Charging Task Scheduling for Directional Wireless Charger Networks

Haipeng Dai*, Ke Sun*, Alex X. Liu*†, Lijun Zhang*, Jiaqi Zheng*, and Guihai Chen*
*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210024, CHINA
†Department of Computer Science and Engineering, Michigan State University, East Lansing 48823, MI, USA
{haipengdai,zlj,gchen}@nju.edu.cn,kesun@smail.nju.edu.cn,alexliu@cse.msu.edu,jiaqi369@gmail.com

ABSTRACT

This paper studies the problem of cHarging tAsk Scheduling for direcTional wireless chargEr networks (HASTE), *i.e.*, given a set of rotatable directional wireless chargers on a 2D area and a series of offline (online) charging tasks, scheduling the orientations of all the chargers with time in a centralized offline (distributed online) fashion to maximize the overall charging utility for all the tasks. We prove that HASTE is NP-hard. Then, we prove that a relaxed version of HASTE falls within the realm of maximizing a submodular function subject to a partition matroid constraint, and propose a centralized offline algorithm that achieves $(1-\rho)(1-\frac{1}{e})$ approximation ratio to address HASTE where ρ is the switching delay of chargers. Further, we propose a distributed online algorithm and prove it achieves $\frac{1}{2}(1-\rho)(1-\frac{1}{e})$ competitive ratio. We conduct simulations, and field experiments on a testbed consisting of 8 off-the-shelf power transmitters and 8 rechargeable sensor nodes. The results show that our distributed online algorithm achieves 92.97% of the optimal charging utility, and outperforms the comparison algorithms by up to 26.19% in terms of charging utility.

KEYWORDS

Charging task, Scheduling, Directional wireless chargers

1 INTRODUCTION

The last decade has witnessed the rapid development of Wireless Power Transfer (WPT) technology, which enjoys huge advantages such as no contact, reliable power supply, and ease of maintenance compared to traditional wired power supply technologies. WPT technology has numerous applications, including wireless identification and sensing platform (WISP) [1], wireless rechargeable sensor networks [2, 3], electric vehicles [4], wireless powered drone aircraft [5], *etc.*. As per the record provided by Wireless Power Consortium, the number of registered WPT products from its 214 member companies, including IT leaders Samsung, Philips, and Huawei, has surged to 848 [6]. By a recent report, 35% of consumers in the United States have used WPT products [7].

In this paper, we consider the problem of cHarging tAsk Scheduling for direcTional wireless chargEr networks (HASTE) aiming for maximizing the overall charging utility of offline/online charging tasks. We adopt the directional charging model for wireless chargers and rechargeable devices for which the power charging area for a charger and the power receiving area for a device are modeled as sectors [8, 9]. A rechargeable device can be charged via wireless by a charger with non-zero power if and only if they are located in each other's covered sector. All wireless chargers can freely adjust its orientation in $[0, 2\pi)$. Moreover, a charging task initiated by a rechargeable device consists of five elements: the position and orientation of its associated device, the release time and end time of the task, and its required charging energy. We define the task's charging utility as a linear and bounded function with its harvested energy from its release time to its end time. With these models, we consider offline/online charging task scheduling. In the offline scenario, information for all charging tasks is known a priori, and thereby the scheduling policies for all chargers at any moment can be determined beforehand. To accommodate practical concerns, we assume that each charger needs an amount of time for switching, which we call *switching delay.* In the online scenario, charging tasks stochastically arrive, and chargers reschedule their orientations in realtime. Nevertheless, in addition to *switching delay*, each charger needs an additional amount of time for recomputing the scheduling policies with negotiating with neighboring chargers, which we call *rescheduling delay.* To avoid global management effort and reduce update cost, we desire a distributed and local algorithm which is scalable with network size. To sum up, we state our problem HASTE as follows. Given a set of rotatable directional wireless chargers on a 2D area and a series of offline (online) charging tasks, scheduling the orientations of all the chargers with time in a centralized offline (distributed online) fashion to maximize the overall charging utility for all the tasks.

First, there exist numerous literatures [10–17] studying on the mobile charging case where one single or multiple chargers travel in a field to charge wireless rechargeable devices to guarantee their normal working, which are fundamentally different from ours. Second, the other works consider wireless charger networks consisted of static wireless chargers such as [18–22], but none of them investigate charging task scheduling for directional wireless charger networks.

We are faced with three major challenges. The first challenge is that HASTE is non-linear and is NP-hard. HASTE is nonlinear because that the orientation of chargers can be freely scheduled; a task can be either covered by a charger and have a certain constant power increment or not with no power increment, which has the flavor of 0-1 integer programming; the charging utility function is linear but bounded, let alone that we extend our results to the case where the utility function is a general concave function. In addition, by reducing from the classical NP-hard separate assignment problem, we prove that HASTE is NP-hard. The second challenge is how to design an efficient centralized offline algorithm for

HASTE in the offline scenario while considering the switching delay of chargers. The switching delay happens if and only if a charger's next intended orientation is different from its current orientation, which implies that the switching delay as well as its caused performance loss is history-dependent. Moreover, the performance loss is difficult to evaluate as there are potentially multiple tasks that are affected by a charger's switching delay, and the charging utility function for tasks is non-linear. The third challenge is how to design an efficient distributed online algorithm for HASTE in the online scenario where all chargers are asynchronous and the rescheduling delay needs to be considered. To the best of our knowledge, there are neither existing distributed online algorithms directly applicable to our problem even when the rescheduling delay is omitted, nor existing online algorithms that deal with the case in our considered scenario with rescheduling delay being concerned for which the response is delayed and the algorithm is not truly "online".

To address the first challenge, we propose that rather than considering all possible orientations in $[0, 2\pi)$ for chargers, we can safely consider a limited number of orientations for them without causing performance loss, and therefore, extract the so-called "dominant task sets" as the corresponding sets of covered tasks. Then, we neglect the switching delay for wireless chargers, and thus reformulate the original continuous optimization problem into a discrete optimization problem HASTE-R. Further, we prove that the reformulated problem is exactly a problem of maximizing a submodular function subject to a partition matroid constraint. To address the second challenge, based on the theoretical results obtained by addressing the first challenge, we tailor the TABULARGREEDY algorithm proposed in [23] to address HASTE-R as it can achieve an approximation ratio between $\frac{1}{2}$ and $1-\frac{1}{e}$ ($1-\frac{1}{e}$ as default in our setting) depending on the value of a control parameter and resulting in different time complexity. Further, to bound the performance loss of switching delay, we exploit the concavity of the utility function and consider all the caused performance loss for all impacted tasks in the worst case, and prove that the switching delay introduces a constant factor of $1-\rho$ in the ultimate achieved approximation ratio for HASTE, *i.e.*, $(1-\rho)(1-\frac{1}{e})$, of the proposed algorithm, where ρ is the switching delay. To address the third challenge, we propose a distributed online algorithm to HASTE. We first prove that if the rescheduling delay is neglected, its achieved global charging utility is the same as that of the centralized offline algorithm. Further, by leveraging the concavity of the utility function and the submodularity of the objective function, we bound the performance loss of scheduling delay, and prove that our distributed online algorithm achieves $\frac{1}{2}(1-\rho)(1-\frac{1}{e})$ competitive ratio.

We conducted simulations and field experiments to evaluate our proposed algorithms. Our simulation results show that our proposed distributed online algorithm can achieve 92.97% of the optimal charging utility, outperform the other two comparison algorithms by 10.96%. Our experimental results show that our distributed online algorithm outperforms the comparison algorithms by up to 26.19% on average.

2 RELATED WORK

First, there exist some literatures focus on mobile charging scenarios where one single or multiple chargers travel in a field to charge rechargeable devices deployed there to make them work perpetually, which are fundamentally different from ours. For example, [10, 11] study the charging efficiency issues of wireless chargers. [12, 13] concentrate on reducing the service delay of mobile chargers. [14–16] optimize the overall network performance such as data routing, data collection, and task assignment. We refer readers to the survey [17] for more related works.

Second, the other works (*e.g.*, [8, 9, 24]) are dedicated to wireless charger networks consisted of static wireless chargers, but none of them consider charging task scheduling for directional wireless charger networks. On one hand, some of them study wireless charging issues but overlook the detrimental effect of the electromagnetic radiation (EMR) to human health, *e.g.*, Dai *et al.* first proposed the empirical directional charging model, and investigated the problem of ominidirectional charging with directional chargers in [8] and the directional wireless charger placement problem in [9]. The others [18–22] take the EMR safety into consideration by guaranteeing that the EMR intensity at any point in the area does not exceed a predefined EMR threshold, *e.g.*, Dai *et al.* presented and studied how to schedule non-adjustable chargers [18] and adjustable chargers [19] to maximize the charging utility for chargers under the EMR safety constraint.

3 PROBLEM FORMULATION

3.1 Preliminaries

Suppose there is a set of directional wireless chargers $S = \{s_1, ..., s_n\}$ located in a 2D plane Ω, which can continuously rotate with orientation angle within $[0\ 2\pi)$. Suppose there are also some rechargeable devices located in Ω, which either keep static or dynamically join or leave the wireless charger network. These rechargeable devices launch (wireless) charging tasks and sending them to chargers now and then, and the chargers accordingly schedule their orientations to serve the tasks. Formally, charging tasks are defined by a five-tuple $\mathcal{T}_j =< o_j, \phi_j, t_r^j, t_e^j, E_j >$ where o_j denotes the position of the rechargeable device that raises the task, ϕ_j is the orientation of the device, t_r^j and t_e^j are the release time and end time of the task, and E_j is required charging energy. We adopt a discrete time model for which the time is divided into multiple slots with uniform duration T_s. For simplicity, we assume that t_r^j is exactly at the beginning of a time slot while t_e^j is at the end of a time slot. We will show in the discussion to Lemma 4.2 that even t_r^j and t_e^j are not aligned with time slots. We summarize the notations used in this paper in Table 1.

We adopt the general and practical directional charging model proposed in [8, 9]. As Figure 1 shows, a charger s_i with working orientation denoted by vector $\overrightarrow{r_{\theta_i}}$ can only charge devices in a *charging area* in the shape of a sector with *charging angle* A_s and radius D. A rechargeable device o_j with orientation denoted by vector $\overrightarrow{r_{\phi_j}}$ can only receive non-zero power in a *receiving area* in the shape of a sector with *receiving angle* A_o and radius D. The charging power from s_i to o_j is given by

$$P_r(s_i,\theta_i,o_j,\phi_j)=\begin{cases}\frac{\alpha}{(||s_io_j||+\beta)^2}, & 0\le ||s_io_j||\le D,\\ & \overrightarrow{s_io_j}\cdot\overrightarrow{r_{\theta_i}}-\|s_io_j\|cos(A_s/2)\ge 0,\\ and & \overrightarrow{o_js_i}\cdot\overrightarrow{r_{\phi_j}}-\|o_js_i\|cos(A_o/2)\ge 0.\\ 0, & otherwise\end{cases}$$

where α and β are two known constants, and $||s_io_j||$ is the distance between s_i and o_j. Further, if a device o_j is covered by more than one chargers, its received power is the sum of the received power from all chargers [8, 9].

A charger can either keep its orientation unchanged during the same time slot, or switch its orientation in the starting ρ $(0<\rho<1)$ portion of a time slot, which we call *switching delay*, and keep static in the rest $1-\rho$ portion of the time slot. We argue that this assumption makes sense because typically a charging task can last up to tens of minutes or even more than an hour, the duration of time slots can be set to a few minutes, and the switching time for commercial rotatable heads or cradles [25] on which the chargers are mounted or soft switching of smart antennas of chargers [26, 27] is commonly a few seconds or even shorter. We assume that a charger stops emitting power during its switching. For convenience of exposition, we define $\theta_i=\emptyset$ for a charger during its switching process, and further define $P_r(s_i,\emptyset,o_j,\phi_j)=0$. In the offline case, we assume the information for all charging tasks are known a prior, then the scheduling policies for all time slots for each charger are determined beforehand. In the online case, we assume the charging tasks stochastically arrive, and chargers recompute their scheduling policies in an on-the-fly fashion. Especially, we assume each charger needs τ $(\tau\in\mathbb{Z}_+)$ number of time slots, which we name as *rescheduling delay*, for negotiation with neighboring chargers and computation to update its future scheduling policies, and then, if necessary, starts switching with a delay of ρ time slot. Typically, the rescheduling delay much less than the duration of charging tasks. In this paper, we assume the latter is at least two times that of the former, *i.e.*, $t_e^j-t_r^j\ge 2\tau T_s$ where T_s is the duration of a time slot.

We adopt a linear and bounded charging utility model for harvested energy for a task that is similar to [9]. That is, the charging utility for a task is first proportional to the harvested energy of its associated device, and then reaches a constant if the harvested energy exceeds a predetermined threshold, *i.e.*,

$$\mathcal{U}(x)=\begin{cases}\frac{1}{E_j}\cdot x, & x\le E_j\\ 1, & x>E_j\end{cases}\tag{1}$$

where E_j is the required charging energy of charging task $\mathcal{T}_j$.

3.2 Problem Formulation and Hardness Analysis

Let $\theta_i(t)$ $(\theta_i:\mathbb{R}_{\ge 0}\mapsto\{[0\ 2\pi)\cup\emptyset\})$ be the function of orientation for charger s_i with time t. Suppose the value of $\theta_i(t)$ at the k_{th} time slot is $\theta_{i,k}$ if charger s_i is not switching; otherwise, $\theta_i(t)$ is set to $\emptyset$ and the charging power of s_i is zero. Then, for a charging task $\mathcal{T}_j$, its harvested power at time t is given by $\sum_{i=1}^n P_r(s_i,\theta_i(t),o_j,\phi_j)$, and its aggregate harvested energy during its whole life is $\int_{t_r^j}^{t_e^j}\sum_{i=1}^n P_r(s_i,\theta_i(t),o_j,\phi_j)dt$. And the overall (weighted) charging utility is $\sum_{j=1}^m w_j\cdot\mathcal{U}(\int_{t_r^j}^{t_e^j}\sum_{i=1}^n P_r(s_i,\theta_i(t),o_j,\phi_j)dt)$ where w_j is the weight of charging task $\mathcal{T}_j$. Our task is to determine the decision variables $\theta_{i,k}$ defined in $\theta_i(t)$ for all the chargers so that the overall charging utility is maximized.

Table 1: Notations and symbols used in this paper

Symbol	Description
s_i	The i_{th} directional wireless charger, or its position
n	Number of directional wireless chargers
θ_i $(\theta_i(t))$	Orientation of charger s_i (its function with time t)
$\theta_{i,k}$	The value of $\theta_i(t)$ at the k_{th} time slot if charger s_i is not switching
$\mathcal{T}_j$	The j_{th} charging task
o_j	Position of the rechargeable device that raises charging task $\mathcal{T}_j$, or the j_{th} rechargeable device
ϕ_j	Orientation of the rechargeable device that raises charging task $\mathcal{T}_j$, or the orientation of device o_j
t_r^j (t_e^j)	Release time (end time) of charging task $\mathcal{T}_j$
E_j	Required charging energy of charging task $\mathcal{T}_j$
m	Number of charging tasks
A_s	Charging angle of chargers
A_o	Receiving angle of devices
T_s	Duration of a time slot
$P_r(.)$	Charging power function
α,β	Constants in the charging model
D	Radius of charging/receiving area
ρ	Switching delay
τ	Rescheduling delay
$\mathcal{U}(.)$	Charging utility function
w_j	Weight of charging task $\mathcal{T}_j$
$\boldsymbol{\mathcal{T}}_i$	Set of charging tasks that cover charger s_i
$\varGamma_i$ $(\varGamma_i^p)$	Set of dominant task sets for charger s_i (the p_{th} dominant task set in $\varGamma_i$)
$\varGamma_{i,k}$ $(\varGamma_{i,k}^p)$	Set of dominant task sets for charger s_i at the k_{th} time slot (the p_{th} dominant task set in $\varGamma_{i,k}$)
K	Number of considered time slots for all tasks
C	Number of colors
$N(s_i)$	Neighbors of charger s_i (two chargers are neighbors to each other if and only if they cover at least one charging task in common)
K_i	Number of considered time slots for all tasks observed by charger s_i

With all above, we define the cHarging tAsk Scheduling for direcTional wireless chargEr networks (HASTE) as follows.

$$\textbf{(P1)}\quad \max_{\theta_{i,k}}\quad \overline{\mathcal{U}}=\sum_{j=1}^m w_j\cdot\mathcal{U}(\int_{t_r^j}^{t_e^j}\sum_{i=1}^n P_r(s_i,\theta_i(t),o_j,\phi_j)dt)$$

$$s.t.\quad \theta_i(t)=\begin{cases}\begin{cases}\emptyset,\ kT_s<t\le(k+\rho)T_s\\ \theta_{i,k},(k+\rho)T_s<t\le(k+1)T_s\end{cases} & ,\theta_{i,k}\ne\theta_{i,k-1}\\ \theta_{i,k},kT_s<t(k+1)T_s, & otherwise\end{cases}$$

$$where\ k\in\mathbf{Z}_0^+,\ and\ \theta_i(0)=\emptyset$$

$$0\le\theta_{i,k}<2\pi.$$

The following theorem shows the complexity of HASTE.

Theorem 3.1. *HASTE is NP-hard.*

Proof. Basically, we can prove the NP-hardness of HASTE by reducing from the NP-hard separate assignment problem [28]. We omit details to save space. □

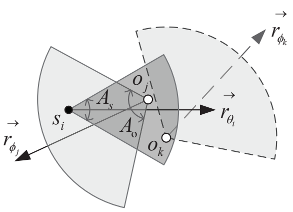

Figure 1: Directional charging model (o_j can receive power from s_i while o_k cannot)

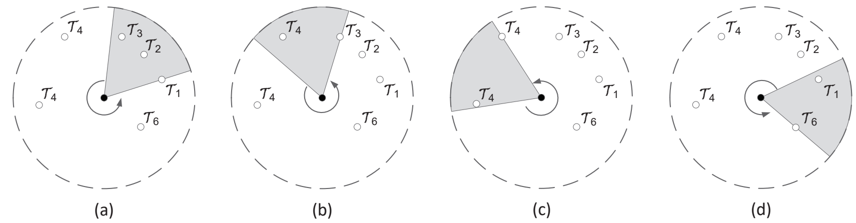

Figure 2: A toy example of dominant task sets extraction

4 PROBLEM REFORMULATION

In this section, we first propose a dominant task sets extraction algorithm for chargers to reduce the continuous solution space for HASTE to a discrete one with limited choices. Then, we consider a relaxed version of HASTE, *i.e.*, HASTE-R, and prove it falls into the realm of maximizing a submodular function subject to a partition matroid constraint, which assists the further algorithm design.

4.1 Extraction of Dominant Task Sets

Though each charger can continuously rotate within $[0\ 2\pi)$, we do NOT need to consider all possible orientations. Instead, we only need to consider the following specific ones.

Definition 4.1. (**dominant task set**) Given a set of tasks $\boldsymbol{\mathcal{T}}_i^1$ covered by a charger s_i with some orientation, if there doesn't exist another set of tasks $\boldsymbol{\mathcal{T}}_i^2$ covered by s_i with some other orientation such that $\boldsymbol{\mathcal{T}}_i^1 \subset \boldsymbol{\mathcal{T}}_i^2$, then $\boldsymbol{\mathcal{T}}_i^1$ is a *dominant task set*.

We describe our algorithm for extracting dominant task sets in Algorithm 1. Basically, the considered charger rotates for 2π and extracts the dominant task sets one by one. We use a toy example for illustration. As shown in Figure 2(a), the charger first covers task $\mathcal{T}_1$, then rotates to cover tasks $\mathcal{T}_2$ and $\mathcal{T}_3$ sequentially. Further, $\mathcal{T}_4$ cannot be added in the current covered set as otherwise $\{\mathcal{T}_1, \mathcal{T}_2\}$ will be missed, and therefore, $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ is a dominant task set. Then, the charger continues to cover $\mathcal{T}_4$ by removing $\mathcal{T}_1$ and $\mathcal{T}_2$ from the current set, as shown in Figure 2(b). Similarly, as $\mathcal{T}_5$ cannot be covered by the charger without missing $\mathcal{T}_3$, $\{\mathcal{T}_3, \mathcal{T}_4\}$ is added as a dominant task set. Algorithm 1 proceeds until the charger rotates for 2π, as depicted in Figure 2(c) and (d). After all, the obtained dominant task sets are $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}, \{\mathcal{T}_3, \mathcal{T}_4\}, \{\mathcal{T}_4, \mathcal{T}_5\}$ and $\{\mathcal{T}_6, \mathcal{T}_1\}$.

4.2 Problem Relaxation and Reformulation

As the switching delay is hard to be analyzed for optimization, we first consider a relaxed version of HASTE, HASTE-R, by neglecting the switching delay of all chargers, and then analyze HASTE. Suppose the obtained set of dominant task sets for charger s_i is Γ_i, the p_{th} dominant task set in Γ_i is Γ_i^p. Let $x_{i,k}^p$ be a binary indicator denoting whether the p_{th} dominant task set in Γ_i in the k_{th} time slot is selected or not. For convenience, we define

---

Algorithm 1: Dominant Task Sets Extraction

Input: The wireless charger s_i, all charging tasks $\{\mathcal{T}_j\}_{j=1}^m$
Output: All dominant task sets

1 Find the subset of charging tasks in $\{\mathcal{T}_j\}_{j=1}^m$ that cover s_i, say $\boldsymbol{\mathcal{T}}_i$;
2 Initialize the orientation of the charger to 0;
3 Rotate the charger anticlockwise to cover the tasks in $\boldsymbol{\mathcal{T}}_i$ one by one until there is some covered task is going to be uncovered. During the rotating process, if the rotated angle is larger than 2π, then terminate;
4 Add the current covered set of tasks to the collection of dominant task sets;
5 Rotate the charger anticlockwise until a new task in $\boldsymbol{\mathcal{T}}_i$ is included in the covered set. During the rotating process, if the rotated angle is larger than 2π, then terminate. If not, goto Line 3.

$$P_r(s_i, o_j) = \begin{cases} \frac{\alpha}{(||s_i o_j|| + \beta)^2}, & 0 \le ||s_i o_j|| \le D, \\ 0, & otherwise. \end{cases}$$

Then, the problem HASTE-R can be formulated as

$$\textbf{(RP1)} \max_{x_{i,k}^p} \quad \overline{U}_R = \sum_{j=1}^{m} w_j \cdot \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_i^p \ni o_j, \\ i \in [n],\, p \in [|\Gamma_i|]}} x_{i,k}^p P_r(s_i, o_j) T_s \Big)$$

$$s.t. \quad \sum_{p=1}^{|\Gamma_i|} x_{i,k}^p = 1, \ (x_{i,k}^p \in \{0, 1\})$$

where $x_{i,k}^p$s are the decision variables, Γ_i^p is the p_{th} dominant task set in Γ_i. We first give the following definitions.

Definition 4.2. [29] (**submodular set function**) Let S be a finite ground set. A real-valued set function $f : 2^S \to \mathbb{R}$ is normalized, monotonic and submodular if and only if it satisfies the following conditions, respectively: (1) $f(\emptyset) = 0$; (2) $f(A \cup \{e\}) - f(A) \ge 0$ for any $A \subseteq S$ and $e \in S \backslash A$; (3) $f(A \cup \{e\}) - f(A) \ge f(B \cup \{e\}) - f(B)$ for any $A \subseteq B \subseteq S$ and $e \in S \backslash B$.

Definition 4.3. [29] (**matroid**) A *matroid* $\mathcal{M}$ is a strategy $\mathcal{M} = (S, L)$ where S is a finite ground set, $L \subseteq 2^S$ is a collection of independent sets, such that: (1) $\emptyset \in L$; (2) if $X \subseteq Y \in L$, then $X \in L$; (3) if $X, Y \in L$, and $|X| < |Y|$, then $\exists y \in Y \backslash X$, $X \cup \{y\} \in L$.

Definition 4.4. [29] (**partition matroid**) Given $S = \bigcup_{i=1}^k S_i'$ is the disjoint union of k sets, $l_1, l_2, \ldots, l_k$ are positive integers, a *partition matroid* $\mathcal{M} = (S, \mathcal{I})$ is a matroid where $\mathcal{I} = \{X \subset S : |X \cap S_i'| \le l_i \text{ for } i \in [k]\}$.

First, we define $\Gamma_{i,k} = \Gamma_i$ ($k \in [K]$) as the set of dominant task sets for charger s_i at the k_{th} time slot, where K is the total number of time slots and the notation $[n] = \{1, 2, \ldots, n\}$, and define $\Gamma_{i,k}^p$ as the p_{th} dominant task set in $\Gamma_{i,k}$. Then, we define $\Theta_{i,k}^p$ as the corresponding scheduling policy for $\Gamma_{i,k}^p$, *i.e.*, the orientation that covers $\Gamma_{i,k}^p = \Gamma_i^p$, for charger s_i at the k_{th} time slot, define $\boldsymbol{\Theta}_{i,k} = \{\Theta_{i,k}^p\}_{p \in [|\Gamma_{i,k}|]}$ as the set of scheduling policies for s_i at the k_{th} time slot, and define a ground set of all scheduling policies $S = \{\boldsymbol{\Theta}_{i,k}\}_{i \in [n], k \in [K]}$. Further, we define the scheduling policies for all chargers at all K time slots as X, which is subject to $|X \cap \boldsymbol{\Theta}_{i,k}| \leq 1$. As $\boldsymbol{\Theta}_{i,k}$s are disjoint sets, we write the independent sets as

$$\mathcal{I} = \{X \subseteq S : |X \cap \boldsymbol{\Theta}_{i,k}| \leq 1 \text{ for } i \in [n], k \in [K]\}. \quad (2)$$

Lemma 4.1. *The constraint in the problem* ***RP1*** *can be written as a partition matroid on the ground set S.*

Accordingly, problem **RP1** can be rewritten as
(RP2)

$$\max_X f(X) = \sum_{j=1}^{m} w_j \cdot \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j,\, i \in [n], \\ p \in \{p | \Theta_{i,k}^p = X \cap \boldsymbol{\Theta}_{i,k}\}}} P_r(s_i, o_j) T_s\Big)$$

$$s.t. \quad X \in \mathcal{I}$$

For **RP2**, we have the following lemma.

Lemma 4.2. The objective function $f(X)$ in **RP2** is a monotone submodular set function.

Proof. First, when there are no active scheduling policies, *i.e.*, $X = \varnothing$, the received energy for any task is zero, then $f(X) = 0$. Second, let A be a set of scheduling strategies in S and $e \in S \backslash A$. For simplicity, define

$$g(X, j) = \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j,\, i \in [n], \\ p \in \{p | \Theta_{i,k}^p = X \cap \boldsymbol{\Theta}_{i,k}\}}} P_r(s_i, o_j) T_s\Big)$$

as the achieved utility for task $\mathcal{T}_j$. It is easy to see that $g(A \cup \{e\}, j) - g(A, j) \geq 0$ because there are possibly more chargers cover task $\mathcal{T}_j$ as all possible dominant task sets that cover $\mathcal{T}_j$, *i.e.*, $\Gamma_{i,k}^p$ ($i \in [n]$, $p \in \{p | \Theta_{i,k}^p = A \cap \boldsymbol{\Theta}_{i,k}\}$) would be enlarged as A becomes $A \cup \{e\}$, and the utility function $\mathcal{U}(.)$ is non-decreasing. Hence we have

$$f(A \cup \{e\}) - f(A) = \sum_{j=1}^{m} w_j \cdot [g(A \cup \{e\}, j) - g(A, j)] \geq 0.$$

Third, let A and B be two sets such that $A \subseteq B \subseteq S$ and element $e \in S \backslash B$. On one hand, it is easy to see that

$$\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j, \\ i \in [n],\, p \in P_1}} P_r(s_i, o_j) T_s - \sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j, \\ i \in [n],\, p \in P_2}} P_r(s_i, o_j) T_s$$

$$= \sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j, \\ i \in [n],\, p \in P_3}} P_r(s_i, o_j) T_s - \sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j, \\ i \in [n],\, p \in P_4}} P_r(s_i, o_j) T_s$$

where $P_1 = \{p | \Theta_{i,k}^p = \{A \cup e\} \cap \boldsymbol{\Theta}_{i,k}\}$, $P_2 = \{p | \Theta_{i,k}^p = A \cap \boldsymbol{\Theta}_{i,k}\}$, $P_3 = \{p | \Theta_{i,k}^p = \{B \cup e\} \cap \boldsymbol{\Theta}_{i,k}\}$, and $P_4 = \{p | \Theta_{i,k}^p = B \cap \boldsymbol{\Theta}_{i,k}\}$. On the other hand, it is clear that

$$(\mathcal{U}(x_1 + \Delta x) - \mathcal{U}(x_1)) - (\mathcal{U}(x_2 + \Delta x) - \mathcal{U}(x_2)) \geq 0,$$

for any $x_2 \geq x_1 \geq 0$ and $\Delta x \geq 0$ due to the concavity of the charging utility function $\mathcal{U}(.)$. Consequently, we have $[g(A \cup \{e\}, j) - g(A, i, q)] - [g(B \cup \{e\}, j) - g(B, j)] \geq 0$, and therefore,

$$[f(A \cup \{e\}) - f(A)] - [f(B \cup \{e\}) - f(B)]$$
$$= \sum_{j=1}^{m} w_j \cdot \{[g(A \cup \{e\}, j) - g(A, j)] - [g(B \cup \{e\}, j) - g(B, j)]\} \geq 0.$$

In summary, by Definition 4.2 we conclude that $f(X)$ is a monotone submodular set function. □

Discussion: First, if the charging utility function is concave, it is easy to verify that the three properties for the submodular set function still hold. Therefore, all the performance guarantees for our following centralized offline and distributed online algorithms still keep valid. Second, if t_r^j and t_e^j are not aligned with time slots, the received power at each time slot will not be affected and the aggregate charging energy for each task in a specific time slot is proportional to its active time duration in this time slot. Therefore, $f(X)$ is still monotone submodular.

5 CENTRALIZED OFFLINE ALGORITHM

In this section, we propose a centralized offline algorithm to address HASTE. After proved that HASTE-R is a problem of maximizing a submodular function under a partition matroid constraint, we can either use a simple greedy algorithm that achieves $\frac{1}{2}$ approximation ratio [30], or a randomized algorithm with optimal approximation guarantees, *i.e.*, $1 - \frac{1}{e}$ approximation ratio, which is, however, too computationally demanding to practically implement. In this paper, we tailor the TABULARGREEDY algorithm [23] to address HASTE-R as it achieves an approximation ratio between $\frac{1}{2}$ and $1 - \frac{1}{e}$ which corresponds to different levels of time complexity by using a control parameter. This provides flexibility in practical applications. We first propose some useful concepts.

- *S-C tuple*. An S-C tuple is a tuple of a scheduling policy for a charger at a time slot and a color from a palette $[C]$ of C colors (note that here color and palette have no concrete meaning, and they are only used to assist sampling). A set $Q \subseteq S \times [C]$ consists of S-C tuples which can be regarded as labeling each scheduling policy for a charger with one or more colors.
- *S-C tuple sampling function.* We associate with each partition $\boldsymbol{\Theta}_{i,k}$ a color $c_{i,k}$. For any set $Q \subseteq S \times [C]$ and vector $\vec{c} = (c_{1,1}, \ldots, c_{n,1}, \ldots, c_{1,K}, \ldots, c_{n,K})$, we define S-C tuple sampling function as

$$\mathtt{sample}_{\vec{c}}(Q) = \bigcup_{i \in [n], k \in [K]} \{x \in \boldsymbol{\Theta}_{i,k} : (x, c_{i,k}) \in Q\}. \quad (3)$$

In other words, $\mathtt{sample}_{\vec{c}}(Q)$ returns a set containing each item x that is exactly labeled with the color $c_{i,k}$ assigned by $\vec{c}$ to the partition $\boldsymbol{\Theta}_{i,k}$ that contains x.

Algorithm 2: Centralized Offline Algorithm to HASTE

Input: Integer C, set of scheduling policies $\boldsymbol{\Theta}_{i,k}$ for charger s_i ($i \in [n]$, $k \in [K]$), objective function $f(.)$
Output: Scheduling policies for all chargers X

$Q \leftarrow \varnothing$;
for *all* $c \in [C]$ **do**
  for *all* $i \in [n]$, $k \in [K]$ **do**
    $e_{i,k,c} \leftarrow \arg\max_{x \in \boldsymbol{\Theta}_{i,k} \times \{c\}} \mathbb{F}(Q + x)$;
    $Q \leftarrow Q \cup e_{i,k,c}$;
for *all* $i \in [n]$, $k \in [K]$ **do**
  Choose $c_{i,k}$ uniformly at random from $[C]$;
$X \leftarrow \mathtt{sample}_{\vec{c}}(Q)$, where $\vec{c} = (c_{1,1}, \ldots, c_{n,1}, \ldots, c_{1,K}, \ldots, c_{n,K})$.
return X

- *Expected charging utility function after S-C tuple sampling.* It is defined as $\mathbb{F}(Q) = \mathbb{E}(f(\mathtt{sample}_{\vec{c}}(Q)))$ as the expected value of $f(\mathtt{sample}_{\vec{c}}(Q))$ when each color $c_{i,k}$ in $\vec{c}$ is selected uniformly at random from $[C]$.

We present our centralized offline algorithm in Algorithm 2. Following Theorem 2 in [23] and assuming $C \to +\infty$ by default, we have the following theorem.

Theorem 5.1. Algorithm 2 achieves $(1-\rho)(1-\frac{1}{e})$ approximation ratio for HASTE, and its time complexity is $O(C(nmK)^2)$ where ρ is the switching delay, C, n, and m are the color number, charger number, and task number, respectively, K is the number of considered time slots.

Proof. Suppose the optimal charging utility for HASTE is $\overline{\mathcal{U}}^*$, and that for HASTE-R is $\overline{\mathcal{U}}^*_R$. Apparently, we have

$$\overline{\mathcal{U}}^*_R \geq \overline{\mathcal{U}}^*. \tag{4}$$

Further, suppose the output X of Algorithm 2 achieves overall charging utility $\overline{\mathcal{U}}_R$ for HASTE-R, *i.e.*,

$$\overline{\mathcal{U}}_R = \sum_{j=1}^{m} w_j \cdot \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j,\, i \in [n],\\ p \in \{p \mid \Theta_{i,k}^p = X \cap \boldsymbol{\Theta}_{i,k}\}}} P_r(s_i, o_j) T_s\Big),$$

and achieves $\overline{\mathcal{U}}$ ($\overline{\mathcal{U}} \leq \overline{\mathcal{U}}_R$) for HASTE by taking the switching delay into consideration. Consider the worst case, *i.e.*, every charger needs to rotate at the beginning of each time slot and lead to switching delay, which results in a time duration of $(1-\rho)T_s$ for effective charging in each time slot for all tasks, then we have

$$\begin{aligned}\overline{\mathcal{U}} &\geq \sum_{j=1}^{m} w_j \cdot \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j,\, i \in [n],\\ p \in \{p \mid \Theta_{i,k}^p = X \cap \boldsymbol{\Theta}_{i,k}\}}} P_r(s_i, o_j)(1-\rho) T_s\Big) \\ &\geq (1-\rho) \sum_{j=1}^{m} w_j \cdot \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j,\, i \in [n],\\ p \in \{p \mid \Theta_{i,k}^p = X \cap \boldsymbol{\Theta}_{i,k}\}}} P_r(s_i, o_j) T_s\Big) \\ &= (1-\rho)\overline{\mathcal{U}}_R. \end{aligned} \tag{5}$$

The second inequality in the above formula is due to the concavity of the charging utility function. Following the classical results in [23] and letting $C \to +\infty$, we have $\overline{\mathcal{U}}_R \geq (1-\frac{1}{e})\overline{\mathcal{U}}^*_R$. Combining it with Equs. (4) and (5), we obtain $\overline{\mathcal{U}} \geq (1-\rho)(1-\frac{1}{e})\overline{\mathcal{U}}^*$, which means Algorithm 2 achieves $(1-\rho)(1-\frac{1}{e})$ approximation ratio. We omit the time complexity analysis to save space. □

6 DISTRIBUTED ONLINE ALGORITHM

In this section, we propose a distributed online algorithm to address HASTE. We face two main challenges. First, we need to adapt the centralized offline algorithm to HASTE, whose relaxed version HASTE-R is a submodular function maximization problem, to cater to the distributed online scenario where all chargers are asynchronous and charging tasks randomly arrive. Nevertheless, to the best of our knowledge, there are no distributed online schemes for maximizing a submodular function with or without constraints. Second, the response of each charger has a delay of up to $\tau + \rho$ time slots, that is, τ number of time slots for computation and negotiation with neighboring chargers and, possibly, plus ρ time slot for switching delay. This setting is fundamentally different from existing ones of online scheduling problems and invalidates traditional online algorithms. We address these challenges by proposing a distributed online algorithm that achieves $\frac{1}{2}(1-\rho)(1-\frac{1}{e})$ competitive ratio.

6.1 Algorithm Description

To begin with, we present some concepts to assist analysis.

- *Neighbors of a charger.* We say two chargers are neighbors to each other if and only if they cover at least one charging task in common. We assume that the communication range of wireless chargers is at least twice of their charging range, and therefore, the neighboring wireless chargers can communicate with each other. The set of neighbors of charger s_i is denoted as $N(s_i)$.
- *Local charging utility function.* The local charging utility function for charger s_i is defined as the aggregated charging utility of all charging tasks that can be charged by s_i, *i.e.*, $\boldsymbol{\mathcal{T}}_i$. Denote by X_i as the set of scheduling policies of s_i, and $\mathcal{X}_i$ the set of scheduling polices of s_i and its neighbors $N(s_i)$, we can formally express the local charging utility function for HASTE-R as

$$f_i(X_i) = \sum_{\mathcal{T}_j \in \boldsymbol{\mathcal{T}}_i} w_j \mathcal{U}\Big(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s} \sum_{\substack{\Gamma_{i,k}^p \ni o_j,\, s_{i'} \in \{s_i\} \cup N(s_i),\\ p \in \{p \mid \Theta_{i,k}^p = \mathcal{X}_i \cap \boldsymbol{\Theta}_{i,k}\}}} P_r(s_{i'}, o_j) T_s\Big)$$

  where K_i is the number of considered time slots for all tasks $\boldsymbol{\mathcal{T}}_i$ observed by charger s_i.
- *Local expected charging utility function after S-C tuple sampling.* Similar to the expected charging utility function after S-C tuple sampling defined in Section 5, we define $\mathbb{F}_i(Q_i) = \mathbb{E}(f_i(\mathtt{sample}_{\vec{c}}(Q_i)))$ as the expected value of $f_i(\mathtt{sample}_{\vec{c}}(Q_i))$ when each color $c_{i,k}$ in $\vec{c}$ is selected uniformly at random from $[C]$.
- *Control message.* The control message exchanged between wireless chargers is expressed as $msg(ID, TIM, COL, CMD, \Delta\mathbb{F}_i^*(Q_i), e_i^{k*})$. The field ID is the charger ID; TIM is the index of the time slots; COL is an integer between 1 and C, which stands for the parameter c in the centralized offline algorithm; CMD can be UPD which indicates an update command; and $\Delta\mathbb{F}_i^{k*}(Q_i)$ is the "maximum" marginal increment for the local expected charging

Algorithm 3: Distributed Online Algorithm to HASTE (at each wireless charger s_i)

Input: Neighbor set $N(s_i)$
Output: Scheduling policy X_i

Update the set of charging tasks that can cover charger s_i, *i.e.*, $\boldsymbol{\mathcal{T}}_i$ to include the new arrived tasks;
Compute the dominant task sets and determine all possible scheduling policies $\boldsymbol{\Theta}_{i,k}$;
Exchange the information of dominant task sets and scheduling policies with the neighbors, and thus derive the local charging utility function $f_i(.)$;
$Q_i \leftarrow \varnothing$;
for k *from* 1 *to* K_i **do**
  for c *from* 1 *to* C **do**
    Calculate $\Delta\mathbb{F}_i^{k*}(Q_i)$ and obtain e_i^{k*};
    Broadcast $msg(i, k, c, NULL, \Delta\mathbb{F}_i^{k*}(Q_i), e_i^{k*})$;
    while $\Delta\mathbb{F}_i^{k*}(Q_i) > 0$ **do**
      if $\Delta\mathbb{F}_j^{k*}(Q_j)$ *of all neighbors* $s_j \in N(s_i)$ *are collected and all their colors are equal to c, and* $\Delta\mathbb{F}_i^{k*}(Q_i)$ *is larger than any of them* **then**
        $Q_i \leftarrow Q_i \cup (e_i^{k*}, c)$;
        Broadcasts $msg(i, k, c, UPD, \Delta\mathbb{F}_i^{k*}(Q_i), e_i^{k*})$;
        break;
      if $msg(j, k, c, UPD, \Delta\mathbb{F}_j^{k*}(Q_j), e_j^{k*})$ *is received* **then**
        Update the stored scheduling policy of its neighbor s_j at the k_{th} time slots to e_j^{k*};
        Calculate $\Delta\mathbb{F}_i^{k*}(Q_i)$ and obtain e_i^{k*};
        Broadcast $msg(i, k, c, NULL, \Delta\mathbb{F}_i^{k*}(Q_i), e_i^{k*})$;
        continue;
      if $msg(j, k, c, NULL, \Delta\mathbb{F}_j^{k*}(Q_j), e_j^{k*})$ *is received* **then**
        Update $\Delta\mathbb{F}_j^{k*}(Q_j)$ and e_j^{k*} for the neighbor s_j;
        continue;
for c *from* 1 *to* C **do**
  Choose c_k^i uniformly at random from $[C]$;
$X_i \leftarrow \texttt{sample}_{\vec{c}}(Q_i)$, where $\vec{c} = (c_1^i, \ldots, c_{K_i}^i)$.
return X_i

utility function after S-C tuple sampling for charger s_i for all possible scheduling policies at the k_{th} time slot, and e_i^{k*} is the corresponding scheduling policy.

We show our distributed online algorithm in Algorithm 3, which is invoked at charger s_i upon arrival of new charging tasks that can be charged by s_i. Each charger accordingly updates the set of charging tasks $\boldsymbol{\mathcal{T}}_i$, all possible scheduling policies in all K_i time slots $\boldsymbol{\Theta}_{i,k}$, and the local charging utility function $f_i(.)$. Then, each charger s_i enumerates all C colors in all K_i time slots. For each color c at the k_{th} time slot, s_i computes $\Delta\mathbb{F}_i^{k*}(Q_i)$ and the corresponding scheduling policy e_i^{k*}, and broadcasts them to its neighbors. Note that $\Delta\mathbb{F}_i^{k*}(Q_i)$ for charger s_i is obtained by greedily choosing the scheduling policies that yield the maximum additional local expected charging utility in all K_i time slots in an increasing order, and therefore, e_i^{k*} is a set of K_i scheduling policies. Meanwhile, s_i receives the control messages sent from its neighbors. If it collects the messages from all its neighbors and finds that it has the maximum value of $\Delta\mathbb{F}_i^{k*}(Q_i)$ (if there are two or more chargers have the same value of $\Delta\mathbb{F}_i^{k*}(Q_i)$ which leads to a tie, we break it based on the IDs of these chargers), s_i adds the S-C tuple (e_i^{k*}, c) to its global S-C tuple set Q_i, and broadcasts the update command to its surrounding neighbors.

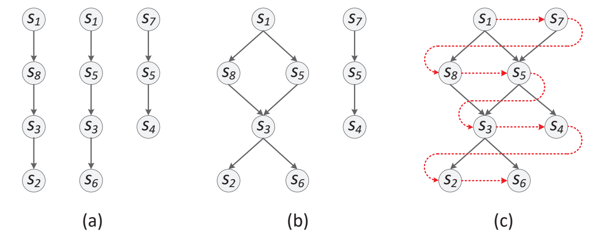

Figure 3: An example of directed acyclic graph construction

Otherwise, if it receives an update command from one of its neighbors, s_i updates the stored scheduling policy for the neighbor, recomputes $\Delta\mathbb{F}_i^{k*}(Q_i)$ and e_i^{k*}, and repeats the above negotiation procedure. After traversing all C colors for all K_i time slots, Algorithm 3 obtains a set of S-C tuples Q_i, and applies a sampling function on Q_i to get a solution X_i.

6.2 Theoretical Analysis

Theorem 6.1. *Algorithm 3 achieves $\frac{1}{2}(1-\rho)(1-\frac{1}{e})$ competitive ratio for HASTE, and its time complexity is $O(C(|N(s_i)||\boldsymbol{\mathcal{T}}_i|K_i)^2)$, its communication cost is $O(CK_i(|N(s_i)|)^2)$ where ρ is the switching delay, C is the number of colors, $N(s_i)$ is the set of neighbors of charger s_i, $\boldsymbol{\mathcal{T}}_i$ is the set of tasks that can cover s_i, K_i is the number of considered time slots for all tasks in $\boldsymbol{\mathcal{T}}_i$.*

Proof. First, we ignore the rescheduling delay of chargers, and prove that the scheduling policies determination processes at all chargers for the online algorithm can be organized in a global order. As the processes of determining scheduling policies for difference colors $c \in [C]$ are in different loops as shown in Algorithm 2, we can equivalently think of the processes of determining scheduling policies for difference colors being isolated from each other and executed in order. For each color, it is clear that the process of determining scheduling policies for a charger s_i and its neighbors is executed in order, which can be expressed as a directed chain with a directed edge between s_i and s_j indicating that the scheduling policies of s_i is determined just left behind that of s_j. Figure 3(a) shows an instance for order chains for nodes s_1, s_3, and s_5. Next, we combine these chains by merging the same nodes. For example, Figures 3(b) and 3(c) illustrate the resulted directed graph when we combine two directed chains corresponding to s_1 and s_3 by merging the two nodes for s_1 and s_3; and further combine the directed chain of s_5 by merging the node for s_5. Then, we obtain a directed graph G, which must be acyclic as otherwise we can always find a charger s_i determining its scheduling policies ahead of itself and thus a contradiction arises. Finally, we apply the well-known linear time topological sorting algorithm presented in [31], to order all the chargers. For example, the red dotted lines in Figure 3(c) connecting all the nodes indicate a topological sort of $s_1 \to s_7 \to s_8 \to s_5 \to s_3 \to s_4 \to s_2 \to s_6$.

Second, clearly the "maximum" marginal increment for the local expected charging utility function after S-C tuple sampling for charger s_i, *i.e.*, $\Delta\mathbb{F}_i^{k*}(Q_i)$, computed by each charger is exactly equal to the "maximum" marginal increment for the global expected charging utility function after S-C tuple sampling. Then, all chargers can be regarded as sequentially determining their scheduling policies based on the global knowledge of the expected charging utility function after S-C tuple sampling as that in the centralized algorithm.

Third, in Algorithm 3, the loop for enumerating all time slots is outside the loop for enumerating all colors. This is critical because as such, the process of being interrupted by arrivals of new charging tasks, recomputing the new scheduling policies and carrying out these new polices for Algorithm 3 can be equivalently viewed as the fluent process with all charging tasks are known a priori. Though Algorithm 3 differs from Algorithm 2 in that the latter has the loop for enumerating all time slots being inside the loop for enumerating all colors, it makes no difference in the ultimate performance guarantee. We omit detail analysis to save space.

To sum up, we claim that Algorithm 3 achieves the same performance as Algorithm 2. Next, we consider rescheduling delay. First, we neglect the switching delay as for HASTE-R. Suppose the global solution X based on the outputs X_i of Algorithm 3 achieves charging utility $\overline{\mathcal{U}}_R$ for HASTE-R, *i.e.*,

$$\overline{\mathcal{U}}_R=\sum_{j=1}^{m} w_j\cdot\mathcal{U}(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s}\sum_{\substack{\Gamma_{i,k}^p\ni o_j,\, i\in[n],\\ p\in\{p|\Theta_{i,k}^p=X\cap\boldsymbol{\Theta}_{i,k}\}}} P_r(s_i,o_j)T_s).$$

Due to rescheduling delay, the reaction of each charger for a newly arrived charging task is delayed for $\tau\cdot T_s$ time. Therefore, it can be equivalently considered that there is no rescheduling delay for chargers under the setting where the first τ time slots of all the charging tasks are "cut off". Suppose X achieves charging utility $\overline{\mathcal{U}}'_R$ for this setting, *i.e.*,

$$\overline{\mathcal{U}}'_R=\sum_{j=1}^{m} w_j\cdot\mathcal{U}(\sum_{k=t_r^j/T_s+\tau+1}^{t_e^j/T_s}\sum_{\substack{\Gamma_{i,k}^p\ni o_j,\, i\in[n],\\ p\in\{p|\Theta_{i,k}^p=X\cap\boldsymbol{\Theta}_{i,k}\}}} P_r(s_i,o_j)T_s).$$

Obviously, we have $\overline{\mathcal{U}}_R\geq\overline{\mathcal{U}}'_R$ as each task misses the opportunity to harvest charging power at its first τ time slots. Assume the optimal overall charging utility for the above setting is $\overline{\mathcal{U}}_R^{'*}$, then we have

$$\overline{\mathcal{U}}_R\geq\overline{\mathcal{U}}'_R\geq(1-\frac{1}{e})\overline{\mathcal{U}}_R^{'*}. \tag{6}$$

Further, assume that the optimal overall charging utility for HASTE-R is $\overline{\mathcal{U}}_R^*$ and its corresponding solution is X^*. Due to the concavity of the charging utility function, we have

$$\begin{aligned}\overline{\mathcal{U}}_R^*=&\sum_{j=1}^{m} w_j\cdot\mathcal{U}(\sum_{k=t_r^j/T_s+1}^{t_e^j/T_s}\sum_{\substack{\Gamma_{i,k}^p\ni o_j,\, i\in[n],\\ p\in\{p|\Theta_{i,k}^p=X^*\cap\boldsymbol{\Theta}_{i,k}\}}} P_r(s_i,o_j)T_s)\\ \leq&\sum_{j=1}^{m} w_j\cdot\mathcal{U}(\sum_{k=t_r^j/T_s+1}^{k=t_r^j/T_s+\tau}\sum_{\substack{\Gamma_{i,k}^p\ni o_j,\, i\in[n],\\ p\in\{p|\Theta_{i,k}^p=X^*\cap\boldsymbol{\Theta}_{i,k}\}}} P_r(s_i,o_j)T_s)\\ &+\sum_{j=1}^{m} w_j\cdot\mathcal{U}(\sum_{k=t_r^j/T_s+\tau+1}^{t_e^j/T_s}\sum_{\substack{\Gamma_{i,k}^p\ni o_j,\, i\in[n],\\ p\in\{p|\Theta_{i,k}^p=X^*\cap\boldsymbol{\Theta}_{i,k}\}}} P_r(s_i,o_j)T_s)\\ \leq&\overline{\mathcal{U}}_R^{*1}+\overline{\mathcal{U}}_R^{*2}.\end{aligned} \tag{7}$$

Note that $\overline{\mathcal{U}}_R^{*1}$ and $\overline{\mathcal{U}}_R^{*2}$ denote the first and second terms at the right hand side of the second inequality. We have

$$\overline{\mathcal{U}}_R^{*2}\leq\overline{\mathcal{U}}_R^{'*}, \tag{8}$$

as the latter is optimal under the same setting. Second, recall that all the charging tasks have a duration of at least $2\tau T_s$ where τ is the switching delay, which indicates $t_e^j/T_s-(t_r^j/T_s+\tau+1)+1\geq(t_r^j/T_s+\tau)-(t_r^j/T_s+1)+1$. Thus, the duration of each task regarding $\overline{\mathcal{U}}_R^{*2}$ is greater than or equal to that of the corresponding task regarding $\overline{\mathcal{U}}_R^{*1}$. Notice that we can move the starting time points of all tasks regarding $\overline{\mathcal{U}}_R^{*1}$ for τ time slots along the time dimension to make them aligned with the corresponding tasks regarding $\overline{\mathcal{U}}_R^{*2}$, we have

$$\overline{\mathcal{U}}_R^{*1}\leq\overline{\mathcal{U}}_R^{'*}. \tag{9}$$

Combining Equs. (6), (7), (8), and (9), we obtain $\overline{\mathcal{U}}_R\geq\frac{1}{2}(1-\frac{1}{e})\overline{\mathcal{U}}_R^*$. Thus Algorithm 3 achieves $\frac{1}{2}(1-\frac{1}{e})$ competitive ratio. By similar analysis on switching delay as in the proof to Theorem 5.1, the achieved competitive ratio of Algorithm 3 is $\frac{1}{2}(1-\rho)(1-\frac{1}{e})$. We omit the analysis of time complexity and communication cost to save space. □

7 SIMULATION RESULTS

In this section, we perform simulations to evaluate the performance of our proposed algorithms. We omit the simulation results for the centralized offline algorithm to save space.

7.1 Evaluation and Baseline Setup

The considered field is a $50\,m\times50\,m$ square area, and wireless chargers and charging tasks are uniformly distributed in this field. We set $\alpha=10000$, $\beta=40$, $D=20\,m$, $n=50$, $m=200$, $w_j=\frac{1}{200}$, $T_s=1\,min$, $\rho=\frac{1}{12}$, $\tau=1$, $A_s=\pi/3$, $A_o=\pi/3$, respectively. The required charging energy and duration of charging tasks are randomly selected in $[5\,kJ\ 20\,kJ]$ and $[10\,min\ 120\,min]$, respectively. Each data point in the figures stands for an averaging result for 100 random topologies. We propose two algorithms named GreedyUtility and GreedyCover for comparison. For GreedyUtility, each charger greedily picks the orientation that leads to maximum charging utility while ignoring the scheduling policies of its neighboring chargers. For GreedyCover, the difference compared with GreedyUtility is that each charger greedily selects the orientation covers the maximum number of tasks.

7.2 Distributed Online Algorithm Evaluation

7.2.1 Impact of Charging Angle A_s. Our simulation results show that on average HASTE outperforms GreedyUtility and GreedyCover by 3.33% and 4.47% (at most 5.59% and 7.59%), respectively, in terms of A_s. We denote by HASTE-DO the distributed online algorithm for HASTE in the following figures. Figure 4 demonstrates that the charging utilities of

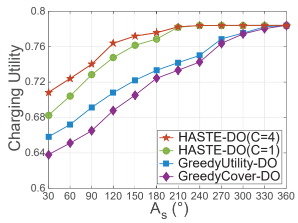

Figure 4: A_s vs. charging utility

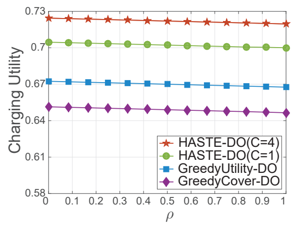

Figure 5: ρ vs. charging utility

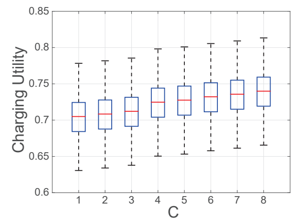

Figure 6: C vs. charging utility

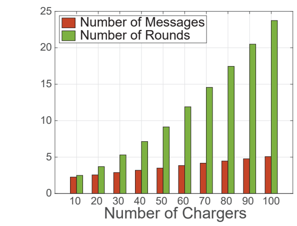

Figure 7: Communication cost

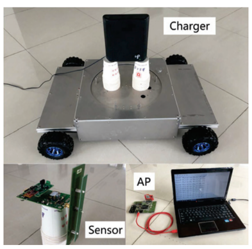

Figure 10: Testbed

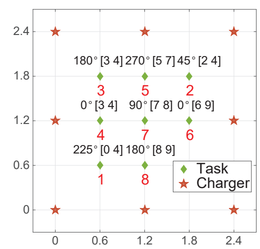

Figure 11: Topology

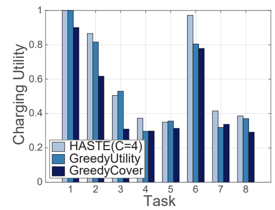

Figure 12: Charging utility for centralized offline algorithm

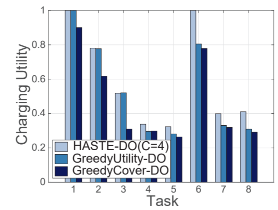

Figure 13: Charging utility for distributed online algorithm

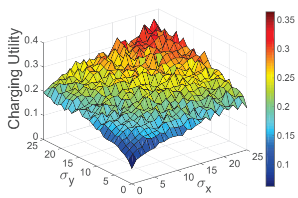

Figure 8: Charging utility vs. Guassian distribution variance

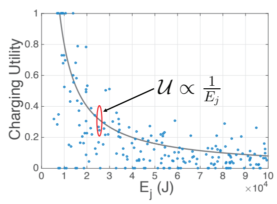

Figure 9: Charging utility vs. required charging energy

HASTE, GreedyUtility, and GreedyCover smoothly increase with the charging angle of chargers A_s, and reach the same maximum overall charging utility when $A_s = 360°$. This is because the larger the charging angle, the larger the chance that a charger can cover more charging tasks with the same orientation, and each charger covers the same set of tasks regardless of its orientations if $A_s = 360°$. The solution for HASTE with $C = 4$ always outperforms that with $C = 1$ with a gain of 0.77% on average (at most 2.59%).

7.2.2 Impact of Switching Delay ρ. Our simulation results show that on average HASTE outperforms GreedyUtility and GreedyCover by 5.20% and 7.3% (at most 5.20% and 7.31%), respectively, in terms of ρ. Figure 5 shows that the charging utilities for all the algorithms steadily decrease with switching delay ρ. HASTE with $C = 4$ outperforms HASTE with $C = 1$ by 1.98%. When the switching delay is even up to one time slot, *i.e.*, $\rho = 1$, the charging utilities for all the algorithms only slightly degrade compared with $\rho = 0$. This is because most chargers keep still most of the time, and thus the caused performance loss is little.

7.2.3 Impact of Color Number C. Our simulation results show that on average the achieved charging utility of HASTE steadily increases with color number C. Figure 6 demonstrates the box plot of the charging utilities of HASTE when the color number C increases from 1 to 8. We can see that both of the maximum and minimum charging utilities of HASTE steadily increase with C. Moreover, on average the average charging utility of HASTE increases by 3.08% when the color number C increases by 1. Besides, the variance of charging utility for all the eight colors is at most 8.42×10^{-3}, which indicates the stable performance of our algorithm.

7.2.4 Communication Cost. Our simulation results show that the number of messages and the number of rounds for a time slot increase quadratically and linearly, respectively, with the number of chargers. We set C to 1, and plot the average numbers of messages and rounds in Algorithm 3 in Figure 7. We can see that when the number of chargers increases from 10 to 100, the numbers of messages and rounds increase by 223.77% and 952.29%, respectively. The number of rounds linearly increases because the number of neighboring chargers linearly increases. Further, as the number of messages in each round also grows proportionally to the number of neighboring chargers, it grows quadratically with the number of chargers.

7.3 Insights

First, we study the impact of distribution for charging tasks on the charging utility. Suppose there are 50 tasks distributed in a $50\,m \times 50\,m$ area, and $A_o = A_s = \pi/3$. The required charging energy and charging duration for all tasks are randomly chosen from $[5\,kJ\ \ 20\,kJ]$ and $[10\,min\ \ 120\,min]$, respectively. The positions of tasks are randomly generated following a 2D Gaussian distribution with both x- and y-coordinates obeying a Gaussian distribution with $\mu = 25$. Figure 8 shows that generally the charging utility increases with either σ_x or σ_y. This is because with a higher degree of uniformness of positions, the phenomenon that some tasks are over-charged while the others are starved out can be largely avoided, and according to the concavity of the charging utility function, the overall charging utility will be enhanced. Second, we study the impact of E_j on the individual charging utility of each charger. We uniformly distribute 50 chargers and 200 tasks. Figure 9 shows that generally the charging utility first can achieve 1 for a small E_j, and then rapidly decreases when E_j continues growing. The maximum individual charging utility is approximately inversely proportional to E_j. The reason is that to achieve the same charging utility, a task with a higher required E_j needs a higher average charging power from its surrounding chargers, which is not cost efficient. Thus, higher E_j leads to lower charging utility.

8 FIELD EXPERIMENTS

We have conducted field experiments to evaluate our scheme. We implemented our proposed schemes on a textbed which consists of eight TX91501 power transmitters produced by Powercast [32] with charging angle of about $60°$, eight rechargeable sensor nodes with receiving angle of about $120°$, and an AP that connects to a laptop for reporting data collected from the nodes as shown in Figure 10. Each power transmitter is mounted on a rotatable platform atop a mobile robot, and thus can be freely rotated. Figure 11 shows the topology of our testbed, where the eight power transmitters are placed at the boundaries of a $2.4\,m \times 2.4\,m$ square area, and the eight sensor nodes are placed inside the square area. We mark the orientation angle and the release and end time (in time slots) on the top of each task associated with a sensor node in Figure 11. The required charging energy for all tasks is set to be in $[3\,J\ 5\,J]$. We set $\alpha = 41.93$, $\beta = 0.6428$, $D = 4\,m$, $\rho = \frac{1}{12}$, $\tau = 1$, $A_s = \pi/3$, $A_o = 2\pi/3$, $w_j = \frac{1}{8}$, based on our empirical results, and set $T_s = 1\,min$. Figures 12 and 13 show the charging utility for each task for the three algorithms, *i.e.*, HASTE (with $C = 4$), GreedyUtility, and GreedyCover, for the centralized offline and distributed online settings, respectively. We can see that on average HASTE basically has the best charging utility for all tasks, and respectively outperforms GreedyUtility and GreedyCover by 8.32% and 26.49% for the centralized offline algorithm; and by 10.40% and 26.19% for the distributed online algorithm. Moreover, task 1 and task 6 have the largest two charging utility for both the algorithms as they have the largest two charging task duration.

9 CONCLUSION

The key novelty of this paper is on proposing the first scheduling algorithm for charging tasks in directional wireless charging networks. The key contributions of this paper are proposing a centralized offline algorithm and a distributed online algorithm both with performance guarantee, and conducting both simulations and field experiments for evaluation. The key technical depth of this paper is in transforming the problem into maximizing a submodular function subject to a partition matroid constraint, bounding the performance loss caused by the switching delay and proving the approximation ratio for the centralized offline algorithm, making the centralized offline algorithm distributed and online and proving its competitive ratio. Our simulation and field experimental results show that our proposed distributed online algorithm can achieve 92.97% of the optimal charging utility and outperform the other two comparison algorithms.

REFERENCES

[1] A. P. Sample *et al.*, "Design of an rfid-based battery-free programmable sensing platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 11, pp. 2608–2615, 2008.
[2] L. Xie *et al.*, "Wireless power transfer and applications to sensor networks," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 140–145, 2013.
[3] "http://www.powercastco.com/pdf/p2110-evb-reva.pdf."
[4] S. Li *et al.*, "Wireless power transfer for electric vehicle applications," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 3, no. 1, pp. 4–17, 2015.
[5] S. F. Bush, *Smart grid: Communication-enabled intelligence for the electric power grid.* John wiley & sons, 2014.
[6] "https://www.wirelesspowerconsortium.com/."
[7] "https://www.wirelesspowerconsortium.com/blog/273/wireless-power-market-surges-as-usage-leaps-forward."
[8] H. Dai *et al.*, "Omnidirectional chargability with directional antennas," in *IEEE ICNP*, 2016, pp. 1–10.
[9] ——, "Optimizing wireless charger placement for directional charging," in *IEEE INFOCOM*, 2017, pp. 2922–2930.
[10] L. Chen *et al.*, "Charge me if you can: Charging path optimization and scheduling in mobile networks," in *ACM MobiHoc*, 2016, pp. 101–110.
[11] S. Nikoletseas *et al.*, "Interactive wireless charging for energy balance," in *IEEE ICDCS*, 2016, pp. 262–270.
[12] L. He *et al.*, "Esync: An energy synchronized charging protocol for rechargeable wireless sensor networks," in *ACM MobiHoc*, 2014, pp. 247–256.
[13] ——, "Mobile-to-mobile energy replenishment in mission-critical robotic sensor networks," in *IEEE INFOCOM*, 2014, pp. 1195–1203.
[14] C. Wang *et al.*, "Improve charging capability for wireless rechargeable sensor networks using resonant repeaters," in *IEEE ICDCS*, 2015, pp. 133–142.
[15] ——, "A hybrid framework combining solar energy harvesting and wireless charging for wireless sensor networks," in *IEEE INFOCOM*, 2016, pp. 1–9.
[16] Z. Dong *et al.*, "Energy synchronized task assignment in rechargeable sensor networks," in *IEEE SECON*, 2016, pp. 1–9.
[17] X. Lu *et al.*, "Wireless charging technologies: Fundamentals, standards, and network applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1413–1452, 2016.
[18] H. Dai *et al.*, "Safe charging for wireless power transfer," in *IEEE INFOCOM*, 2014, pp. 1105–1113.
[19] ——, "SCAPE: Safe charging with adjustable power," in *IEEE ICDCS*, 2014, pp. 439–448.
[20] S. Nikoletseas *et al.*, "Low radiation efficient wireless energy transfer in wireless distributed systems," in *IEEE ICDCS*, 2015, pp. 196–204.
[21] H. Dai *et al.*, "Radiation constrained wireless charger placement," in *IEEE INFOCOM*, 2016, pp. 1–9.
[22] ——, "Radiation constrained scheduling of wireless charging tasks," in *ACM MobiHoc*, 2017, pp. 17–26.
[23] D. Golovin *et al.*, "Online submodular maximization under a matroid constraint with application to learning assignments," *arXiv preprint arXiv:1407.1082*, 2014.
[24] S. He *et al.*, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1931–1942, 2013.
[25] "http://www.sevenoak.biz/Pantiltheads/SK-EBH01.html."
[26] R. Zhang *et al.*, "MIMO broadcasting for simultaneous wireless information and power transfer," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 1989–2001, 2013.
[27] Z. Ding *et al.*, "Application of smart antenna technologies in simultaneous wireless information and power transfer," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 86–93, 2015.
[28] L. Fleischer *et al.*, "Tight approximation algorithms for maximum general assignment problems," in *ACM SODA*, 2006, pp. 611–620.
[29] S. Fujishige, *Submodular functions and optimization.* Elsevier, 2005, vol. 58.
[30] G. L. Nemhauser *et al.*, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
[31] T. H. Cormen, *Introduction to algorithms.* MIT press, 2009.
[32] "www.powercastco.com."