Energy Synchronized Task Assignment in Rechargeable Sensor Networks

Zheng Dong, Cong Liu
University of Texas at Dallas

Lingkun Fu, Peng Cheng
Zhejiang University

Liang He
University of Michigan at Ann Arbor

Yu Gu‡, Wei Gao‡
IBM Research-Austin‡
University of Tennessee†

Chau Yuen
Singapore University of Technology and Design

Tian He
University of Minnesota

Abstract

Wireless rechargeable sensor networks have recently emerged as a promising platform that can effectively solve the power constraint problem suffered by traditional battery-powered systems. The problem of determining the best charging routes for maximizing charging efficiency has been studied extensively. However, the task assignment problem, which plays a crucial role in efficiently utilizing the harvested energy and thus minimize the charging delay, has received rather limited attention. In this paper, we study the problem of assigning a given set of tasks in a wireless rechargeable sensor network while maximizing the charger’s velocity to minimize the charging delay. We first propose an online task assignment algorithm, namely Lower Bound assignment (L-B), that yields a quantifiable lower bound on the charging velocity while guaranteeing a feasible assignment. This algorithm further enables the transformation of our considered task assignment problem into a variation of the classical multiple knapsack problem. We then present a fully polynomial-time approximation scheme with a \((2 + \varepsilon)\)-approximation ratio, namely ACT, that is built upon an existing greedy algorithm designed for the original knapsack problem. Extensive experimental results presented herein demonstrate that ACT is able to achieve near-optimal performance in most cases, and can achieve more than 15% performance improvement compared to the baseline algorithms.

1 Introduction

Wireless sensor networks (WSNs) have been widely adopted in a variety of application domains. In a typical scenario, sensors are deployed in a certain region for executing a large number of sensing, computing, and communication tasks. Unfortunately, stringent energy constraints posed by traditional battery-powered WSNs have become a critical bottleneck that impedes long-term operations of such systems.

Through recent advances in wireless recharging technologies [15], wireless rechargeable sensor networks (WRSNs) have emerged as a promising platform to effectively solve the energy constraint problem [6, 7, 9]. A WRSN often deploys one or more mobile chargers that traverse along an existing infrastructure and replenish the dissipated energy of sensor nodes [8, 7, 30]. For many practical applications of WRSNs, the charging delay is not negligible and plays a critical role in minimizing the system decision/operation cycle time [7, 30]. Prior works have focused on investigating the issue of determining the best traversing routes while assuming constant charger traversing velocity for minimizing the overall charging delay [7] and maximizing the network lifetime [30].

Figure 1: The charger wirelessly charges the sensor nodes along the road segment before each monitoring period.

While identifying the optimal or near-optimal routes is essential to minimize the charging delay, for many applications, such as warehouse and traffic monitoring [22, 17], the charger’s traversing route is typically fixed due to geographical constraints. An example motivational scenario is vehicle monitoring along a road network [11], as illustrated in Fig. 1. Sensor nodes are deployed along the road and the sensing range of each sensor node covers part of the road segment. The road segment can be entirely covered by all the sensor nodes. A service vehicle carries wireless chargers and travels along the road network to power these sensor nodes. The monitoring task is composed by a set of sensing tasks utilizing sensors including image, magnetic, sound and radiation [11, 17]. During a monitoring period, due to the computational and energy capacity limitation, a single sensor node normally can not handle all tasks. Thus, different sensing tasks need to be assigned and distributed to different nodes for accomplishing the monitoring task collectively. These nodes gather different sensing data to be fused at the sink. In such scenarios with fixed traversing routes, identifying better routes becomes relatively easy. Rather, maximizing the charger’s traversing velocity through optimally utilizing the charged energy on sensor nodes becomes critical in order to minimize the charging delay.

This problem of maximizing the charger’s traversing velocity is challenging because sensor nodes in a WRSN need to be wireless charged to a degree such that they harvest e-
nough energy for executing a given number of sensing and computing tasks, each of which possesses a specific energy requirement. This clearly conflicts with the objective of maximizing the charger’s traversing velocity, where a faster charging velocity may cause reduced amount of energy harvested by sensor nodes (the detailed charging model is given in Sec. 2). Moreover, under a specific traversing velocity, the amount of energy harvested by different sensor nodes varies due to their different geographical locations. Our observation herein is that the strategy of assigning tasks to sensor nodes is critical in maximizing the traversing velocity. A judicious task assignment may lead to a reduced amount of energy that sensor nodes require to complete all given tasks and better utilize the energy harvested by each individual sensor, thus allowing an increased charging velocity. Motivate by this, we investigate in this paper the problem of maximizing the charger’s traversing velocity (thus minimizing the charging delay) while guaranteeing a feasible task assignment.

Specifically, we consider the problem of assigning a given set of tasks in a WRSN while the charger travels along a fixed route charging sensor nodes. Our objective is to minimize the charging delay while ensuring that sensor nodes harvest enough energy for executing all tasks. To achieve this objective, we design novel online task assignment algorithms that seek to minimize the amount of required energy harvested by sensor nodes, thus yielding the maximum feasible charging velocity.

The major contributions of this paper are listed as follows:

- We propose a new task assignment algorithm, namely LB (Lower Bound) assignment, in WRSNs, which seeks to minimize the charging delay. LB is an online algorithm, meaning that tasks are assigned dynamically at runtime while the charger is traversing and charging the sensor nodes. LB yields an upper bound on the total amount of required energy that sensor nodes need to harvest in order to execute all given tasks. It thus yields a quantifiable lower bound on the charging velocity while guaranteeing a feasible assignment. We also analytically prove that the ratio of the charging velocity achieved by LB over an upper bound on the charging velocity is a constant factor that can be computed provided specific task and system parameters.

- Moreover, the LB algorithm is significant in the sense that it further enables us to formulate the task assignment problem as a multiple knapsack problem with variable capacity for each knapsack. Based upon this problem formulation, we present an improved polynomial time approximation scheme (ACT) with \((2 + \epsilon)\)-approximation ratio. Our proposed ACT leverages a classical multiple knapsack solution and applies several efficient optimization techniques.

- We have conducted extensive experiments, which demonstrate that our proposed algorithms are able to achieve near-optimal performance compared to the optimal solution, and can achieve > 15% performance improvement compared to a baseline algorithm.

The rest of this paper is organized as follows. Sec. 2 describes our system model and problem formulation. In Sec. 3, we present the LB assignment and the resulting lower bound on the charger velocity. Then in Sec. 4, we present a PTAS with an approximation ratio of \(2 + \epsilon\) that yields a faster charger velocity. We evaluate our design in Sec. 5. We discuss related works in Sec. 6 and conclude in Sec. 7.

2 Preliminaries

In this section, we present models used for the wireless charging and the task assignment problem in wireless rechargeable sensor network.

2.1 Mobile Wireless Charging Model

Wireless recharging has become a promising technology to address sustainable problem for battery-powered devices. Much recent work [18, 20, 24] has shown that rechargeable sensor nodes can harvest energy from ambient radio frequency signals. Wireless Identification and Sensing Platform (WISP) [25], developed by Intel Research, is one of the most representative wireless rechargeable sensor node platform for many sensing applications such as Passive Data Logger (PDL) [32] and daily activity recognition [1].

To model the wireless charging power, He et al. [9] propose a wireless recharging model based on the Friis’ free space equation as follows:

\[
P_r = \frac{G_s G_r \eta}{L_p} \left(\frac{\lambda}{4\pi(d + \beta)}\right)^2 P_0,\]

where \(d\) is the distance between the sensor node and charger. All other parameters are constant based on the environment and device settings. \(P_0\) is the source power, \(G_s\) is the source antenna gain, \(G_r\) is the receive antenna gain, \(L_p\) is polarization loss, \(\lambda\) is the wavelength, \(\eta\) is rectifier efficiency, and \(\beta\) is a parameter to adjust the Friis’ free space equation for short distance transmission. This model has been widely used in prior work [7, 9, 27]. We have also validated this model by conducting measurements-based experiments, which study the relationship between the charged power and the distance between a charger and a sensor node. The experiments were conducted on WISP and the measured results are shown in Fig. 2.

According to Eq. (1), the only variable that affects the charged power of a sensor node is its distance from the charger within a given environment. Thus, for readability, we abbreviate the charging model in Eq. (1) as:

\[
P = \kappa(d).
\]

Based on this charging model, we calculate the charging
power for a mobile charger. In Fig. 3, the charger travels along a trajectory \( AB \), charging the sensor node \( s \) continuously. The charger’s location is a function of time \( t \), which can be denoted as \((x(t), y(t))\), and the coordinates of sensor node \( s \) is denoted \((x_s, y_s)\). Therefore, the distance between the sensor node \( s \) and the charger can be computed as follows:

\[
f_a(t) = \sqrt{(x(t) - x_s)^2 + (y(t) - y_s)^2}.
\]

Then based on Eqs. (2) and (3), the charging power at time \( t \) for the sensor node \( s \) is \( \kappa(f_a(t)) \) and the total amount of charged energy \( E_s(v) \) on sensor node \( s \) can be denoted as:

\[
E_s(v) = \int_0^L \kappa(f_a(t)) \, dt,
\]

where \( v \) is the charger’s velocity and \( L \) is the length of \( AB \). From Eq. (4), we can see that given a fixed travel trajectory of the charger, the total amount of charged energy on the sensor \( s \), \( E_s(v) \), only depends on the charger’s travel velocity.

### 2.2 Problem Statement

We consider a wireless rechargeable sensor network with a set \( S = \{s_1, s_2, s_3, \ldots, s_m\} \) of \( m \) stationary sensor nodes, which need to execute a set \( T = \{T_1, T_2, T_3, \ldots, T_n\} \) of \( n \) tasks. \(^{1}\)

Each task \( T_i \) requires \( e_i \) amount of energy for execution, \( e_i \), equals to the sensing power of \( s_i \) multiplied by the monitoring period. The locations of individual nodes can be obtained with solutions proposed in [27] and represented as \((x_j, y_j)\) for node \( j \). Given the location of each node and the charger velocity \( v \) along a trajectory, we can estimate the charged energy on a node \( s_j \), denoted \( E_j(v) \), using Eq. (4). Therefore, the total amount of charged energy on all sensor nodes, denoted \( E_{\text{total}} \), is:

\[
E_{\text{total}} = \sum_{j=1}^{m} E_j(v)
\]

Since \( E_j(v) \) is a monotonically decreasing function of \( v \) according to Eq. (4), \( E_{\text{total}} \) is also a monotonically decreasing function of \( v \). Consequently, our objective is to find the maximum velocity \( v_{\text{max}} \), for the charger, while guaranteeing that all the tasks can be completed.

Energy is an essential constraint for the task assignment problem. A node can not execute a task if the required amount of energy for executing the task is more than the harvested energy. Formally, let \( A \) denote a task assignment, where \( x_{i,j} \) is a binary decision variable. Let \( x_{i,j} = 1 \), if \( T_i \) is assigned to sensor node \( s_j \), and 0 otherwise. In order to execute all assigned tasks on \( s_j \), the amount of charged energy on \( s_j \) should be more than the required amount of energy, i.e.,

\[
\sum_{i=1}^{n} x_{i,j} e_i \leq E_j(v).
\]

A task assignment is feasible under a charger’s traversing velocity \( v \) if under this assignment, all sensors have enough charged energy for executing their assigned tasks.

If both Eq. (6) and \( \sum_{i=1}^{n} x_{i,j} e_i = n \) hold, then the charged energy on \( m \) nodes is sufficient to execute all \( n \) tasks. Therefore, we can formulate our problem as the following integer linear programming (ILP):

\[
\begin{align*}
\text{max} & \quad v \\
\text{subject to} : & \quad \sum_{j=1}^{m} x_{i,j} \leq 1 \quad (8) \\
& \quad \sum_{i=1}^{n} x_{i,j} \leq E_j(v) \quad (9) \\
& \quad \sum_{j=1}^{m} \sum_{i=1}^{n} x_{i,j} = n \quad (10) \\
& \quad x_{i,j} \in \{0,1\} \quad (11)
\end{align*}
\]

### 3 A Feasible Lower Bound on Velocity

Although solving the above ILP guarantees an optimal solution, it takes exponential time to search the solution space, which is not practical. In this section, we present the LB algorithm, which is an online task assignment algorithm that yields an upper bound on the total amount of required energy that sensor nodes need to harvest in order to execute all given tasks (Theorem 1). Thus, LB yields a quantifiable lower bound on the charging velocity while guaranteeing a feasible assignment. Before describing the algorithm, we first give necessary definitions.

**Definition 1.** Let \( E_1, E_2, \ldots, E_m \) denote the amount of charged energy on sensor nodes \( s_1, s_2, \ldots, s_m \), when the charger travels with speed \( v_{\text{max}} \) along its trajectory. Let \( e_{\text{sum}} \) denote \( \sum_{i=1}^{n} e_i \). We index tasks such that \( e_i \leq e_j \) holds if \( i \leq j \).

For conciseness, let \( T_{\text{lab}} \) denote \( \{T_1, T_2, \ldots, T_m\} \) with \( m \) tasks.

**Theorem 1.** A feasible traveling velocity \( v_{\text{lb}} \) for the charger yields a total charged energy \( E(v_{\text{lb}}) = \max \{ \sum_{k=1}^{n} e_k + (m-1) \cdot e_m, \sum_{k=1}^{n} e_k + (m-1) \cdot e_m \} \), which guarantees that
there exists a feasible task assignment.

Proof. We prove Theorem 1 by showing that our proposed LB algorithm guarantees to yield a feasible task assignment under \( V_{\text{act}} \). The LB algorithm is a 2-Stage process (described next) and the order of these 2 stages cannot be inverted.

Stage 1 of LB: Assign \( \{T_{m+1}, T_{m+2}, \ldots, T_n\} \) to sensor nodes in order. Before assigning \( T_{m+i} \) \((1 \leq i \leq n-m)\), we denote the total amount of remaining energy on all sensor nodes as \( E_r \) and it holds that:

\[
E_r = E(v_{jb}) - \sum_{j=m+1}^{n} e_j
\]

\[
= \max \{ \sum_{k=m+1}^{n} e_k + (m-1) \cdot e_n - \sum_{j=m+1}^{n} e_j, \sum_{k=1}^{n} e_k + (m-1) \cdot e_m - \sum_{j=m+1}^{n} e_j \} \tag{12}
\]

\[
\geq \sum_{k=m+1}^{n} e_k + (m-1) \cdot e_n - \sum_{j=m+1}^{n} e_j
\]

\[
\geq \sum_{k=m+1}^{n} e_k + (m-1) \cdot e_n - \sum_{j=m+1}^{n} e_j
\]

Since \( E_r \geq m \cdot e_{m+i} \), the average amount of remaining energy on each sensor node is no less than \( e_{m+i} \). Thus, we can always find a sensor node with an amount of remaining energy no less than \( e_{m+i} \). Then, we assign \( T_{m+i} \) to this sensor node. By applying this argument inductively on \( i \), we are able to find sensor nodes to which \( T_{m+1}, T_{m+2}, \ldots, T_n \) can be assigned. Stage 2 of LB: Assign tasks in \( T_{\text{sub}} \) to sensor nodes. After Stage 1, we denote the total amount of remaining energy on all sensors as \( E_{tr} \) and it holds that:

\[
E_{tr} = E(v_{lb}) - \sum_{j=m+1}^{n} e_j
\]

\[
= \max \{ \sum_{k=m+1}^{n} e_k + (m-1) \cdot e_n - \sum_{j=m+1}^{n} e_j, \sum_{k=1}^{n} e_k + (m-1) \cdot e_m - \sum_{j=m+1}^{n} e_j \} \tag{13}
\]

\[
\geq \sum_{k=m+1}^{n} e_k + (m-1) \cdot e_n - \sum_{j=m+1}^{n} e_j
\]

\[
\geq \sum_{k=m+1}^{n} e_k + (m-1) \cdot e_n - \sum_{j=m+1}^{n} e_j
\]

Since \( E_{tr} \geq \sum_{k=1}^{m-1} e_k + m \cdot e_m \), there exists at least one sensor node \( s_i \) that has a remaining energy greater than \( e_1 \). Thus, we can assign \( T_1 \) to \( s_i \). Similarly, if we assign tasks in \( T_{\text{sub}} \) in order, then after assigning \( T_{i-1} \) \((2 \leq i \leq m)\), the total amount of remaining energy on all sensor nodes is at least \( \sum_{k=1}^{m-1} e_i + m \cdot e_m - \sum_{j=1}^{i-1} e_j = \sum_{k=1}^{m} e_k + (m-1) \cdot e_m \). Thus, we can always find a sensor node \( s_i \) whose remaining energy is no less than \( e_m \). We can assign \( T_i \) to \( s_i \).

After completing this 2-Stage process, we have assigned all \( n \) tasks to sensor nodes with a total charged energy of \( E(v_{lb}) \). The theorem thus follows.

Intuitively, since \( \sum_{i=1}^{n} e_i \) denotes the total amount of energy required to execute the \( n \) tasks, Theorem 1 shows that the amount of redundant energy is related to the number of sensor nodes. Note that in order to guarantee a feasible task assignment, the above-mentioned 2-Stage LB algorithm only relies on the total amount of harvested energy on all sensor nodes. Thus, it can be executed online, which implies that sensor charging, task assignment, and task execution can be operated at the same time regardless of the amount of harvested energy on each individual sensor node. While the charger is traveling and charging the sensor nodes, tasks can be assigned to sensor nodes that harvest enough energy. All \( n \) tasks are guaranteed to be assigned after the 2-Stage process (i.e., when the total amount of harvested energy on all sensor nodes reaches \( E(v_{lb}) \)).

Example. We use an example to illustrate the 2-Stage task assignment, as shown in Fig. 4. In this figure, notation \( s_i(j) \) denotes the amount of remaining energy on sensor node \( s_i \) and the number shown in the task block denotes the amount of energy required by that task. For this system, the task set is \( \{T_1, T_2, T_3\} \) and the sensor set is \( \{s_1, s_2\} \). According to Theorem 1, \( E_o = \max \{e_{\text{sum}} + e_2, e_{\text{sum}} + (e_3 - e_1)\} = 13 \). Fig. 4(b) shows the amount of harvested energy increasing with time \( t \) for \( s_1 \) and \( s_2 \), when the charger travels with \( v_{lb} \) and finally the total amount of harvested energy on \( s_1 \) and \( s_2 \) is 13. Since the 2 stages in Theorem 1 cannot be inverted, we have to assign \( T_3 \) first to guarantee a feasible task assignment. In Fig. 4(b), we observe that the amount of harvested energy on \( s_2 \) is 5 at time \( t_1 \). Therefore, we assign \( T_3 \) to \( s_2 \) at \( t_1 \). After assigning \( T_3 \), the algorithm enters Stage 2, we start to assign \( T_{\text{sub}} \). At time \( t_1 \), the amount of harvested energy on \( s_1 \) is 4.5, which is larger than \( e_1 \). Thus we also assign \( T_1 \) to \( s_1 \) at \( t_1 \). Fig. 4(c) shows the system status after assigning \( T_3 \) and \( T_1 \). In Fig. 4(b), we can see that the total amount of harvested energy on \( s_1 \) is 5 at \( t_2 \). So after executing \( T_1 \), the amount of remaining energy is enough to execute \( T_2 \). \( T_2 \) can thus be assigned to \( s_1 \) at \( t_2 \). Fig. 4(d) shows the final task assignment.

Approximation Ratio of LB. The LB algorithm yields a feasible velocity \( v_{lb} \). We now derive the approximation ratio between \( v_{lb} \) and the maximum possible velocity \( v_{\text{max}} \).

According to Eq. (5), the total amount of harvested energy on all sensor nodes \( E \) is a monotonically decreasing function of the charger’s traveling velocity \( v \), therefore, the inverse function for \( v \) can be described as:

\[
v = E^{-1}(E_{\text{total}}), \tag{14}
\]

where \( E_{\text{total}} \) is the total amount of harvested energy on all sensors. \( v \) is also a monotonically decreasing function of \( E_{\text{total}} \).

Theorem 2.

\[
E^{-1}(E(v_{lb})) \leq \frac{v_{lb}}{v_{\text{max}}} \leq 1. \tag{15}
\]

Proof. Assuming when the charger travels with \( v_{\text{max}} \), the total amount of harvested energy on all sensor nodes is
Since \( E(v_{\text{max}}) \) must yield a feasible task assignment to accomplish all tasks, it is clear that \( E(v_{\text{max}}) \geq e_{\text{sum}} \) must hold (\( e_{\text{sum}} \) is defined in Def. 1. Since \( v \) is a monotonically decreasing function of \( E^{-1}(E_{\text{total}}) \), \( E^{-1}(E(v_{\text{max}})) \leq E^{-1}(e_{\text{sum}}) \). Therefore, \( v_{lb} = E^{-1}(E(v_{lb})) \leq E^{-1}(e_{sum}) \). Since \( v_{lb} \leq v_{\text{max}}, \frac{v_{lb}}{v_{\text{max}}} \leq 1 \). Hence, the theorem follows.

**Example.** Since \( v = E^{-1}(E_{\text{total}}) \) is an inverse function of Eq. (5), we can quantitatively calculate the approximation ratio \( E^{-1}(E(v_{lb})) \) of LB with given system parameters. Consider one of the settings used in our experiments for example system is given by \( 0 \leq v \leq 12 \)](lb)

\[ 0 \leq E_{\text{total}} \leq 100 \]  

Let \( v_{lb} \) denote \( E^{-1}(e_{sum}) \). According to Theorem 2, \( v_{lb} \leq v_{\text{max}} \leq v_{lb} \). In the following sections, we introduce a binary search algorithm to find \( v_{\text{max}} \).

### 4 Practical Binary Compression

In the previous section, we have shown that the LB algorithm yields a lower bound \( v_{lb} \) (Theorem 1) and an upper bound \( v_{ub} \) (Theorem 2) on the feasible velocity for the charger. This implies that for any velocity \( \leq v_{lb} \), we can always find a corresponding feasible task assignment, and for any velocity \( > v_{ub} \), we can not find a feasible task assignment. Thus, the optimal velocity \( v_{\text{max}} \) falls in the interval \([v_{lb}, v_{ub}]\). According to Theorem 2, \([v_{lb}, v_{ub}]\) is expected to yield a rather limited solution space in most cases. In this section, we present novel techniques that find \( v_{ub} \) by compressing the limited solution space of \([v_{lb}, v_{ub}]\).

#### 4.1 Task Scheduling for A Given Energy Distribution

According to the mobile charging model in Sec.2, we know that the amount of harvested energy on each sensor node only depends on the charger’s traveling speed \( v \). In other words, each \( v \in [v_{lb}, v_{ub}] \) corresponds to an energy distribution on these sensor nodes. Our key observation is that since the amount of harvested energy on each sensor node decreases monotonously when the charger’s velocity increases, we can apply the binary search algorithm to reduce the searching space efficiently.

In the rest of this section, we investigate how to derive a feasible task assignment under a given \( v \) using the binary search algorithm. For any given \( v \), the amount of harvested energy on each sensor node is fixed. Our objective is thus changed to determine whether all tasks can be completed by sensor nodes with fixed energy distributions.

The Task Assignment Problem (TAP) can be formulated by the following ILP:

\[
\text{max} \quad z = \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \quad (16)
\]

subject to:

\[
\sum_{j=1}^{k} x_{ij} \leq 1 \quad (17)
\]

\[
\sum_{i=1}^{n} x_{ij} \times e_{i} \leq E_{j}(v) \quad (18)
\]

\[
x = \{0, 1\} \quad (19)
\]

where \( z \) denotes the number of tasks executed by the rechargeable sensor nodes, where the charger travels with speed \( v \). If \( z = n \), then the corresponding \( v \) guarantees that all \( n \) tasks can be assigned.

### 4.2 Multiple Knapsack Problem

According to the above formulation, we observe that TAP has many similarities with the Multiple Knapsack Problem (MKP). MKP can be defined as follows: given a set of \( n \) items and a set of \( m \) knapsacks (\( m \leq n \)) such that each item \( i \) has a profit \( p_{i} \) and size \( s_{i} \), and each knapsack \( j \) has a capacity \( c_{j} \), how to select \( m \) disjoint subsets of items such that the total profit of the selected items is maximized while satisfying the constraint that each subset can only be assigned to a knapsack whose capacity is no less than the total size of items in the subset.

For TAP, the amount of charged energy to each sensor node corresponds to the capacity of each knapsack; the consumed energy to accomplish each task corresponds to the size of each item. Our objective is to assign tasks to these sensor nodes as many as possible while guaranteeing that all sensor nodes have enough charged energy to execute the assigned tasks.

Therefore, TAP is a special case of MKP, where the profit for each item equals 1. The solutions proposed to tackle MKP can also be used to solve TAP.
4.3 Our Proposed PTAS Algorithm.

MKP is a nature extension of the single knapsack problem, which has been studied extensively [12, 19, 3]. MKP is NP-hard [12]. Many solutions were proposed to seek the optimal profit for MKP [10] [13] [3]. In [3], a polynomial time greedy scheme is designed, which yields a $(2 + \epsilon)$-approximation algorithm. The time complexity of this algorithm is $O(m\log(1/\epsilon) + m/\epsilon^4)$, using the results of Lawler [16] for the knapsack problem.

Our proposed algorithm is built upon this greedy scheme [3], while preserving the theoretical property of the $(2 + \epsilon)$ approximation ratio. For brevity, we denote this greedy scheme as “GREEDY”. Under GREEDY, the sensor nodes are considered as independent components. Each sensor is considered in turn and tasks are assigned to each considered sensor by applying the PTAS [26] designed for the single knapsack problem on the remaining tasks. Thus, GREEDY seeks to maximize the number of tasks assigned to each individual sensor. However, our key observation herein is that maximizing the number of tasks assigned to each sensor may not maximize the total number of tasks assigned to all sensor nodes. Thus, based upon GREEDY, we present an improved algorithm by globally exploring the correlations among sensor nodes in order to maximize the total number of assigned tasks, as described next.

Assuming a feasible solution given by [3] is $F_i (i = 1, 2, \ldots, m)$, where $F_i$ is the set of tasks assigned to $s_i$. First we give necessary definitions.

**Definition 2.** Let $P = \bigcup_{i=1}^{m} F_i$ and $R = T - P$, where $T$ is the task set $\{T_1, T_2, \ldots, T_n\}$, $P$ is the set of tasks that have already been assigned, and $R$ is the set of remaining tasks that cannot be assigned under GREEDY. Let $z_i$ denote the total number of assigned tasks on $s_i$ and $c_i$ denote the current reminder energy on $s_i$. Let $g_j$ denote the index of the sensor where $T_j$ is assigned.

Our algorithm applies the following three optimization techniques on the task assignment obtained by GREEDY. As demonstrated by the experiments presented later, these optimization techniques are effective in practice for increasing the number of assigned tasks. After executing the GREEDY algorithm, no sensor has enough remaining energy to execute tasks in $R$ in the resulting task assignment. Thus, the idea behind these optimization techniques is to increase the remaining energy on sensor nodes in order to allow more tasks to be assigned. Note that the order of applying these optimization techniques can be arbitrarily determined. Although different orderings may yield different performance, they are incomparable in general and the resulting performance heavily depends on the task and system parameters.

**Task Swapping.** This optimization technique increases the remaining energy on sensor nodes by swapping the assigned tasks with the unassigned tasks. The swapping process can be described as follows: for each $T_i (T_i \in R)$, search $P$ to find the minimum $k$ which satisfies $e_i < e_k$. Then swap $T_i$ for $T_k$ such that the remaining energy is increased on sensor node $g_k$ while the total number of assigned tasks does not decrease. This process ends until $R = \emptyset$, or for each $T_i (T_i \in R)$, we cannot find a $T_k$ ($T_k \in P$) that satisfies $e_i < e_k$.

Fig. 5 presents an example illustrating this swapping process. After executing the Greedy algorithm, task $T_1$ is not assigned, and $e_1 = 4$, as illustrated in Fig. 5(a). $T_2$ ($e_2 = 6$) is assigned on $s_1$. If we swap $T_2$ for $T_1$ on $s_1$, the total number of assigned tasks on sensor nodes is not decreased and the amount of remaining energy on $s_1$ is increased. So we swap $T_2$ for $T_1$. Fig. 5(b) shows the task assignment after swapping. The complexity of this process is $O(n^2)$.

**Task Moving.** This optimization technique seeks to increase the remaining energy on sensor nodes by moving tasks from one sensor node to another. The moving process can be described as follows: for each sensor node $s_i (1 \leq i \leq m)$, search $P$ to find the minimum $k$ which satisfies $c_i \leq c_k$ and $c_i < c_{g_k} - e_k$. Then move $T_k$ from $s_{g_k}$ to $s_i$ such that the remaining energy on $s_{g_k}$ is increased up to $c_i$. The process ends until all sensor nodes are tested.

As shown in Fig. 5(b), after swapping, if $T_4$ (with $e_4 = 3$) is moved from $s_2$ to $s_1$, then the amount of remaining energy on $s_2$ becomes 5, which is larger than $c_1 = 4$. Therefore, we move $T_4$ to $s_1$. Fig. 5(c) shows the assignment after the moving step. The complexity of this moving step is $O(m \times n^2)$.

**Task Exchanging.** This optimization technique increases the amount of remaining energy on sensor nodes by exchanging the assigned tasks on different sensor nodes. For each $T_i (T_i \in P)$, we exchange $T_i$ with $T_k$ ($T_k \in P$) if any task $T_j (T_j \in R)$ can be assigned to sensor $g_i$ or sensor $g_k$ after this exchange process. This exchanging step is executed on all pairs of tasks.

An illustrating example is shown in Fig. 5. After the moving step (Fig. 5(c)), the amount of remaining energy on $s_2$ is 5. If we exchange $T_3$ (with $e_3 = 5$) and $T_1$ (with $e_1 = 4$), the

![Figure 5: Example task assignments illustrating the three optimization techniques.](image)
amount of remaining energy on $s_2$ becomes 6, which allows $T_2$ to be assigned to $s_2$. We thus exchange $T_1$ and $T_2$, and the resulting task assignment is shown in Fig. 5(d). We then assign $T_3 \in R$ to $s_2$. Fig. 5(d)(e) shows the final task assignment after applying all the three optimization techniques. As seen, applying these techniques enables all the tasks in this example system to be successfully assigned. The complexity of the exchanging process is also $O(n^2)$. Note that this process ends when the sum of the remaining energy on any two sensors is smaller than the energy required by the unassigned task that consumes the minimum energy among all unassigned tasks.

5 Evaluation

To evaluate our proposed task assignment algorithms, we have conducted extensive experiments assuming different network settings and different trajectories for the mobile charger.

5.1 Experiment Setup

We simulated a network consisting of 500 wireless rechargeable nodes randomly deployed in an area of $100m \times 100m$ two-dimensional square. Regarding the charging model as given in Eq. (1), we set $G = G_n \big(\frac{\lambda}{4\pi}\big) = 36$ and $\beta = 30$. A total number of 5000 tasks are generated for each experiment. We simulate an application scenario where 9 categories of tasks need to be performed, as shown in Table. 1. The power consumption values listed in this table are obtained from the corresponding sensors’ data sheet. The executing time of a task is randomly selected from $[1s, 10s]$.

5.2 Baseline Settings

Since the literature on investigating the task assignment problem in RWSNs is limited, we set the greedy algorithm [3] discussed in Sec.4 as the baseline for comparison purposes. We also compare our proposed approach with two other methods: the lower bound $v_{act}$ and the upper bound $v_{max}$ as discussed in Sec.3.

Results. We investigate the performance of our algorithm with different system parameters including the number of wireless rechargeable nodes, the number of tasks and different kinds of traveling trajectories.

5.3 Impact of Sensor Node Number

We explore the scalability of our design and investigate the impact of the number of nodes on the feasible charger velocity, as shown in Fig.6(a). In the experiment, the number of sensor nodes is varied from 100 to 1000. We can see that ACT yields near optimal velocity, which is close to the upper bound $v_{max}$. Another observation is that both ACT and the greedy algorithms yield better performance when more sensors are involved in the system. This is because the density of sensor nodes in the square increases when the number of sensors increases, which implies that the charger may charge more sensor nodes at the same time and the total harvested energy thus increases. This further implies that the charger can travel with a faster velocity. However, when the number of sensor nodes is more than 900, ACT and the greedy algorithm achieve similar performance. Also the charger’s velocity stops increasing. The reason is because the sensor on which the assigned tasks require the maximum amount of energy gives an upper bound of the charger’s velocity. Thus, whenever this sensor has been charged with enough energy under a certain velocity, the velocity cannot increase any further. Fig.6(b) shows the total amount of charged energy under difference approaches. As seen in this figure, ACT also performs quite close to the upper bound $v_{max}$ under all cases. Fig. 6(c) shows the standard deviation of remaining energy on sensors with the sensor numbers varying from 100 to 500. Our algorithm yields smaller deviations under all cases, which implies that the amount of remaining energy on sensors under our algorithm is more balanced.

5.4 Impact of Task Number

Fig. 7 shows the experimental results investigating the impact due to different number of tasks. In the experiment the number of tasks is varied from 2500 to 5500. Intuitively, with a larger number of tasks and a fixed number of sensors, it is more likely that sensors need to harvest more energy in order to execute all of the tasks. This in turn implies that the charger may need to decrease its speed. This observation is verified in Fig. 7(a) and Fig. 7(b), where a clear decreasing trend for the charger’s velocity and a clear increasing trend for the total amount of charged energy on all sensors can be
observed, respectively. As seen in Fig. 7(a), ACT achieves quite close performance to the upper bound, yielding a faster speed than the greedy algorithm. The corresponding results on the total amount of harvested energy under the four tested algorithms are shown in Fig. 7(b). Fig. 7(c) shows the standard deviation of the amount of remaining energy on sensors with the task number varying from 2500 to 5500 under ACT and the greedy algorithm. ACT gets smaller deviations under all cases, which means that the amount of remaining energy on sensor nodes for ACT is more balanced, regardless of the number of tasks.

5.5 Impact of the Task Average Energy

In this experiment, we study the impact due to different distributions of the energy consumption of tasks by varying the average energy consumption of tasks from 0.05J to 2J, and the results are shown in Fig. 8. In Fig. 8(a), we can see that the charger’s velocity decreases as the average energy consumption of tasks increases. This is because with a larger average energy consumption, tasks need more energy, thus the charger needs to travel with a lower speed to charge more energy on these sensors. An increasing trend of the amount of harvested energy on all sensor nodes is observed in Fig. 8(b) when the average energy consumption of tasks increases.

5.6 System Insight

In order to reveal the insight about why our proposed algorithm ACT can significantly reduce charging delay, we conduct an experiment to observe the amount of remaining energy on each sensor node while the harvested energy can accomplish all tasks with ACT. For comparison, we also show the results under the greedy algorithm [26] with the same system settings. Intuitively, an algorithm becomes more efficient when the amount of remaining energy on each sensor node is small after accomplishing all the tasks (all the sensor nodes use up their harvested energy is the best case). Fig. 9 shows the cumulative distribution of sensor node number when the amount of remaining energy varies from 0 J to 20 J. The cumulative ratio of sensor number associated with the remaining energy J represents the number of sensors with remaining energy less than J. As seen in this figure, for each remaining energy value, ACT is able to achieve a higher cumulative ratio of sensor number than the greedy algorithm. This implies that ACT is able to better utilize the harvest energy, leaving a smaller number of sensors with large remaining energy after assigning all tasks. Given that the same set of tasks is tested under both algorithm, the greedy algorithm fails to assign as many tasks as ACT because it yields a greater remaining energy.

6 Related Work

Recently, the research problem of improving charging efficiency in wireless rechargeable sensor networks has received much attention [15, 28, 31, 4, 23]. Many pioneer work has focused on hardware design to improve charging efficiency [15, 28]. Kurs et al. [15] improve the overall output efficiency of charging multiple devices. Sample et al. [28] design a scheme of analog circuitry for WISP node to obtain
an efficient conversion of the incoming RF energy. From the view of networks, some recent work has investigated the network charging coordination to improve charging performance. He et al. [9] consider the static reader deployment in a wireless rechargeable sensor networks so that the nodes can harvest enough energy for continuous operation. In [30], Xie et al. consider the joint design of traveling path of mobile wireless charging vehicle (WCV), flow routing among the network, and charging time of WCV at each stopping point, and propose a near-optimal solution with guaranteed accuracy. The authors in [21] build a proof-of-concept prototype of wireless charging system for sensor networks and conduct experiments to evaluate its feasibility and performance in small-scale networks. Bin et al. [29] investigate how to minimize charging cost by reducing energy consumption rate and improving recharging efficiency.

To the best of our knowledge, none of the prior work considers the task assignment problem along with the charging delay minimization problem. We consider the problem of finding the maximum charging velocity and a corresponding feasible task assignment such that each sensor has enough charged energy to execute the assigned tasks. This distinguishes this paper from the prior work.

7 Conclusion

In this paper, we study a general scenario of randomly deployed rechargeable wireless sensor network, where a charger travels along a fixed trajectory to charge energy on the sensor nodes that need to execute a set of tasks. Our objective is to find the maximum velocity for the mobile charger while ensuring a corresponding feasible task assignment. We first propose an online task assignment algorithm LB that yields a quantifiable lower bound on the charging velocity while guaranteeing a feasible assignment. LB further enables us to transform the considered task assignment problem into a variation of the classical multiple knapsack problem. We then present an improved task assignment algorithm, namely ACT, that built upon an existing greedy algorithm designed for the original knapsack problem. The effectiveness of our proposed algorithm has been demonstrated by extensive experiments. In the future work, it would be interesting to explore the proposed techniques to address the problem of minimizing the overall system delay including both the charging delay and the task execution delay.

8 Acknowledgments

This work was supported by NSF grants OISE-1427824, CNS-1527727, CNS-1456656, CNS-1526769, A*Star SERC project (grant number 142 02 00043) and NSFC grant 61550110244.

9 References


To the best of our knowledge, none of the prior work considers the task assignment problem along with the charging delay minimization problem. We consider the problem of finding the maximum charging velocity and a corresponding feasible task assignment such that each sensor has enough charged energy to execute the assigned tasks. This distinguishes this paper from the prior work.

7 Conclusion

In this paper, we study a general scenario of randomly deployed rechargeable wireless sensor network, where a charger travels along a fixed trajectory to charge energy on the sensor nodes that need to execute a set of tasks. Our objective is to find the maximum velocity for the mobile charger while ensuring a corresponding feasible task assignment. We first propose an online task assignment algorithm LB that yields a quantifiable lower bound on the charging velocity while guaranteeing a feasible assignment. LB further enables us to transform the considered task assignment problem into a variation of the classical multiple knapsack problem. We then present an improved task assignment algorithm, namely ACT, that built upon an existing greedy algorithm designed for the original knapsack problem. The effectiveness of our proposed algorithm has been demonstrated by extensive experiments. In the future work, it would be interesting to explore the proposed techniques to address the problem of minimizing the overall system delay including both the charging delay and the task execution delay.

8 Acknowledgments

This work was supported by NSF grants OISE-1427824, CNS-1527727, CNS-1456656, CNS-1526769, A*Star SERC project (grant number 142 02 00043) and NSFC grant 61550110244.

9 References