GC-EgyTimeD: Graph Coloring Inspired Energy-Time Decoupled Optimal Wireless Energy Redistribution in WSNs

Author1, Author2, Author3

***Abstract*—Wireless energy redistribution based on wireless power transfer technology is a core building block for supporting perpetual wireless sensor networks (WSNs) charged with moving or fixed wireless chargers, and it is especially important for prolong the network normal operation time when WCs could not to charge the network timely. In this paper, we investigated the wireless power transfer based energy redistribution (WPTERD) problem in WSNs, which is to redistribute the energy among network nodes so that all nodes' expected energy amount are satisfied when possible, meanwhile guaranteeing that the energy lost in this process is minimized and the time length of the redistribution process is minimized. We propose a two-step approach which decouples the joint energy-time optimization into to two sub-problems named WPTERD-Egy and WPTERD-Time, which focus exclusively on the optimization in energy and time, respectively. In the first step, we formulate the WPTERD-Egy problem as a linear programming (LP) problem, which returns the optimal time lengths of the nodes' energy transmissions leading to minimum energy lost. The objective of the WPTERD-Time problem in the second step is to find a schedule of energy transmission time slices with minimum makespan meanwhile guaranteeing some energy restrictions. When ignoring the energy restrictions, the corresponding WPTERD-Time problem is called Energy Transmission Task Scheduling (ETTS) problem. We prove that both WPTERD-Time and ETTS are NP-hard, and propose an algorithm named Least conflicting-neighbor-set-weight Last Sequential Scheduling (LNSWL-SS) for solving the ETTS problem, on the inspiration of a graph coloring algorithm. Basing on LNSWL-SS, combining solutions to WPTERD-Egy and WPTERD-Time, we propose an algorithm named Graph Coloring inspired Egy-Time Decoupling (GC-EgyTimeD) to solve the WPTERD problem. We also obtain some approximation ratios of LNSWL-SS for 2D WSNs and 3D WSNs, respectively. Numerical simulations illustrate the effectiveness and efficiency of GC-EgyTimeD, which return schedules with minimum energy lost and approximately minimum makespan. By exploiting parallel opportunities of energy transmissions, GC-EgyTimeD reduces makespan by about more than 60% when compared with a schedule without exploiting the parallel energy transmission opportunities, at the expense of 10% more energy transmission switches.**

***Index Terms*—Wireless energy redistribution problem, wireless power transfer, task scheduling problem, wireless charging, graph coloring problem.**

I. Introduction

RECENT breakthroughs in the areas of Wireless Power Transfer(WPT) technology [1], [2] and rechargeable batteries [3] open up a new dimension to prolonging the lifetime of wireless sensor networks (WSNs). 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 sensor nodes in a relative long distance (>10m away)[4]. It has been validated that sensor node could harvest radio power of 6mW from a wireless charger with 4W transmission power over a distance of 12 meters. The received radio power is 20mW and the transition efficiency is 30 percent [5].

WPT will have a profound impact on the design of WSNs attributed to its obvious advantages, and hence many research efforts have been devoted to exploiting WPT to enhance WSNs [6]-[13]. Some works focus on the application of using dedicated fixed or mobile nodes called wireless chargers (WCs) for powering the normal nodes in WSNs using WPT. Due to the limited number of WCs, some nodes may could not obtain power from the WCs directly. This is true even if there are mobile WCs, because that there may be some restrictions on their movement, or the WCs may be fail temporally. Hence, WPT based Energy ReDistribution (WPTERD) is a core building block for supporting perpetual WSNs.

In recent years some research efforts have been devoted on WPTERD-like application, where the nodes exchange energy in peer-to-peer mode when coming into vicinity of each other during moving, so as to make the nodes' remained energy become more balanced. However, the intrinsic broadcasting feature of energy transmitting signals, which means multiple nodes may harvest energy simultaneously from the signal transmitted by a single node, is not exploited, and hence leads to lower energy efficiency.

In this paper, we focus on the WPTERD problem for WSNs: given a WSN with nodes all equipped with wireless power transceivers and limited energy storages containing some known energy, the task is to redistribute the energy in the network among all nodes through WPT, assuring that the energies of the nodes after the redistribution all do not below their energy expectations, meanwhile the energy loss is minimized and the time length (called 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, outlined in Fig. 1. They 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

The authors are with the Department of
Manuscript received XXX, XX, 2018; revised XXX, XX, 2018.

thus be solved easily to obtain the time lengths of the nodes' energy transmissions. With the results in the first step, the remanent of the WPTERD problem becomes a task scheduling problem of the nodes' energy transmissions with some energy restrictions resulted from energy storage limitations, which is named WPTERD-Time. When ignoring the energy restrictions, the corresponding WPTERD-Time problem is called Energy Transmission Task Scheduling (ETTS) problem. We prove that both WPTERD-Time and ETTS are NP-hard, and propose an algorithm named Least conflicting-neighbor-set-weight Last Sequential Scheduling (LNSWL-SS) for solving the ETTS problem, on the inspiration of a graph coloring algorithm. A pair of nodes is said to be conflicting neighbors of each other so long as one node's energy transmitting signals can reach the other one. All conflicting neighbors of one node make up the conflicting-neighbor-set of the node. A node's conflicting-neighbor-set-weight is the accumulated weights of the node's conflicting-neighbors. Basing on LNSWL-SS, combining solutions to WPTERD-Egy and WPTERD-Time, we propose an algorithm named Graph Coloring inspired Egy-Time Decoupling (GC-EgyTimeD) to solve the WPTERD problem. Numerical simulations illustrate the effectiveness and efficiency of GC-EgyTimeD, which return schedules with minimum energy loss and approximately minimum makespan. By exploiting parallel opportunities of energy transmissions, GC-EgyTimeD reduces makespan by about 30% when compared with a schedule without using the parallel opportunities.

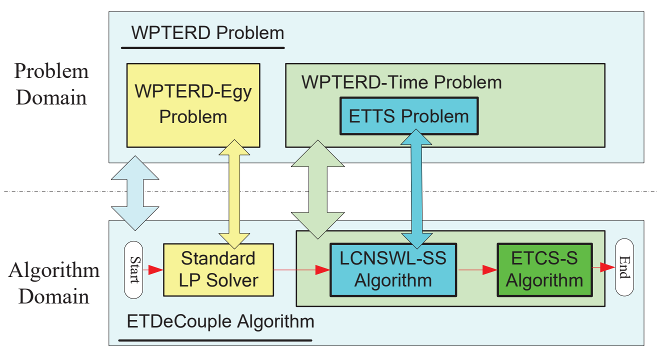


Fig. 1. The outline of our two-step energy-time decoupling approach to the WPTERD problem.

Our main contributions in this paper are as follows:

- We the Graph Coloring inspired Egy-Time Decoupling (GC-EgyTimeD) algorithm to solve the WPTERD problem efficiently and approximately by dividing it into two sub-problems WPTERD-Egy and WPTERD-Time and solve them in two steps successively. This approach exploits the broadcast nature of wireless signals well, and implicitly realizes the multi-hop energy transmission easily.
- For the WPTERD-Egy problem, we prove that its optimal solutions must be disjoint solutions, where energy transmission time intervals of neighboring nodes should not overlap in time line. With these properties, the WPTERD-Egy problem is then formulated as a Linear Programming (LP) problem, with can be solved easily.
- For the WPTERD-time problem and its sub-problem ETTS, we prove that they are both NP-hard by using reduction from the Graph Coloring (GC) problem.
- For the ETTS problem, we propose Least conflicting-neighbor-set-weight Last Sequential Scheduling(LNSWL-SS) algorithm on the inspiration of the well-known least degree last algorithm for the GC problem. Additionally, we prove that, if the approximation ratios of LNSWL-SS for 2D WSNs and 3D WSNs are 6 and 24, respectively. When restricted to cases where all nodes the same energy transmission radius, then the approximation ratios of LNSWL-SS for 2D WSNs and 3D WSNs are 3 and 12, respectively.

For easy reference, main abbreviations in the paper are collected in Table I.

TABLE I
ABBREVIATIONS.

Abbreviations	Whole words/centences
WPT	Wireless Power Transfer;
WPTERD	Wireless Power Transfer based Energy Re-Distribution;
WPTERD-Egy	WPTERD problem with only the energy lost minimization 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 energy restrictions imposed by energy storage limitations;
LNSWL-SS	Least conflicting-neighbor-set-weight Last Sequential Scheduling algorithm for solving ETTS;
ETCS-S	energy-transmission concurrent set scheduling algorithm, which together with LNSWL-SS, solves WPTERD-Time;
GC-EgyTimeD	Graph Coloring inspired Egy-Time Decoupling algorithm for solving WPTERD;
NSW	neighbor-set-weight;
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 described in Section III. Analysis results and the method to solve the WPTERD-Egy problem are described in Section IV. In Section V, the NP-hard property of the ETTS problem and the WPTERD-Time problem are proved. In Section VI, we provide the LNSWL-SS algorithm for solving the ETTS problem, prove its approximation ratios, and propose ETCS-S algorithm, which together with LNSWL-SS, solves the WPTERD-Time problem approximately. The whole algorithm GC-EgyTimeD which combines ETCS-S, LNSWL-SS, and LP solver is provided in Section VII. Simulation results are provided and discussed in Section VIII. Finally, we conclude this work 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 [6], focusing on a WSN containing a fixed wireless charger (WC) as the only energy source for the network, the problem of distributing the energy injected from the access point among all nodes through multi-hop WPT was investigated. In the approach there, the energy transfers in the network 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 [7]. In this work, considering the energy demand 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 assumed to be located at network nodes), a shortest path tree rooted at this location that covers all the nodes is constructed using the Dikjstra's algorithm. Then in the second step, the problem is transformed to a Mixed Integer Linear Programming (MILP) problem making use of the trees constructed in the first step, and the MILP is solved using mature optimization software packages.

The works in [4], [8] 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 more. The authors showed that CCSP is NP-hard, and proposed a greedy algorithm based on sub-modular set cover problem as well as a genetic algorithm for the CCSP. However, only one-hop WPT is considered in these works.

Multi-hop wireless power transmission is considerably different from multi-hop data transmission. For data, different transmissions are usually different since they convey different data. For energy, however, energy from any source are equivalent, we need not explicitly construct paths and restrict energy flows along a certain graph. 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 the wireless charging of WSNs is of importance.

B. Energy Redistribution Among Nodes

Some series of recent works [9], [10], [11], [12], [13] focus on a problem similar to our WPTERD problem for mobile social and sensor networks, which consist of human-carried mobile devices. In these works, energy redistribution is realized using peer-to-peer energy exchange, which happens only when nodes coming into vicinity of each other because of the moving of the person carrying the node.

Firstly in [9], the authors investigated the problem of selecting a subset of nodes to charge the rest nodes in the network such that all nodes can continue normal operations without battery depletion. In [10], the authors investigated the problem of efficiently reaching an energy balance among network nodes distributively though peer-to-peer energy exchange. The problem was investigated for two different assumptions: lossless 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 [11]. 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 [12] and [13], respectively.

In these works, energy exchange only happens when two nodes coming into vicinity of each other. As a consequence, their analyses and algorithms do not exploit the intrinsic broadcasting nature of wireless signals as well as the capability 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 in-efficiency.

III. System Model

A. Problem Model

We consider a WSN consisting of n static nodes $U=\{u_1, u_2, \ldots, u_n\}$, 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 storage of capacity limits described as function $e_{\mathrm{U}}:\mathcal{N}(n)\to\mathcal{R}^+$, *i.e.*, the capacity limit of node u_i is $e_{\mathrm{U}}(i)$. We assume node u_i's energy should always do not below its lower limit $e_{\mathrm{L}}(i)$. Besides the traditional wireless transmitter/receivers (transceivers) for data communication, these nodes are all equipped with wireless energy transceivers dedicated for transmitting/harvesting energy. Suppose node u_i currently has energy $e_{\mathrm{B}}(i)$, $i\in\mathcal{N}(n)$. We are required to redistribute the energy among the nodes in by using WPT, with the objective that node u_i's energy $e_{\mathrm{F}}(i)$ at the finish of the redistribution process should not below its expectation $e_{\mathrm{E}}(i)$, $i\in\mathcal{N}(n)$. We express the objective as $e_{\mathrm{F}}\succeq e_{\mathrm{E}}$ for short. We assume that the energy expectation of node u_i has a lower limit $e_{\mathrm{E,L}}(i)$. We also assume $e_{\mathrm{L}}\prec e_{\mathrm{E,L}}\preceq e_{\mathrm{E}}\prec e_{\mathrm{U}}$ and $e_{\mathrm{L}}\prec e_{\mathrm{B}}\prec e_{\mathrm{U}}$ always hold. For convenient, function e_{U} is sometimes called as list or vector when causing no confusion, which also applies to other similar symbols. For easy reference, we list the symbols and their meanings in Table II.

Assume node u_i always transmits energy with a constant power $p(i)$, $i\in\mathcal{N}(n)$. When node u_i is transmitting energy, the energy power harvested by node u_j from u_i's signal is expressed as $p_{\mathrm{H}}(j)=c(j,i)*p(i)$, where the energy harvesting coefficient $c(i,j)$ abstracts the effects of many factors such as the distance between the nodes, the environment, the hardware restriction, etc. 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 $c(i,i)=-1$, $i\in\mathcal{N}(n)$. We collect all energy harvesting coefficients into a matrix $\mathbf{C}=[c(i,j)]_{i,j\in\mathcal{N}(n)}$.

TABLE II
NOTATIONS AND MEANINGS.

Notation	Meaning
$\mathcal{N}(n)$	The set of positive integers $\{1,2,\ldots,n\}$;
$\mathcal{R}^+$	The set of positive real value in range $[0,+\inf]$;
$p,\mathbf{p}$	The function/list/vector of the node's constant energy transmitting power, node u_i's energy transmit power is $p(i)$;
$[.]^{\mathrm{T}}$	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 transmitting time lengths $\mathbf{t}\triangleq[t(1),t(2),\ldots,t(n)]^{\mathrm{T}}$;
$e_{\mathrm{B}},\mathbf{e}_{\mathrm{B}}$	The function/list/vector of the nodes' energy at the beginning time where it is $e_{\mathrm{B}}(i)$ for u_i. $\mathbf{e}_{\mathrm{B}}\triangleq[e_{\mathrm{B}}(1),e_{\mathrm{B}}(2),\ldots,e_{\mathrm{B}}(n)]^{\mathrm{T}}$;
$e_{\mathrm{F}},\mathbf{e}_{\mathrm{F}}$	The function/list/vector of the nodes' final energy after the energy redistribution process finishes where it is $e_{\mathrm{F}}(i)$ for u_i. $\mathbf{e}_{\mathrm{F}}\triangleq[e_{\mathrm{F}}(1),e_{\mathrm{F}}(2),\ldots,e_{\mathrm{F}}(n)]^{\mathrm{T}}$;
$e_{\mathrm{U}},\mathbf{e}_{\mathrm{U}}$	The function/list/vector of the nodes' energy upper limits where it is $e_{\mathrm{U}}(i)$ for u_i. $\mathbf{e}_{\mathrm{U}}\triangleq[e_{\mathrm{U}}(1),e_{\mathrm{U}}(2),\ldots,e_{\mathrm{U}}(n)]^{\mathrm{T}}$;
$e_{\mathrm{L}},\mathbf{e}_{\mathrm{L}}$	The function/list/vector of the nodes' energy lower limits where it is $e_{\mathrm{L}}(i)$ for u_i. $\mathbf{e}_{\mathrm{L}}\triangleq[e_{\mathrm{L}}(1),e_{\mathrm{L}}(2),\ldots,e_{\mathrm{L}}(n)]^{\mathrm{T}}$;
$e_{\mathrm{E}},\mathbf{e}_{\mathrm{E}}$	The function/list/vector of the nodes' energy expectations where it is $e_{\mathrm{E}}(i)$ for u_i. $\mathbf{e}_{\mathrm{E}}\triangleq[e_{\mathrm{E}}(1),e_{\mathrm{E}}(2),\ldots,e_{\mathrm{E}}(n)]^{\mathrm{T}}$;
$e_{\mathrm{E,L}},\mathbf{e}_{\mathrm{E,L}}$	The function/list/vector of the nodes' lower limits of energy expectations where it is $e_{\mathrm{E,L}}(i)$ for u_i. $\mathbf{e}_{\mathrm{E,L}}\triangleq[e_{\mathrm{E,L}}(1),e_{\mathrm{E,L}}(2),\ldots,e_{\mathrm{E,L}}(n)]^{\mathrm{T}}$;
$\mathbf{1}$ or $\mathbf{0}$	Proper size column vectors with elements all one or all zero, respectively;
$\mathbf{1}_{condi}$	The indication function of condition $condi$, which has value 1 when $condi$ is true, and 0 otherwise;
$N(i)$	The set of neighbors of node u_i, two nodes u_i and u_j are neighbors of each other if $c(i,j)+c(j,i)>0$;
$N_{H,\mathrm{y}}(v)/N_{H,\mathrm{n}}(v)$	The set of node v's neighbors in graph H together with/without node v itself;
$w(S)$	The accumulated weight of the nodes in S;
$\varpi(G)$	$\varpi(G)\triangleq\max_{H\subseteq G}(\min_{v\in H}w(N_{H,\mathrm{y}}(v)))$;

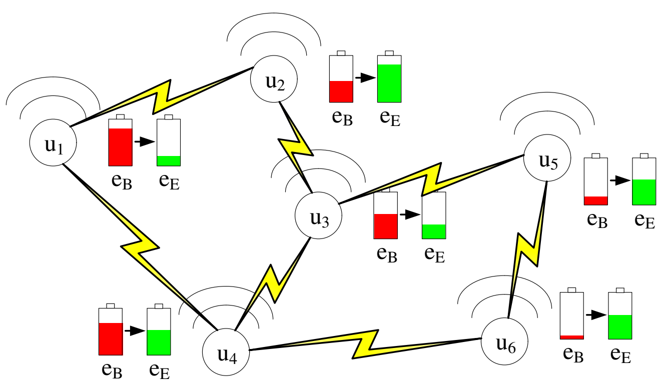


Fig. 2. An example network with WPT based energy redistribution.

We assume that the energy harvesting coefficients of the nodes are constant. Moreover, multiple simultaneous energy transmissions encountered at a node are assumed to be additive, which is the *energy harvesting additive assumption* widely adopted in the literature[15], [16], [17]. In other words, if a set of nodes $\{u_j|j\in U_{\mathrm{s}}\}$ transmit energy with power $\{p_j|j\in U_{\mathrm{s}}\}$ to another node u_j simultaneously, then the energy power harvested by u_j is $p_{\mathrm{H}}(j)=\sum_{k\in U_{\mathrm{s}}}c(j,k)*p(k)$. Harvested energy should be stored in the energy storage for later use. Excessive energy harvested by node u_i is lost when the energy storage is full of energy.

To fulfil the objective of the WPTERD problem, we have to find an energy transmission schedule $s_{\mathrm{c}}\triangleq\{s(1),s(2),\ldots\}$ where $s(i)\triangleq(u_{\mathrm{s}}(i),t_{\mathrm{b}}(i),t_{\mathrm{e}}(i))$ represents a schedule item, which means to let node $u_{\mathrm{s}}(i)\in U$ transmit energy in time slice $[t_{\mathrm{b}}(i),t_{\mathrm{e}}(i)]$. Given a schedule, if each node u_i can only exist in at most one schedule item, then this schedule is *non-preemptive*, otherwise it is *preemptive*. We mainly consider preemptive schedules in this paper. An energy transmission schedule is valid if we have $e_{\mathrm{F}}\succeq e_{\mathrm{E}}$ after performing the energy redistribution process according to the schedule. A valid schedule with *maximum accumulated final energy* $E_{\mathrm{C}}(s_{\mathrm{c}})=\sum_{i=1}^{n}e_{\mathrm{F}}(i)$ is called an *optimal schedule*. For the WPTERD problem, it is obvious that *maximizing accumulated final energy* is equivalent to *minimizing the energy loss*, hence we use the two phases alternatively for convenience. We denote $t_{\mathrm{E}}(s_{\mathrm{c}})\triangleq\max_{i:s(i)\in s_{\mathrm{c}}}t_{\mathrm{e}}(i)$ and call it the *schedule length* of s_{c}. Short schedules are preferred. In the literature of job/task scheduling, it has the *makespan* [14]. We will use the two concepts alternatively for convenience.

Formally, we state our WPTERD problem as follows. *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 e_U (vector), energy storage lower limit list e_L, energy transmitting power list p, beginning energy list e_B, expected energy list e_E, the task is to find a valid energy transmission schedule s_c with maximum accumulated final energy $E_C(s_c)$ and further with minimum makespan $t_E(s_c)$.*

B. Energy Harvesting Model

The energy harvesting model determines the energy harvesting coefficients. Although any energy harvesting model can be applied, we adopt a model where energy harvesting coefficient is determined using Eq.(1). In (1), α,β,γ are known constants determined 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 the energy transmitting power $p=1$. $\gamma\in[2,6]$ represents the channel fading effect and it is 4 in free space.

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

Our energy harvesting model extends the model in [16] in two aspects: (1) adds one new parameter γ, which consists with the popular wireless signal transmission model; (2) makes the energy coverage radius depends on the energy transmitting power, which is more practical. Furthermore, the model is not restricted to 2D or 1D space, in fact 3D space also applies.

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

IV. Solve the WPTER-Egy Problem

By analyzing some properties of the WPTERD problem, we propose a two-step approach which solves two embedded sub-problems named WPTERD-Egy and WPTERD-Time in turn. The two sub-problems focus only on the optimization in energy and time, respectively. In the first step, by formulating the WPTERD-Egy problem based on interesting property of the problem in a linear programming (LP) form, we obtain the optimal energy transmitting time lengths of the nodes leading to minimum energy loss. With the results in the first step, the remanent work of the WPTERD problem becomes the WPTERD-Time problem, which is to find a minimum makespan schedule of energy transmission operations not violating the nodes' energy limits. We call a continuous time interval of a node's energy transmission operation as an *energy transmission time slice*, hence the WPTERD-Time problem is to schedule the energy transmission time slices. We use *time slice* to represent *energy transmission time slice* by default for simplicity.

We will provide our analysis 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 $s_{\mathrm{c}}{\in}S_{\mathrm{C}}$, we can sort all items of $t_{\mathrm{b}}(i)$ and $t_{\mathrm{e}}(i)$ into a list $T_{\mathrm{s}}(s_{\mathrm{c}}){=}[t_1, t_2, \dots, t_L]$ in ascending order. Here the list is assumed to have length L. The time points in $T_{\mathrm{s}}(s_{\mathrm{c}})$ divide the time interval $[0, t_{\mathrm{L}}]$ into time slots $\{ts(1), ts(2), \dots, ts(n)\}$, where $ts(i)$ represents the slot $(t_{i-1}, t_i]$. For each slot $ts(i)$, we can obtain the set of nodes $U_{\mathrm{T}}(i)$ who are scheduled to transmit energy in slot $ts(i)$, and meanwhile all the others $U_{\mathrm{H}}(i){=}U{-}U_{\mathrm{T}}(i)$ 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 $e_i(s_{\mathrm{c}}, t)$, then it can be expressed recursively as Eq.(2).

The WPTERD-Egy problem can then be formulated as Eq.(3).

$$
\begin{array}{lll}
(\mathbf{P1}) & \max\limits_{s_{\mathrm{c}}\in S_{\mathrm{C}}} & \sum_{u_i\in U} e_i(s_{\mathrm{c}}, t_{\mathrm{E}}(s_{\mathrm{c}})) \\
 & \textit{s.t.} & e_i(s_{\mathrm{c}}, t_E(s_{\mathrm{c}})) \geq e_{\mathrm{E}}(i), \quad i\in\mathcal{N}(n); \\
 & & e_i(s_{\mathrm{c}}, t_{\mathrm{E}}(s_{\mathrm{c}})) \leq e_{\mathrm{U}}(i), \quad i\in\mathcal{N}(n); \\
 & & e_i(s_{\mathrm{c}}, t) \leq e_{\mathrm{U}}(i), \quad i\in\mathcal{N}(n), t\in[0, t_{\mathrm{E}}(s_{\mathrm{c}})]; \\
 & & e_i(s_{\mathrm{c}}, t) \geq e_{\mathrm{L}}(i), \quad i\in\mathcal{N}(n), t\in[0, t_{\mathrm{E}}(s_{\mathrm{c}})];
\end{array}
\tag{3}
$$

The WPTERD-Egy problem has several challenges: (1) it is nonlinear due to the lower and upper limits of nodes energy; (2) the solution space S_{C} is infinitely larger because that the variables t_i can take any real values from the continuous time interval $[0, t_{\mathrm{E}}(s_{\mathrm{c}})]$, and $t_{\mathrm{E}}(s_{\mathrm{c}})$ also requires 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 time of the time slots. Hence, to overcome the challenges, we will focus only on determining the optimal energy transmission time lengths leading to minimum energy loss in the WPTERD-Egy problem.

The restrictions on each node's energy of not violating the lower and upper energy limits during the energy redistribution process are postponed to the WPTERD-Time problem, and hence are ignored in the WPTERD-Egy problem.

B. Analyses of the WPTERD-Egy Problem

During the energy redistribution process, excessive energy harvested of a node is lost when its energy storage is full of energy. Given an energy transmission schedule, if some harvested energy are lost at some nodes, we say that *this schedule has energy upper limit violations*.

About energy transmission schedules with energy upper limit violations, we have the following lemma.

Lemma 1: For any valid energy transmission schedule with some energy upper limit violations, there must be some valid energy transmission schedules with more accumulated final energy.

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

- **case1:** $\exists v_i{\in}N(v_0)$ **with** $e_{\mathbf{F}}(v_i){<}e_{\mathbf{U}}(v_i)$. For the purpose to generate a better schedule with no violation, we assume each node is equipped with an auxiliary energy storage which can store the lost energy because of the violation on its energy upper limits. 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 the energy is e_{v0}) by an additional energy transmission time slice of length $e_{v0}/p(v_0)$. The 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 becomes $e'_{\mathrm{F}}(v_i){=}\max(e_{\mathrm{F}}(v_i){+}c(v_i, v_0){*}e(v_0), e_{\mathrm{U}}(v_i))$. With the given condition that $e_{\mathrm{F}}(v_i){<}e_{\mathrm{U}}(v_i)$, we have $e'_{\mathrm{F}}(v_i){>}e_{\mathrm{F}}(v_i)$. Meanwhile, all other nodes' new final energy will at least not decrease. Thus, total accumulated 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** $e_{\mathbf{F}}(v_i){=}e_{\mathbf{U}}(v_i)$. In this case, there must be a node v_1 with $e_F(v_1){<}e_{\mathrm{U}}(v_1)$ 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 also assume 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 $e'_{\mathrm{F}}(a_i){=}e_{\mathrm{F}}(a_i){=}e_{\mathrm{U}}(v_i)$, $i{\in}\mathcal{N}(k{-}1)$, $e'_{\mathrm{F}}(a_k){>}e_{\mathrm{F}}(a_k)$. The time lengths of the additional transmissions $t(a_i)$, $i{\in}\mathcal{N}(k{-}1)$, can be obtained by solving the following Eq.(4). Furthermore, we can obtain a new schedule s_1 by scheduling these additional energy transmissions to earlier times such that violations at a_i, $i{\in}\mathcal{N}(k{-}1)$, do not happen.

Combining above cases, the lemma follows. ■

Based on Lemma 1, we have the following theorem further.

$$e_i(s_{\rm c},t)=\begin{cases} e_{\rm B}(i), & t{=}0;\\ \min\left\{e_{\rm U}(i),\max\left[0,e_i(s_{\rm c},t_{j-1}){+}\left(\mathbf{1}_{i\in U_{\rm H}(j)}\sum\limits_{k\in U_{\rm T}(j)}p(k)c(k,i){-}\mathbf{1}_{i\in U_{\rm T}(j)}p(i)\right)(t{-}t_{j-1})\right]\right\}, & t{\in}(t_{j-1},t_j],\\ & j{\in}\mathcal{N}(L); \end{cases} \tag{2}$$

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

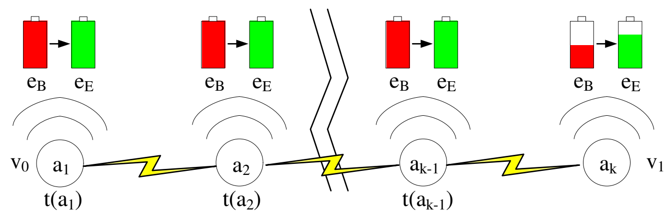


Fig. 3. The path connects v_0 and v_1 with $e_{\rm F}(v_1){<}e_{\rm U}(v_1)$.

Theorem 1: Optimal energy transmission schedules with maximum accumulated final energy (*i.e.*, minimum energy loss) must have no upper limit violations.

Proof: It can be proved easily by contradiction. If there is an optimal schedule s_1 with 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 in s_1 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 accumulated final energy $E_{\rm C}$, we have the following lemma.

Lemma 2: For the two energy transmission schedules s_1 and s_2 obtained as above and assume they both have no energy upper limit violations, we have $E_{\rm C}(s_2){>}E_{\rm C}(s_1)$.

Proof: Basing on the *energy harvesting additive assumption*, and according to Eq.(2), we analyze 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.*, $e_k(s_1,t_{\rm E}(s_1)){=}e_k(s_2,t_{\rm E}(s_2))$, $\forall k{\in}\mathcal{N}(n),k{\neq}i,j$. It is also easy to notice that, 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 lengths of u_i and u_j's energy transmitting time intervals in s_1 are respectively $t(i)$ and $t(j)$. Furthermore, assume that the total length of overlapped parts of their time slices has length $t_{\rm o}$. Thus, the energy harvested by u_i from u_j in s_1 is $c_{i,j}{*}p(j){*}(t(j){-}t_{\rm o})$. Contrastively, since that the time slices of u_i and u_j in s_2 are disjoint, 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 $e_i(s_2,t_{\rm E}(s_2)){\geq}e_i(s_1,t_{\rm E}(s_1))$. The same analysis applies to u_j and we have $e_j(s_2,t_{\rm E}(s_2)){\geq}e_j(s_1,t_E(s_1))$. With the fact $c_{i,j}{+}c_{j,i}{>}0$ for neighboring nodes u_i and u_j, we know that at least one $e_k(s_2,t_{\rm E}(s_2)){\geq}e_k(s_1,t_{\rm E}(s_1))$, $k{\in}\{i,j\}$, strictly holds. Considering the given precondition that there are no energy upper limit violations, all harvested energy will be able to be stored, hence we have Eq.(5).

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

- **case2: the set of neighboring nodes whose time slices overlap contains three or more nodes**. Without loss of generality, we assume the set contains k nodes $U_{\rm s}{=}\{u_{s1},u_{s2},\dots,u_{sk}\}$. Following the same way as in case1, we can analyze the energy of each node in $U_{\rm s}$ harvested from the other nodes in U_s. We can thus obtain that each node's harvested energy in s_2 must be greater than that in s_1. Again with the fact that there are no energy upper limit violations, we have $E_{\rm C}(s_2){>}E_{\rm C}(s_1)$.

Combining above cases, the lemma follows. ■

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

Theorem 2: Optimal energy transmission schedules with maximum accumulated 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 not disjoint. By Theorem 1, we know that s_1 must have no 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 total length of node u_i's energy transmission time slices (called as time length of u_i for short) in a schedule, and let $t\triangleq\{t(1), t(2), \ldots, t(n)\}$ be the list of all 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 not exceeding the energy storage limits when solving the WPTERD-Time problem, we only consider the restriction of the energy limits on the final energy e_{F} here.

$$e_{\mathrm{F}}(i)=e_{\mathrm{B}}(i)-p(i)t_i+\sum_{j\in\mathcal{N}(n),j\neq i} c(i,j)p(j)t_j, i\in\mathcal{N}(n). \tag{6}$$

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

$$\begin{array}{lcll} (\mathbf{P2}) & \max\limits_{t=\{t(1),t(2),\ldots,t(n)\}} & \sum_{i\in\mathcal{N}(n)} e_{\mathrm{F}}(i) & \\ & s.t. & Eq.(6); & \\ & & e_{\mathrm{F}}(i)\geq e_{\mathrm{E}}(i), & i\in\mathcal{N}(n); \\ & & e_{\mathrm{F}}(i)\leq e_{\mathrm{U}}(i), & i\in\mathcal{N}(n); \\ & & t_i\geq 0, & i\in\mathcal{N}(n). \end{array} \tag{7}$$

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

$$\begin{array}{lcl} (\mathbf{P3}) & \max\limits_{\mathbf{t}} & \mathbf{1}^{\mathrm{T}}*\mathbf{e}_{\mathrm{F}} \\ & s.t. & \mathbf{e}_{\mathrm{B}}+\mathbf{C}*\mathbf{p}*\mathbf{1}^{\mathrm{T}}*\mathbf{t}=\mathbf{e}_{\mathrm{F}}; \\ & & \mathbf{e}_{\mathrm{F}}\succeq\mathbf{e}_{\mathrm{E}}; \\ & & \mathbf{e}_{\mathrm{F}}\preceq\mathbf{e}_{\mathrm{U}}; \\ & & \mathbf{t}\succeq 0. \end{array} \tag{8}$$

Problem **P2** (*i.e.*, **P3**) is a standard linear programming problem, which can be solved efficiently using mature software optimization packages, such as Gurobi, Cplex, Nlopt, and Snopt, Knitro, Conopt, Stoaminlp, Minlpsolve.

V. Analyses of the WPTERD-Time Problem

Given an energy transmission schedule, we can group the schedule items into item sets $S^i=\{s^i_1, s^i_2, \ldots, s^i_{m_i}\}$, $i\in\mathcal{N}(n)$. Thus, S^i contains all m_i schedule items where node u_i should transmit energy. Thus, the total energy transmission time length of u_i contains m_i time slices with length $t_{\mathrm{e}}(s^i_j)-t_{\mathrm{b}}(s^i_j)$, $j\in\mathcal{N}(m_i)$.

Once the energy transmission time lengths of the nodes have been determined in the previous section, the remanent work to solve the WPTERD problem is to schedule the nodes' energy transmission time slices along the time line so that the makespan of the schedule is minimum meanwhile assuring the energy limits are not violated during the process. This problem is the WPTERD-Time problem, which is what the second step will solve. Violating the upper limits leads to energy loss thus destroying the optimality in term of maximum final energy. Violating the lower limits leads to invalid schedule.

To minimize the makespan of a schedule, we should try to make more energy transmission time slices overlap along the time line, but with the prerequisite that the time slices of neighboring nodes should not overlap with each other, otherwise the optimality in term of energy will be destroyed, as shown by 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 $\mathbf{C}=\{c(i,j)\}$, energy storage capacity limit list e_U (vector), energy storage lower limit list e_L, energy transmitting power list p, beginning energy list e_B, expected energy list e_E, the energy transmission time length vector t, the task is to find a time slice schedule s_c for the time length vector t with minimum time span, meanwhile the energy limits are not violated during the whole process*. It can be formulated formally as Eq.(9).

$$\begin{array}{ll} (\mathbf{P4}) & \min_{s_c\in S_C}\max_{s(i)\in s_c} t_{\mathrm{e}}(i) \\ s.t. & e_i(s_{\mathrm{c}}, t_{\mathrm{E}}(s_{\mathrm{c}}))\geq e_{\mathrm{E}}(i), \quad i\in\mathcal{N}(n); \\ & e_i(s_{\mathrm{c}}, t_{\mathrm{E}}(s_{\mathrm{c}}))\leq e_{\mathrm{U}}(i), \quad i\in\mathcal{N}(n); \\ & e_i(s_{\mathrm{c}}, t)\leq e_{\mathrm{U}}(i), \quad i\in\mathcal{N}(n), t\in[0, t_{\mathrm{E}}(s_{\mathrm{c}})]; \\ & e_i(s_{\mathrm{c}}, t)\geq e_{\mathrm{L}}(i), \quad i\in\mathcal{N}(n), t\in[0, t_{\mathrm{E}}(s_{\mathrm{c}})]; \\ & [t_{\mathrm{b}}(i), t_{\mathrm{e}}(i)]\cap[t_{\mathrm{b}}(j), t_{\mathrm{e}}(j)]=\emptyset, u_{\mathrm{s}}(i)=u_{\mathrm{s}}(j) \\ & \quad , s_i, s_j\in s_c; \\ & [t_{\mathrm{b}}(i), t_e(i)]\cap[t_{\mathrm{b}}(j), t_e(j)]=\emptyset, u_s(i)\neq u_s(j) \\ & , c(u_{\mathrm{s}}(i), u_{\mathrm{s}}(j))+c(u_{\mathrm{s}}(j), u_{\mathrm{s}}(i))>0, s_i, s_j\in s_{\mathrm{c}}; \\ & \sum_{\left\{\substack{s(j)=i,\\ s(j)\in s_{\mathrm{c}}}\right\}} (t_e(j)-t_b(j))=t(i), \quad i\in\mathcal{N}(n); \\ & 0\leq t_{\mathrm{b}}(i)<t_{\mathrm{e}}(i), \quad s_i\in s_{\mathrm{c}}; \end{array} \tag{9}$$

In the WPTERD-Time problem, besides the restrictions of the energy limits, 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.

$$\left\{\begin{array}{lccl} V(G)=\{ & u_i & | & i\in\mathcal{N}(n)\}, \\ E(G)=\{ & e(i,j) & | & c(i,j)+c(j,i)>0, i\in\mathcal{N}(n)\}. \end{array}\right. \tag{10}$$

Besides the restrictions of the energy limits, the core work of the WPTERD-Time problem is an energy transmission task scheduling (ETTS) problem stated as follow: *Given a set of n energy transmission tasks with time lengths $t=[t_1, t_2, \ldots, t_n]$ and the corresponding conflict graph $G(V,E)$, to find an energy transmission task schedule with minimum makespan*. We can solve the WPTERD-Time problem basing on a solution to the ETTS problem. However, 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 graph coloring (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 different colors.

For the GC problem with graph $G(V,E)$, we create an ETTS problem with $|V(G)|$ tasks all with unit time length, and meanwhile it utilizes $G(V,E)$ as its conflict graph. It is obvious that the creation of the ETTS problem instance from the GC problem instance is polynomial. Furthermore, it is easy to notice that there is a one-to-one map between the coloring solutions of the GC problem and the task schedules of the ETTS problem. To be specific, we divided the time line for the ETTS problem into time slots with unit length, then let the nodes whose corresponding vertices are colored with 1 in the coloring solution of the GC problem instance transmit energy in the 1st time slot, and then those nodes corresponding to color 2 should transmit in the 2nd slot, and so on. The makespan of the ETTS problem is just the number of the colors in the GC problem. As a conclusion, there is a polynomial reduction from the NP-hard GC problem to our ETTS problem. The theorem follows. ■

***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 build an instance of WPTERD-Time problem with its parameters constructed as follows. We can use any matric as $\mathbf{C}$ on condition that it satisfies: (1) diagonal elements take value -1; (2) all elements corresponding to edges in G take values from (0,1); (3) all other elements take value 0; (4)$c(i,j){+}c(j,i){>}0$ for all edges $(i,j){\in}E(G)$. $e_{\mathrm{L}}{=}0$, $e_{\mathrm{U}}{=}\inf$, $p{=}1$, $e_{\mathrm{B}}{=}\max(\mathbf{t}){+}1$, and $e_{\mathrm{E}}{=}e_{\mathrm{B}}{+}\mathbf{C}{*}\mathbf{t}$. This can obviously be done in polynomial time.

Then it is easy to check that any schedule to the instance of the ETTS problem is a valid energy transmission schedule to the constructed instance of the WPTERD-time problem, and vise visa. In other words, the solutions to the ETTS problem instance and those to the WPTERD-Time problem have a one-to-one map. Hence, the theorem follows. ■

VI. Solve the WPTERD-Time Problem

Theorem 3 and Theorem 4 imply that no polynomial algorithms can solve the two problems optimally. We propose an approximate algorithm to solve it in two steps. In the first step, it solves the ETTS problem to obtain the collection of energy-transmission concurrent sets. An energy-transmission concurrent set is a set of nodes that transmit energy concurrently for a certain time length, this time length value is association of the set. Step 1 just returns a collection of energy-transmission concurrent sets, whereas the sequence of the energy transmissions is not determined. Instead, final energy transmission schedule is determined in the second step based on the energy-transmission concurrent set collection meanwhile considering the energy limits.

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

Inspired by the smallest-degree-last algorithm for approximately solving the GC problem, we propose an three-step algorithm for the ETTS problem. In the first step, it determines a sequence in which nodes having least conflicting-neighbor-set-weight (CNSW) ordered at last. Here the CNSW of a node represents the sum of the weights of the node'‘s conflicting neighbors. 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 conflicting neighbors. Finally it return the list of schedule items in format of (concurrent-task-set, time-length), such a schedule item represents a decision that the nodes in concurrent-task-set should transmit energy concurrently for a period with length of time-length. We name the algorithm as Least CNSW Last Sequential Scheduling (LNSWL-SS) algorithm.

The pseudo code of LNSWL-SS is shown in Alg. 1. The while-loop between lines 2-7 implements the first stage. A property value of graph G, defined as $\varpi(G){\triangleq}\max_{H\subseteq G}(\min_{v\in H} w(N_{H,\mathrm{y}}(v))$, is also obtained accompanyingly, which denotes the largest ϖ such that G contains a subgraph H in which each node's CNSW is at least ϖ. 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) ρ times of the optimal value.

For general graphs, the graph coloring problem is NP-hard even for any fixed number of colors $k{\geq}3$ [18]. Furthermore, it is hard to approximate, *i.e.*, the problem of approximating the chromatic number with any constant ratio is also NP-hard [19]. 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 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 $m_{\mathrm{LNSWL}}{\leq}\varpi(G)$ 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 $v_{\mathrm{List}}[1{:}n]$. For the first node $v_{\mathrm{List}}(1)$, its energy transmission operation can definitely be scheduled in $[0,\varpi(G)]$. Now assume that the scheduled time slices for the nodes in $v_{\mathrm{List}}[1{:}i]$ are all in $[0,\varpi(G)]$. According to the definition of $\varpi(G)$ and the construction of the list $v_{\mathrm{List}}[1{:}n]$, we must have $w(N(v_{\mathrm{List}}(i{+}1))){\leq}\varpi(G)$, otherwise

Algorithm 1 Least Conflicting-Neighbor-Set-Weight Last Sequential Scheduling(LNSWL-SS) algorithm

Require: $G(V, E, W)$: weighted conflict graph of the problem;

Ensure: $v_{\text{List}}[1{:}n]$: the node list representing the scheduling sequence;
S_{si}: the set of schedule items;
S_{cts}: the collection of (concurrent-task-set, time-length) schedule items;
$\varpi(G)$: $\varpi(G) \triangleq \max_{H\subseteq G}(\min_{v\in H} w_{\text{N}}(H, v))$;

1: Initialize $G'{=}G(V, E)$, $\varpi(G){=}0$, $v_{\text{List}}{=}[]$;
2: **while** $G'{\neq}\emptyset$ **do**
3: $\quad\varpi(G){=}\max(\varpi(G), \min_{v_j\in V(G')} w_{\text{N}}(G', v_j))$;
4: $\quad v_i{=}\arg\min_{v_j\in V(G')} w_{\text{N}}(G', v_j)$; $v_{List} = [v_i, v_{\text{List}}]$;
5: $\quad V'{=}V(G'){-}v_i$; $E'{=}E(G'){-}\{e(i,k)|e(i,k){\in}V(G')\}$;
6: $\quad G'{=}G(V', E')$;
7: **end while**
8: **for** $i{=}1{:}n$ **do**
9: $\quad$ Assign all time slices unused by any scheduled items of the nodes in $N(G', v_i)$, (i.e., the conflicting neighboring nodes of v_i in G'), and insert the corresponding schedule items into S_{si};
10: **end for**
11: Sort the beginning and ending time values of all schedule items in S_{si} into a list $T_{\text{s}}(s_{\text{c}}){=}[t_1, t_2, \ldots, t_L]$ in ascending order(here we assume the list has length L), we thus obtain $L{-}1$ time slots with slot i occupies time interval $(t_i, t_{i+1}]$.
12: **for** $1{=}1{:}1{:}L{-}1$ **do**
13: $\quad$ Obtain the set $U_{\text{T}}(i)$ of concurrent tasks in time slot i;
14: $\quad S_{\text{cts}}(i).CTS{=}U_{\text{T}}(i)$;$S_{\text{cts}}(i).tLen{=}t_{i+1}{-}t_i$;
15: **end for**
16: **return** $S_{\text{si}}, S_{\text{cts}}, \varpi(G)$;

$\varpi(G)$ 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 [20]. First, the makespan m_{OPT} of any optimal schedule must not be smaller than the weight $w_{c,G}$ of the maximum clique, we have Eq.(11).

$$m_{\text{OPT}} \geq w_{c,G} \tag{11}$$

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).

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

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 $N(v^*, H) \overset{\text{def}}{=} H \cap N(v^*)$. The nodes on the division lines can be assumed to be in either of the two adjacent sectors deterministically.

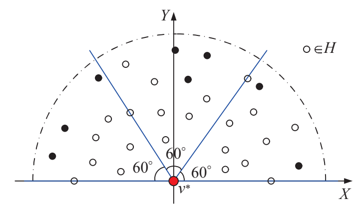


Fig. 4. Divide the top half disk into 3 sectors.

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).

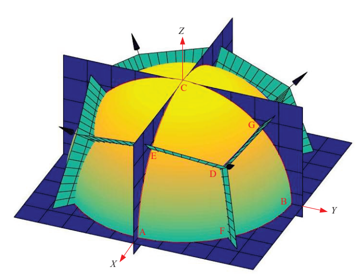


Fig. 5. Divide the top half sphere into 12 sectors.

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.

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

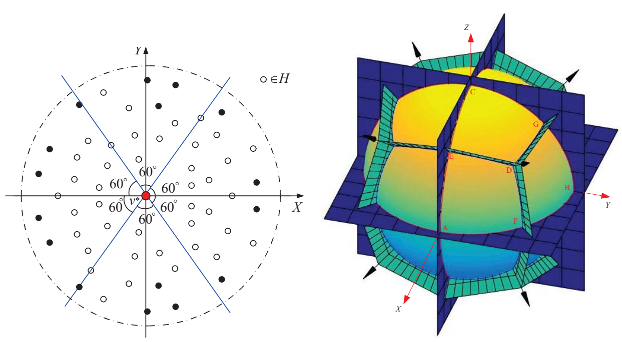
Because that the four nodes are extreme nodes in the sector, the distance between any points in the sector must be not longer than $\max\{l_{DA}, l_{DE}, l_{DF}, l_{EF}\}<r$. Thus, all nodes in each section must form a clique.

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.



(a) 6 sectors in 2D WSN (b) 24 sectors in 3D WSN

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 in Alg. 1.

Algorithm 2 The ETCS-S algorithm

Require: $S_{\text{cts}}, \mathbf{e_B}, \mathbf{e_U}, \mathbf{e_L}, \mathbf{e_E}, \mathbf{C}, \mathbf{p}$;
Ensure: TS_{ss}: the time-set schedule sequence;
1: $num(i)=0, \forall u_i\in U$; enQueue$(Q, S_{\text{cts}})$;
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: $i_2=\arg\min_{i\in s1.CTS}(e_t(i)-e_{\text{L}}(i))/p(i)$; $t_2=(e_t(i_2)-e_{\text{L}}(i_2))/p(i_2)$;
6: **if** $t_2>0$ **then**
7: $t3=\min\{t_1,t_2\}$;
8: $e_{t1}=e_t-\mathbf{C}*\mathbf{p}*tv_{s1.CTS}(t3)$;
9: **while** $\exists i\in N(s1.CTS), e_{t1}(i)>e_{\text{U}}(i)$ **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: $e_t=e_t-\mathbf{C}*\mathbf{p}*tv_{s1.CTS}(t3); e_t=\min(e_t, e_{\text{U}})$;
16: $t(i)=t(i)-t3, i\in s1.CTS$;
17: $st1.CTS=s1.CTS$; $st1.timeLen=t3$;
18: $ST_{\text{ss}}=[ST_{\text{ss}}, st1]$;$s1.timeLen=s1.timeLen-t3$;
19: $s1.CTS=\{i|i\in s1.CTS, t(i)>0\}$;
20: **if** $s1.timeLen>0$ **then**
21: enQueue$(Q, s1)$;
22: **end if**
23: **else**
24: $num(i_2)=num(i_2)+1$;
25: **if** $num(i_2)>num_{\text{lim}}$ **then**
26: $s1.CTS=\{i_2\}$; $s1.timeLen=s1.TimeLen$;
27: enQueue$(Q, s1)$;
28: **end if**
29: **end if**
30: **end while**
31: **return** ST_{ss};

The 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 in Alg. 1. 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}.

There are cases that two neighboring nodes are both with full energy $e_{\text{B}}{=}e_{\text{U}}$ before the re-distribution. In this case, the bisection search (code lines 9-14 in Alg. 2) will fail to return a positive value. In fact, there is no way to avoid EUL-violation while keeping the energy optimality. One such case is shown in Fig.7. The parameters are as follows: energy upper limit $e_{\text{U}}{=}[10,10,10]$, energy at the beginning $e_{\text{B}}{=}[10,10,1]$, energy expectation $e_{\text{E}}{=}[6,6,2]$, and energy transmission power $p{=}[1,1,1]$. It is easy to check that the optimal energy transmission time-length solution to the corresponding WPTERD-Egy problem is $t{=}[5,5,0]$. However, whichever of the two nodes, u_1 and u_2, performs the energy transmission first, EUL-violation will happen at the other node. To solve this dilemma, we compromise the energy optimality by allowing a short time period ϵ for energy loss/overflow (code lines 11-13). After that period, the two nodes can perform energy transmissions alternatively: one node keeps transmitting until the other reaches its EUL, or its energy transmission time expectation is used up. We call this scheme as ϵ-scheme. The energy changing processes along time of the nodes using this scheme are shown in Fig 8. Since that energy harvesting coefficient is quite small, the time-length of a node's continuous energy transmission time slice will increase rapidly. With small enough ϵ, the amount of energy loss may be trivial. Indeed, the energy loss approaches 0 as $\epsilon{\to}0$.

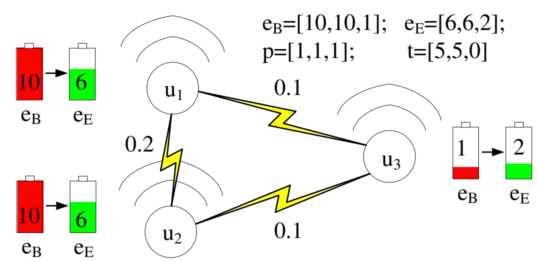


Fig. 7. An example of the WPTERD-Egy problem without practical schedules assuring energy optimality.

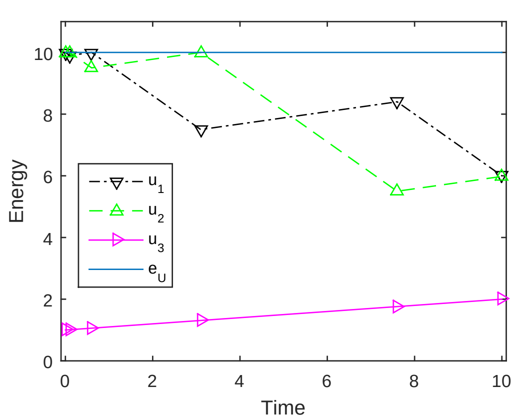


Fig. 8. The energy changing processes along time of the nodes of a schedule for the example in Fig.7.

VII. Solve the WPTERD Problem

Based on previous analyses results and the preliminary algorithms proposed, we can thus solve the WPTERD problem by decoupling the problem into WPTERD-Egy problem and WPTERD-Time problem. The two problems focus exclusively on the optimization in energy and time, respectively. Based on this approach, we propose our algorithm named Graph Coloring inspired Energy-Time Decoupling (GC-EgyTimeD) algorithm. GC-EgyTimeD firstly solve the **P3** problem using mature LP optimization tools to obtain $\mathbf{t}$. Then, it obtains S_{cts} by using the LNSWL-SS algorithm to solve the corresponding ETTS problem, and at last generates a schedule TS_{ss} from S_{cts} by using the ETCS-S algorithm. The pseudo code of GC-EgyTimeD is shown in Alg.3.

Algorithm 3 The GC-EgyTimeD algorithm

Require: $\mathbf{e_B}, \mathbf{e_U}, \mathbf{e_L}, \mathbf{e_E}, \mathbf{C}, \mathbf{p}$;
Ensure: TS_{ss}: the time-set schedule sequence;
[$\mathbf{t}$]=solveP3($\mathbf{e_B}, \mathbf{e_O}, \mathbf{e_U}, \mathbf{C}, \mathbf{p}$);%using to solve and obtain $\mathbf{t}$;
create $G(V,E,W)$ for the corresponding ETTS problem using $\mathbf{t}$;
[S_{cts}]=LNSWL-SS($G(V,E,W)$);
[TS_{ss}]=ETCS-S($S_{\text{cts}}, \mathbf{e_B}, \mathbf{e_U}, \mathbf{e_L}, \mathbf{e_E}, \mathbf{C}, \mathbf{p}$);
return TS_{ss};

By using the GC-EgyTimeD algorithm 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 but easily. Meanwhile, the broadcasting nature of radio signals is well exploited to achieve the most energy efficient way to conduct the energy redistribution.

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*, *Schedule Makespan*, and *Switch Number*. The energy loss ratio metric is calculated as $\frac{\sum_{i\in\mathcal{N}(n)}(e_{\text{B}}(i)-e_{\text{F}}(i))}{\sum_{i\in\mathcal{N}(n),e_{\text{E}}(i)>e_{\text{B}}(i)}(e_{\text{E}}(i)-e_{\text{B}}(i))}$, which implies the side 'cost' for redistribute energy in the network using the algorithm. The makespan metric represents the time span of the energy transmissions of a schedule, which reveals the time efficiency of a schedule. The switch number metric represents the number of 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 smaller energy loss, smaller schedule 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 concurrency, even for not-neighboring nodes. In AlgNoConCur, the time for each continuous energy transmission slice is determined in a way similar to that in the ETCS-S algorithm in Alg. 2. Additionally, it is obvious that the maximum clique of the weighted graph $G(V, E, W)$ of the WPTERD-Time 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 weight clique problem and use it as a baseline for performance evaluation. Thus, totally three algorithms are tested in our simulations: GC-EgyTimeD, AlgNoConCur, 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-needing nodes* η=0.3, and *the amount of energy required by these energy-needing nodes* $e_{\rm h}$=5, $e_{\rm U}$=100, $e_{\rm L}$=20. The energy harvesting coefficients are set as α=0.3, β=1, γ=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 for 2D case without sacrificing the effectiveness of the simulation results). $e_{\rm B}(i)$, $i{\in}\mathcal{N}(n)$, are randomly selected in $[e_{\rm L}, e_{\rm U})$ following the uniform distribution. Randomly select $\lceil n{*}\eta \rceil$ energy-needing nodes and set $e_{\rm E}(i)$ of each node u_i in this set to be $e_{\rm E}(i){+}e_{\rm h}$. All the others nodes' energy expectations are set to $e_{\rm L}$. 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' performance are obtained by performing a simulation set consisting of similar simulation configurations. The simulation configurations in such simulation set only difference in the value of this parameter, whereas all the other parameters take their default values. We conduct simulation experiments for inspecting the effects of the main parameters. All simulation results verify the efficiency of our algorithms. Here we only provide the simulation results for parameter n for space limitation.

In the simulation experiment for n, we let n take values in range from 10 to 100 with step size 10. The results are shown in Fig.9. The energy loss ratio switch number metric is not applicable to LBClique. Since that the other three algorithms make schedules based the shared solution t of **P3**, there are no distinctive differences in the energy loss ratio metric, as shown in Fig.9(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 can harvest energy from the single energy transmission. These two factors both lead to less energy loss when transmitting energy to others. The results in Fig.9(b) show that, GC-EgyTimeD usually obtains makespans no more than 120% of LBClique. These results imply that results of GC-EgyTimeD are nearly optimal. Compared with AlgNoConCur, GC-EgyTimeD reduces the makespan metric by more than 60% when n=100, and it becomes more effective in this metric as n increases. These results imply that GC-EgyTimeD can exploit the concurrent energy transmitting opportunities well. This is obtained at the cost of a little more operation status switches, as shown in Fig.9(c). Compared with AlgNoConCur, when n=100, GC-EgyTimeD incurs about 5% more switches.

IX. Conclusion

In this paper, we propose a two-step approach to solve the WPTERD problem by solving two embedded sub-problems WPTERD-Egy and WPTERD-Time in turn. In the first step, by formulating the WPTERD-Egy problem based on interesting property of the problem in a linear programming (LP) form, we obtain the optimal energy transmitting time lengths of the nodes leading to minimum energy loss. With the results in the first step, the remanent work of the WPTERD problem becomes the WPTERD-Time problem, which is to find a minimum makespan schedule not violating the nodes' energy limits. We prove that the WPTERD-Time problem is NP-hard, and then we propose Graph Coloring inspired Energy-Time Decoupling (GC-EgyTimeD) algorithm to solve it. Numerical simulations illustrate the effectiveness and efficiency of GC-EgyTimeD, which return schedules with minimum energy loss and approximately minimum makespan. By exploiting parallel opportunities of energy transmissions, GC-EgyTimeD reduces makespan by more than 60% when compared with a schedule without exploiting the parallel energy transmission opportunities.

Acknowledgment

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

References

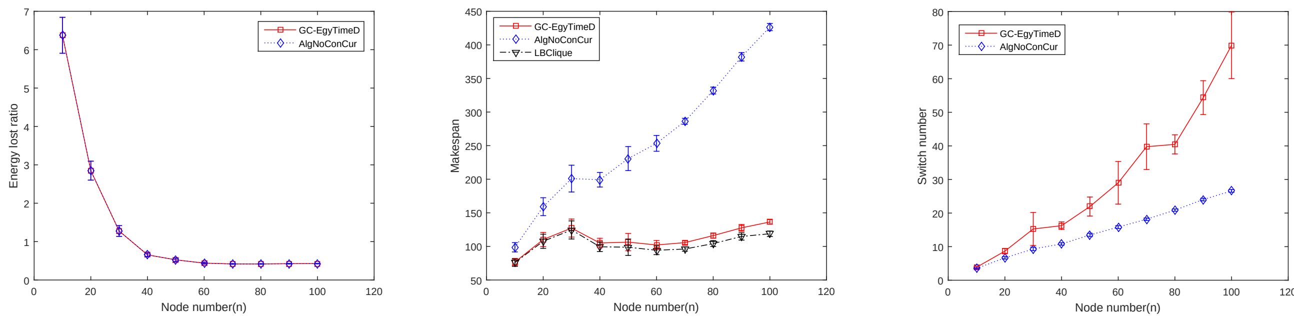
[1] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances", *Science*, vol.317, no.5834, pp.83-86, 2007.

[2] 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.

[3] K. Kang, Y. S. Meng, J. Breger, C. P. Grey, and G. Ceder, "Electrodes with high power and high capacity for rechargeable lithium batteries", *Science*, vol.311, no.5763, pp. 977-980, 2006.

[4] 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.

[5] 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.



(a)energy loss ratio (b)makespan (c)switch number

Fig. 9. The effect of node number n on performance of the algorithms.

[6] 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.

[7] T. Rault, A. Bouabdallah, Y. Challal, "Multi-hop wireless charging optimization in low-power networks," IEEE Global Communications Conference, 2013, Atlanta, United States. pp.484-489, 2013.

[8] 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.

[9] 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, TX, USA, 8-12 Dec. 2014, pp.207-212.

[10] S. Nikoletseas, 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.

[11] S. Nikoletseas, 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.

[12] A. Madhja, S. Nikoletseas, 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.

[13] A. Madhja, S. Nikoletseas, 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, CA, USA, 19-22 Mar. 2017, pp.1-6.

[14] Dániel Marx, "Graph colouring problems and their applications in scheduling," *Periodica Polytechnica SER. EL. ENG.*, vol. 48, no.1, pp.11-16, 2004.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.