of the relationship of the angle threshold θ and the angle between $\overrightarrow{gs_x}$ and $\overrightarrow{gs_y}$. The candidate cones are then generated according to the three cases shown in Fig. 6 (i), (ii), and (iii).

Below, we analyze the PB-GCS algorithm. In step 1, some initialization tasks are done in O(n) time. In step 2, for any grid point g, a sphere S centered at g of radius R is formed. If S covers k sensor nodes, then at most $4C_2^k$ candidate cones are generated, where $k \leq n$. Since there are p grid points, the number of candidate cones generated is then $4pC_2^k \leq pn^2 = O(pn^2)$ and the time complexity to generate the candidate cones is also $O(pn^2)$. Step 3 is a repeat-until loop, and at each iteration of the loop, PB-GCS first selects the cone w from C that covers the most unmarked nodes in SN. Since there are at most pn^2 cones and each cone can cover at most n nodes, the selection can be done in O(pn^3) time. PB-GCS then checks if every unmarked node covered by w can be marked by adjusting and checking the coverage degree of the node. The checking takes O(n) time. A cone can be selected only once in the repeat-until loop, so there are at most pn^2 iterations in the loop. Thus, the time complexity of step 3 is O(p^2n^5). Since step 3 has the highest time complexity among all steps, we have that the time complexity of the PB-GCS algorithm is O(p^2n^5).

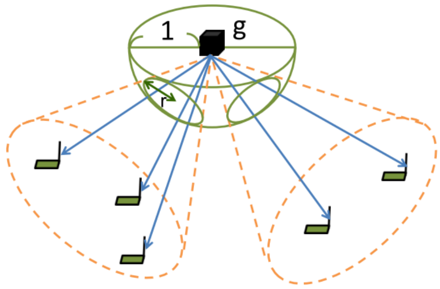

Figure 5. The illustration of the projection of cones and vectors onto the surface of the unit sphere centered at g

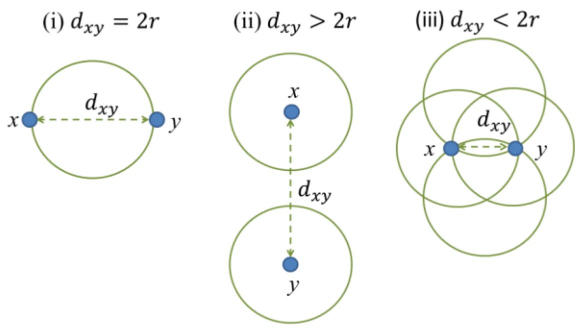

Figure 6. Three cases for the PB-GCS algorithm to generate candidate cones

IV. Simulation Results

In this section, we show the simulation results of the two proposed algorithms. The simulator is implemented in C++ language and the simulation settings are shown in Table I. The sensor nodes are assumed to be deployed randomly in a 20 x 15 m plane with a set of grid points of the separation of 1 m to deploy wireless chargers. For simplicity, we only consider the cases in which sensor nodes require 1-coverage or 2-coverage of chargers; that is, some sensor nodes need the coverage of only 1 charger and some others need the coverage of 2 chargers to make their energy sustainable. For a sensor node, we consider its worst-case power consumption and worst-case charging efficiency to estimate its coverage requirement. To be more precise, we consider the maximum power consumption per unit time (say, per day) and the minimum charging efficiency of the sensor node, which occurs when the sensor is located at an extreme point of a charger cone. We use x%-y% to mean that x% of sensor nodes need 1-coverage and y% of sensor nodes need 2-coverage of chargers. Every case of a simulation setting is run 100 times to obtain the average number of total chargers needed to meet the coverage requirement of all sensor nodes.

TABLE I. Simulation Settings

Item	Parameter
Sensor Node Plane	$20 \times 15\ (m^2)$
Number of Sensor Nodes	50, 100, 150, 200, 250
Effective Charging Distance	3 (m)
Angle Threshold	30°
Height of Grid Points	2.3 (m)
Separation of Grid Points	1 (m)

Fig. 7 shows the most ideal case of 100%-0%, in which all sensor nodes need the 1-coverage requirement; i.e., only 1 wireless charger is necessary for each sensor node to be energy-sustainable. The average numbers of wireless chargers necessary for NB-GCS and PB-GCS to cover 50, 100,…, 250 sensor nodes are shown and are marked as NB-GCS: 0-100% and PB-GCS: 0%-100%, respectively. As shown in Fig. 8, PB-GCS is better than NB-GCS in terms of the number of wireless chargers needed. When the number of sensor nodes is low, the difference between the two algorithms is not obvious. However, when the number of sensor nodes increases, the difference between them is significant.

Fig. 8 shows the Comparisons of PB-GCS and NB-GCS for the cases where 80% of nodes require 1-coverage and 20% of nodes require 2-covearge of chargers. Again, PB-BCS outperforms NB-GCS in terms of the number of chargers needed. Similarly, the superiority of PB-BCS is more significant when the number of sensor nodes is larger.

Fig. 9 summarizes the comparisons of PB-GCS and NB-GCS of Fig. 7 and Fig. 8 by showing histograms of the average numbers of deployed wireless chargers for 50, 100,…, 250 sensor nodes. By Fig. 9, the average number of wireless chargers for the case of PB-GCS: 80%-20% is even smaller than that that for the case of NB-GCS: 100%-0%. Because the number of candidate cones of PB-GCS is $O(pn^2)$, while the number of candidate cones of NB-GCS is $O(pn)$, where p is the number of grid points and n is the number of sensor nodes, PB-GCS has higher probability to select better candidate cones than NB-GCS.

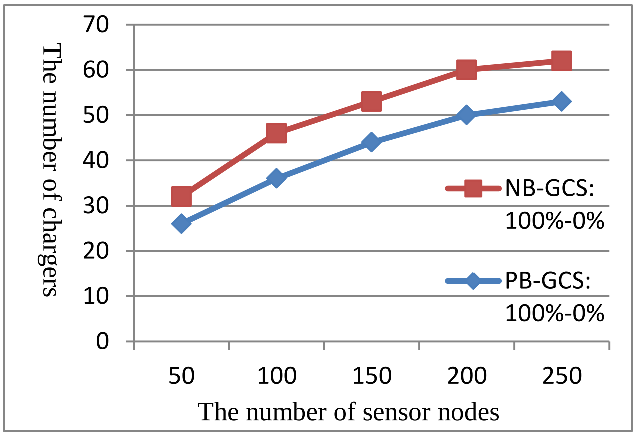

Figure 7. Comparisons of PB-GCS and NB-GCS for the cases in which all sensor nodes require 1-coverage of chargers

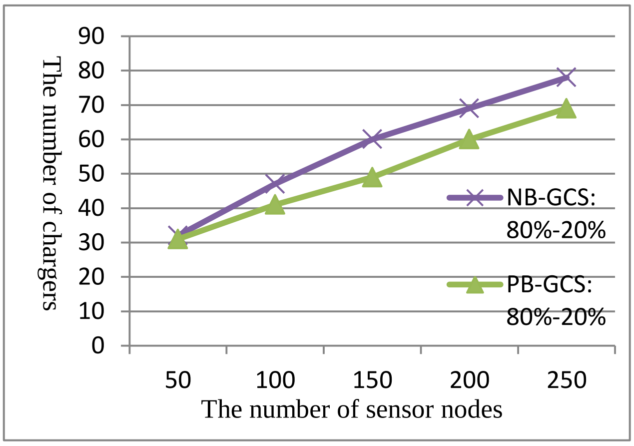

Figure 8. Comparisons of PB-GCS and NB-GCS for the cases in which 80% of sensor nodes require 1-coverage and 20% of sensor nodes require 2-coveage of chargers

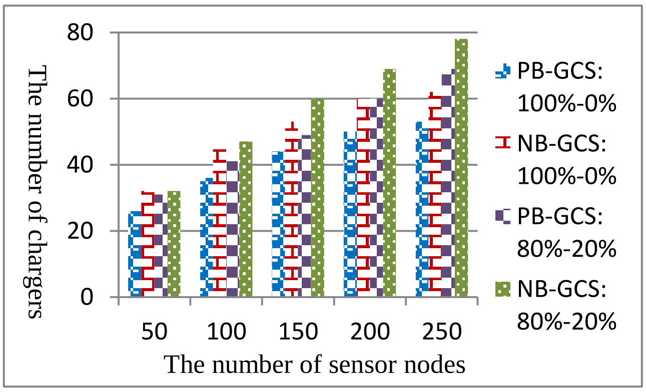

Figure 9. Summarized comparisons of PB-GCS and NB-GCS

V. Conclusion

In this paper, we propose two greedy algorithms, the node based greedy cone selecting (NB-GCS) algorithm and the pair based greedy cone selecting (PB-GCS) algorithm, to solve the wireless charger deployment optimization (WCDO) problem in a WRSN for deploying as few as possible wireless chargers to cover all sensor nodes to make the WRSN sustainable. We have conducted simulation experiments and analyzed the time complexity of the two algorithms. As shown in the paper, PB-GCS is better than NB-GCS in terms of the number of chargers deployed. However, NB-GCS has a lower time complexity (i.e., $O(p^2n^3)$) than that (i.e., $O(p^2n^5)$) of PB-GCS, where n is the number of sensor nodes and p is the number of grid points. In the future, we plan to perform more simulation experiments under more practical cases (e.g., cases considering charging efficiency models). We also plan to design more efficient algorithms (e.g., those without the limitation that chargers must be deployed at grid points) to solve the WCDO problem.

References

[1] S. He, J.-M. Chen, F. Jiang, D. Y. Yau, G. Xing, and Y.-X. X. Sun, "Energy Provisioning in Wireless Rechargeable Sensor Networks," *Proc. IEEE IFONCOM*, pp. 2006-2014, 2011.

[2] T.-C. Chiu, Y.-Y. Shih, A.-C. Pang, J.-Y. Jeng, and P.-C. Hsiu, "Mobility-aware charger deployment for wireless rechargeable sensor networks," *Proc. Network Operations and Management Symposium (APNOMS)*, pp. 1-7, 2012.

[3] K. Li, H. Luan, and C.-C. Shen, "Qi-Ferry: Energy-Constrained Wireless Charging in Wireless Sensor Networks," *IEEE Wireless Communications and Networking Conference*, pp. 2515–2520, 2012.

[4] A. M. Zungeru, L.-M. Ang, SRS. Prabaharan, and K. P. Seng, "Radio Frequency Energy Harvesting and Management for Wireless Sensor Networks," Green Mobile Devices and Networks: Energy Optimization and Scavenging Techniques, pp. 341-368, 2012.

[5] S. Escolar, S. Chessa, and J. Carretero, "Optimization of Quality of Service in Wireless Sensor Networks Powered by Solar Cells," Proc. 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications (ISPA), pp. 269-276, 2012.

[6] Fu-Da-Tong Technology Company website: http://www.rfidpower.com.tw/index.html

[7] PowerCast Corp. website: http://www.powercastco.com/

[8] Y. K. Tan, and S. K. Panda, "Review of Energy Harvesting Technologies for Sustainable Wireless Sensor Network," in Sustainable Wireless Sensor Networks, edited by Winston Seah and Yen Kheng Tan, Dec. 2010.

[9] W. Haboubi, H. Takhedmit, O. Picon, and L. Cirio, "A GSM-900 MHz and WiFi-2.45 GHz dual-polarized, dual-frequency antenna dedicated to RF energy harvesting applications," Proc. European Conference on Antennas and Propagation (EuCAP), pp. 3959 – 3961, Apr. 2013.