IEEE TRANSACTIONS ON
MOBILE COMPUTING

IEEE Xplore® Digital Library
http://ieeexplore.ieee.org/Xplore

Combining Solar Energy Harvesting with Wireless Charging for Hybrid Wireless Sensor Networks

Cong Wang, Ji Li, Yuanyuan Yang and Fan Ye

Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

***Abstract*—The application of wireless charging technology in traditional battery-powered wireless sensor networks (WSNs) grows rapidly recently. Although previous studies indicate that the technology can deliver energy reliably, it still faces regulatory mandate to provide high power density without incurring health risks. In particular, in clustered WSNs there exists a mismatch between the high energy demands from cluster heads and the relatively low energy supplies from wireless chargers. Fortunately, solar energy harvesting can provide high power density without health risks. However, its reliability is subject to weather dynamics. In this paper, we propose a hybrid framework that combines the two technologies - cluster heads are equipped with solar panels to scavenge solar energy and the rest of nodes are powered by wireless charging. We divide the network into three hierarchical levels. On the first level, we study a discrete placement problem of how to deploy solar-powered cluster heads that can minimize overall cost and propose a distributed $1.61(1+\epsilon)^2$-approximation algorithm for the placement. Then we extend the discrete problem into continuous space and develop an iterative algorithm based on the Weiszfeld algorithm. On the second level, we establish an energy balance in the network and explore how to maintain such balance for wireless-powered nodes when sunlight is unavailable. We also propose a distributed cluster head re-selection algorithm. On the third level, we first consider the tour planning problem by combining wireless charging with mobile data gathering in a joint tour. We then propose a polynomial-time scheduling algorithm to find appropriate hitting points on sensors' transmission boundaries for data gathering. For wireless charging, we give the mobile chargers more flexibility by allowing partial recharge when energy demands are high. The problem turns out to be a Linear Program. By exploiting its particular structure, we propose an efficient algorithm that can achieve near-optimal solutions. Our extensive simulation results demonstrate that the hybrid framework can reduce battery depletion by 20% and save vehicles' moving cost by 25% compared to previous works. By allowing partial recharge, battery depletion can be further reduced at a slightly increased cost. The results also suggest that we can reduce the number of high-cost mobile chargers by deploying more low-cost solar-powered sensors.**

***Keywords*-Wireless sensor networks, solar energy harvesting, wireless charging, mobile data gathering, facility location problem, partial recharge.**

I. INTRODUCTION

Driven by the prevalent energy needs from mobile devices yet relatively stagnant battery technology, wireless charging took off recently as a convenient way to power small electronics. Its application in wireless sensor networks (WSNs) has been investigated [5]–[13]. By instrumenting wireless energy transmitters on mobile chargers (MCs) [5]–[10] or at strategic locations [11]–[13], sensors can be charged conveniently without wires or plugs. Although wireless charging is a promising technique that can power hundreds of nodes reliably, rising energy demands in the network also increase the risks of electromagnetic exposure [12]. As a result, energy transmitters must comply with standards from Federal Communication Commission and limit their emitting power to human safe power densities ($< 1mW/cm^2$ [3]). Nevertheless, nodes at data aggregation points (such as cluster heads in a clustered WSN) usually consume very high energy ($10-100mW$) due to data traffic. Thus limiting transmission power at wireless chargers can easily cause battery depletion and network interruption on such nodes.

In the meanwhile, there is another competitive technique for environmental energy harvesting that has low risk yet much higher power density. As shown in [21], among a variety of harvesting techniques, solar harvesting through photovoltaic conversion enjoys the highest power density ($15mW/cm^2$), which is renewable and risk-free. In practice, a solar panel commensurate with sensor's size is sufficient to meet the energy demands of cluster heads. However, availability of sunlight is subject to dynamics in the environment. Not only weather conditions would have a direct impact on the harvesting rates, but also a series of spatial-temporal factors such as sunrise, sunset times, locations and their surroundings would affect deployment decisions of harvesting sensors.

Realizing the pros and cons of both technologies, in this paper, we propose a hybrid framework to make use of their advantages and overcome their drawbacks. In the new framework, a majority of nodes are wireless-powered nodes (WNs) due to the low costs of charging coils. On the other hand, due to the relatively higher manufacturing and deploying costs, a small number of solar-powered nodes (SNs) are responsible for aggregating data. Normally, a fleet of MCs roam over the field to serve recharge requests from WNs and collect data from SNs. In contrast to WNs, SNs' energy from the ambient source is self-sufficient. This scheme provides effective energy replenishment at cluster heads so that they can complete high volume data transmissions. Meanwhile, the rest of WNs can be recharged by MCs on demand. The hybrid framework raises several new challenges. First, how many SNs are needed and where should we deploy them such that the total cost is minimized? Second, how to guarantee robustness of the network when sunlight is unavailable (e.g., cloudy/raining days)? Third, how to schedule the MCs to complete wireless charging and data gathering in the same tour? Can we further optimize system cost and improve network performance compared to previous approaches?

To answer these questions, in this paper we first study a placement problem in discrete form where SNs are deployed among the known WN locations. We formulate it into a *facility location problem* [26]–[29] to minimize the total cost of packet routing and node deployment. Due to its NP-hardness, we use the primal-dual method to develop a distributed $1.61(1+\epsilon)^2$-factor algorithm suitable for WSN applications based on the centralized paradigm in [28]. Then we show the locations of SNs can be further optimized within a cluster in continuous space and propose an iterative mechanism based on the Weiszfeld algorithm [30]. We also demonstrate how our algorithms can adapt to seasonal variations of sunlight by adjusting their locations accordingly. Second, we theoretically analyze network energy balance and propose a method to maintain such balance during cloudy/raining days. We find that using a smaller cluster size is effective to reduce energy consumptions and develop a distributed algorithm to appoint some selected WNs as temporary cluster heads until solar energy becomes available. Finally, we optimize MCs' routes for the joint wireless charging and mobile data gathering problem. Different from [5] in which MCs visit exact node locations, we point out that for data gathering, it is only necessary for MCs to move into SN's transmission range. Based on this observation, we give a polynomial-time route improvement algorithm that can take shortcuts through SN's neighborhood to further reduce the cost. Since some nodes may require expedited recharge due to limited lifetime, we allow the MCs to perform *partial recharge* rather

Digital Object Identifier 10.1109/TMC.2017.2732979

than to refill batteries to full capacity [5]–[10]. Our objective is to simultaneously maximize the time MCs spend in recharging and prevent nodes from energy depletion, which is formulated into a Linear Programming problem. For easy implementation on MCs, we propose an efficient algorithm based on the particular structure of the lifetime constraints and validate its near-optimality by extensive simulations.

The contribution of this paper is multi-fold. First, we propose a hybrid framework to overcome the constraints of wireless charging and environmental harvesting techniques. To the best of our knowledge, this is the first work dealing with WSNs using such hybrid energy sources. Second, we formulate the SN placement problem into a facility location problem and propose the first distributed $1.61(1+\epsilon)^2$-factor approximation algorithm for sensor applications. Third, we propose a method to maintain network robustness by reducing energy consumption. Fourth, we give a route improvement algorithm that saves an extra 25% moving energy on MCs and surpasses the algorithm in [40] by additional 5%. The algorithm can also be used in a general setting for the Traveling Salesmen Problem with Neighborhood (TSPN) and provide solutions very close to the exact solutions found by exhaustive search. We also give MCs more flexibility to only partially refill the battery in case of high energy demand and propose an efficient algorithm that yields solutions within 5% to optimality. Finally, we conduct extensive simulations to evaluate the performance of the framework compared to WSNs that are solely wireless-powered [5]–[13].

The rest of the paper is organized as follows. Section II conducts a literature review of the related works. Section III presents the network model and assumptions. Section IV studies the placement problem of SNs. Section V provides theoretical analysis and discusses how to maintain energy balance using WNs. Section VI optimizes MC's migration routes and explores partial recharge for further improvements. Section VII evaluates the new framework and Section IX concludes the paper.

II. Related Works

A. Wireless Charging

Wireless charging technology has developed at an unprecedented pace recently [1], [2] and its application has been considered in battery-powered WSNs [5]–[13]. In [5], optimization of wireless charging and mobile data gathering is studied by combining the two utilities on a single MC. Based on the products from Powercast [4], wireless charging is explored in [6] to evaluate the new impact on deployment patterns and packet routing. In [7], an $\mathcal{O}(k^2 k!)$ greedy algorithm (where k is the number of nodes) is developed to maximize network lifetime whereas the moving cost of wireless chargers is not considered. Upon realizing this problem, minimization of chargers' moving cost is considered in [8], [9]. In [8], a distributed real-time energy information gathering protocol is proposed first. Then based on the updated energy information, a weighted-sum algorithm considering nodes' lifetimes and MCs' moving costs is developed. The problem is further extended to jointly consider MCs' recharge capacities and sensors' dynamic battery deadlines in [9]. In [10], a joint routing and wireless charging scheme is proposed to improve network utilization and prolong network lifetime. Similarly, in [11], deployment problems of wireless chargers are studied to extend network lifetime.

However, since many wireless charging systems are radiation-based, exposure to radio-frequency energy implies potential health risks. The negative biological effects include increased possibility of tumor and other impairments. Therefore, the regulation limits the maximum transmitter output power to be under 1W [3]. In practice, the omnidirectional propagation of electromagnetic wave causes health risks in all directions. Due to regulations of emitting power level on a single charger, multiple wireless chargers are considered in [12], [13]. In [12], the problem of how to adjust the transmitting power of wireless chargers such that overall electromagnetic exposure does not exceed a threshold is studied. A charger placement problem is formulated to guarantee all the locations to satisfy the safety requirements. In [13], optimization of "useful" energy transferred from chargers to nodes under safety concerns is considered. However, since the accumulative emitting power from multiple chargers is still restricted, nodes cannot perform energy-consuming applications.

B. Environmental Energy Harvesting

Environmental energy harvesting provides another alternative to extend network lifespan. Renewable energy from the environment such as solar, wind, vibration and thermal can be used effectively to power sensor nodes. Due to the dynamics in environmental energy sources, a majority of previous efforts focus on energy management of sensors [14]–[17]. In [14], a power management scheme to maximize sensors' duty cycles is proposed. Energy is profiled based on moving average to predict future income and the duty cycles are adjusted accordingly. In [15], the problem of maximizing quality of coverage is investigated for solar-powered WSNs. A nonlinear optimal control algorithm is proposed to dynamically allocate solar energy during the day and preserve energy for the night. Harvesting energy from light sources indoors is investigated in [16]. Energy allocation algorithms based on experimental measurements are developed to optimize energy storage on sensors. Joint energy management and resource allocation is considered in [17] for optimizing network performance. However, an inevitable drawback of these earlier works is that the network operations would be disrupted when those ambient energy sources are unavailable (e.g,. during cloudy/raining days in a solar harvesting network). In contrast, our proposed framework in this paper incorporates a combination of hybrid energy sources so that steady and productive network performance can be guaranteed.

C. Tour Planning of Mobile Vehicles

Tour planning of mobile vehicles for data collection in WSNs has been studied, see, for example in [18]–[20]. The problem shares similarities to the well-known Traveling Salesmen Problem with Neighborhood (TSPN) [37], [38], [40] in which the salesman aims to find the shortest tour through city neighborhoods of arbitrary shapes. In the context of WSNs, due to the omnidirectional propagation of electromagnetic waves, the neighborhoods are usually assumed to be circles. In [18], the tour planning problem is formulated into a mixed-integer program and a spanning tree covering algorithm is proposed. However, the algorithm requires the vehicles to visit exact node locations for data gathering, which is usually unnecessary in practice. If the node's transmission range is considered, the performance can be further improved. The method proposed in [19] attempts to improve current solutions for TSPN. It first determines the shortest TSP routes among sensors without considering their transmission ranges. Then it searches along transmission boundaries to find the best hitting points such that the tour length is minimized. In [20], a progressive tour construction method is proposed. It exploits the overlaps of transmission ranges of neighboring nodes so that the mobile vehicle can take shortcuts for cost savings. This method works effectively when there exist overlaps among nodes' transmission ranges. Different from [20] where nodes' transmission ranges may have overlaps, in this paper, multi-hop clusters are formed such that one-hop transmission neighborhoods of cluster heads (SNs) are disjoint. In addition, different from all the previous works, a migration tour may incorporate both WNs and SNs in our framework. Therefore, we focus on how to minimize tour lengths

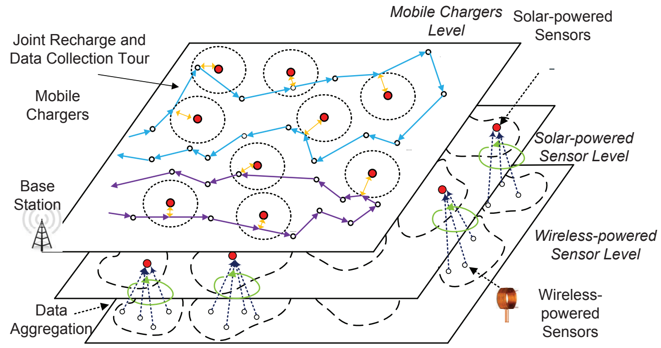


Fig. 1. Overview of a three-level network hierarchy.

by taking advantage of SNs' neighborhoods in a joint wireless charging and data gathering tour.

III. Network Model and Assumptions

In this section, we give an overview of the network model and assumptions of the new framework. Based on the energy sources, there are two types of nodes in the framework: wireless-powered nodes and solar-powered nodes. For brevity, we denote them by "WNs" and "SNs" respectively. Based on the functionality of network components, we divide the network into three hierarchical levels as shown in Fig. 1: wireless-powered sensor, solar-powered sensor and mobile charger levels.

The bottom level (wireless-powered sensor level) has N WNs uniformly randomly distributed on a square field of side length L. Since charging coils can be cheaply manufactured, WNs are deployed in high density to perform basic sensing missions such as environmental readings, target tracking, etc. In particular, to monitor location-dependent solar radiation strength, each node has an illuminance sensor and reports its reading with other data to the base station. The energy consumption for transmitting a packet follows the widely adopted model [23], $e_t = e_0 + e_1 d_r^{\alpha}$, where e_t is the transmitting energy, e_0 and e_1 are the energy consumed in electronics and amplifiers, d_r is the distance between the transmitter and the receiver ($d_r \leq r$, r is the transmission range), and α is the path loss exponent. To perform sensing tasks, sensors also consume e_s energy for each packet. For message exchange, we assume the network is connected. We also assume an event-driven data generation model [22]. Each WN generates packets independently following a Poisson process with average rate λ. Each WN is powered by a 780 mAh rechargeable NiMH battery and the recharge time T_r is 78 minutes [24]. If the energy drops below a threshold, e.g., 50%, it sends out a request to the MCs for scheduling energy replenishment.

The middle level (solar-powered sensor level) is comprised of self-sustaining, energy harvesting nodes. Normally, when solar energy income is sufficient, SNs act as cluster heads for aggregating sensed data. However, when energy supply is not enough during cloudy/raining days, the network re-selects WNs as cluster heads so they can rely on consistent wireless energy supply from the MCs. To minimize routing and deploying costs, SNs should be deployed at advantageous locations. Due to varying nature of sun's angles during a year, building obstructions and tree shades may exhibit different spatial-temporal patterns. Therefore, SN locations should be re-calculated based on the updated data once in a while (e.g., several weeks). The energy harvesting rates are modeled according to [33], which will be discussed in Section V. We assume the size of solar panel is chosen to be large enough to harvest enough energy for aggregating all the data. We consider a commercially available panel of $10 \times 10 cm^2$ size which is connected to a 3V, 2150 mAh lithium-ion battery.

The top level (mobile charger level) manages a fleet of m MCs through the base station. Proposed in [5], MCs are equipped with high-capacity batteries and powerful antennas for energy replenishment and data collection. Coordination among the MCs is conducted via long range communications to exchange status, position, energy request, etc. They also have positioning devices (e.g., GPS, gyroscope, etc) to locate sensor positions so that they can approach them in close proximity for wireless charging at high efficiency. We assume each node is only recharged by one MC at a time. Depending on updated geographical solar energy distribution, MCs can deploy SNs at appropriate locations. In case some nodes in the recharge sequences are bound to deplete their energy, the MCs can expedite the process by partially refilling nodes' batteries. Since SNs and MCs are the main components to sustain network operations and their manufacturing costs are much higher than WNs, we assume their monetary expenses are p_s (for SNs) and p_m (for MCs), respectively and $p_s < p_m$. In this paper, we have made additional assumptions: 1) The duration of MC's tour is dominated by battery recharging time; 2) Data is aggregated at the SNs by the shortest path routing (Dijsktra's algorithm) so routing paths are fixed; 3) For scalability, WNs do not upload their data to the MC directly. Finally, we summarize the important notations in Table I.

TABLE I
List of Notations

Notation	Definition
N	Number of wireless-powered sensors
s	Number of solar-powered sensors (calculated by algorithm)
m	Number of mobile chargers
L	Side length of squared sensing field
r	Transmission range of sensor nodes
e_t, e_r, e_s	Energy consumption to transmit, receive, generate a packet
λ	Average packet rate of Poisson distributed traffic
p_s, p_m	Monetary expenses of SNs and MCs, respectively
C_h	Battery capacity of sensor nodes
T_r	Recharge time of sensor's battery from zero to full capacity
v	Average moving speed of MCs

IV. Solar-powered Sensor Layer: Placement Problem

In this section, we study the Solar-powered sensor Placement Problem (SPP). It determines where to place SNs such that the total cost is minimized. In SPP, there are two types of costs: packet routing cost and sensor deploying cost. According to the energy model on packet transmissions [23], the routing cost is proportional to the number of hops thus the distance to the cluster head. The deploying cost is related to expense p_s and the strength of sunlight at a specific location. Since harvested energy exhibits slow variation due to seasonal changes of sun's angle, solar radiations at a fixed location also change slowly during a year. According to the illuminance readings from sensors, we denote the average solar strength at sensor location i by l_i. The deploying cost can be defined as the ratio p_s to l_i, which can be explained as the price we pay to gather a unit of solar energy from a specific location. If more SNs are deployed, nodes would have less relaying distance to the SNs. Thus, less routing cost can be achieved. On the other hand, more SNs would increase the deploying cost so our objective is to minimize the sum of routing cost and deploying cost.

These observations suggest that our problem is in close analogy to the classic *Facility Location Problem* (FLP) [26]–[29], which is NP-hard. In FLP, a set of facilities and cities are given. There is an opening cost associated with each facility and a transportation cost between any pair of facility and city. The goal is to connect each city to an open facility while minimizing the sum of transportation and opening cost. Due to NP-hardness, obtaining an optimal solution in polynomial time is infeasible. In practice, approximation algorithms that can achieve certain factors to the optimal solution are always preferred. After the first polynomial time 3.16-approximation algorithm [26], there has been encouraging progress in improving the approximation ratio and running time. An $\mathcal{O}(n^2 \log n)$ algorithm is proposed in [27] with an approximation ratio of 3. This bound is soon improved by [28] from 1.86 to 1.61 which is very close to the upper limit 1.46-ratio that polynomial-time algorithms can achieve [29].

However, the aforementioned efforts only focus on centralized algorithms whereas distributed implementation of FLP is rare in the literature. In dynamic wireless environments, a centralized algorithm requires the collection of variables across multiple dimensions to form global knowledge, which is usually time-consuming and not cost-effective. To this end, we propose a distributed $1.61(1+\epsilon^2)$-approximation algorithm based on the centralized approach in [28]. Next, we first formalize SPP and illustrate the centralized 1.61-approximation algorithm. Then we propose a distributed version of the algorithm. Since the locations of SNs are not necessarily constrained to WN locations, we further improve the solution using the Weiszfeld algorithm [30]. Finally, we discuss how to re-deploy SNs in order to adapt variations in solar strength at different times.

A. Placement of Solar-powered Sensors

In this subsection, we formalize the problem and describe both the centralized and distributed versions of the algorithm.

1) Centralized Placement Algorithm: First, let us formalize SPP. We denote the sets of SNs and WNs by $\mathcal{S}$ and $\mathcal{N}$, respectively. For simplicity, we first study discrete SPP by assuming SNs can only be co-located at WNs' locations, $\mathcal{S} \subset \mathcal{N}$. We consider a graph $G = (V, E)$ where vertices are sensor nodes and edges are connections. c_{ij} is the routing cost between nodes i and j, which is the energy consumed for transmitting packets. f_i is the deploying cost of SN i, and $f_i = p_s/l_i$, where l_i is the solar strength at node i. Since the energy consumed by WNs for data transmissions ultimately comes from the MCs, to convert c_{ij}'s energy units into monetary cost, we scale c_{ij} by how much the base station has paid for consuming per watt of energy to recharge MC's battery. The decision variable x_{ij} is 1 if WN j is assigned to SN i; otherwise, it is 0. y_i is 1 if we place an SN at i; otherwise, it is 0. Initially, all WNs are candidate locations for SNs. Our objective is to minimize the total cost by finding the locations for SNs.

$$\mathbf{P1}: \quad \min \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{N}} c_{ij}x_{ij} + \sum_{i\in\mathcal{S}} f_i \tag{1}$$

Subject to

$$\sum_{i\in\mathcal{S}} x_{ij} = 1; j \in \mathcal{N} \tag{2}$$

$$x_{ij} \le y_i; i \in \mathcal{S}, j \in \mathcal{N} \tag{3}$$

$$x_{ij}, y_i \in \{0,1\}; i \in \mathcal{S}, j \in \mathcal{N} \tag{4}$$

Constraints (2) and (3) impose that each WN is only connected to one SN. A centralized 1.61-approximation algorithm is proposed in [28]. Since we will use it as a guideline for designing the distributed algorithm, we briefly describe the centralized algorithm below. For each SN i, we introduce a set $\mathcal{B}_i$ to represent its connected WNs ($\mathcal{B}_i \subseteq \mathcal{N}$). In each step, the algorithm selects the node i^* with the minimum average cost

$$i^* = \arg\min_{i\in\mathcal{N}} \big[\sum_{j\in\mathcal{B}_i} c_{ij} + f_i - \sum_{j\in\mathcal{B}'_i} (c_{i'j} - c_{ij}) \big] / |\mathcal{B}_i|. \tag{5}$$

Node i' is a deployed SN that WN j has already connected to. $\mathcal{B}'_i$ is the set of these already connected WNs which would be benefited by altering their connections to the new SN i. Hence, a saving of routing cost $\sum_{j\in\mathcal{B}'_i}(c_{i'j} - c_{ij})$ should be deducted from the total cost. To find the minimum average cost for each candidate SN i, we can sort the cost in an ascending order and select the least one. This would result in $|\mathcal{B}_i|$ WNs being chosen each time. After i^* is found, we deploy an SN at its location and update all the WNs in $\mathcal{B}_i \bigcup \mathcal{B}'_i$ to connect with node i^*. The iteration continues to add SNs until all WNs are connected to them. The centralized algorithm has $\mathcal{O}(N^3)$ complexity and is summarized in Table II.

TABLE II
CENTRALIZED 1.61-FACTOR SN PLACEMENT ALGORITHM

```
Input: Set of WN $\mathcal{N}$.
Output: Set of SN $\mathcal{S}$ and $\mathcal{B}_i$, $i \in \mathcal{S}$.
While ($\mathcal{N} \neq \emptyset$)
Find $i^* = \arg\min_{i\in\mathcal{N}} \big( \sum_{j\in\mathcal{B}_i} c_{ij} + f_i - \sum_{j\in\mathcal{B}'_i} (c_{i'j} - c_{ij}) \big) / |\mathcal{B}_i|$.
Deploy $i^*$, connect $\forall j \in \mathcal{B}_i \bigcup \mathcal{B}'_i$ to $i^*$, $\mathcal{N} \leftarrow \mathcal{N} - \mathcal{B}_i$.
End While
```

TABLE III
DISTRIBUTED $1.61(1+\epsilon)^2$-FACTOR ALGORITHM FOR WN j

```
If $j$ is not connected to any SN,
  send a message to $i$ with offer $a_j \leftarrow \max(a_j - c_{ij}, 0)$.
Else If $j$ is connected to a deployed SN $i'$,
  send a message to $i$ with offer $a_j \leftarrow \max(c_{i'j} - c_{ij}, 0)$.
End If
Raise offer $a_j \leftarrow (1+\epsilon)a_j$.
```

2) ($1.61 + \epsilon^2$)-factor Distributed Algorithm: To understand the nature of the problem, we first formulate the dual problem of **P1**. The introduction of dual variables will help design the distributed problem.

$$\mathbf{P2}: \quad \max_{j\in\mathcal{N}} a_j \tag{6}$$

Subject to

$$a_j - b_{ij} \le c_{ij}; i \in \mathcal{S}, j \in \mathcal{N} \tag{7}$$

$$\sum_{j\in\mathcal{N}} b_{ij} \le f_i; i \in \mathcal{S} \tag{8}$$

$$a_i, b_{ij} \ge 0; i \in \mathcal{S}, j \in \mathcal{N} \tag{9}$$

Here, we can think the dual variable a_j as a monetary offer from node j to the total expense for deploying an SN. Constraints (7)-(8) can be combined into $\sum_{j\in\mathcal{N}} \max(a_j - c_{ij}, 0) \le f_i$ for SN $i \in \mathcal{S}$. It means that if offer a_j is raised for all the WNs at the same pace, and at the moment the total offer minus the total cost is equivalent to f_i, an SN can be successfully deployed at i. This method is known as the *dual ascent procedure* [28] and its 1.61-factor approximation is proved in [28] following the centralized paradigm. Based on [28], we propose a distributed $1.61(1+\epsilon)^2$-approximation algorithm next, where ϵ is a small fraction greater than zero.

First, WNs will send out their offers to SNs. If a WN is not connected to any $i \in \mathcal{S}$, the value of the offer is set to $\max(a_j - c_{ij}, 0)$; otherwise, the value is set to $\max(c_{i'j} - c_{ij}, 0)$. At the other side, SNs receive the offering messages from WNs. If an SN j is not yet deployed while its received total offers $\sum_{j\in\mathcal{N}} \max(a_j - c_{ij}, 0)$ are greater than or equal to f_i, we can successfully deploy an SN at i. If j is deployed and the offer value $a_j = c_{ij}$, we connect j to i by sending a connection message to j. In the next round, WNs increase their offers a_j by a ratio of $(1+\epsilon)$. After the locations for SNs have been calculated, the WNs send out deploying requests to the MCs. Then an MC is dispatched from the base station to deploy SNs at their designated locations. The distributed algorithm on WNs and SNs is summarized in Table III and Table IV and has the following properties.

Property 1: In principle, the distributed and centralized algorithms are equivalent.

TABLE IV
DISTRIBUTED $1.61(1+\epsilon)^2$-FACTOR ALGORITHM FOR SN i

```
Receive offering messages from WNs.
If $i$ is not yet deployed AND $\sum_{j\in\mathcal{N}} \max(a_j - c_{ij}, 0) \ge f_i$
  deploy an SN at $i$'s location, $\mathcal{S} \leftarrow \mathcal{S} + i$.
  $\forall j \in \mathcal{N}$, $\mathcal{B}_i \leftarrow \mathcal{B}_i + j$. Connect $j$ to SN $i$.
Else If $i$ has been deployed AND $a_j = c_{ij}$
  connect $j$ to $i$, $\mathcal{B}_i \leftarrow \mathcal{B}_i + j$, $\mathcal{B}'_i \leftarrow \mathcal{B}'_i - j$.
  Send a connection request message to $j$.
End If
```

Proof: See Section X-A. ■

Property 2: The distributed algorithm terminates in $\mathcal{O}(\log_{1+\epsilon} f_m)$ rounds, where $f_m = \max f_i, i \in \mathcal{N}$. The total message overhead is $\mathcal{O}((\log_{1+\epsilon} f_m)N^2)$.

Proof: Clearly, when the offering amount a_j increases at a rate $(1+\epsilon)$, reaching the maximum value of f_i requires $\mathcal{O}(\log_{1+\epsilon} f_m)$. In each round, the message overhead is bounded by $\mathcal{O}(N^2)$ so the overall message overhead is $\mathcal{O}((\log_{1+\epsilon} f_m)N^2)$. ■

Property 3: The distributed algorithm achieves $1.61(\epsilon+1)^2$-factor approximation to the optimal solution.

Proof: See Section X-B. ■

B. Placement of SNs in Continuous Space

Since MCs enjoy the freedom to place SNs at any feasible location in the field, in this subsection, we explore the placement of SNs in continuous space. Clearly, the continuous problem is much harder than its discrete version (SPP). Thus we start from the discrete results and relax them into the continuous domain. Intuitively, the asymptotic behavior of the discrete problem should approach the continuous problem when the number of nodes is infinite. However, in our problem, the number is limited. Thus discrete results from the SPP provide a feasible, sub-optimal solution to the continuous problem.

Our objective is to re-locate SNs in the continuous domain so the total cost which consists of the intra-cluster routing cost and the deploying cost is minimal. We find that, in most cases, variations of geographical solar radiation are quite small in a cluster (unless some spots are covered by shades) so the total cost is usually dominated by the routing cost. Changing SNs to new locations might reduce the routing cost (by ΔC) but lead to an increase of deploying cost (by ΔF). If the reduction in routing cost is higher than the increment in deploying cost ($\Delta C - \Delta F > 0$), there is still an extra saving in the total cost. To minimize the total cost, a naive approach is to divide the field into grids and enumerate through all possible locations. This method obviously requires enormous computation power and its accuracy also depends on the density of the grid.

We shift our focus to minimizing intra-cluster routing cost, and at the same time, we look for locations that offer the largest overall savings $\Delta C - \Delta F$. From SPP, a set $\mathcal{S}$ is obtained ($\mathcal{S} \subseteq \mathcal{N}$) with the corresponding cluster set $\mathcal{B}_i$, $\forall i \in \mathcal{S}$. We observe that the problem resembles the well-known *Weber problem* [30], [31] which finds the geometric mean for the set of $\mathcal{B}_i$ nodes. A well-known algorithm to solve the problem is to use an iterative procedure due to Weiszfeld [30]. In order to consider the deploying cost also, we introduce an additional step to the original Weiszfeld algorithm.

The algorithm is illustrated below. First, we initialize SN's location x_0 at node i's coordinates (from the output of SPP). The Weiszfeld algorithm uses a recursive function $W(\cdot)$ to find the optimal cost. The computation of the k-th iteration is,

$$W(x_k) = [\sum_{j=1}^{|\mathcal{B}_i|} \frac{\alpha_j}{c(x_k, j)}] / [\sum_{j=1}^{|\mathcal{B}_i|} \frac{1}{c(x_k, j)}] \qquad (10)$$

where $c(x_k, j)$ is the routing cost of transmitting a packet from location x_k to node j, and α_j are the coordinates of node $j \in \mathcal{B}_i$. The iteration continues by executing $x_{k+1} = W(x_k)$ until location changes are less than a small error bound ε. Since it has been proved in [31] that the Weiszfeld algorithm can converge to an optimum, the intra-cluster routing cost $C_k = \sum_{j=1}^{|\mathcal{B}_i|} \frac{1}{c(x_k, j)}$ for the k-th iteration is larger than C_{k+1} in the $(k+1)$-th iteration. However, the corresponding deploying cost may not share the same property due to the relative randomness in geographical solar energy distribution. Thus, an additional step of finding the minimal deploying cost $k_m = \arg\min_{1 \le i \le n} (C_i + F_i)$ is needed when the iteration is over. If the initial location has the minimal cost, the SN does not need to change its location. Otherwise, we re-locate the SN to x_{k_m} as the final solution. The algorithm is summarized in Table V.

TABLE V
EXTENDED WEISZFELD ALGORITHM FOR A CLUSTER

```
Initialize SN's location $x_0$ to $i$'s coordinates obtained by SPP.
While $x_{k+1} - x_k > \varepsilon$
  $W(x_k) = [\sum_{j=1}^{|\mathcal{B}_i|} \frac{\alpha_j}{c(x_k,j)}] / [\sum_{j=1}^{|\mathcal{B}_i|} \frac{1}{c(x_k,j)}]$, $x_{k+1} = W(x_k)$.
  Record deploying cost $F_k$ at $x_k$.
End While
$n \leftarrow k$ is the number of iterations until convergence.
Find $k_m = \arg\min_{1 \le i \le n} (C_i + F_i)$ and place SN at location $x_{k_m}$.
```

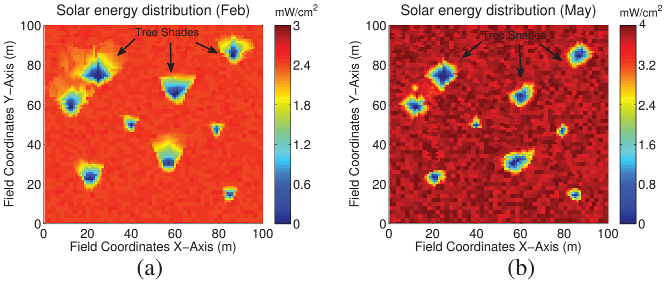


Fig. 2. Heatmaps of geographic solar energy distribution in different months. (a) February. (b) May.

Remarks: The approximation ratio for the discrete problem does not hold for the general continuous facility location problem. Unlike the discrete problem which we can formulate into an integer programming problem, the continuous problem usually involves nonlinear optimizations and complex iterative algorithms [32]. Our solution first utilizes the discrete solution for allocation of SNs (i.e., clustering WNs with SNs/determining the number of SNs). In this way, we can narrow down the search space of SNs within established clusters. Although this method may not be optimal, it offers a reasonable and fast solution for such an NP-hard problem.

C. Spatial-temporal Solar Variations

We now demonstrate spatial-temporal solar variations and explain how our proposed algorithm adapts to these changes. During different seasons of a year, the sun's angle towards earth surface varies slowly. Consequently, the harvested energy at different locations reflects such changes due to building obstructions, tree shades, etc. Fig. 2 shows the heatmaps of solar energy distributions on our campus gathered at different locations in February and May (Longitude at North 40°). We can see that the areas falling into tree shades are quite different. This has a direct impact on the deploying cost f_i at a location. Second, the strength of solar radiation also varies dramatically. In February, the maximum level is 3 mW/cm^2 and an increase of 13% is observed in May. These observations suggest that SN locations should be re-calculated after some time T_c. Otherwise, they might be covered in shades with limited harvesting capabilities.

Our algorithm fully exploits the distributed nature of WSNs. During operation, each node records solar strength at its location periodically and maintains a trailing average for the past T_c time. f_i is updated every T_c time accordingly. Once a new deployment is initiated, SNs' locations are calculated using the distributed algorithm. An example of the algorithm is shown in Fig 5(a). After their locations are found, an MC is dispatched to re-locate the corresponding SNs to designated locations. Then data is first gathered via multi-hop forwarding at the SNs and then collected by the MC during recharge.

V. WIRELESS-POWERED SENSOR LEVEL: MAINTAINING ENERGY BALANCE

In this section, we study the wireless-powered sensor level. Our main objective is to maintain network energy balance on WNs in different scenarios. First, we derive energy balance when SNs operate during sunny days. To facilitate our analysis, we denote the number of SNs obtained by the SPP algorithm as $s = |\mathcal{S}|$ and the maximum hop count from WN to its assigned SN as h. We assume that a total budget B for s SNs and m MCs, $sp_s + mp_m \leq B$. We explore the relationship between s and m given their manufacturing costs p_s and p_m ($p_s < p_m$). Second, during cloudy/raining days, energy balance might be broken. In this case, we further study how to regain such balance by refilling the energy gap. We propose to utilize several WNs to act as temporary cluster heads for aggregating data. A numerical range of WN cluster heads is first derived followed by a distributed algorithm to determine which WNs should be selected.

A. Energy Balance

First, let us consider energy consumptions in the network. For s shortest path routing trees rooted at SNs, the total energy consumption is

$$\begin{aligned} E_c &= \sum_{j \in \mathcal{N}} \big[\lambda(e_t + e_s) + \sum_{i \in \mathcal{C}_j} \lambda(e_t + e_r)\big]T \\ &\leq \sum_{i=1}^{h} \big[N_i(e_t + e_s) + \sum_{\substack{j=i+1,\\ i \neq h}}^{h} N_j(e_t + e_r)\big]\lambda s T \qquad (11) \\ &= \big[\big(\frac{2}{3}h^3 - \frac{1}{2}h^2 - \frac{1}{6}h\big)(e_t + e_r) + h^2(e_t + e_s)\big]\pi r^2 \rho \lambda s T \end{aligned}$$

where $\mathcal{C}_j$ is the set of child nodes of $j \in \mathcal{N}$, $N_i = (2i-1)\pi r^2 \rho$. The inequality holds because 1) a cluster can be estimated as a circle of radius $R = hr$ which consists of h concentric rings [25]; 2) summation of consumptions from all circle-shaped clusters has overlapping areas between neighboring clusters.

The harvested solar energy can be estimated by the empirical model proposed in [33]. The model provides a year-round analysis of solar radiations from weather stations and relates power levels to a quadratic equation on the time t of the day,

$$E = (a_1(t + a_2)^2 + a_3)(1 - \sigma). \qquad (12)$$

The shape of Eq. (12) is determined by parameters $a_1 - a_3$ that vary seasonally for different months. For example, for the month of May, $a_1 = -1.1$, $a_2 = -13.5$ and $a_3 = 43.5$. t_1 and t_2 are the respective time of sunrise and sunset ($t_1 = -\sqrt{-\frac{a_3}{a_1}} - a_2, t_2 = \sqrt{-\frac{a_3}{a_1}} - a_2$). σ is the percentage of cloud cover from weather reports. For T days, energy harvested by SNs is

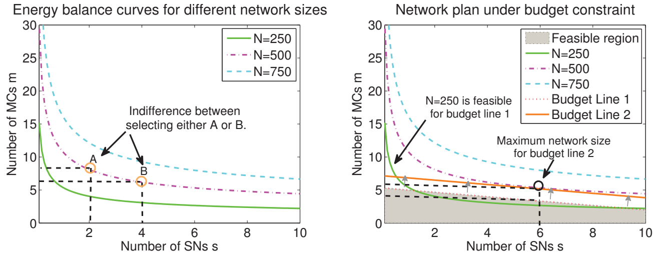
$$E_s = s \sum_{i=1}^{T} \int_{t_1}^{t_2} \big[a_1(t + a_2)^2 + a_3\big](1 - \sigma_i)dt \qquad (13)$$

The wireless energy replenished by MCs into the network is governed by the battery charging rates C_h/T_r. Thus, the amount of wireless energy replenished by m MCs in T can be calculated by $E_w = (mTC_h)/T_r$, $m \neq 0$. Then network energy balance is achieved when

$$\begin{aligned} &E_c \leq E_s + E_w \\ &E_c < \big[\big(\frac{2}{3}h^3 - \frac{1}{2}h^2 - \frac{1}{6}h\big)(e_t + e_r) + h^2(e_t + e_s)\big]\pi r^2 \rho \lambda s T \\ &\leq s \sum_{i=1}^{T} \int_{t_1}^{t_2} \big[a_1(t + a_2)^2 + a_3)\big](1 - \sigma_i)dt + \frac{mTC_h}{T_r} \qquad (14) \end{aligned}$$

Since $\pi(hr)^2 s \geq L^2$, we have $\sqrt{L^2/(sr^2\pi)} \leq h$. By plugging it into Eq. (14) and taking approximation $e_t \approx e_r$, we obtain a relationship between s and m

$$\frac{LT_r e_t \rho \lambda}{3\sqrt{\pi} C_h r}\big(\frac{4L^2}{\sqrt{s}} - \pi r^2 \sqrt{s}\big) - Xs + \big(\frac{1}{2}e_t + e_s\big)\frac{L^2 \rho \lambda T_r}{C_h} \leq m, \qquad (15)$$



(a) (b)

Fig. 3. Network plans under budget constraints. (a) Energy balance curves for different network sizes. (b) Optimal choices under budget constraints.

where X is

$$X = \big[\frac{T_r}{C_h T} \sum_{i=1}^{T} (1 - \sigma_i)\big]\big[\frac{a_1}{3}t^3 + a_1 a_2 t^2 + (a_1 a_2^2 + a_3)t\big]\Big|_{t=t_1}^{t=t_2}. \qquad (16)$$

B. Analysis of Budget and System Cost

The relationship between s and m in Eq. (15) can be explained graphically. Fig. 3(a) shows a group of energy balance curves when $N = 250 \sim 750$. Any point on a curve serves the same purpose for balancing network energy and there is no preference between choosing SNs or MCs as long as the balance holds. In fact, these curves can also be interpreted as the *indifference curves* in microeconomics. An indifference curve shows a collection of different goods between which the consumer is indifferent and every point on the curve results in the same utility. For example, when $N = 500$, point A requires 2 SNs and 8 MCs, which is equivalent to point B of 4 SNs and 6 MCs. Note that keeping adding SNs reduces MCs at a diminishing marginal rate. This is because that when the number of SNs is small, adding more SNs helps alleviate routing cost significantly (saving energy); however, this benefit gradually diminishes as more SNs are deployed.

Based on the budget, we can find whether a network plan is feasible to maintain energy balance. Since s is determined by the SPP algorithm, the corresponding number of MCs m can be found from the budget line $m = -\frac{P_m}{P_s}s + \frac{B}{P_s}$ (point (s, m)). If this point is above a balance curve, it means that the corresponding network size can satisfy energy balance. For example, point $(6, 3)$ on budget line 1: $m = -\frac{1}{3}s + 5$ is above the balance curve of $N = 250$ which indicates that the selection of $s = 6$, $m = 3$ is feasible. The feasible region for budget line 1 is marked as the shaded area. Furthermore, to find the maximum network size a given budget can sustain, we gradually increase N until point (s, m) is no longer above the balance curve. For point $(6, 5)$ on budget line 2, the maximum network size is $N = 500$. As the network size increases, the budget should be increased as well, which is to shift the budget line upward in Fig. 3(b). In this way, network administrators can quickly find an appropriate network plan, given the budget and available choices of SNs and MCs.

C. Adaptive Re-selection of Cluster Heads

An imperfection of solar energy is that sunlight is not always available. For example, during raining seasons, the network could experience consecutive cloudy or raining days and SNs are unable to harvest enough energy. To sustain network operation, our framework should adaptively switch cluster heads to WNs for aggregating sensed data. In this section, we first discuss how to maintain energy balance in the absence of solar energy. Then, we propose an algorithm to re-select cluster heads among WNs.

1) Maintaining Energy Balance: Since the number of MCs m is fixed for a network plan, we cannot expect more energy income from the energy replenishing side. To this end, we should reduce energy consumptions to restore energy balance. That is, the energy gap during cloudy days should be filled by reducing consumption for at least the same amount. Intuitively, introducing more cluster heads can effectively reduce energy consumption because more aggregation points would shorten packet relay paths. In other words, this is equivalent to having smaller k-hop clusters ($k < h$). The following property validates our intuition.

Property 4: For a network originally clustered by s SNs with cluster size $h > 2$, in the case that solar energy is unavailable, we can always restore network energy balance by reducing cluster size.

Proof: See Section X-C. ■

An anomaly is $h = 1, 2$. Because the original cluster sizes are already very small, we cannot reduce energy consumption further by having smaller clusters. In these cases, a notification message should be sent to request for more MCs. However, one-hop mobile data gathering only occurs in the worst case. Normally, we have $1 < k < h$ so our objective is to calculate how many WN cluster heads are needed given k. From the previous subsection, the solar radiation model indicates the energy harvested peaks when $\sigma \approx 0$ (perfect weather condition). Let us denote the number of new WN cluster heads by s' ($s' > s$). The maximum amount of energy harvested is E_s^* when $\sigma = 0$ in the ideal case. $E_c(h)$ is the energy consumption with h-hop clusters. $X_c(k)$ is the energy consumption for each k-hop cluster (plugging k into Eq. (11) and getting rid of s'). Since we require $E_s^* \leq E_c(h) - s'X_c(k)$, a range for s' is

$$\frac{L^2}{\pi(kr)^2} < s' \leq \frac{E_c(h) - E_s^*}{X_c(k)}. \tag{17}$$

By fixing k, any s' satisfying Eq. (17) will guarantee energy balance of the network. Next, we develop a distributed algorithm to find WN cluster heads.

2) Head Re-selection Problem: We now further explore the *Head Re-selection Problem* (HRP) which finds k-hop clusters with s' cluster heads satisfying Eq. (17). On one hand, since WN cluster heads will be traversed by MCs for data collection, the number of such nodes should be minimized to save MCs' moving cost. On the other hand, for heads to cover all the nodes within k hops, s' should be sufficiently large; otherwise, clusters will exceed k hops and more likely break the energy balance. Hence, our objective is to select a minimum number of heads and ensure that the shortest path from any node to its nearest head does not exceed k hops. It is not difficult to see that HRP is the *minimum k-hop dominating set problem* which is NP-hard [34].

A distributed algorithm for this problem is proposed in [34] for ad-hoc networks. The algorithm requires two rounds of k-hop message flooding for all the nodes. Since flooding is usually less preferred in energy constrained WSNs, we will not adopt the algorithm in [34]. Instead, we leverage the range in Eq. (17) as a basis for HRP. That is, as s' grows, hop distance from a node to its nearest head should decrease. Thus, we can start from the lower bound and increase s' iteratively until all the nodes are covered in k hops or the upper bound is reached. To find which WNs should become cluster heads, we extend the *furthest first traversal* algorithm proposed in [35]. The algorithm selects the node with the maximum distance from the current node to become the head in the next round. Unfortunately, the algorithm cannot be applied directly to our problem because: 1) it is centralized and not efficient to implement in distributed WSNs; 2) it may lead to inefficient selections. A new head might be chosen in the vicinity of an established one thereby causing a large overlap between neighboring clusters. This is not efficient and may also violate Eq. (17). Hence, we leverage the principle of *furthest first traversal* and propose a new distributed algorithm.

When gathered data at an MC indicates solar energy is not sufficient to support SNs, the MC sends a *head notification* message to any arbitrary WN whose battery has just been replenished and sets a counter to 0. The message specifies the cluster size k hops ($k = h - 1$ initially and decreased by 1 in each trial till $k = 1$).

TABLE VI

DISTRIBUTED HEAD RE-SELECTION ALGORITHM FOR WN i, $i \in \mathcal{N}$

```
MC sends *HeadMsg* to a WN (with enough energy), counter $c \leftarrow 0$,
sets HeadMsg.hop to $k(k < h)$. Set of cluster heads $\mathcal{H} \leftarrow \emptyset$.
**If** Recv(HeadMsg.ID $= i$ AND $c \leq \frac{E_c(h)-E_s^*}{X_c(k)}$)
  $d_{ij} = \min_{j \in \mathcal{N}}$ HopCount$(i, j)$ (Bellman-Ford-SPT($i$)),$\mathcal{H} \leftarrow \mathcal{H} + i$.
  Send *new routing msg* regarding new head $i$ to all the nodes.
  Set time-out period $T$ waiting for *resume* messages.
  **If** Recv(ResumeMsg.ID $= i$) within $T$
    $u = \arg\max \min_{j \in \mathcal{N}}$ HopCount$(i, j)$, $c \leftarrow c + 1$.
    Send *HeadMsg* to $u$.
  **Else**
    Clustering is completed and broadcast *complete msg*.
  **End If**
**Else If** Recv(NewRoutingMsg.ID is $i$)AND $\min_{j \in \mathcal{H}}$HopCount$(i, j) > k$.
  Send ResumeMsg to the new head.
**Else If** Recv(NewRoutingMsg.ID is $i$)AND $\min_{j \in \mathcal{H}}$HopCount$(i, j) \leq k$.
  Send JoinMsg to $u = \arg\min_{j \in \mathcal{H}}$ HopCount$(i, j)$,
  Declare as cluster member of $u$ ($\mathcal{B}_u \leftarrow \mathcal{B}_u + i, \mathcal{N} \leftarrow \mathcal{N} - i$).
**Else If** Recv(HeadMsg.ID $= i$ AND $c > \frac{E_c - E_s^*}{X_c(k)}$)
  $k \leftarrow k - 1$, broadcast a *restart* message.
**Else** Forward message according to routing entries.
**End If**
```

Upon receiving the head notification message, the WN declares itself as a new head and builds a shortest path tree (e.g., using Bellman-Ford algorithm [36]). Each node also maintains a routing entry to store minimum hop distance to a head. Those entries are updated when a new shortest path tree is formed. If a node j's entry indicates the minimum hop distance to a head i is less than or equal to k, it sends a *join* message to i to "join" the cluster as a member. Otherwise, it sends a *resume* message to node i to let the head selection continue. Within a timeout period, if the head receives a resume message, it means that there still exist some node(s) uncovered and the selection process should continue.

If a resume message is received, the head computes a shortest path tree using the Bellman-Ford algorithm. To avoid inefficient head selection, nodes should also report to the head whether they are cluster members or not. Then a new *head notification* message is generated and sent along the shortest path tree to the node with the maximum hop distance and enough battery energy which is not a cluster member yet. The counter is then increased by one. Otherwise, if no resume message is received during the timeout period, the head declares that clustering is successful by sending a *complete* message to all the heads. Upon receiving the complete message, heads report to the MC of cluster information. Note that if the counter exceeds the upper bound in Eq. (17), the current k is not feasible to maintain energy balance so it should be further decreased. In this case, the head should broadcast a message to restart the whole process and choose a smaller k. The psuedocode of this algorithm is given in Table VI. Based on *Property 4*, the distributed HRP algorithm can always find a set of cluster heads in $\mathcal{O}(hS)$ rounds and the worse case message overhead is $\mathcal{O}(hSN^2)$, where S is the upper bound in Eq. (17).

VI. MOBILE CHARGER LAYER: JOINT WIRELESS CHARGING AND MOBILE DATA GATHERING

In this section, we focus on optimizing trajectories of MCs. Proposed in [5], the instrumenting radio modules on MCs realize joint wireless charging and mobile data gathering on a single MC. This design certainly reduces manufacturing cost of MCs and system cost. However, because the effective wireless charging range is very limited (0.5-1m), the method in [5] requires the MC to stop at the exact WN location to perform simultaneous data gathering and recharge. However, in our framework, since SNs are powered by solar energy, it is only necessary for the MC to enter the transmission range ("touch" the transmission boundaries) to collect data from SNs. This creates opportunities to further optimize MCs' trajectories.

A. *Planning of Joint Data Gathering and Recharging Tours*

1) Initial Center Tour: We assume MC i has been assigned a touring sequence. The sequence defines an ordered set of nodes that starts from the base station b, traverses through WNs w_i and SNs (cluster heads) a_j, $w_i \in \mathcal{N}, a_j \in \mathcal{S}$, and finally returns to the base station for uploading data and recharging MC's own battery. Recharge scheduling algorithm proposed in [9] can be used conveniently to take recharging and data gathering requests together and calculate an initial touring sequence for each MC. Normally, the cluster size is larger than one hop ($h > 1$), and the transmission range around SNs form disjoint disks with identical radius. Since the initial sequence does not distinguish an SN from a WN and stops at the center of SN's transmission radius, we call it "Initial Center Tour" and denote its length as L_c. For such a tour with n WNs and s SNs, we have the following property.

Property 5: For an optimal tour with length L_r^*, when L_r^* is much larger than transmission range r, L_c is within $\left(1+\frac{8}{\pi}+\epsilon\right) \approx 3.55+\epsilon$ to the optimal L_r^*.

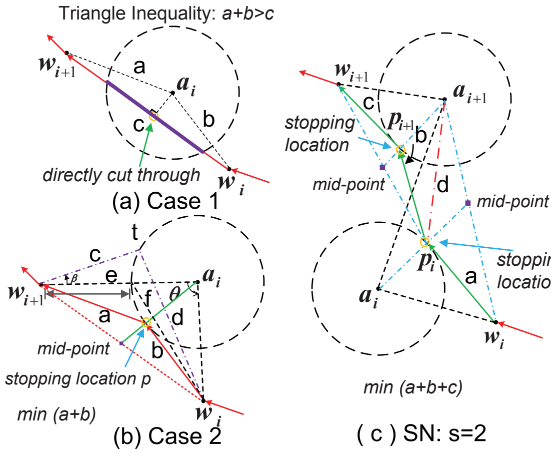
Proof: See Section X-D. ■

2) Exhaustive Search: Although the initial center tour guarantees a $(3.55+\epsilon)$-factor approximation, the solution can be further improved if the MC can take shortcuts through the disks. This problem is known as the *Traveling Salesmen Problem with Neighborhoods* and no efficient solution exists [37], [38]. However, in our problem, we can take advantage of WNs in the sequence and greatly reduce computation complexity. For all WNs in a sequence (and the base station), we order them in pairs (b, w_1), (w_1, w_2), ..., (w_n, b). For each pair (w_i, w_{i+1}), there could be at most s SNs in between. Let us start the analysis with $s = 1$. Fig. 4 shows that there are two cases: 1) the path connecting w_i and w_{i+1} directly cuts through the disk (Fig. 4(a)). In this case, the MC does not need to change directions. It only stops for a period of data uploading time (several minutes) in the disk. 2) the disk does not intersect with the path so there should exist a point on the boundaries of the disk that can minimize the path ($\min(a+b)$ in Fig. 4(b)). A naive approach is to divide the disk perimeter into l segments and find out which one yields the minimum distance. The method is used in [19] to find optimal hitting points on disk boundaries and its accuracy is proportional to l.

A closer look at Fig. 4(b) suggests a possible reduction of search space. Let us define the angle between lines connecting $w_i a_i$ and $w_{i+1} a_i$ as θ. For a point t on the arc outside the sector, there is an angle $\beta > 0$ between lines $w_{i+1} t$ and $w_{i+1} a_i$. Clearly, $c+d > e+f$ so any point t outside the sector gives an inferior solution compared to a point within the sector of θ. Thus, we can narrow down the search space to the points on the arc within angle θ between $w_i a_i$ and $w_{i+1} a_i$, so examination of only a fraction $f = \frac{\theta l}{2\pi}$ points is enough. Nevertheless, computation complexity of exhaustive search still grows exponentially when there are s SNs between w_i and w_{i+1} (with complexity $\mathcal{O}(f^s)$), thus a faster method is needed.

3) Minimizing Sum of Squared Distance: Exhaustive search quickly turns out to be impractical in reality. If the problem can be solved analytically, computational complexity can be greatly reduced. Since the expressions of distance involving square roots tend to yield intractable computations, we minimize the sum of squared distance instead. In fact, sum of squared distance has been used in many applications such as the well-known K-means algorithm [36]. The estimation error to the actual sum of distance will be evaluated by simulations. Likewise, we start our analysis from $s = 1$ and derive the following property.

Property 6: The point p on disk a_i that minimizes $d = (|w_i p|)^2 + (|w_{i+1} p|)^2$ is the intersection between line $w_m a_i$ and the disk, where w_m is the mid-point between the coordinates of w_i and w_{i+1}.



(a) Case 1

(b) Case 2

(c) SN: s=2

Fig. 4. Analysis of shortest path through feasible regions around SNs.

TABLE VII
ROUTE IMPROVEMENT ALGORITHM FOR MCS

```
Input: Sequence $\langle b, w_1, \ldots, a_i, \ldots, w_j, \ldots, w_n, b \rangle$.
Set of SNs between $w_j$ and $w_{j+1}$, $\mathcal{S}_j$, $\mathcal{S} = \bigcup_{j \in \mathcal{N}} \mathcal{S}_j$.
Output: Coordinates $(x_i, y_i)$ MC should visit near $a_i$.
While $\mathcal{S}_j \neq \emptyset$
  For $a_i \in \mathcal{S}_j$, find coordinates of WNs $w_j$, $w_{j+1}$ in sequence.
  If $a_{i+1}$ is also between $w_j$, $w_{j+1}$.$x_{j+1} = x_{a_{i+1}}$, $y_{j+1} = y_{a_{i+1}}$.
  Else $(x_{j+1}, y_{j+1})$ is set to $w_{j+1}$'s coordinates.
 End If
  Establish cartesian coordinate system originated at center of $a_i$.
  $x_i = (x_j + x_{j+1})r/\sqrt{(x_j + x_{j+1})^2 + (y_j + y_{j+1})^2}$,
  $y_i = (y_j + y_{j+1})r/\sqrt{(x_j + x_{j+1})^2 + (y_j + y_{j+1})^2}$.
  $\mathcal{S}_j \leftarrow \mathcal{S}_j - a_i$.
End While
```

Proof: See Section X-E. ■

Next, we consider the case of $s = 2$ without a direct cut as illustrated in Fig. 4(c). To use our method, computing each touching point on a disk needs two fixed points. For two disks, since the touching points can change simultaneously, minimization of sum of distance $(a+b+c)$ by considering multiple variables is very difficult analytically. Instead, we use the center of a_{i+1} as the reference point and calculate p_i on disk a_i to minimize $(a+d)$ first. Then, based on p_i, we calculate p_{i+1} on a_{i+1} to $\min(b+c)$. In this way, each computation only involves one variable. The method can be easily extended to the case when there are s SNs between w_i and w_{i+1}, so a total of $\mathcal{O}(s)$ computation is needed, which reduces the exponential-$\mathcal{O}(f^s)$ exhaustive search algorithm to linear time. We summarize the route improvement algorithm in Table VII.

B. *Optimizing Recharging Time*

We now optimize the allocation of MCs' recharge time for different WNs to avoid battery depletion. After the touring sequences have been found, the MCs need to traverse through WNs and the neighborhoods of SNs for wireless charging and data gathering. We assume the energy recharged into sensors' batteries is roughly proportional to the recharging time. In fact, most batteries exhibit such property as the longer they are being recharged, the more energy will be stored (under the battery capacity). An ideal situation is to replenish all the nodes to full capacity and hopefully no sensor depletes its battery before the recharge begins [8]. However, due to the nontrivial recharge time, the MCs have to reside at the WN locations for some time (e.g., 30-78 mins). This may easily cause energy depletion of subsequent nodes in the sequence.

1) Linear Program Formulation: We take a new approach that gives the MCs more flexibility so they can recharge more sensors in fixed time and sustain their battery energy at working levels.

That is, instead of fully replenishing sensors' batteries [5]–[10], we allow partial recharge. In case a node is bound to deplete its energy, the MC can expedite all recharge schedules before that node.

Let us denote the touring sequence found in Section VI-A3 by $\langle 1, 2, \cdots, i, \cdots, n \rangle$. There are two types of sojourn time at sensor nodes. For WN i, the MC stays for a period of t_i for recharge; for SN j, the MC spends a fixed time τ_j to collect data packets. $\tau_j = p_j/B$, where p_j is the amount of data at SN j's buffer and b is the data rate. A special case is when the MC's trajectory directly cuts through SN's transmission range (Fig. 4(a)). If $\frac{p_j}{b} \leq \frac{d_c}{v}$, $\tau_j = \frac{d_c}{v}$ where d_c is the length of chord within SN's transmission range. It means that if the data transmission time is less than MC's traveling time inside SN's transmission range, the MC can collect all the data without stopping. We denote MC's traveling time between two consecutive nodes in the sequence by $\tau_{i,i+1} = d_{i,i+1}/v$. Each WN i has a residual lifetime L_i. Our objective is to maximize the sum of recharge time under a pre-determined packet delay T_d. At the same time, we should also guarantee that all the recharge requests are met before their lifetime expirations. It can be formulated as a Linear Programming (LP) problem,

$$\mathbf{P4}: \quad \max \sum_{i \in \mathcal{N}} t_i \tag{18}$$

Subject to

$$\sum_{i=1}^{j-1} (\tau_{i,i+1} + t_i) + \sum_{i<j, i \in \mathcal{S}} \tau_i \leq L_j; 2 \leq j \leq n \tag{19}$$

$$0 < t_i \leq T_r; 1 \leq i \leq n \tag{20}$$

$$\sum_{i=1}^{n-1} \tau_{i,i+1} + \sum_{i=1}^{n} t_i + \sum_{i \in \mathcal{S}} \tau_i \leq T_d \tag{21}$$

Constraint (19) in fact contains $(n-1)$ constraints and each one ensures that for a sensor, the sum of all the time spent during recharging, data gathering and traveling for all previous nodes in the sequence does not exceed its lifetime. Constraint (20) states that the maximum recharge time is to replenish the battery to full capacity and Constraints (21) imposes a delay bound T_d for the entire recharge sequence.

2) Recharge Time Assignment: Although the problem can be calculated by standard LP solvers (e.g., using the simplex method), their worst case performance may take exponential time [41]. Thus, in this subsection, we develop a new algorithm by exploiting the particular structure of the problem. Since $\tau_{i,i+1}$ and τ_i can be calculated (constants) once the recharge sequence has been determined, we can simplify Constraints (19) and (21) as, $\sum_{i=1}^{j-1} t_i < L'_j$, $\sum_{i=1}^{n} t_i < T'_d$, where

$$L'_j = L_j - \sum_{i=1}^{j-1} \tau_{i,i+1} - \sum_{i<j, i \in \mathcal{S}} \tau_i, T'_d = T_d - \sum_{i=1}^{n-1} \tau_{i,i+1} - \sum_{i \in \mathcal{S}} \tau_i. \tag{22}$$

In other words, L'_j and T'_d represent the maximum recharge time from node 1 to $j-1$ and the entire sequence respectively. The level of partial recharge is assigned dynamically and there are generally two cases.

Case I: $\forall j \in \mathcal{N}$, $L'_j \geq (j-1)T_r$ and $T'_d \geq nT_r$. In this case, the MC can recharge all the nodes to full capacity, i.e., $t_i^* = T_r, \forall i \in \mathcal{N}$ and $\sum_{i \in \mathcal{N}} t_i^* = nT_r$.

Case II: $\exists j \in \mathcal{N}$, $L'_j < (j-1)T_r$ or $T'_d < nT_r$. In this case, recharge time prior to j may not take T_r so new assignments of recharge time should be performed. In fact, the maximization should take the value of a node's lifetime when its lifetime constraint is tight. Based on this observation and the iterative structure of Constraint (19), we propose a time assignment algorithm. It assigns recharge time proportional to the energy demands. The algorithm starts from the node with the minimum lifetime $j_m = \arg\min(L'_j, T'_d), j \in \mathcal{N}$. Then for node i in the sequence $(1 \leq i < j_m)$, an amount of $t_i = L'_{j_m} d_i / (\sum_{j=1}^{j_m-1} d_j)$ recharge time is assigned. d_i is the energy demand from node i. If assigned $t_i > T_r$, t_i should be bounded by T_r according to Constraint (20). The remaining time $t_i - T_r$ can be evenly distributed among the other nodes from $i+1$ to j_m. For the node's lifetime, we have the following property.

TABLE VIII
RECHARGE TIME ASSIGNMENT ALGORITHM

```
Initialize $j_s \leftarrow 1$, $j_m = \arg\min(\mathcal{L}', T')$
While $j_m \neq n$
  Initialize time variables $\varphi_i \leftarrow 0$ $(j_s \leq i \leq j_m - 1)$.
  For $i$ from $j_s$ to $j_m - 1$
    $t_i = \min(T_r, L'_{j_m} d_i / \sum_{j=j_s}^{j_m-1} d_j + \varphi_i)$.
    If $t_i = T_r$, for $i + 1 \leq k \leq j_m - 1$
      $\varphi_k \leftarrow \varphi_k + (L'_{j_m} d_i / \sum_{i=j_s}^{j_m-1} d_i - T_r)/(j_m - i - 1)$
    End
  End For
  $j_s \leftarrow j_m$, update $\mathcal{L}' \leftarrow \{L'_{j_m+1}, \cdots, L'_n, T'\}$
  Find the next, $j_m = \arg\min(\mathcal{L}', T')$
End While
```

Property 7: For the time assignment of node i prior to j_m in the recharge sequence $i < j_m$, the constraint $\sum_{k=1}^{i-1} t_k < L'_i$ holds.

Proof: The property can be proved since

$$\sum_{k=1}^{i-1} t_k = L'_{j_m} (\sum_{k=1}^{i-1} d_k / \sum_{k=1}^{j_m-1} d_k) < L'_{j_m} < L'_i. \tag{23}$$

■

Based on *Property 7*, we can see that once j_m has been selected and recharge time is assigned for all the nodes before j_m, the lifetime constraints still hold for these nodes. Hence, we can proceed to find the next node with minimum lifetime from (j_m+1) to n and repeat the same procedure. The iteration continues until the recharge sequence is exhausted. In most cases, the time complexity of the algorithm is $\mathcal{O}(n^2)$ since it needs to iterate through $\mathcal{O}(n)$ nodes and each one requires $\mathcal{O}(n)$ time assignments. In the worst case, the algorithm needs $\mathcal{O}(n^3)$ since an extra $\mathcal{O}(n)$ assignments are needed once the calculated recharge time exceeds T_r. The algorithm is summarized in Table VIII.

C. An Example of Proposed Hybrid Framework

Finally, we demonstrate a complete example of the proposed framework in Fig. 5. In Fig. 5(a), 8 SNs are placed to organize 250 WNs into clusters. Their initial locations calculated by the distributed SPP algorithm are marked by triangles and improved by the intra-cluster Weiszfeld algorithm shown as the crosses. In case of shortage of sunlight, Fig. 5(b) shows the results from the HRP algorithm to re-allocate cluster heads to WNs. Here, the loss in energy harvested from the 8 SNs can be compensated by introducing 13 WN cluster heads to reduce hop distance. For wireless charging and data gathering, Fig. 5(c) shows an initial center tour that covers 9 energy requests and 7 data uploading sites (SNs). The route is improved by our algorithm in Fig. 5 (d) with a saving of 17% moving energy on the MCs.

VII. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of the framework by a discrete-event simulator and compare it with a network solely relied on wireless energy [5], [9], [11]. In the simulation, all

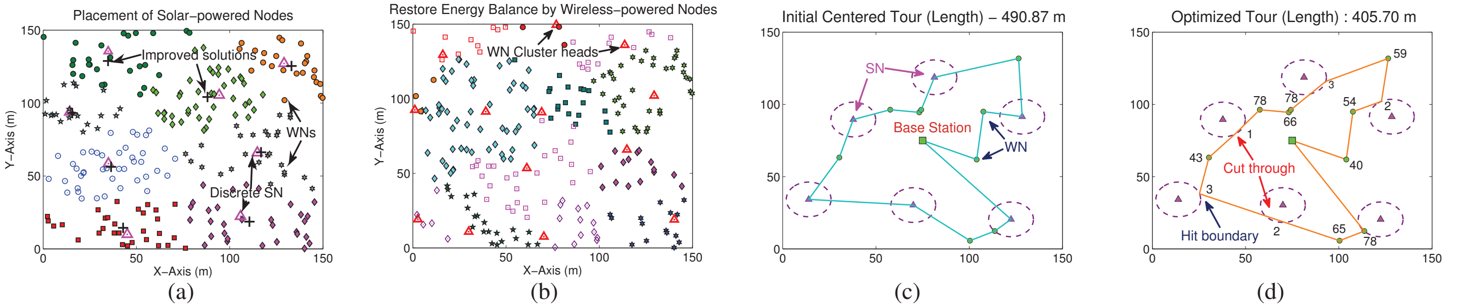


Fig. 5. A complete example of the framework. (a) Placement of SNs. (b) Restoring energy balance by WNs. (c) Initial center tour. (d) Optimized joint tour with MC's stopping time.

the cluster heads are replenished by the MCs in the *wireless-powered framework*. $N = 500$ nodes are uniformly randomly distributed over a square field of $L = 150$ m. Sensors have identical transmission range of $r = 12$ m, and consume $e_s = 0.05J$ for generating a sensing packet and $e_t = e_r = 0.02J$ for transmitting/receiving a packet [23]. Each time slot is 1 min. We have assumed an event-driven data generation model. Events occur at an average rate every 20 s and a packet is generated for each event. Thus, the Poisson traffic rate is 3 pkt/min. WNs have battery capacity $C_h = 780mAh$ and require $T_r = 78$ mins for recharge [24]. SNs have battery with larger capacity $2150mAh$ [42]. The MC has battery packs and weighs around 20 lbs. Based on the calculations from [43], the MC consumes energy at a rate of 5 J/m while moving at $v = 1$ m/s. We use real meteorological trace at Stony Brook, NY from [39], which has a complete archive of weather conditions. The simulation time starts from December and lasts for 12 months. The results are averaged over a number of 50 runs.

A. *Evaluation of Algorithms*

First, we evaluate the performance of the proposed algorithms.

1) Extended Weiszfeld algorithm: We validate the convergence property of the extended Weiszfeld algorithm for finding the SN locations in continuous space. For clarity, we randomly pick 4 SNs and trace the evolution of their total costs in Fig. 6(a). We normalize the total cost for better visualization. First, we observe that the Weiszfeld algorithm can converge extremely fast (within 5-6 iterations) and offer an additional 2-4% energy saving. Second, since the deploying cost is usually much less than the routing cost in our problem, the total costs are dominated by the changes in the routing costs. Therefore, the optimal values reach the minimum when the algorithm converges. Otherwise, the algorithm will use the minimum total cost during the iteration process as the final solution (explained in Section IV-B).

2) Route Improvement Algorithm: We evaluate the route improvement algorithm by comparing it with the algorithms proposed in [5], [40]. The algorithm in [40] continuously finds the closest hitting points on the boundaries of the disks and we call it *nearest insertion algorithm*. The tour passing through the disk centers is used in [5] for joint wireless charging and data gathering and we denote it by *initial center tour*. Fig. 6(b) compares MC's moving energy using the three methods. First, we can see that our algorithm provides an average of 25% energy saving compared to the initial center tour. In fact, more energy saving can be achieved with a larger transmission range since an MC only needs to visit the transmission boundaries for gathering data. Second, the results further indicate 5-7% improvements over the nearest insertion algorithm [40]. This is because that selecting the closest hitting point on a disk cannot guarantee that the sum of distance to the neighboring nodes is minimal. In contrast, our algorithm finds a point on the disk that minimizes the sum of squared distance. To examine the gap between minimizing the sum of squared distance and the actual distance, we conduct more evaluations in Fig. 6(c) by considering a joint route comprised of WNs and SNs. Since WNs outnumber SNs by a considerable amount, we maintain a 10 to 1 ratio between WNs and SNs. To provide a baseline, an exact solution is found by exhaustive search using [19].

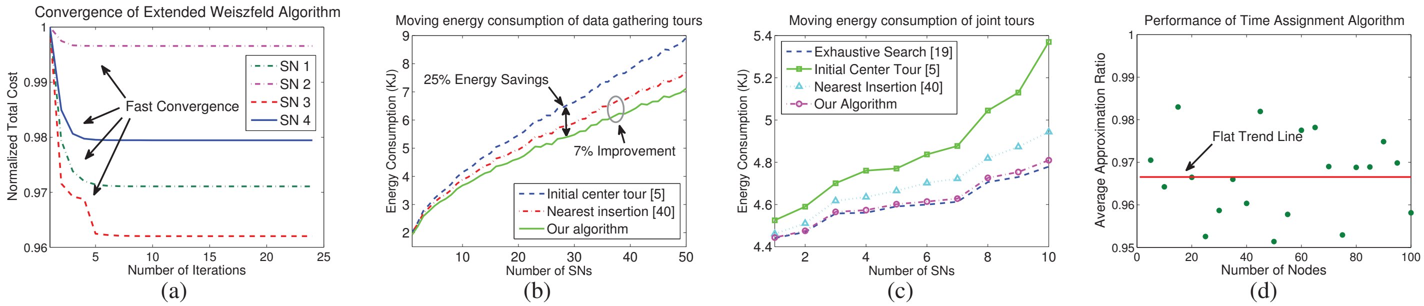
Hence, we examine this approximation ratio by simulations in Fig. 6(c), which shows "surprising" results that minimizing the sum of square distance has only 1% difference compared to the sum of actual distance in the joint data gathering and recharge problem. Recall that our route improvement algorithm (Table VII) has made an approximation by taking the mid-point which minimizes the sum of square distance of route segments. Surprisingly, this algorithm only has an average of 1% difference to the solution found by exhaustive search. We found it surprising because our linear-time algorithm achieves almost the same results as the exponential exhaustive search method.

In addition, for mixed WNs and SNs in a joint tour, our algorithm enjoys even more improvement. This is because that for the touring sequence that only consists of SNs, we have taken further approximation. As shown in Fig. 4(c), when there are more than 2 consecutive SNs in the touring sequence, we take the center of the second circle a_{i+1} as a reference point while calculating the hitting point on the first circle a_i whereas the optimal reference point should be on the arc of the second circle. Thus, more consecutive SNs may introduce some inaccuracies. However, for the joint tour, we have at most 2-3 consecutive SNs in our test so the performance is better than Fig. 6(b).

3) Recharge Time Assignment Algorithm: We also evaluate the performance of the recharge time assignment algorithm and compare with optimal solutions from standard LP solver. Recharge sequences from 5 to 100 nodes are evaluated with nodes' battery energy and lifetime randomly distributed. The results are averaged over 100 simulation runs. Fig. 6 (d) illustrates difference between our algorithm and the optimal solution. We observe that our algorithm is able to achieve results very close (within 5%) to the optimal LP solver. As we increase the number of nodes in the sequence, the approximation ratio does not degrade as indicated by the flat trend line. In the simulation, we discover that in most cases, our algorithm can find the optimal solution. However, in some special cases, there are several consecutive nodes with limited lifetimes that require partial recharge of previous nodes. Our algorithm conducts one recharge time assignment at a time until Constraints (19) and (21) for all the nodes are satisfied. In contrast, the optimal LP solver considers all the nodes together and assigns recharge time in one shot. For these cases, our approach may not always yield optimal results but remains very close to them.

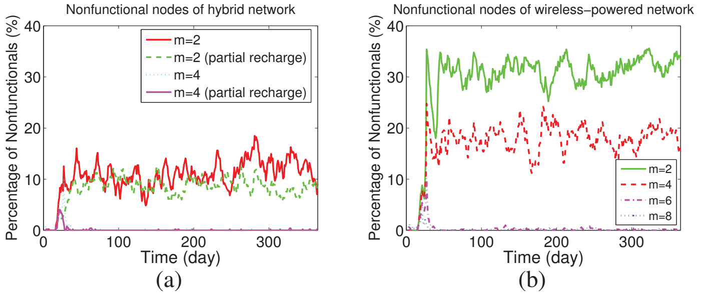
B. *Nonfunctional Nodes*

One of the key performance metrics for recharging is nonfunctional nodes. Once a node's battery is depleted, it stops working and becomes nonfunctional until its battery is replenished. To sustain perpetual operations, nodes should be alive all the time; otherwise,



(a) (b) (c) (d)

Fig. 6. Evaluation of the proposed algorithms. (a) Convergence of Weiszfeld algorithm. (b) Improvements of tour for only data gathering sites (SNs). (c) Improvements of joint wireless charging and mobile data gathering tour. (d) Average approximation ratios of the recharge time assignment algorithm.



(a) (b)

Fig. 7. Number of nonfunctional nodes. (a) Hybrid framework. (b) Wireless-powered framework.

they will degrade sensing qualities and node communications. Fig. 7 compares the percentage of nonfunctional nodes between hybrid (full or partial recharge) and wireless-powered frameworks [5], [9], [11]. The SPP algorithm generates $s = 11$ SNs. To compare the performance, we change the number of MCs m. Fig. 7(a) shows the results from the hybrid framework when $m = 2 \sim 4$. We can see that 2 MCs can keep the percentage of nonfunctional nodes around 10% and 4 MCs can almost achieve perpetual operations. Partial recharge offers even better performance with less than 10% nonfunctional nodes when $m = 2$.

In contrast, $m = 2$ for wireless-powered network results in 30% nonfunctional nodes in Fig. 7(b) and an increase to $m = 6$ still barely eliminates all battery depletions at equilibrium. These observations clearly demonstrate that the hybrid framework can improve network performance significantly. Since for a wireless-powered network, cluster heads consume energy much faster, MCs need to visit them more frequently, which reduces the chances for other nodes to get recharged. However, SNs are replenished by solar energy which has much higher power density so MCs have more leverage to take care of the rest of the network.

C. System Cost

The cost to maintain network operations is another important performance metric. In our framework, there are two types of costs. The first type comes from network maintenance which basically involves energy expenditures at the MCs while moving and recharging. Since the energy expense for recharging is necessary for sensor nodes, we focus on the energy cost while the MCs are moving. Fig. 8(a) and (b) trace the evolution of MCs' moving energy cost when our goals are maintaining nonfunctional percentage under 15% and 1%, respectively. First, let us examine the cost brought by partial recharge. Although it is not obvious on Fig. 8(a), by taking their mean values, we are able to see that partial recharge results in slightly higher cost (3.57 KJ vs. 3.49 KJ). This is because that partial recharge allows MCs to recharge more nodes in a fixed time period (MCs move more frequently). Second, we compare the moving cost to wireless-powered networks. Since it requires more MCs to maintain energy balance, their moving costs inevitably increase which are almost doubled if evaluated by their mean values (mean 6.91 KJ vs. 3.49 KJ of the hybrid network). Similar results are observed in Fig. 8(b) when the objective is to maintain nonfunctional percentage below 1%. The difference is that the gap between the two curves becomes even wider (mean value 36 KJ vs. 105 KJ), which indicates that the wireless-powered network almost needs 3 times the energy of the hybrid network. From these results, we can see that hybrid networks are more energy efficient for higher performance requirements.

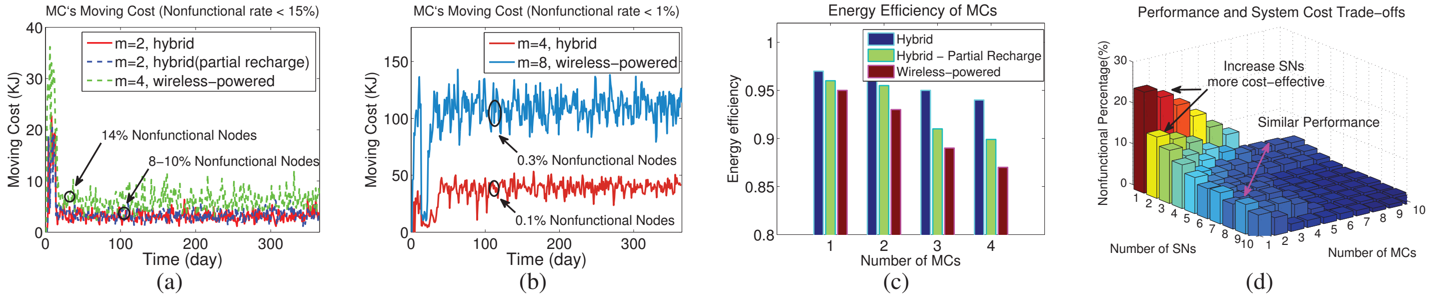
We have also evaluated energy efficiency for different numbers of MCs in Fig. 8(c). Energy efficiency is defined as the ratio between energy replenished and the total energy consumed on the MCs. The hybrid network enjoys the highest overall energy efficiency as more than 90% energy can be used for recharge while partial recharge has slightly lower efficiency due to possibly more movements. The wireless-powered network has the lowest energy efficiency since the MCs have to move even more often to recharge cluster heads.

The second type of cost is the fixed cost of SNs and MCs. The theoretical results in Section V indicate that by having more SNs, we are able to save the cost on MCs since they are usually more expensive. To see the trade-offs between the numbers of SNs and MCs, we show the impact on nonfunctional percentage in Fig. 8(d) in which the X and Y axes represent the numbers of SNs and MCs and the Z axis represents the percentage of nonfunctional nodes. We observe that introducing SNs is more cost-effective to reduce nonfunctional percentage compared to MCs. For example, when we have 1 MC and 1 SN, having 1 more SN helps reduce nonfunctional percentage from 23% to 13% while having 1 more MC still results in over 20% nonfunctional nodes. With more SNs, even fewer MCs can achieve similar performance. For example, 8 SNs and 1-2 MCs result in similar performance to 6-8 MCs and 1 SN. These results suggest that hybrid networks are more cost-effective and energy efficient than wireless-powered networks.

D. Harvested Energy and Message Overhead

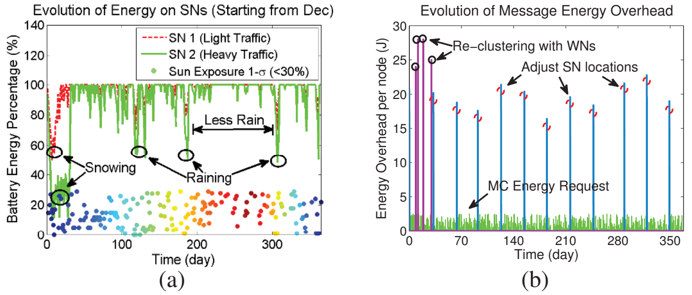
To validate our algorithm design, we evaluate the evolution of SN's energy and network message overhead. Fig. 9(a) traces SNs' energy with weather conditions represented by percentage of solar exposure $(1 - \sigma)$ obtained in [39]. We focus on two typical nodes with light and heavy data traffic. We can see that through the month of December, energy storage continuously declines due to weak solar strength in winter. In addition, there are also several consecutive snowing days so SNs are unable to harvest enough energy and re-selection of cluster heads among WNs is needed. This gives SNs opportunities to recover their energy (during 40-50 days). For the remaining simulation, although a few consecutive raining days are observed, the energy gaps are quickly filled. This is because that solar radiation has strengthened since spring and energy storage is sufficient to sustain network operations. For example, from summer to fall, SNs can maintain their battery energy over 80% due to favorable weather conditions and strong solar radiation.

Fig. 9(b) demonstrates the energy consumed by exchanging control messages such as SN clustering, re-selection of WN cluster



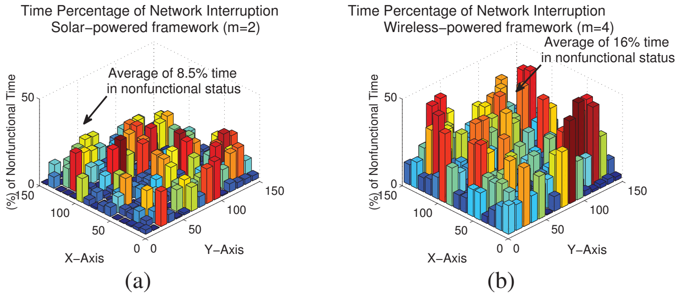
(a) (b) (c) (d)

Fig. 8. Evaluation of system cost. (a) Evolution of MC's moving cost when nonfunctional rate is less than 15%. (b) Evolution of MC's moving cost when nonfunctional rate is less than 1%. (c) Energy efficiency of MCs. (d) Trade-offs between performance and system cost.



(a) (b)

Fig. 9. Evolution of harvested solar energy and message energy overhead. (a) Energy variations of SNs. (b) Message energy overhead.



(a) (b)

Fig. 10. Geographical distributions of service interruption. (a) Hybrid framework $m = 2$. (b) Wireless-powered framework $m = 4$.

heads and energy information requests. For each month, MCs initiate a new calculation of SN's placement pattern to reflect the updated geographical solar radiations (e.g., Fig. 2). These message overhead is shown as the blue spikes at the beginning of each month. Note that in the simulation, when SNs' energy is less than 30% due to insufficient solar energy, they request to re-cluster by WNs. The overhead is represented by those higher spikes (in purple), which corresponds to the time when SNs' energy drops in Fig. 9(a). Our results indicate that re-clustering using WNs only occur during winter time though there are some consecutive raining periods in other seasons. From Fig. 9, we have validated that our algorithm can adapt to weather conditions effectively.

E. Geographical Distributions of Service Interruption

Finally, we examine geographical distributions of service interruptions. Our objective is to see how long nodes are in nonfunctional status and their geographical distributions. Since cluster heads are responsible for aggregating and uploading sensed data, their survivals are critical for the entire network. A breakdown may lead to severe packet loss, network interruption and extended data latency. For fair comparison, we use the results from Section VII-B and set $m = 2$ for the hybrid framework and $m = 4$ for the wireless-powered framework so that both cases have a similar number of nonfunctional nodes. From Fig. 10(a), we observe that the distribution of nonfunctional nodes is quite even in the hybrid framework. In contrast, nodes around the cluster heads (including the heads as well) are more prone to deplete battery energy in the wireless-powered network (20% more nonfunctional time in Fig. 10(b)). On average, a node in the hybrid framework has only 8.5% time in nonfunctional status whereas it would experience 16% nonfunctional time in the wireless-powered network. The sharp contrast is because that for the wireless-powered network, MCs need to not only take care of cluster heads but also their surrounding areas. This may cause the MCs to move frequently between head locations and overwhelm their recharge capabilities. However, for the hybrid framework, MCs do not need to recharge SNs so the resources can be re-distributed among WNs to reduce their nonfunctional rates.

VIII. Discussions

We now discuss a few interesting issues worth future study. First, for tractability, our solution breaks the optimization problem into many sub-problems. Indeed, system cost can be improved in a cross-layer fashion. For example, the planning of SNs could be associated with route management for the MC to reduce system cost. Such cross-layer optimization brings variables from different aspects to determine the number, allocation, location of sensors together with route planning of the MC. Second, whether the MC should visit the exact SN locations can depend on the SN's energy. In case it has low energy income, the MC should move close to avoid draining sensor's battery. We plan to study these interesting problems in the future.

Second, a fixed time period has been used to adjust SN's locations. A more efficient solution is to develop an adaptive scheme that can reflect solar variations with minimal overhead. If micro-climate solar logs can be obtained, we can predict spatial-temporal energy distributions based on historical data. Then we can schedule re-location more efficiently. However, these data are usually available at low resolutions (up to 10 km) [44]. Thus, in our future work, we will exploit the MC for gathering such solar data.

IX. Conclusions

In this paper, we consider a hybrid framework that combines the advantages of wireless charging and solar energy harvesting technologies. We study a three-level network consisting of SNs, WNs and MCs levels. First, we study how to minimize the total cost of deploying a set of SNs. The problem is formulated into a facility location problem and a $1.61(1+\epsilon^2)$-factor distributed algorithm is proposed. The solution is further improved by using intra-cluster Weiszfeld algorithm in continuous space. Second, we examine the energy balance in the network and develop a distributed head re-selection algorithm to designate some WNs as cluster heads when solar energy is not available during raining/cloudy days. Third, we focus on how to optimize the joint tour consisting of both wireless charging and data gathering sites for the MCs. A linear-time algorithm is proposed that can approach very closely to the exact solution and reduce at least 5% MC's moving energy compared to previous solutions. We also propose to partially refill sensors' energy to further reduce battery depletion and develop an efficient algorithm to solve the problem with high accuracy. Finally, based on real weather data, we demonstrate through simulations the effectiveness and efficiency of the hybrid framework that can improve network performance significantly.

X. Proofs of Properties

A. *Proof of Property 1*

We sequentialize the distributed algorithm into execution rounds. For the distributed algorithm, each round consists of a number of message sending and receiving by respective WNs and SNs. In each round, the total offers received from all the nodes are $\sum_{j\in\mathcal{N}} a_j = \sum_{j\in\mathcal{N}} c_{ij} + f_i$. For some nodes already connected, the new offers are $\sum_{j\in\mathcal{N}} a_j = \sum_{j\in\mathcal{N}} c_{i'j} - c_{ij}$, which should be deducted from the total offers to reflect the adjusted value. We can see that this result is exactly the term in (5). Since the offer value is increased at a rate $(1+\epsilon)$, an SN that meets the lowest total offer will be selected in the earliest time, which is equivalent to selecting the least average cost in the centralized algorithm. Therefore, we can see that the mechanism of the distributed algorithm is analogous to the centralized algorithm in [28].

B. *Proof of Property 3*

Our proof of *Property 3* is based on [28]. First, denote optimal offers in the centralized algorithm [28] by a_j and the distributed algorithm by a'_j, $j \in \mathcal{N}$. A *Factor Revealing LP* is constructed by [28]. For SN i, $k = |\mathcal{B}_i|$, the optimal solution is to solve the following maximization problem

$$\textbf{P3}: \quad z_k = \max \sum_{j=1}^{k} a_j \big/ (\sum_{j=1}^{k} c_{ij} + f_i) \tag{24}$$

Subject to

$$a_j \le a_{j+1}, \forall j \in \{1, 2, \ldots, k-1\} \tag{25}$$

$$\sum_{j=1}^{k} \max(a_j - c_{il}, 0) \le f_i, \forall l \in \{1, 2, \ldots, k\} \tag{26}$$

$$a_j \le a_l + c_{ij} + c_{il}, \forall j, l \in \{1, 2, \ldots, k\} \tag{27}$$

$$a_j, c_{ij}, c_{il}, f_i \ge 0, \forall j, l \in \{1, 2, \ldots, k\} \tag{28}$$

For the maximization problem to be bounded, a_j should also be bounded. It implies that at least one of the constraints of (26) and (27) is tight (i.e., changing from inequality into equality). *Case 1*: Eq. (26) is tight; *Case 2*: Eq. (27) is tight, thus a_l is also bounded.

For the distributed algorithm, we can formulate it into a similar Factor Revealing LP except that constraint (26) becomes $\sum\limits_{j=1}^{k} \max\left(\frac{a'_j}{1+\epsilon} - c_{il}, 0\right) \le f_i$ and constraint (27) becomes $\frac{a'_j}{1+\epsilon} \le a'_l + c_{ij} + c_{il}$ since increasing offers at the same pace in the centralized scheme would deploy SN i at most $(1+\epsilon)$ time earlier compared to the distributed algorithm. For Case 1, since $f_i \ge 0$ and the constraint is tight, $a_j - c_{il} \ge 0$ and $\frac{a'_j}{1+\epsilon} - c_{il} \ge 0$ hold for the centralized and distributed algorithms, respectively. Thus, $\frac{a'_j}{a_j} \le 1+\epsilon$. For Case 2: $\frac{a'_j}{1+\epsilon} = a'_l + c_{ij} + c_{il}$ and $\sum\limits_{l=1}^{k} \max(\frac{a'_l}{1+\epsilon} - c_{il}, 0) \le f_i$ is also tight to bound a'_l. The latter suggests that the ratio in Case 1 $\frac{a'_l}{a_l} \le 1+\epsilon$ can be applied here:

$$\frac{a'_j}{1+\epsilon} \le a_l(1+\epsilon) + c_{ij} + c_{il} \le (1+\epsilon)(a_l + c_{ij} + c_{il}) \tag{29}$$

Then by taking the ratio of Eq. (29) to Eq. (27), we have $\frac{a'_j}{a_j} \le (1+\epsilon)^2$ for Case 2. Since the approximation ratio to the optimal solution a^*_j is proved by [28], $\frac{a_j}{a^*_j} \le 1.61$. Thus, our algorithm has at most $\frac{a'_j}{a^*_j} \le 1.61(1+\epsilon)^2$ approximation to the optimal solution.

C. *Proof of Property 4*

By assumption, for SNs to successfully aggregate and transmit data, $E^*_s \ge sh^2r^2\pi\rho\lambda(e_t + e_r)T$. Plugging this into Eq. (14), we have

$$\frac{mC_h}{T_r} > \left[\left(\frac{2}{3}h^3 - \frac{3}{2}h^2 - \frac{1}{6}h\right)(e_t + e_r) + h^2(e_t + e_s)\right]\pi r^2 \rho\lambda s \tag{30}$$

For $h > 2$, $\frac{2}{3}h^3 - \frac{3}{2}h^2 - \frac{1}{6}h > 0$ so $\frac{mC_h}{T_r} > h^2r^2\pi\rho(e_t + e_s)\lambda s > N(e_t + e_s)\lambda$. $N(e_t + e_s)\lambda$ is exactly the energy consumed by sensors to generate and transmit data in one-hop communication to the MCs and the inequality states that the recharging rates from MCs is enough to support one-hop communications. Thus, we have proved that for $h > 2$, in the worst case, we can always use one-hop mobile data gathering to restore energy balance.

D. *Proof of Property 5*

Our proof is based on [38]. Since SNs can be represented by disjoint disks, the sum of feasible areas for s SNs is $s\pi r^2$. We consider a larger disk of radius $2r$ so any point in disk of radius r can be enclosed. The total area $s\pi r^2$ should be less than the area swept by the disk of $2r$,

$$s\pi r^2 + n\pi r_w^2 \le 4rL^*_r + 4r^2\pi. \tag{31}$$

The wireless charging range r_w is much smaller than r ($r_w \ll r$). If we enforce the MC to go through disk centers, an extra distance less than $2r$ has to be made (entering and leaving the center). Thus, L_c is bounded by

$$\begin{aligned} L_c &\le L^*_r + 2rs \le L^*_r + 2r\frac{4rL^*_r + 4r^2\pi - n\pi r_w^2}{\pi r^2} \\ \frac{L_c}{L^*_r} &\le \left(1 + \frac{8}{\pi}\right) + \frac{8r}{L^*_r} \le 1 + \frac{8}{\pi} + \epsilon \end{aligned} \tag{32}$$

In the first step, we use Eq. (31) for s. In the last step, we omit the last term $n(\frac{r_w}{r})^2$, as $\frac{r_w}{r} \approx 0$. Since r is much smaller than L^*_r, we denote $\frac{8r}{L^*_r} \le \epsilon$ where ϵ is a small fraction close to 0.

E. *Proof of Property 6*

Although the property seems to be true by visual judgment of Fig. 4(b), a geometric proof is difficult. Thus, we calculate p in terms of cartesian coordinates. Denote coordinates of w_i,w_{i+1} and p as $(x_i, y_i), (x_{i+1}, y_{i+1}), (x, y)$, respectively. Assume the origin of coordinate system resides at the disk center. The function of the disk is $x^2 + y^2 = r^2$. We use the *Lagrangian multiplier* method to find minimal sum of squared distance. After taking partial derivatives, the variables are

$$\begin{aligned} \frac{L_x}{\partial x} &= 4x - 2(x_i + x_{i+1}) + 2x\lambda, \\ \frac{L_y}{\partial y} &= 4y - 2(y_i + y_{i+1}) + 2y\lambda \\ \frac{L_\lambda}{\partial \lambda} &= x^2 + y^2 - r^2 \end{aligned} \tag{33}$$

After some calculations, the coordinates for p are

$$\begin{aligned} x &= \frac{(x_i + x_{i+1})r}{\sqrt{(x_i + x_{i+1})^2 + (y_i + y_{i+1})^2}}, \\ y &= \frac{(y_i + y_{i+1})r}{\sqrt{(x_i + x_{i+1})^2 + (y_i + y_{i+1})^2}}. \end{aligned} \tag{34}$$

On the other hand, the coordinates of w_m are $(\frac{x_i+x_{i+1}}{2}, \frac{y_i+y_{i+1}}{2})$. We plug the function of line $w_m a_i$, $y = \frac{y_i+y_{i+1}}{x_i+x_{i+1}}x$ into the disk function of a_i, and obtain two intersection points

$$x = \pm\sqrt{\frac{r^2}{(\frac{y_i+y_{i+1}}{x_i+x_{i+1}})^2 + 1}}, y = \pm\sqrt{\frac{r^2}{(\frac{x_i+x_{i+1}}{y_i+y_{i+1}})^2 + 1}} \tag{35}$$

We can see one of the solutions in Eq. (35) is exactly Eq. (34), so the property is proved.

XI. Acknowledgments

The work in this paper was supported in part by the grant from US National Science Foundation under grant number ECCS-1307576.

REFERENCES

[1] Powermat Technology, Website: “http://powermat.com/”.

[2] UBeam Technology, Website: “http://ubeam.com/”.

[3] FCC Rules for ISM bands, “http://www.afar.net/tutorials/fcc-rules”.

[4] PowerCast Corp, “http://www.powercastco.com.”

[5] M. Zhao, J. Li and Y. Yang, “A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks,” *IEEE Trans. Mobile Computing*, vol. 13, no. 12, pp. 2689-2705, 2014.

[6] B. Tong, Z. Li, G. Wang and W. Zhang, “How wireless power charging technology affects sensor network deployment and routing,” *IEEE ICDCS*, 2010.

[7] Y. Peng, Z. Li, W. Zhang and D. Qiao, “Prolonging sensor network lifetime through wireless charging,” *IEEE RTSS*, 2010.

[8] C. Wang, J. Li, F. Ye and Y. Yang, “NETWRAP: An NDN based real-time wireless recharging framework for wireless sensor networks,” *IEEE Trans. Mobile Computing*, vol. 13, no. 6, pp. 1283-1297, 2014.

[9] C. Wang, J. Li, F. Ye and Y. Yang, “A mobile data gathering framework for wireless rechargeable sensor networks with vehicle movement costs and capacity constraints,” *IEEE Trans. Computers*, vol. 65, no. 8, pp. 2411-2427, 2016.

[10] Z. Li, Y. Peng, W. Zhang, and D. Qiao, “J-RoC: a joint routing and charging scheme to prolong sensor network lifetime,” *IEEE ICNP*, 2011.

[11] S. He, J. Chen, F. Jiang, D. Yau, G. Xing and Y. Sun, “Energy provisioning in wireless rechargeable sensor networks,” *IEEE Trans. Mobile Computing*, vol. 12, no. 10, pp. 1931-1942, 2013.

[12] H. Dai, Y. Liu, G. Chen, X. Wu and T. He, “SCAPE: Safe charging with adjustable power,” *IEEE ICDCS*, 2014.

[13] S. Nikoletseas, R. Theofanis and R. Christoforos, “Low radiation efficient wireless energy transfer in wireless distributed systems,” *IEEE ICDCS*, 2015.

[14] A. Kansal, J Hsu, S. Zahedi and M. Srivastava. “Power management in energy harvesting sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, 2007.

[15] B. Gaudette, V. Hanumaiah, M. Krunz and S. Vrudhula, “Maximum quality of cover with connectivity in solar powered active wireless sensor networks,” *ACM Trans. on Sen. Netw. (TOSN)*, vol. 10, no. 4, pp. 1-59, 2014.

[16] R. Margolies, M. Gorlatova, J. Sarik, G. Stanje, J. Zhu, P. Miller, M. Szczodrak, B. Vigraham, L. Carloni, P. Kinget, I. Kymissis, G. Zussman, “Energy Harvesting Active Networked Tags (EnHANTs): Prototyping and Experimentation,” *ACM Trans. Sen. Netw*, vol. 11, no. 4, pp. 1-62, Nov. 2015.

[17] R. S. Liu, K. W. Fan, Z. Zheng and P. Sinha, ”Perpetual and Fair Data Collection for Environmental Energy Harvesting Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 4, pp. 947-960, 2011.

[18] M. Ma, Y. Yang and M. Zhao, “Tour planning for mobile data-gathering mechanisms in wireless sensor networks,” *IEEE Trans. Vehicular Technology (TVT)*, vol. 62, no. 4, pp. 1472-1483, May 2013.

[19] B. Yuan, M. Orlowska and S. Sadiq, “On the optimal robot routing problem in wireless sensor networks,”*IEEE Trans. Knowledge and Data Engineering*, vol. 19, no. 9, 2007.

[20] L. He, J. Pan and J. Xu, “A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements,” *IEEE Trans. Mobile Computing*, vol. 12, no. 7, pp. 1308-1320, 2013.

[21] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman and M. Srivastava,“Design considerations for solar energy harvesting wireless embedded systems,” *IEEE IPSN*, 2005.

[22] V. Rai and R. N. Mahapartra, “Lifetime modeling of a sensor network,”*IEEE DATE*, 2005.

[23] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” *IEEE HICSS*, 2000.

[24] Panasonic Ni-MH battery handbook, “http://www2.renovaar.ee/userfiles/Panasonic_Ni-MH_Handbook.pdf”.

[25] X. Wu, G. Chen and S. Das, “Avoiding energy holes in wireless sensor networks with nonuniform node distribution,” *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 5, 2008.

[26] D. Shmoys, E. Tardos and K. Aardal, “Approximation algorithms for facility location problems,” *ACM STOC*, 1997.

[27] K. Jain and V. Vazirani, “Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation,” *JACM* vol. 48, no. 2, pp. 274-296, 2001.

[28] K. Jain, M. Mahdian, E. Markakis, A. Saberi and V. Vazirani, “Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP,” *JACM*, vol. 50, no. 6, pp. 795-824, 2003.

[29] S. Guha and S. Khuller, “Greedy strikes back: improved facility location algorithms,” *Journal of Algorithms*, vol. 31, no. 1, pp. 228-248, 1999.

[30] E. Weiszfeld and F. Plastria, “(Translated version) On the point for which the sum of the distances to n given points is minimum,” *Annals of Op. Research*, vol. 167, no. 1, pp. 7-41, 2009.

[31] H. Kuhn, “A note on Fermat’s problem,” *Mathematical Programming*, vol. 4, no. 1, pp. 98-107, 1973.

[32] J. Eyster, J. White and W. Wierwille, “On solving multifacility location problems using a hyperboloid approximation procedure”, *IIE Transactions*, vol. 5, no. 1, pp. 1-6, 1973.

[33] N. Sharma, J. Gummeson and D. Irwin, “Cloudy computing: leveraging weather forecasts in energy harvesting sensor systems,” *IEEE SECON*, 2010.

[34] A. Amis, R. Prakash, T. Vuong and D. Huynh,“Max-min d-cluster formation in wireless ad hoc networks,” *IEEE INFOCOM*, 2000.

[35] T. Gonzalez, “Clustering to minimize the maximum intercluster distance,” *Theoretical Computer Science*, vol. 38, pp. 293-306, 1985.

[36] T.H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, MIT Press, 2001.

[37] S. Shmuel and O. Schwartz, “On the complexity of approximating TSP with neighborhoods and related problems,” *Computational Complexity*, 2006.

[38] A, Dumitrescu and J. Mitchell, “Approximation algorithms for TSP with neighborhoods in the plane,” *ACM Symp. Discrete Algorithms*, 2001.

[39] Weather underground: “www.wunderground.com/history/”.

[40] K. Elbassioni, A. Fishkin, N. Mustafa and R. Sitters, “Approximation algorithms for Euclidean group TSP,” *ICALP*, 2005.

[41] C. H. Papadimitriou and K. Steiglitz, “Combinatorial optimization: algorithms and complexity,” *Dover Publications*, 1998.

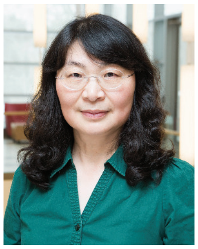
[42] Lenmar AA battery, http://www2.lenmar.com

[43] Battery Calculator, “http://www.evsource.com/battery_calculator.php.”

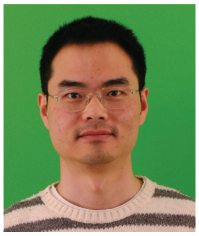
[44] Solar GIS, http://www.nrel.gov/gis/data_solar.html

Cong Wang received BEng in Information Engineering from the Chinese University of Hong Kong and M.S. from Columbia University. He received the PhD degree in Electrical Engineering from Stony Brook University in 2017. He will join Old Dominion University as an Assistant Professor in the ECE Department in fall 2017. His research interests include wireless sensor networks, performance evaluation of network protocols and algorithms.

Ji Li received B.S. in Electrical Engineering from Harbin Engineering University, Harbin, China, and M.S. from Zhejiang University, Hangzhou, China. He is currently a PhD student in Electrical and Computer Engineering at Stony Brook University. His research interests include wireless sensor networks and embedded systems.



Yuanyuan Yang received the BEng and MS degrees in computer science and engineering from Tsinghua University and the MSE and PhD degrees in computer science from Johns Hopkins University. She is a SUNY Distinguished Professor of computer engineering and computer science and the Associate Dean for Academic Affairs in the College of Engineering and Applied Sciences at Stony Brook University, New York. Her research interests include wireless networks, data center networks and cloud computing. She has published over 360 papers in major journals and refereed conference proceedings and holds seven US patents in these areas. She is currently the Associate Editor-in-Chief for IEEE Transactions on Cloud Computing and an Associate Editor for ACM Computing Surveys. She has served as an Associate Editor-in-Chief and Associated Editor for IEEE Transactions on Computers and Associate Editor for IEEE Transactions on Parallel and Distributed Systems. She has also served as a general chair, program chair, or vice chair for several major conferences and a program committee member for numerous conferences. She is an IEEE Fellow.



Fan Ye is an Assistant Professor in the ECE Department of Stony Brook University. He got his Ph.D. from the Computer Science Department of UCLA, M.S. and B.E. from Tsinghua University. He has published over 60 peer reviewed papers that have received over 8000 citations according to Google Scholar. He has 21 granted/pending US and international patents/applications. He was the co-chair for the Mobile Computing Professional Interests Community at IBM Watson for two years. He received IBM Research Division Award, 5 Invention Achievement Plateau awards, Best Paper Award for International Conference on Parallel Computing 2008. His current research interests include mobile sensing platforms, systems and applications, Internet-of-Things, indoor location sensing, wireless and sensor networks.