

# Time-expanded Method Improving Throughput in Dynamic Renewable Networks

Jianhui Zhang<sup>\*†</sup>, Siqi Guan<sup>\*</sup>, Jiacheng Wang<sup>\*</sup>, Liming Liu<sup>\*</sup>, HanXiang Wang<sup>\*</sup> and Feng Xia<sup>†</sup>

<sup>\*</sup>College of Computer Science & Tech., Hangzhou Dianzi Univ., Hangzhou 310018 China.

<sup>†</sup> School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC 3353, Australia

<sup>‡</sup>Corresponding author e-mail: jh\_zhang@hdu.edu.cn

**Abstract**—In the Dynamic Rechargeable Networks (DRNs), the existing studies usually consider the spatio-temporal dynamics of the harvested energy so as to maximize the throughput by efficient energy allocation. However, the network dynamics have seldom been considered simultaneously including the time variable link quality, communication power and battery charge efficiency. Furthermore, the wireless interference brings extra challenge. To take these dynamics into account together, this paper studies the quite challenging problem, the network throughput maximization in the DRNs, by proper energy allocation while considering the additional affection of wireless interference. We introduce the Time-Expanded Graph (TEG) to describe the above dynamics in a feasible easy way, and then look into the scenario where there is only one pair of source-target firstly. To maximize the throughput, this paper designs the Single Pair Throughput maximization (SPT) algorithm based on TEG while considering the wireless interference. In the case of multiple pairs of source-targets, it's quite complex to solve the network throughput maximization problem directly. This paper introduces the Garg and Könemanns framework and then designs the Multiple Pairs Throughput (MPT) algorithm to maximize the overall throughput of all pairs. MPT is a fast approximation solution with the ratio of  $1-3\epsilon$ , where  $0 < \epsilon < 1$  is a small positive constant. This paper also conducts the extensive numerical evaluation based on the simulated data and the data collected by our real system. The numerical simulation results demonstrate the throughput improvement of our algorithms.

**Index Terms**—Energy-Harvesting System; Dynamic Renewable Networks; Time-Expanded Graph; Throughput Maximization

## I. INTRODUCTION

Recent developments of the energy harvesting systems, *e.g.*, solar cells, backscatter energy and ultra-low-power transceivers can support the self-sustainable and perpetual operations of the networks [1]–[6] or the energy neutral operation [7]–[9]. Such renewable networks have some real application scenarios, such as sensor networks, ad hoc networks, and edge networks. In some cases, the energy harvesting system is power-weak and its harvested energy is relatively limited sometimes compared to the demand and usually spatial-temporally dynamic [2] [8]. The previous works usually take factors like the clock synchronization [10] and the dynamic of harvested energy into account, but seldom the network dynamics at the same time, such as time variable link quality and transceiving power, and imperfect charge efficiency [3],

which have not been concerned simultaneously. When including all of the above dynamics together at the same time, it's quite challenging to explore the harvested energy fully, such as to maximize the throughput, in the Dynamic Rechargeable Networks (DRNs). Furthermore, it must inevitably avoid the wireless interference when maximizing the throughput with wireless communication. So it brings some new great challenges to the existing works on the Throughput Maximization Problem (TMP) in the DRNs [11] [12].

In this paper, we take both of the network dynamics and wireless interference into account fully and study the throughput maximization problem in the DRNs. Firstly, we consider one scenario where there is a single pair of source-target, between which there may be multiple paths. It's quite challenging to solve the TMP directly because of its NP-hardness. We then introduce the Time-Expanded Graph (TEG) to find solution in a relatively easy way. For the case of the single pair, this paper finds that the wireless interference can be clearly described with the help of TEG and be dealt with easily. We thus design the Single Pair Throughput maximization (SPT) algorithm based on TEG. For the case of the multiple pairs, each pair of source-target has its own throughput demand and the problem becomes quite challenging. We introduce the Garg and Könemanns framework [13] to solve its dual problem instead of the original problem TMP itself, which enhances us having interesting way to present the Multiple Pairs Throughput maximization (MPT) algorithm.

**Summary of key contributions.** This paper adopts TEG to design interesting solution to TMP and gives the theoretical analysis. The key contributions of this paper are list below:

- 1) This paper fully considers the factors of the network dynamics and wireless interference at the same time in the DRNs. Compared to the previous studies which only consider a few dynamic factors, this paper consider the influence of multiple complex factors on TMP problem simultaneously, which is a difficult challenge to solve. We introduce the TEG to describe the factors to reduce the complexity of the way to solve the problem.
- 2) Based on TEG, for the first scenario of single pair of source-target, this paper presents the SPT algorithm and illustrates the priority to find the maximum throughput in the DRNs. For the second scenario of the multiple pairs of source-targets, it becomes quite challenging to solve

the throughput maximization problem directly. This paper introduces the Garg and Könemanns framework and proposes the MPT algorithm based on TEG, which is an approximation solution with the ratio of  $1-3\epsilon$ .

- 3) We conduct numerical evaluation for our methods based on the simulated data and the data collected from the real system. The simulation results show that the performance of both SPT and MPT exceeds classic algorithm. Additionally, the amount of harvested energy has great impact on the throughput of both SPT and MPT.

This paper is organized as follows. The related works are reviewed in the section II. The section III presents the system model to describe the network dynamics and formulates the TMP problem. The TEG is designed and TMP is transformed to be the corresponding problem in Section IV. The SPT is designed for the single pair of source-target in the section V. While the section VI presents the MPT algorithm for multiple pairs of source-targets and gives the theoretical analysis. The performance of our solution is evaluated by the numerical simulation in the section VII. The works of this whole paper are summarized in the section VIII.

Most symbols used in this paper are summarized in Table I.

TABLE I: Symbol and meaning

Sym.	Description	Sym.	Description
$v$	Node	$N$	# of nodes
$V$	Node set	$M$	# of edges
$e$	Edge	$E$	Edge set
$T$	Period	$\tau$	Time slot
$h$	Harvested energy	$b$	Remaining energy
$G$	Graph	$\phi$	Consumed energy
$\eta$	Transmission power	$\theta$	Receiving power
$B$	Battery capacity	$c$	Edge capacity
$v(t)$	$c$ -node of $t$ at $\tau$	$f$	Throughput
$G^T$	TEG	$V^T$	TEG node set
$E^T$	TEG edge set	$F$	Set of source-target pairs
$\lambda$	Charge efficiency	$P$	Path set
$p$	Single path	$r$	Link quality
$I$	Interference function	$D$	Objective value function
$l$	Length function	$w, \epsilon, \delta$	Constants

## II. RELATED WORK

There are two closely related topics, the duty cycling and the throughput maximization in the DRNs [1] [14]. Some new works have been contributed to the two topics in renewable wireless networks in order to achieve the neutral energy operation [2] [8], or efficient energy utilization [2] [4] [8], which may be affected by some factors, such as imperfect charge efficiency and link quality [15].

### A. Renewable Network Dynamic

In the renewable networks, one direction focuses on the tiny energy harvesting, such as solar sensor network or backscatter energy harvesting [16]. Facing the relatively limited harvested energy, the duty cycle was adjusted independently in the early studies since they only counted the amount of harvested energy available in each period [7]. They estimated the short or long term dynamics of the energy source to design an

appropriate power subsystem (*i.e.*, solar panel size and energy store capacity), and then dynamically computed the sustainable performance level at runtime. Zhang *et al.* improved the energy efficiency by considering the value of information of the delivered data [2]. Some other works considered the duty cycling time cooperation among one-hop neighboring nodes by analyzing the dynamics of the harvested energy [8] [9]. Chan *et al.* developed a novel framework enabling an adaptive duty cycling scheme to allow each node to obtain key QoS metrics based on the Markov process decision by an adaptive reinforcement learning algorithm [17].

Another group of works designed the sleep-scheduling algorithms to take full advantage of the energy harvesting capability efficiently [18], or to maximize network lifetime at low latency [19], or to balance the energy consumption in networks [20]. The previous works did not pay much attention to the energy efficiency, which can be very low because of some factors, such as the imperfect charge efficiency. Zhang *et al.* studied the stochastic duty cycling under the imperfect charge efficiency when the harvested energy is stored [3]. When the factors are time-variable, such as variable charge efficiency, link quality and so on, the network is dynamic. The existing works on duty cycling or sleep-scheduling have not paid much attention to the network dynamics, which can affect greatly on the network lifetime.

### B. Throughput Maximization

A certain amount of works have been devoted to the throughput maximization in energy harvesting multi-hop networks [1] [21], but these works did not consider the network dynamics fully.

Huang *et al.* investigated the throughput maximization problem over a finite horizon of multiple transmission blocks by assuming the deterministic energy harvesting model under which the energy arrival time and the harvested amount were known prior to transmission [22]. Vaze *et al.* considered a wireless communication channel between a single energy harvesting source-target pair, to maximize the mutual information or the achievable rate between them over a fixed number of slots [23]. Xu *et al.* studied the end-to-end throughput maximization problem for optimal time and power allocation in energy harvesting cognitive radio networks in order to decrease the transmission powers of second users by employing multi-hop transmission with Time Division Multiple Access (TDMA) [24]. Mehrabi *et al.* proposed a general framework for network throughput maximization problem in energy harvesting wireless sensor networks when the data was collected from one-hop stationary sensor nodes by using a path-constrained mobile sink [25]. By considering the QoS with respect to diverse data traffic demands and communication reliability, the network throughput was optimized in MIMO-based wireless powered underground sensor networks [26]. These related works mainly studied the dynamic/deterministic energy harvesting but not the network dynamics fully.

There are another two groups of related works, rate assignment/control [27] [28] and the duty cycled routing in

renewable sensor networks or multi-hop networks [11] [12], which usually did not take wireless interference or network dynamics into account together.

### III. SYSTEM MODEL AND PROBLEM FORMULATION

#### A. System Model

This paper considers a renewable dynamic wireless network represented by a graph  $G(V, E)$ , where  $V = \{v_i, i = 1, \dots, N\}$  and  $E = \{e_{ij}, \forall v_i, v_j \in V\}$  and  $|E| = M$ .  $V$  is the set of renewable nodes, each of which can harvest environmental energy and store in its battery with the capacity  $B > 0$ . Each node also has limited storage space to delay its received or sampled data, and its size is assumed to be  $w|\tau|$ , where  $|\tau|$  is the slot length and  $w$  is a positive constant.  $E$  is a set of directional edges among the nodes with each edge associated a capacity  $c(e)$ . Each edge  $e_{ij}$  represents a directional wireless communication link from  $v_i$  to  $v_j$ .

The harvested energy is usually spatio-temporally dynamic. Furthermore, this paper also considers the network dynamics on the wireless communication and energy storage. The link quality and energy consumption on communication change over time and among different edges and nodes. The battery suffers from nonlinear charge property [29]. Let  $\eta_i(\tau)$  and  $\theta_i(\tau)$  denote the transmission and receiving power of node  $v_i$  during time slot  $\tau$ , where  $\eta_i(\tau) > 0$  and  $\theta_i(\tau) > 0$ ,  $\forall v_i \in V, \tau \in T$ , where period  $T$  is divided into  $m$  equal time slots. Let  $r_e(\tau)$  denote the link quality on edge  $e$  at time slot  $\tau$ , and  $\lambda_i(\tau)$  denote the charge efficiency of node  $v_i$  at time slot  $\tau$  when storing the harvested energy, and  $0 < \lambda_i(\tau) < 1$ .

When taking the above dynamic factors into account together, for a certain throughput  $f$  on edge  $e$ , the energy consumptions of its transmitter and receiver are different and time-variable as shown in the following equation:

$$\phi_{\text{tx}}^i(\tau) = f\eta_i(\tau)/r_e(\tau), \phi_{\text{rx}}^j(\tau) = f\theta_j(\tau)/r_e(\tau), \forall \tau \in T; \quad (1)$$

where  $\phi_{\text{tx}}^i(\tau)$  and  $\phi_{\text{rx}}^j(\tau)$  are the energy consumptions of the transmitter  $v_i$  and receiver  $v_j$  during  $\tau$ . For edge  $e_{ij}$ , its throughput is determined by the minimal one among its transmitter and receiver,  $f_{e_{ij}} = \min\{f_{v_i}, f_{v_j}\}$ . The edge throughput is time-dependent and constrained by the edge capacity as follows:

$$f_e(\tau) \leq c(e), \forall e \in E, \tau \in T; \quad (2)$$

Let  $h_i(\tau)$  denote the harvested energy during slot  $\tau$  by  $v_i$ . The harvested energy profile is defined by  $\{h_i(\tau), \tau \in T\}$ . Let  $b_i(t_k)$  denote the remaining energy of node  $v_i$  until the beginning of slot  $\tau_{k+1}$ . It should be no less than zero at any time slot. The consumed energy  $\phi_i(\tau_k)$  of each node  $v_i$  during slot  $\tau_k$  should be at most the sum of the remaining  $b_i(t_{k-1})$  at the beginning of the slot and the harvested energy  $h_i(\tau_k)$  during the slot. By introducing the variable charge efficiency, it leads to the following equations:

$$b_i(t_k) = b_i(t_{k-1}) + \lambda_i(\tau_k)[h_i(\tau_k) - \phi_i(\tau_k)]_+ - |\phi_i(\tau_k) - h_i(\tau_k)|_+$$

$$0 \leq b_i(t_k) \leq B_i, \phi_i(\tau_k) \geq 0, \quad \forall v_i \in V, \tau_k \in T; \quad (3)$$

where  $\phi_i(\tau_k)$  is the energy consumed on either transmission or receiving. The operator  $|x|_+ = x$ , if  $x > 0$ , and  $|x|_+ = 0$ , otherwise. The battery of each node is assigned with an initial value,  $b_i(t_0)$ ,  $0 \leq b_i(t_0) \leq B_i$ .

**Interference function.** Another challenging factor is the wireless interference. Let  $I$  denote the interference model. Suppose each node cannot receive and transmit data simultaneously.  $I_e$  denotes the interference set of edges which are interfered by the edge  $e$  and the edge itself.  $I'_e$  denotes the set of edges interfering the edge  $e$ . Let  $p^\circ$  denote the single path in the network  $G$ , and  $I_{p^\circ}$  denote the union of the inference sets of edges from the path  $p^\circ$ , i.e.,  $I_{p^\circ} = \cup_{e \in p^\circ} I_e$ .

#### B. Problem Formulation

This paper studies the problem how to maximize the path throughput in the DRNs. For a given pair of source-target, there may be more than one single path. The throughput  $f_{p^\circ}$  of each single path  $p^\circ$  depends on the throughput of each edge on it, and is also time-dependent.

*Definition 1 (Single Path Throughput):* Suppose that a single path  $p^\circ$  contains a set of edges  $E_{p^\circ}$ . Its throughput  $f_{p^\circ}$  is the minimum among the throughput of edges from  $E_{p^\circ}$ , i.e.,  $f_{p^\circ} = \min_{e \in E_{p^\circ}, \tau \in T} f_e(\tau)$ .

Let  $P_{st}$  be the set of paths from  $v_s$  to  $v_t$ ,  $v_s \neq v_t$ , and the throughput between them is thus  $f(v_s, v_t) = \sum_{p^\circ \in P_{st}} f_{p^\circ}$ . Given a set  $V_{st}$  of source-target node pairs, the TMP problem is formulated to maximize their overall throughput under the constraints (2) and (3) and wireless interference  $I$  as the following optimization problem:

$$\begin{aligned} \mathcal{P}_1 : \quad & \max \quad \sum_{(v_s, v_t) \in V_{st}} f(v_s, v_t) \\ & \text{s.t. Constraints (2) and (3), and } I. \end{aligned} \quad (4)$$

The problem  $\mathcal{P}_1$  is obviously NP-hard since it becomes the set cover problem if the constraints in (4) were reduced to only the wireless interference  $I$ .

### IV. TEG AND PROBLEM TRANSFORMATION

This section designs the TEG and transforms the problem  $\mathcal{P}_1$  to the one under TEG.

#### A. TEG

Let  $G^T(V^T, E^T)$  be the corresponding TEG of  $G(V, E)$ , where  $V^T$  and  $E^T$  are the node and edge sets respectively. Denote the duration between two moments  $t_{k-1}$  and  $t_k$  by time slot  $\tau_k$ . We firstly present the process to construct the TEG from  $G$  as the following steps:

- 1) Create  $m + 1$  copies,  $v(t_k), k = 0, 1, \dots, m$ , for each node  $v \in V$ . Name the newly created node as *copied node* ( $c$ -node for short) and assign it with the parameter vector  $[h(t), \eta(t), \theta(t)]$ . Enclose all  $c$ -nodes in the  $c$ -node set  $V^T$ .
- 2) Link each  $c$ -node  $v(t_k)$  to its next  $c$ -node  $v(t_{k+1})$ , and the newly created directional link is called *self-edge* ( $s$ -edge for

short). Denote it by  $e^s(\tau_{k+1})$  and set its capacity as  $c(e^s)$ . Include these  $s$ -edges into the edge set  $E^s$ .

3) For each edge  $e \in E$  in  $G$ , create a set of *copied-edges* ( $c$ -edge for short)  $e^c(\tau)$ ,  $\tau \in T$ . Assign each  $c$ -edge with a parameter vector  $[c(e^c), r_{e^c}(\tau)]$ , which are edge capacity and link quality. Include these  $c$ -edges into the edge set  $E^c$ .

By above process, we can obtain that  $V^T = \{v(t_i), i = 0, 1, \dots, m, v \in V\}$  and  $E^T = E^s \cup E^c$  in the TEG  $G^T$ . In the following context, we adopt  $e$  to represent  $e^s$  or  $e^c$  without loss of the generality. Each source or target node in the node pair set  $V_{st}$  is labelled with the extra information, the time, such as  $v_i(t)$ . The source  $v_s$  and target  $v_t$  in  $G$  correspond to  $v_s(t_0)$  and  $v_t(t_m)$  in  $G^T$  respectively. When no confusion, the following context will omit the moment labels,  $t_0$  and  $t_m$ . Let the set  $F$  denote the set of the pairs of source-targets in TEG corresponding to  $V_{st}$ . Recall that each node has limited storage space to delay the received data so the capacity of  $s$ -edge is limited by the storage space given in Section III-A. As to the two types of edges in TEG, and their capacities are different and defined as  $c(e) = |\tau|$  if  $e \in E^c$  and  $c(e) = w|\tau|$  if  $e \in E^s$ , where  $w > 1$  is a positive constant. The throughput of the paths going through one single edge is constrained by the edge's capacity and we have the following constraint based on (2).

$$\sum_{p \in F} \sum_{e \in p} f_p \leq c(e), \quad \forall \tau \in T, e \in E^T; \quad (5)$$

Figure 1(a) shows an original sample of the DRNs, which is transformed to its corresponding TEG in Figure 1(b). The detailed process is illustrated as follows. The network  $G(V, E)$  composes of  $V = \{v_i, i = 1, \dots, 4\}$  and  $E = \{e_i, i = 1, \dots, 6\}$ . In the TEG of Figure 1(b), the horizontal axis is the time-coordinate and the vertical axis is the node-coordinate. Each node is created  $m+1$  copies, *i.e.*, the donut-circle nodes in Figure 1(b). For example, the device  $v_1$  has  $m+1$  copies  $v_1(t_k)$ ,  $k = 0, \dots, m$ , *i.e.*, the  $m+1$   $c$ -nodes list horizontally in the last row. From each  $c$ -node to its next one, there is a directional  $s$ -edge such as the brown dash-line arrow from  $v_1(t_0)$  to  $v_1(t_1)$ . Each edge in  $E$  is created a series of copies, called  $c$ -edges. For example, edge  $e_2$  connects  $v_1$  to  $v_2$  in the network, and its corresponding  $c$ -edges are the ones from  $v_1(t_k)$  to  $v_2(t_{k+1})$  in TEG, where  $k = 0, 1, \dots, m-1$ .

**T-path.** After constructing the TEG, let  $t$ -path denote any

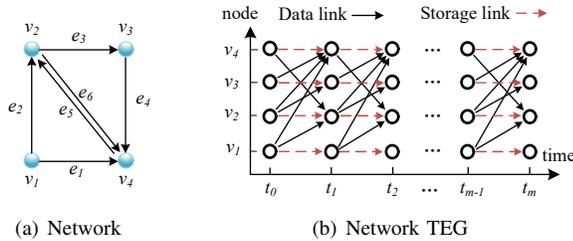


Fig. 1: (a) shows a sample network and (b) creates its TEG, in which the horizontal axis indicates the extended time and the time span from  $t_{i-1}$  to  $t_i$  represents slot  $\tau_i$ .

single path in it. For example, there are two  $t$ -paths in Figure 2, and its definition is as follows:

**Definition 2 (Single  $t$ -path):** A single  $t$ -path, denoted by  $p_i$ , is composed of a series of directional edges including the  $s$ -edges and the  $c$ -edges from the source  $c$ -node  $v_s^i(t_0)$  to the target one  $v_t^i(t_m)$ .

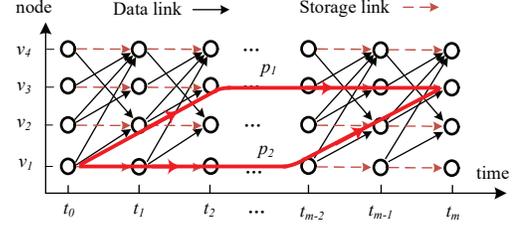


Fig. 2: There are two  $t$ -paths,  $p_1$  and  $p_2$ , in TEG.

**From single path to  $t$ -path.** Since each node has  $m$  corresponding  $c$ -nodes, each single path  $p_i^o$  in the original network  $G$  may correspond a set of  $t$ -paths  $p_i$  from the source  $v_s^i$  to the target  $v_t^i$  in TEG. For example, there is a path  $v_1 \rightarrow v_2 \rightarrow v_3$  in the network of Figure 1(a). The corresponding  $t$ -paths can be  $v_1(t_0) \rightarrow v_2(t_1) \rightarrow v_3(t_2) \dots v_3(t_m)$  or  $v_1(t_0) \rightarrow v_1(t_1) \dots \rightarrow v_2(t_{m-1}) \rightarrow v_3(t_m)$  and so on as shown in Figure 2. From the example, it's easy to notice that the repeated  $s$ -edge on a  $t$ -path means that the original node delays data to transmit at next time slot.

The original network may contain cycle but each single  $t$ -path does not contain cycle since all edges directions are along the time axis. Meanwhile, any pair of edges can interfere each other only when they are active simultaneously, *i.e.*, during the same time slot. We have the following obvious relationship between a single path and its corresponding  $t$ -path as the below claims:

**Claim 1:** Each single path has a set of corresponding  $t$ -paths, which has the following properties: (1) Any  $t$ -path does not contain cycle; (2) Any  $t$ -path in TEG contains at most  $m$   $s$ -edges and  $c$ -edges; (3) In TEG, the interference can exist only among the  $c$ -edges during same time slot.

Given a path, its throughput is determined by the bottleneck, *i.e.*, the edge with the minimum throughput among all of its edges.

**Claim 2:** The throughput of any  $t$ -path is restricted by the  $c$ -edge with the minimum throughput on it.

## B. TMP Problem under TEG

With the TEG, the wireless interference model in Section III are transformed as follows. TEG contains two types of edges:  $s$ -edge and  $c$ -edge. Each  $s$ -edge has no interference with other edges while  $c$ -edge has probability to interference with each other during same time slot. Recall the definition of  $I'_e$ .  $I'_{e^s} = \emptyset$  for any  $e^s \in E^s$  since any  $e^s$  does not create wireless signal. The interference edge set  $I'_e$  of any  $c$ -edge must contain no  $s$ -edge so it has  $I'_e \cap E^s = \emptyset, \forall e \in E^c$ . Besides the constraint in (5) caused by the throughput, the impact of the wireless interference  $I$  on each  $c$ -edge's capacity should be included,

which is given as follows:

$$f_e(\tau) + \sum_{e'(\tau) \in I'_{e(\tau)}} f_{e'}(\tau) \leq c(e), \quad \forall \tau \in T, e', e \in E^T \quad (6)$$

The constraint (6) means that the overall throughput of the  $c$ -edges in the interference range in each time slot should be no bigger than the edge capacity. Otherwise, there must be conflict among the edges. Notice that any  $s$ -edge satisfies the constraint naturally since it is not under the impact of interference model, i.e.,  $I'_{e(\tau)} = \emptyset$ . So in the constraint (6), we let  $e \in E^T$  and the constraint (6) thus is suitable for both  $s$ -edge and  $c$ -edge.

In TEG, each  $c$ -node updates its remaining energy with Equation (3). The only difference is that the variables in the equation have new meanings.  $b_i(t)$  is the remaining energy of  $c$ -node  $v_i(t)$ .  $\phi_i(t)$  is the energy consumed by  $c$ -node  $v_i(t)$  and can be either transmission or receiving energy consumption. In TEG,  $P_i$  is the set of paths from  $v_s^i$  to  $v_t^i$  and the throughput between them is thus  $f(v_s^i, v_t^i) = \sum_{p \in P_i} f_p$ , where  $(v_s, v_t)_i \in F$ .

The problem  $\mathcal{P}_1$  can thus be transformed to the corresponding one  $\mathcal{P}_2$  in TEG as follows:

$$\begin{aligned} \mathcal{P}_2 : \quad & \max \quad \sum_{(v_s, v_t)_i \in F} f((v_s, v_t)_i) \\ & \text{s.t. Constraints (3) and (6)} \\ & f_p \geq 0, \quad \forall p \in P_i, (v_s, v_t)_i \in F \end{aligned}$$

## V. SINGLE SOURCE-TARGET ALGORITHM

The preliminary question is to find the maximum throughput between a given pair of nodes, which may contain some paths between the nodes. The problem is NP-hard as stated in Section III-B and different from the traditional network flow problem, which cannot deal with the dynamic of harvested energy and wireless interference. This section presents the novel algorithm SPT to maximize the throughput for a single pair of source-target by considering the factors of the network dynamics and interference.

### A. Single T-path Construction

Firstly, this section discusses the single  $t$ -path construction to maximize its throughput by determining the following two factors, cycle avoidance and energy allocation.

**Circle avoidance.** The original network  $G$  may contain cycle while  $t$ -path in TEG as Claim 1. Any cycle on single path cannot increase the throughput since it's constrained by the bottleneck of the edge with the minimum throughput on it. Furthermore, any cycle on single path must consume extra energy since there must be at least one node spending energy on the same throughput twice or more. It's necessary to avoid any circle on any path. It's quite easy to find cycle in a  $t$ -path since it must go through some one node twice in the original network when it has one cycle. Correspondingly, the node will appear in its  $t$ -path as two  $c$ -nodes of the same node and there is at least one  $c$ -edge among them. For example, there is a path  $v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_2$  in  $G$  as the example in Figure 1. Its

corresponding  $t$ -path is  $v_2(t_0) \rightarrow v_3(t_1) \rightarrow v_4(t_2) \rightarrow v_2(t_3)$ . There are two  $c$ -nodes  $v_2(t_0)$  and  $v_2(t_3)$  of the same node  $v_2$  and there is at least one  $c$ -edge, such as  $(v_2(t_0), v_3(t_1))$ , between  $v_2(t_0)$  and  $v_2(t_4)$ . Accordingly, we can find any cycle on the corresponding  $t$ -path by the following claim.

*Claim 3 (Finding cycle):* A single path in the original network  $G$  contains cycle if there are at least two  $c$ -nodes including the source and target of the same node and exists at least one  $c$ -edge between them on its corresponding  $t$ -path.

**Energy allocation.** Each  $t$ -path contains the relay  $c$ -nodes, the source and target  $c$ -nodes. In order to maximize the throughput of each relay  $c$ -node  $v_i(t_k)$ , it has to spend energy on both transmission and receiving on single  $t$ -path properly so that it has the equal transceiving throughput. From the view of moment  $t_k$ , it acts as receiver in previous slot and as transmitter in latter slot by recalling that the duration from moment  $t_{k-1}$  to  $t_k$  indicates slot  $\tau_k$ . For example, a relay  $c$ -node  $v_i(t_k)$  is a receiver during slot  $\tau_k$  on edge  $e$  and is a transmitter during  $\tau_{k+1}$  on another edge  $e'$ . When it's the receiver on the edge  $e$ , the energy allocation depends on the remaining energy  $b_i(t_{k-1})$ , the harvested energy  $h_i(\tau_k)$  and the link quality  $r_e(\tau_k)$ . When it's the transmitter on the edge  $e'$ , it depends on  $b_i(t_k)$ ,  $h_i(\tau_{k+1})$  and  $r_{e'}(\tau_{k+1})$ . Since this section concerns the  $t$ -path containing no cycle, it goes through one same relay  $c$ -node once. The way to allocate energy is presented as the following algorithm.

---

### Algorithm 1 Energy allocation method for single $c$ -node

---

**Input:** A given  $c$ -node  $v_i(t_k)$  and the energy profile  $h_i(\tau_j), j = 1, \dots, m$ .

**Output:** The energy allocated for receiving and transmission of  $v_i(t)$ , i.e.,  $\phi_{rx}^i(t_k)$  and  $\phi_{tx}^i(t_k)$ .

---

- 1: Define  $\mu_1$  and  $\mu_2$  to be two temporary variables;
  - 2: Let  $\mu_1 = \frac{r_e(\tau_k)\eta_i(t_{k+1})}{r_{e'}(\tau_{k+1})\theta_i(t_k)}$ ;  $\mu_2 = \frac{b_i(t_{k-1})+h_i(\tau_{k+1})+h_i(\tau_k)}{1+\mu_1}$ ;
  - 3: **if**  $\mu_2 > h_i(\tau_k)$  **then**
  - 4:     **if**  $\mu_2 \leq h_i(\tau_{k-1})$  **then**
  - 5:          $\phi_{rx}^i(t_k) = [b_i(t_{k-1}) + h_i(\tau_{k+1}) + h_i(\tau_k)] / (1 + \mu_1)$ ;
  - 6:          $\phi_{tx}^i(t_k) = [b_i(t_{k-1}) + h_i(\tau_{k+1}) + h_i(\tau_k)] / (1 + 1/\mu_1)$ ;
  - 7:     **else**
  - 8:          $\phi_{rx}^i(t_k) = b_i(t_{k-1}) + h_i(\tau_k)$ ;
  - 9:          $\phi_{tx}^i(t_k) = h_i(\tau_{k+1})$ ;
  - 10:     **end if**
  - 11: **else**
  - 12:      $\phi_{rx}^i(\tau_k) = [b_i(t_k) + h_i(\tau_{k+1})] / [\mu_1 + \lambda_i(\tau_k)]$ ;
  - 13:      $\phi_{tx}^i(\tau_{k+1}) = [b_i(t_k) + h_i(\tau_{k+1})]\mu_1 / [\mu_1 + \lambda_i(\tau_k)]$ ;
  - 14: **end if**
- 

### B. Algorithm Design

The idea of SPT is to search the  $t$ -path with the maximal throughput and then to update the energy of  $c$ -nodes on it iteratively till there exists no  $t$ -path with positive throughput. Given a pair of source-target  $v_s$  and  $v_t$  in  $G$ , SPT firstly finds all  $t$ -paths from  $v_s(t_0)$  to  $v_t(t_m)$  in TEG, which are included

in the set  $P_{st}$ . According to Claim 1, TEG contains no cycle and we can adopt Breadth First Search (BFS) to find the set  $P_{st}$ . The second step is to find the maximal throughput path  $p_{max}$  according to Claim 2 by the energy allocation in Algorithm 1 and recalling Definition 1. Notice that each relay  $c$ -node  $v_i(t)$  on its  $t$ -path receives data in previous time slot and then transmits it, and thus allocates its energy by the way given in Algorithm 1. The third step is to update the remaining energy of the  $c$ -nodes on  $p_{max}$ , and the capacity of  $c$ -edges interfered by those edges on the path  $p_{max}$ . Remove any path containing the  $c$ -edges with non-positive updated capacity from  $P_{st}$ . Repeat the second and third steps till there exists no  $t$ -path with positive throughput.

We illustrate the above process by an example with Figure 2. Assume the source and target  $c$ -nodes are  $v_1(t_0)$  and  $v_3(t_m)$ . Firstly, SPT finds all  $t$ -paths from  $v_1(t_0)$  to  $v_3(t_m)$  by BFS and obtain the set  $P_{13}$ . Secondly, SPT calculates the throughput of each  $c$ -edge in  $P_{13}$  according to the available energy of its transmitter and receiver  $c$ -nodes and the throughput of each path in the set, and finds the  $t$ -path with the maximal throughput. For example, suppose that the maximal throughput  $t$ -path is  $p_1$  in Figure 2. Since  $p_1$  contains the following  $c$ -nodes:  $p_1 = v_1(t_0) \rightarrow v_2(t_1) \rightarrow v_3(t_2) \rightarrow v_3(t_3) \cdots v_3(t_m)$ . Thirdly, update the remaining energy of the following  $c$ -nodes, such as  $v_1(t_i)$ ,  $i = 0, 1, \dots, m$ ;  $v_2(t_j)$ ,  $j = 0, 1, \dots, m$ ;  $v_3(t_k)$ ,  $k = 0, 1, \dots, m$ . Suppose that the  $c$ -edge  $e$ 's capacity is  $c(e)$  and is interfered by the edge  $e(v_1(t_0), v_2(t_1))$ , which has the bottleneck throughput  $f$ . So  $e$ 's capacity  $c(e)$  is updated as  $c(e) - f$ . Repeat the above process until no  $t$ -path with positive throughput. The algorithm SPT is summarized in Algorithm 2.

### C. Algorithm Analysis.

*Theorem 1:* Algorithm 2 can be terminated in time  $O(m(M + 2N))$ .

*Proof:* The TEG has  $mM$   $c$ -edges,  $mN$   $s$ -edges and  $(m + 1)N$   $c$ -nodes totally. In Algorithm 2, the  $c$ -nodes at  $t_0$  cannot be searched besides the source  $v_s(t_0)$  so the step 3 takes  $O(m(M + 2N))$  time to implement the BFS. Each round finds at least one bottleneck  $c$ -edge so there are at most  $mM$  round to finish the “while” loop. Therefore, Algorithm 2 can be terminated at time  $O(m(M + 2N))$ . ■

Algorithm 2 provides a feasible solution to the problem  $\mathcal{P}_2$ , and accordingly to the TMP formulated in  $\mathcal{P}_1$ . The reason is that in each round of the “while” loop, the steps 5 and 8 ensure the energy consumption satisfying the energy constraint in (3) and capacity constraint in (5), and the step 9 deals with wireless interference by reducing the capacity of each interfered  $c$ -edge so as to saturate the constraint in (6).

## VI. MULTIPLE SOURCE-TARGETS SCHEME

When there are multiple pairs of source-targets, each of which has its own throughput demand, this section introduces Garg and Könemanns framework [13] to design a simple and fast approximation solution to the problem  $\mathcal{P}_2$ , which is formulated as the maximum concurrent flow problem.

---

### Algorithm 2 SPT

---

**Input:**  $G^T(V^T, E^T)$ ,  $F$ , and  $I$ .

**Output:** The set  $P_{st}$  of  $t$ -paths to maximize the overall throughput from  $v_s(t_0)$  and  $v_t(t_m)$ .

- 1: Initially the flow  $f = 0$  for all edges in  $G^T$ ;
  - 2: Let  $P_{st} = \phi$ , and define a temporary path set  $P' = \phi$ ;
  - 3: Search all  $t$ -paths from  $v_s$  and  $v_t$  by BFS, and include them into the set  $P'$ .
  - 4: Exclude the paths containing cycle from  $P'$  by the method in Claim 3.
  - 5: Calculate the throughput of each  $c$ -edge, accordingly that of each  $t$ -path, in  $P_{st}$  with its maximum available energy according to Algorithm 1;
  - 6: **while**  $P' \neq \phi$  **do**
  - 7: Find the  $t$ -path  $p_{max}$  with the maximum throughput  $f_{max}$  in  $P_{st}$ ;
  - 8: Update the remaining energy of each  $c$ -node on  $p_{max}$  according to the allocated energy calculated in Step 5, and update the remaining capacity of the edges on  $f_{max}$ ;
  - 9: Update the capacity of each  $c$ -edge  $e$  interfered by those on  $p_{max}$ , i.e.,  $c(e) - f_{max}$ ,  $\forall e \in I_{p_{max}}$ ;
  - 10: Move  $p_{max}$  from  $P'$  to  $P_{st}$ ;
  - 11: **end while**
- 

### A. Garg and Könemanns Framework

Garg and Könemanns framework was proposed originally for the maximum concurrent flow problem, where each commodity flow has its own flow demand. Given a directed network  $G^T(V^T, E^T)$  with the set  $F$  of source-target pairs  $(v_s, v_t)_i \in F$ ,  $1 \leq i \leq K$ , each pair of source-target  $(v_s, v_t)_i$  has its own throughput demand  $d_i$ . The maximum concurrent flow problem in this paper is to find flow paths  $P_i$  for each pair  $(v_s, v_t)_i$  under the constraint of throughput demand and to maximize the overall throughput of all pairs. Notice that each commodity, i.e., each pair of source-target, may compose of several flows and each flow is supported by one path. Let  $P_i$  denote the set of flow paths for the pair  $(v_s, v_t)_i$  in  $G^T(V^T, E^T)$  and  $P = \cup_{(v_s, v_t)_i \in F} P_i$ . Let  $f_p$  denote the flow variable on path  $p$ ,  $p \in P$ . The maximum concurrent flow problem can be given as the original problem of  $\mathcal{P}_3$  in the following formulation:

$$\begin{array}{ll}
 \mathcal{P}_3 : \text{ Original problem} & \text{Dual problem} \\
 \max \sum \omega & D(l) \triangleq \min \sum_{e \in E^T} l(e)c(e) \\
 \text{s.t. } \sum_{e \in P_i} f_p \leq c(e), \forall e \in E^T & \text{s.t. } \sum_{e \in P} l(e) \geq z_i, \forall p \in P_i \\
 \sum_{p \in P_i} f_p \geq \omega d_i, \forall (v_s, v_t)_i \in F & \sum_{i=1}^K d_i \cdot z_i \geq 1 \\
 f_p \geq 0, \forall p \in P & l(e) \geq 0, \forall e \in E^T
 \end{array}$$

The dual problem of  $\mathcal{P}_3$  is to assign a length function  $l(e)$  to each edge  $e \in E^T$  and a nonnegative throughput variable  $z_i$  to each pair of source-target  $i$  so that the length of every path in  $P_i$  is no less than  $z_i$ , the total volume of the product of the throughput variable and demand is no less than 1, and the objective  $\sum_{e \in E^T} l(e)c(e)$  of the dual problem is minimized.

Garg and Könemann proposed an approximation algorithm for the above dual problem so as to maximize the corresponding maximum concurrent flow [13].

Garg and Könemann algorithm assigns a flow  $f$ , and a length function  $l(e)$  to each edge  $e$ . Initially,  $f = 0$ , and  $l(e) = \frac{\delta}{c(e)}$ ,  $\forall e \in E^T$ , i.e., no routed flow, where  $\delta = ((1 - \epsilon)/|E^T|)^{1/\epsilon}$  is quite small value while  $\epsilon < 1$  is previously given positive small value. The algorithm runs in phases and each phase contains several iterations. In iteration  $i$ , it routes  $d_i$  units of flow for commodity  $i$  from  $v_s^i$  to  $v_t^i$  by the following steps. In each step, it finds the shortest path  $p_i$  for the pair of source-target  $(v_s, v_t)_i \in F$  with the current edge length function  $l(e)$ , where  $e$  is either  $c$ -edge or  $s$ -edge. The bottleneck  $f_i$  of  $p_i$  then can be found and then the minimum one between  $f_i$  and remaining demand  $d_i'$  of  $(v_s, v_t)_i$  is selected to be the increased flow for the path  $p_i$ . The length of the edges on the path is accordingly updated by multiplying the length  $l(e)$  of each edge  $e$  of  $p_i$  by a factor of  $1 + \epsilon \frac{\min\{f_i, d_i'\}}{c(e)}$ . After the path length updating, let  $z_i$  be the length of the shortest path in  $P_i$ , i.e.,  $z_i = \min_{p_j \in P_i} l(p_j)$ . Let  $l_{min}^i$  denote the shortest path from  $v_s^i$  to  $v_t^i$  with the length function  $l$ . Let  $\alpha(l) = \sum_{i=1}^K d_i z_i$  for all source-target pairs in  $F$ . So minimizing  $D(l)$  under the dual constraints is equivalent to minimizing  $\beta \triangleq \min_l D(l)/\alpha(l)$  by computing edge length  $l(e)$  for the edges. The algorithm stops in limited time as given in the following lemma when the objective value is at least one, i.e.,  $D(l) \geq 1$  and  $\beta \geq 1$ . When  $\beta < 1$ , Fleischer provided a standard procedure for scaling the problem so that  $\beta \geq 1$  [30]. Notice that the final flow produced by the above procedure may exceed the edge capacities of some edges. So it needs to scale the final flow  $f$  down by the maximum  $f_m$  among the overflow edges so as to obtain a feasible solution. Because the algorithm only increases the flow along paths with the length smaller than one and the length update rule of the algorithm ensures that the length of edges on the paths is exponential in their overflow, we can conclude that this maximum overflow is not very large as the following lemmas.

**Lemma 2:** (see [13]). Let  $f$  be the flow obtained by the above process and scale it down by  $\log_{1+\epsilon} \frac{1}{\delta}$ . The final flow then is feasible.

**Lemma 3:** (see [13]). If  $\beta \geq 1$ ,  $\frac{|f|}{\log_{1+\epsilon} \frac{1}{\delta}} \geq (1 - 3\epsilon)OPT$ , where  $OPT$  is the value of the optimal flow and  $\epsilon$  is a constant.

**Lemma 4:** (see [13]). For any  $\epsilon > 0$ , there is an algorithm that can compute  $(1 - 3\epsilon)$ -approximation to the maximum concurrent flow problem in time  $O(K(mM\delta)^2)$ .

**$\mathcal{P}_2$  Transformation.** With the concept of network flow, the path associated with throughput can be treated as the network flow, and the problem  $\mathcal{P}_2$  can be rewritten as the maximum concurrent flow problem. For a pair of source-target nodes  $v_s$

and  $v_t$  in TEG, a flow  $f_{st}$  is a throughput associated path from  $v_s$  to  $v_t$ . Recall that the moment labels  $t_0$  and  $t_m$  are omitted in the following context when no confusion. Each flow ensures the constraints (3) and (6) saturated. The solution to  $\mathcal{P}_2$  is equivalent to finding a series of flows for all pairs of source-targets. This section transfers the problem  $\mathcal{P}_2$  to the maximum multicommodity flow problem with a linear program. Let  $P_i$  denote all possible flows for source-target pair  $(v_s, v_t)_i$ , and  $P$  denoted the union of all such flow sets, i.e.,  $P = \cup_{(v_s, v_t)_i \in F} P_i$ . The problem  $\mathcal{P}_2$  then becomes the following one in TEG.

$$\begin{aligned} \mathcal{P}_4 : \max \quad & \omega \\ \text{s.t.} \quad & \sum_{p \in P_i} f_p \geq \omega d_i, \quad \forall (v_s, v_t)_i \in F \\ & \text{Constraints (3) and (6)} \\ & f \geq 0, \quad \forall f \in P_i, (v_s, v_t)_i \in F \end{aligned}$$

### B. Networked Energy Allocation

It's quite complex to solve the maximum multicommodity flow problem  $\mathcal{P}_4$  directly. By recalling TEG is a directed graph and each edge in it has its own capacity, this paper can follow Garg and Könemann's framework [13]. This section presents a fast approximation algorithm, named *MPT*, for the dual problem of  $\mathcal{P}_4$  instead of the solution for the original problem itself. For a given  $t$ -path  $p$  containing several edges  $e \in p$ , the throughput of each edge is constrained not only by the constraint (6), but also by its available energy for transceiving when it acts as receiver during slot  $\tau$  and as transmitter at  $\tau+1$ . Recall  $s$ -edge does not consume energy. Let the available throughput be  $f'$  under the energy update in Equation (3). By replacing  $f$  by  $f'$  in  $\mathcal{P}_4$ , we can omit the constraint (3) and obtain an equivalent problem, denoted by  $\mathcal{P}'_4$ . The dual problem of  $\mathcal{P}'_4$  thus has the same form with the dual problem of  $\mathcal{P}_3$ .

Since the dual problem of  $\mathcal{P}_3$  is NP-hard, this paper designs the approximation algorithm *MPT* for it. The approximation algorithm for an optimization problem is usually evaluated theoretically by the metric: approximation ratio. Denote the theoretical performance of the approximation and optimal algorithms for the problem by *Appro* and *Opt* respectively. The approximation ratio of the approximation algorithm is  $\rho$  if  $\frac{Appro}{Opt} \leq \rho$  for a minimization problem, or  $\frac{Appro}{Opt} \geq \rho$  for a maximization problem.

The idea of the algorithm *MPT* for the dual of  $\mathcal{P}_3$  is to find the shortest paths between the source-targets iteratively, and to determine the available flow, i.e., throughput, through the edges on the path under the constraints of the wireless interference and the available energy. The algorithm *MPT* consists of three steps. In the first one, it finds the shortest paths for each pair of source-targets from the set  $F$  by the Dijkstra algorithm. In the second step, it allocates the energy to find the available throughput for the shortest paths and determines the final throughput under the constraints given in the dual of  $\mathcal{P}_3$ . In the third step, it updates the lengths and capacities of edges and the remaining energy of each  $c$ -node which are

contained in the founded paths. The edges interfered by those on the shortest paths are also updated. The three steps are iteratively proceed and repeated till all pairs of source-targets reach their demands or  $D(l) \triangleq \min \sum_{e \in E^c} c'(e)l(e) \geq 1$ . The details of the algorithm MPT are presented in Algorithm 3.

---

**Algorithm 3** MPT
 

---

**Input:** Information of the harvested energy of all devices and  $G^T(V^T, E^T)$ ;  $d$  for each pair of source-target; flow  $f = 0$ .

**Output:** The throughput  $f_\tau$  of each  $c$ -node  $v(t)$  in each time slot  $\tau$ .

```

1: for each pair of source-targets  $(v_s, v_t)_i \in F$  do
2:   Initialize the throughput of each pair of source-target
   by  $f_i = 0$ , and assign each pair with a demand  $d_i > 0$ ;
3: end for
4:  $D = 0$ ;  $\delta = (\frac{1-\epsilon}{|E^T|})^{1/\epsilon}$ ;
5: for each edge  $e \in E^T$  do
6:    $l(e) = \frac{\delta}{c(e)}$ ;  $D += l(e) \cdot c(e)$ ;
7: end for
8: Define a temporary demand variable for each pair  $d'$  of
   source-target  $(v_s, v_t)_i$ ;
9: while  $D < 1$  do
10:  for each  $(v_s, v_t)_i \in F$  do
11:   if  $d'_i \neq 0$  then
12:    Let  $d'_i = d_i$ ; /* $d'_i$  is the remaining demand of the
    pair  $(v_s, v_t)_i$ */
13:    Find a shortest path  $p_i$  in  $G^T$  for each  $(v_s, v_t)_i$ 
    w.r.t the length function  $l(\cdot)$ ;
14:    Allocate energy for each node on  $p_i$ , and then
    calculate the available throughput  $f'_e$  for each edge
     $e$  on  $p_i$  by the method given in Algorithm 1;
15:    Find the available bottleneck capacity  $f'_i(e')$  on  $p_i$ ,
    where  $e \in p_i$ , and let  $f'_{p_i} \leftarrow f'_i(e')$ ;
16:    Calculate the increased available throughput for the
    source-target pair  $i$ :  $\Delta f'_i \leftarrow \min\{f'_i(e), d'_i\}$ ;
17:    for each edge  $e \in p_i \cup I'_{e: e \in p_i}$  do
18:      $l(e) *= (1 + \epsilon \cdot \frac{\Delta f'_i}{c(e)})$ ;  $D += \epsilon \cdot \frac{\Delta f'_i}{c(e)}$ ;  $d'_i -=$ 
      $\Delta f'_i$ ;
19:    end for
20:  end if
21: end for
22: end while
23: for each pair  $(v_s, v_t)_i \in F$  do
24:   $f'_i = f'_i |\tau| / \log_{1+\epsilon} \frac{1+\epsilon}{\delta}$ ; /*throughput scaling*/
25: end for

```

---

In the step 17 of Algorithm 3,  $I'_{e: e \in p_i}$  is the sets of edges, which were interfered by those on the path  $p_i$  under the wireless interference model  $I$ . It means that the available throughput on all of the interfered edges is reduced. For example, in Figure 2, suppose that the  $c$ -edge  $e_{12} : v_1(t_0) \rightarrow v_2(t_1)$  interferes the edge  $e_{42} : v_4(t_0) \rightarrow v_2(t_1)$  and the throughput on  $e_{12}$  is  $f_{e_{12}}$ . The available throughput of  $e_{42}$  is then less than the maximum one between  $c(e_{42}) - f_{e_{12}}$  and 0.

We give an example to show the mechanism of Algorithm 3

in Figure 2. After the lengths of all  $c$ -edges are initialized, suppose that the  $t$ -path  $p_1$  is the shortest from the source  $v_1(t_0)$  to the target  $v_3(t_m)$ . The algorithm MPT calculates the energy allocation for all the  $c$ -nodes on  $p_1$  by the method given in Algorithm 1 and then obtains the available throughput of each  $c$ -edge of  $p_1$ . The bottleneck capacity of  $p_1$  then can be found to be  $f'(p_1)$  and the increased throughput  $\Delta f'(p_1)$  from  $v_1(t_0)$  to  $v_3(t_m)$  can be calculated. Notice that it needs to update the lengths of not only each  $c$ -edge on  $p_1$  but also the edges interfered by those on  $p_1$ , such as  $e_{42} : v_4(t_0) \rightarrow v_2(t_1)$ . With the interference model  $I$ , the edges interfered by the edges on  $p_1$  also update their maximal available throughput and all interfered edges are included in the set  $I'_{e: e \in p_i}$ . Increase  $D$  by  $\epsilon \cdot \frac{\Delta f'_i}{c(e)}$  and reduce the demand  $d_i$  for the source-target pair  $(v_s, v_t)_i$  by  $\Delta f'_i$ . MPT repeats above process till  $D \geq 1$ .

With the reference [13], the Garg and Könemann algorithm needs at most  $O(\epsilon^{-2} km \log L \cdot T_{sp})$  time, where  $L$  is the maximum number of edges on a path between any source-target pair, and  $T_{sp}$  is the time required to compute the shortest s-t path in a graph with non negative edge-weights. This paper applies the existing results to obtain the theoretical performance of Algorithm 3. With Lemmas 2, 3 and 4, and Claim 1, we can obtain the following theorem. Notice that the numbers of edges and  $c$ -nodes in TEG are  $m(M + N)$  and  $(m + 1)N$ .

**Theorem 5:** The algorithm MPT has the time complexity  $O(\epsilon^{-2} km^2 (M + N)^2 \log m \log[(m + 1)N])$  in the TEG  $G^T(V^T, E^T)$ , and gives an approximation solution with at least  $1 - 3\epsilon$  times of the optimum for any constant  $\epsilon$ , where  $0 < \epsilon \leq 1/3$ .

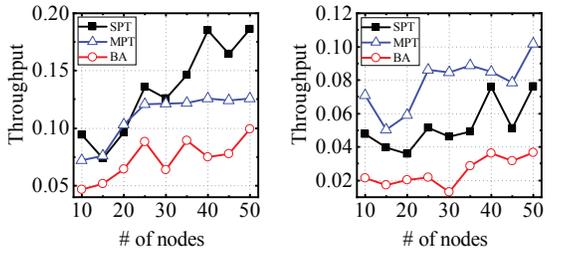
## VII. NUMERICAL EVALUATION

This section evaluates the performance of our algorithms with the numerical simulation based on the simulated data and the data collected from the real system.

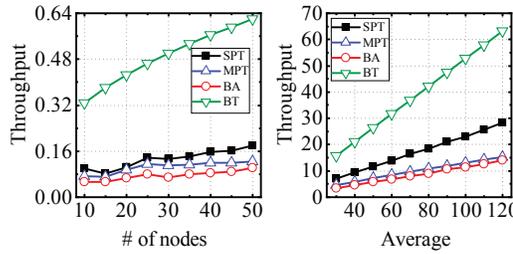
### A. Simulation Setting

**Collected data from real system.** This paper conducts the experiment with 20 solar sensor nodes, each of which composes of a solar panel, a TelosB sensor node and a controller. The controller can control the harvested energy to charge the battery or to support the TelosB. TelosB sensor node can sense and record the voltage of the solar power. The solar panel has the power of 5V and 100mA with the size of 70mm×28mm. We carry out a seven-day experiment and select out 12 typical groups of energy profile data, which records the power value of the harvested energy in 24 hours from 8 a.m to the next 8 a.m. The 12 groups represent different solar conditions, sunny in whole day, sunny in half day, cloudy in whole day, cloudy in half day, and raining day. Each group of energy profile data contains variable solar power values over time different from other groups, and has its own average and variance.

**Simulation environment.** This paper also conducts extensive simulation with different amounts of nodes deployed in a



(a) Isomorphic collected data (b) Heterogeneous collected data



(a) Throughput vs average (b) Throughput vs variance

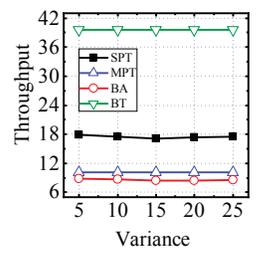


Fig. 3: The throughput under the collected data and simulated data.

Fig. 4: Harvested energy vs throughput.

$100 \times 100$  square area. The transceiving range and the interference range of each node are set to be 15 and 30 respectively. Since the transceiving powers are time variable, the simulation allows each node to set different power level at each time slot, which are generated randomly by the simulation with different average and variance. The link quality is also time variable so the simulation assigns each edge  $e$  with different quality value  $0.5 < r(e) \leq 0.95$  at each time slot that is selected randomly from 0.5 to 0.95 with interval 0.05.

There are few researches that consider network dynamics and wireless interference simultaneously. Therefore, this section gives a basic algorithm BA to present a group of typical algorithms in the energy-harvesting/renewable wireless networks. BA assumes that the transceiving powers are static and calculates the duty cycle straightforwardly, *i.e.*, the ratio of the amount of active time to the period length, according to the amount of harvested energy with or without energy harvesting prediction. This section also gives another comparison data curve, named “Best Throughput (BT)”. It is the optimal throughput among the source and the target when assuming they can use the harvested energy fully, and communicate with the minimal power and perfect link quality.

### B. Result analysis

This section analyzes the performance of our algorithms, SPT and MPT, on the throughput under different parameter settings. In the following figures, the curves of SPT for CD and MPT for CD illustrate the performance of the Algorithm SPT and MPT when the data of harvested energy are collected from the real system, where CD means the collected data and MPT curve is the throughput of each pair of source-target on average. CD contains 12 groups of harvested energy profiles. Besides the collected, the rest data of harvested energy are generated by simulation program when setting different averages and variances. Each node randomly selects one group in each sample point, which is repeated with 50 times in simulation. In SPT, BA and BT, there is only one pair of source-target. In MPT, the numbers of source-targets are randomly selected from the duration  $[2, N/2]$ .

We evaluate the performances of SPT and MPT in comparison to BA and BT based on the collected and simulated data. In Figure 3(a), the performances of BA, SPT and MPT are estimated by randomly allocating one group data to each

node from the 12 groups of the CD. The results show that SPT and MPT are better than BA on throughput. When the number of nodes increases, the advantage of SPT is more obvious than MPT while MPT keeps stable. From the figure, the throughput of BA may be consistent with that of MPT when the number of nodes becomes larger. But this situation won’t happen, because the purpose of BA is that all the collected energy is always the last to use, and there must be a loss due to the charging efficiency. While in MPT, the collected energy can be used immediately, thus reducing the energy waste. Figure 3(b) shows that the performances of SPT, MPT, BA rank always from top to bottom for heterogeneous collected data. MPT is 49.07% better than SPT, and 210.47% better than BA.

Figure 3(c) illustrates the performance of the algorithms when they all adopt the simulated data of the harvested energy. Both SPT and MPT have better performance than BA and the performance increases with the number of nodes. SPT is 77.33% better than BA, and MPT is 39.45% better than BA. Therefore, the throughput of SPT and MPT is significantly higher than that of BA. In the figure, SPT accounts for 27.14% of BT on average and MPT accounts for 21.34%. This is because BT is calculated with the optimal value of network dynamic factors, *e.g.*, best link quality and largest collected energy. While in our algorithms, except for the controllable independent variables, dynamic factors are randomly generated, so the performances of our algorithms are worse than BT.

This section also investigates the factors affecting the throughput of all algorithms by simulation. Figure 4 illustrates that the amount of harvested energy has much impact on all algorithms, where the data of harvested energy is generated randomly by simulation with different averages and variances. So each point in the figure is the average throughput in all cases where the numbers of nodes are set to be from 10 to 50 with interval 5. Figure 4(a) shows that the throughput of all algorithms increases with the average value of harvested energy. But the performance of all algorithms has no much change when the variance increases as shown in Figure 4(b).

Figure 5 plots the demand completion ratio of MPT when each point is also set to be the average of throughput in the cases of setting the horizontal axis as average or variance when the numbers of nodes are set to be from 10 to 50 with interval 5. The demand completion ratio is the ratio of the implemented

demand to the given one by MPT. It's also greatly impacted by the average of the harvested energy while the number of nodes and variance have almost no impact on it.

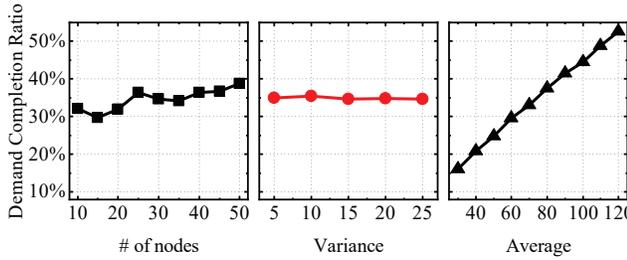


Fig. 5: Properties of demand completion ratio.

## VIII. CONCLUSION

This paper takes the network dynamics into account together including the spatio-temporally variable harvested energy, transceiving power and link quality. Meanwhile, the wireless interference and the imperfect charge efficiency are described as the time variable function of edge. The problem to find the maximum throughput between the given pairs of source-targets is shown to be NP-hard. This paper introduces TEG to give a quite new interesting way to design the algorithms respectively for the single and multiple pairs of source-targets with different demands, and finds the feasible solution for the first case and the approximation one for the second case.

## REFERENCES

- [1] Jelena Marašević, Clifford Stein, and Gil Zussman. Max-min fair rate allocation and routing in energy harvesting networks: Algorithmic analysis. In *ACM MobiHoc*, pages 367–376, Philadelphia, PA, USA, 11-14 August 2014.
- [2] Jianhui Zhang, Zhi Li, and Shaojie Tang. Value of information aware opportunistic duty cycling in solar harvesting sensor networks. *IEEE Transactions on Industrial Informatics*, 12(1):348–360, Feb 2016.
- [3] Jianhui Zhang, Siwen Zheng, Tianhao Zhang, Mengmeng Wang, and Zhi Li. Charge-aware duty cycling methods for wireless systems under energy harvesting heterogeneity. *ACM Transactions on Sensor Networks*, 16(2):15, January 2020.
- [4] Chi Lin, Yanhong Zhou, Fenglong Ma, Jing Deng, Lei Wang, and Guowei Wu. Minimizing charging delay for directional charging in wireless rechargeable sensor networks. In *IEEE INFOCOM*, pages 1819–1827, Paris, France, April 29 - May 2 2019.
- [5] Pengzhan Zhou, Cong Wang, and Yuanyuan Yang. Self-sustainable sensor networks with multi-source energy harvesting and wireless charging. In *IEEE INFOCOM*, pages 1828–1836, Paris, France, April 29 - May 2 2019.
- [6] Kunyi Chen, Hong Gao, Zhipeng Cai, Quan Chen, and Jianzhong Li. Distributed energy-adaptive aggregation scheduling with coverage guarantee for battery-free wireless sensor networks. In *IEEE INFOCOM*, pages 1018–1026, Paris, France, April 29 - May 2 2019.
- [7] Bernhard Buchli, Felix Sutton, Jan Beutel, and Lothar Thiele. Dynamic power management for long-term energy neutral operation of solar energy harvesting systems. In *ACM SenSys*, pages 31–45, Memphis, Tennessee, November 3-6 2014.
- [8] Mohammed Bahbahani and Emad Alsusa. A cooperative clustering protocol with duty cycling for energy harvesting enabled wireless sensor networks. *IEEE Transactions on Wireless Communications*, 17(1):101–111, Jan 2018.
- [9] Deepak Sharma and Amol P. Bhondekar. An improved cluster head selection in routing for solar energy-harvesting multi-heterogeneous wireless sensor networks. *Wireless Personal Communications*, 11 May 2019.
- [10] X. Zhang, H. Chen, K. Lin, Z. Wang, J. Yu, and L. Shi. Rmts: A robust clock synchronization scheme for wireless sensor networks. *Journal of Network and Computer Applications*, 135(JUN.):1–10, 2019.
- [11] Euhanna Ghadimi, Olaf Landsiedel, Pablo Soldati, Simon Duquennoy, and Mikael Johansson. Opportunistic routing in low duty-cycle wireless sensor networks. *ACM Transactions on Sensor Networks*, 10(4), 2014.
- [12] Sanam Shirazi Beheshtiha, Hwee-Pink Tan, and Masoud Sabaei. Opportunistic routing with adaptive harvesting-aware duty cycling in energy harvesting WSN. In *The 15th International Symposium on Wireless Personal Multimedia Communications*, pages 90–94, 2012.
- [13] Naveen Garg and Jochen Könenmann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *IEEE FOCS*, pages 300–339, Washington, DC, USA, 1998.
- [14] Qun Li, Javed Aslam, and Daniela Rus. Online power-aware routing in wireless ad-hoc networks. In *ACM MobiCom*, pages 97–107, Rome, Italy, 16-21 July 2001.
- [15] Andrea Castagnetti, Alain Pegatoquet, Trong Nhan Le, and Michel Auguin. A joint duty-cycle and transmission power management for energy harvesting WSN. *IEEE Transactions on Industrial Informatics*, 10(2):928–936, May 2014.
- [16] Tao Wu, Panlong Yang, Haipeng Dai, Wanru Xu, and Mingxue Xu. Charging oriented sensor placement and flexible scheduling in rechargeable wsn. In *IEEE INFOCOM*, pages 73–81, Paris, France, April 29 - May 2, 2019.
- [17] Wai Hong Ronald Chan, Pengfei Zhang, Ido Nevat, Sai Ganesh Nagarajan, Alvin C. Valera, Hwee-Xian Tan, and Natarajan Gautam. Adaptive duty cycling in sensor networks with energy harvesting using continuous-time markov chain and fluid models. *IEEE Journal on Selected Areas in Communications*, 33(12):2687–2700, Dec 2015.
- [18] Longbo Huang. Optimal sleep-wake scheduling for energy harvesting smart mobile devices. *IEEE Transactions on Mobile Computing*, 16(5):1394–1407, 2017.
- [19] Rongrong Zhang, Amiya Nayak, and Jihong Yu. Sleep scheduling in energy harvesting wireless body area networks. *IEEE Communications Magazine*, 57(2):95–101, 2019.
- [20] Mithun Mukherjee, Lei Shu, R. Venkatesha Prasad, Di Wang, and Gerhard P. Hancke. Sleep scheduling for unbalanced energy harvesting in industrial wireless sensor networks. *IEEE Communications Magazine*, 57(2):108–115, 2019.
- [21] Marios Gatzianas, Leonidas Georgiadis, and Leandros Tassioulas. Control of wireless networks with rechargeable batteries. *IEEE Transactions on Wireless Communications*, 9(2):581–593, 2010.
- [22] Chuan Huang, Rui Zhang, and Shuguang Cui. Throughput maximization for the gaussian relay channel with energy harvesting constraints. *IEEE Journal on Selected Areas in Communications*, 31(8):1469–1479, 2013.
- [23] Rahul Vaze, Rachit Garg, and Neetish Pathak. Dynamic power allocation for maximizing throughput in energy-harvesting communication system. *IEEE/ACM Transactions on Networking*, 22(5):1621–1630, 2014.
- [24] Chi Xu, Meng Zheng, Wei Liang, Haibin Yu, and Ying-Chang Liang. End-to-end throughput maximization for underlay multi-hop cognitive radio networks with rf energy harvesting. *IEEE Transactions on Wireless Communications*, 16(6):3561–3572, 2017.
- [25] Abbas Mehrabi and Kiseon Kim. General framework for network throughput maximization in sink-based energy harvesting wireless sensor networks. *IEEE Transactions on Mobile Computing*, 16(7):1881–1896, 2017.
- [26] Guanghua Liu, Zhi Sun, and Tao Jiang. Joint time and energy allocation for QoS-aware throughput maximization in mimo-based wireless powered underground sensor networks. *IEEE Transactions on Communications*, 67(2):1400–1412, 2019.
- [27] Shusen Yang and Julie A. Mccann. Distributed optimal lexicographic max-min rate allocation in solar-powered wireless sensor networks. *ACM Transactions on Sensor Networks*, 11(1):1–35, July 2014.
- [28] Hui Liang, Xiaohui Zhao, and Zan Li. Optimal energy cooperation policy in fusion center-based sustainable wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 69(6):6401–6414, 2020.
- [29] Yu Luo, Lina Pu, Yanxiao Zhao, Wei Wang, and Qing Yang. A nonlinear recursive model based optimal transmission scheduling in rf energy harvesting wireless communications. *IEEE Transactions on Wireless Communications*, 19(5):3449–3462, 2020.
- [30] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13(4):505–520, 2000.