Particle Swarm Optimization for Charger Deployment in Wireless Rechargeable Sensor Networks

Yen-Chung Chen
Department of Computer Science and Information
National Central University
Taoyuan City, Taiwan

Jehn-Ruey Jiang
Department of Computer Science and Information
National Central University
Taoyuan City, Taiwan

***Abstract*—In Wireless Rechargeable Sensor Networks (WRSNs), wireless chargers can recharge batteries of sensor nodes so that they can operate sustainably. Since wireless chargers are costly and have limited charging distances and angles, how to apply as few as possible chargers to cover all sensor nodes and satisfy their energy requirements is thus an important and challenging problem. This paper proposes the PSCD (Particle Swarm Charger Deployment) algorithm using the Particle Swarm Optimization (PSO) concept to nearly optimize WRSN charger deployment. PSCD estimates charging efficiency according to the distance and angle between chargers and sensor nodes. It then, on the basis of PSO, utilizes the local optimal result and the global optimal result to adjust locations and antenna orientations of chargers to make WRSNs sustainable. We perform experiments using practical wireless chargers to obtain charging efficiency data. Based on the data, PSCD is simulated and compared with two related heuristic greedy algorithms to show its superiority.**

Keywords—Wireless Rechargeable Sensor Network; Particle Swarm Optimization; Sustainability; Wireless Charger Deployment

I. INTRODUCTION

A *wireless* rechargeable *sensor network (WRSN)* [1] consists of many *rechargeable sensor nodes*, some *wireless chargers*, and one or more *sink nodes*. Sensor nodes are usually powered by batteries and can sense environmental phenomena (e.g., temperature, humidity, atmospheric pressure, and light intensity) and forward the sensed data to sink nodes via multi-hop wireless communications. As shown in Figure 1, WRSN sensor nodes can also utilize energy harvesting technologies to harvest energy from different sources [2], such as solar power, thermal power, radio frequency (RF) waves, air flow, and vibrations, for recharging their batteries to make the WRSN sustainable. Due to the sustainability of WRSNs, they have attracted much research attention [3-9] in the last few years.

Energy harvesting technologies are divided into two classes: (1) charger-based energy harvesting and (2) ambient energy harvesting. The former deploys specific devices, called *chargers*, to emit energy, and utilizes *harvesters* to harvest the energy. The latter applies ambient energy harvesting devices, such as a solar panel and a thermoelectric generator, to harvest energy from ambient environments. It is more difficult for the latter to control the amount of harvested energy due to the unstable nature of ambient energy sources. This is problematic if sensor nodes have specific energy requirements to meet. Therefore, this paper pays attention to charger-based energy harvesting technologies. It practically focuses on indoor WRSNs using the RF-based energy harvesting technology.

The deployment of RF-based wireless chargers (or "wireless chargers" for short) is challenging, since every harvester needs to be covered by at least one charger, i.e., within the *charging range* or *charging distance* of a charger, and the charging range of chargers is usually small. For example, the effective charging range is 3-5 m between the Powercast TX91501-3W-ID charger [10] and the Powercast P2110-EVAL-02 harvester [11]. 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. The paper [1] addressed the wireless charger deployment optimization (WCDO) problem dealing with how to efficiently deploy as few as possible chargers to cover all WRSN sensor nodes for making the WRSN sustainable. It proposed two algorithms, namely the Pair Based Greedy Cone Selection (PB-GCS) algorithm and the Node Based Greedy Cone Selection (NB-GCS) algorithm, to solve the WCDO problem. They are heuristic greedy algorithms and provide approximate solutions to the problem.

This paper proposes the PSCD (Particle Swarm Charger Deployment) algorithm using the Particle Swarm Optimization (PSO) concept for providing better solutions to the WCDO problem than existing solutions. PSCD estimates charging efficiency according to the distance and the angle between chargers and sensor nodes. Based on the charging efficiency estimated, PSCD relies on PSO using the local optimum and global optimum to adjust locations and antenna directions of chargers to achieve near-optimal WCDO solutions. We perform experiments using practical wireless chargers to obtain charging efficiency data. Based on the data, we simulate the PSCD algorithm and compare the simulation results with those of the PB-GCS algorithm and the NB-GCS algorithm. The simulation results show that the proposed PSCD algorithm outperforms the other two algorithms in terms of the number of chargers needed to satisfy the energy requirements of all sensor nodes.

The rest of this paper is organized as follows. Section II presents the WCDO problem definition, and the two existing algorithms to solve the problem. Section III describes the proposed PSCD algorithm and Section IV shows the experimental results, simulation results, and comparisons of algorithms. And finally, Section V concludes the paper.

II. PROBLEM DEFINITION AND EXISTING SOLUTIONS

A. Problem Definition

The study makes the following assumptions. Sensor nodes in the WRSN are deployed in a cuboid with length L, width W and

978-1-5090-0919-0/16/$31.00 ©2016 IEEE

height H; they are located on the ground or object surfaces. Wireless chargers equipped with directional antennas of different *directions* or *orientations* are deployed on the *deployment plane* with height H. All sensor nodes and all chargers are homogeneous. Fig. 1 shows the scenario of a WRSN schematically.

The wireless charger deployment optimization (WCDO) problem deals with, under the above-mentioned assumptions, how to deploy as few as possible chargers in a WRSN to cover all sensor nodes with different energy requirements to make the WRSN sustainable. To be more precise, the WCDO problem is to decide the position of wireless chargers on the deployment plane and the orientation of charger antennas. It is believed the WCDO problem is NP-hard. Nevertheless, the NP-hardness of the WCDO problem has not yet proven.

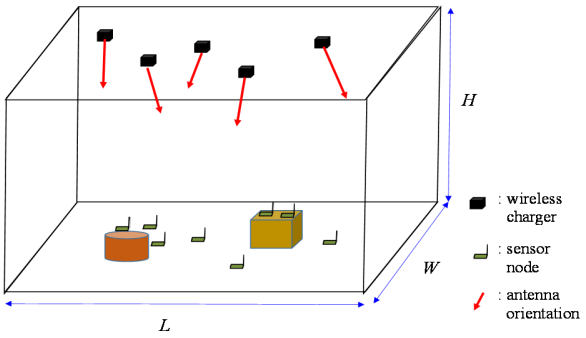

Fig. 1 The scenario of a WRSN.

B. Existing Solutions

The paper [1] proposed two greedy algorithms, called the Node Based Greedy Cone Selection (NB-GCS) and the Pair Based Greedy Cone Selection (PB-GCS), to solve the WCDO problem. The algorithms assume that the wireless chargers are deployed on grid points on the deployment plane with height H and with grid point separation G (refer to Fig. 2).

The algorithms also assume the effective charging space of wireless chargers is a cone, called a *charger cone*. As shown in Fig. 3, 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* 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, it is assumed that the sensor node can be charged effectively by the charger; otherwise, the sensor node cannot be charged effectively. The point E in Fig. 3 is an extreme point within the charger cone having the least energy efficiency; it is on the inner side of the charger cone surface and its distance to the cone apex is R.

The NB-GCS and PB-GCS algorithms assume that every sensor node has an *energy requirement* (or *coverage requirement*) as the number of chargers to cover it. In practice, it is assumed every sensor node derives the requirement by calculating the ratio $\lceil EC/CE \rceil$ of its estimated energy consumption (EC) over the charging efficiency (CE) of the extreme point E, where the energy consumption and the charging efficiency are both of the unit of mW.

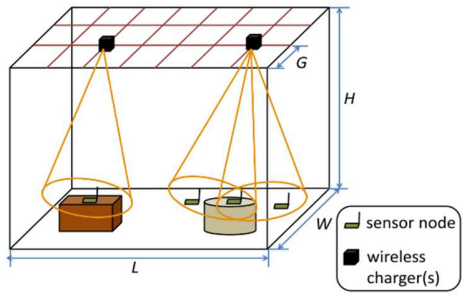

Fig. 2. The WRSN deployment cuboid and deplyment plane of grid points.

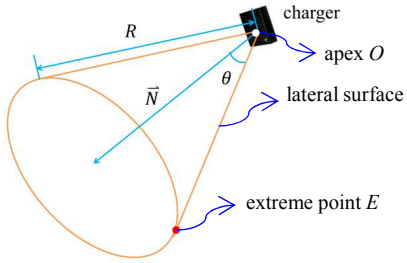

Fig. 3. A charger cone and its parameters

The two algorithms first unmark all sensor nodes, which means that the coverage requirements of all sensor nodes are not satisfied. They have two stages: cone generation and cone selection. In the first stage, the NB-GCS algorithm generates cones on a "node by node" basis, while, the PB-GCS algorithm, on a "node-pair by node-pair" basis. In the second stage, both the algorithms 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.

Below we describe the two algorithms in details. It is assumed that each sensor node in $SN = \{s_1, s_2, \ldots, s_n\}$ has the coverage requirement $CN = \{c_1, c_2, \ldots, c_n\}$ where c_i indicates sensor node s_i needs to be covered by at least c_i chargers to meet the coverage requirement. 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 $s_{a_1}, s_{a_2}, \ldots, s_{a_n}$, then k candidate cones will be generated that take vectors going from g to $s_{a_1}, s_{a_2}, \ldots, s_{a_k}$ as directions, respectively. The basic idea of NB-GCS is to adjust the directions 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}, $1 \le x$, $y \le k$ and $x \ne y$. Note that "+" stands for vector composition, and $\vec{u}_x$ (resp., $\vec{u}_y$) stands for a vector going from grid point g to sensor node s_{a_x} (resp., s_{a_y}), and $Cone(\overrightarrow{u_x})$ (resp., $Cone(\overrightarrow{u_y})$) stands for a cone taking R as the effective charging distance, θ as the effective charging angle, g as the apex, and $\overrightarrow{u_x}$ (resp., $\overrightarrow{u_y}$) as the symmetrical axis direction (or orientation). Also note that $|Cone(\overrightarrow{u_x})|$ stands for the number of sensor nodes in SN covered by $Cone(\overrightarrow{u_x})$. 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 $Cone(\overrightarrow{u_x})$ for the adjusted $\overrightarrow{u_x}$. Please refer to [1] for more details.

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 $s_{a_1}, s_{a_2}, \ldots, s_{a_k}$, then at most $4C_2^4$ 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, call the *test angle* below in this paper, between the vectors associated with the two nodes is smaller than the double of the effective charging angle θ. There are three cases for the test. (Case 1) If the angle is equal to 2θ, then one candidate cone is generated. (Case 2) If it is larger than 2θ, then two candidate cones are generated. (Case 3) If it is less than 2θ, then four candidate cones are generated.

To calculate the test angle between every pair of sensor nodes s_{a_x} and s_{a_y}, $1 \le x$, $y \le k$ and $x \ne y$, the PB-GCS algorithm projects vectors $\vec{u}_x$ and $\vec{u}_y$, and cones $Cone(\overrightarrow{u_x})$ and $Cone(\overrightarrow{u_y})$ associated with the sensor nodes onto the surface of the unit sphere centered at g. As shown in Fig. 4, 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 i and j be two projection points of $\vec{u}_x$ and $\vec{u}_y$ onto the unit sphere, respectively, and d_{xy} be the Euclidean distance between the two projection points i and j. As shown in Fig. 5 (i), (ii) and (iii), there are three cases of the relationship between d_{xy} and r. The three cases correspond to the cases of the test angle is (Case 1) equal to, (Case 2) less than, and (Case 3) larger than 2θ, respectively. For (Case 1), one candidate cone is generated. For (Case 2), two candidate cones are generated. For (Case 3), four candidate cones are generated. The candidate cones are generated according to the three cases shown in Fig. 5 (i), (ii), and (iii).

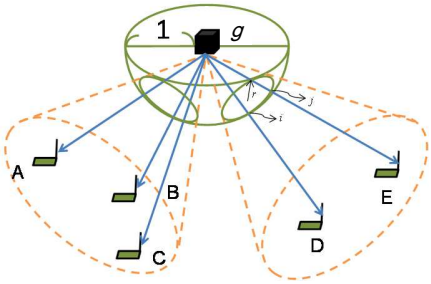

Fig 4. The illustration of the projection of cones and vectors onto the surface of the unit sphere centered at g.

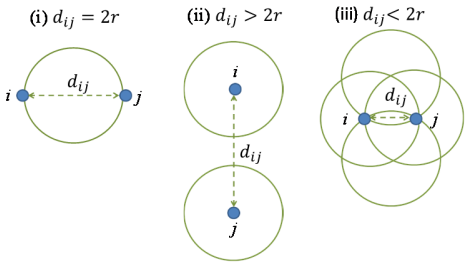

Fig. 5. Three cases for the PB-GCS algorithm to generate candidate cones.

After generating candidate cones, both the NB-GCS algorithm and the PB-GCS algorithm greedily choose the candidate cone covering the largest number of unmarked sensor nodes to keep the selected (or deployed) chargers as few as possible. If a sensor node's requirement is met, it is marked. The cone selection proceeds until all sensor nodes' requirements are satisfied. Please refer to [1] for more details.

III. The Proposed PSO-based Algorithm

A. Particle Swarm Optimization

The concept of Particle Swarm Optimization (PSO) [12] is first introduced by Kennedy and Eberhart in 1995. It can be used to develop evolutionary algorithms based on the foraging behavior of birds, described below. Birds of a swarm forage food at arbitrary positions in an area. Every bird does the forage based on its own experience; it keeps the local best known position of finding food. If birds find food, they notify each other of the food position. Therefore, every bird can also forage food based on the swarm experience; it also keeps the global best known position of finding food.

In a PSO algorithm, a particle in the solution space of a problem is regarded as a bird in a swarm of birds. The particle i is associated with a position p_i in an n-dimension solution space, a velocity v_i, and the particle's best known (i.e., local optimal) position L_p_i; the whole swarm of particles has the global best known (i.e., global optimal) position G_p. The local optimal and the global optimal position is derived on the basis of the fitness function. The fitness function returns a larger (resp., smaller) value for a solution closer to the optimal solution, which has the largest (resp., smallest) fitness function value. Note that every particle keeps the local optimal value and the global optimal value.

In a PSO algorithm, each particle adjusts its position and velocity iteration by iteration. After a period of time, most of the particles gradually move near the global optimal position, which corresponds to the approximate optimal solution to the problem.

With input parameters m, n, w, $c1$, and $c2$, the general steps of a PSO algorithm are shown below:

Step1: Generate m particles, and set the initial positon p_i and the initial velocity v_i in the n-dimension space for every particle i.

Step2: Calculate the fitness function value for every particle according to its position, and update the particle's local best known position L_p_i and global best known positon G_p.

Step3: Update the velocity and the position of every particle i as follows:

$$v_i(t+1) = w \times v_i(t) + c1 \times rand() \times \left(L\_p_i - p_i(t)\right) + c2 \times rand() \times (G\_p - p_i(t)) \quad (1)$$

$$p_i(t+1) = p_i(t) + v_i(t+1) \quad (2)$$

Step4: Check if the termination condition holds. If so, the algorithm terminates; otherwise, the algorithm continues by going to Step2. The termination condition may be that the algorithm has run the maximum number of iterations or that the fitness function value of the global optimal position does not increase by a specific value.

Below, we explain Eq. (1) and Eq. (2) used in Step3. In the equations, $p_i(t)$ is the position, and $v_i(t)$ is the velocity of

particle i at the t^{th} iteration. L_p_i is the local best known position, and G_p is the global best known position. The parameter w is the inertial weight, which is suggested to have a value between 0.8 and 1.2. A larger value of w leads to a search of a more global scope, while a smaller value leads to a search of a more local scope. The parameters $c1$ and $c2$ are *learning factors*; specifically, $c1$ is called the *cognitive factor* and $c2$ is called the *social factor*. They are suggested to have values between 0 and 4 (a common setting is $c1$=$c2$=2). Furthermore, the function $rand()$ is a function returning a uniform random value in the open range between 0 and 1.

B. The Proposed Algorithm

This subsection introduces the proposed PSCD (Particle Swarm Charger Deployment) algorithm to solve the WCDO problem on the basis of PSO. Like the NB-GCS algorithm and the PB-GCS algorithm, the proposed PSCD algorithm assumes that the deployment space is a cuboid with the length L, width W, and height H, and with sensor nodes distributed on the surface of the ground and objects. Unlike the two existing algorithms that assume wireless chargers are deployed at grid points on a deployment plane of height H, the PSCD algorithm assumes wireless chargers can be deployed at any point on the deployment plane of height H. Furthermore, the PSCD algorithm does not take the charger cone modeling, which assumes a sensor node can (resp., cannot) be charged effectively if it is inside (outside) the charger cone of the charger. Instead, it uses a function taking two parameters, namely (1) the charging distance between the charger and the sensor node, and (2) the charging angle between the direction of the charger antenna and the vector going from the charger to the sensor node, to calculate charging efficiency. In practice, through experimental data and interpolation, the charging efficiency (or energy received by harvesters) can be calculated very accurately.

In the PSCD algorithm, a charger is modeled as a particle i with a 5-dimension positon p_i and a 5-dimension velocity v_i, as defined below. Note that in the following context, we use the terms "sensor node" and "particle" interchangeably.

$$p_i(t) = \big(C_i(t), N_i(t)\big), \text{where } C_i(t) = (X_i, Y_i) \text{ and } N_i = (x_i, y_i, z_i) \quad (3)$$

$$v_i(t) = \big(v\_C_i, v\_N_i(t)\big) \quad (4)$$

Note that $C_i(t)$=(X_i, Y_i, Z_i), which corresponds to the charger position of particle i on the deployment plane of height H. Since all points on the deployment plane has the same Z-coordinate value H, the charger position only needs to keep the X- and Y-coordinate values, which are X_i and Y_i for particle i. Also note that $N_i(t)$ corresponds to the charger antenna direction of particle i, which is a 3-dimension vector (x_i, y_i, z_i). Similarly, the 5-dimesion velocity $v_i(t)$ of particle i consists of two components: v_C_i and v_N_i. The former is a 2-dimension vector representing the velocity of the X- and Y-coordinate values of the particle position, while the latter is a 3-dimension vector representing the velocity of antenna direction of the particle.

For every particle or charger, the PSCD algorithm evaluates its goodness by calculating its associate fitness function value. The fitness function f_i for particle i returns the accumulated energy received by all sensor nodes covered by the charger according to parameters. Assume particle i covers k sensor nodes $S_1, S_2, \dots, S_k$, and let d_{ij} denote the distance between particle i and sensor node S_j, and let $\emptyset\big(\overrightarrow{N_i}, \overrightarrow{C_iS_j}\big)$ denote the angle between the antenna direction vector $\overrightarrow{N_i}$ of particle i and the vector starting from particle i at position C_i and going to sensor node S_j. The fitness function of particle i is defined as follows.

$$f_i = \sum_{j=1}^{k} Energy(d_{ij}, \emptyset\big(\overrightarrow{N_i}, \overrightarrow{C_iS_j}\big)) \quad (5)$$

In Eq. (5), the function $Energy(d_{ij}, \emptyset\big(\overrightarrow{N_i}, \overrightarrow{C_iS_j}\big)$ returns the energy that is sent by charger i (i.e., particle i) and is received by sensor node S_j. The function can do the calculation according to the well known Friis transmission equation [13] considering the factors of d_{ij} and $\emptyset\big(\overrightarrow{N_i}, \overrightarrow{C_iS_j}\big)$. However, this paper performs the calculation by interpolating experimental data, as will be shown in the next section.

The fitness function can evaluate how good a particle is. Therefore, the algorithm can derive for every particle i the best local optimal fitness function value that is ever seen and stores corresponding particle information in L_C_i and L_N_i. The algorithm also keeps the global optimal fitness function value and stores corresponding particle information in G_C and G_N. By the particle information stored, the PSCD algorithm updates every particle at every iteration according to the following equations.

$$v_i(t+1) = w \times v_i(t) + c1 \times rand() \times (L\_C_i - C_i(t), L\_N_i) + c2 \times rand() \times (G\_C - C_i(t), G\_N) \quad (6)$$

$$p_i(t+1) = (C_i(t), (0,0,0)) + v_i(t+1) \quad (7)$$

Similar to the PSO notations mentioned earlier, $rand()$ is a function returning a uniform random value in the open range between 0 and 1, and t and t+1 stand for the iteration numbers in Eq. (6) and Eq. (7). Furthermore, w is the inertial weight, $c1$ is the cognitive factor, and $c2$ is the social factor. And $v_i(t)$ is the velocity of particle i, $C_i(t)$ is the deployment plane position of particle i, and N_i is the antenna direction vector of particle i.

Below we elaborate the steps of the PSCD algorithm:

Algorithm PSCD (Particle Swarm Charger Deployment)

Input: (1) PSO parameters: m, w, $c1$, and $c2$. (2) A set of WRSN sensor nodes with specified energy requirements within the cuboid of the length L, width W and height H. (3) The maximal iteration number B for calculating the information of a charger. (4) The experimental data for calculating the function *Energy*( ).

Output: A set CS of chargers with calculated positions on the deployment plane of height H and with calculated antenna directions.

Step1: Generate m particles and set the initial positon C_i, the initial antenna vector N_i, and the initial velocity v_i for every particle i.

Step2: Calculate the value of the fitness function f_i for particle i according to Eq. (5). Keep particle information corresponding to the local optimal fitness function value in L_C_i and L_N_i. Among

all local optimal values, derive the global optimal value and keep particle information corresponding to the value in *G_C* and *G_N*.

Step3: Update particle information according to Eqs. (6) and (7).

Step4: If the number of iterations does not exceed the specified number *B*, then go to Step2.

Step5: Deploy charger *C* according to the particle information stored in *G_C* and *G_N*. Add *C* into *CS*.

Step6: Check for every sensor node covered by the deployed charger *C* if its energy requirement is satisfied. If so, mark the sensor node. If all sensor nodes are marked, output *CS* and stop; otherwise, go to Step1.

IV. Experiments and Simulations

A. Experiments

In this subsection, we show results of extensive experiments in indoor environments for calculating the function *Energy*(). The devices used in the experiments are Powercast TX91501-3W-ID wireless charger and P2110CSR-EVB power harvester working in the 915 MHz band, as shown in Fig. 6. The former is a wireless charger with a patch directional antenna with 8dBi gain and 60° beamwidth in both horizontal and vertical directions [10]. The latter is a power harvester with a dipole omni-directional antenna with 1dBi gain. It can harvest energy from RF waves, provide intermittent/pulsed power output up to 4.21 V, and connect to rechargeable batteries including Alkaline, Lithium Ion, and Ni-MH.

As shown in Fig. 7, a wireless charger and a power harvester are set up. The distance *R* of the two devices is set to be 0.5 m, 1.0 m, …, or 6.0 m for experiments. As shown in Fig. 8, the orientation of the charger's patch antenna is specified by the charging angel θ, which is set to be 0°, 15°, …, or 90°.

By a multimeter, we measure the voltage (V) and current (I) of the output of the harvester 10 times for every distance R=0.5 m, 1.0 m, …, and 6.0 m, and for every charging angel θ=0°, 15°, …, and 90°. The measured results are then averaged as experimental data, and then applied the formula P=I×V to derive the received power. Table I shows partial experimental results of the received power (energy). Note that the voltage values smaller than 0.01 V or the current values smaller than 0.01 mA are not measurable and such cases lead to "not available" (i.e., "-") symbols shown in the Table. By interpolating data in Table I, we can easily estimate the received energy for any charging distances and charging angles.

(a) (b)

Fig. 6. (a) Powercast TX91501-3W-ID wireless charger, and (b) P2110CSR-EVB power harvester.

Fig. 7. The experimental setting of a wireless charger and a power harvester.

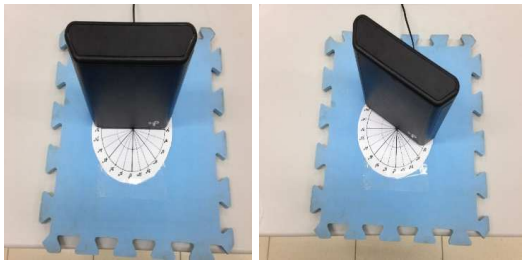

Fig. 8. Experimental settings for different charging angels.

TABLE I. EXPERIMENTAL DATA OF THE RECEIVED ENERGY*

R \ θ	0°	15°	30°	45°	60°	75°	90°
0.5 m	149.46	141.46	113.25	95.34	53.76	17.64	10.58
1.0 m	32.84	32.21	20.38	12.35	7.77	2.10	0.01
1.5 m	18.95	15.07	12.59	8.34	2.28	--	--
2.0 m	11.70	9.94	8.57	6.03	--	--	--
2.5 m	5.50	4.02	2.29	0.93	--	--	--
3.0 m	2.10	1.05	0.70	--	--	--	--
3.5 m	0.76	--	--	--	--	--	--

*in mW

B. Simulations

This section shows the simulation results and comparisons of the proposed PSCD algorithm, and the two related algorithms, namely the PB-GCS algorithm and the NB-GCS algorithm. The simulator is implemented in the C++ language and has the simulation settings shown in Table II. In summary, in the simulations, the WRSN sensor nodes are distributed within a 20 m (L) × 15 m (W) × 2.3 m (H) cuboid. For the two related algorithms, the chargers are deployed at grid points on the deployment plane at height of 2.3 m, and the grid point separation is 1.8 m. The charging distance is 3 *m*, and the charging angle is 30°. It is noted that 30 experiments were executed per simulation case for averaging the simulation results.

TABLE II. SIMULATION SETTINGS

WRSN cuboid	20 m (L) × 15 m (W) × 2.3 m (H)
Number of sensor nodes	50, 100, 150, 200, or 250
Height of chargers	2.3 m
Average power consumption per sensor	0.6 mW, or 0.6 mW ~1.4 mW
Number of particles, m	200
Inertial weight, w	2
Cognitive factor, $c1$	2
Social factor, $c2$	2
PSO iterations, B	100

Fig. 9 shows the comparisons of NB-GCS, PB-GCS and PSCD in terms of the number of the chargers needed to satisfy the energy requirement of 0.6 mW. By Table I, 0.6 mW energy requirement implies 1-coverage requirement for NB-GCS and PB-GCS under the charger cone of the charging distance of 3 m and the charging angle of 30°. Fig. 10 shows the comparisons of NB-GCS, PB-GCS and PSCD in terms of the number of the chargers needed to satisfy the energy requirement of 0.6 mW ~ 1.4 mW. By Table I, any energy requirements larger than 0.7 mW imply coverage requirements of 2 or more chargers (i.e., 2^{+}-coverage) for NB-GCS and PB-GCS under the charger cone of the charging distance of 3 m and the charging angle of 30°.

By Fig. 9 and Fig. 10, we can observe that PSCD significantly outperforms NB-GCS and PB-GCS in terms of number of chargers needed to meet the energy requirements of 0.6 mW and 0.6 mW~1.4 mW. Especially for the high energy requirement of 0.6 mW~1.4 mW, the superiority of PSCD over NB-GCS and PB-GCS is much more significant. This is because PSCD deploys chargers at any points on the deployment plane, but NB-GCS and PB-GCS deploy chargers at only grid points on the plane. This is also because PSCD checks for every sensor node if its energy requirement is satisfied by interpolating practical experimental data, but NB-GCS and PB-GCS do the checking by coverage comparison on the basis of the energy received at the extreme point *E* of the charger cone.

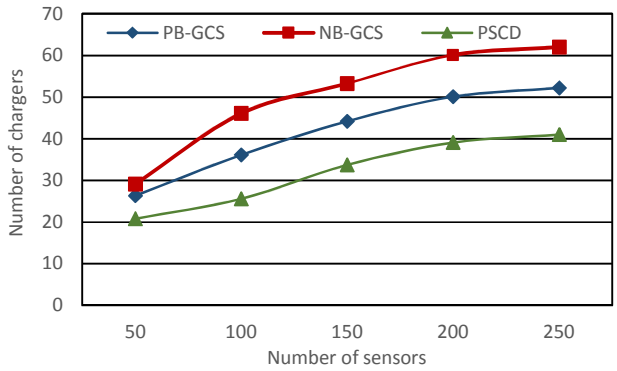

Fig. 9. The comparisons of NB-GCS, PB-GCS and PSCD in terms of the number of the chargers for the energy requirement of 0.6 mW.

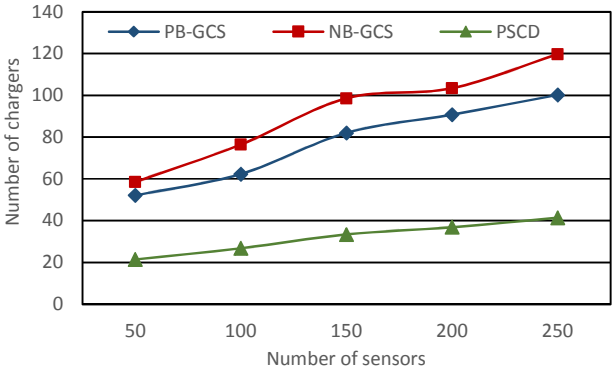

Fig. 10. The comparisons of NB-GCS, PB-GCS and PSCD in terms of the number of the chargers for the energy requirement of 0.6 mW~1.4 mW.

V. Conclusion

In his paper, we propose the PSCD algorithm using the PSO concept to nearly optimize WRSN charger deployment. On the basis of PSO, the PSCD algorithm utilizes the local optimal result and the global optimal result to adjust locations and antenna orientations of chargers to make WRSNs sustainable. We perform experiments and simulations to compare the PSCD algorithm with two related heuristic greedy algorithms, namely the PB-GCS algorithm and the NB-GCS algorithm. The comparison results show that the proposed PSCD algorithm outperforms the other two algorithms in terms of the number of chargers needed to make WRSNs sustainable. In the future, we plan to apply the meta-optimization concept to further improve the solution by automatically adjusting the PSO parameter settings.

Acknowledgment

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

References

[1] J.-H. Liao, and J.-R. Jiang, "Wireless charger deployment optimization for wireless rechargeable sensor networks," in *Proc. of the 7th International Conference on Ubi-Media Computing (UMEDIA 2014)*, 2014.

[2] Y. Tan, K. Panda, and S. K, "Review of energy harvesting technologies for sustainable wireless sensor network," *Sustainable Wireless Sensor Networks*, W. Seah, and Y. K. Tan, Eds., pp. 15–43,2010

[3] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Minimizing charging delay in wireless rechargeable sensor networks," in *Proc. of IEEE INFOCOM*, pp. 2922-2930, 2013.

[4] H. Dai, L. Xu, X. Wu, C. Dong, and G .Chen, "Impact of mobility on energy provisioning in wireless rechargeable sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*; pp. 962-967 , 2013.

[5] S. He, J. Chen, F. Jiang, D. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 1931-1942, 2013.

[6] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining sensor networks perpetually via wireless recharging mobile vehicles," in *Proc. of IEEE 39th conference on local computer networks (LCN)*, pp. 270-278, 2014.

[7] W. Xu, W. Liang, X. Lin, G. Mao, and X. Ren, "Towards perpetual sensor networks via deploying multiple mobile wireless chargers," in *Proc. of 43rd IEEE International Conference on Parallel Processing (ICPP)*, pp. 80–89, 2014.

[8] X. Ren, W. Liang, and W. Xu, "Quality-aware target coverage in energy harvesting sensor networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 8–21, 2015.

[9] X. Ye, and W. Liang, "Charging utility maximization in wireless rechargeable sensor networks," *Wireless Networks*, pp. 1-13, 2016.

[10] Powercast, "TX91501-3W-ID wireless charger user manual," available online: http:// www.powercastco.com/PDF/TX91501-manual.pdf (accessed on 30 May 2016).

[11] 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).

[12] J. Kennedy, and R. Eberhart, "Particle swarm optimization," *in Proc. of IEEE International Conference on Neural Networks*, 1995.

[13] J. A. Shaw, "Radiometry and the Friis transmission equation," *Am. J. Physics*, vol. 81, no. 33, pp. 33–37, 2013.