Energy-Aware Multiple Mobile Chargers Coordination for Wireless Rechargeable Sensor Networks

Lei Mo, Angeliki Kritikakou, Shibo He

HAL Id: hal-02140283
https://inria.hal.science/hal-02140283

Submitted on 27 May 2019

Energy-Aware Multiple Mobile Chargers Coordination for Wireless Rechargeable Sensor Networks

Lei Mo*, *Member, IEEE,* Angeliki Kritikakou*, and Shibo He†, *Member, IEEE*

Abstract—**Wireless charging provides dynamic power supply for Wireless Sensor Networks (WSNs). Such systems, are typically considered under the scenario of Wireless Rechargeable Sensor Networks (WRSNs). With the use of mobile chargers (MCs), the flexibility of WRSNs is further enhanced. However, the use of MCs poses several challenges during the system design. The coordination process has to simultaneously optimize the scheduling, the moving time and the charging time of multiple MCs under limited system resources (time and energy). Efficient methods that jointly solve these challenges are generally lacking in the literature. In this paper, we address the multiple MCs coordination problem under multiple system requirements. Firstly, we aim at minimizing the energy consumption of MCs, guaranteeing that every sensor will not run out of energy. We formulate the multiple MCs coordination problem as a mixed-integer linear programming and derive a set of desired network properties. Secondly, we propose a novel decomposition method to optimally solve the problem, as well as to reduce the computation time. Our approach divides the problem into a subproblem for the MC scheduling and a subproblem for the MC moving time and charging time, and solves them iteratively by utilizing the solution of one into the other. The convergence of proposed method is analyzed theoretically. Simulation results demonstrate the effectiveness and scalability of the proposed method in terms of solution quality and computation time.**

Keywords—*Wireless rechargeable sensor networks, mobile charger coordination, perpetual operation, mixed-integer linear program, decomposition method*

I. INTRODUCTION

With the rapid development of wireless energy transfer technology [2], Wireless Rechargeable Sensor Networks (WRSNs) have become an important research topic [3], [4]. Unlike the traditional Wireless Sensor Networks (WSNs), WRSNs are able to circumvent the limitations introduced by the energy-constrained sensor nodes, and, thus, achieving a perpetual network lifetime. To recharge a large-scale sensor network, it is necessary to use multiple mobile chargers (MCs). Compared with single MC dispatch problem, multiple MCs dispatch problem further involves the coordination among the MCs [5], [6]. By properly coordinating the MCs, we can finish the complex charging task and also enhance the scalability and the robustness of the system. The multiple MCs coordination problem usually considers WRSNs with three main components: stationary sensor nodes, MCs and one stationary base

* L. Mo and A. Kritikakou are with the University of Rennes, INRIA, CNRS, IRISA, 35042 Rennes Cedex, France. E-mail: lei.mo@inria.fr, angeliki.kritikakou@irisa.fr
†S. He is with the College of Control Science and Engineering, Zhejiang University, 310027 Hangzhou, China. E-mail: s18he@zju.edu.cn (*Corresponding author*)

Ref.	Objective		MC Engrgy		Coordination		Solution	
	Sig.	Multi.	Un-limt.	Limt.	Cen.	Dis.	Non-opt.	Opt.
[1]		√	√			√		√
[7]	√		√		√		√	
[8]	√		√		√			√
[9]	√		√		√		√	
[10]	√		√		√			√
[11]	√		√		√		√	
[12]		√		√	√		√	
[13]	√			√	√		√	
[14]	√			√	√		√	
[15]	√			√	√		√	
[16]	√			√	√	√	√	
[17]	√			√	√			√
[18]	√			√	√			√
[19]	√			√	√	√	√	
Our		√		√	√			√

TABLE I.   CLASSIFICATION OF SOME MULTIPLE MCs CHARGING OPTIMIZATION APPROACHES

station (BS). The sensor nodes periodically report their residual energies to the BS. Based on the collected information, the BS makes a charging decision and sends the control commands to the MCs, which will then schedule their movements and adjust their moving time and charging time. During the movement, the MCs stop at the working points and charge the sensor nodes. The charging process operates in cycles. At each cycle, the MCs start from the BS, move forward the sensor nodes to charge them and return to the BS to charge or replace the battery after finish the assigned energy charging tasks.

The prior work on multiple MCs coordination optimization can be classified according to the following criteria: whether 1) the objective is single, i.e., the problem optimizes only the MC scheduling, or multiple, i.e., the problem jointly optimizes the MC scheduling, the moving and the charging time (time required for the MCs to charge the sensor nodes), 2) the energy of the MC is limited or unlimited, 3) the coordination mechanism is distributed or centralized, and 4) the solution is optimal or non-optimal. Table I provides a summary of several representative works in the literature. The majority of the works in this area deals with the single objective of MC scheduling and utilizes centralized mechanisms to coordinate the MCs. Although the distributed coordination is considered in [1], [16], [19], the coordination process is divided into several phases and a distributed mechanism only exists in some specific phases. However, if multiple system requirements are taken into account, it is difficult to realize a fully distributed mechanism. By jointly considering the scheduling, the moving time and the charging time of the MCs, the replenished energy of the sensor nodes can be optimized. Therefore, we can avoid to fully charge the sensor nodes, leading to a better usage of system resources (time and energy). On the other hand, the number of sensor nodes is usually much larger than the number of MCs, and, thus, one MC requires to charge several sensor nodes in one cycle. If an MC does not have enough energy to

finish the assigned tasks (e.g., moving or charging tasks), this MC has to return to the BS for battery charging or replacement. Therefore, the number of available MCs is time-varying, which influences the scheduling, the moving time and the charging time of the MCs. Moreover, the charging optimization problems usually aim to enhance the charging efficiency [8], [12], [13], [18] or reduce the energy consumption of the MCs [1], [7], [9]. Therefore, the problem becomes more realistic, when the constraints related to the limited MC energy budget are taken into account.

It is worth noting that the extensions from: 1) the single objective case to the multiple objectives case, or 2) the MC energy unbounded case to the MC energy bounded case, are not straightforward. This is due to the fact that: 1) additional variables and constraints need to be added into the problem so as to satisfy the new requirements, and 2) the coupling between the optimization variables makes the problem hard to solve. Heuristics are popular methods to solve charging optimization problems [7], [9], [11]–[15]. They are able to find a feasible (non-optimal) solution in a short time. However, they are sensitive to the modification of the problem structure and the parameters, while it is hard to guarantee the quality of the solution. Finding the optimal solution is very important. By doing so, we can find out how far is the system performance based on a feasible solution from the optimal one, and how to improve heuristic methods based on the quality of the solution. Since the charging behaviors of the MCs are coordinated by the BS (which has more computing resources than the sensor nodes and the MCs), BS is able to run complex MC coordination algorithms, including algorithm for optimal solution.

In this work, we address the multiple MCs coordination problem to perform the sensors charging task. The primary goal is to achieve perpetual operation, i.e., guarantee that the residual energy of each sensor node will never fall below the operational energy level. On this basis, we aim to minimize the total energy consumption of MCs by determining the scheduling of the MCs and adjusting their moving and charging time. According to this work, we answer the following questions: 1) What is the condition to achieve perpetual operation? 2) If this is possible, how to schedule the MCs to perform the charging tasks such that the total energy consumption of MCs is minimized? 3) Is there a way to achieve optimal solution while avoiding high computational complexity? The main distinction of this paper from previous work is that: the scheduling, the charging time and the moving time of multiple MCs are jointly optimized, under the constraint that the energy budget of the MCs is limited and none of the sensors will run out of its energy.

A. Related Work

In the literature, the MC scheduling is usually converted into the traveling salesman problem (TSP) [5], where a MC constructs a tour of all nodes only once. For the multiple MCs case, the TSP can be extended to the multiple traveling salesman problem (m-TSP) [7]. Based on the weighted sum of the traveling time and the residual lifetime of nodes, a heuristic algorithm is proposed in [7] to select the nodes to recharge. In [8], the point-to-multipoint energy transfer is considered and the aim is to maximize the number of nodes that are charged at each stop to reduce the charging delay. A TSP with multiple time windows (TSPMTW) is formulated and solved using constraint programming model. The aim in [9] is to minimize

the sum of traveling distances of all chargers. This can be formulated as a q-root TSP problem, i.e., find q closed tours covering all locations such that the total length of the q tours is minimized. Considering WRSNs with negligible charging time, the problem of minimizing the number of MCs for maintaining the operations of the networks is studied in [10]. This problem can be optimally solved with linear complexity. The problem of maximizing the network coverage is considered in [11], where each sensor monitors a circular area. The focus of [1] is to jointly schedule the MCs and adjust their charging time so that the energy consumption of the MCs is minimized. The authors design a decentralized method to find the optimal solution. However, the energy of the MCs is unlimited in all above studies, compared to our work.

The MC energy is limited in [12]–[16]. The authors in [12] jointly optimize the BS positioning and the MC scheduling. They maximize the charging efficiency, while reducing the number of required MCs and BSs. The considered problem is solved by a three-step design: 1) scheduling planning, 2) candidate BS identification, and 3) BS deployment and scheduling assignment. In [13], the aim is to maximize energy efficiency, while including no sensor node outage. A profitable TSP considering deadlines is formulated and a greedy algorithm and an adaptive algorithm are used to solve the problem. In [14], the problem of minimizing the number of the chargers is proved to be NP-hard and an approximation algorithm is proposed to solve the problem. Under similar context to [9], a q-root TSP is developed in [15] to schedule the MCs, while further aiming to minimize the number of MCs. In [16], based on the energy status of the sensors and the MCs, the network is divided into several regions. Each MC is assigned to a network region and charges the sensors in that region. However, the above studies mainly focus on the heuristic methods, compared to our work.

To enhance the flexibility of the charging process, the MCs are allowed to transfer energy not just to the sensor nodes, but also between themselves [17], [18]. The MCs are divided into two groups in [19], one that charges the sensor nodes and the other charges the MCs. Such kind of two-layer architecture can be considered as future work. Moreover, the above studies mainly focus on the MC scheduling problem, since the MC charging time optimization is not taken into account except in [1]. Note that different scheduling schemes lead to different charging time decision. The scheduling problem and the charging time problem should be jointly addressed to find the optimal solution.

With respect to the coordination mechanism, some works propose distributed approach. In [16], [19], the distributed mechanism exists among the region partition phase, i.e., the MCs communicate with each other to make the region partition decision. The results in [16], [19] show that the performance of the distributed coordination is lower than the centralized one, since distributed coordination utilizes local network information instead of global information to make the decision. In [1], the MCs communicate with each other to make the charging time decision under the given MC scheduling decision determined by the BS. However, the actions between the sensors and the MCs are still coordinated by the BS in [1], [16], [19]. The BS collects and processes the energy information from the sensors and sends it to the MCs. Different from our preliminary results in [1] where the MC energy constraints are not taken into account and the solution is based on classical Benders decomposition (BD) approach [20], this paper considers that the MC energy budget is limited and proposes a novel solution

to further reduce the computation time.

B. Contributions

Our main contributions focus on how to formulate the joint-design problem and how to solve this problem efficiently. More precisely:

1) We propose a novel multiple MCs coordination framework for WRSNs that jointly optimizes the scheduling, the moving time and the charging time of the MCs. Solving these correlated problems separately leads to non-optimal solution. To enhance charging efficiency, based on the energy consumption model of the sensor nodes and the charging model of the MCs, we derive a lifetime-based charging sequence for the sensor nodes and divide the sensor nodes into several groups according to the number of available MCs. We also obtain a set of desired network properties, such as how to determine a sensor node that requires charging in the current charging cycle or not, and how much energy a sensor node should be replenished so as to achieve perpetual operation. The joint-design problem of minimizing the energy consumption of the MCs, while guaranteeing the perpetual operation of the WRSNs, is formulated as an MILP problem.

2) We present an Optimal multiple MCs Coordination algorithm, referred to as OMC, to solve the joint-design problem. Since the moving time and the charging time of the MCs is influenced by the MC scheduling decision, the OMC decomposes the overall problem into an integer linear programming (ILP)-based master problem (MP) for the MC scheduling and a linear programming (LP)-based slave problem (SP) for the MC moving and charging time. We prove that by solving the MP and the SP in each iteration, we obtain a lower bound and an upper bound for the overall problem. Moreover, in each iteration, by constructing a new constraint according to the solution to the SP and adding this new constraint to the MP in next iteration, the gap between the lower bound and the upper bound gradually reduces. Unlike the classical BD approach, we prove that by relaxing the MP to an LP to find a feasible solution and by replacing the optimal solution to the MP with the feasible solution during the iteration between the MP and the SP, the optimality of the solution to the overall problem is still guaranteed.

3) Finally, we conduct extensive simulations results to analyze the quality of the solution, the computation time and the scalability of the proposed OMC algorithm. The obtained results show that the proposed method is able to achieve the optimal solution with reduced computation time compared to state-of-the-art optimal methods. Moreover, we present how we can further balance the computation time with the quality of the solution by controlling the iteration process between the MP and the SP.

C. Paper Organization

The remainder of this paper is organized as follows. Section II presents the system model and formulates the problem. Then, the multiple MCs coordination mechanism is described in Section III. Finally, Section IV shows the simulation results and Section V concludes this work.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first present the sensor energy consumption model and the wireless energy charging model. Then, we formulate the multiple MCs coordination problem as an MILP that takes perpetual operation and energy efficiency into account. The notations followed in this paper are: for a matrix $\boldsymbol{a}$, a_{ij} is the $(i,j)^{th}$ element of $\boldsymbol{a}$; for a vector $\boldsymbol{b}$, b_i is the ith element of $\boldsymbol{b}$; $(\cdot)^T$ is the operator for the transpose of a matrix/vector. Let $\boldsymbol{x} = [x_1 \ldots, x_n]^T$ and $\boldsymbol{y} = [y_1 \ldots, y_n]^T$. $\boldsymbol{x} \preceq \boldsymbol{y}$ represents $x_i \leq y_i$, $(1 \leq i \leq n)$. In the following, node is used in the place of sensor node.

A. System Model

We consider a WRSN with n nodes $\{s_1, \ldots, s_n\}$ and m MCs $\{c_1, \ldots, c_m\}$. For each node s_i $(1 \leq i \leq n)$, its energy is mainly consumed for the data transmitting and receiving. We adopt the following energy consumption model [21]:

$$r_i = \rho \sum_{k=1, k \neq i}^{n} f_{ki} + \sum_{j=1, j \neq i}^{n} C_{ij} f_{ij} + C_{ib} f_{ib}, \quad (1)$$

where f_{ki} (f_{ij}) and f_{ib} are the data flow rate from node s_k to node s_i (from node s_i to node s_j) and from node s_i to BS, respectively. ρ and C_{ij} (C_{ib}) are the rate of energy consumption for receiving a unit of data, and transmitting a unit of data from node s_i to node s_j (BS), respectively. We assume that the system runs with a given routing protocol [22], and, thus, the energy consumption rate r_i of node s_i is invariant with time.

On the other hand, we consider the following wireless energy charging model [23], [24]:

$$p_r = \frac{G_s G_r \kappa}{L_p} \left[\frac{\omega}{4\pi(d+\zeta)} \right]^2 p_0 = \frac{\varsigma}{(d+\zeta)^2} p_0, \quad (2)$$

where $\varsigma = \frac{G_s G_r \kappa \omega^2}{16 L_p \pi^2}$, p_0 is the source power of the MC, p_r is the received power of the node, d is the distance between the MC and the node, G_s is the source antenna gain, G_r is the receive antenna gain, L_p is the polarization loss, ω is the wavelength, κ is the rectifier efficiency and ζ is a parameter to adjust the *Friis'* free space equation for the short distance transmission [25]. We consider point-to-point energy transfer [8], where a MC charges only one sensor node at a time by approaching it at a very close distance so the charging process has the maximum efficiency possible. As the experiment shown in [26], the charging efficiency $\eta = \frac{p_r}{p_o} \approx 6\%$ when $d \approx 0$. Moreover, we assume that all the MCs have the same charging power.

B. Problem Formulation

The parameters and the variables used during the problem formulation are summarized in Table II.

1) Determine Charging Sequence:

Definition 2.1: If n nodes are charged once, this process is called one charging cycle.

Definition 2.2: If m MCs are scheduled to charge nodes, at most m nodes can be charged simultaneously. This process is called one charging round.

Since the number of nodes is usually much larger than the number of MCs $(n \gg m)$, one charging cycle contains several charging rounds. To achieve perpetual operation, the charging process contains two phases: 1) it determines the sets of the nodes that are charged in each cycle and in each round,

Parameters

n	number of nodes
m	number of MCs
$d_{\max}$	maximum distance between two nodes
$e_{\max}$	maximum capacity of node battery
$e_{\min}$	minimum energy for a node to be operational
p_0	source power of MC
p_r	received power of node
η	charging efficiency of MC
r_i	energy consumption rate of node s_i
e_i^k	initial energy of node s_i in cycle $\mathcal{C}^k$
L_i^k	lifetime of node s_i
v	moving speed of MC c_j
τ_s	battery charging/replacement time of MC c_j
ϵ	moving energy related coefficient of MC c_j
E_j	initial energy of MC c_j in cycle $\mathcal{C}^k$
$E_{j,l}^k$	residual energy of MC c_j at the beginning of round $\mathcal{R}_l^k$
$t_{j,l}^{\max}$	maximum charging time of MC c_j in round $\mathcal{R}_l^k$
σ	maximum number of rounds in one cycle
ϱ	maximum charging and moving energy consumed by a MC in one round
$m_{l,1}^k$	number of schedulable MCs in the round $\mathcal{R}_l^k$ that estimated in round $\mathcal{R}_1^k$ $(l \geq 1)$
$\sigma_{i,1}^k$	number of rounds in cycle $\mathcal{C}^k$ that estimated in round $\mathcal{R}_1^k$ with respect to node s_i
h_l^k	number of schedulable MCs in round $\mathcal{R}_l^k$
p_l^k	number of nodes require charging in round $\mathcal{R}_l^k$
σ_i^k	number of rounds left in cycle $\mathcal{C}^k$ when charging node s_i
τ_l^k	total moving and charging time in rounds $\{\mathcal{R}_1^k, \ldots, \mathcal{R}_l^k\}$
τ_d	time interval between two adjacent rounds
θ_l^k	predefined time threshold
ϖ_l^k	maximum waiting time of nodes in round $\mathcal{R}_l^k$
ι_l^k	index of node in round $\mathcal{R}_l^k$ that has the shortest lifetime

Variables

$q_{ij,l}^k$	$= \begin{cases} 1 & \text{if MC } c_j \text{ is scheduled to charge node } s_i \\ 0 & \text{otherwise} \end{cases}$
$t_{ij,l}^k$	time of MC c_j spends to charge node s_i
$g_{ij,l}^k$	moving time of MC from node s_j to node s_i

TABLE II.   Main notations

and 2) it decides how much energy should be replenished for each node. It is worth noting that different nodes $\{s_1, \ldots, s_n\}$ have different initial energy levels $\{e_1^k, \ldots, e_n^k\}$ and different energy consumption rates $\{r_1, \ldots, r_n\}$. To evaluate the energy charging priorities of the nodes, we sort all the nodes according to their lifetimes in an increasing order:

$$\boldsymbol{L}^k = \{L_1^k, L_2^k, \ldots, L_n^k\}, \tag{3}$$

where $L_i^k = \frac{e_i^k - e_{\min}}{r_i}$ is the lifetime of node s_i. There is no need to charge all the nodes $\{s_1, \ldots, s_n\}$ in one cycle $\mathcal{C}^k$, since some nodes have enough energy to work until a later cycle (e.g., $\mathcal{C}^{k+1}$). Therefore, in cycle $\mathcal{C}^k$, all the nodes $\{s_1, \ldots, s_n\}$ can be divided into two sets: 1) the serving set $\mathcal{S}^k$, and 2) the

non-serving set $\bar{\mathcal{S}}^k$. If $s_i \in \mathcal{S}^k$, this node is charged in the cycle $\mathcal{C}^k$, else, it can be charged in a later cycle.

Lemma 2.1: The cycle $\mathcal{C}^k$ contains at most σ^k rounds, where σ satisfies the inequalities:

$$\sum_{l=1}^{\sigma^k - 1} m_{l,1}^k < n, \tag{4}$$

$$\sum_{l=1}^{\sigma^k} m_{l,1}^k \geq n. \tag{5}$$

Proof: For the MC c_j, the worst case happens when 1) all the nodes $\{s_1, \ldots, s_n\}$ require charging in one cycle, 2) the nodes charged by the MC c_j in each round require fully charging (i.e., the replenished energy is $e_{\max} - e_{\min}$), and 3) the distances between the nodes that are charged by the MC c_j in two adjacent rounds are the maximum (i.e. the moving distance is $d_{\max}$). Therefore, the maximum charging and moving energy consumed by the MC c_j in one round is

$$\varrho = \frac{e_{\max} - e_{\min}}{\eta} + d_{\max}\epsilon,$$

where ϵ is an energy related coefficient.

If the initial energy of the MC c_j in the cycle $\mathcal{C}^k$ is E_j, it can perform the charging task at least

$$\rho_j^k = \left\lfloor \frac{E_j}{\varrho} \right\rfloor = \left\lfloor \frac{E_j \eta}{e_{\max} - e_{\min} + d_{\max}\epsilon\eta} \right\rfloor$$

rounds (i.e., $\{\mathcal{R}_1^k, \ldots, \mathcal{R}_{\rho_j^k}^k\}$). Then, the MC c_j has to return to the BS for battery charging/replacement. This process takes at most $\tau_s + \frac{2d_{\max}}{v}$ time, where v is the moving speed of MC. Since the charging time and the moving time of the MC c_j in one round takes at most $\frac{e_{\max} - e_{\min}}{p_r} + \frac{d_{\max}}{v}$ time, the battery charging/replacement of the MC c_j takes at most

$$\phi = \left\lceil \frac{p_r(\tau_s v + 2d_{\max})}{v(e_{\max} - e_{\min}) + p_r d_{\max}} \right\rceil \tag{6}$$

rounds. Therefore, the MC c_j can be scheduled again after the round $\mathcal{R}_{\rho_j + \phi}^k$. We assume that the available energy of the MC c_j after each battery charging/replacement is the same (i.e., E_j).

Based on the number of rounds that the MCs $\{c_1, \ldots, c_m\}$ can charge the nodes (i.e., $\{\rho_1^k, \ldots, \rho_m^k\}$), we can estimate how many MCs are available in the round $\mathcal{R}_l^k$ (e.g., $m_{l,1}^k$ MCs). Note that n nodes require charging in the cycle $\mathcal{C}^k$. Based on the number of available MCs in each round (i.e., $\{m_{1,1}^k, m_{2,1}^k, \ldots\}$), we can estimate how many rounds are included in the cycle $\mathcal{C}^k$, e.g., σ^k rounds, where σ^k satisfies (4) and (5). ∎

Remark 2.1: In *Lemma* 2.1, the moving and charging energy consumed by the MC c_j in one round is assumed to be ϱ. However, the real consumed energy of the MC c_j is no more than ϱ, and, thus, the real number of rounds in the cycle $\mathcal{C}^k$ is no more than σ^k.

Lemma 2.2: If the lifetime of the node s_i $(1 \leq i \leq n)$ satisfies the inequality:

$$L_i^k < (\sigma_{i,1}^k + \sigma^{k+1} - 1)\varrho + \tau_d, \tag{7}$$

$s_i \in \mathcal{S}^k$, else, $s_i \in \bar{\mathcal{S}}^k$.

Proof: For the node s_i, the worst case happens when 1) all the nodes $\{s_1, \ldots, s_n\}$ require to be charged in the cycles $\mathcal{C}^k$ and $\mathcal{C}^{k+1}$, 2) the charging time and the moving time of the MC in each round of the cycles $\mathcal{C}^k$ and $\mathcal{C}^{k+1}$ is maximum (i.e.,

ϱ), and 3) this node is charged again in the last round of cycle $\mathcal{C}^{k+1}$. As illustrated in Fig. 1, if the lifetime of node s_1 satisfies the inequality $L_1^k \geq (\sigma_{1,1}^k + \sigma^{k+1} - 1)\varrho + \tau_d$, $s_1 \in \bar{\mathcal{S}}^k$, else, $s_1 \in \mathcal{S}^k$. This inequality implies that if s_1 is a non-serving node, it should have the enough energy to work $\sigma_{1,1}^k$ rounds in the current cycle $\mathcal{C}^k$ and $\sigma^{k+1} - 1$ rounds in the next cycle $\mathcal{C}^{k+1}$, where each round in the cycles $\mathcal{C}^k$ and $\mathcal{C}^{k+1}$ takes ϱ time.

For the ith node in the sequence $\boldsymbol{L}^k$ (i.e., s_i), we assume that n_i nodes before it do not require charging and all the nodes after it require charging. Therefore, $n - n_i$ nodes require charging in the cycle $\mathcal{C}^k$. Based on the number of available MCs in each round (i.e., $\{m_{1,1}^k, m_{2,1}^k, \ldots\}$), we can estimate how many rounds exist in the cycle $\mathcal{C}^k$ (e.g., $\sigma_{i,1}^k$ rounds). Therefore, if (7) is satisfied, $s_i \in \mathcal{S}^k$, else, $s_i \in \bar{\mathcal{S}}^k$. ∎
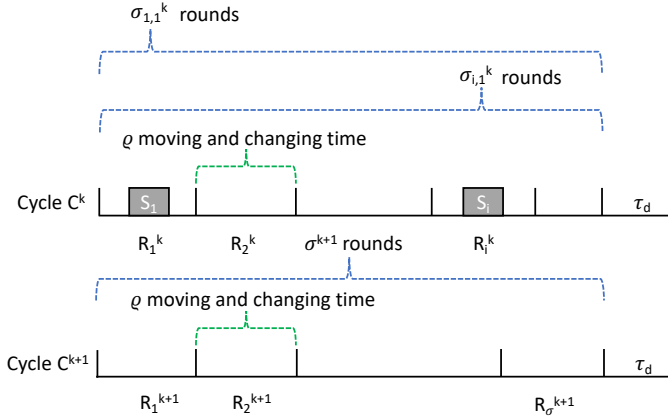

Fig. 1. An example to determine if node s_i requires charging in the current cycle $\mathcal{C}^k$ or not.

Remark 2.2: Note that 1) the real number of rounds in the cycle $\mathcal{C}^{k+1}$ is no more than σ, 2) the node s_i may not be charged in the last round of cycle $\mathcal{C}^{k+1}$, and 3) some nodes after the node s_i may not require charging. The real waiting time of node s_i is no more than $(\sigma_{i,1}^k + \sigma^{k+1} - 1)\varrho + \tau_d$.

Let $\tilde{n}^k$ ($\tilde{n}^k \leq n$) denote the number of nodes in the serving set $\mathcal{S}^k$. Therefore, the refined charging sequence is

$$\tilde{\boldsymbol{L}}^k = \{\underbrace{L_1^k, \ldots, L_{m_{1,1}^k}^k}_{\mathcal{R}_1^k}, \underbrace{L_{m_{1,1}^k+1}^k, \ldots, L_{m_{1,1}^k+m_{2,1}^k}^k}_{\mathcal{R}_2^k}, \ldots, L_{\tilde{n}^k}^k\},$$
(8)

and the refined charging cycle is defined as

Definition 2.3: One charging cycle means that the nodes in the serving set $\mathcal{S}^k$ are charged once.

Based on the refined charging sequence given by (8), we can schedule the MCs to charge the nodes.

2) Optimize Charging Behavior: Without loss of generality, we assume that the current round is $\mathcal{R}_l^k$. If the residual energy of the MC c_j satisfies the inequality:

$$E_{j,l}^k \geq \frac{e_{\max} - e_{\min}}{\eta} + 2d_{\max}\epsilon,$$
(9)

the MC c_j is available in the round R_l^k, else, it has to return to the BS for battery charging/replacement. (9) shows that the available MC should have enough energy for moving, charging a node and returning to the BS. We assume that h_l^k ($h_l^k \leq m$) MCs are available in the round $\mathcal{R}_l^k$.

Lemma 2.3: If the replenished energy of node s_i satisfies the constraint:

$$\underline{E}_i^k \leq \sum_{j=1}^{h_l^k} p_r t_{ij,l}^k \leq \overline{E}_i^k,$$
(10)

where

$$\underline{E}_i^k = [(\sigma_i^k + \sigma^{k+1} - 1)\varrho + \tau_d]r_i - (e_i^k - \tau_{l-1}^k r_i),$$
(11)

$$\overline{E}_i^k = e_{\max} - (e_i^k - \tau_{l-1}^k r_i),$$
(12)

no matter in which round of the next cycle this node is placed, it will never deplete its residual energy before being charged again.

Proof: If the residual energy of a node is larger than the energy consumed during the waiting time to be charged, this node can work perpetually. For the ith node in the sequence $\tilde{\boldsymbol{L}}^k$ (i.e., s_i), the worst case happens when 1) all the nodes after it (i.e., $\{s_{i+1}, \ldots, s_{\tilde{n}}\}$) require charging in the current cycle $\mathcal{C}^k$, 2) all the nodes (i.e., $\{s_1, \ldots, s_n\}$) require charging in the next cycle $\mathcal{C}^{k+1}$, 3) each round in the cycles $\mathcal{C}^k$ and $\mathcal{C}^{k+1}$ takes ϱ time, and 4) this node is charged again in the last round of cycle $\mathcal{C}^{k+1}$.

With the residual energy $E_{j,l}^k$, the MC c_j is able to perform the charging task from round $\mathcal{R}_l^k$ to round $\mathcal{R}_{l+\rho_{j,l}}^k$. Then, it is available again after the round $\mathcal{R}_{l+\rho_{j,l}^k+\phi}^k$, where

$$\rho_{j,l}^k = \left\lfloor \frac{E_{j,l}^k}{\varrho} \right\rfloor = \left\lfloor \frac{E_{j,l}^k \eta}{e_{\max} - e_{\min} + d_{\max}\epsilon\eta} \right\rfloor,$$

and ϕ is given by (6). Therefore, we can estimate how many MCs can be scheduled in the rounds $\{\mathcal{R}_l^k, \mathcal{R}_{l+1}^k, \ldots\}$ (e.g., $\{m_{l,l}^k, m_{l+1,l}^k, \ldots\}$). Note that the number of uncharged nodes is known when charging the node s_i (i.e., $\tilde{n} - i + 1$ sensors). Based on the number of available MCs in the later rounds (i.e., $\{m_{l,l}^k, m_{l+1,l}^k, \ldots\}$), we can estimate how many rounds are left in the cycle $\mathcal{C}^k$ (e.g., σ_i^k rounds). Since the cycle $\mathcal{C}^{k+1}$ contains at most σ^{k+1} rounds, the maximum waiting time of node s_i to be charged again is $(\sigma_i^k + \sigma^{k+1} - 1)\varrho + \tau_d$. On the other hand, since each node s_i has a maximum energy level $e_{\max}$, the replenished energy of node s_i should not exceed $e_{\max} - (e_i - \tau_{l-1}^k r_i)$, where $e_i - \tau_{l-1}^k r_i$ is the residual energy of node s_i in the current round $\mathcal{R}_l^k$. Therefore, we have (10). ∎

When determining the sequence (8), we consider the worst case, which means $s_i \in \bar{\mathcal{S}}^k$ even if (7) is satisfied. Substituting (7) into (11), we have

$$\underline{E}_i^k > [\tau_{l-1}^k + (\sigma_i^k - \sigma_{i,1}^k)\varrho]r_i - e_{\min}.$$

Based on different values of the parameters, the lower bound of replenished energy $\underline{E}_i^k$ can be either positive or negative. If $\underline{E}_i^k < 0$, $s_i \in \bar{\mathcal{S}}$, since this node has enough energy to work until at the end of next cycle, and, thus, it can be removed from the charging sequence. Note that in the last round of a cycle, the number of sensors is no more than the number of MCs. We assume that p_l^k ($p_l^k \leq h_l^k$) nodes require charging in the round $\mathcal{R}_l^k$.

Since one MC is responsible for at most one node and each node is charged by one MC, the MC scheduling variable $q_{ij,l}^k$ is bounded by

$$\sum_{i=1}^{p_l^k} q_{ij,l}^k \leq 1, \ 1 \leq j \leq h_l^k,$$
(13)

$$\sum_{j=1}^{h_l^k} q_{ij,l}^k = 1, \ 1 \le i \le p_l^k. \tag{14}$$

On the other hand, since the charging time of the MC c_j in the round $\mathcal{R}_l^k$ is limited by its maximum charging time $t_{j,l}^{\max}$, we have

$$0 \le t_{ij,l}^k \le t_{j,l}^{\max} q_{ij,l}^k, \ 1 \le i \le p_l^k, \ 1 \le j \le h_l^k. \tag{15}$$

Note that in *Lemma* 2.3, the maximum waiting time of a node s_i is assumed to be $(\sigma_i^k + \sigma^{k+1} - 1)\varrho + \tau_d$. It means the interval between two adjacent rounds $\mathcal{R}_l^k$ and $\mathcal{R}_{l+1}^k$ should not exceed ϱ. Therefore, we introduce a (continuous) variable $g_{ij,l}^k$ to adjust the moving time of the MCs, and let the variables $q_{ij,l}^k$, $t_{ij,l}^k$ and $g_{ij,l}^k$ satisfy the constraints:

$$g_{ij,l}^k \ge q_{ij,l}^k \frac{d_{ij}}{v}, \ 1 \le i \le p_l^k, \ 1 \le j \le h_l^k, \tag{16}$$

$$0 \le t_{ij,l}^k + g_{ij,l}^k \le \theta_l^k, \ 1 \le i \le p_l^k, \ 1 \le j \le h_l^k, \tag{17}$$

where the time threshold θ_l^k is given by (18). For the nodes that are charged in the round $\mathcal{R}_l^k$, the number of the nodes in the later rounds is the same, and, thus, the number of remaining residual rounds is the same, which is denoted as σ_l^k. We assume that n_{l-1}^k nodes require charging and $\bar{n}_{l-1}^k$ nodes do not require charging in the previous rounds $\{\mathcal{R}_1^k, \dots, \mathcal{R}_{l-1}^k\}$.

Lemma 2.4: To ensure that the interval between two adjacent rounds is smaller than ϱ, we can set

$$\theta_l^k = \min\{\varrho, \varpi_{l+1}^k, \dots, \varpi_{l+\sigma_l^k}^k\}, \tag{18}$$

where $\varpi_j^k = L_{\iota_j^k} - \tau_{l-1}^k - (j-l)\varrho - \frac{d_{\max}}{v}$ and $\iota_j^k = n_{l-1}^k + \bar{n}_{l-1}^k + \sum_{i=l}^{j-1} m_{i,l}^k + 1 \ (l+1 \le j \le l+\sigma_l^k)$.

Proof: To determine the value of time threshold θ_l^k, we need to know how long the nodes in the later rounds $\{\mathcal{R}_{l+1}^k, \dots, \mathcal{R}_{l+\sigma_l}^k\}$ can survive. Note that $m_{l,l}^k$ MCs can be scheduled in the round $\mathcal{R}_l^k$ and the previous rounds $\{\mathcal{R}_1^k, \dots, \mathcal{R}_{l-1}^k\}$ contain $n_{l-1}^k + \bar{n}_{l-1}^k$ nodes. The node $s_{\iota_{l+1}^k}$ in the next round $\mathcal{R}_{l+1}^k$ has the shortest lifetime, where $\iota_{l+1}^k = n_{l-1}^k + \bar{n}_{l-1}^k + m_{l,l}^k + 1$. Since $q_{ij,l}^k$, $t_{ij,l}^k$, and $g_{ij,l}^k$ are variables, the exact charging and moving time in the round $\mathcal{R}_l^k$ is unknown. Therefore, we set $\tau_l^k = \tau_{l-1}^k + \varrho$, since ϱ is an upper bound of the charging and moving time in one round. To guarantee that the nodes in the round $\mathcal{R}_{l+1}^k$ can be charged in time, the maximum waiting time of the nodes in the round $\mathcal{R}_{l+1}^k$ should not exceed $\varpi_{l+1}^k = L_{\iota_{l+1}^k} - \tau_l^k - \frac{d_{\max}}{v} = L_{\iota_{l+1}^k} - \tau_{l-1}^k - \varrho - \frac{d_{\max}}{v}$.

For the nodes in the later round $\mathcal{R}_j^k$ $(l+1 \le j \le l+\sigma_l^k)$, the maximum waiting time should not exceed $\varpi_j^k = L_{\iota_j^k} - \tau_{j-1}^k - \frac{d_{\max}}{v} = L_{\iota_j^k} - \tau_{l-1}^k - (j-l)\varrho - \frac{d_{\max}}{v}$, where $\iota_j^k = n_{l-1}^k + \bar{n}_{l-1}^k + \sum_{i=l}^{j-1} m_{i,l}^k + 1$. Note that the time threshold θ_l^k should not exceed the maximum charging and moving time ϱ. Therefore, we have (18). ∎

Theorem 2.1: The sufficient condition to achieve perpetual operation is that the constraints (10), (16) – (18) are satisfied.

Proof: The constraint (10) makes sure that all nodes will not run out of their residual energies before being charged again, under the condition that the interval between two adjacent rounds is smaller than a predefined time threshold, which is guaranteed by the constraints (16) – (18). ∎

For the objective function, we consider minimizing the total energy consumption of the MCs. Based on this objective and all the aforementioned constraints, the Primal Problem (PP) is formulated as

$$\textbf{PP}: \min_{\boldsymbol{Q}_l^k, \boldsymbol{T}_l^k, \boldsymbol{G}_l^k} \Phi = \sum_{i=1}^{p_l^k} \sum_{j=1}^{h_l^k} (q_{ij,l}^k d_{ij}\epsilon + p_0 t_{ij,l}^k) \tag{19}$$
$$\text{s.t. } (10) - (18).$$

where $\boldsymbol{q}_l^k = [q_{ij,l}^k]$, $\boldsymbol{t}_l^k = [t_{ij,l}^k]$, and $\boldsymbol{g}_l^k = [g_{ij,l}^k]$. Since the binary and the continuous variables are coupled with each other linearly, the PP is an MILP problem.

III. MULTIPLE MCs COORDINATION SCHEME

In this section, we present an optimal multiple MCs coordination (OMC) algorithm to solve PP. The structure of OMC is shown in Fig. 2. To solve PP, the most important step is to find a proper MC scheduling decision $\boldsymbol{q}_l^k$. If the value of $\boldsymbol{q}_l^k$ is determined, PP is reduced to a LP problem, which has a simpler structure and it is easier to solve. Based on this idea, OMC tackles the problem by iteratively solving an ILP-based master problem (MP) to determine the MC scheduling and a LP-based slave problem (SP) to determine the MC charging and moving time.
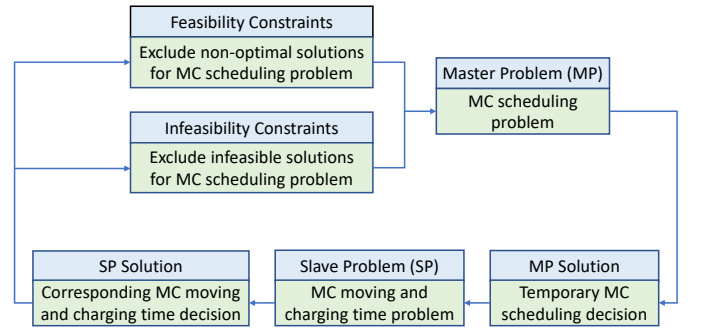

Fig. 2. The structure of OMC algorithm.

A. Formulations of MP and SP

For convenience, the matrices and the vectors are used to denote the constraints and the variables. Therefore, the PP is reformulated as

$$\textbf{PP1}: \min_{\boldsymbol{x},\boldsymbol{y}} \Phi(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{g}^T\boldsymbol{x} + \boldsymbol{f}^T\boldsymbol{y} \tag{20}$$
$$\text{s.t. } \begin{cases} \boldsymbol{A}\boldsymbol{x} \preceq \boldsymbol{b}_1, \\ \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{y} \preceq \boldsymbol{b}_2, \end{cases}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors of binary and continuous variables, respectively. $\boldsymbol{g}$ and $\boldsymbol{f}$ are the vectors of the objective function coefficients. $\boldsymbol{A}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are the matrices of the coefficients in the constraints. $\boldsymbol{b}_1$ is an u-dimensional vector and $\boldsymbol{b}_2$ is an v-dimensional vector.

Note that the objective function of the PP1 contains the binary variables $\boldsymbol{x}$ as well as the continuous variables $\boldsymbol{y}$, while the MP only considers the binary variables $\boldsymbol{x}$. To facilitate the iteration between the MP and the SP, we introduce an auxiliary (continuous) variable $\hat{\Phi}$ into the MP as the objective function, where $\hat{\Phi}$ and Φ have the same physical meaning. Based on the

structure of the PP1, the corresponding MP is

$$\textbf{MP} : \Phi_L = \min_{\boldsymbol{x}, \hat{\Phi}} \ \hat{\Phi} \tag{21}$$

$$\text{s.t.} \begin{cases} \boldsymbol{Ax} \preceq \boldsymbol{b}_1, \\ C_1 : \hat{\Phi} \geq \boldsymbol{g}^T \boldsymbol{x} + \boldsymbol{\lambda}(i)^T (\boldsymbol{Cx} - \boldsymbol{b}_2), \ \forall i \in \mathcal{A}, \\ C_2 : 0 \geq \boldsymbol{\varphi}(j)^T (\boldsymbol{Cx} - \boldsymbol{b}_2), \ \forall j \in \mathcal{B}, \end{cases}$$

where C_1 and C_2 are the sets of feasibility constraints (FCs) and infeasibility constraints (ICs), respectively. They are generated from the solution to the dual slave problem (DSP) (23). $\mathcal{A}$ and $\mathcal{B}$ are the sets of iterations where the DSP has bounded and unbounded solutions, respectively. $\boldsymbol{\lambda}(i)$ is the solution to the DSP at the ith iteration, while $\boldsymbol{\varphi}(j)$ is the solution to the dual feasibility check problem (DFCP) (26) at the jth iteration.

Let $(\boldsymbol{x}(l), \hat{\Phi}(l))$ denote the solution to the MP at the lth iteration. Therefore, the corresponding MP is

$$\textbf{SP} : \Phi_U = \min_{\boldsymbol{y} \succeq 0} \ \Phi(\boldsymbol{x}(l), \boldsymbol{y}) = \boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{f}^T \boldsymbol{y} \tag{22}$$

$$\text{s.t.} \ \ \boldsymbol{Cx}(l) + \boldsymbol{Dy} \preceq \boldsymbol{b}_2,$$

Comparing the SP with the PP1, we observe that their formulations are the same, except that the binary variables $\boldsymbol{x}$ in the SP are fixed.

Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ denote the optimal solution to the PP1. We have $\Phi^* = \Phi(\boldsymbol{x}^*, \boldsymbol{y}^*)$. Compared with the PP1, the MP only considers the MC scheduling variables $\boldsymbol{x}$, whereas the constraints with respect to the charging and moving time variables $\boldsymbol{y}$ are relaxed. Solving the MP yields a lower bound Φ_L of Φ^*. On the other hand, since the MC scheduling decision $\boldsymbol{x}(l)$ may be just a feasible solution (not optimal yet), solving the SP with $\boldsymbol{x}(l)$ yields an upper bound Φ_U of Φ^*. Therefore, we have $\Phi_L \leq \Phi^* \leq \Phi_U$. To reduce the gap between Φ_L and Φ_U, a new constraint FC (or IC) is added into C_1 (or C_2) at each iteration. When the gap is smaller than a predefined threshold ε, the optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is found.

B. Iterations between MP and SP

1) Step 1 - Initialization: Initialize the iteration counter $l = 0$, the MP solution $\boldsymbol{x}(0)$, the lower bound $\Phi_L = -\infty$, and the upper bound $\Phi_U = \infty$. The sets C_1 and C_2 are set to null. The initial solution $\boldsymbol{x}(0)$ can be given arbitrarily, as long as it satisfies the constraint $\boldsymbol{Ax}(0) \preceq \boldsymbol{b}_1$.

2) Step 2 - Solving SP: In this paper, rather than solving the SP directly, we solve its dual problem. This is because the SP and the DSP are equivalent due to the strong duality [27], and the new constraints can be constructed according to the solution of the DSP.

To construct the dual of the SP, we introduce Lagrange multipliers $\boldsymbol{\lambda} \triangleq [\lambda_i]$ $(1 \leq i \leq v)$ to the SP. Therefore, the DSP is

$$\textbf{DSP} : \max_{\boldsymbol{\lambda} \succeq 0} \ \boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{\lambda}^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2) \tag{23}$$

$$\text{s.t.} \ \ \boldsymbol{f} + \boldsymbol{D}^T \boldsymbol{\lambda} \succeq 0.$$

Since the DSP is an LP, it can be solved very fast using standard algorithms, such as simplex method or interior point method [28].

3) Step 3 - Solving MP: Based on the solution to the DSP, we have:

1) If the DSP is *infeasible*, the SP has an unbounded solution. Therefore, the PP1 is infeasible.

2) If the DSP has a *bounded solution* $\boldsymbol{\lambda}(l)$, $\mathcal{A} \leftarrow \{m\} \cup \mathcal{A}$. We have $\boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{f}^T \boldsymbol{y}(l) = \boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{\lambda}(l)^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2)$ due to the strong duality, where $\boldsymbol{y}(l)$ is the solution to the SP at the lth iteration. At the same time, the upper bound $\Phi_U(l)$ is updated by

$$\Phi_U(l) = \min\{\Phi_U(l-1), \boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{\lambda}(l)^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2)\}.$$

Since we have

$$\hat{\Phi}(l) < \boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{f}^T \boldsymbol{y}(l) = \boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{\lambda}(l)^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2),$$

the stopping criterion is not met, and, thus, $\boldsymbol{x}(l)$ is a non-optimal solution to the PP1. To avoid selecting again the non-optimal solution $\boldsymbol{x}(l)$, a new FC

$$\hat{\Phi} \geq \boldsymbol{g}^T \boldsymbol{x} + \boldsymbol{\lambda}(l)^T (\boldsymbol{Cx} - \boldsymbol{b}_2) \tag{24}$$

is generated and added into C_1 at the $(l+1)^{th}$ iteration.

3) If the DSP has an *unbounded solution* (i.e., $\boldsymbol{g}^T \boldsymbol{x}(l) + \boldsymbol{\lambda}(l)^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2) = +\infty$), $\mathcal{B} \leftarrow \{l\} \cup \mathcal{B}$. Due to the strong duality, the SP has no feasible solution under the given $\boldsymbol{x}(l)$. Note that the feasibility of the SP is related to the constraints rather than the objective function. This problem may be feasible if the positive (continuous) variables $\boldsymbol{\xi} \triangleq [\xi_i]$ $(1 \leq i \leq v)$ are introduced to relax the constraints. Based on this idea, we construct a Feasibility Check Problem (FCP)

$$\textbf{FCP} : \min_{\boldsymbol{\xi}, \boldsymbol{y} \succeq 0} \ \boldsymbol{1}^T \boldsymbol{\xi} \tag{25}$$

$$\text{s.t.} \ \ \boldsymbol{Cx}(l) + \boldsymbol{Dy} \preceq \boldsymbol{b}_2 + \boldsymbol{\xi}.$$

Since the FCP is an LP, the strong duality exists between the FCP and its dual problem. Instead of solving FCP, we solve its dual problem. To construct the dual of the FCP, we introducing Lagrange multipliers $\boldsymbol{\varphi} = [\varphi_i]$ $(1 \leq i \leq v)$ to the FCP. Therefore, the dual of the FCP (DFCP) is

$$\textbf{DFCP} : \max_{\boldsymbol{\varphi} \succeq 0} \ \boldsymbol{\varphi}^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2) \tag{26}$$

$$\text{s.t.} \begin{cases} \boldsymbol{1} - \boldsymbol{\varphi} \succeq 0, \\ \boldsymbol{D}^T \boldsymbol{\varphi} \succeq 0. \end{cases}$$

The DFCP can be solved by a similar method used to solve the DSP. Let $\boldsymbol{\xi}(l)$ and $\boldsymbol{\varphi}(l)$ denote the solutions to the FCP and the DFCP at the lth iteration, respectively. If SP exists infeasible constraints, the corresponding variables are non-zero, while the others are zero. Due to the strong duality, we have

$$\boldsymbol{1}^T \boldsymbol{\xi}(l) = \boldsymbol{\varphi}(l)^T (\boldsymbol{Cx}(l) - \boldsymbol{b}_2) > 0.$$

To avoid selecting again the infeasible solution $\boldsymbol{x}(l)$, a new IC

$$0 \geq \boldsymbol{\varphi}(l)^T (\boldsymbol{Cx} - \boldsymbol{b}_2) \tag{27}$$

is generated and added into C_2 at the $(l+1)^{th}$ iteration. Since solving DFCP and solving FCP are equivalent, and $\boldsymbol{\varphi}^T (\boldsymbol{Cx} - \boldsymbol{b}_2)$ is a function with respect to $\boldsymbol{x}$ but not $\boldsymbol{1}^T \boldsymbol{\xi}$ (i.e., $0 \geq \boldsymbol{1}^T \boldsymbol{\xi}$ is an invalid constraint for the MP), we construct the IC through the solution to the DFCP rather than the solution to the FCP.

For the new generated FC (24) or IC (27), all the parameters are constant except $\hat{\Phi}$ and $\boldsymbol{x}$. They are the variables to the MP. When the MP is solved, the iteration counter l increases, and Step 2 to Step 3 is repeated. The iteration stops when $\Phi_U(l) - \Phi_L(l) \leq \varepsilon$ is satisfied.

C. Convergence Analysis

From the MP (21), we observe that the real constraints are $\boldsymbol{Ax} \preceq \boldsymbol{b}_1$ and C_2. C_1 can be treated as the objective function. Therefore, the MP can be solved by only considering the binary variables $\boldsymbol{x}$. Let $\boldsymbol{\lambda}(k)$ denote the bounded solution to the DSP at the kth iteration ($k \in \mathcal{A}$). Comparing the following ILP problem

$$\hat{\Phi}_r(k) = \min_{\boldsymbol{x}} \ \boldsymbol{g}^T\boldsymbol{x} + \boldsymbol{\lambda}^T(k)(\boldsymbol{Cx} - \boldsymbol{b}_2) \tag{28}$$

$$\text{s.t.} \begin{cases} \boldsymbol{Ax} \preceq \boldsymbol{b}_1, \\ C_2 : 0 \geq \boldsymbol{\varphi}(j)^T(\boldsymbol{Cx} - \boldsymbol{b}_2), \ \forall j \in \mathcal{B}, \end{cases}$$

with the MP, we have $\hat{\Phi}(l) = \max_{\forall k \in \mathcal{A}}\{\hat{\Phi}_r(k)\}$. Although the MP includes a continuous variable $\hat{\Phi}$, it can be solved by only considering the binary variables $\boldsymbol{x}$.

Lemma 3.1: The lower bound $\Phi_L(l)$ and the upper bound $\Phi_U(l)$ on the optimal objective function value Φ^* are derived from the solution to the MP and the SP, respectively.

Proof: Without loss of generality, we assume that $\hat{\Phi}(l) = \hat{\Phi}_r(k)$ ($k \in \mathcal{A}$). Therefore, we have

$$\hat{\Phi}(l) = \hat{\Phi}_r(k) = \min_{\boldsymbol{x}} \ \boldsymbol{g}^T\boldsymbol{x} + \boldsymbol{\lambda}^T(k)(\boldsymbol{Cx} - \boldsymbol{b}_2)$$

$$\leq \boldsymbol{g}^T\boldsymbol{x}^* + \boldsymbol{\lambda}^T(k)(\boldsymbol{Cx}^* - \boldsymbol{b}_2) \tag{29a}$$

$$\leq \max_{\boldsymbol{\lambda} \succeq 0} \ \boldsymbol{g}^T\boldsymbol{x}^* + \boldsymbol{\lambda}^T(\boldsymbol{Cx}^* - \boldsymbol{b}_2) \tag{29b}$$

$$= \Phi^*, \tag{29c}$$

where (29a) holds since $\boldsymbol{x}^*$ is not the optimal solution to the problem (28) with $\boldsymbol{\lambda}(k)$. (29b) holds since $\boldsymbol{\lambda}(k)$ is not the optimal solution to the problem (23) with $\boldsymbol{x}^*$. (29c) holds since solving the problem (23) with $\boldsymbol{x}^*$ we obtain Φ^*. From (29), we observe that $\hat{\Phi}(l)$ is a lower bound of Φ^*.

Depending on the solution of the DSP, the value of the objective function of the DSP can be either finite or infinite. If the DSP has an unbounded solution, i.e., $\boldsymbol{g}^T\boldsymbol{x}(l) + \boldsymbol{\lambda}(l)^T(\boldsymbol{Cx}(l) - \boldsymbol{b}_2) = +\infty$, it is obvious that $+\infty$ is an upper bound of Φ^*. Therefore, we focus on the case when the DSP has a bounded solution. Note that

$$\Phi_U(l) = \min\{\Phi_U(l-1), \boldsymbol{g}^T\boldsymbol{x}(l) + \boldsymbol{\lambda}(l)^T(\boldsymbol{Cx}(l) - \boldsymbol{b}_2)\}$$

$$= \min_{1 \leq i \leq l}\{\boldsymbol{g}^T\boldsymbol{x}(i) + \boldsymbol{\lambda}(i)^T(\boldsymbol{Cx}(i) - \boldsymbol{b}_2)\}, \tag{30}$$

and

$$\boldsymbol{g}^T\boldsymbol{x}(i) + \boldsymbol{\lambda}(i)^T(\boldsymbol{Cx}(i) - \boldsymbol{b}_2) = \min_{\boldsymbol{y} \succeq 0} \ \Phi(\boldsymbol{x}(i), \boldsymbol{y})$$

$$\geq \min_{\boldsymbol{y} \succeq 0} \ \Phi(\boldsymbol{x}^*, \boldsymbol{y}) = \Phi^*, \tag{31}$$

where (31) holds since $\boldsymbol{x}^*$ is the optimal solution to the problem (20). From (30) and (31), we observe that $\Phi_U(l)$ is an upper bound of Φ^*. ∎

Remark 3.1: Note that the SP has the same formulation as the PP1 except that the binary variables $\boldsymbol{x}$ in the SP are fixed. In addition, the SP and the DSP are equivalent due to the strong duality. From (29), we observe that the auxiliary variable $\hat{\Phi}$ has the same physical meaning as the objective function of DSP,

and, thus, $\hat{\Phi}$ has the same physical meaning as the objective function of PP1.

Lemma 3.2: The lower bound sequence $\{\Phi_L(0), \ldots \Phi_L(l)\}$ is increasing, while the upper bound sequence $\{\Phi_U(0), \ldots \Phi_U(l)\}$ is decreasing.

Proof: With iteration l increasing, more constraints are added into the MP. Therefore, the feasible region of the MP shrinks. Since the MP is a minimization problem, the non-optimal values of Φ^* (i.e., $\{\hat{\Phi}(0), \ldots, \hat{\Phi}(l)\}$) are excluded by the constraints C_1 and C_2, and, thus, $\Phi_L(l+1) = \hat{\Phi}(l+1)$ is larger than the previous lower bounds $\{\Phi_L(0), \ldots, \Phi_L(l)\}$.

We assume that $\Phi_U(l+1) > \Phi_U(l)$. This contradicts the fact that $\Phi_U(l+1) = \min\{\Phi_U(l), \boldsymbol{\lambda}(l+1)^T(\boldsymbol{Cx}(l+1) - \boldsymbol{b}_2)\}$. Therefore, $\Phi_U(l+1)$ is smaller than the previous upper bounds $\{\Phi_U(0), \ldots, \Phi_U(l)\}$. ∎

Theorem 3.1: With the FC (24) and the IC (27) added into the MP, the algorithm converges.

Proof: Based on *Lemma* 3.1 and *Lemma* 3.2, as well as the fact that the non-optimal and the infeasible solutions of the binary variables $\boldsymbol{x}$ are excluded by the FCs and the ICs, and the dimension of the binary variables $\boldsymbol{x}$ is finite, the algorithm converges in a finite number of iterations. ∎

Theorem 3.2: The FC and the IC generated by solving the DSP with $\bar{\boldsymbol{x}}(l)$ do not exclude the optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, where $\bar{\boldsymbol{x}}(l)$ is an arbitrary feasible solution to the MP at the lth iteration.

Proof: If the DSP has a bounded solution $\bar{\boldsymbol{\lambda}}(l)$ with $\bar{\boldsymbol{x}}(l)$, the corresponding FC is

$$\hat{\Phi} \geq \boldsymbol{g}^T\boldsymbol{x} + \bar{\boldsymbol{\lambda}}(l)^T(\boldsymbol{Cx} - \boldsymbol{b}_2). \tag{32}$$

On the other hand, if the DSP has an unbounded solution with $\bar{\boldsymbol{x}}(l)$, the corresponding IC is

$$0 \geq \bar{\boldsymbol{\varphi}}(l)^T(\boldsymbol{Cx} - \boldsymbol{b}_2), \tag{33}$$

where $\bar{\boldsymbol{\varphi}}(l)$ is solution to the DFCP with $\bar{\boldsymbol{x}}(l)$. In the following, we prove that the optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to the PP1 does not violate the constraints (32) and (33).

If the DSP has a bounded solution $\bar{\boldsymbol{\lambda}}(l)$ with $\bar{\boldsymbol{x}}(l)$, suppose that $\boldsymbol{x}^*$ and Φ^* violate the FC (32). Therefore, we have

$$\Phi^* < \boldsymbol{g}^T\boldsymbol{x}^* + \bar{\boldsymbol{\lambda}}(l)^T(\boldsymbol{Cx}^* - \boldsymbol{b}_2). \tag{34}$$

However, (34) contradicts the fact that Φ^* is the optimal value to the DSP with $\boldsymbol{x}^*$:

$$\Phi^* = \max_{\boldsymbol{\lambda} \succeq 0} \ \boldsymbol{g}^T\boldsymbol{x}^* + \boldsymbol{\lambda}(\boldsymbol{Cx}^* - \boldsymbol{b}_2)$$

$$\geq \boldsymbol{g}^T\boldsymbol{x}^* + \bar{\boldsymbol{\lambda}}(l)^T(\boldsymbol{Cx}^* - \boldsymbol{b}_2).$$

Therefore, the FC (32) does not exclude the optimal solution $\boldsymbol{x}^*$. On the other hand, if the DSP has an unbounded solution with $\bar{\boldsymbol{x}}(l)$, $\bar{\boldsymbol{x}}(l)$ is excluded by the IC (33). Since $\boldsymbol{x}^* \neq \bar{\boldsymbol{x}}(l)$, $\boldsymbol{x}^*$ does not violate the IC (33). ∎

Theorem 3.3: With the FC (32) and the IC (33) added into the MP, the algorithm converges.

Proof: Since the dimension of binary variables $\boldsymbol{x}$ is finite, according to *Theorem* 3.2, the solution converges to the global optimal value within a finite number of iterations. ∎

Remark 3.2: Since MP is an ILP, this problem is still hard to solve directly compared with the LP-based SP. Moreover, the size of MP will increase with the number of iterations, since at each iteration a new FC or IC is added into the MP. Therefore, the computational complexity of OMC is dominated by the cost of solving the MP at each iteration. Based on *Theorem* 3.3, we

can circumvent the above difficulties by replacing the optimal solution to the MP $\boldsymbol{x}(k)$ with the feasible solution $\bar{\boldsymbol{x}}(k)$ during the iteration between the MP and the SP. Such a feasible solution can be efficiently found by using the heuristics, such as feasibility pump (FP) method [29].

Algorithm 1 summarizes the implementation details of the OMC algorithm.

Algorithm 1: Optimal Multiple MCs Coordination (OMC) Algorithm

---

Input: System parameters $\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{f}, \boldsymbol{b}_1, \boldsymbol{b}_2$;

Output: Optimal scheduling decision $\boldsymbol{x}^*$, optimal charging and moving time decision $\boldsymbol{y}^*$;

Set initial values: $l = 0$, $\Phi_L(0) = -\infty$, $\Phi_L(0) = +\infty$, $\{\boldsymbol{x}(0)|\boldsymbol{Ax}(0) \preceq \boldsymbol{b}_1\}$, ε;

C_1 and C_2 are set to null;

while $\Phi_U(l) - \Phi_L(l) > \varepsilon$ **do**

    Solve MP (21) to obtain temporary scheduling decision $\bar{\boldsymbol{x}}(k)$;

    Update lower bound $\Phi_L(l) = \max_{\forall k \in \mathcal{A}}\{\hat{\Phi}_r(k)\}$;

    Solve DSP (23) with $\bar{\boldsymbol{x}}(k)$;

    if *DSP has bounded or unbounded solution* **then**

        if *Solution is bounded* **then**

            $\mathcal{A} \leftarrow \{l\} \cup \mathcal{A}$;

            Add FC: $\hat{\Phi} \geq \boldsymbol{g}^T \boldsymbol{x} + \bar{\boldsymbol{\lambda}}(l)^T(\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}_2)$ to C_1;

            Update upper bound $\Phi_U(l) = \min\{\Phi_u(l-1), \bar{\boldsymbol{\lambda}}(l)^T(\boldsymbol{C}\bar{\boldsymbol{x}}^T(l) - \boldsymbol{b}_2)\}$;

        else

            $\mathcal{B} \leftarrow \{l\} \cup \mathcal{B}$;

            Add IC: $0 \geq \bar{\boldsymbol{\varphi}}(l)^T(\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}_2)$ to C_2;

        end

    else

        PP1 (20) is infeasible;

    end

    $l \leftarrow l + 1$;

end

Optimal scheduling decision $\boldsymbol{x}^* = \boldsymbol{x}(l)$;

Submit $\boldsymbol{x}^*$ into PP1 and solve it to obtain optimal charging and moving time decision $\boldsymbol{y}^*$;

IV. SIMULATION

We consider a WRSN as result case study. The mobile robots and the sensor nodes (e.g., Mica2) that equipped with the Powercast chargers and receivers are employed as MCs and rechargeable sensor nodes, respectively. The nodes and the MCs are randomly deployed in a 100 m $\times$ 100 m area to monitor the environment and charge the nodes. The BS is at $(50, 50)$ m. The system parameters are summarized in Table III and are adopted from [21], [26]. For the energy model given by (1), λ_1 and λ_2 are the distance-independent and the distance-dependent constant terms, respectively. d_{ij} is the distance between the nodes s_i and s_j. θ is the path loss index. For a regular AA battery, its nominal cell voltage and the quantity of electricity is 1.2 V/2.5 Ah. Since two AA batteries provide an average voltage 2.4 V for the Mica2 node and the operating limit is 2.1 V, we set $e_{\min} = 2.1 \times 2.5 \times 3600 = 18900$ J and $e_{\max} = 2.4 \times 2.5 \times 3600 = 21600$ J. Note that different MCs and sensor nodes only change the values of the problem parameters $\{\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{g}, \boldsymbol{f}, \boldsymbol{b}_1, \boldsymbol{b}_2\}$. They do not affect the

structure of our problem formulation. The simulations are performed on a PC with quad-core 2.5 GHz Intel i7 processor and 16 GB RAM, and the algorithms are implemented in Matlab 2016a.

Sensor node s_i characteristics	
$r_i = \rho \sum_{k=1, k \neq i}^{n} f_{ki} + \sum_{j=1, j \neq i}^{n} C_{ij} f_{ij} + C_{ib} f_{ib}$	
$\rho = 50$ nJ/b	$f_{ij} \in [1, 10]$ kb/s
$C_{ij} = \lambda_1 + \lambda_2(d_{ij})^\theta$	$\lambda_1 = 50$ nJ/b
$\lambda_2 = 0.0013$ pJ/(b.m^4)	$\theta = 4$
$e_{\min} = 18900$ J	$e_{\max} = 21580$ J
$e_i^k \in [19000, 20000]$ J	$\Delta_s = 1$ s
Mobile charger c_j characteristics	
$\eta = 6$ %	$p_0 = 5$ W
$v = 1$ m/s	$\epsilon = 1$ J
$E_j^k \in [1000, 5000]$ J	$\tau_s = 10000$ s, $\tau_d = 50000$ s

Tuned parameters		
	n	m
Min/Max/Step	25/50/5	5/15/5

TABLE III.   SYSTEM PARAMETERS

We present the following evaluation results: 1) the system performance (the energy status of the nodes) with the proposed OMC method; 2) the convergence iteration and the computation time of OMC and the trade-off between the solution quality and the computation time; 3) the algorithm performance (the solution quality and the computation time) comparison of OMC with: i) optimal approaches: decentralized Benders decomposition (DBD) [1], branch and bound method (B&B) [30], and branch and cut method (B&C) [31], and ii) evolutionary approach: genetic algorithm (GA) [32]; and 4) the system performance (the energy consumption of the MCs) using OMC, m-TSP [7] and region partition [16] methods to schedule the MCs and charge the nodes.

Let $g_l(k) = \sqrt{\sum_{s_i \in \mathcal{R}_l^1}(e_i^* - e_i(k))^2 / p_l^1}$ denote the charging error of the lth round at step k, where the charging error is defined as the mean squared error (MSE) between the desired energy levels $\{e_i^*\}$ and the residual energy levels $\{e_i(k)\}$ of the nodes. Fig. 3 shows the charging error in the first cycle $\mathcal{C}^1$, with $n = 25$ and $m = 5$. Based on the refined charging sequence $\tilde{\boldsymbol{L}}^1$, the serving set $\mathcal{S}^1$ contains 22 nodes (i.e., $\tilde{n}^1 = 22$). The numbers of charged nodes p_l^1 and available MCs h_l^1 in each round $\mathcal{R}_l^1$ are listed in Table IV. From it, we observe

$\mathcal{R}_l^1$	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$
p_l^1	5	5	4	4	3
h_l^1	5	5	4	4	4

TABLE IV.   THE REAL NUMBER OF CHARGED NODES IN CYCLE $\mathcal{C}^1$.

that the cycle $\mathcal{C}^1$ contains five rounds $\{\mathcal{R}_1^1, \mathcal{R}_2^1, \mathcal{R}_3^1, \mathcal{R}_4^1, \mathcal{R}_5^1\}$ and the total number of charged nodes in this cycle is less than 22. For some nodes in the charging sequence $\tilde{\boldsymbol{L}}^1$, the lower bounds of the replenished energy $\{\underline{E}_i^1\}$ are negative, and, thus, these nodes can be removed from the charging sequence. Fig. 3 shows that there is no gap between two adjacent rounds, since the residual energy of the nodes decreases gradually without charging. The earlier the MCs start the next charging round, the less energy each node requires to be replenished.

Note that the nodes in the sequence $\tilde{\boldsymbol{L}}^1$ are sorted according to their lifetimes in an increasing order. The desired energy levels of the nodes in the previous rounds are higher than the desired energy levels of the nodes in the later rounds. Moreover, at the beginning of charging process, the charging errors of previous rounds are larger than the charging errors of later rounds. Fig. 4 shows the MSE between the minimum optional energy $e_{\min}$ and the residual energy levels $\{e_i(k)\}$ of the nodes (i.e., $g_e(k) = \sqrt{\sum_{i=1}^n (e_i(k) - e_{\min})^2/n}$) in five cycles $\{\mathcal{C}^1, \mathcal{C}^2, \mathcal{C}^3, \mathcal{C}^4, \mathcal{C}^5\}$, with $n = 25$ and $m = 5$. Due to the introduction of the constraints (10), (16) – (18), the nodes will never run out their residual energies before being charged again.
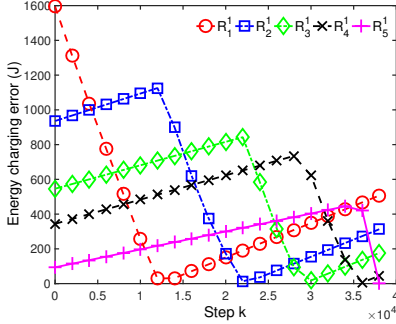
Fig. 3.   The charging error of the nodes in the first cycle $\mathcal{C}^1$.
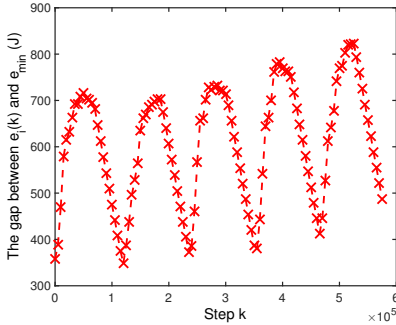
Fig. 4.   The energy status of the nodes in five cycles.

We define the OMC convergence iteration as the number of iterations required by the OMC to converge. The convergence iteration of OMC under different n (the number of nodes) and m (the number of MCs) parameters in one round is compared in Fig. 5. Since the number of charged nodes in one round is no more than m, we set $n = m$ and change the value of n from 5 to 25 with a step of 5. With the value of n increasing, more variables and constraints are involved into the PP, and, thus, more iterations are required to find the optimal solution. Fig. 6 compares the computation time of OMC under different n and m parameters in one cycle ($\mathcal{C}^1$), where the OMC computation time is defined as the time required to solve PP. Fig. 6 shows that under the given number of MCs (m), the computation time of OMC is almost linearly increased with the number of nodes (n). This is because in each round the number of charged nodes, as well as the number of available MCs, is no more than m, while the number of rounds will increase with n in one cycle. For the rest of the rounds, we obtain the similar results.

Fig. 7 and Fig. 8 evaluate the algorithm performance (the solution quality and the computation time) of OMC with the stopping criteria ε varying. We set $n = 50$ and $m = 15$ and change the value of ε between the range of $[0.1, 1, 10, 100]$.
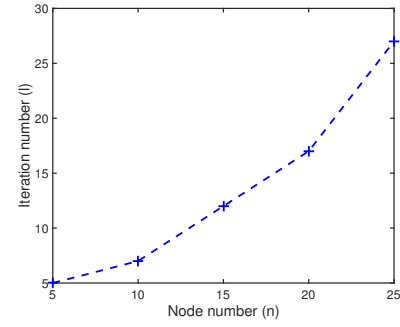

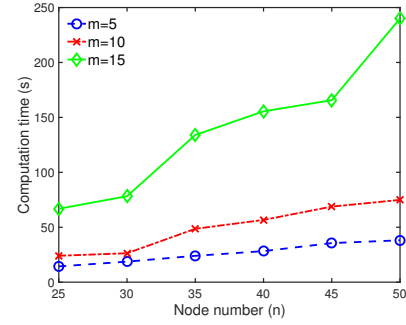Fig. 5.   Convergence iteration of OMC with n varying ($n = m$).

Fig. 6.   Computation time of OMC and with m and n varying.

Under the given ε, the iterations between the MP and the SP stop when the gap between the bounds $\Phi_U(l)$ and $\Phi_L(l)$ is smaller than ε and the DSP is feasible under the solution to the MP (i.e., $\boldsymbol{x}(l)$). Accordingly, by adjusting the value of ε, we can control the quality of the solution. From Fig. 8 and Fig. 7, we observe that the solution quality and the computation time decrease, when ε increases. This characteristic inspires us a feasible PP solution can be found by using the following way: we stop the iterations between the MP and the SP when the SP is feasible for the first time (assume that at the lth iteration). Therefore, the corresponding MC moving time and charging time decision $\boldsymbol{y}(l)$ is obtained under the given MC scheduling decision $\boldsymbol{x}(l)$.
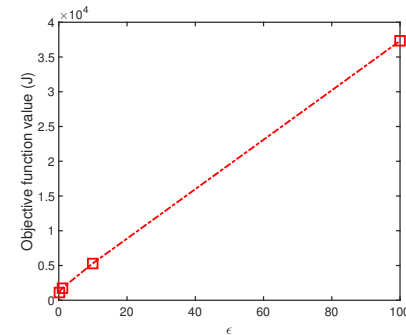

Fig. 7.   Solution quality of OMC with ε varying.

The solutions to the PP found by the OMC, DBD, B&B, B&C and GA are compared in Fig. 9. Let $E_o(m, n)$, $E_d(m, n)$, $E_b(m, n)$, $E_c(m, n)$ and $E_g(m, n)$ denote the objective function values of the PP (the total energy consumption of the MCs) achieved by the OMC, DBD, B&B, B&C and GA under the given m and n parameters, respectively. The box plot of "DBD vs OMC" shows the statistical property of the data set $\{(E_d(m, n) - E_o(m, n))/E_d(m, n)\}$ for all tuned m and n parameters. For the rest of algorithm comparisons, the
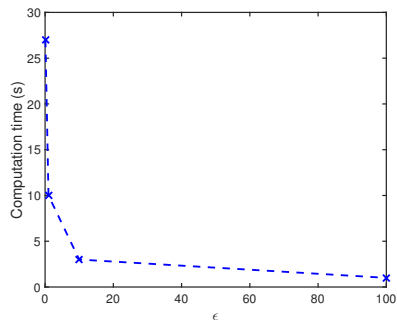
Fig. 8. Computation time of OMC with ε varying.

definition of the data sets is similar to "DBD vs OMC". On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points that are not considered outliers, while the outliers are plotted individually using the '+' symbol. Fig. 9 shows that the solutions given by the OMC, DBD, B&B and B&C are the same. Therefore, the OMC also finds the optimal solution, since B&B and B&C are able to find the optimal solution for the MILP problem. Note that PP is a minimization problem and there is no guarantee of convergence to a global optimum for GA. The solution provided by the OMC has a lower objective function value than GA. The convergence of GA is sensitive to the choice of the genetic operators, the mutation probability and the selection criteria, while fine-tuning of these parameters is often required.
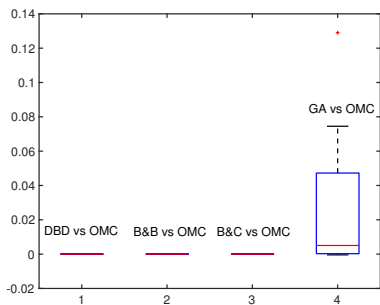

Fig. 9. Energy gain of OMC, DBD, B&B, B&C and GA.

Fig. 10 compares the computation time of the OMC, DBD, B&B, B&C and GA under different n and m parameters. Let $T_o(m,n)$, $T_d(m,n)$, $T_b(m,n)$, $T_c(m,n)$ and $T_g(m,n)$ denote the computation time of OMC, DBD, B&B, B&C and GA required to solve the PP under the given m and n parameters, respectively. The box plot of "DBD vs OMC" shows the statistical property of the data set $\{(T_d(m,n)-T_o(m,n))/T_d(m,n)\}$ for all tuned m and n parameters. Similarly, we construct the data sets for the rest of algorithm comparisons. With the values of n and m increasing, the computation time of all compared algorithms grows. However, OMC has a shorter computation time than DBD, B&B, B&C and GA. Although the SP of DBD is solved in a distributed manner, the reduction of computation time is limited, since the SP is a LP problem. Moreover, the MP of DBD is still an ILP problem, which is more difficult to solve than the LP-based MP of OMC. B&C, which combines the benefits of B&B and Gomory cutting scheme, can better balance optimality, efficiency and stability. Usually, B&C has a faster convergence speed than B&B. Note that the compu-

tational complexity of an optimization problem highly relates to the number of variables and constraints. Solving the smaller problems with less variables and constraints (i.e., MP and SP) iteratively is more efficient than solving a single large problem. This result is in line with the comparison in [33]. Compared with OMC, the structure of GA is more complex, since in each iteration GA needs to generate new populations through several procedures, such as selection, reproduction, mutation and crossover. Therefore, GA has a longer computation time.
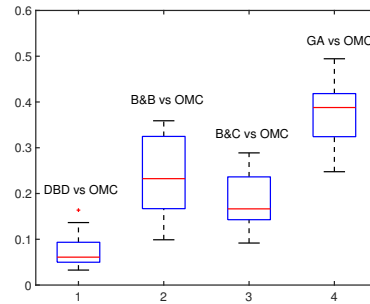

Fig. 10. Time gain of OMC, DBD, B&B, B&C and GA.

The basic idea of the m-TSP method is the use of a graph to model the charging process, where the vertexes and the edges represent the sensor nodes and the charging costs (i.e., the moving and charging time of the MCs), respectively. The aim of the m-TSP method is to minimize the charging costs under the constraint of visiting all the vertexes. On the other hand, the region partition method divides the sensors into several clusters and each cluster is assigned one MC, based on the energy status of the sensors and the MCs. The MC with higher energy level is assigned to the cluster that contains the sensors requiring to be replenished with higher amount of energy. For all methods, a MC will go back to the BS for battery charging/replacement when it has no enough energy to finish the assigned tasks, event if it is in the middle of charging phase. Fig. 11 compares the total energy consumption of the MCs using the OMC, m-TSP and region partition methods for one cycle. We set $m = 5$ and change the value of n from 25 to 50 with a step of 5. In contrast to the m-TSP and the region partition methods, the moving and the charging energy of the MCs are jointly optimized by the OMC method. This implies that there is no need to fully charge all the sensors. The replenished energy of a sensor is as much as it is required so as to make sure that this node has the enough energy to work until being charged again. As shown in Fig. 11, the OMC achieves lower energy consumption than m-TSP and region partition. Achieving a lower MC energy consumption, the times of the MCs back to the BS during the charging process can be reduced. In this way, the charging efficiency is further enhanced and the constraints with respect to the network perpetual operation can be more easily satisfied.

V. CONCLUSION

In this paper, we studied the problem of coordinating the multiple MCs to charge the sensor nodes, with the aim of enhancing the energy efficiency of MCs, while keeping the network operating perpetually. This problem has been formulated as an MILP problem, which jointly optimized the scheduling, the moving time and the charging time of the MCs. To find an optimal solution with reduced computation time,
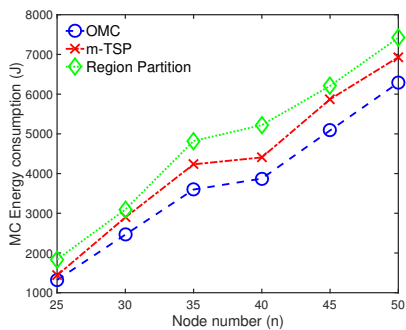
Fig. 11. MC energy consumption with OMC, m-TSP and region partition.

we proposed an OMC algorithm. It decomposes the multiple MCs coordination problem into two correlated subproblems: an MP for the MC scheduling and a SP for the MC moving time and charging time, and solves the subproblems iteratively. We proved that the proposed OMC algorithm converges to the optimal solution through a finite number of iterations. We also proved that the optimality of the solution is still guaranteed when the optimal solution to the MP is replaced with an arbitrary solution during the iteration between the MP and the SP. The results show that the desired system requirements are satisfied using the proposed OMC algorithm. We can also achieve a trade-off between the solution quality and the computation time by adjusting the stopping criteria of the OMC algorithm.

REFERENCES

[1] L. Mo, P. You, X. Cao, Y. Song, and J. Chen, "Decentralized multi-charger coordination for wireless rechargeable sensor networks," in *Proc. IEEE International Performance Computing and Communications Conference*, 2015, pp. 1–8.

[2] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, no. 5834, pp. 83–86, 2007.

[3] O. B. Akan, O. Cetinkaya, C. Koca, and M. Ozger, "Internet of hybrid energy harvesting things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 736–746, 2018.

[4] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting internet of things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1389–1399, 2018.

[5] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless charging technologies: fundamentals, standards, and network applications," *IEEE Commun. Surv. Tut.*, vol. 18, no. 2, pp. 1413–1452, 2016.

[6] F. Sangare, Y. Xiao, D. Niyato, and Z. Han, "Mobile charging in wireless-powered sensor networks: optimal scheduling and experimental implementation," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7400–7410, 2017.

[7] C. Wang, J. Li, F. Ye, and Y. Yang, "NETWRAP: an NDN based real-timewireless recharging framework for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1283–1297, 2014.

[8] T. Rault, "Avoiding radiation of on-demand multi-node energy charging with multiple mobile chargers," *Comput. Commun.*, vol. 134, pp. 42 – 51, 2019.

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

[10] R. Beigel, J. Wu, and H. Zheng, "On optimal scheduling of multiple mobile chargers in wireless sensor networks," in *Proc. International Workshop on Mobile Sensing, Computing and Communication*, 2014, pp. 1–6.

[11] L. Jiang, X. Wu, G. Chen, and Y. Li, "Effective on-demand mobile charger scheduling for maximizing coverage in wireless rechargeable sensor networks," *Mobile Netw. Appl.*, vol. 19, no. 4, pp. 543–551, 2014.

[12] G. Jiang, S. Lam, Y. Sun, L. Tu, and J. Wu, "Joint charging tour planning and depot positioning for wireless sensor networks using mobile chargers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2250–2266, 2017.

[13] C. Wang, J. Li, F. Ye, and Y. Yang, "Recharging schedules for wireless sensor networks with vehicle movement costs and capacity constraints," in *Proc. IEEE International Conference on Sensing, Communication, and Networking*, 2014, pp. 468–476.

[14] H. Dai, X. Wu, G. Chen, L. Xu, and S. Lin, "Minimizing the number of mobile chargers for large-scale wireless rechargeable sensor networks," *Comput. Commun.*, vol. 46, pp. 54 – 65, 2014.

[15] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining sensor networks perpetually via wireless recharging mobile vehicles," in *Proc. IEEE Conference on Local Computer Networks*, 2014, pp. 270–278.

[16] A. Madhja, S. Nikoletseas, and T. P. Raptis, "Distributed wireless power transfer in sensor networks with multiple mobile chargers," *Comput. Netw.*, vol. 80, pp. 89 – 108, 2015.

[17] S. Zhang, J. Wu, and S. Lu, "Collaborative mobile charging," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 654–667, 2015.

[18] C. Lin, Y. Wu, Z. Liu, M. S. Obaidat, C. W. Yu, and G. Wu, "GTCharge: a game theoretical collaborative charging scheme for wireless rechargeable sensor networks," *J. Syst. Softw.*, vol. 121, pp. 88–104, 2016.

[19] A. Madhja, S. Nikoletseas, and T. P. Raptis, "Hierarchical, collaborative wireless energy transfer in sensor networks with multiple mobile chargers," *Comput. Netw.*, vol. 97, pp. 98–112, 2016.

[20] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math.*, vol. 4, no. 1, pp. 238–252, 1962.

[21] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. D. Sherali, and S. F. Midkiff, "Multi-node wireless energy charging in sensor networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 437–450, 2015.

[22] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: a survey," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 2, pp. 551–591, 2013.

[23] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, 2013.

[24] Y. Shu, H. Yousefi, P. Cheng, J. Chen, Y. J. Gu, T. He, and K. G. Shin, "Near-optimal velocity control for mobile charging in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1699–1713, 2016.

[25] W. Na, J. Park, C. Lee, K. Park, J. Kim, and S. Cho, "Energy-efficient mobile charging for wireless power transfer in internet of things networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 79–92, 2018.

[26] Y. Peng, Z. Li, W. Zhang, and D. Qiao, "Prolonging sensor network lifetime through wireless charging," in *Proc. IEEE Real-Time Systems Symposium*, 2010, pp. 129–139.

[27] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University, 2004.

[28] L. Mo, X. Cao, Y. Song, and A. Kritikakou, "Distributed node coordination for real-time energy-constrained control in wireless sensor and actuator networks," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4151–4163, 2018.

[29] M. Fischetti, F. Glover, and A. Lodi, "The feasibility pump," *Mathematical Programming*, vol. 104, no. 1, pp. 91–104, 2005.

[30] S. Boyd and J. Mattingley, "Branch and bound methods," *Notes for EE364b, Stanford University*, pp. 1–11, 2007.

[31] S. Albert, "Solving mixed integer linear programs using branch and cut algorithm," Master's thesis, North Carolina State University, 1999.

[32] E. Rothberg, "An evolutionary algorithm for polishing mixed integer programming solutions," *INFORMS J. Comput.*, vol. 19, no. 4, pp. 534–541, 2007.

[33] C. Randazzo and H. P. L. Luna, "A comparison of optimal methods for local access uncapacitated network design," *Ann. Oper. Res.*, vol. 106, no. 1, pp. 263–286, 2001.

Lei Mo (S'13–M'17) is currently a Postdoctoral Fellow with INRIA Rennes research center, France. He received the B.S. degree from College of Telecom Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou, China, in 2007, and the Ph.D. degree from College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, in 2013. From 2013 to 2015, he was a research fellow with the Department of Control Science and Engineering, Zhejiang University, China. From 2015 to 2017, he was a research fellow with INRIA Nancy research center, France. His current research interests include networked estimation and control in wireless sensor and actuator networks, cyber-physical systems, task mapping and resources allocation in embedded systems. He serves as an Associate Editor for KSII Transactions on Internet and Information Systems, Journal of Computer and Journal of Electrical and Electronic Engineering. He also serves as a Guest Editor for IEEE Access and Journal of Computer Networks and Communications and a TPC Member for several international conferences.

Angeliki Kritikakou is currently an Associate Professor at University of Rennes 1 and IRISA - INRIA Rennes research center. She received her Ph.D. in 2013 from the Department of Electrical and Computer Engineering at University of Patras, Greece and in collaboration with IMEC Research Center, Belgium. She worked for one year as a Postdoctoral Research Fellow at the Department of Modelling and Information Processing (DTIM) at ONERA in collaboration with Laboratory of Analysis and Architecture of Systems (LAAS) and the University of Toulouse, France. Her research interests include embedded systems, real-time systems, mixed-critical systems, hardware/software co-design, mapping methodologies, design space exploration methodologies, memory management methodologies, low power design and fault tolerance.
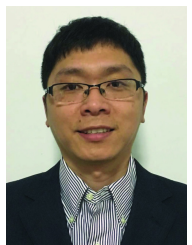
Shibo He (M'13) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2012. From Nov. 2010 to Nov. 2011, he was a visiting scholar with the University of Waterloo, Waterloo, ON, Canada. He was an Associate Research Scientist from March 2014 to May 2014, and a postdoctoral scholar from May 2012 to February 2014, with Arizona State University, Tempe, AZ, USA. He is currently a Professor at Zhejiang University. His research interests include wireless sensor networks, crowd sensing and big data analysis. Dr. He serves on the editorial board of IEEE Transactions on Vehicular Technology, Springer Peer-to-Peer Networking and Application, KSII transactions Internet and Information Systems, and is a guest editor of Elsevier Computer Communications and Hindawi International Journal of Distributed Sensor Networks. Dr. He is the recipient of IEEE Asia-Pacific outstanding researcher award, 2015.