Theoretical Computer Science 621 (2016) 73–81

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Broadcasting in weighted trees under the postal model [☆]

Yu-Hsuan Su [a], Ching-Chi Lin [b,*], D.T. Lee [a,c,d]

[a] Graduate Institute of Electronic Engineering, National Taiwan University, Taipei 10617, Taiwan
[b] Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 20224, Taiwan
[c] Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan
[d] Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan

ARTICLE INFO

Article history:
Received 14 May 2015
Received in revised form 25 January 2016
Accepted 26 January 2016
Available online 2 February 2016
Communicated by R. Klasing

Keywords:
Algorithm
Broadcast center
Weighted tree
Postal model

ABSTRACT

Given a weighted graph $G = (V, E)$, the broadcasting problem is to find a broadcast center such that the maximum communication time from the center to all vertices is minimized. For this problem, Slater et al. [29] proposed an $O(n)$-time algorithm in unweighted trees and Koh and Tcha [23] designed an $O(n \log n)$-time algorithm in unweighted trees under the telephone model. In this paper, we strengthen the results of Slater et al. by showing an $O(n)$-time algorithm in weighted trees under the postal model. The algorithm computes the set of broadcast centers, determines the broadcast time of the broadcast centers, and an optimal broadcast scheme from one of the centers to all vertices in the tree. We further show that the broadcast time of any vertex in the tree can also be determined in $O(n)$ time.

1. Introduction

The broadcasting problem has been extensively studied for several decades; see [10,16,19–21] for books and survey papers. Given a weighted graph $G = (V, E)$, the broadcasting problem is to find a broadcast center such that the maximum communication time from the center to all vertices is minimized. In this paper, we consider the broadcasting problem in a weighted tree $T = (V, E)$, where the weight $w(u, v)$ of each edge $\overline{u, v} \in E(T)$ denotes the transmission time required to transmit messages between two adjacent vertices. In communication networks, each link may need *connection time* α to set up connection between two adjacent vertices, and may have different message *transmission time* β to complete the message transmission. For practical applications in network technology, there are several kinds of communication models considered in the literature of communication studies, e.g., postal model, telephone model, and k-broadcasting model. We consider the problem under the postal model as described below.

The *postal model* [2,3,7] makes distinction between connection time α and transmission time β, where $\alpha > 0$ is assumed to be a constant and $\beta \geq 0$ varies from edge to edge. More specifically, the postal model assumes that the sender requires α time to set up a connection. After the sender sets up the connection, the sender is allowed to set up another connection to the next receiver while the sender is still transmitting the messages to the current receiver. For example, the sender v first sets up the connection with the receiver u_1 in α time, and then v can set up another connection to the next receiver u_2 while still transmitting the messages to u_1. The message transmission between two vertices is referred to as a "call". A

☆ A preliminary version of this work was presented at COCOON 2010. This work is partially supported by the Ministry of Science and Technology of Taiwan under the Grant Nos. MOST103-2221-E-001-041, MOST103-2221-E-001-042, MOST104-2221-E-001-031, and MOST104-2218-E-001-032.

* Corresponding author.
E-mail addresses: yhsuq@eda.ee.ntu.edu.tw (Y.-H. Su), lincc@mail.ntou.edu.tw (C.-C. Lin), dtlee@iis.sinica.edu.tw (D.T. Lee).

http://dx.doi.org/10.1016/j.tcs.2016.01.031

sender can only set up one connection to make a call to a receiver per α time and a sender can only transmit the messages to adjacent vertices. A call is said to be completed only after the sender completes transmitting the messages to the receiver. That is, if the transmission time β between the sender v and receiver u is $w(u, v)$, and the receiver u is the ith vertex called by the sender, then the call from v to u will be completed after $\alpha i + w(u, v)$ time. The receiver u cannot forward the messages until the sender completes the transmission.

In contrast to the postal model is the *telephone model* [1,5,23,26,29], in which the sender is allowed to set up another connection to the next receiver only after completing the transmission of messages to the current receiver. That is, only after the sender completes the transmission can the sender set up another connection to the next receiver. For example, the sender v first sets up a connection with the receiver u_1 after α time, then the sender v can only set up another connection to the next receiver u_2 after $\alpha + w(u_1, v)$ time. Meanwhile, the *k-broadcasting model* [13,14,17,24,27] is a generalization of the telephone model in which the sender is allowed to set up connections to k receivers simultaneously. For all the postal, telephone and k-broadcasting models, the receiver cannot forward the messages until the receiver finishes the receipt of the messages from the sender.

Some notations are introduced in order to give a formal definition of the broadcasting problem. Given a weighted tree $T = (V, E)$, the *broadcast time* of v, denoted as $b(v, T)$, is the minimum time required to broadcast a message from v to all vertices in T. The *broadcast time* of T, denoted as $b(T)$, is the minimum broadcast time of any vertex $v \in V(T)$, i.e., $b(T) = \min\{b(v, T) | \ v \in V(T)\}$. The *broadcast center* $BC(T)$ of T is the set of vertices with the minimum broadcast time, i.e., $BC(T) = \{v \mid v \in V(T),\ b(v, T) = b(T)\}$.

Problem Definition. Given a weighted tree $T = (V, E)$ in which the weight $w(u, v) \geq 0$ of an edge $\overline{u, v}$ represents the transmission time between them, the broadcasting problem is to compute the set of broadcast centers $BC(T)$ and determine the broadcast time $b(T)$, under the postal model with a constant connection time $\alpha > 0$. We use n to denote the size of the tree T.

1.1. Previous work

The telephone model is considered one of the most popular communication models in the literature. In 1978, Slater et al. [29] showed that computing the broadcast time of a given vertex in an arbitrary unweighted graph is NP-complete under the telephone model. Furthermore, they also provided an $O(n)$-time algorithm to compute the set of broadcast centers and determine the broadcast time for unweighted trees under the telephone model. Jakoby et al. [22] further showed that the problem is still NP-complete even when restricted to graphs with bounded degree at least 3 and planar graphs of degree 3. On the other hand, much of the research has focused on the design of approximation algorithms [7,9,11,25] and heuristics algorithms [4,12,18,28].

Furthermore, some of the research has focused on the design of polynomial-time algorithm for trees with different communication models. By adopting Slater et al.'s algorithm, Koh and Tcha [23] extended the results in [29] to provide an $O(n \log n)$-time algorithm for weighted trees under the telephone model. Dessmark et al. [6] provided algorithms to construct a minimum broadcasting scheme for almost trees and partial k-trees under the telephone model. For the k-broadcasting model, Harutyunyan et al. [17] presented a linear-time algorithm for finding the k-broadcast time of any vertex in the tree. For the p-center problem of a given unweighted tree and a given time, Farley et al. [8] showed an algorithm to determine a smallest set of subtrees covering vertices of the tree such that broadcasting can be completed within the given time in each subtree. For the nonadaptive model, Harutyunyan et al. [15] gave a polynomial-time algorithm to determine the minimum nonadaptive broadcast time for trees. Furthermore, given an nonadaptive broadcast time, they also showed how to construct the largest possible trees.

Note that our problem becomes the broadcasting problem for unweighted trees under the telephone model when $\alpha = 1$ and $\beta = 0$ for all edges.

1.2. Contributions

In this paper, we propose an $O(n)$-time algorithm for the broadcasting problem in weighted trees under the postal model. Similar to the algorithm by Koh and Tcha [23], which solved the broadcasting problem in weighted trees under the telephone model, our algorithm is based on the concept of Slater *et al.*'s algorithm [29]. But unlike their algorithm which uses a priority queue and a sorting procedure, resulting in an $O(n \log n)$-time algorithm, we develop an $O(n)$-time algorithm by using a new observation and a non-sorting labelling method. The two major refinements lead to a time complexity improvement from $O(n \log n)$ to $O(n)$. We further show that the broadcast time of any vertex in the tree can also be determined in $O(n)$ time.

For determining the broadcast centers $BC(T)$ in an unweighted tree T, Slater et al. use a bottom-up approach to iteratively update the labels of vertices and remove leaf nodes with the smallest label. Since the edges are unweighted, the value of labels are integers in $\{0, \ldots, n-1\}$. They exploit this fact to create data structures to avoid the use of a priority queue and a sorting procedure. For our broadcasting problem in a weighted tree T, since the labels are real numbers in our problem, it is not clear how to avoid the use of priority queue, which takes $O(\log n)$ time per operation, if one needs to select a vertex with the smallest label. However, we prove in Lemma 5 that it is sufficient to select between any two

Algorithm 1 Algorithm BROADCAST.

Input: A weighted tree $T = (V, E)$ with weight $w(u, v) \geq 0$ for $\overline{u, v}$ in $E(T)$.
Output: The broadcast time $b(T)$ and the set of broadcast centers $BC(T)$.
for each leaf $\ell \in T$ **do** $t(\ell) \leftarrow 0$;
let $BC(T) \leftarrow \emptyset$, $W \leftarrow \emptyset$ and $T' \leftarrow T$;
while $|V(T')| \geq 2$ **do**
  select two leaves u_x and u_y in T' arbitrarily;
  let $t(u) \leftarrow \min\{t(u_x), t(u_y)\}$, $W \leftarrow W \cup \{u\}$ and $T' \leftarrow T' - \{u\}$;
  let v be the vertex adjacent to u in T';
  if v is a leaf in T' **then**
    suppose that v is adjacent to labelled vertices $u_1, u_2, \ldots, u_k$ in W such that
    $t(u_1) + w(u_1, v) \geq t(u_2) + w(u_2, v) \geq \ldots \geq t(u_k) + w(u_k, v)$;
    let $t(v) \leftarrow \max\{t(u_i) + w(u_i, v) + \alpha i|\ 1 \leq i \leq k\}$;
  end if
end while
let κ be the only vertex left in T', $b(T) \leftarrow t(\kappa)$ and $BC(T) \leftarrow \{\kappa\}$;
let the neighbors of κ in T be $u_1, u_2, \ldots, u_k$ such that
$t(u_1) + w(u_1, \kappa) \geq t(u_2) + w(u_2, \kappa) \geq \ldots \geq t(u_k) + w(u_k, \kappa)$;
let h be the smallest integer such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$;
$BC(T) \leftarrow BC(T) \cup \{u_i \mid (1)\ w(u_i, \kappa) = 0$ and $i \leq h$; or
  $(2)\ b(u_i, T(u_i, \kappa)) = b(\kappa, T(\kappa, u_i))\}$.

labelled vertices the one with the smaller label. With this new observation, we can perform this task in $O(n)$ time. As for update of labels of vertices in T, we propose a new non-sorting procedure, and show in Theorem 10 its details.

1.3. Organization

The rest of the paper is organized as follows. Section 2 describes the algorithm BROADCAST which computes the set of broadcast centers $BC(T)$ and determines the broadcast time $b(T)$ under the postal model. In Section 3, we provide the correctness proof and complexity analysis of the algorithm. Finally, we give concluding remarks and suggest some direction for future work in Section 4.

2. Algorithm BROADCAST

The algorithm adopts a greedy strategy to process the vertices in T in a bottom-up manner. For an edge $\overline{u, v}$ in T, the removal of this edge will result in two subtrees, each of which contains v and u respectively. The subtree of T containing v is denoted as $T(v, u)$, and the subtree of T containing u is denoted as $T(u, v)$. Recall that $b(v, T)$ denotes the minimum time required to broadcast a message from v to all vertices in T. Suppose that $u_1, u_2, \ldots, u_k$ are the neighbors of v in T such that $b(u_1, T(u_1, v)) + w(u_1, v) \geq b(u_2, T(u_2, v)) + w(u_2, v) \geq \ldots \geq b(u_k, T(u_k, v)) + w(u_k, v)$. If we want to broadcast messages from v to all vertices in T, it is not difficult to see that an optimal broadcast scheme from v to its neighbors would be ordered as $u_1, u_2, \ldots, u_k$. Hence, the broadcast time from v to all vertices in T is $b(v, T) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha i|\ 1 \leq i \leq k\}$.

Based on the above observation, the concept of our algorithm BROADCAST is as below. The algorithm initially assigns a label $t(u) = 0$ to each leaf node u. It then removes a leaf u in the current tree T iteratively and assigns a label $t(v)$ to the vertex v, which becomes a leaf in the tree $T - \{u\}$. We always keep the leaf vertex with the largest label when removing a leaf node. It is one of the most significant differences of our algorithm from the algorithm by Slater et al. [29]. We will show that the tree $T - \{u\}$ still contains a broadcast center in T after removing the vertex u. Each vertex in T would be labelled exactly once except the last remaining vertex, which gets labelled twice. Let κ denote the last remaining vertex in the rest of this paper. For a vertex $v \neq \kappa$ in T, let v' be the neighbor of v such that v' is on the path from v to κ. We will show that $t(v) = b(v, T(v, v'))$, which is the minimum broadcast time from v to all vertices in $T(v, v')$. The algorithm is detailed above (Algorithm 1).

In the beginning of the algorithm, we set $t(\ell) = 0$ for each leaf ℓ in T and the set of processed vertices $W = \emptyset$. Next, in each iteration of the while loop, to keep the vertex with the largest label, we arbitrarily select two leaves u_x and u_y in the current tree T. Suppose that $t(u) = \min\{t(u_x), t(u_y)\}$ and v is the vertex adjacent to u in the current tree T, we remove u from the current tree T. If v becomes a leaf in $T - \{u\}$, we set $t(v) = \max\{t(u_i) + w(u_i, v) + \alpha i|\ 1 \leq i \leq k\}$, where $u_1, u_2, \ldots, u_k \in W$ are the neighbors of v with $t(u_i) + w(u_i, v) \geq t(u_{i+1}) + w(u_{i+1}, v)$ for $1 \leq i \leq k - 1$. After the execution of the while loop, we have only one vertex left, denoted as κ, which is one of the broadcast centers in T. Let the neighbors of κ in T be $u_1, u_2, \ldots, u_k$ such that $t(u_1) + w(u_1, \kappa) \geq t(u_2) + w(u_2, \kappa) \geq \ldots \geq t(u_k) + w(u_k, \kappa)$. Suppose that h is the smallest integer such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$. We will prove that the neighbor u_i of κ is a broadcast center in T if and only if $w(u_i, \kappa) = 0$ and $i \leq h$.

Refer to Fig. 1 for an illustrative example. Suppose that we have $\alpha = 2$. In the beginning, we set $t(u) = 0$ for each leaf u of T in Fig. 1(a). Next, by the rules of removing vertices and assigning labels, one vertex is removed from T' for each iteration of the while loop. The vertices in T are removed with respect to the ordering $v_1, v_2, \ldots, v_{11}$. For example, in Fig. 1(b), two vertices v_5 and v_9 are selected. Since $t(v_5) < t(v_9)$, v_5 was removed from T' and we have $W = \{v_1, v_2, v_3, v_4\} \cup \{v_5\}$. Refer

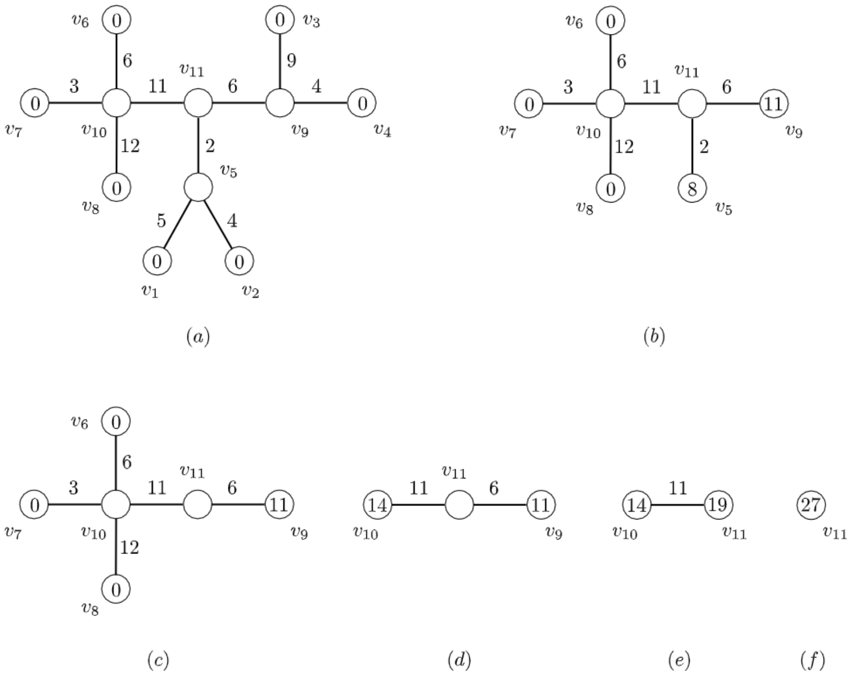

Fig. 1. The intermediate execution steps of BROADCAST algorithm. The vertices in T are removed with respect to the ordering $v_1, v_2, \ldots, v_{11}$.

to Fig. 1(c) for the result of removing v_5 from T'. After removing vertices v_6, v_7 and v_8, v_{10} becomes a leaf in T'. Then, we have $t(v_{10}) = 14$ by the rule of assigning labels in Steps 6 and 7. Refer to Fig. 1(d) for the result of labelling v_{10} in T'. Finally, we have exactly two vertices v_{10} and v_{11} left in Fig. 1(e). Again, by the rules of removing vertices and assigning labels, v_{10} was removed and v_{11} was relabelled. The vertex v_{11} is a broadcasting center of T. Notice that v_{11} has been labelled twice during the execution of algorithm, once in Fig. 1(d) and again in Fig. 1(e). According to Steps 12 and 13, we have $BC(T) = \{v_{11}\}$.

3. Correctness and complexity analysis

Let v be an arbitrary vertex in T and $u_1, \ldots, u_k$ be the neighbors of v. Let $\bar{T} = \bigcup_{1 \le i \le k} \{(v, u_i) + T(u_i, v)\}$ be a rooted tree of T with v as the root, and the neighbors of v are ordered so that $b(u_i, T(u_i, v)) + w(u_i, v) \ge b(u_{i+1}, T(u_{i+1}, v)) + w(u_{i+1}, v)$ for $1 \le i \le k-1$.

Intuitively, in order to shorten the broadcast time from v to all vertices in $\bar{T}$, we will transmit the message to u_1 first, then u_2, u_3, and so on. The following lemma shows that $u_1, \ldots, u_k$ is in fact an optimal broadcast scheme to broadcast messages from v to its neighboring vertices in $\bar{T}$. Then, it follows that the minimum time required to broadcast from v to all vertices in $\bar{T}$ is $b(v, \bar{T}) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha i \mid 1 \le i \le k\}$.

Lemma 1. *Let v be an arbitrary vertex in T and $u_1, \ldots, u_k$ be the neighbors of v. Let $\bar{T} = \bigcup_{1 \le i \le k} \{(v, u_i) + T(u_i, v)\}$ be a rooted tree of T with v as the root, and the neighbors of v are ordered so that $b(u_i, T(u_i, v)) + w(u_i, v) \ge b(u_{i+1}, T(u_{i+1}, v)) + w(u_{i+1}, v)$ for $1 \le i \le k-1$. Then, $u_1, u_2, \ldots, u_k$ is an optimal broadcast scheme to broadcast messages from v to neighboring vertices in $\bar{T}$. Consequently, $b(v, \bar{T}) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha i \mid 1 \le i \le k\}$.*

Proof. Suppose that π and π' are two permutations of $\{1, 2, \ldots, k\}$ such that v sends the message to u_i at time $\alpha\pi(i)$ and $\alpha\pi'(i)$, respectively. Suppose $\pi(i) = i$ and π' is an optimal broadcast scheme such that $b(v, \bar{T}) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha\pi'(i) \mid 1 \le i \le k\}$. Then, it suffices to show that $\max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha\pi'(i) \mid 1 \le i \le k\} \ge \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha\pi(i) \mid 1 \le i \le k\}$.

If $\pi = \pi'$, then we are done. Otherwise, let $s(\pi')$ denote the smallest index such that $\pi'(i) \ne i$. Without loss of generality, we suppose that the permutation π' is chosen so that the index $s(\pi')$ is maximized. Suppose further that we have $s(\pi') = g$ and $\pi'(h) = g$. By exchanging the elements of indices g and h in π', we get a new permutation π''. Since $\pi'(g) = \pi''(h) > g$ and $b(u_g, T(u_g, v)) + w(u_g, v) \ge b(u_h, T(u_h, v)) + w(u_h, v)$, we have $\max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha\pi'(i) \mid 1 \le i \le k\} \ge \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha\pi''(i) \mid 1 \le i \le k\}$. If $\pi'' = \pi$, then we are done. Otherwise, we have $s(\pi'') > g$, contradicting to the assumption that $s(\pi')$ is maximized. Therefore, π is an optimal broadcast scheme from v to its neighboring vertices $u_1, u_2, \ldots, u_k$ in $\bar{T}$. $\square$

Then we show an important result, on which the correctness proof of our algorithm is based, and also some useful properties for finding a broadcast center.

Lemma 2. *For each vertex v in T in Algorithm BROADCAST, we have $t(v) = b(v, T(v, v'))$, where v' is the neighbor of v on the path from v to κ.*

Proof. Let $v_1, v_2, \ldots, v_n$ be the elimination order of vertices in T such that v_i is removed from T' before v_j if $i < j$. We prove the statement by induction on the number of vertices. Let vertex v'_1 be the neighbor of v_1 on the path from v_1 to $\kappa = v_n$. Clearly, v_1 is a leaf in T and so we have $t(v_1) = b(v_1, T(v_1, v'_1)) = 0$. Suppose that the statement holds for $i \le k$. We consider $i = k + 1$ below.

We first consider the case when v_{k+1} is a leaf in T. Clearly, $t(v_{k+1}) = b(v_{k+1}, T(v_{k+1}, v'_{k+1})) = 0$, where v'_{k+1} is the neighbor of v_{k+1} on the path from v_{k+1} to κ. Next, we consider the case when v_{k+1} is an internal vertex in T. Suppose that $v'_{k+1}, u_1, \ldots, u_\ell$ are the neighbors of v_{k+1} in T, and without loss of generality, let us assume that $t(u_i) + w(u_i, v) \ge t(u_{i+1}) + w(u_{i+1}, v)$, for $i = 1, \ldots, \ell - 1$. Notice that $u_i \in \{v_1, \ldots, v_k\}$ for $1 \le i \le \ell$. By induction hypothesis, we have $t(u_i) = b(u_i, T(u_i, v_{k+1}))$ for $1 \le i \le \ell$. Therefore, by Lemma 1, we have $b(v_{k+1}, T(v_{k+1}, v'_{k+1})) = \max\{t(u_i) + w(u_i, v_{k+1}) + \alpha i \mid 1 \le i \le \ell\}$. Meanwhile, according to Step 7 of algorithm BROADCAST, we have $t(v_{k+1}) = \max\{t(u_i) + w(u_i, v_{k+1}) + \alpha i \mid 1 \le i \le \ell\}$. It follows that $t(v_{k+1}) = b(v_{k+1}, T(v_{k+1}, v'_{k+1}))$. □

Lemma 3. *For any edge $\overline{x_1, x_2} \in E(T)$, if $b(x_1, T(x_1, x_2)) \le b(x_2, T(x_2, x_1))$, then the following two statements hold:*

1. *$b(x_1, T) = b(x_2, T(x_2, x_1)) + w(x_1, x_2) + \alpha$; and*
2. *$b(x_2, T) \le b(x_1, T)$.*

Proof. We first show the correctness of statement one. Suppose $u_1, u_2, \ldots, u_k$ are the neighbors of vertex x_1 in T such that $b(u_i, T(u_i, x_1)) + w(u_i, x_1) \ge b(u_{i+1}, T(u_{i+1}, x_1)) + w(u_{i+1}, x_1)$ for $1 \le i \le k - 1$. Since $b(x_2, T(x_2, x_1)) \ge b(x_1, T(x_1, x_2))$, we have $b(x_2, T(x_2, x_1)) + w(x_2, x_1) \ge b(x_1, T(x_1, x_2))$. According to Lemma 1, $u_1 = x_2$. Then, we have

$$\begin{aligned} b(x_1, T) &= \max\Big\{b\big(x_2, T(x_2, x_1)\big) + w(x_2, x_1) + \alpha, \\ &\qquad \max\Big\{b\big(u_i, T(u_i, x_1)\big) + w(u_i, x_1) + i\alpha \mid 2 \le i \le k\Big\}\Big\} \\ &= \max\Big\{b\big(x_2, T(x_2, x_1)\big) + w(x_1, x_2) + \alpha, b\big(x_1, T(x_1, x_2)\big) + \alpha\Big\} \\ &= b\big(x_2, T(x_2, x_1)\big) + w(x_1, x_2) + \alpha. \end{aligned}$$

Next, we show the correctness of statement two. Consider a broadcast scheme from x_2 to all vertices in T that x_2 passes the messages to x_1 in the beginning of transmission and the broadcast scheme for $T(x_2, x_1)$ and $T(x_1, x_2)$ are optimal. Clearly, if x_2 transmits messages to all vertices in T using the above broadcast scheme, then the maximum communication time will be greater than or equal to $b(x_2, T)$. Hence, we have $b(x_2, T) \le \max\{b(x_1, T(x_1, x_2)) + w(x_1, x_2) + \alpha,\ b(x_2, T(x_2, x_1)) + \alpha\}$. Further, since $b(x_1, T(x_1, x_2)) \le b(x_2, T(x_2, x_1))$, we have $b(x_2, T) \le b(x_2, T(x_2, x_1)) + w(x_1, x_2) + \alpha = b(x_1, T)$. □

Lemma 4. *For each vertex v in $T - \{\kappa\}$, we have $b(v, T) = b(v', T(v', v)) + w(v, v') + \alpha$, where v' is the neighbor of v on the path from v to κ.*

Proof. According to Lemma 3, to prove $b(v, T) = b(v', T(v', v)) + w(v, v') + \alpha$, it suffices to show that $b(v, T(v, v')) \le b(v', T(v', v))$. Suppose that κ and τ are the only two vertices left with $t(\tau) \le t(\kappa)$ when $|T'| = 2$ during the execution of the algorithm.

First, we consider the case when τ is on the path from v to κ, refer to Fig. 2(*a*) for an illustrative example. Notice that $T(v, v')$ and $T(\kappa, \tau)$ are subtrees of $T(\tau, \kappa)$ and $T(v', v)$, respectively. Therefore $b(v, T(v, v')) \le b(\tau, T(\tau, \kappa))$ and $b(\kappa, T(\kappa, \tau)) \le b(v', T(v', v))$. Since $t(\tau) \le t(\kappa)$ implies $b(\tau, T(\tau, \kappa)) \le b(\kappa, T(\kappa, \tau))$, we have $b(v, T(v, v')) \le b(\tau, T(\tau, \kappa)) \le b(\kappa, T(\kappa, \tau)) \le b(v', T(v', v))$. It then follows that $b(v, T(v, v')) \le b(v', T(v', v))$.

Next, we consider the case when τ is not on the path from v to κ, refer to Fig. 2(*b*) for an illustrative example. Since κ and τ are the only two vertices left in T' when $|T'| = 2$, according to Lemma 2, we have $b(v, T(v, v')) = t(v) \le t(\tau) = b(\tau, T(\tau, \kappa))$. Further, $T(\tau, \kappa)$ is a subtree of $T(v', v)$, which implies that $b(\tau, T(\tau, \kappa)) \le b(v', T(v', v))$. Thus, we have $b(v, T(v, v')) \le b(\tau, T(\tau, \kappa)) \le b(v', T(v', v))$, which completes the proof. □

In the following, we show that after removing a leaf u in the current tree T', $T' - \{u\}$ still contains a broadcast center by keeping the vertex with the largest label. It leads to the fact that the last remaining vertex κ is a broadcast center.

Lemma 5. *Suppose that a leaf u is deleted in the current tree T' in the ith iteration of the while loop. Then we have the following results:*

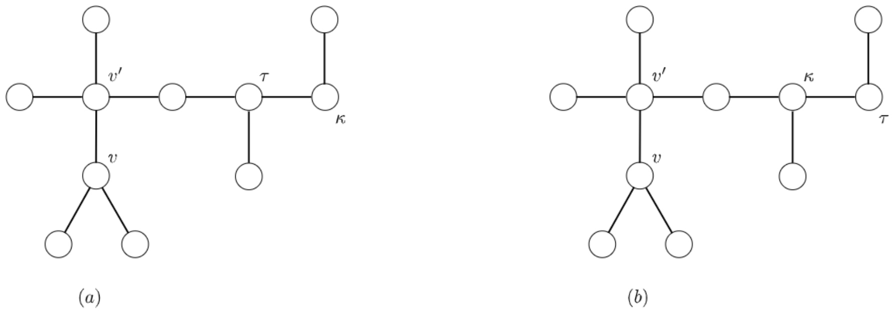

Fig. 2. An example depicts the relative positions between vertices v, τ, and κ. (a) τ is on the path from v to κ. (b) τ is not on the path from v to κ.

1. *$BC(T) \cap V(T'') \neq \emptyset$, where $T'' = T' - \{u\}$; and*
2. *the last remaining vertex $\kappa \in BC(T)$ and $b(\kappa, T) = b(T)$.*

Proof. Suppose that the leaf u is deleted in the ith iteration and v is the vertex adjacent to u in the current tree T'. To prove $BC(T) \cap V(T'') \neq \emptyset$, it suffices to show that $b(v, T) \leq b(u, T)$. We first consider the case when the current tree T contains exactly two vertices u and v. Note that by the choice of u and Lemma 2, $b(v, T(v, u)) = t(v) \geq t(u) = b(u, T(u, v))$. According to Lemma 3, since $b(v, T(v, u)) \geq b(u, T(u, v))$ with $\overline{u, v} \in E(T)$, we have $b(v, T) \leq b(u, T)$.

Next, we consider the case when the current tree T' contains at least three vertices u, v, and y. Suppose that u and y are the two leaves selected in the ith iteration of the while loop and $t(u) \leq t(y)$. Let y' be the neighbor of y on the path from y to v. Similarly, it suffices to show that $b(v, T) \leq b(u, T)$. Once again, we prove that by showing $b(v, T(v, u)) \geq b(u, T(u, v))$ according to Lemma 3. Suppose to the contrary that $b(v, T(v, u)) < b(u, T(u, v))$. Since $T(y, y')$ is a subtree of $T(v, u)$, we have $b(y, T(y, y')) \leq b(v, T(v, u))$. By $t(u) \leq t(y)$, we have $b(u, T(u, v)) = t(u) \leq t(y) = b(y, T(y, y'))$. This implies that $b(y, T(y, y')) \leq b(v, T(v, u)) < b(u, T(u, v)) \leq b(y, T(y, y'))$, a contradiction. Therefore, it is in fact that $b(v, T(v, u)) \geq b(u, T(u, v))$. Hence according to Lemma 3, since $b(v, T(v, u)) \geq b(u, T(u, v))$ with $\overline{u, v} \in E(T)$, we have $b(v, T) \leq b(u, T)$. □

The neighborhood $N(v)$ of a vertex v is the set of all vertices adjacent to v in T. Intuitively, a tree may contain more than one broadcast center. Below we show that the only candidates for being a broadcast center of T are the neighbors of κ. More specifically, we will prove that for each vertex $u_i \in N(\kappa)$, u_i is a broadcast center of T if and only if one of the following conditions holds:

1. $w(u_i, \kappa) = 0$ and $i \leq h$, where h is the smallest integer such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$; and
2. $b(u_i, T(u_i, \kappa)) = b(\kappa, T(\kappa, u_i))$.

Lemma 6. *If v is a broadcast center in T, then $v \in N(\kappa) \cup \{\kappa\}$.*

Proof. Suppose that s is a vertex in T such that $s \notin N(\kappa) \cup \{\kappa\}$ and s' is the neighbor of s such that s' is on the path from s to κ. To prove the lemma, it suffices to show that $b(s, T) > b(\kappa, T)$. Let t be the neighbor of κ such that t is on the path from s to κ. Since $T(\kappa, t)$ is a proper subtree of $T(s', s)$, $b(s', T(s', s)) \geq \alpha + w(t, \kappa) + b(\kappa, T(\kappa, t))$. Therefore, by Lemma 4, we have $b(s, T) \geq 2\alpha + w(s, s') + w(t, \kappa) + b(\kappa, T(\kappa, t))$.

Further, since κ is the only vertex left in the algorithm, we have $b(\kappa, T(\kappa, t)) \geq b(t, T(t, \kappa))$. It follows from Lemma 3 that $b(t, T) = \alpha + w(t, \kappa) + b(\kappa, T(\kappa, t))$ and $b(t, T) \geq b(\kappa, T)$. Since $\alpha > 0$, we have $b(s, T) \geq 2\alpha + w(s, s') + w(t, \kappa) + b(\kappa, T(\kappa, t)) > \alpha + w(t, \kappa) + b(\kappa, T(\kappa, t)) = b(t, T) \geq b(\kappa, T)$, which completes the proof. □

Lemma 7. *For each vertex $u_i \in N(\kappa)$, if condition (1) or condition (2) holds, then we have $u_i \in BC(T)$.*

Proof. We first prove that condition (1) $w(u_i, \kappa) = 0$ and $i \leq h$ implies $u_i \in BC(T)$. Suppose that u_i passes the messages to κ in the beginning of transmission and the broadcast scheme for κ is $(u_1, u_2, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k)$. To prove $u_i \in BC(T)$, it suffices to show that $b(u_i, T) \leq b(T)$. We first consider the vertices in $T(u_\ell, \kappa)$ when $\ell < i$. The vertices in $T(u_\ell, \kappa)$ finish the receipt of the message sent from u_i at $\alpha + w(u_i, \kappa) + t(u_\ell) + w(u_\ell, \kappa) + \alpha\ell$ time. Since $w(u_i, \kappa) = 0$ and $\ell < i \leq h$, the vertices in $T(u_\ell, \kappa)$ would finish the receipt of messages no later than $b(T)$.

Then, we consider the case when $\ell = i$. One can see that the vertices in $T(u_i, \kappa)$ finish the receipt of the message at $\alpha + t(u_i)$. Clearly, we have $\alpha + t(u_i) \leq t(u_i) + w(u_i, \kappa) + \alpha i \leq b(T)$. Therefore, the vertices in $T(u_i, \kappa)$ also finish the receipt no later than $b(T)$. Finally, we consider the case when $\ell > i$. The vertices in $T(u_\ell, \kappa)$ still finish the receipt of the message at $w(u_i, \kappa) + t(u_\ell) + w(u_\ell, \kappa) + \alpha(\ell - 1) + \alpha$ time. Since $w(u_i, \kappa) = 0$, we also have $t(u_\ell) + w(u_\ell, \kappa) + \alpha\ell \leq b(T)$. So, we have $b(u_i, T) \leq b(T)$ as desired.

Next, we prove that condition (2) $b(u_i, T(u_i, \kappa)) = b(\kappa, T(\kappa, u_i))$ implies $u_i \in BC(T)$. Since $b(u_i, T(u_i, \kappa)) = b(\kappa, T(\kappa, u_i))$, we have

$$\begin{aligned} b(\kappa, T) &= \max\{b(u_i, T(u_i, \kappa)) + w(\kappa, u_i) + \alpha, b(\kappa, T(\kappa, u_i)) + \alpha\} \\ &= \max\{b(\kappa, T(\kappa, u_i)) + w(\kappa, u_i) + \alpha, b(u_i, T(u_i, \kappa)) + \alpha\} \\ &= b(u_i, T). \end{aligned}$$

It follows that we have $u_i \in BC(T)$. □

Lemma 8. *For each vertex $u_i \in N(\kappa)$, if neither condition (1) nor condition (2) holds, then we have $u_i \notin BC(T)$.*

Proof. Note that κ is the last vertex left in the execution of the algorithm. Therefore, in order to shorten the broadcast time from u_i to all vertices in T. We will transmit the messages to κ in the beginning of transmission. Further, the optimal broadcast scheme for κ becomes $(u_1, u_2, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k)$. To prove $u_i \notin BC(T)$, it suffices to show that $b(u_i, T) > b(T)$.

First, we will show that if $w(u_i, \kappa) > 0$ and $b(u_i, T(u_i, \kappa)) \neq b(\kappa, T(\kappa, u_i))$, then $u_i \notin BC(T)$. Suppose that j is the smallest integer such that $t(u_j) + w(u_j, \kappa) + \alpha j = b(T)$. We first consider the subcase when $i \neq j$. The vertices in $T(u_j, \kappa)$ finish the receipt of the message from u_i no less than $w(u_i, \kappa) + t(u_j) + w(u_j, \kappa) + \alpha j$. Since $w(u_i, \kappa) > 0$, we have $w(u_i, \kappa) + t(u_j) + w(u_j, \kappa) + \alpha j > b(T)$. Next, we consider subcase when $i = j \neq 1$. Similarly, the vertices in $T(u_{j-1}, \kappa)$ finish the receipt of the message from u_i no less than $\alpha + w(u_j, \kappa) + t(u_{j-1}) + w(u_{j-1}, \kappa) + (j-1)\alpha$. Since $w(u_j, \kappa) > 0$ and $t(u_{j-1}) + w(u_{j-1}, \kappa) \geq t(u_j) + w(u_j, \kappa)$, we have $\alpha + w(u_j, \kappa) + t(u_{j-1}) + w(u_{j-1}, \kappa) + (j-1)\alpha > b(T)$. It follows that $b(u_i, T) > b(T)$. Finally, we consider the subcase when $i = j = 1$. Since κ is the last remaining vertex and $b(u_i, T(u_i, \kappa)) \neq b(\kappa, T(\kappa, u_i))$, we have $b(\kappa, T(\kappa, u_i)) > b(u_i, T(u_i, \kappa))$. It follows that

$$\begin{aligned} b(u_i, T) &= \max\{b(\kappa, T(\kappa, u_i)) + w(\kappa, u_i) + \alpha, b(u_i, T(u_i, \kappa)) + \alpha\} \\ &> \max\{b(u_i, T(u_i, \kappa)) + w(\kappa, u_i) + \alpha, b(\kappa, T(\kappa, u_i)) + \alpha\} \\ &= b(\kappa, T). \end{aligned}$$

Thus, we have $u_i \notin BC(T)$.

Next, we will prove that if $i > h$ and $b(u_i, T(u_i, \kappa)) \neq b(\kappa, T(\kappa, u_i))$, then $u_i \notin BC(T)$. Again, we consider the vertices in $T(u_h, \kappa)$. Since $i > h$, the vertices in $T(u_h, \kappa)$ finish the receipt of the message from u_i at $w(u_i, \kappa) + t(u_h) + w(u_h, \kappa) + \alpha h + \alpha$ time. By the definition of h, we obtain $b(u_i, T) > b(T)$. □

Since the only candidates for being a broadcast center of T are the neighbors of κ, the set of broadcast centers $BC(T)$ is a star. Then we have the following corollary.

Corollary 9. *The set of broadcast centers $BC(T)$ is a star.*

Below we provide the correctness proof and the analysis of running time of the algorithm.

Theorem 10. *Given a weighted tree $T(V, E)$, the set of broadcast centers $BC(T)$ and the broadcast time $b(T)$ can be obtained in $O(n)$ time.*

Proof. Combining Lemmas 5 to 8, we obtain the correctness proof of the algorithm. In the following, we will show that the algorithm can be implemented in $O(n)$ time. We first observe that Steps 1 and 2 take $O(n)$ time. Suppose that $u_1, u_2, \ldots, u_k$ are the neighbors of v in W such that $t(u_1) + w(u_1, v) \geq t(u_2) + w(u_2, v) \geq \ldots \geq t(u_k) + w(u_k, v)$. Then, the label of v is the maximum value of $\{t(u_i) + w(u_i, v) + \alpha i\}$, i.e., $t(v) = \max\{t(u_i) + w(u_i, v) + \alpha i \mid 1 \leq i \leq k\}$. To determine the label of v, there is an intuitive implementation which takes $O(k \log k)$ time to sort the values $t(u_1) + w(u_1, v), t(u_2) + w(u_2, v), \ldots, t(u_k) + w(u_k, v)$. It follows that the overall time complexity of the while loop is $O(n \log n)$. However, if Step 7 takes $O(k)$ time to assign the label $t(v)$ for each vertex v in T, then the while loop can be implemented in $O(n)$ time. Below, we show how to determine $t(v)$ in $O(k)$ time by providing a non-sorting labelling method.

If a vertex u_p satisfies $t(u_1) + w(u_1, v) \geq t(u_p) + w(u_p, v) + \alpha k$ with $k \geq p$, then $t(u_1) + w(u_1, v) + \alpha > t(u_p) + w(u_p, v) + \alpha p$. Note that the vertex u_p would have no influence on the label of v. Therefore, in order to determine $t(v)$, we only need to consider the neighbor u_i of v such that $t(u_i) + w(u_i, v) > t(u_1) + w(u_1, v) - \alpha k$. Based on the above observation, we construct the following k linked lists. For each list[i], with $0 \leq i \leq k-1$, the list[i] contains the vertices u_j such that $\alpha i \leq (t(u_1) + w(u_1, v)) - (t(u_j) + w(u_j, v)) < \alpha(i+1)$. Note that the vertex u_1 with $t(u_1) + w(u_1, v) = \max\{t(u_i) + w(u_i, v)\}$ can be determined by simply visiting the neighbors of v once. For each neighbor u_i of v in $N(v) - \{u_1\}$, if $t(u_i) + w(u_i, v) \leq t(u_1) + w(u_1, v) - \alpha k$, then we discard it. Otherwise, we insert the vertex into the front of its corresponding linked list. Therefore, it takes $O(k)$ time to construct these k linked lists.

We use num[i] to denote the number of vertices in the list[i] and let acc[i] $= \sum_{j=0}^{i}$ num[j] for $0 \le i \le k-1$. Further, let u_{i^*} be the vertex in the list[i] such that $t(u_{i^*}) + w(u_{i^*}, v) = \min\{t(u_j) + w(u_j, v) \mid u_j$ belongs to the list[i]$\}$. Clearly, for $0 \le i \le k-1$, the values num[i], acc[i], and the vertex u_{i^*} can be determined in $O(k)$ time. For any given vertices u_x and u_y belonging to the same linked list with $x < y$, since $y - x \ge 1$ and $(t(u_x) + w(u_x, v)) - (t(u_y) + w(u_y, v)) < \alpha$, we have $t(u_y) + w(u_y, v) + \alpha y > t(u_x) + w(u_x, v) + \alpha x$. Therefore,

$$\begin{aligned} t(v) &= \max\{t(u_i) + w(u_i, v) + \alpha i \mid 1 \le i \le k\} \\ &= \max\{t(u_{i^*}) + w(u_{i^*}, v) + \alpha \mathrm{acc}[i] \mid 0 \le i \le k-1\}. \end{aligned}$$

So the label of v, $t(v)$, can be determined in $O(k)$ time. By the above labeling method, the while loop can be completed in $O(n)$ time for n vertices.

Next, we show that the smallest integer h such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$ can also be determined in $O(n)$ time in Step 12. Let q be the smallest integer such that $t(u_{q^*}) + w(u_{q^*}, \kappa) + \alpha \mathrm{acc}[q] + \alpha > b(T)$. Clearly, the list[$q$] contains the vertex u_h. We will show that either $h = \mathrm{acc}[q]$ or $h = \mathrm{acc}[q] - 1$. For any given vertices u_x and u_y belonging to the same linked list with $y \ge x + 2$, since $(t(u_x) + w(u_x, v)) - (t(u_y) + w(u_y, v)) < \alpha$, we have $(t(u_y) + w(u_y, \kappa) + y\alpha) - (t(u_x) + w(u_x, \kappa) + x\alpha) > \alpha$. Note that $t(u_{q^*}) + w(u_{q^*}, \kappa) + \alpha \mathrm{acc}[q] \le b(T)$. Therefore, for any vertex u_i belonging to list[q] with $i \le \mathrm{acc}[q] - 2$, we have $t(u_i) + w(u_i, \kappa) + \alpha i < b(T) - \alpha$. It follows that $h = \mathrm{acc}[q]$ or $h = \mathrm{acc}[q] - 1$. Hence, h can be determined in $O(n)$ time. These prove the correctness and time complexity of the algorithm. □

In the following, we will show that an optimal broadcast scheme by which the broadcast center broadcasts its messages to all vertices in T can be determined in $O(n)$ time.

Theorem 11. *An optimal broadcast scheme by which the broadcast center broadcasts its messages to all vertices in T can be determined in $O(n)$ time.*

Proof. Suppose that $u_1, u_2, \ldots, u_k$ are the neighbors of v in T such that $t(u_1) + w(u_1, v) \ge t(u_2) + w(u_2, v) \ge \ldots \ge t(u_k) + w(u_k, v)$. We first construct the following k linked lists, for each list[i], with $0 \le i \le k-1$, the list[i] contains the vertices u_j such that $\alpha i \le (t(u_1) + w(u_1, v)) - (t(u_j) + w(u_j, v)) < \alpha(i+1)$. Suppose further that the list[k] contains the vertices u_j such that $t(u_1) + w(u_1, v) \ge t(u_j) + w(u_j, v) + \alpha k$. Let num[$i$] denote the number of vertices in the list[i] and acc[i] $= \sum_{j=0}^{i}$ num[j] for $0 \le i \le k-1$. Further, let u_{i^*} be the vertex in the list[i] such that $t(u_{i^*}) + w(u_{i^*}, v) = \min\{t(u_j) + w(u_j, v) \mid u_j$ belongs to the list[i]$\}$. We place the vertex u_{i^*} at the end of the list[i] for $0 \le i \le k$.

Then, we assume that $u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(k)}$ is a traversal ordering of the lists such that the list[p] is traversed before the list[q] if $p < q$. When traversing a list, we traverse the list from the beginning to the end of the list sequentially. Clearly, the ordering $u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(k)}$ can be determined in $O(k)$ time. Since

$$\begin{aligned} t(v) &= \max\{t(u_i) + w(u_i, v) + \alpha i \mid 1 \le i \le k\} \\ &= \max\{t(u_{i^*}) + w(u_{i^*}, v) + \alpha \mathrm{acc}[i] \mid 0 \le i \le k-1\} \\ &= \max\{t(u_{\pi(i)}) + w(u_{\pi(i)}, v) + \alpha \pi(i) \mid 1 \le i \le k\}, \end{aligned}$$

$u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(k)}$ is an optimal broadcast scheme by which v broadcasts messages to its neighboring vertices in T, which completes the proof. □

Suppose T is a rooted tree with v as the root. Then, using a similar method of the above arguments in Theorems 10 and 11, one can show that the broadcast time $b(v, T)$ and an optimal broadcast scheme from v to all vertices of T can be determined in a bottom-up manner in $O(n)$ time. Thus, we have the following theorem.

Theorem 12. *Given a vertex $v \in V(T)$, the broadcast time $b(v, T)$ and an optimal broadcast scheme from v to all vertices in T can be determined in $O(n)$ time.*

4. Conclusion and future work

We have proposed a non-sorting linear-time algorithm for the broadcasting problem in a weighted tree under the postal model. The algorithm BROADCAST computes the set of broadcast centers, determines the broadcast time $b(T)$, and an optimal broadcast scheme from the broadcast center to all vertices in T. Also, given a vertex $v \in V(T)$, the broadcast time $b(v, T)$ and an optimal broadcast scheme from v to all vertices in T can be determined in linear time.

Below we present some open problems related to the broadcasting problem in weighted trees under the postal model, i.e., the broadcasting p-center problems, the broadcasting p-median problems, and the broadcasting problems with bounded communication time.

The broadcasting p-center problem is a generalization of the broadcasting problem. In the broadcasting p-center problem, we want to locate p centers on a network and partition the set of n vertices in p subsets, such that the maximum broadcast time from the centers to the associated subsets of vertices is minimized. On the other hand, in the broadcasting p-median problem, instead of minimizing the maximum broadcast time, we minimize the sum of communication times from the centers to the associated subsets of vertices. Tsou et al. [30] showed that the broadcasting p-median problem is NP-hard in general graphs. Furthermore, they also provided a linear-time algorithm for the broadcasting p-median problem in weighted trees when $p = 1$. Together with the results of this paper, it would be interesting to further consider both problems with $p > 1$, i.e., design polynomial-time algorithms for the broadcasting p-center problem and the broadcasting p-median problem when $p > 1$.

The broadcasting problem with bounded communication time is to find a vertex $v \in V(T)$ such that the sum of communication time of v is minimized under the constraint that the maximum communication time is no greater than a given constant value $c > 0$.

Acknowledgements

We thank A. Liestman and Elaheh Kamaliha for pointing out an error in the proof of Lemma 8 of an earlier version.

References

[1] B. Baker, R. Shostak, Gossips and telephones, Discrete Math. 2 (3) (1972) 191–193.
[2] A. Bar-Noy, S. Guha, J. Naor, B. Schieber, Message multicasting in heterogeneous networks, SIAM J. Comput. 30 (2) (2000) 347–358.
[3] A. Bar-Noy, S. Kipnis, Designing broadcasting algorithms in the postal model for message-passing systems, Math. Syst. Theory 27 (5) (1994) 431–452.
[4] R. Beier, J.F. Sibeyn, A powerful heuristic for telephone gossiping, in: Proceedings of the 7th International Colloquium on Structural Information and Communication Complexity, 2000, pp. 17–35.
[5] R.T. Bumby, A problem with telephones, SIAM J. Algebr. Discrete Methods 2 (1) (1981) 13–18.
[6] A. Dessmark, A. Lingas, H. Olsson, H. Yamamoto, Optimal broadcasting in almost trees and partial k-trees, in: Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science, 1998, pp. 432–443.
[7] M. Elkin, G. Kortsarz, Sublogarithmic approximation for telephone multicast, J. Comput. System Sci. 72 (4) (2006) 648–659.
[8] A.M. Farley, A. Proskurowski, Broadcasting in trees with multiple originators, SIAM J. Algebr. Discrete Methods 2 (4) (1981) 381–386.
[9] U. Feige, D. Peleg, P. Raghavan, E. Upfal, Randomized broadcast in networks, Random Structures Algorithms 1 (4) (1990) 447–460.
[10] P. Fraigniaud, E. Lazard, Methods and problems of communication in usual networks, Discrete Appl. Math. 53 (1–3) (1994) 79–133.
[11] P. Fraigniaud, S. Vial, Approximation algorithms for broadcasting and gossiping, J. Parallel Distrib. Comput. 43 (1) (1997) 47–55.
[12] H.A. Harutyunyan, C. Jimborean, New heuristic for message broadcasting in networks, in: Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications, 2014, pp. 517–524.
[13] H.A. Harutyunyan, A.L. Liestman, Improved upper and lower bounds for k-broadcasting, Networks 37 (2) (2001) 94–101.
[14] H.A. Harutyunyan, A.L. Liestman, k-Broadcasting in trees, Networks 38 (3) (2001) 163–168.
[15] H.A. Harutyunyan, A.L. Liestman, K. Makino, T.C. Shermer, Nonadaptive broadcasting in trees, Networks 57 (2) (2011) 157–168.
[16] H.A. Harutyunyan, A.L. Liestman, J.G. Peters, D. Richards, Broadcasting and gossiping, in: Handbook of Graph Theory, second edition, Chapman and Hall/CRC, 2013, pp. 1477–1494.
[17] H.A. Harutyunyan, A.L. Liestman, B. Shao, A linear algorithm for finding the k-broadcast center of a tree, Networks 53 (3) (2009) 287–292.
[18] H.A. Harutyunyan, B. Shao, An efficient heuristic for broadcasting in networks, J. Parallel Distrib. Comput. 66 (1) (2006) 68–76.
[19] S.M. Hedetniemi, S.T. Hedetniemi, A.L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (4) (1988) 319–349.
[20] J. Hromkovič, R. Klasing, B. Monien, R. Peine, Dissemination of information in interconnection networks (broadcasting & gossiping), in: Combinatorial Network Theory, Kluwer Academic Publishers, Dordrecht, 1995, pp. 125–212.
[21] J. Hromkovič, R. Klasing, A. Pelc, P. Ružička, W. Unger, Dissemination of Information in Communication Networks-Broadcasting, Gossiping, Leader Election, and Fault-Tolerance, Springer-Verlag, Berlin, 2005.
[22] A. Jakoby, R. Reischuk, C. Schindelhauer, The complexity of broadcasting in planar and decomposable graphs, Discrete Appl. Math. 83 (1–3) (1998) 179–206.
[23] J.-M. Koh, D.-W. Tcha, Information dissemination in trees with nonuniform edge transmission times, IEEE Trans. Comput. 40 (10) (1991) 1174–1177.
[24] J.-C. König, E. Lazard, Minimum k-broadcast graphs, Discrete Appl. Math. 53 (1–3) (1994) 199–209.
[25] G. Kortsarz, D. Peleg, Approximation algorithms for minimum-time broadcast, SIAM J. Discrete Math. 8 (3) (1995) 401–427.
[26] R. Labahn, The telephone problem for trees, Elektron. Informationsverarb. Kybernet. 22 (9) (1986) 475–485.
[27] E. Lazard, Broadcasting in DMA-bound bounded degree graphs, Discrete Appl. Math. 37 (38) (1992) 387–400.
[28] P. Scheuermann, G. Wu, Heuristic algorithms for broadcasting in point-to-point computer networks, IEEE Trans. Comput. 33 (9) (1984) 804–811.
[29] P.J. Slater, E.J. Cockayne, S.T. Hedetniemi, Information dissemination in trees, SIAM J. Comput. 10 (4) (1981) 692–701.
[30] C.-H. Tsou, G.-H. Chen, H.-I. Yu, C.-C. Lin, The broadcast median problem in heterogeneous postal model, J. Comb. Optim. 25 (4) (2013) 602–616.