

# Graph Coloring Inspired Optimal Algorithm for Wireless Energy Redistribution in WSNs

Author1, Author2, Author3

## Abstract

Wireless energy redistribution using Wireless Power Transfer (WPT) is of key importance for enhancing Wireless Sensor Networks (WSNs). In this paper, we investigated the WPT based energy redistribution (WPTERD) problem in WSNs, which is to redistribute the energy among nodes to meet the energy demands of nodes meanwhile optimal in energy efficiency and time. We propose a two-step approach by decoupling the energy-time joint optimization into two problems named WPTERD-Egy and WPTERD-Time, which focus exclusively on the optimization in energy and time, respectively. We transform WPTERD-Egy as a linear programming (LP) problem by exploiting its properties, and obtain optimal time-lengths of the nodes' energy transmissions leading to minimum energy lost. The core of WPTERD-Time is the Energy Transmission Task Scheduling (ETTS) problem. Proving that ETTS and WPTERD-Time are both NP-hard, we propose Least neighbor-set-weight Last Sequential Scheduling (LNSWL-SS) algorithm on the inspiration of a graph coloring algorithm, and using which we solve WPTERD-Time to obtain an energy transmission schedule with approximately minimum makespan. Finally Graph Coloring inspired Egy-Time Decoupling (GC-EgyTimeD) algorithm is constructed on LNSWL-SS for the WPTERD problem. Some approximation ratios of LNSWL-SS are obtained. Simulations show that GC-EgyTimeD return schedules with minimum energy lost and approximately minimum makespan. By exploiting concurrent energy transmission opportunities, GC-EgyTimeD reduces makespan by about more than 65% when compared with a schedule without concurrent transmissions.

## Index Terms

Wireless energy redistribution problem, wireless power transfer, task scheduling problem, wireless charging, graph coloring problem.

The authors are with the Department of

Manuscript received XXX, XX, 2018; revised XXX, XX, 2018.

## I. INTRODUCTION

Wireless Power Transfer (WPT) is the transmission of electric energy from a power source to a receiver without a conductor. With recent advances in WPT, it is possible to charge sensors in a relative long distance ( $>10\text{m}$  away)[1]. It has been validated that a sensor could harvest radio power of  $6\text{mW}$  over a distance of 12 meters from a Wireless Charger (WC) transmitting with  $4\text{W}$  power. The received radio power is  $20\text{mW}$  and the transition efficiency is 30 percent [2]. The breakthroughs in WPT technology [3], [4] open up a new dimension to prolonging the lifetime of wireless sensor networks (WSNs). Many research efforts have been devoted to exploiting WPT to enhance WSNs [5]-[12]. Some works focus on using WCs to power the nodes in WSNs using WPT. Due to the limited number of WCs, some nodes may not able to obtain power from the WCs directly. This is true even if there are mobile WCs because of reachability restrictions or temporal failure. Hence, WPT based Energy ReDistribution (WPTERD) is a core building block for supporting perpetual WSNs.

Some recent efforts have been devoted to the application of energy exchanging among nodes in peer-to-peer mode when nodes coming into vicinity of each other during moving, so as to make the nodes' remanent energy more balanced. However, the intrinsic broadcasting feature of energy transmitting signals is not exploited, and hence leads to lower energy efficiency.

In this paper, we focus on the WPTERD problem: given a WSN consisting of nodes equipped with wireless power transceivers and limited energy storages with some initial energy, each node has its energy expectation, the task is to redistribute the energy in the WSN among all nodes through WPT, assuring that each node's final energy after the redistribution do not below its corresponding energy expectation, meanwhile the energy loss is minimized and the time-length (*i.e.*, makespan) of the energy redistribution process is minimized.

We solve the WPTERD problem using a two-step approach which decouples the joint energy-time optimization into to two sub-problems named WPTERD-Egy and WPTERD-Time, as outlined in Fig. 1. The two problems focus exclusively on the optimization in energy and time, respectively. In the first step, basing on two proved properties resulted from the widely adopted *energy harvesting additive assumption*, the WPTERD-Egy problem is formalized as a Linear Programming (LP) problem, and can thus be solved to obtain the optimal time-lengths of the nodes' energy transmissions. The remanent of the WPTERD problem thus becomes the WPTERD-Time problem, which is a task scheduling problem of the nodes' energy transmis-

sions with some restrictions of not violating some energy limits. At the core of the WPTERD problem is a Energy Transmission Task Scheduling (ETTS) problem, where there are no energy-limit restrictions. We prove that both WPTERD-Time and ETTS are NP-hard, and propose an algorithm named Least neighbor-set-weight Last Sequential Scheduling (LNSWL-SS) for solving the ETTS problem, inspired by a well-known graph coloring algorithm. We build our algorithm Graph Coloring inspired Egy-Time Decoupling (GC-EgyTimeD) basing on LNSWL-SS and solutions to WPTERD-Egy and WPTERD-Time. Numerical simulations illustrate the effectiveness and efficiency of GC-EgyTimeD, which return schedules with minimum energy loss and approximately minimum makespan. Main abbreviations are listed in Table I for reference.

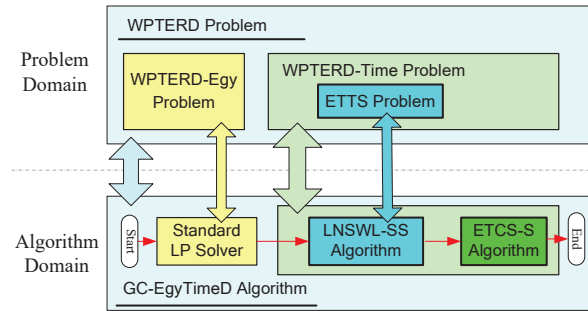


Fig. 1. Outline of the GC-EgyTimeD algorithm for the WPTERD problem.

Our main contributions in this paper are as follows:

- We propose the GC-EgyTimeD algorithm to solve the WPTERD problem efficiently and approximately by dividing it into WPTERD-Egy and WPTERD-Time and solve them successively. This approach exploits the broadcast nature of wireless signals well, and implicitly realizes the multi-hop energy transmission.
- For the WPTERD-Egy problem, we prove that its optimal solutions must be disjoint (neighboring nodes should not transmit energy simultaneously) and no energy limit violations. These properties enable us to formulate WPTERD-Egy as a Linear Programming (LP) problem and thus can be solved easily.
- For the WPTERD-time problem and the ETTS problem, we prove that they are both NP-hard by reduction from the Graph Coloring (GC) problem.
- We propose the LNSWL-SS algorithm for the ETTS problem on the inspiration of the well-known Least-degree-last algorithm for the GC problem. We also prove that, for equal-radius WSNs, the approximation ratios of LNSWL-SS for 2D WSNs and 3D WSNs are 3

and 12, respectively. The approximation ratios are respectively 6 and 24 when applied to non-equal-radius WSNs.

TABLE I  
ABBREVIATIONS.

Abbreviations	Whole words/sentences and meanings
WPT	Wireless Power Transfer;
WPTERD	Wireless Power Transfer based Energy Re-Distribution problem;
WPTERD-Egy	WPTERD problem with only the minimum energy lost objective;
WPTERD-Time	WPTERD problem with only the minimum makespan objective;
ETTS	Energy Transmission Task Scheduling problem, which is the core of WPTERD-Time but without the restrictions caused by energy limits;
LNSWL-SS	Least neighbor-set-weight Last Sequential Scheduling algorithm, which solves ETTS;
ETCS-S	Energy Transmission Concurrent Set Scheduling algorithm, which together with LNSWL-SS solves WPTERD-Time;
GC-EgyTimeD	Graph Coloring inspired Energy-Time Decoupling algorithm, which solves WPTERD;
NSW	Neighbor-Set-Weight, which is the sum of the weights of a node's neighbors as well as the node itself;
EUL,ELL	Energy-Upper-Limit, Energy-Lower-Limit;

The rest of the paper is organized as follows. Related work is discussed in Section II. The models of the problem are provided in Section III. The WPTERD-Egy problem is analyzed and solved in Section IV. In Section V, the WPTERD-Time problem is analyzed and the NP-hard property of ETTS and WPTERD-Time are proved. In Section VI, we provide LNSWL-SS and prove its approximation ratios. GC-EgyTimeD is provided in Section VII. Simulation results are provided and discussed in Section VIII. Finally a conclude is made in Section IX.

## II. RELATED WORK

Existing works using WPT most related to our work fall into two topics: (1) charging WSNs with fixed WCs; (2) energy redistribution among nodes in WSNs.

### A. Charging WSNs with fixed WCs

In [5], focusing on WSNs containing a fixed wireless charger (WC) as the only energy source for the network, the problem of distributing the energy injected from the WC among all nodes through multi-hop WPT was investigated. In the approach there, the energy transfers are modeled and formulated as multi-hop energy flows, and algorithms were proposed.

WSNs with multiple WCs and multi-hop power transfer technique were focused in [6]. In this work, considering the energy demands of the nodes, the energy loss that occurs during an energy transmission, and the energy capacity limits of the WCs, the problem of determining the minimum number of WCs needed for perpetual operation of the WSN was investigated. The approach there solves the problem in two steps. In the first step, for each possible location of the WCs (WCs are embedded in nodes), a shortest-path tree rooted at this location covering the network is constructed using the Dijkstra's algorithm. Then in the second step, the problem is transformed to a Mixed Integer Linear Programming (MILP) problem making use of the trees and solved using optimization software packages.

The works in [1], [7] also devoted to WSNs with multiple WCs, but with an emphasis on exploiting the possibility of multiple WCs charging a single node simultaneously. A more complicated energy harvesting model considering the nonlinear superposition charging effect of simultaneously arrived energy signals was proposed there. Basing on this model, the joint charging utilities for all possible set of the WCs are pre-calculated and then be used to solve the Concurrent Charging Scheduling Problem (CCSP), whose objective is to find a schedule for the WCs so as to minimize the time spent on providing each sensor node with at least  $E$  energy. The authors showed that CCSP is NP-hard, and proposed a greedy algorithm. However, only one-hop WPT is considered in these works.

Multi-hop power transmissions are quite different from multi-hop data transmissions. Different data transmissions are not equivalent since they convey different data. For power transmissions, however, energy from any source are equivalent, we need not explicitly construct paths and restrict energy flows along a certain path. In some sense, opportunistic routing in the wireless

network routing realm is more suitable for energy transmission. Hence, finding other mechanisms of multi-hop WPT for charging WSNs is of importance.

### B. Energy Redistribution among Nodes

Some series of recent works [8], [9], [10], [11], [12] focus on a problem for mobile social and sensor networks consisting of human-carried mobile devices. In these works, energy redistribution is realized using peer-to-peer energy exchange.

Firstly in [8], the authors investigated the problem of selecting a subset of nodes to charge the others such that all nodes can continue normal operations without battery depletion. In [9], the authors investigated the problem of efficiently reaching an energy balance among nodes distributively though peer-to-peer energy exchange. The problem was investigated in cases of two different assumptions: loss-less power exchange and lossy power exchange. Three energy exchange protocols were designed, analyzed and evaluated there. Then the problem was extended to the weighted version in [10]. The uniform balance version and the weighted balance version of the peer-to-peer energy redistribution problem among star-backbone nodes in WSNs were investigated in [11] and [12], respectively.

In these works, energy exchange only happens when two nodes coming into vicinity of each other because of the moving of the person carrying the node. As a consequence, their analyses and algorithms do not exploit the intrinsic broadcasting nature of wireless signals as well as the ability of WPT's transferring energy over a distance. Ignoring the possibility of multiple nodes simultaneously harvesting energy from a single node's energy transmitting signals inevitably leads to energy low-efficiency.

## III. SYSTEM MODEL

### A. Problem Model

We consider a WSN consisting of  $n$  static nodes  $U = \{u_1, u_2, \dots, u_n\}$ , where node  $u_i$  is positioned at  $(x(i), y(i), z(i))$ . An example network is shown in Fig.2. The nodes are equipped with energy storages with limited capacity. Besides the traditional wireless transmitter/receivers (transceivers) for data communication, these nodes are all equipped with wireless energy transceivers dedicated for transmitting/harvesting energy. For each node  $u_i$ ,  $i \in \mathcal{N}(n)$ , there is some parameters: beginning energy  $e_B(i)$ , final energy  $e_F(i)$ , energy expectation  $e_E(i)$ , energy upper limit  $e_U(i)$ , and energy lower limit  $e_L(i)$ .  $e_B(i)$  represents node  $u_i$ 's initial energy amount at the

beginning time;  $e_F(i)$  represents  $u_i$ 's final energy after the energy redistribution process;  $e_E(i)$  represents  $u_i$ 's energy expectation.  $e_U(i)$  represents the capacity limit of  $u_i$ 's energy storage. It is required that node  $u_i$ 's instantaneous energy  $e_E(i, t)$  at time  $t$  should always be in range  $[e_L(i), e_U(i)]$ ,  $i \in \mathcal{N}(n)$ ,  $t \in [0, +\infty]$ .

We are required to redistribute the energy among the nodes by using WPT, with the objective that node  $u_i$ 's final energy  $e_F(i)$  at the finish of the redistribution process should not below its expectation  $e_E(i)$ , i.e.,  $e_F(i) \geq e_E(i)$ ,  $i \in \mathcal{N}(n)$ . We express this objective as  $e_F \succeq e_E$  for short. To facilitate simulation, we further assume  $e_L \prec e_B \prec e_U$ , and assume  $e_E(i)$  take values in range  $[e_{E,L}(i), e_{E,U}(i)]$ , where we have  $e_L \prec e_{E,L} \preceq e_E \preceq e_{E,U} \prec e_U$ . For convenient,  $e_U$  is sometimes called as list or vector when no confusion, which also applies to other similar symbols. The symbols are listed in Table II for easy reference.

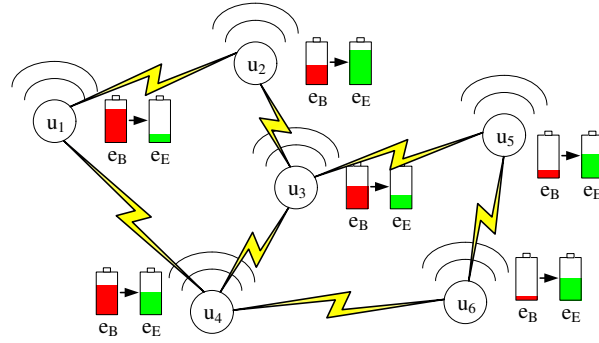


Fig. 2. An example network with WPT based energy redistribution.

Assume node  $u_i$  always transmits energy with a constant power  $p(i)$ . When  $u_i$  is transmitting energy, the energy power harvested by  $u_j$  from  $u_i$ 's signal is expressed as  $p_H(j) = c(j, i) * p(i)$ , where the energy harvesting coefficient  $c(i, j)$  embodies the effects of many factors such as the distance between nodes, environment, hardware restrictions. Energy harvesting coefficients are always non-negative. If  $c(i, j) + c(j, i) > 0$ , then we say that  $u_i$  and  $u_j$  are neighbors. Furthermore, because of the energy conservation principle in practice, energy loss is inevitable during the wireless energy redistribution process, hence we assume  $\sum_{j \neq i, j \in \mathcal{N}(n)} c(j, i) < 1$ ,  $i \in \mathcal{N}(n)$ . For completeness, we let  $c(i, i) = -1$ ,  $i \in \mathcal{N}(n)$ . The energy harvesting coefficients are assumed to be constant. We collect all energy harvesting coefficients into a matrix  $\mathbf{C} = [c(i, j)]_{i, j \in \mathcal{N}(n)}$ .

Multiple simultaneous energy transmissions encountered at a node are assumed to be additive in energy harvesting applications, which is the *energy harvesting additive assumption* widely

TABLE II  
SYMBOLS AND MEANINGS.

Notation	Meaning
$\mathcal{N}(n)$	The set of positive integers $\{1, 2, \dots, n\}$ ;
$p, \mathbf{p}$	The list/vector of the nodes' energy transmission power, where node $u_i$ 's energy transmission power is $p(i)$ ;
$[\cdot]^T$	Transpose operation of the input matrix or vector;
$\mathbf{C}$	The matrix of the energy harvesting coefficients $[c(i, j)]_{i, j \in \mathcal{N}(n)}$ , $c(i, j)$ is the energy harvesting coefficient for node $u_i$ receives energy from $u_j$ ;
$\mathbf{t}$	The vector of the nodes' energy transmission time-lengths $\mathbf{t} \stackrel{\text{def}}{=} [t(1), t(2), \dots, t(n)]^T = [t_1, t_2, \dots, t_n]^T$ ;
$e_B, \mathbf{e}_B$	The list/vector of the nodes' beginning energy $\mathbf{e}_B \stackrel{\text{def}}{=} [e_B(1), e_B(2), \dots, e_B(n)]^T$ ;
$e_F, \mathbf{e}_F$	The list/vector of the nodes' final energy $\mathbf{e}_F \stackrel{\text{def}}{=} [e_F(1), e_F(2), \dots, e_F(n)]^T$ ;
$e_U, \mathbf{e}_U$	The list/vector of the nodes' energy upper limits $\mathbf{e}_U \stackrel{\text{def}}{=} [e_U(1), e_U(2), \dots, e_U(n)]^T$ ;
$e_L, \mathbf{e}_L$	The list/vector of the nodes' energy lower limits $\mathbf{e}_L \stackrel{\text{def}}{=} [e_L(1), e_L(2), \dots, e_L(n)]^T$ ;
$e_E, \mathbf{e}_E$	The list/vector of the nodes' energy expectations $\mathbf{e}_E \stackrel{\text{def}}{=} [e_E(1), e_E(2), \dots, e_E(n)]^T$ ;
$e_{E,L}, \mathbf{e}_{E,L}$	The list/vector of the nodes' energy expectation lower limits $\mathbf{e}_{E,L} \stackrel{\text{def}}{=} [e_{E,L}(1), e_{E,L}(2), \dots, e_{E,L}(n)]^T$ ;
$e_{E,U}, \mathbf{e}_{E,U}$	The list/vector of the nodes' energy expectation upper limits $\mathbf{e}_{E,U} \stackrel{\text{def}}{=} [e_{E,U}(1), e_{E,U}(2), \dots, e_{E,U}(n)]^T$ ;
$\mathbf{1}$ or $\mathbf{0}$	Proper size column vectors with elements all one or all zero, respectively;
$\mathbf{1}_{\text{condi}}$	The indication function of condition <i>condi</i> , which has value 1 when <i>condi</i> is true, and 0 otherwise;
$w(G)$	The sum of the weights of the nodes in graph/set $G$ ;
$N(u_i, G)$	The set of neighbors of node $u_i$ in graph $G$ , two nodes $u_i$ and $u_j$ are neighbors of each other if $c(i, j) + c(j, i) > 0$ ;
$w_{\text{NSW}}(v, H)$	$w_{\text{NSW}}(v, H) \stackrel{\text{def}}{=} \sum_{u \in \{N(v, H), v\}} w(u)$ , which is the sum of the weights of node $v$ 's neighbors in graph $H$ together with node $v$ itself;



adopted in the literature[14], [15], [16]. To be specific, if a set of nodes  $\{u_j|j \in U_s\}$  transmit energy with power  $\{p_j|j \in U_s\}$  to another node  $u_j$  simultaneously, then the energy power harvested by  $u_j$  is  $p_H(j) = \sum_{k \in U_s} c(j, k) * p(k)$ . Harvested energy should be stored in the energy storage for later use. Excessive energy harvested by a node is lost when its energy storage is full. If some harvested energy is lost at some nodes, we say that there are *Energy Upper Limit (EUL) violations*. If a node's instantaneous energy becomes lower than its energy lower limit, we say that there are *Energy Lower Limit (ELL) violations*. EUL violations and ELL violations are both called as energy violations.

To meet the requirement of  $e_F \succeq e_E$ , we are required to find an energy transmission schedule  $s_c \triangleq \{s(1), s(2), \dots\}$  where  $s(i) \stackrel{\text{def}}{=} (u_s(i), t_b(i), t_e(i))$  represents a schedule item, which means to let node  $u_s(i) \in U$  transmit energy in time period  $[t_b(i), t_e(i)]$ .

Given a schedule, if each node can only transmit energy in one continuous time period, then this schedule is *non-preemptive*, otherwise it is *preemptive*. We mainly consider preemptive schedules in this paper. A schedule is valid if we have  $e_L \preceq e_E(\cdot, t) \preceq e_U$  (i.e., no energy violations) during performing the schedule and  $e_F \succeq e_E$  after performing the schedule.

A valid schedule with *maximum total final energy*  $E_C(s_c) = \sum_{i=1}^n e_F(i)$  is called an *optimal schedule*. It is obvious that maximizing total final energy is equivalent to *minimizing energy loss*, we hence use the two phases alternatively for convenience. We denote  $t_E(s_c) \stackrel{\text{def}}{=} \max_{i: s(i) \in s_c} t_e(i)$  and call it the *schedule length* of  $s_c$ . Short schedules are preferred. It is usually named *makespan* in the literature of job/task scheduling [13]. We will use the two concepts alternatively for convenience.

**Now, we can state our WPTERD problem formally as follows.** *Given a WSN consisting of a set of static nodes  $U$  in a given space with energy harvesting coefficient matrix  $\mathbf{C} = \{c(i, j)\}$ , energy upper limit list  $e_U$  (vector), energy lower limit list  $e_L$ , energy transmitting power list  $p$ , beginning energy list  $e_B$ , energy expectation list  $e_E$ , the task is to find a valid energy transmission schedule  $s_c$  with maximum total final energy  $E_C(s_c)$  and further with minimum makespan  $t_E(s_c)$ .*

## B. Energy Harvesting Model

The energy harvesting coefficients are determined by energy harvesting model. We adopt a model as described in Eq.(1). Here  $\alpha, \beta, \gamma$  are constants affected by hardware parameters of the energy transceivers as well as the surrounding environment, and  $D$  is the farthest distance that

the energy transmitting signal can reach when  $p=1$ .  $\gamma \in [2, 6]$  represents the channel fading effect and is 4 in free space.

$$c(d) = \begin{cases} \frac{\alpha}{(\beta+d)^\gamma}, & d \leq D * p^{1/\gamma}; \\ 0, & d > D * p^{1/\gamma}, \end{cases} \quad (1)$$

The model in Eq.(1) extends the model in [15] in two aspects: (1) adds the channel fading factor  $\gamma$ , which consists with the popular wireless signal transmission model; (2) makes the energy coverage radius depends on energy transmitting power. Furthermore, the model can be applied to 1D, 2D, and 3D spaces.

*Comment: the analyses and proposed algorithm in later sections do not depend on certain energy harvesting model. The model in Eq.1 is only used in the simulation experiments in Section VIII.*

#### IV. SOLVE THE WPTERD-EGY PROBLEM

To overcome the challenges, we propose a two-step approach as shown in Fig. 1 to solve the WPRERD problem by decoupling the joint energy-time optimization into two embedded sub-problems named WPTERD-Egy and WPTERD-Time, which focus exclusively on the optimization in energy and time, respectively. In the first step, based on some interesting properties of the WPTERD-Egy problem, we reformulate the WPTERD-Egy problem as a linear programming (LP) problem. The LP problem is then solved using optimization software packages to obtain an optimal solution, which contains a set of expected energy transmitting time-lengths of the nodes leading to minimum energy loss. With these results, the remanent work of the WPTERD problem becomes the WPTERD-Time problem, which is to find a schedule of energy transmission operations with minimum makespan meanwhile without energy violations. We call a continuous period of a node's energy transmission operation as an *energy transmission time slice*, the WPTERD-Time problem is thus to schedule the energy transmission time slices. We use *time slice* to refer to *energy transmission time slice* by default for simplicity.

We will provide our analyses and treatment on the WPTERD-Egy problem in this section. The treatments on the WPTERD-Time problem are provided in later sections.

##### A. Problem Formulation of the WPTERD-Egy Problem

Let  $S_C$  be the space of all valid energy transmission schedules. Given a schedule  $s_c \in S_C$ , we can sort all items of  $t_b(i)$  and  $t_e(i)$  into a list  $T_s(s_c) = [t_1, t_2, \dots, t_L]$  in ascending order. Here the

$$e_i(s_c, t) = \begin{cases} e_B(i), & t=0; \\ \min \left\{ e_U(i), \max \left[ 0, e_i(s_c, t_{j-1}) + \left( \mathbf{1}_{i \in U_H(j)} \sum_{k \in U_T(j)} p(k)c(k, i) - \mathbf{1}_{i \in U_T(j)} p(i) \right) (t - t_{j-1}) \right] \right\}, & t \in (t_{j-1}, t_j], \\ & j \in \mathcal{N}(L); \end{cases} \quad (2)$$

list is assumed to have length  $L$ . The time points in  $T_s(s_c)$  divide the time interval  $[0, t_L]$  into time slots  $\{ts(1), ts(2), \dots, ts(n)\}$ , where  $ts(i)$  represents slot  $(t_{i-1}, t_i]$ . For each slot  $ts(i)$ , we can obtain the set of nodes  $U_T(i)$  who are scheduled to transmit energy in the slot, meanwhile all the others in  $U_H(i) = U - U_T(i)$  to harvest energy.

During the energy redistribution process, node  $u_i$ 's energy changes along time  $t \in [0, t_L]$ . Denoting the function of  $u_i$ 's energy on time  $t$  and energy transmission schedule  $s_c$  as  $e_i(s_c, t)$ , then it can be expressed recursively as Eq.(2). The WPTERD-Egy problem can then be formulated as Eq.(3).

$$\begin{aligned} (\mathbf{P1}) \quad & \max_{s_c \in S_C} \sum_{u_i \in U} e_i(s_c, t_E(s_c)) \\ \text{s.t.} \quad & e_i(s_c, t_E(s_c)) \geq e_E(i), \quad i \in \mathcal{N}(n); \\ & e_i(s_c, t_E(s_c)) \leq e_U(i), \quad i \in \mathcal{N}(n); \\ & e_i(s_c, t) \leq e_U(i), \quad i \in \mathcal{N}(n), t \in [0, t_E(s_c)]; \\ & e_i(s_c, t) \geq e_L(i), \quad i \in \mathcal{N}(n), t \in [0, t_E(s_c)]; \end{aligned} \quad (3)$$

The WPTERD-Egy problem has several challenges: (1) it is nonlinear due to the lower and upper limits of energy; (2) the solution space  $S_C$  is infinitely larger because that the variables  $t_i$  can take any real values from a continuous time interval  $[0, t_E(s_c)]$ , and  $t_E(s_c)$  also required to be determined.

Further inspections show that, what significantly affect the nodes' final energy are their energy transmission time-lengths, not the beginning and ending times of the time slots. We can hence focus only on determining the optimal energy transmission time-lengths leading to minimum energy loss.

*The restrictions on no energy violations during the energy redistribution process are postponed to the WPTERD-Time problem, and hence are ignored in the WPTERD-Egy problem.*

### B. Analyses on the WPTERD-Egy Problem

**Lemma 1:** For WPTERD problem with  $e_B \succ e_F$ , for any valid energy transmission schedule with EUL violations, there must be some valid schedules with larger total final energy.

*Proof:* Without loss of generality, we denote the given schedule with EUL violations as  $s_0$ , and assume an EUL violation is happened at node  $v_0 \in U$ . Then we analyze the situation by classifying into the following two cases:

- **case1:**  $\exists v_i \in N(v_0)$  with  $e_F(v_i) < e_U(v_i)$ . To generate a better schedule with no EUL violation, we assume each node is equipped with an auxiliary energy storage which can store the lost energy resulted from EUL violations. Then, after  $s_0$  finishes, we can let node  $v_0$  try to use-up the energy in its auxiliary storage (assume the amount of energy is  $e_{v0}$ ) by an additional energy transmission of time-length  $e_{v0}/p(v_0)$ . Node  $v_i$  will harvest energy  $c(v_i, v_0) * p(v_0) * e_{v0}/p(v_0) = c(v_i, v_0) * e(v_0)$  from this energy transmission, and its new final energy will become  $e'_F(v_i) = \max(e_F(v_i) + c(v_i, v_0) * e(v_0), e_U(v_i))$ . With the given condition that  $e_F(v_i) < e_U(v_i)$ , we have  $e'_F(v_i) > e_F(v_i)$ . Meanwhile, all other nodes' new final energy will at least not decrease. Thus, total final energy after  $v_0$ 's last energy transmission will be greater than that of  $s_0$ . Furthermore, we can obtain a new schedule  $s_1$  by scheduling  $v_0$ 's last energy transmission to earlier times such that the violation at  $v_0$  does not happens, which is true since that we can extend the makespan of the whole schedule when necessary. The new schedule  $s_1$  may still have violations at other nodes, but the violation at node  $v_0$  is avoided.
- **case2:**  $\forall v_i \in N(v_0)$  we have  $e_F(v_i) = e_U(v_i)$ . In this case, there must be a node  $v_1$  with  $e_F(v_1) < e_U(v_1)$  in the network such that there is a path  $\{v_0 = a_1, a_2, \dots, a_k = v_1\}$  connects  $v_0$  and  $v_1$ , as illustrated in Fig.3. As in the previous case, we assume that each node is equipped with an auxiliary energy storage. Then, after  $s_0$  finishes, we can let node  $a_i, i \in \mathcal{N}(k-1)$ , to make additional energy transmissions in turn so as to use-up the excessive energy  $e_{v0}$  at  $v_0$ , meanwhile make  $e'_F(a_i) = e_F(a_i) = e_U(v_i), i \in \mathcal{N}(k-1), e'_F(a_k) > e_F(a_k)$ . The time-lengths of the additional transmissions  $t(a_i), i \in \mathcal{N}(k-1)$ , can be obtained by solving Eq.(4). Furthermore, we can obtain a new schedule  $s_1$  by scheduling these additional energy transmissions to earlier times such that no violations happen at  $a_i, i \in \mathcal{N}(k-1)$ .

Combining above cases, the lemma follows. ■

**Theorem 1:** For WPTERD problem with  $e_B \succ e_F$ , optimal energy transmission schedules with

$$\begin{bmatrix}
-1*p(a_1) & c(a_1, a_2)*p(a_2) & \dots & 0 & 0 \\
c(a_2, a_1)*p(a_1) & -1*p(a_2) & \dots & 0 & 0 \\
0 & c(a_3, a_2)*p(a_2) & \dots & 0 & 0 \\
0 & 0 & \ddots & -1*p(a_{k-2}) & c(a_{k-2}, a_{k-1})*p(a_{k-1}) \\
0 & 0 & 0 & c(a_{k-1}, a_{k-2})*p(a_{k-2}) & -1*p(a_{k-1})
\end{bmatrix}
\begin{bmatrix}
t(a_1) \\
t(a_2) \\
\vdots \\
t(a_{k-2}) \\
t(a_{k-1})
\end{bmatrix}
=
\begin{bmatrix}
-e_{v0} \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\quad (4)$$

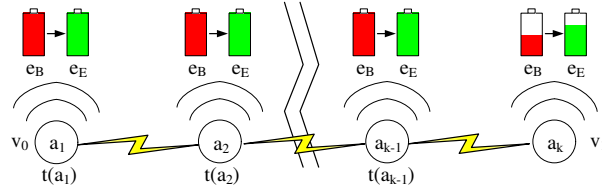


Fig. 3. The path connects  $v_0$  and  $v_1$  where  $e_F(v_1) < e_U(v_1)$ .

maximum total final energy (*i.e.*, minimum energy loss) must have no EUL violations.

*Proof:* It can be proved easily by contradiction. If there is an optimal schedule  $s_1$  with EUL violations, then by applying Lemma 1 we can obtain a new better schedule  $s_2$ , which contradicts with the assumption that  $s_1$  is optimal. ■

For any energy transmission schedule  $s_1$ , if there is a set of neighboring nodes whose time slices overlap (partly or completely), then we can obtain a new schedule  $s_2$  by just moving the overlapped time slices apart meanwhile guaranteeing that they all do not overlap with all other nodes' time slices. This is absolutely possible since that we can always increase  $t_L$  when necessary. We call a schedule where neighboring nodes' time slices do not overlap as a *disjoint schedule*, otherwise an *overlap schedule*.

For the relative priority of the two schedules  $s_1$  and  $s_2$  in term of final energy, we have the following lemma.

**Lemma 2:** For the two energy transmission schedules  $s_1$  and  $s_2$  obtained as above, assume they both have no EUL violations, we have  $E_C(s_2) > E_C(s_1)$ .

*Proof:* Basing on the *energy harvesting additive assumption*, and using Eq.(2), we analyze in the following cases:

- **case1: the set of neighboring nodes whose time slices overlap only contains two nodes.**

Assume the two neighboring nodes as  $u_i$  and  $u_j$ . Replacing  $s_1$  with  $s_2$  will not affect

the amount of energy harvested by any other node  $u_k$ , *i.e.*,  $e_k(s_1, t_E(s_1)) = e_k(s_2, t_E(s_2))$ ,  $\forall k \in \mathcal{N}(n), k \neq i, j$ . Furthermore, the energy harvested by  $u_i$  and  $u_j$  from all the other nodes are kept unchanged when replacing  $s_1$  with  $s_2$ . Now we analyze the harvested energy of  $u_i$  from  $u_j$  in  $s_1$  and  $s_2$ , respectively. We assume the time-lengths of  $u_i$  and  $u_j$ 's energy transmissions in  $s_1$  are respectively  $t(i)$  and  $t(j)$ . Furthermore, assume that the time-length of the overlapped parts of their transmissions is  $t_0$ . Thus, the energy harvested by  $u_i$  from  $u_j$  in  $s_1$  is  $c_{i,j} * p(j) * (t(j) - t_0)$ . Contrastively, the energy harvested by  $u_i$  from  $u_j$  in  $s_2$  becomes  $c_{i,j} * p(j) * t(j)$ . Since that  $c(i, j) \geq 0$ ,  $u_i$ 's final energy  $e_i(s_2, t_E(s_2)) \geq e_i(s_1, t_E(s_1))$ . The same analysis applies to  $u_j$  and so we have  $e_j(s_2, t_E(s_2)) \geq e_j(s_1, t_E(s_1))$ . With the fact  $c_{i,j} + c_{j,i} > 0$  for neighboring nodes  $u_i$  and  $u_j$ , we know that at least one of the equations  $e_k(s_2, t_E(s_2)) \geq e_k(s_1, t_E(s_1))$ ,  $k \in \{i, j\}$ , strictly holds. Considering the given pre-condition that there are no EUL violations, all harvested energy will be able to be stored, hence we have.

$$\begin{aligned}
E_C(s_2) &= e_i(s_2, t_E(s_2)) + e_j(s_2, t_E(s_2)) \\
&\quad + \sum_{k \in \mathcal{N}(n), k \neq i, j} e_k(s_2, t_E(s_2)) \\
&> e_i(s_1, t_E(s_1)) + e_j(s_1, t_E(s_1)) \\
&\quad + \sum_{k \in \mathcal{N}(n), k \neq i, j} e_k(s_1, t_E(s_1)) \\
&= E_C(s_1).
\end{aligned} \tag{5}$$

- **case2: the set of neighboring nodes whose time slices overlap contains three or more nodes.** Without loss of generality, we assume this set contains  $k$  nodes  $U_s = \{u_{s1}, u_{s2}, \dots, u_{sk}\}$ . As in case1, we can analyze the energy of each node in  $U_s$  harvested from the others in  $U_s$ . We can thus obtain that each node's harvested energy in  $s_2$  must be not smaller than that in  $s_1$ , and at least one strictly holds. With the fact that there are no EUL violations, we have  $E_C(s_2) > E_C(s_1)$ .

Combining above cases, the lemma follows. ■

By combining the previous Theorem 1 and Lemma 2, we can obtain the following theorem easily.

**Theorem 2:** For WPTERD problem with  $e_B \succ e_F$ , optimal energy transmission schedules with maximum total final energy (*i.e.*, minimum energy loss) must be disjoint.

*Proof:* We prove it by contradiction. Assume there is an optimal solution  $s_1$  which is an overlap schedule. Theorem 1 tells us that  $s_1$  must have no EUL violations. Then by using

Lemma 2 on  $s_1$ , we can obtain a better schedule  $s_2$ , which contradicts with the assumption that  $s_1$  is optimal. ■

### C. Reformulate and Solve the WPTERD-Egy Problem

Theorem 2 tells us that, to solve the WPTERD-Egy problem, we need only to consider disjoint schedules, which enables us to re-formulate the WPTERD-Egy problem as a more simplified linear programming (LP) problem and thus easy to solve.

Let  $t_i$  be the time-length of node  $u_i$ 's energy transmission time slices (called as time-length of  $u_i$  for short) in a schedule, and let  $t \triangleq \{t_1, t_2, \dots, t_n\}$  be the list of the nodes' time-lengths, then the final energy of node  $u_i$  after performing the schedule can be expressed as Eq.(6). Since that we postpone the treatment for assuring no energy violations during the energy redistribution process to the time when solving the WPTERD-Time problem, the restriction of the energy limits can be only applied to the final energy  $e_F$  here.

$$e_F(i) = e_B(i) - p(i)t_i + \sum_{\substack{j \in \mathcal{N}(n), \\ j \neq i}} c(i, j)p(j)t_j, i \in \mathcal{N}(n) \quad (6)$$

The WPTERD-Egy problem for determining  $\mathbf{t}$  can then be expressed as **P2** in Eq.(7).

$$\begin{aligned} (\mathbf{P2}) \quad & \max_{t=\{t_1, t_2, \dots, t_n\}} \sum_{i \in \mathcal{N}(n)} e_F(i) \\ & s.t. \quad Eq.(6); \\ & \quad e_F(i) \geq e_E(i), \quad i \in \mathcal{N}(n); \\ & \quad e_F(i) \leq e_U(i), \quad i \in \mathcal{N}(n); \\ & \quad t_i \geq 0, \quad i \in \mathcal{N}(n). \end{aligned} \quad (7)$$

Using the matrix and vector symbols defined in Table II, problem **P2** can be expressed in matrix form as **P3** in Eq.(8).

$$\begin{aligned} (\mathbf{P3}) \quad & \max_{\mathbf{t}} \quad \mathbf{1}^T * \mathbf{e}_F \\ & s.t. \quad \mathbf{e}_B + \mathbf{C} * \mathbf{p} * \mathbf{1}^T * \mathbf{t} = \mathbf{e}_F; \\ & \quad \mathbf{e}_F \succeq \mathbf{e}_E; \\ & \quad \mathbf{e}_F \preceq \mathbf{e}_U; \\ & \quad \mathbf{t} \succeq 0. \end{aligned} \quad (8)$$

Problem **P2** (i.e., **P3**) is a standard LP problem, which can be solved efficiently using mature software optimization packages, such as Gurobi, Cplex, Nlopt, etc.

## V. ANALYSES OF THE WPTERD-TIME PROBLEM

Given an energy transmission schedule, we can group the schedule items into sets  $S^i = \{s_1^i, s_2^i, \dots, s_{m_i}^i\}$ ,  $i \in \mathcal{N}(n)$ . Here set  $S^i$  contains  $m_i$  schedule items, where node  $u_i$  should transmit energy. Thus, the total energy transmission period of  $u_i$  contains  $m_i$  time slices with lengths  $t_e(s_j^i) - t_b(s_j^i)$ ,  $j \in \mathcal{N}(m_i)$ .

Once the energy transmission time-lengths of the nodes have been determined, the remanent work to solve the WPTERD problem is to schedule the nodes' energy transmission time slices along the time line, with the objection that the makespan of the schedule is minimum meanwhile no energy violations happen during the process. This is the WPTERD-Time problem, which is what the second step will solve. EUL violations lead to energy loss thus destroying the energy optimality. ELL-violations lead to invalid schedule.

To minimize the makespan of a schedule, we should try to make energy transmission time slices overlap along the time line. However, to preserve energy optimality of a schedule, the time slices of neighboring nodes should not overlap with each other, according to Theorem 2.

The WPTERD-Time problem can be stated formally as follow. *Given a set of static nodes  $U$  in a given space with energy harvesting coefficient matrix  $C = \{c(i, j)\}$ , energy storage capacity limit list  $\mathbf{e}_U$  (vector), energy storage lower limit list  $\mathbf{e}_L$ , energy transmitting power list  $\mathbf{p}$ , beginning energy list  $\mathbf{e}_B$ , energy expectation list  $\mathbf{e}_E$ , energy transmission time-length vector  $\mathbf{t}$ , the task is to find a time slice schedule  $s_c$  for  $\mathbf{t}$  with minimum makespan, meanwhile no energy violation happens during the process.* It can be formulated as Eq.(9).



$$\begin{aligned}
(\mathbf{P4}) \quad & \min_{s_c \in S_c} \max_{s(i) \in s_c} t_e(i) \\
s.t. \quad & e_i(s_c, t_E(s_c)) \geq e_E(i), \quad i \in \mathcal{N}(n); \\
& e_i(s_c, t_E(s_c)) \leq e_U(i), \quad i \in \mathcal{N}(n); \\
& e_i(s_c, t) \leq e_U(i), \quad i \in \mathcal{N}(n), t \in [0, t_E(s_c)]; \\
& e_i(s_c, t) \geq e_L(i), \quad i \in \mathcal{N}(n), t \in [0, t_E(s_c)]; \\
& [t_b(i), t_e(i)] \cap [t_b(j), t_e(j)] = \emptyset, u_s(i) = u_s(j) \\
& \quad , s_i, s_j \in s_c; \\
& [t_b(i), t_e(i)] \cap [t_b(j), t_e(j)] = \emptyset, u_s(i) \neq u_s(j) \\
& \quad , c(u_s(i), u_s(j)) + c(u_s(j), u_s(i)) > 0, s_i, s_j \in s_c; \\
& \sum_{\substack{s(j)=i, \\ s(j) \in s_c}} (t_e(j) - t_b(j)) = t(i), \quad i \in \mathcal{N}(n); \\
& 0 \leq t_b(i) < t_e(i), \quad s_i \in s_c;
\end{aligned} \tag{9}$$

In the WPTERD-Time problem, besides the energy limit restrictions, the main restrictions are that neighboring nodes' time slices should not overlap. The neighboring information of the WPTERD-Time problem can be more conveniently expressed as a graph  $G(V, E)$  defined as Eq.(10). This graph is usually called *conflict graph* in the literature of task scheduling.

$$\begin{cases} V(G) = \{ u_i \mid i \in \mathcal{N}(n) \}, \\ E(G) = \{ e(i, j) \mid c(i, j) + c(j, i) > 0, i \in \mathcal{N}(n) \}. \end{cases} \tag{10}$$

The WPTERD-Time problem contains the following energy transmission task scheduling (ETTS) problem as its core: *Given a set of  $n$  energy transmission tasks with time vector  $t = [t_1, t_2, \dots, t_n]$  and the corresponding conflict graph  $G(V, E, W)$ , to find an energy transmission task schedule with minimum makespan.* The ETTS removes the restrictions of energy limits, we can however solve the WPTERD-Time problem basing on a solution to the ETTS problem. Unfortunately, the ETTS problem is hard to solve, as shown in the following Theorem 3, which is proved by inducing from the well-known NP-hard graph coloring (GC) problem.

**Theorem 3:** The ETTS problem is NP-hard.

*Proof:* We prove it by providing a polynomial reduction from the NP-hard GC problem to the ETTS problem. The GC problem is to find a coloring for a given graph  $G(V, E)$  with minimum number of colors. A coloring of  $G$  using  $k$  colors is a function  $c: V \rightarrow \mathcal{N}(k)$  such that adjacent vertices in  $G$  are assigned with different colors.

For a GC problem instance with graph  $G(V, E)$ , we construct an ETTS problem instance with  $|V(G)|$  unit-length tasks and conflict graph  $G(V, E)$ , and its time-length vector is all 1. The construction of the ETTS problem instance from the GC problem instance is obviously polynomial. Furthermore, there is a one-to-one map between the coloring solutions of the GC instance and the task schedules of the ETTS instance. To be specific, we divided the time line in the ETTS problem into time slots with unit length, then for each graph coloring solution, we can let the nodes colored 1 transmit energy in the 1st time slot, and then those nodes colored 2 transmit in the 2nd slot, and so on. The makespan of such an ETTS solution equals the number of the colors in the GC solution. As a conclusion, there is a polynomial reduction from the NP-hard GC problem to our ETTS problem. ■

**Theorem 4:** The WPTERD-Time problem is NP-hard.

*Proof:* We prove it by providing a polynomial reduction from the ETTS problem to the WPTERD-Time problem.

For the ETTS problem with conflict graph  $G(V, E)$  and task time-length list  $\mathbf{t}$ , we can construct a WPTERD-Time instance with parameters constructed as follows. Use any matrix  $\mathbf{C}$  on condition that it satisfies: (1) diagonal elements take value -1; (2) all elements corresponding to edges in  $G$  take values from (0,1), and all other elements take value 0; (3)  $c(i, j) + c(j, i) > 0$ ,  $\forall (i, j) \in E(G)$ . Furthermore,  $e_L = 0$ ,  $e_U = \inf$ ,  $p = 1$ ,  $e_B = \max(\mathbf{t}) + 1$ , and  $e_E = e_B + \mathbf{C} * \mathbf{t}$ . This can obviously be done in polynomial time. Furthermore, it is easy to check that any schedule to the ETTS instance is a valid energy transmission schedule to the WPTERD-time problem constructed above, and vice versa. In other words, the solutions to the ETTS instance and those to the WPTERD-Time instance have a one-to-one map. The theorem follows. ■

## VI. SOLVE THE WPTERD-TIME PROBLEM

Theorem 3 and Theorem 4 imply that no polynomial algorithms can solve the WPTERD-Time problem and the ETTS problem optimally. Our approach is to approximately solve it in two steps. In the first step, it solves the ETTS problem to obtain the collection of Energy-Transmission Concurrent (ETC) sets. An ETC set is a set of nodes that is scheduled to transmit energy concurrently for a certain time-length associated with the set. Step 1 just returns a collection of ETC sets, whereas the sequence of the energy transmissions is postponed to the second step, which determines the operation sequence of the ETC sets assuring no EUL and ELL violations.

The two steps are implemented as algorithms named as LNSWL-SS and ETCS-S, as shown in Fig. 1.

#### A. Step1: The LNSWL-SS Algorithm for the ETTS Problem

Inspired by the smallest-degree-last algorithm [19] for approximately solving the GC problem, we propose LNSWL-SS for the ETTS problem, which is a three-step algorithm. In the first step, it determines a sequence in which nodes having least neighbor-set-weight (NSW) are ordered at last. Hence we name it Least NSW Last Sequential Scheduling (LNSWL-SS) algorithm. Here a node's NSW means the sum of the weights of the node's neighbors and itself. Then in the second step, it makes schedule decisions for the tasks greedily following this sequence. When making schedule decisions for each task, it is scheduled to run in any time slice not occupied by any of its neighbors already scheduled. Finally it returns the list of schedule items in format of 2-tuple (concurrent-task-set, time-length). A schedule item represents a decision that the nodes in concurrent-task-set should transmit energy concurrently for time time-length. LNSWL-SS works on a weighted graph  $G(V, E, W)$  which encapsulates the ETTS problem's conflict graph  $G(V, E)$  and task time-length vector  $\mathbf{t}$  by assigning weight  $W(i)=t(i)$ .

The pseudo code of LNSWL-SS is shown in Alg. 1. The while-loop between code lines 2-7 implements the first step. A property value of graph  $G$ , defined as  $\varpi(G) \stackrel{\text{def}}{=} \max_{H \subseteq G} (\min_{v \in H} w(N_{\text{NSW}}(v, H))$ , is also obtained accompanying, which denotes the largest  $\varpi$  such that  $G$  contains a sub-graph  $H$  in which each node's NSW is at least  $\varpi$ . The for-loop between code lines 8-10 implements the second step. Code lines 12-15 implements the last step.

#### B. Performance Ratio of the LNSWL-SS Algorithm

The quality of the solutions returned by an approximate algorithm can be coarsely implied by its performance ratio. Performance ratio of an algorithm for an maximization (minimization) problem is a constant  $\rho \leq 1 (\geq 1)$  such that, for any problem instance, the value of any solution returned by the algorithm is at least (at most)  $\rho$  times of the optimal value.

For general graphs, the graph coloring problem is NP-hard even for any fixed number of colors  $k \geq 3$  [17]. Furthermore, it is hard to approximate, *i.e.*, the problem of approximating the chromatic number with any constant ratio is also NP-hard [18]. Fortunately, for the ETTS problem embedded in the WPTERD-Time problem, whose conflict graph is the intersection graph of the nodes' energy signal coverage disks, our LNSWL-SS algorithm has constant approximation

**Algorithm 1** LNSWL-SS algorithm for the ETTS problem**Require:**  $G(V, E, W)$ : weighted graph of the problem;**Ensure:**  $\varpi(G)$ ;  $S_{si}$ : the set of schedule items;  $S_{cts}$ : the collection of (concurrent-task-set, time-length) schedule items;

---

```

1: Initialize  $G'=G(V, E)$ ,  $\varpi(G)=0$ ,  $v_{List}=[]$ ;
2: while  $G' \neq \emptyset$  do
3:    $\varpi(G) = \max(\varpi(G); \min_{v_j \in V(G')} w_{NSW}(v_j, G'))$ ;
4:    $v_i = \arg \min_{v_j \in V(G')} w_{NSW}(v_j, G')$ ;  $v_{List} = [v_i, v_{List}]$ ;
5:    $V' = V(G') - v_i$ ;  $E' = E(G') - \{e(i, k) | e(i, k) \in V(G')\}$ ;
6:    $G' = G(V', E')$ ;
7: end while
8: for  $i=1:n$  do
9:   Assign all time slices unused by any scheduled items of the nodes in  $N(G', v_i)$ , and insert
   the corresponding schedule items into  $S_{si}$ ;
10: end for
11: Sort the beginning and ending time values of all items in  $S_{si}$  into a list  $T_s(s_c)=[t_1, t_2, \dots, t_L]$ 
   in ascending order(here assume its length is  $L$ ), we thus obtain  $L-1$  time slots with slot  $i$ 
   occupies time interval  $(t_i, t_{i+1}]$ .
12: for  $l=1:L-1$  do
13:   Obtain the set  $U_T(i)$  of concurrent tasks in time slot  $i$ ;
14:    $S_{cts}(i).CTS = U_T(i)$ ;  $S_{cts}(i).tLen = t_{i+1} - t_i$ ;
15: end for
16: return  $S_{si}, S_{cts}, \varpi(G)$ ;

```

---

ratios. We denote the makespan of the solution returned by LNSWL-SS for the ETTS problem as  $m_{LNSWL}$ , and denote the makespan of the corresponding optimal solution as  $m_{OPT}$ .

**Lemma 3:** The solution returned by LNSWL-SS for the ETTS problem must have makespan  $m_{LNSWL}$  not greater than  $\varpi(G)$ .

*Proof:* To prove that  $m_{LNSWL} \leq \varpi(G)$  is equivalent to prove that all time slices of the nodes in a schedule returned by LNSWL-SS must fall into time interval  $[0, \varpi(G)]$ . We will prove it by induction. LNSWL-SS determines the time slices of the nodes in the sequence of  $v_{List}[1:n]$ .

For the first node  $v_{\text{List}}(1)$ , its energy transmission operation can definitely be scheduled in  $[0, \varpi(G)]$ . Now assume that the scheduled time slices for the nodes in  $v_{\text{List}}[1:i]$  are all in  $[0, \varpi(G)]$ . According to the definition of  $\varpi(G)$  and the construction of the list  $v_{\text{List}}[1:n]$ , we must have  $w(N(v_{\text{List}}(i+1))) \leq \varpi(G)$ , otherwise  $\varpi(G)$  will be updated as  $\varpi(G) = w(N(v_{\text{List}}(i+1)))$  by code line3 in Alg. 1 at that time. Hence, we can surely schedule node  $v_{\text{List}}(i+1)$ 's energy transmission operations in between the time slices occupied by its neighbors in  $[0, \varpi(G)]$ . The lemma follows. ■

We call a WSN where all nodes' energy signal coverage disks have equal radius as an equal-radius WSN, whereas other WSNs are called as non-equal-radius WSNs.

**Lemma 4:** For equal-radius WSNs in 2D space, the approximation ratio of LNSWL-SS is 3.

*Proof:* Our proof mimics the proof of the theorem in [19]. First, the makespan  $m_{\text{OPT}}$  of any optimal schedule must not be smaller than the weight  $w_{c,G}$  of the maximum clique, we have Eq.(11).

$$m_{\text{OPT}} \geq w_{c,G} \quad (11)$$

Let  $H$  be a subgraph of  $G$  where the nodes' NSWs are at least  $\varpi(G)$ , and let  $v^* \in H$  be the node having smallest NSW. Using the definition of  $\varpi(G)$  and Lemma 3, we have Eq.(12).

$$w_{\text{NSW}}(v^*, H) \geq \varpi(G) \geq m_{\text{LNSWL}} \quad (12)$$

Since that  $v^*$  has the smallest  $Y$ -coordinate in  $H$ , all nodes in  $H$  must be in the half plane above the horizontal line through node  $v^*$ . Let  $r$  be the radius of the nodes' energy signal coverage disks. We divide the top half disk with radius  $r$  centered at  $v^*$  into three equal sectors as shown in Fig.4, where the blank circles represent the nodes in  $N(v^*, H) \stackrel{\text{def}}{=} H \cap N(v^*)$ . The nodes on the division lines can be assumed to be in either of the two adjacent sectors deterministically.

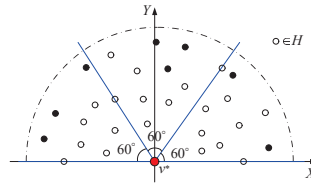


Fig. 4. Divide the top half disk into 3 sectors.

Because that all nodes have energy signal coverage radius  $r$ , the nodes in each sector must form a clique. Denote the node set in the three sectors as  $S_1$ ,  $S_2$ , and  $S_3$ , respectively. Assume the maximum clique of  $G$  has weight  $w_{c,G}$ , then we have Eq.(13).

$$\begin{aligned}
 w_{\text{NSW}}(v^*, H) &= w(S_1 \cap H + S_2 \cap H + S_3 \cap H - 2 * w(v^*)) \\
 &\leq w(S_1 \cap H + S_2 \cap H + S_3 \cap H) \\
 &= w(S_1 \cap H) + w(S_2 \cap H) + w(S_3 \cap H) \\
 &\leq 3 * w_{c,G}
 \end{aligned} \tag{13}$$

Combining the equations from Eq.(11) to Eq.(13), we obtain  $m_{\text{LNSWL}} \leq 3 * m_{\text{OPT}}$ , *i.e.*, the performance ratio of LNSWL-SS is 3. The lemma follows. ■

**Lemma 5:** For equal-radius WSNs in 3D space, the approximation ratio of LNSWL-SS is 12.

*Proof:* The proof is similar to that in the previous lemma but with some adaptations to the 3D space. The differences lie in the following aspects: (1)  $v^*$  represents the node with smallest  $Z$  coordinate value in  $H$ ; (2) divide the top half sphere into 12 sectors as shown in Fig.5 (firstly use the planes  $X=0$  and  $Y=0$  to split the top half sphere into 4 equal sectors, then divide each sector into 3 sectors further using three planes, each of which contains the line through point  $(0,0,0)$  and the center point of the sector's spherical face meanwhile perpendicular to the splitting planes).

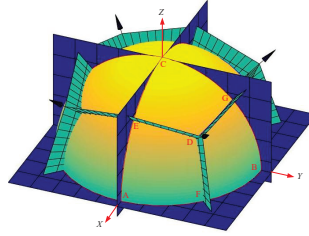


Fig. 5. Divide the top half sphere into 12 sectors.

It is easy to notice that any two points in any of the 12 sectors has distance not greater than  $r$ . For example, the distances between the points  $A$ ,  $D$ ,  $E$ ,  $F$  in Fig.5 can be obtained easily as follows.

$$\begin{aligned}
 l_{DA} &= r \sqrt{\left(\frac{1}{\sqrt{3}}\right)^2 + 2 * \left(\frac{1}{\sqrt{3}}\right)^2} && \approx 0.9194r < r \\
 l_{DE} = l_{DF} &= r \sqrt{2\left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{3}}\right)^2 + \left(\frac{1}{\sqrt{3}}\right)^2} && \approx 0.6058r < r \\
 l_{EF} &= r \sqrt{2 * \left(\frac{1}{\sqrt{2}}\right)^2} && = r
 \end{aligned} \tag{14}$$

Because that the four nodes are extreme nodes in the sector, the distance between any points in the sector must be not longer than  $\max\{l_{DA}, l_{DE}, l_{DF}, l_{EF}\} < r$ . Thus, all nodes in each section must form a clique.

Applying the analysis process in proving Lemma 4 to the 3D case with above adaptations, we obtain  $\frac{m_{\text{LNSWL}}}{m_{\text{OPT}}} \leq 12$ . ■

**Lemma 6:** For non-equal-radius WSNs in 2D (3D) space, the approximation ratio of LNSWL-SS is 6 (24).

*Proof:* These can be proved by applying some adaptations for non-equal-radius WSNs to the analysis process used in the previous lemmas. To be specific, assume  $H$  is a subgraph of  $G$  such that the nodes' NSWs in  $H$  are at least  $\varpi(G)$ .

Assume  $v^* \in H$  is the node with the smallest radius  $r_1$ . All other nodes in  $H$  must lie around  $v^*$  instead of in a half space around  $v^*$ . For assuring that all nodes in a sector form a clique, for the 2D case, we divide the disk centered at  $v^*$  with radius  $r_1$  into 6 equal sectors as in Fig.6(a). For the 3D case, we divide the sphere centered at  $v^*$  with radius  $r_1$  into 24 equal sectors as in Fig.6(b). Because that  $v^*$  has the smallest radius, the nodes in  $H$  fall into each sector form a clique.

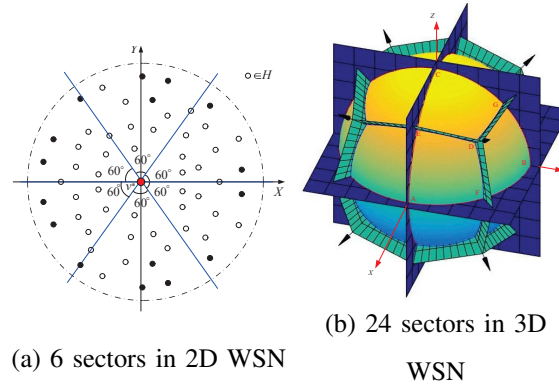


Fig. 6. Divide the space around  $v_*$  with radius  $r_1$  equally.

With these adaptations, we can obtain the lemma by applying the analysis used in proving previous lemmas. ■

### C. Step2: Schedule Considering Energy Limits

This step determines a  $(set, time)$  sequence of energy transmission time slices, where the  $set$  in a sequence item determines a set of nodes that should transmit energy simultaneously, and the

*time* in the item determines the length of the time slice. We propose the Energy-Transmission Concurrent Set Scheduling (ETCS-S) algorithm to determine the schedule sequence making use of the collection of concurrent task sets obtained by LNSWL-SS.

The pseudo code of ETCS-S is shown in Alg. 2. It firstly inserts all items in  $S_{cts}$  obtained by LNSWL-SS into a queue. We call these items as candidate schedule items. ETCS-S determines the available time-slice length  $t3$  for the candidate schedule items in the queue in turn (code lines 4-14).  $t3$  is mainly determined by whichever is the earliest among the three values: (1) the earliest time that one node's remaining time (which means the difference between the required time-length and the accumulated time-lengths of already scheduled items for the node) becomes 0 (code line 4); (2) the earliest time that some node reaches its lower energy limit  $e_L$  (code line 5); (3) the earliest time that some node reaches its upper energy limit, which is determined using the bisection search method when necessary (code lines 8-14). The other lines update the candidate schedule items correspondingly and generating new candidate schedule items when necessary determined using a threshold parameter  $num_{lim}$ .

## VII. SOLVE THE WPTERD PROBLEM

Based on the analyses in previous sections, we propose an algorithm to solve the WPTERD problem by combining the preliminary algorithms proposed. The outline of the final algorithm is already demonstrated in Fig. 1. The algorithm decouples the energy-time joint optimization into the WPTERD-Egy problem and the WPTERD-Time problem, which focus exclusively on the optimization in energy and time, respectively. The graph-coloring inspired LNSWL-SS algorithm is the core of our algorithm, hence we name it Graph-Coloring inspired Energy-Time decoupling (GC-EgyTimeD) algorithm. The pseudo code of GC-EgyTimeD is shown in Alg.3. GC-EgyTimeD firstly solve the **P3** problem using mature LP optimization software packages to obtain  $t$ . Then, it obtains  $S_{cts}$  by using the LNSWL-SS algorithm to solve the corresponding ETTS problem, and at last generates a schedule  $ST_{ss}$  from  $S_{cts}$  by using the ETCS-S algorithm.

*By using GC-EgyTimeD to carefully schedule the energy transmissions of the nodes in WSNs, we are able redistribute energy in the network, which implicitly realizes the multi-hop energy flow efficiently yet easily. Meanwhile, the broadcasting nature of radio signals is well exploited to achieve the most energy efficiency.*



## VIII. PERFORMANCE EVALUATION

### A. Performance Metrics

We conduct numerical simulations using Matlab 2015a on a computer with Win10-bit64, 2.21GHz i7-CPU, and 8GB Memory. Three performance metrics are used: *Energy Loss Ratio*, *Makespan*, and *Switch Number*. The energy loss ratio metric is calculated as  $\frac{\sum_{i \in \mathcal{N}(n)} (e_B(i) - e_F(i))}{\sum_{i \in \mathcal{N}(n), e_E(i) > e_B(i)} (e_E(i) - e_B(i))}$ , which implies the side 'cost' for redistribute energy in the network. The makespan metric represents the time-span of a schedule. The switch number metric represents the number of continuous energy transmissions slices of the nodes in the schedule, which is obtained by counting the starts of energy transmissions. This metric implies the number of node-status changes in a schedule. We obviously prefer algorithms with smaller energy loss ratio, smaller makespan, and smaller switch number.

### B. Simulation Setup

For comparison purpose, besides GC-EgyTimeD, we implement another algorithm denoted as AlgNoConCur, where the nodes transmit energy one by one, *i.e.*, no concurrency energy transmissions even for non-neighboring nodes. In AlgNoConCur, the time-length of each continuous energy transmission slice is determined in a way similar to that in ETCS-S in Alg. 2. Additionally, it is obvious that the maximum clique-weight of the weighted graph  $G(V, E, W)$  of the ETTS problem is a lower bound for the makespan of an optimal schedule. We use a greedy-based algorithm denoted as LBCLique to approximately solve the maximum clique-weight problem and use it as a baseline for performance evaluation. Thus, totally three algorithms are tested in our simulations: GC-EgyTimeD, AlgNoConCur, and LBCLique.

Main parameters and the default values in our simulations include *number of nodes*  $n=100$ , *side length*  $L=10$  of the square network region, *energy transmission power*  $p=1$ , *ratio of energy-demanding nodes*  $\eta=0.3$ , and *the amount of energy demanded by these energy-demanding nodes* is  $e_h=5$ ,  $e_U=100$ ,  $e_L=20$ ,  $e_{E,L}=30$ ,  $e_{E,U}=90$ . The energy harvesting mode parameters are set as  $\alpha=0.3$ ,  $\beta=1$ ,  $\gamma=2$ , and  $D=4$ . A set of particular values for these parameters is called a **simulation configuration**. For each simulation configuration, 300 problem instances are generated and treated using the tested algorithms in turn. The WPTERD-Egy problem of randomly generated WSNs may have no LP solutions at all, in such cases new problem instances are generated repeatedly until a valid problem instance is obtained. In each problem instance,  $n$  nodes are

randomly placed in the square region  $L \times L$  (Although our analysis applies to 3D space, we restrict our simulations to 2D space, which does not damage the effectiveness of the simulations).  $e_B(i)$ ,  $i \in \mathcal{N}(n)$ , are randomly selected in  $[e_{E,L}, e_{E,U})$  following the uniform distribution. Randomly select  $\lceil n * \eta \rceil$  energy-demanding nodes and set  $e_E(i)$  of each selected node to be  $e_B(i) + e_h$ . The others nodes' energy expectations are set to  $e_{E,L}$ . Results of the performance metrics are collected and averaged to obtain the final results for the simulation configurations. The 95% confidence intervals of the performance metrics are also calculated.

### C. Simulation Results

In our simulation experiments, the effects of a parameter on the algorithms' performances are obtained by performing a simulation set consisting of similar simulation configurations. The simulation configurations in such a simulation set only different in the value of this selected parameter, whereas all other parameters take default values. We conduct simulation experiments for inspecting the effects of the main parameters. All simulation results verify the efficiency of our algorithm. Here we only provide the simulation results for parameter  $n$  because of space limitation.

In the simulation experiment for  $n$ , we let  $n$  take values from 10 to 100 with step size 10. The results are shown in Fig.7. The metrics of energy loss ratio and switch number do not applies to LBCLique. Since that the other two algorithms make schedules based the shared solution  $t$  of **P3**, there are no differences in the energy loss ratio metric, as shown in Fig.7(a). As  $n$  increases, the energy loss ratio decreases quickly. This is reasonable since that, with higher node density, the energy harvesting coefficients will be much larger, and more nodes will be able to harvest energy from a single energy transmission. These two factors both lead to less energy loss. The results in Fig.7(b) show that, GC-EgyTimeD usually obtains makespans no more than 115% of LBCLique. These results imply that the solutions of GC-EgyTimeD are nearly optimal. Compared with AlgNoConCur, the makespan metric of GC-EgyTimeD is decreased by more than 65% when  $n=100$ , and it becomes more effective in this metric as  $n$  increases. These results imply that GC-EgyTimeD can exploit concurrent energy transmission opportunities well. This is obtained at the cost of more node-status switches, as shown in Fig.7(c). Compared with AlgNoConCur, when  $n=100$ , GC-EgyTimeD incurs about 2.3 times more switches.

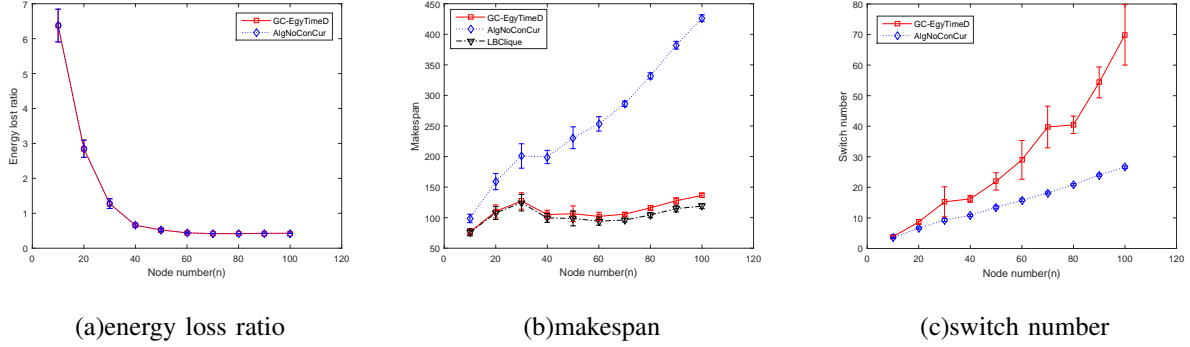


Fig. 7. Effects of node number  $n$  on performance metrics of the algorithms.

## IX. CONCLUSION

In this paper, we investigated WPTERD problem for redistributing energy among nodes optimally in energy efficiency and time. Inspired by a graph coloring algorithm, we propose GC-EgyTimeD to solve the WPTERD problem by decoupling the energy-time joint optimization into two problems WPTERD-Egy and WPTERD-Time. They focus exclusively on the optimization in energy and time, respectively. We transform WPTERD-Egy as a linear programming (LP) problem by exploiting its properties, and obtain optimal time-lengths of the nodes' energy transmissions leading to minimum energy lost. Then the WPTERD-Time is solved to obtain an energy transmission schedule with quasi-minimum makespan. Numerical simulations illustrate the effectiveness and efficiency of GC-EgyTimeD, which return schedules with minimum energy lost and approximately minimum makespan.

## ACKNOWLEDGMENT

This research was funded by Natural Science Foundation of China (No.XXXXXXXXXX, XXXXXXXXXX, XXXXXXXXXX).

## REFERENCES

- [1] P. Guo, X. F. Liu, S. J. Tang, J. N. Cao, "Concurrently wireless charging sensor networks with efficient scheduling," *IEEE Transactions on Mobile Computing*, vol.16, no.9, pp.2450-2463, Sep. 2017.
- [2] P. Nintanavongsa, U. Muncuk, D. R. Lewis, and K. R. Chowdhury, "Design optimization and implementation for RF energy harvesting circuits," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol.2, no.1, pp.24-33, 2012.
- [3] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances", *Science*, vol.317, no.5834, pp.83-86, 2007.

- [4] B. Cannon, J. Hoburg, D. Stancil, and S. Goldstein, "Magnetic resonant coupling as a potential means for wireless power transfer to multiple small receivers", *IEEE Transactions on Power Electronics*, vol.24, no.7, pp.1819-1825, July 2009.
- [5] L. Xiang, J. Luo, K. Han, and G. T. Shi, "Fueling wireless networks perpetually: a case of multi-hop wireless power distribution," in *Proc. IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC2013)*, London, UK, 8-11 Sep. 2013, pp.1994-1999.
- [6] T. Rault, A. Bouabdallah, Y. Challal, "Multi-hop wireless charging optimization in low-power networks," *IEEE Global Communications Conference(GlobeCom2013)*, Atlanta, USA, 2013, pp.484-489.
- [7] P. Guo, X. F. Liu, T. F. Tang, S. J. Tang, J. N. Cao, "Practical concurrent wireless charging scheduling for sensor networks," in *Proc. IEEE 36th International Conference on Distributed Computing Systems(ICDCS2016)*, Nara, Japan, 27-30 Jun. 2016, pp.741-742.
- [8] E. Bulut, M. E. Ahsen, B. K. Szymanski, "Opportunistic wireless charging for mobile social and sensor networks", in *Proc. IEEE 6th International Workshop on Management of Emerging Networks and Services(Globecom2014)*, Austin, USA, 8-12 Dec. 2014, pp.207-212.
- [9] S. Nikolettseas, T. P. Raptis, C. Raptopoulos, "Energy balance with peer-to-peer wireless charging," in *Proc. IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems(MASS2016)*, Brasilia, Brazil, 10-13 Oct. 2016, pp.101-108.
- [10] S. Nikolettseas, T. P. Raptis, C. Raptopoulos, "Wireless charging for weighted energy balance in populations of mobile peers," *Ad Hoc Networks*, vol.60, pp.1-10, 2017.
- [11] A. Madhja, S. Nikolettseas, C. Raptopoulos, "Energy aware network formation in peer-to-peer wireless power transfer," in *Proc. ACM 19th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM2016)*, Malta, Nov. 13-17, 2016, pp.43-50.
- [12] A. Madhja, S. Nikolettseas, T. P. Raptis, C. Raptopoulos, and D. Tsolovos, "Peer-to-peer wireless energy transfer in populations of very weak mobile nodes," in *Proc. IEEE Wireless Communications and Networking Conference Workshops (WCNCW2017)*, San Francisco, USA, 19-22 Mar. 2017, pp.1-6.
- [13] Dániel Marx, "Graph colouring problems and their applications in scheduling," *Periodica Polytechnica SER. EL. ENG.*, vol.48, no.1, pp.11-16, 2004.
- [14] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol.12, no.10, pp.1931-1942, Oct. 2013.
- [15] H. P. Dai, Y. H. Liu, G. H. Chen, X. B. Wu, T. He, A. X. Liu, H. Z. Ma, "Safe charging for wireless power transfer," *IEEE/ACM Transactions on Networking*, vol.25, no.6, pp.3531-3544, Dec. 2017.
- [16] H. P. Dai, H. Z. Ma, A. X. Liu, and G. H. Chen, "Radiation constrained scheduling of wireless charging tasks," *IEEE/ACM Transactions on Networking*, vol.26, No.1, pp.314-326, Feb. 2018.
- [17] M. R. Garey, and D. S. Johnson, "Computers and intractability," *A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.
- [18] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," *Journal of the ACM*, vol.45, no.3, pp.501-555, May 1998.
- [19] M.V. Marathe, H. Breu, H.B. Hunt, S.S. Ravi, D. J. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol.25, no.2, pp.59-68, Mar. 1995.

**Algorithm 2** The ETCS-S algorithm**Require:**  $S_{cts}$ ,  $e_B$ ,  $e_U$ ,  $e_L$ ,  $e_E$ ,  $C$ ,  $p$ ;**Ensure:**  $TS_{ss}$ : the time-set schedule sequence;

```

1:  $num(i) = 0, \forall u_i \in U$ ;  $enqueue(Q, S_{cts})$ ;
2: while  $nonEmpty(Q)$  do
3:    $s1 = dequeue(Q)$ ;
4:    $i_1 = \arg \min_{i \in s1.CTS} t(i)$ ;  $t_1 = t(i_1)$ ;
5:    $i_2 = \arg \min_{i \in s1.CTS} (e_t(i) - e_L(i)) / p(i)$ ;  $t_2 = (e_t(i_2) - e_L(i_2)) / p(i_2)$ ;
6:   if  $t_2 > 0$  then
7:      $t3 = \min\{t_1, t_2\}$ ;
8:      $e_{t1} = e_t - C * p * tv_{s1.CTS}(t3)$ ;
9:     while  $\exists i \in N(s1.CTS), e_{t1}(i) > e_U(i)$  do
10:      Find the maximum  $t3$  that leads to no EUL violation using the bisection method;
11:      if  $t3 < \epsilon$  then
12:         $t3 = \epsilon$ ; break;
13:      end if
14:    end while
15:     $e_t = e_t - C * p * tv_{s1.CTS}(t3)$ ;  $e_t = \min(e_t, e_U)$ ;
16:     $t(i) = t(i) - t3, i \in s1.CTS$ ;
17:     $s1.CTS = s1.CTS$ ;  $s1.tLen = t3$ ;
18:     $ST_{ss} = [ST_{ss}, s1]$ ;  $s1.tLen = s1.tLen - t3$ ;
19:     $s1.CTS = \{i | i \in s1.CTS, t(i) > 0\}$ ;
20:    if  $s1.tLen > 0$  then
21:       $enqueue(Q, s1)$ ;
22:    end if
23:  else
24:     $num(i_2) = num(i_2) + 1$ ;
25:    if  $num(i_2) > num_{lim}$  then
26:       $s1.CTS = \{i_2\}$ ;  $s1.tLen = s1.tLen$ ;
27:       $enqueue(Q, s1)$ ;
28:    end if
29:  end if
30: end while

```

31: **return**  $ST_{ss}$ ;

---

**Algorithm 3** The GC-EgyTimeD algorithm
 

---

**Require:**  $e_B, e_U, e_L, e_E, C, p$ ;

**Ensure:**  $ST_{ss}$ : the (set,time) schedule sequence;

- 1:  $[t] = \text{solveP3}(e_B, e_O, e_U, C, p)$ ;
  - 2: Construct ETTS problem using  $t$  and create  $G(V, E, W)$ ;
  - 3:  $[S_{cts}] = \text{LNSWL-SS}(G(V, E, W))$ ;
  - 4:  $[ST_{ss}] = \text{ETCS-S}(S_{cts}, e_B, e_U, e_L, e_E, C, p)$ ;
  - 5: **return**  $ST_{ss}$ ;
-