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 \((1 - \rho)(1 - \frac{1}{2})\) approximation ratio to address HASTE where \(\rho\) is the switching delay of chargers. Further, we propose a distributed online algorithm and prove it achieves \(\frac{1}{2}(1 - \rho)(1 - \frac{1}{2})\) 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 their 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

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
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 TABULARGREED algorithm proposed in [23] to address HASTE 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 $\rho$ 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 omnidirectional 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 $S = \{s_1, \ldots, s_n\}$ located in a 2D plane $\Omega$, which can continuously rotate with orientation angle within $[0, 2\pi]$. Suppose there are also some rechargeable devices located in $\Omega$, 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 $T_j = < o_j, \phi_j, t^l_j, t^r_j, E_j >$ where $o_j$ denotes the position of the rechargeable device that raises the task, $\phi_j$ is the orientation of the device, $t^l_j$ and $t^r_j$ 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 $t^l_j$ is exactly at the beginning of a time slot while $t^r_j$ is at the end of a time slot. We will show in the discussion to Lemma 4.2 that even $t^l_j$ and $t^r_j$ 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 $\overrightarrow{r_{\phi_j}}$ 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

$$P_r(s_i, \theta_i, o_j, \phi_j) = \begin{cases} \frac{\alpha}{(\|s_i o_j\|)^2}, & 0 \leq \|s_i o_j\| \leq D, \\ \frac{\alpha}{s_i o_j}, & \|s_i o_j\| > D, \\ 0, & \text{otherwise} \end{cases}$$

where $\alpha$ and $\beta$ are two known constants, and $\|s_i o_j\|$ 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 $\rho$ ($0 < \rho < 1$) 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 $\theta_0 = \emptyset$ 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 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 $\tau$ ($\tau \in \mathbb{Z}_+$) 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 $\rho$ 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., $t^{\rho} - t^\rho \geq 2\tau T_s$ 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.,

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

where $E_j$ is the required charging energy of charging task $T_j$.

### 3.2 Problem Formulation and Hardness Analysis

Let $\theta_i(t)$ ($\theta_i : \mathbb{R}_{\geq 0} \rightarrow \{\emptyset, 2\pi \} \cup \emptyset$) 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 $T_j$, 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 $t^m \sum_{i=1}^n P_r(s_i, \theta_i(t), o_j, \phi_j) dt$. And the overall (weighted) charging utility is $\sum_{j=1}^m w_j U(t^m \sum_{i=1}^n P_r(s_i, \theta_i(t), o_j, \phi_j) dt)$ where $w_j$ is the weight of charging task $T_j$. 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.

With all above, we define the charging task scheduling for directional wireless charger networks (HASTE) as follows.

(P1) $\max_{\theta_i(k)} U = \sum_{j=1}^m w_j \cdot t^m \left\{ \sum_{i=1}^n P_r(s_i, \theta_i(t), o_j, \phi_j) dt \right\}$

s.t. $\theta_i(t)$

$\begin{cases} \emptyset, k T_s < t \leq (k + \rho) T_s \\ \theta_i(k), (k + \rho) T_s < t \leq (k + 1) T_s \\ \theta_i(k), k T_s < t < (k + 1) T_s \end{cases}$

otherwise

where $k \in \mathbb{Z}_+^*$ and $\theta_i(0) = \emptyset$

$0 \leq \theta_i(k) < 2\pi$.

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

We describe our algorithm for extracting dominant task sets in Algorithm 1. Basically, the considered charger rotates for $2\pi$ 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 $T_1$, then rotates to cover tasks $T_2$ and $T_3$ sequentially. Further, $T_4$ cannot be added in the current covered set as otherwise $\{T_1, T_2\}$ will be missed, and therefore, $\{T_1, T_2, T_3\}$ is a dominant task set. Then, the charger continues to cover $T_1$ by removing $T_1$ and $T_2$ from the current set, as shown in Figure 2(b). Similarly, as $T_5$ cannot be covered by the charger without missing $T_4$, $\{T_3, T_4\}$ is added as a dominant task set. Algorithm 1 proceeds until the charger rotates for $2\pi$, as depicted in Figure 2(c) and (d). After all, the obtained dominant task sets are $\{T_1, T_2, T_3\}, \{T_3, T_4\}, \{T_4, T_5\}$ and $\{T_6, T_1\}$.

**Algorithm 1:** Dominant Task Sets Extraction

**Input:** The wireless charger $s_i$, all charging tasks $\{T_j\}_{j=1}^m$

**Output:** All dominant task sets

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

Then, the problem HASTE-R can be formulated as

$$(RP1) \max \sum_{i,k} x_{i,k} \cdot \mathcal{U}(\sum_{k=t_i/T_s+1}^{t_i/T_s} \sum_{j \in [n]} \sum_{\mathcal{T}_p \supseteq \mathcal{T}_i} x_{i,k}^p \cdot \mathcal{P}_t(s_i, o_j) T_s)$$

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

where $x_{i,k}^p$ are the decision variables, $I_i^p$ is the $p_{th}$ dominant task set in $I_i$. 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 $f : 2^S \rightarrow \mathbb{R}$ is normalized, monotonic and submodular if and only if it satisfies the following conditions, respectively: (1) $f(\emptyset) = 0$; (2) $f(A \cup \{e\}) - f(A) \geq 0$ for any $A \subseteq S$ and $e \in S \setminus A$; (3) $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ for any $A \subseteq B \subseteq S$ and $e \in S \setminus B$.

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

**Definition 4.4.** [29] (partition matroid) Given $S = \bigcup_{i=1}^k S_i$, $S_i$ 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 $\mathcal{I} = \{X \subseteq S : |X \cap S_i| \leq l_i \text{ for } i \in [k]\}$.
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 \( \Gamma_{i,k}^{\phi} \) as the \( p_{th} \) dominant task set in \( \Gamma_{i,k} \). Then, we define \( \Theta_{i,k}^{\phi} \) as the corresponding scheduling policy for \( \Gamma_{i,k}^{\phi} \), i.e., the orientation that covers \( \Gamma_{i,k}^{\phi} = \Gamma_i^{\phi} \), for charger \( s_i \) at the \( k_{th} \) time slot, define \( \Theta_{i,k} = \{ \Theta_{i,k}^{p} \}_{p \in [P]} \) as the set of scheduling policies for \( s_i \) at the \( k_{th} \) time slot, and define a ground set of all scheduling policies \( S = \{ \Theta_{i,k} \}_{i \in [m], k \in [K]} \). Further, we define the scheduling policies for all chargers at all \( K \) time slots as \( X \), which is subject to \( |X \cap \Theta_{i,k}| \leq 1 \). As \( \Theta_{i,k} \) are disjoint sets, we write the independent sets as

\[
I = \{ X \subseteq S : |X \cap \Theta_{i,k}| \leq 1 \text{ for } i \in [n], k \in [K] \}. 
\]

**Lemma 4.1.** The constraint in the problem \( \text{RP1} \) can be rewritten as a partition matroid on the ground set \( S \).

Accordingly, problem \( \text{RP1} \) can be rewritten as (RP2)

\[
\max_{X} f(X) = \sum_{j=1}^{m} w_j \cdot U(\sum_{k=t_{j}/T_s}^{t_{j}/T_s+1} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s)\]

s.t. \( X \in I \)

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., \( X = \emptyset \), the received energy for any task is zero, then \( f(X) = 0 \). Second, let \( A \) be a set of scheduling strategies in \( S \) and \( e \in S \setminus A \). For simplicity, define

\[ g(X, j) = U(\sum_{k=t_{j}/T_s}^{t_{j}/T_s+1} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s) \]

as the achieved utility for task \( T_j \). It is easy to see that \( g(A \cup \{e\}, j) - g(A, j) \geq 0 \) because there are possibly more chargers covering task \( T_j \) as all dominant possible task sets that cover \( T_j \), i.e., \( \Gamma_{i,k} \) \((i \in [n], p \in \{ p | \Theta_{i,k}^{p} = A \cap \Theta_{i,k} \}) \) would be enlarged as \( A \) becomes \( A \cup \{e\} \) and the utility function \( U(.) \) 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 \( e \in S \setminus B \). On one hand, it is easy to see that

\[ \sum_{k=t_{j}/T_s}^{t_{j}/T_s+1} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s \]

\[ \frac{1}{t_{j}/T_s} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s - \frac{1}{t_{j}/T_s} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s \]

\[ = \frac{1}{t_{j}/T_s} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s - \frac{1}{t_{j}/T_s} \sum_{i \in [n], j \in [n]} P_r(s_i, o_j)T_s \]

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

\[ (U(x_1 + \Delta x) - U(x_1)) - (U(x_2 + \Delta x) - U(x_2)) \geq 0, \]

for any \( x_2 \geq x_1 \geq 0 \) and \( \Delta x \geq 0 \) due to the concavity of the charging utility function \( U(.) \). 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 hold. Second, if \( t_{j}^{\phi} \) and \( t_{j}^{\phi} \) 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 \( \Theta_{i,k} \) a color \( c_{i,k} \). For any set \( Q \subseteq S \times [C] \) and vector \( \vec{c} = (c_1, 1, \ldots, c_n, 1, \ldots, c_K, \ldots, c_K) \), we define S-C tuple sampling function as

\[
\text{Sample}_Q(p) = \bigcup_{i \in [n], k \in [K]} \{ x \in \Theta_{i,k} : (x, c_{i,k}) \in Q \}.
\]
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 \( U_R \geq (1 - \rho)T^* \). Combining it with Eqs. (4) and (5), we obtain \( U \geq (1 - \rho)(1 - 2\epsilon)T^* \), which means Algorithm 2 achieves \( (1 - \rho)(1 - 2\epsilon) \) 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, \( \tau \) number of time slots for computation and negotiation with neighboring chargers and, possibly, plus \( \rho \) 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 \( \frac{1}{2}(1 - \rho)(1 - \frac{1}{2}\epsilon) \) 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 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., \( T_i \). Denote by \( X_i \) the set of scheduling policies of \( s_i \), and \( X_i \) the set of scheduling policies of \( s_i \) and its neighbors \( N(s_i) \), we can formally express the local charging utility function for HASTE-R as

\[
U_i(X_i) = \frac{w_j \cdot U_i(T_j)}{1 - \rho T_i} \sum_{k=1}^{m} P_r(s_i, o_j) T_k
\]

where \( K_i \) is the number of scheduled charging tasks for charger \( s_i \). When each color \( c_{i,k} \) is selected uniformly at random from \( [C] \),

- **Control message.** The control message exchanged between wireless chargers is expressed as \( \text{msg}(ID, T, CMD) \). The field \( ID \) is the charger ID; \( T \) is the index of the time slot; \( CMD \) is an integer between 1 and \( C \), which stands for the parameter \( c \) in the centralized offline algorithm; \( CMD \) can be \( \text{UPD}^c \) which indicates an update command; and \( \Delta F^c(Q) \) 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 $X_i$
1. Update the set of charging tasks that can cover charger $s_i$, i.e., $T_i$, to include the new arrived tasks;
2. Compute the dominant task sets and determine all possible scheduling policies $\Theta_{i,k}$;
3. Exchange the information of dominant task sets and scheduling policies with the neighbors, and thus derive the local charging utility function $f_i(\cdot)$;
4. $Q_i \leftarrow \emptyset$;
5. for $k$ from 1 to $K_i$, do
   for $c$ from 1 to $C$ do
      Calculate $\Delta_E^{k+}(Q_i)$ and obtain $e_i^{k+}$;
      Broadcast $msg(i, c, NULL, \Delta_E^{k+}(Q_i), e_i^{k+})$;
      while $\Delta_E^{k+}(Q_i) > 0$ do
         if $\Delta_E^{k+}(Q_i)$ of all neighbors $s_j \in N(s_i)$ are collected and all their colors are equal to $c$, and $\Delta_E^{k+}(Q_i)$ is larger than any of them then
            $Q_i \leftarrow Q_i \cup \{e_i^{k+}, c\}$;
            Broadcasts $msg(i, k, c, UPD, \Delta_E^{k+}(Q_i), e_i^{k+})$;
         break;
         if $msg(j, k, c, UPD, \Delta_E^{k+}(Q_i), e_j^{k+})$ is received then
            Update the stored scheduling policy of its neighbor $s_j$ at the $k_{th}$ time slots to $e_j^{k+}$;
            Calculate $\Delta_E^{k+}(Q_i)$ and obtain $e_i^{k+}$;
            Continue;
         if $msg(j, k, c, NULL, \Delta_E^{k+}(Q_i), e_j^{k+})$ is received then
            Update $\Delta_E^{k+}(Q_i)$ and $e_i^{k+}$ for the neighbor $s_j$;
            Continue;
22. for $c$ from 1 to $C$ do
23. Choose $c_i$ uniformly at random from $[C]$;
24. $X_i \leftarrow \text{sample}(Q_i)$, where $\bar{e} = (e_1, \ldots, e_K)$.
25. 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 $T_i$, all possible scheduling policies in all $K_i$ time slots $\Theta_{i,k}$, and the local charging utility function $f_i(\cdot)$. 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 $\Delta_E^{k+}(Q_i)$ and the corresponding scheduling policy $e_i^{k+}$, and broadcasts them to its neighbors. Note that $\Delta_E^{k+}(Q_i)$ 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 $\Delta_E^{k+}(Q_i)$ (if there are two or more chargers have the same value of $\Delta_E^{k+}(Q_i)$ 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.

Otherwise, if it receives an update command from one of its neighbors, $s_i$ updates the stored scheduling policy for the neighbor, recomputes $\Delta_E^{k+}(Q_i)$ 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 $\frac{1}{2}(1-\rho)(1-\frac{1}{C})$ competitive ratio for HASTE, and its time complexity is $O(C(|N(s_i)|)\sum_{k=1}^{K_i}T)$, its communication cost is $O(CK_i(|N(s_i)|)^2)$ where $\rho$ is the switching delay, $C$ is the number of colors, $N(s_i)$ is the set of neighbors of charger $s_i$, $T_i$ is the set of tasks that can cover $s_i$, $K_i$ is the number of considered time slots for all tasks in $T_i$.

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 $s_1, s_2, s_3$, and $s_4$. 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 $s_2$ 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 charging path merging the node for $s_i$.

Figure 3: An example of directed acyclic graph construction
sampling for charger $s_i$, i.e., $\Delta P_i^k(Q_i)$, 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 policies 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 $\bar{U}_R$ for HASTE-R, i.e.,

$$\bar{U}_R = \sum_{j=1}^{m} w_j \cdot U(\sum_{k=t_j^i/T_s+1}^{t_j^i/T_s+\tau+1} P_r(s_i, \alpha_j) T_s).$$

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 $\tau$ time slots of all the charging tasks are “cut off”. Suppose $X$ achieves charging utility $\bar{U}_R$ for this setting, i.e.,

$$\bar{U}_R = \sum_{j=1}^{m} w_j \cdot U(\sum_{k=t_j^i/T_s+\tau+1}^{t_j^i/T_s+\tau+1} P_r(s_i, \alpha_j) T_s).$$

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

$$\bar{U}_R \geq \bar{U}_R' \geq (1 - \frac{1}{e}) \bar{U}_R^*.$$  \hspace{1cm} (6)

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

$$\bar{U}_R = \sum_{j=1}^{m} w_j \cdot U(\sum_{k=t_j^i/T_s+1}^{t_j^i/T_s+\tau+1} P_r(s_i, \alpha_j) T_s) \leq \sum_{j=1}^{m} w_j \cdot U(\sum_{k=t_j^i/T_s+\tau+1}^{t_j^i/T_s+\tau+1} P_r(s_i, \alpha_j) T_s) \leq \sum_{j=1}^{m} w_j \cdot U(\sum_{k=t_j^i/T_s+\tau+1}^{t_j^i/T_s+\tau+1} P_r(s_i, \alpha_j) T_s) \leq (1 - \frac{1}{e}) \bar{U}_R^*.$$  \hspace{1cm} (7)

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

$$\overline{U}^2_R \leq \overline{U}_R^*.$$  \hspace{1cm} (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 $\tau$ is the switching delay, which indicates $t_j^i/T_s - (t_j^i/T_s + \tau + 1) + 1 \geq (t_j^i/T_s + \tau) - (t_j^i/T_s + 1) + 1$. Thus, the duration of each task regarding $\overline{U}^1_R$ is greater than or equal to that of the corresponding task regarding $\overline{U}_R$.

We denote by $HASTE-DO$ the centralized offline algorithm to save space.

$$\overline{U}^2_R \leq \overline{U}_R^*.$$  \hspace{1cm} (9)

Combining Eqs. (6), (7), (8), and (9), we obtain $\overline{U}_R \geq \frac{1}{2}(1 - \frac{1}{e}) \overline{U}_R$. 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 $\frac{1}{2}(1 - (1 - \frac{1}{e})$.

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 $50m \times 50m$ square area, and wireless chargers and charging tasks are uniformly distributed in this field. We set $\alpha = 100000$, $\beta = 40$, $D = 20m$, $m = 50$, $m = 200$, $w_j = \frac{1}{20}, T_s = 1min$, $\rho = \frac{1}{12}, \tau = 1, A_s = \pi/3, A_r = \pi/3$, respectively. The required charging energy and duration of charging tasks are randomly selected in $[5kJ 20kJ]$ and $[10min 120min]$, 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 Evaluation.

7.2 Distributed Online Algorithm

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
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 $8.42 \times 10^{-3}$, 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 $50 \times 50 m$ area, and $A_0 = A_\pi = \pi/3$. The required charging energy and charging duration for all tasks are randomly chosen from $[5 \text{kJ}, 20 \text{kJ}]$ and $[10 \text{min}, 120 \text{min}]$, 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 $\sigma_x$ or $\sigma_y$. 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 testbed 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 × 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 $\alpha = 41.93$, $\beta = 0.6428$, $D = 4 m$, $\rho = \frac{\pi}{3}$, $\tau = 1$, $A_0 = \pi/3$, $A_0 = 2\pi/3$, $w_1 = \frac{1}{3}$, based on our experimental results, and $T_1 = 1 min$. 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

[6] [https://www.wirelesspowerconsortium.com/].
[7] [https://www.wirelesspowerconsortium.com/blog/273/wireless-power-market-surges-as-usage-leaps-forward.].
[25] [http://www.sevenoak.biz/Pantiltheads/SK-EBH01.html].
[32] [http://www.powercastco.com/].