A Hybrid Framework Combining Solar Energy Harvesting and Wireless Charging for 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*—Recently, there have been a growing number of applications that power wireless sensor networks (WSNs) by wireless charging technology. Although previous studies indicate that wireless charging can deliver energy reliably, it still faces regulatory challenges 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 that wireless charging can provide. Fortunately, solar energy harvesting can provide high power density which is also risk-free. However, it is subject to weather dynamics. Therefore, 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. First, we study a placement problem on 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. Second, we establish an energy balance in the network and explore how to maintain such balance when sunlight is unavailable. Third, we consider combining wireless charging and mobile data gathering in a joint tour in such networks, and propose a polynomial-time scheduling algorithm. Our extensive simulation demonstrates that the hybrid framework can reduce battery depletion by 20% and save system cost by 25% compared to previous results.**

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

I. Introduction

Wireless charging technology is a promising solution to meet the prevalent energy needs from mobile electronics to low-power devices. Its application in wireless sensor networks (WSNs) has been studied extensively recently [1]–[4]. By launching wireless energy transmitters on mobile chargers (MCs) [1], [2] or at strategic locations [3], sensors can be charged conveniently without wires or plugs. In [1], optimization of wireless charging and mobile data gathering is studied by combining the two functions on a single MC. The work of [2] further considers MC's recharge capacity and nodes' lifetimes. In [3], wireless chargers are deployed at strategic locations for maximum coverage. 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 [4]. As a result, energy transmitters must comply with standards from Federal Communication Commission (FCC) and limit their emitting power to human-safe power density ($1mW/cm^2$ [5]). 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.

Meanwhile, to find a way that is risk-free but has much higher power density, another competitive technique called environmental energy harvesting has been considered. As shown in [6], 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 from 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.

After realizing both technologies have their pros and cons, in this paper, we propose a hybrid framework to combine their advantages and overcome their drawbacks. In the new framework, a majority of nodes are wireless-powered nodes (WNs) because charging coils can be cheaply manufactured from copper wires. On the other hand, due to 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 resolve recharge requests from WNs and collect data from SNs. In contrast to WNs, energy on SNs from the ambient source is self-sufficient. This scheme realizes effective energy replenishment at cluster heads so that they can complete high volume of 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 compared to the previous approach in [1]?

To answer these questions, in this paper we first study a placement problem in discrete sense where SNs are deployed at known WN locations. We formulate it into a *facility location problem* [9]–[12] 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 [11]. We also demonstrate how our algorithm can adapt to seasonal variations of sunlight. 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 propose 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 [1] 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 propose a polynomial time route improvement algorithm that can take shortcuts through SN's neighborhood for additional cost saving.

We make several contributions in this paper. First, we propose a hybrid framework to overcome the disadvantages of wireless charging and environmental harvesting techniques. To the best of our knowledge, this is the first work considering WSNs based on 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 find a way to maintain network robustness by reducing energy consumptions. Fourth, we propose a route improvement algorithm that saves an extra 25% moving energy on MCs and surpasses the algorithm in [21] 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. Finally, we conduct extensive simulations to evaluate the performance of the framework compared to WSNs that are solely wireless-powered [1]–[4].

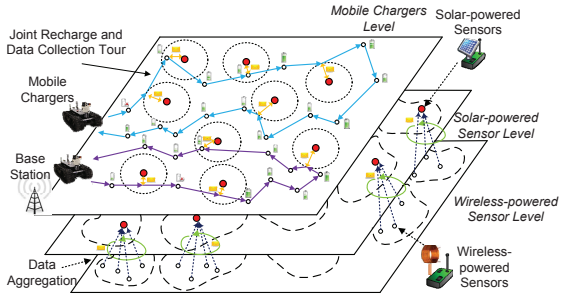


Fig. 1. An overview of 3-level network hierarchy.

The rest of the paper is organized as follows. Section II presents network model and assumptions. Section III studies the placement problem of SNs. Section IV provides theoretical analysis and discusses how to maintain energy balance using WNs. Section V optimizes MC's migration routes. Section VI evaluates the new framework by simulations and Section VII concludes the paper.

II. 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. As shown in Fig. 1, the network consists of three levels: wireless-powered sensor, solar-powered sensor and mobile charger levels.

The bottom 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 of transmitting an l-bit packet follows the widely adopted model [7], $e_t = e_0 + e_1 r^\alpha$, where e_t is the transmitting energy, e_0 and e_1 are the energy consumed in electronics and amplifiers, r is the transmission distance and α is the path loss exponent. To perform sensing tasks, it also consumes e_s energy for each packet. For message exchange, we assume the network is connected. Each WN generates packets independently from others following a Poisson process with average rate λ. Each WN is powered by a 750 mAh rechargeable NiMH battery and the recharge time T_r is 75 mins. If the energy drops below a threshold, e.g., 50%, it sends out a request to the MCs for scheduling energy replenishment.

The 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 [13], which will be discussed in Section IV. We assume the size of solar panel is chosen to be large enough to harvest enough energy for aggregating all the data. A commercially available panel of $10 \times 10 cm^2$ size is adopted which is connected to a 3V, 2150 mAh lithium-ion battery.

The top level manages a fleet of m MCs through the base station. Proposed in [1], MCs are equipped with high-capacity batteries and powerful antennas for energy replenishment and data collection respectively. 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 they can approach the locations in close proximity for wireless charging in high efficiency. We assume each node is only recharged by one MC at a time and the emitting power at the wireless charger also complies with FCC's regulations so minimal health risks would be created. Depending on updated geographical solar energy distributions, MCs can deploy SNs at appropriate locations. 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 (SNs) and p_m (MCs) and $p_s < p_m$.

III. 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 [7], 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 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 between p_s and 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) [9]–[12], 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 cost 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 is proposed in [9], there has been encouraging progress in improving the approximation ratio and running time. An $\mathcal{O}(n^2 \log n)$ algorithm is proposed in [10] with an approximation ratio of 3 based on the primal-dual method. This bound is soon improved by [11] from 1.86 to 1.61 which is very close to the upper limit 1.46-ratio that polynomial-time algorithms can achieve [12].

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 [11]. Next, we first formalize SPP and illustrate the centralized 1.61-approximation algorithm. Then we propose a distributed version of the algorithm. 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 the distributed versions of the algorithm.

TABLE I
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

1) Centralized Placement Algorithm: First, let us formalize SPP. We denote the sets of SNs and WNs by $\mathcal{S}$ and $\mathcal{N}$, respectively. We 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, $f_i = p_s / l_i$, 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

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

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

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

Constraints (2) and (3) impose that each WN is only connected to one SN. A centralized 1.61-approximation algorithm is proposed in [11]. As a guideline for 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. It is summarized in Table I.

2) Distributed Algorithm: To understand the nature of the problem, we 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

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

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

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

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 $\sum_{j \in \mathcal{N}} \max(a_j - c_{ij}, 0) \leq f_i$ 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* [11] and 1.61-factor approximation is proved in [11] following the centralized paradigm. Based on [11], we propose a distributed $1.61(1+\epsilon)^2$-approximation algorithm next, where ϵ is a small constant greater than zero.

TABLE II
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 $a_j \leftarrow (1+\epsilon) a_j$.

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

Receive offering messages from WNs.
If i is not yet deployed AND $\sum_{j \in \mathcal{N}} \max(a_j - c_{ij}, 0) \geq f_i$
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

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 $\max(c_{i'j} - c_{ij}, 0)$. On 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 II and Table III and it has the following properties.

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

Proof: We sequentialize the distributed algorithm into execution rounds. For the distributed algorithm, each round consists of a number of message sent and received 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 [11]. ■

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: First, denote optimal offers in the centralized algorithm [11] by a_j and the distributed algorithm by a'_j, $j \in \mathcal{N}$. A

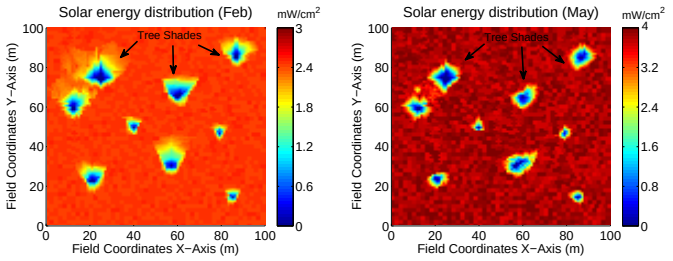


Fig. 2. Geographic solar energy distribution in different months. (a) February. (b) May.

Factor Revealing LP is constructed by [11]. For SN i, $k = |\mathcal{B}_i|$, the optimal solution is to solve the following maximization problem

$$\mathbf{P3}: \quad z_k = \max \sum_{j=1}^{k} a_j / (\sum_{j=1}^{k} c_{ij} + f_i) \tag{10}$$

Subject to

$$a_j \leq a_{j+1}, \forall j \in \{1, \ldots, k-1\} \tag{11}$$

$$\sum_{j=1}^{k} \max(a_j - c_{il}, 0) \leq f_i, \forall l \in \{1, \ldots, k\} \tag{12}$$

$$a_j \leq a_l + c_{ij} + c_{il}, \forall j, l \in \{1, \ldots, k\} \tag{13}$$

$$a_j, c_{ij}, c_{il}, f_i \geq 0, \forall j, l \in \{1, \ldots, k\} \tag{14}$$

For the maximization problem to be bounded, a_j should also be bounded. It implies that at least one of the constraints of (12) and (13) is tight (i.e., changing from inequality into equality). *Case 1*: Eq. (12) is tight; *Case 2*: Eq. (13) is tight, and a_l is also bounded.

For the distributed algorithm, we can formulate it into a similar Factor Revealing LP except that constraint (12) becomes $\sum_{j=1}^{k} \max(\frac{a'_j}{1+\epsilon} - c_{il}, 0) \leq f_i$ and constraint (13) becomes $\frac{a'_j}{1+\epsilon} \leq a'_l + c_{ij} + c_{il}$ 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 $f_i \geq 0$ and the constraint is tight, $a_j - c_{il} \geq 0$ and $\frac{a'_j}{1+\epsilon} - c_{il} \geq 0$ hold for the centralized and distributed algorithms, respectively. Thus, $\frac{a'_j}{a_j} \leq 1 + \epsilon$. For Case 2: $\frac{a'_j}{1+\epsilon} = a'_l + c_{ij} + c_{il}$ and $\sum_{l=1}^{k} \max(\frac{a'_l}{1+\epsilon} - c_{il}, 0) \leq f_i$ is also tight to bound a'_l. The latter suggests that the ratio in Case 1 $\frac{a'_l}{a_l} \leq 1 + \epsilon$ can be applied here,

$$\frac{a'_j}{1+\epsilon} \leq a_l(1+\epsilon) + c_{ij} + c_{il} \leq (1+\epsilon)(a_l + c_{ij} + c_{il}) \tag{15}$$

Then by taking the ratio between Eq. (15) and Eq. (13), we have $\frac{a'_j}{a_j} \leq (1+\epsilon)^2$ for Case 2. Since the approximation ratio to the optimal solution a^*_j is proved by [11], $\frac{a_j}{a^*_j} \leq 1.61$. Thus, our algorithm has at most $\frac{a'_j}{a^*_j} \leq 1.61(1+\epsilon)^2$ approximation to the optimal solution. ■

B. Adapt Solar Variations

Next, we demonstrate how to change SN's locations to adapt solar variations. During different seasons of a year, the sun's angle towards earth surface varies slowly, and consequently, the harvested energy at different locations reflects such changes due to building obstructions, tree shades, etc. Fig. 2 shows the heatmaps of a sensing field on our campus gathered at different locations in February and May (Longitude at North $40°$). We can see that the areas fall 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. After their locations are found, an MC is dispatched to re-locate corresponding SNs to designated locations.

IV. Wireless-powered Sensor Level: Maintain 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 are operative 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

$$\begin{aligned} E_c &= \sum_{j \in \mathcal{N}} \big[\lambda(e_t + e_s) + \sum_{i \in \mathcal{C}_j} \lambda(e_t + e_r)\big] T \\ &\leq \sum_{i=1}^{h} \big[N_i(e_t + e_s) + \sum_{\substack{j=i+1, \\ i \neq h}}^{h} N_j(e_t + e_r)\big] \lambda s T \\ &= \big[(\frac{2}{3}h^3 - \frac{1}{2}h^2 - \frac{1}{6}h)(e_t + e_r) + h^2(e_t + e_s)\big] \pi r^2 \rho \lambda s T \end{aligned} \tag{16}$$

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 [8]; 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 [13]. 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,

$$E = (a_1(t + a_2)^2 + a_3)(1 - \sigma). \tag{17}$$

The shape of Eq. (17) 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 ($t_1 = -\sqrt{-\frac{a_3}{a_1}} - a_2, t_2 = \sqrt{-\frac{a_3}{a_1}} - a_2$) according to [13]. σ is the percentage of cloud cover from weather reports. For T days, energy harvested by SNs is

$$E_s = s \sum_{i=1}^{T} \int_{t_1}^{t_2} \big[a_1(t + a_2)^2 + a_3\big](1 - \sigma_i) dt \tag{18}$$

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

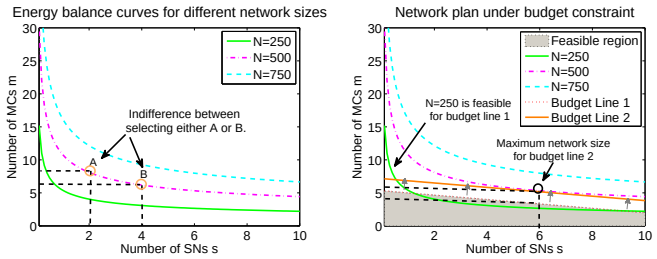


Fig. 3. Network plans under budget constraints. (a) Energy balance curves for different network sizes. (b) Optimal choices under budget constraints.

achieved when

$$E_c \le E_s + E_w$$
$$E_c < \big[(\frac{2}{3}h^3 - \frac{1}{2}h^2 - \frac{1}{6}h)(e_t + e_r) + h^2(e_t + e_s)\big]\pi r^2 \rho\lambda s T$$
$$\le s\sum_{i=1}^{T}\int_{t_1}^{t_2}\big[a_1(t+a_2)^2 + a_3)\big](1-\sigma_i)dt + \frac{mTC_h}{T_r} \quad (19)$$

Since $\pi(hr)^2 s \ge L^2$, we have $\sqrt{L^2/(sr^2\pi)} \le h$. By plugging it into Eq. (19) and taking approximation $e_t \approx e_r$, we obtain a relationship between s and m

$$\frac{LT_r e_t \rho\lambda}{3\sqrt{\pi}C_h r}(\frac{4L^2}{\sqrt{s}} - \pi r^2\sqrt{s}) - Xs + (\frac{1}{2}e_t + e_s)\frac{L^2\rho\lambda T_r}{C_h} \le m, \quad (20)$$

where X is

$$X = \big[\frac{T_r}{C_h T}\sum_{i=1}^{T}(1-\sigma_i)\big]\big[\frac{a_1}{3}t^3 + a_1 a_2 t^2 + (a_1 a_2^2 + a_3)t\big]\Big|_{t=t_1}^{t=t_2}. \quad (21)$$

The relationship between s and m in Eq. (20) can be explained graphically. Fig. 3(a) shows a group of energy balance curves when $N = 250 - 750$. 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 that make no difference to the consumer 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 keep 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 (saves 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 as shown 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.

B. 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) Maintain 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 case solar energy is unavailable, we can always restore network energy balance by reducing cluster size.

Proof: 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. (19),

$$\frac{mC_h}{T_r} > \big[(\frac{2}{3}h^3 - \frac{3}{2}h^2 - \frac{1}{6}h)(e_t + e_r) + h^2(e_t + e_s)\big]\pi r^2\rho\lambda s \quad (22)$$

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. ■

Remarks: 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 (plug k into Eq. (16) and get rid of s'). Since we require $E_s^* \le E_c(h) - s'X_c(k)$, a range for s' is

$$\frac{L^2}{\pi(kr)^2} < s' \le \frac{E_c(h) - E_s^*}{X_c(k)}. \quad (23)$$

By fixing k, any s' satisfying Eq. (23) will guarantee energy balance of the network. Next, we develop a distributed algorithm to find WN cluster heads.

2) Head Re-selection Problem: In this subsection, we further explore the *Head Re-selection Problem* (HRP) which finds k-hop clusters with s' cluster heads satisfying Eq. (23). 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 proved to be NP-hard in [14].

A distributed algorithm for this problem is proposed in [14] 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 [14]. Instead, we leverage the range in Eq. (23) 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 [15]. 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. (23). 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 $k = 1$). 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 [16]). 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. (23), 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 IV. 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. (23).

TABLE IV
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

V. Mobile Charger Layer: Joint Wireless Charging and Mobile Data Gathering

In this section, we focus on optimizing trajectories of the MCs. Proposed by [1], launching radio modules on MCs realizes 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 [1] 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. 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 [2] can be used conveniently to take recharging and data gathering requests together and calculate an initial touring sequence for each MC. Normally, cluster size is larger than one hop ($h > 1$), 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 $(1 + \frac{8}{\pi} + \epsilon) \approx 3.55 + \epsilon$ to the optimal L_r^*.

Proof: Our proof is based on [18]. 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 a 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 $2r$,

$$s\pi r^2 + n\pi r_w^2 \leq 4rL_r^* + 4r^2\pi. \tag{24}$$

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

$$\begin{aligned} L_c &\leq L_r^* + 2rs \leq L_r^* + 2r\frac{4rL_r^* + 4r^2\pi - n\pi r_w^2}{\pi r^2} \\ \frac{L_c}{L_r^*} &\leq (1 + \frac{8}{\pi}) + \frac{8r}{L_r^*} \leq 1 + \frac{8}{\pi} + \epsilon \end{aligned} \tag{25}$$

In the first step, we use Eq. (24) 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 $\frac{8r}{L_r^*} \leq \epsilon$ where ϵ is a small number close to 0. ■

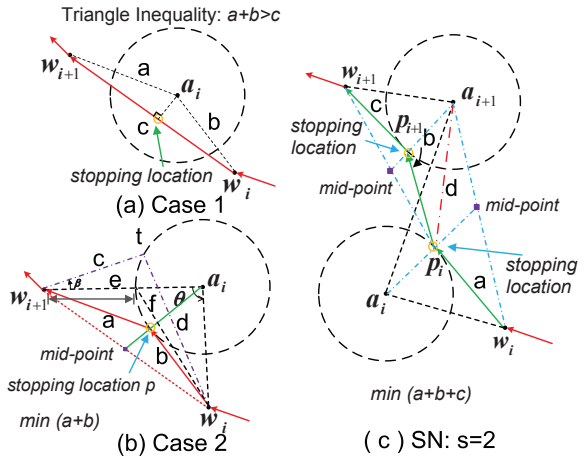


Fig. 4. Analysis of shortest path through feasible regions around SNs.

B. 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 [17], [18]. 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 of $f = \frac{\theta l}{2\pi}$ 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.

C. Minimize 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 [16]. 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 actually 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}.

TABLE V
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: 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

$$\begin{aligned} \frac{L_x}{\partial x} &= 4x - 2(x_i + x_{i+1}) + 2x\lambda, \\ \frac{L_y}{\partial y} &= 4y - 2(y_i + y_{i+1}) + 2y\lambda \\ \frac{L_\lambda}{\partial \lambda} &= x^2 + y^2 - r^2 \end{aligned} \tag{26}$$

After some calculations, the coordinates for p is

$$\begin{aligned} x &= \frac{(x_i + x_{i+1})r}{\sqrt{(x_i + x_{i+1})^2 + (y_i + y_{i+1})^2}}, \\ y &= \frac{(y_i + y_{i+1})r}{\sqrt{(x_i + x_{i+1})^2 + (y_i + y_{i+1})^2}}. \end{aligned} \tag{27}$$

On the other hand, the coordinate of w_m is $(\frac{x_i+x_{i+1}}{2}, \frac{y_i+y_{i+1}}{2})$. 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

$$x = \pm\sqrt{\frac{r^2}{(\frac{y_i+y_{i+1}}{x_i+x_{i+1}})^2 + 1}}, y = \pm\sqrt{\frac{r^2}{(\frac{x_i+x_{i+1}}{y_i+y_{i+1}})^2 + 1}} \tag{28}$$

We can see one of the solutions in Eq. (28) is exactly Eq. (27), so the property is proved. ■

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 $\mathcal{O}(s)$ computations are needed, which reduces the exponential-$\mathcal{O}(f^s)$ exhaustive search algorithm to linear time. We summarize the route improvement algorithm in Table V.

D. An Example of the Hybrid Framework

Finally, we demonstrate a complete example of the 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. In case of shortage of sunlight,

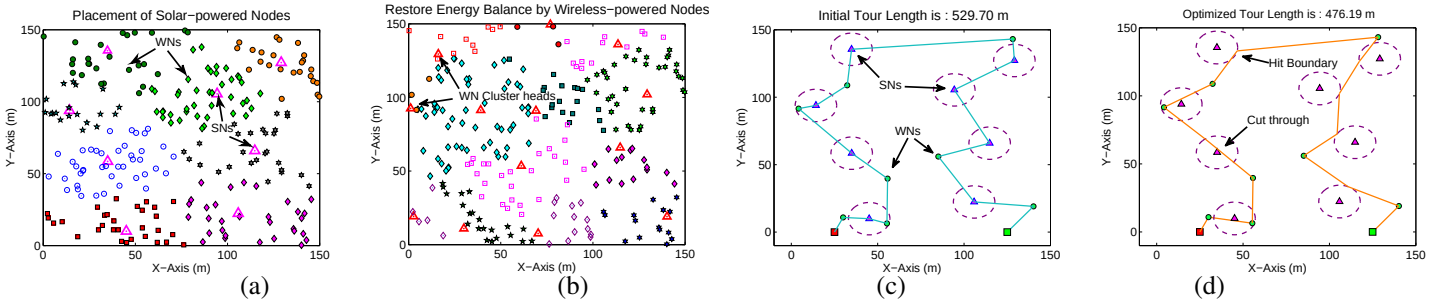


Fig. 5. A complete example of the framework. (a) Placement of SNs. (b) Restore energy balance by WNs. (c) Initial center tour. (d) Optimized joint tour.

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 8 energy requests and 8 data uploading sites (SNs). The route is improved by our algorithm in Fig. 5 (d) with a saving of 10% moving energy on the MCs.

VI. Performance Evaluations

In this section, we evaluate the performance of the framework by a discrete-event simulator and compare it with a network solely relying on wireless energy [1]–[3]. Note that all 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. Each time slot is 1 min and the traffic follows a Poisson distribution with average $\lambda = 3$ pkt/min. WNs have battery capacity $C_h = 780mAh$ and require $T_r = 78$ mins for recharge. SNs have larger capacity of $2150mAh$. MC consumes 5 J/m while moving at $v = 1$ m/s. We use real meteorological trace from [20], which has a complete archive of weather conditions. Since our observation of solar variation lasts from December to May, we set the simulation for 6 months time.

A. Evaluation of Route Improvement Algorithm

First, we evaluate the route improvement algorithm by comparing it with the algorithm proposed in [21]. The algorithm in [21] 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 [1] for joint wireless charging and data gathering and we denote it by *initial center tour*. Fig. 6(a) 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 [21]. 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(b) 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]. Surprisingly, our algorithm has only an average of 1% difference to the exact solution whereas reducing computation complexity from exponential to linear time. In addition, with mixed WNs and SNs, our algorithm also outperforms the nearest insertion algorithm by 5-10%.

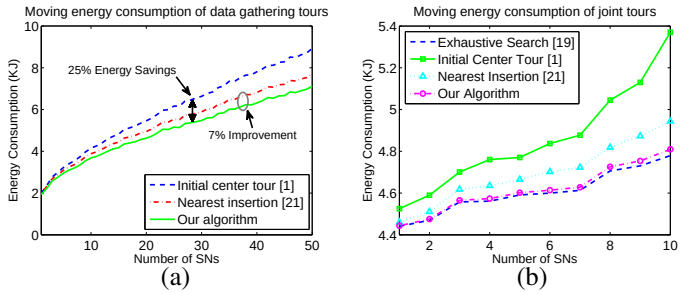


Fig. 6. Evaluating route improvement algorithm. (a) Tour consists of only data gathering sites (SNs). (b) Joint wireless charging and mobile data gathering tour.

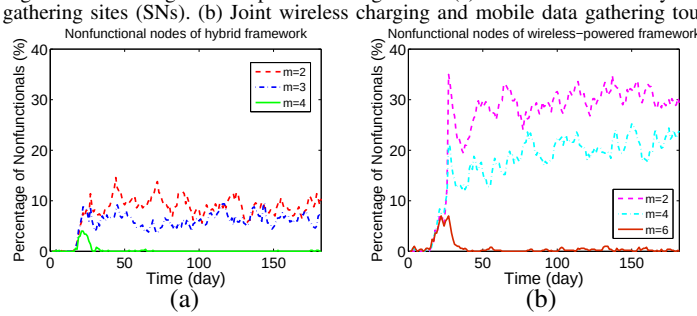


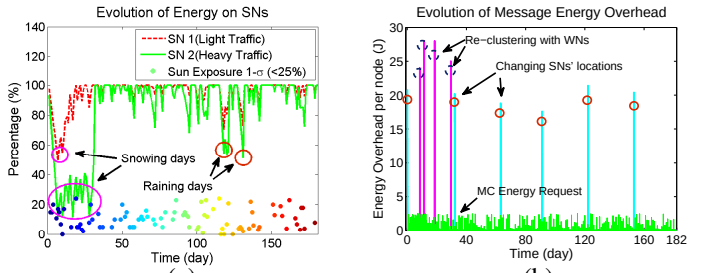
Fig. 7. Number of nonfunctional nodes. (a) Hybrid framework. (b) Wireless-powered framework.

B. Nonfunctional Nodes

One of the key performance metrics 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, they will degrade sensing qualities and node communications. Fig. 7 compares the percentage of nonfunctional nodes between hybrid and wireless-powered frameworks [1]–[3]. The SPP algorithm generates $s = 11$ SNs. To compare 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. 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, thus 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. Second, the results also indicate that having 11 SNs can save 2 MCs while achieving similar performance. This is beneficial in practice since the manufacturing and operating cost of MCs is usually much higher than SNs.

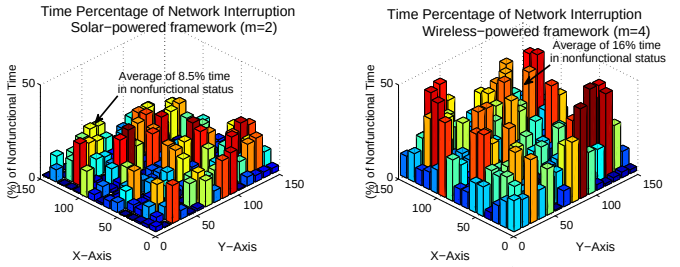
C. Harvested Energy and Message Overhead

To validate our algorithm design, we also evaluate the evolution of SN's energy and network message overhead. Fig. 8(a) traces



(a) (b)

Fig. 8. Evolution of harvested solar energy and message energy overhead. (a) Energy variations of SNs. (b) Message energy overhead.



(a) (b)

Fig. 9. Geographical distributions of service interruption. (a) Hybrid framework $m = 2$. (b) Wireless-powered framework $m = 4$.

SNs' energy with weather conditions represented by percentage of solar exposure $(1 - \sigma)$ obtained in [20]. 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 the 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 during spring and energy storage is sufficient to sustain network operations.

Fig. 8(b) demonstrates the energy consumed by exchanging control messages such as SN clustering, re-selection of WN cluster 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). The 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 25% due to insufficient solar energy, they request to re-cluster by WNs. The overhead is represented by red spikes, which corresponds to the time when SNs' energy drops in Fig. 8(a). From the results of Fig. 8, we have validated that our algorithm can adapt to weather conditions effectively.

D. Geographical Distributions of Service Interruption

Finally, we examine geographical distributions of service interruptions. Our objective is to see how long nodes are in non-functional 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 Sec. VI-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. 9(a), we observe that the distribution of nonfunctional nodes are 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 time for being nonfunctional in Fig. 9(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.

VII. 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 layers. 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. Second, we examine the energy balance in the network and develop a distributed head re-selection algorithm to designate some WNs as heads when solar energy is not available. 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 the exact solution and reduce at least 5% MC's moving energy compared to previous solutions. Finally, based on real weather data, we demonstrate the effectiveness and efficiency of the hybrid framework that can improve network performance significantly.

References

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

[2] C. Wang, J. Li, F. Ye and Y. Yang, "Recharging schedules for wireless sensor networks with vehicle movement costs and capacity constraints," *IEEE SECON*, 2014.

[3] S. He, J. Chen, F. Jiang, D. Yau, G. Xing and Y. Sun,"Energy provisioning in wireless rechargeable sensor networks," *IEEE TMC*, vol. 12, no. 10, pp. 1931-1942, Oct. 2013.

[4] H. Dai, Y. Liu, G. Chen, X. Wu and T. He,"Safe charging for wireless power transfer," *IEEE INFOCOM*, 2014.

[5] Online: "http://www.afar.net/tutorials/fcc-rules".

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

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

[8] X. Wu, G. Chen and S. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE TPDS*, vol.19, no.5, 2008.

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

[10] 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.

[11] 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.

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

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

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

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

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

[17] S. Shmuel and O. Schwartz, "On the complexity of approximating TSP with neighborhoods and related problems," *Computational complexity*, 2006.

[18] A, Dumitrescu and J. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," *ACM Symp. on discrete algorithms*, 2001.

[19] B. Yuan, M. Orlowska and S. Sadiq, "On the optimal robot routing problem in wireless sensor networks,"*IEEE TKDE*, vol. 19, no. 9, 2007.

[20] Weather underground: "www.wunderground.com/history/".

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