J Comb Optim (2013) 25:602–616
DOI 10.1007/s10878-012-9493-8

The broadcast median problem in heterogeneous postal model

Cheng-Hsiao Tsou · Gen-Huey Chen · Hung-I Yu · Ching-Chi Lin

Published online: 26 April 2012

Abstract We propose the problem of finding broadcast medians in heterogeneous networks. A heterogeneous network is represented by a graph $G = (V, E)$, in which each edge has a weight that denotes the communication time between its two end vertices. The overall delay of a vertex $v \in V(G)$, denoted as $b(v, G)$, is the minimum sum of the communication time required to send a message from v to all vertices in G. The broadcast median problem consists of finding the set of vertices $v \in V(G)$ with minimum overall delay $b(v, G)$ and determining the value of $b(v, G)$. In this paper, we consider the broadcast median problem following the heterogeneous postal model. Assuming that the underlying graph G is a general graph, we show that computing $b(v, G)$ for an arbitrary vertex $v \in V(G)$ is NP-hard. On the other hand, assuming that G is a tree, we propose a linear time algorithm for the broadcast median problem in heterogeneous postal model.

Keywords Algorithm · Broadcast median · Heterogeneous network · Postal model

C.-H. Tsou · G.-H. Chen (✉)
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
e-mail: ghchen@csie.ntu.edu.tw

C.-H. Tsou
e-mail: f97922063@csie.ntu.edu.tw

H.-I. Yu
Institute of Information Science, Academia Sinica, Taipei, Taiwan
e-mail: herbert@iis.sinica.edu.tw

C.-C. Lin
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan
e-mail: lincc@mail.ntou.edu.tw

1 Introduction

Broadcasting is one of the most important operations in message-passing systems, such as distributed and parallel systems, and communication networks. The main objective of this operation is to quickly distribute data from a source vertex to the entire network for processing. Applications that make use of broadcasting include scientific computations, database transactions, network management protocols, and so on. Due to its significance, designing efficient broadcasting algorithms has received much attention from researchers over decades (Bar-Noy et al. 2000; Hedetniemi et al. 1988; Khuller and Kim 2007; Khuller et al. 2006, 2008; Lee and Chang 1992; Richards and Liestman 1988; Slater et al. 1981).

In the literature, algorithms implementing the broadcasting operation are asked to minimize the *broadcasting time* from a given source vertex, which is the worst-case time required for sending one message from the source to all vertices in the network. Researchers extended the concept to study the *broadcast center* problem, which aimed to find the set of vertices with minimum broadcasting time. Due to low broadcasting time, this kind of broadcasting is appropriate for applications with frequent data exchange and synchronization. However, the message arrival time of each vertex would be loose since we put emphasis on only the maximum communication time. Therefore, for applications with large amount of independent distributed operations, such as database query or web search, the computation resource of the whole system may not be well utilized in a long-term view. In this paper, we consider a new kind of broadcasting that seeks to minimize the *overall delay* from a given source vertex, which is defined to be the sum of the communication time required to send a message from the source vertex to all vertices in G. Based on this kind of broadcasting, we define the *broadcast median* problem, which is to find the set of vertices with minimum overall delay.

The two most common models for studying message-passing systems are the *telephone model* (Slater et al. 1981) and the *postal model* (Bar-Noy and Kipnis 1994, 1997). The communication in both models is via *calls* between adjacent vertices, each of which transmits a message from one sender to one receiver. In the telephone model, each call takes one unit of time, and each vertex can only participate in one call at any time. The postal model is introduced as a more realistic model, which employs an additional parameter $\alpha > 0$, called the *setup time*. In this model, each call consists of a setup phase and a transmission phase, which take α and one unit of time respectively. Furthermore, a sender is allowed to start a new call to another receiver whenever the setup phase of its last call is finished. For example, suppose that a vertex u wants to transmit a message first to v_1 and then to v_2 at time 0. In its first call, u sets up the connection to v_1 in α time and sends the message to v_1 in one unit of time. So the call from u to v_1 will be completed after time $\alpha + 1$. Since the setup phase of the first call is finished at time α, the call from u to v_2 will be completed after time $\alpha + (\alpha + 1) = 2\alpha + 1$.

In this paper, assuming that the underlying communication network is a *heterogeneous network*, we study the broadcast median problem following a generalized postal model, which is called the *heterogeneous postal model* (Bar-Noy et al. 2000). A heterogeneous network is a network connecting vertices with different operating systems and communication protocols, in which the transmission between distinct pairs of vertices take different time. The heterogeneous network is represented by an edge-weighted graph $G = (V, E)$, in which $V(G)$ is the set of vertices and $E(G)$ is the set of edges connecting vertices. Let $n = |V(G)|$ and $m = |E(G)|$. Each edge $(u, v) \in E(G)$ is associated with a nonnegative weight $w(u, v) \geq 0$, which represents the time required to send a message from u to v, or from v to u. Thus, a call from u to v now takes $\alpha + w(u, v)$ time. Given an edge-weighted graph $G = (V, E)$, the *overall delay* of v, denoted as $b(v, G)$, is the minimum sum of the communication time required to send a message from v to all vertices in G. The overall delay of G, denoted by $b(G)$, is the minimum overall delay among all vertices $v \in V(G)$, i.e., $b(G) = \min\{b(v, G) \mid v \in V(G)\}$. A vertex with the minimum overall delay is a broadcast median of G, and the set of broadcast medians of G is denoted by $BM(G) = \{v \mid v \in V(G), b(v, G) = b(G)\}$.

Problem Definition Given an edge-weighted graph $G = (V, E)$, the broadcast median problem in heterogeneous postal model is to determine the set of broadcast medians $BM(G)$ and compute the overall delay $b(G)$ of G, following the heterogeneous postal model with setup time $\alpha > 0$.

1.1 Related work and contributions

To our best knowledge, the broadcast median problem has not been considered in the literature. In the following, we list some results of the broadcast center problem for comparison.

For the telephone model, Slater et al. (1981) first showed that computing the broadcasting time of a given vertex in an arbitrary unweighted graph is NP-complete. They also provided an $O(n)$-time algorithm to compute the set of broadcast centers and determine the broadcasting time for an unweighted tree. By adopting Slater et al.'s algorithm, Koh and Tcha (1991) extended the result to provide an $O(n \log n)$-time algorithm for a weighted tree. Farley (1980) determined the lower and upper bounds of the time required to broadcast m messages from a given vertex v to all vertices in an unweighted graph.

For the postal model, Bar-Noy and Kipnis (1994) presented an algorithm to determine the broadcasting time from a given vertex to all vertices in an unweighted complete graph. They showed that the algorithm is optimal which runs in $\Theta(\lambda \log n / \log(\lambda + 1))$ time, where $\lambda > 1$ denotes the communication latency, i.e., the time needed from the sender starts sending messages until the receiver completes the transmission. For a weighted tree, Su et al. (2010) provided an $O(n)$-time algorithm to compute the set of broadcast centers and determine its broadcasting time.

In this paper, we show that computing the overall delay $b(v, G)$ of a given vertex $v \in V(G)$ in an arbitrary graph is NP-hard, following the heterogeneous postal model. Then, assuming that the underlying graph is a weighted tree T, we propose a linear time algorithm for the broadcast median problem in the heterogeneous postal model.

1.2 Organization

The rest of the paper is organized as follows. In Sect. 2, we propose the hardness result for computing the overall delay of a given vertex in a general graph. Then, Sect. 3

describes the algorithm that computes the set of broadcast medians of a weighted tree and determines the overall delay of the tree. Finally, we give some concluding remarks and suggest directions for future study in Sect. 4.

2 NP-hardness for general graphs

In this section, we show that it is NP-hard to compute the overall delay of a given vertex in a general graph G, following the heterogeneous postal model. To be more specific, we consider only a special case of this problem, in which the broadcasting is always performed in a *zero graph*, where the weight $w(u,v)=0$ for each edge $(u,v)\in E(G)$ and the setup time $\alpha=1$. We prove that the problem of computing the overall delay of a given vertex under such case, which is called the *overall delay in zero graphs* (OD) problem, is NP-hard. For better understanding, the proof is done in two steps. First, we customize a problem, called the *set overall delay in zero graphs* (SOD) problem, and show its NP-hardness by a reduction from the *three-dimensional matching* (3DM) problem. Then, by extending the reduction concept, the 3DM problem is shown to be reducible to the OD problem, and thus the OD problem is NP-hard. We begin with the problem definitions.

Definition 1 (The Overall Delay in Zero Graphs (OD) Problem) Given a zero graph $G=(V,E)$ with a vertex $u\in V(G)$ and a positive integer d, determine whether $b(u,G)\le d$ or not.

Given a zero graph $G=(V,E)$, the *overall delay* of a subset $V_0\subseteq V(G)$, denoted as $b(V_0,G)$, is the minimum sum of the communication time required to send a message simultaneously from the vertices in V_0 to all vertices in G.

Definition 2 (The Set Overall Delay in Zero Graphs (SOD) Problem) Given a zero graph $G=(V,E)$ with a set $V_0\subseteq V(G)$ and a positive integer d, determine whether $b(V_0,G)\le d$ or not.

Definition 3 (The Three-Dimensional Matching (3DM) Problem) Let $X=\{x_1, x_2,\dots,x_m\}$, $Y=\{y_1,y_2,\dots,y_m\}$, and $Z=\{z_1,z_2,\dots,z_m\}$ be three finite sets. Given a set of ordered triples $M\subseteq X\times Y\times Z$, determine whether there exists a matching of size m, i.e., a subset of M of size m in which any two distinct triples disagree with each other in all three coordinates.

It is well-known that the 3DM problem is NP-complete (Karp 1972). Now, we show that any 3DM instance is polynomially transformable to an instance of the SOD problem. A zero graph G and a vertex set V_0 are constructed from the 3DM instance, such that $b(V_0,G)$ is equal to a specific value if and only if the 3DM instance has a matching of size m, thereby giving the NP-hardness result of the SOD problem. We first define some basic components for constructing G. A *gadget* H_i and its *root* $r(H_i)$ are recursively defined as below. Refer to Fig. 1 for an illustrative example.

- H_0 consists of a single vertex, which is its root $r(H_0)$,

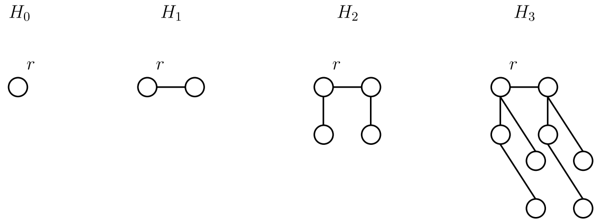

Fig. 1 The gadgets H_0, H_1, H_2, and H_3

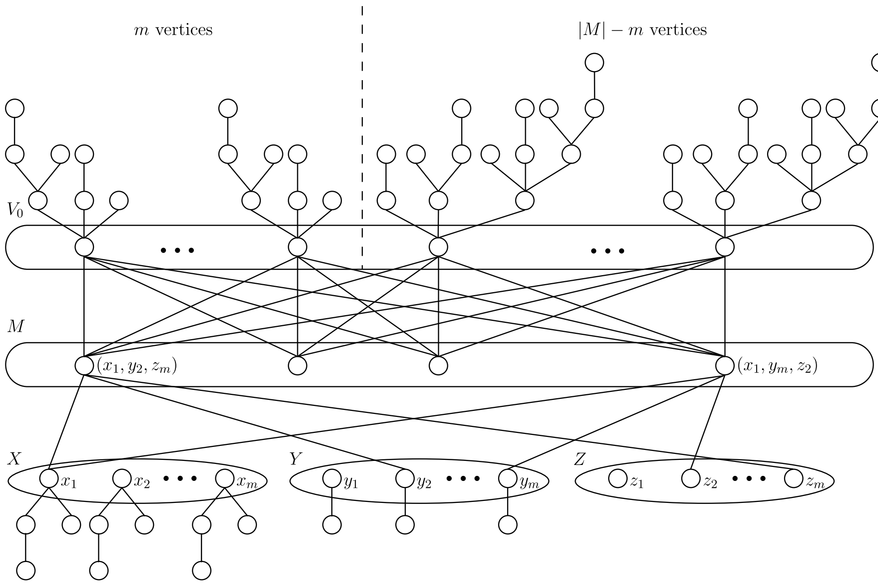

Fig. 2 The graph G corresponding to the problem 3DM

- H_i is obtained by connecting the roots of two gadgets H_{i-1}, and $r(H_i)$ is one of the roots in H_{i-1}.

Given a 3DM instance $I_3 = (X, Y, Z, M, m)$, we construct an SOD instance $I_S = (G, V_0, d)$ as follows, where $d = 49|M|$. Refer to Fig. 2 for an illustrative example.

- For each $x_i \in X$, create a gadget H_2, where $r(H_2)$ represents x_i in G.
- For each $y_i \in Y$, create a gadget H_1, where $r(H_1)$ represents y_i in G.
- For each $z_i \in Z$, create a gadget H_0, where $r(H_0)$ represents z_i in G.
- Create $|M|$ vertices to represent M in G, in which each vertex representing $(x_i, y_j, z_k) \in M$ connects to x_i, y_j, and z_k.
- Create $|M|$ vertices as V_0.
- Connect V_0 and M so that the bipartite graph induced by them is complete.
- Divide V_0 into two disjoint sets V_a and V_b of size m and $|M| - m$, respectively.
- For each $v \in V_a$, create 3 gadgets H_0, H_1, H_2 and connect their roots to v.
- For each $v \in V_b$, create 3 gadgets H_1, H_2, H_3 and connect their roots to v.

There are $(2^2+2^1+2^0)m+2|M|+(2^2+2^1+2^0)m+(2^3+2^2+2^1)(|M|-m) = 16|M|$ vertices in G, so the transformation is done in time polynomial to $|M|$.

Consider the broadcasting in G starting from V_0. Since $\alpha = 1$ and all edge weights are 0 in G, the calls in the broadcasting can be made in a synchronized way, such that a broadcasting sequence from V_0 to all vertices in G can be described by $Q = (V_0, E_1, V_1, E_2, V_2, \dots, E_h, V_h)$, where h is the length of Q. The sequence Q is said to be *feasible* if Q satisfies that, for $i \geq 1$, each edge in E_i is incident to exactly one distinct vertex in V_{i-1}, $V_i = V_{i-1} \cup \{v \mid (u, v) \in E_i \text{ and } u \in V_{i-1}\}$, and $V_h = V(G)$. In a feasible broadcasting sequence Q, we can see that vertices in the set $V_i - V_{i-1}$ receive the message at time i. Thus, the overall delay with respect to Q can be computed as $\sum_{1\leq i\leq h} i|V_i - V_{i-1}|$. A feasible broadcasting sequence is optimal if the overall delay with respect to Q is equal to $b(V_0, G)$.

Lemma 1 *Let G be a zero graph and k be a positive integer. Suppose that $V_0 \subseteq V(G)$ is a vertex set in G such that $|V(G)| = 2^k|V_0|$. Then, we have $b(V_0, G) = ((k-1)2^k + 1)|V_0|$ if and only if there is an optimal feasible broadcasting sequence of length k.*

Proof Sufficiency. Suppose that $Q = (V_0, E_1, V_1, E_2, V_2, \dots, E_k, V_k)$ is an optimal feasible broadcasting sequence of length k. Since $|V_i - V_{i-1}| \leq |V_{i-1}|$, we have $2|V_{i-1}| \geq |V_i|$ for $1 \leq i \leq k$. Combining with the fact that $|V_k| = |V(G)| = 2^k|V_0|$, one can see that $|V_i| = 2^i|V_0|$ for $1 \leq i \leq k$. It follows that $b(V_0, G) = \sum_{1\leq i\leq k} i|V_i - V_{i-1}| = \sum_{1\leq i\leq k} i2^{i-1}|V_0| = ((k-1)2^k + 1)|V_0|$.

Necessity. Suppose that there is no optimal feasible broadcasting sequence of length k. Let $Q = (V_0, E_1, V_1, E_2, V_2, \dots, E_h, V_h)$ be an optimal feasible broadcasting sequence of length $h \neq k$. We prove the statement by showing that $b(V_0, G) > ((k-1)2^k + 1)|V_0|$. Note that $h \neq k$ implies that $h > k$ since $2|V_{i-1}| \geq |V_i|$ and $|V_h| = |V(G)| = 2^k|V_0|$.

Let $\delta_i = |V_i - V_{i-1}|$ for $1 \leq i \leq h$. Clearly, one can see that $\sum_{1\leq i\leq h} \delta_i = |V_h| - |V_0| = (2^k - 1)|V_0|$ and $b(V_0, G) = \sum_{1\leq i\leq h} i\delta_i$. Furthermore, we have $\delta_i \leq 2^{i-1}|V_0|$ as $|V_i - V_{i-1}| \leq |V_{i-1}|$. Let $\sigma_i = 2^{i-1}|V_0|$ for $1 \leq i \leq k$ and $\sigma_i = 0$ for $k < i \leq h$. Then, we have $\sum_{1\leq i\leq k} i\sigma_i = ((k-1)2^k + 1)|V_0|$ and $\sum_{1\leq i\leq h} \sigma_i = |V_0| \sum_{1\leq i\leq k} 2^{i-1} = (2^k - 1)|V_0| = \sum_{1\leq i\leq h} \delta_i$. It follows that $\sum_{1\leq i\leq h}(\delta_i - \sigma_i) = 0$ and $\sum_{k<i\leq h}(\delta_i - \sigma_i) = -\sum_{1\leq i\leq k}(\delta_i - \sigma_i)$. Hence, we have

$$\begin{aligned}
\sum_{1\leq i\leq h} i\delta_i - \big((k-1)2^k + 1\big)|V_0| &= \sum_{1\leq i\leq h} i\delta_i - \sum_{1\leq i\leq h} i\sigma_i \\
&= \sum_{1\leq i\leq k} i(\delta_i - \sigma_i) + \sum_{k<i\leq h} i(\delta_i - \sigma_i) \\
&\geq k \sum_{1\leq i\leq k} (\delta_i - \sigma_i) + (k+1) \sum_{k<i\leq h} (\delta_i - \sigma_i) \\
&> 0,
\end{aligned}$$

since $\sum_{k<i\leq h}(\delta_i - \sigma_i) > 0$. This implies that $b(V_0, G) > ((k-1)2^k + 1)|V_0|$. Thus, the lemma follows. □

By Lemma 1, we have $b(V_0, G) = 49|M|$ if and only if there is an optimal feasible broadcasting sequence of length 4. Next, we show that broadcasting from V_0 in G can be done within 4 time units, i.e., there exists a feasible broadcasting sequence of length 4, if and only if there exists a matching to the 3DM instance I_3. For each gadget H_i in G, the message broadcasted from V_0 is always transmitted to $r(H_i)$ first, and then it requires exact i unit of time to broadcast the message from $r(H_i)$ to all vertices in H_i. Therefore, to have a feasible broadcasting sequence of length 4, $r(H_i)$ must receive the message no later than time $4 - i$.

Based on the observation, broadcasting from $V_0 = V_a \cup V_b$ must be done according to the following order. For each vertex $v \in V_b$, v must first spend 3 units of time to broadcast upward for taking care of the three gadgets H_3, H_2, H_1 connecting to v. Then, v has to broadcast downward in the last step, which covers a vertex in M. To the contrary, all of the m vertices in V_a have to broadcast downward to a subset $U \subseteq M$ in the beginning, so that the gadgets created for X, Y, and Z may receive the message in time. The gadgets H_2, H_1, H_0 connecting to vertices in V_a are handled thereafter. Obviously, U and the $|M| - m$ vertices covered by V_b have to be disjoint and together they form M.

The above broadcasting sequence guarantees that, except the gadgets for X, Y, and Z, all vertices in G receive their own copies of the message within 4 time units. It follows that whether the broadcasting can be done within 4 time units completely depends on whether we can pick an appropriate subset U such that U is able to cover entire X at time 2, entire Y at time 3, and entire Z at time 4. Since $|U| = |X| = |Y| = |Z| = m$, it is easy to see that the existence of such U directly corresponds to the existence of a matching to the instance I_3. Summarizing the above arguments, we prove that there is a matching to I_3 if and only if $b(V_0, G) \le 49|M|$ in I_S, and the transformation is done in time polynomial to $|M|$.

Now, we show that transformation from the 3DM problem to the OD problem can be done in time polynomial to $|M|$, thereby prove the NP-hardness of the OD problem. Again, it is done by transforming the 3DM instance I_3 to an OD instance $I_O = (G^*, u, d^*)$, where $d^* = (k + 4)2^{k+5} + 1$ and k is the smallest positive integer with $2^k \ge |M|$. We construct the instance I_O as follows. Refer to Fig. 3 for an illustrative example.

- Create a gadget H_k and let $r(H_k)$ represent u.
- For each vertex v in H_k, create 4 gadgets H_0, H_1, H_2, H_3 and connect their roots to v.
- Create the SOD instance $I_S = (G, V_0, d)$ as discussed above.
- Arbitrarily choose $|M|$ vertices from H_k and connect them to V_0 in one-to-one correspondence.
- For each vertex v of the remaining $2^k - |M|$ vertices in H_k, create a gadget H_4 and connect its root to v.

Note that there are $2^k + (2^3 + 2^2 + 2^1 + 2^0)2^k + 16|M| + 2^4(2^k - |M|) = 32 \times 2^k = 2^{k+5}$ vertices in G^*. Since $2^{k-1} < |M|$ by definition, we have $2^{k+5} < 64|M|$. Thus, the transformation from I_3 to I_O takes time polynomial to $|M|$. Furthermore, by Lemma 1, the optimal feasible broadcast sequence from u in G^* is of length $k + 5$, if and only if $b(u, G^*) = (k + 4)2^{k+5} + 1$.

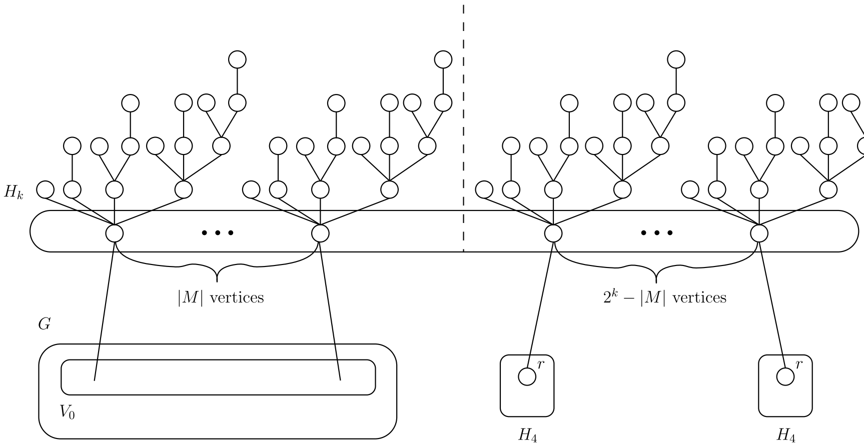

Fig. 3 The graph G^* with $u = r(H_k)$

We claim that the broadcasting from u can be done within $k + 5$ time units if and only if there is a matching of size m to the instance I_3. Note that broadcasting in each gadget H_4 from its root takes 4 time units, and broadcasting in G requires at least 4 time units even when the copies of the message are ready on V_0. Thus, one can observe that to complete the broadcasting within $k + 5$ time units, the roots of all H_4s and the vertices in V_0 have to receive the message no later than time $k + 1$. The only way to meet the above conditions is that u first broadcasts the message to all vertices in H_k within k time units and then all vertices of H_k broadcast downward simultaneously to V_0 and the roots of all H_4s.

After the conditions are satisfied, each vertex in H_k has to broadcast upward to handle its four attached gadgets, each root of H_4s has to broadcast inside its own gadget, and V_0 has to broadcast in G according to the best possible sequence as discussed previously. We have that only a broadcasting following this sequence has the possibility to be done within $k + 5$ time units. Furthermore, whether $k + 5$ time units is enough for broadcasting in G^* is entirely depended on whether broadcasting in G can be done within 4 time units. Combining with the above discussions, we have that $b(u, G^*) \leq (k + 4)2^{k+5} + 1$ if and only if there exists a matching to the 3DM instance I_3. Consequently, an 3DM instance can be polynomially reduced to an OD instance, which implies that the OD problem is NP-hard. Since the OD problem only handles a special case of the computation of the overall delay, we obtain the following theorem.

Theorem 1 *Given a weighted graph $G = (V, E)$ with a vertex $v \in V(G)$, it is NP-hard to compute the overall delay $b(v, G)$ of v, following the heterogeneous postal model with $\alpha > 0$.*

We remark that communicating in zero graphs also satisfies the definition of the telephone model. Thus, the above reduction also serves as an NP-hardness proof for the problem following the telephone model.

3 The algorithm for weighted tree graphs

In this section, we consider the broadcast median problem in heterogeneous postal model on a weighted tree $T=(V,E)$, and propose a linear-time algorithm for computing the set of broadcast medians $BM(T)$ and the overall delay $b(T)$. This algorithm always maintains a subtree of T that contains a broadcast median, while reducing the subtree size. The size reduction is done by removing vertices of T from outside to inside in a greedy manner. Initially, each leaf $\ell \in V(T)$ is assigned a label $num(\ell) \leftarrow 1$. Then, we randomly select two leaves of T and remove the one with smaller label from T, with ties broken arbitrarily. Whenever some vertex v becomes a new leaf due to the removal, a new label $num(v)$ is assigned to v, which is computed as the sum of 1 and the labels $num(u)$ among all neighbors u of v appearing in W, where W denotes the set of vertices removed so far. The process is iterated until only one vertex κ is remained. Note that the removal strategy guarantees that the leaf with largest label in the remaining tree is always kept, which ensures that the remaining subtree contains at least one broadcast median of T, as shown in Sect. 3.1. The property thus gives us that κ is a broadcast median. Furthermore, we have that any vertex v is a broadcast median of T if and only if (1) $|T(\kappa,v)|=|T(v,\kappa)|$, $v\in N(\kappa)$ or (2) $|T(v,\kappa)|=S$ and $w(v,\kappa)=0$, where $S=\max\{|T(v',\kappa)|\colon v'\in N(\kappa)\}$.

In order to compute the overall delay of T, we compute another label $t(v)$ for each vertex $v\in V(T)$. Initially, $t(\ell)\leftarrow 0$ for each leaf ℓ. Suppose that the removal process makes some vertex v become a new leaf. Let $u_1,u_2,\ldots,u_k$ be the neighbors of v appearing in W such that $num(u_i)\ge num(u_{i+1})$ for $1\le i\le k-1$. We then set the label $t(v)=\sum_{i=1}^{k} t(u_i)+num(u_i)\cdot(i\cdot\alpha+w(u_i,v))$, whose meaning can be seen as follows. Let v' be the neighbor of v on the path from v to κ. The edge (v,v') partitions T into two subtrees $T(v,v')$ and $T(v',v)$, which contain v and v' respectively. As will shown in Sect. 3.1, $t(v)=b(v,T(v,v'))$ for $v\ne\kappa$, which is the overall delay of v in the subtree $T(v,v')$, and $t(\kappa)=b(\kappa,T)=b(T)$. We also note that by definition $num(v)=1+\sum_{i=1}^{k} num(u_i)=|T(v,v')|$.

The algorithm is detailed below.

3.1 Correctness and complexity analysis

We begin with a lemma about how to establish an optimal sequence of calls while broadcasting. Let v be an arbitrary vertex in T and $u_1,\ldots,u_k$ are the neighbors of v in T such that $|T(u_1,v)|\ge|T(u_2,v)|\ge\cdots\ge|T(u_k,v)|$. In the following, we show that the best strategy to broadcast from v to all vertices in T is to send the message in the order: the larger, the earlier.

Lemma 2 *Let v be an arbitrary vertex in T and $u_1,\ldots,u_k$ be the neighbors of v in T such that $|T(u_1,v)|\ge|T(u_2,v)|\ge\cdots\ge|T(u_k,v)|$. Then, $u_1,u_2,\ldots,u_k$ is an optimal sequence of calls to broadcast the message from v to neighboring vertices in T. Consequently, $b(v,T)=\sum_{i=1}^{k} b(u_i,T(u_i,v))+|T(u_i,v)|\cdot(i\cdot\alpha+w(u_i,v))$.*

Proof Let π be a permutation of $\{1,2,\ldots,k\}$, where $\pi(i)=i$ for $1\le i\le n$, that represents a sequence of calls, in which u_i receives message from v at time $\alpha\cdot\pi(i)$,

Algorithm 1 Finding Overall Delay and Broadcast Medians

Input: A weighted tree $T = (V, E)$ with weight $w(u, v) \geq 0$ for each $(u, v) \in E(T)$.
Output: The overall delay $b(T)$ and the set of broadcast medians $BM(T)$.

1: **for** each leaf $\ell \in T$ **do** $num(\ell) \leftarrow 1$, $t(\ell) \leftarrow 0$;
2: let $BM(T) \leftarrow \emptyset$, $W \leftarrow \emptyset$ and $T' \leftarrow T$;
3: **while** $|V(T')| \geq 2$ **do**
4: choose two leaves u_x and u_y in T' at random;
5: let u be the vertex in $\{u_x, u_y\}$ such that $num(u) = \min\{num(u_x), num(u_y)\}$;
let $W \leftarrow W \cup \{u\}$ and $T' \leftarrow T' - \{u\}$;
let v be the vertex adjacent to u in T';
6: **if** v is a leaf in T' **then**
7: let $u_1, u_2, \ldots, u_k$ be the neighbors of v in W such that
$num(u_1) \geq num(u_2) \geq \cdots \geq num(u_k)$;
8: let $t(v) \leftarrow \sum_{i=1}^{k} t(u_i) + num(u_i) \cdot (i \cdot \alpha + w(u_i, v))$;
9: let $num(v) \leftarrow 1 + \sum_{i=1}^{k} num(u_i)$;
10: **end if**
11: **end while**
12: let κ be the only vertex left in T', $b(T) \leftarrow t(\kappa)$ and $BM(T) \leftarrow \{\kappa\}$;
let $S = \max\{|T(v', \kappa)|\colon\ v' \in N(\kappa)\}$;
13: $BM(T) \leftarrow BM(T) \cup \{v \in N(\kappa)\colon$ (1) $|T(v, \kappa)| = |T(\kappa, v)|$; or
(2) $|T(v, \kappa)| = S$ and $w(v, \kappa) = 0\}$.

respectively. We show that π is an optimal sequence of calls. Suppose that another permutation π' is an optimal sequence of calls, such that the overall delay $b(v, T)$ of v can be computed as $\sum_{i=1}^{k} b(u_i, T(u_i, v)) + |T(u_i, v)| \cdot (\pi'(i) \cdot \alpha + w(u_i, v))$.

If $\pi' = \pi$, then we are done. Otherwise, let $p(\pi')$ denote the smallest index such that $\pi'(i) \neq i$, which implies $\pi'(i) > i$. Without loss of generality, we suppose that the permutation π' is chosen from all optimal sequences so that $p(\pi')$ is the maximum. Suppose further that we have $\pi'(h) = i$, which implies that $h > i$. By exchanging the elements of indices i and h in π', we get a new permutation π''. Since $\pi'(i) > i$ and $h > i$, one can verify that $\sum_{i=1}^{k} b(u_i, T(u_i, v)) + |T(u_i, v)| \cdot (\pi'(i) \cdot \alpha + w(u_i, v)) \geq \sum_{i=1}^{k} b(u_i, T(u_i, v)) + |T(u_i, v)| \cdot (\pi''(i) \cdot \alpha + w(u_i, v))$. The permutation π'' thus represents another optimal sequence of calls. Since $\pi''(i) = i$, we can see that $p(\pi'') > i$, which contradicts the maximality of $p(\pi')$. This implies that π is an optimal sequence of calls from v to its neighboring vertices $u_1, u_2, \ldots, u_k$ in T, and the equation of $b(v, T)$ follows. □

The following lemma provides the theoretical basis for finding a broadcast median, which is essential for proving the correctness of our algorithm.

Lemma 3 *For any edge $(u, v) \in E(T)$, if $|T(u, v)| \leq |T(v, u)|$, then the following two statements hold*:

1. $b(u, T) = b(v, T(v, u)) + b(u, T(u, v)) + |T(v, u)| \cdot w(v, u) + (n - 1) \cdot \alpha$;
2. $b(v, T) \leq b(u, T)$.

Proof We first show the correctness of statement one. Suppose that $u_1, u_2, \dots, u_k$ are the neighbors of vertex u in T such that $|T(u_i, u)| \geq |T(u_{i+1}, u)|$ for $1 \leq i \leq k-1$. Since $|T(v, u)| \geq |T(u, v)|$, we have $u_1 = v$. Then, according to Lemma 2, we have

$$\begin{aligned} b(u, T) &= b\big(v, T(v, u)\big) + \big|T(v, u)\big| \cdot \big(\alpha + w(v, u)\big) \\ &\quad + \sum_{i=2}^{k} \big(b\big(u_i, T(u_i, u)\big) + \big|T(u_i, u)\big| \cdot \big(i \cdot \alpha + w(u_i, u)\big)\big) \\ &= b\big(v, T(v, u)\big) + \big|T(v, u)\big| \cdot \big(\alpha + w(v, u)\big) \\ &\quad + b\big(u, T(u, v)\big) + \big(\big|T(u, v)\big| - 1\big) \cdot \alpha \\ &= b\big(v, T(v, u)\big) + b\big(u, T(u, v)\big) + \big|T(v, u)\big| \cdot w(v, u) \\ &\quad + (n-1) \cdot \alpha. \end{aligned}$$

Next, we show the correctness of statement two. Consider a sequence of calls from v to all vertices in T that v passes the message to u in the beginning of broadcasting and the broadcasting sequences for u in $T(u, v)$ and v in $T(v, u)$ are optimal. Clearly, the overall delay of v with respect to the above broadcasting sequence would be greater than or equal to $b(v, T)$. Hence, we have

$$\begin{aligned} b(v, T) &\leq b\big(u, T(u, v)\big) + \big|T(u, v)\big| \cdot \big(\alpha + w(v, u)\big) \\ &\quad + b\big(v, T(v, u)\big) + \big(\big|T(v, u)\big| - 1\big) \cdot \alpha \\ &= b\big(u, T(u, v)\big) + b\big(v, T(v, u)\big) + \big|T(u, v)\big| \cdot w(v, u) \\ &\quad + (n-1) \cdot \alpha. \end{aligned}$$

Since $|T(u, v)| \leq |T(v, u)|$ and $w(v, u) \geq 0$, we have that $b(v, T) \leq b(u, T)$. □

Now we are ready to show the correctness of Algorithm 1. The following lemma proves the claim that T' always contains a broadcast median of T, using the fact that the leaf with largest label is always kept.

Lemma 4 *Suppose that a leaf u is deleted in the current tree T' in the ith iteration of the while loop. Then, $BM(T) \cap V(T'') \neq \emptyset$, where $T'' = T' - \{u\}$. Consequently, the last remaining vertex $\kappa \in BM(T)$.*

Proof Let v be the vertex adjacent to u in the current tree T'. To prove $BM(T) \cap V(T'') \neq \emptyset$, it suffices to show that $b(v, T) \leq b(u, T)$, i.e., v is more qualified to be a broadcast median than the removed u. We first consider the case when the current tree T' contains exactly two vertices u and v. Note that by the choice of u, we have $num(u) \leq num(v)$. This implies that $|T(u, v)| \leq |T(v, u)|$. Then, according to Lemma 3, one can see that $b(v, T) \leq b(u, T)$.

Next, we consider the case when the current tree T' contains at least three vertices, including u, v. Let y be the leaf in T' with largest label $num(y)$ before the ith iteration. Since u is removed after this iteration, we have that $u \neq y$ and $num(u) \leq num(y)$ by definition. 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, according to Lemma 3, we prove that by showing $|T(u, v)| \leq |T(v, u)|$. By $num(u) \leq num(y)$, we have $|T(u,v)| \le |T(y,y')|$. Furthermore, one can see that $T(y,y')$ is a subtree of $T(v,u)$. It follows that $|T(u,v)| \le |T(y,y')| \le |T(v,u)|$, which completes the proof. □

The above lemma shows that κ is a broadcast median, which means that $b(\kappa,T) = b(T)$. Next, we show that $b(\kappa,T)$ can be directly obtained from the label $t(\kappa)$.

Lemma 5 *For each vertex v in $T-\{\kappa\}$ in Algorithm 1, we have $t(v) = b(v, T(v,v'))$, where v' is the neighbor of v on the path from v to κ. Furthermore, $t(\kappa) = b(\kappa,T)$ after the last iteration of the while-loop.*

Proof Let $v_1, v_2, \ldots, v_{n-1}$ be the vertices in $T-\{\kappa\}$ arranged in the order of removal, such that v_i is removed from T' before v_j if $i<j$. We prove the statement by induction on the removal order. Let vertex v_1' be the neighbor of v_1 on the path from v_1 to κ. 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 $u_1, \ldots, u_\ell$ are the neighbors of v_{k+1} in $T(v_{k+1}, v'_{k+1})$ such that $|T(u_j, v_{k+1})| \ge |T(u_{j+1}, v_{k+1})|$ for $j = 1, \ldots, \ell-1$. Therefore, by Lemma 2, we have $b(v_{k+1}, T(v_{k+1}, v'_{k+1})) = \sum_{j=1}^{\ell} b(u_j, T(u_j, v_{k+1})) + |T(u_j, v_{k+1})| \cdot (j\cdot\alpha + w(u_j, v_{k+1}))$. According to Step 8 of Algorithm 1, we have $t(v_{k+1}) = \sum_{j=1}^{\ell} t(u_j) + num(u_j)\cdot(j\cdot\alpha + w(u_j, v_{k+1}))$. Since $num(u_j) = |T(u_j, v_{k+1})|$ and $u_j \in \{v_1, \ldots, v_k\}$ for $1 \le j \le \ell$, by induction hypothesis, we have $t(u_j) = b(u_j, T(u_j, v_{k+1}))$ for $1 \le j \le \ell$, and hence $t(v_{k+1}) = b(v_{k+1}, T(v_{k+1}, v'_{k+1}))$. □

Note that a tree may contain more than one broadcast median. Below we show that the only candidates besides κ for being broadcast medians of T are the neighbors of κ. More specifically, we will prove that for each vertex $v \in N(\kappa)$, v is a broadcast median of T if and only if (1) $|T(v,\kappa)| = |T(\kappa,v)|$ or (2) $|T(v,\kappa)| = S$ and $w(v,\kappa) = 0$, where $S = \max\{|T(v,\kappa)|\colon v \in N(\kappa)\}$.

Lemma 6 *If v is a broadcast median of T, then $v \in N(\kappa) \cup \{\kappa\}$. That is, the set of broadcast medians $BM(T)$ forms a star.*

Proof Let s be a vertex in T such that $s \notin N(\kappa) \cup \{\kappa\}$ and s' be the neighbor of s on the path from s to κ. By similar arguments as in Lemma 5, we have $|T(s,s')| \le |T(s',s)|$. Then, according to Lemma 3, we have $b(s',T) \le b(s,T)$. It follows that if s' is not a broadcast median of T, then s is not a broadcast median of T. Thus, we assume that s' is a broadcast median of T in the following discussion.

Suppose that $s' \in N(\kappa)$. Let $u_1, u_2, \ldots, u_k$ denote the neighbors of vertex s' in T such that $s = u_h$ and $|T(u_i, s')| \ge |T(u_{i+1}, s')|$ for $1 \le i \le k-1$. Notice that $|T(s,s')| < |T(s',\kappa)| \le |T(\kappa,s')|$, so we have $s \ne u_1$ and $|T(u_1,s')| > |T(s,s')|$. Hence, we have

$$\begin{aligned} b(s', T) &= b\big(s, T(s, s')\big) + \big|T(s, s')\big| \cdot \big(h \cdot \alpha + w(s', s)\big) \\ &\quad + b\big(s', T(s', s)\big) + \sum_{i=h+1}^{k} \big|T(u_i, s')\big| \cdot \alpha \\ &= b\big(s, T(s, s')\big) + b\big(s', T(s', s)\big) + \big|T(s, s')\big| \cdot w(s', s) \\ &\quad + \left(h \cdot \big|T(s, s')\big| + \sum_{i=h+1}^{k} \big|T(u_i, s')\big|\right) \cdot \alpha. \end{aligned}$$

Since $|T(u_1, s')| > |T(s, s')|$ and $|T(u_i, s')| \geq |T(s, s')|$ for $1 < i < h$, one can verify that $b(s, T) > b(s', T)$ (by the aid of Lemma 3). It follows that we have $s \notin BM(T)$.

Suppose that $s' \notin N(\kappa)$. We have that $|T(s', s'')| \leq |T(s'', s')|$, where s'' is the neighbor of s' on the path from s' to κ, and thus $b(s'', T) \leq b(s', T)$ by Lemma 3. Since s' is a broadcast median of T, so is s''. By applying the arguments recursively, there exists a broadcast median s^* of T, such that the neighbor of s^* on the path from s^* to κ is in $N(\kappa)$ and is also a broadcast median of T. According to the discussions of the previous case, we have $s^* \notin BM(T)$, which is a contradiction. □

Lemma 7 *For each vertex $v \in N(\kappa)$, v is a broadcast median of T if and only if* (1) $|T(v, \kappa)| = |T(\kappa, v)|$ *or* (2) $|T(v, \kappa)| = S$ *and* $w(v, \kappa) = 0$, *where* $S = \max\{|T(v', \kappa)|\colon v' \in N(\kappa)\}$.

Proof Necessity. We first prove that $|T(\kappa, v)| \neq |T(v, \kappa)|$ and $w(v, \kappa) \neq 0$ implies $v \notin BM(T)$. To prove this, it suffices to show that $b(v, T) > b(\kappa, T)$. Since κ is the only vertex left in the algorithm, we have $|T(\kappa, v)| \geq |T(v, \kappa)|$. It follows from Lemma 3 that we have $b(v, T) = b(\kappa, T(\kappa, v)) + b(v, T(v, \kappa)) + |T(\kappa, v)| \cdot w(\kappa, v) + (n-1) \cdot \alpha$. Consider a sequence of calls from κ to all vertices in T that κ passes the message to v in the beginning of broadcasting and the broadcasting sequences for κ in $T(\kappa, v)$ and v in $T(v, \kappa)$ are optimal. Then, we have

$$\begin{aligned} b(\kappa, T) &\leq b\big(v, T(v, \kappa)\big) + \big|T(v, \kappa)\big| \cdot \big(\alpha + w(\kappa, v)\big) \\ &\quad + b\big(\kappa, T(\kappa, v)\big) + \big(\big|T(\kappa, v)\big| - 1\big) \cdot \alpha \\ &= b\big(v, T(v, \kappa)\big) + b\big(\kappa, T(\kappa, v)\big) + \big|T(v, \kappa)\big| \cdot w(\kappa, v) \\ &\quad + (n-1) \cdot \alpha. \end{aligned}$$

Since $|T(\kappa, v)| > |T(v, \kappa)|$ and $w(\kappa, v) > 0$, one can verify that $b(v, T) > b(\kappa, T)$. Thus, we have $v \notin BM(T)$.

Next, we prove that if $|T(\kappa, v)| \neq |T(v, \kappa)|$ and $|T(v, \kappa)| \neq S$, then $v \notin BM(T)$. Similarly, by Lemma 3, we have $b(v, T) = b(\kappa, T(\kappa, v)) + b(v, T(v, \kappa)) + |T(\kappa, v)| \cdot w(\kappa, v) + (n-1) \cdot \alpha$. Let $u_1, u_2, \ldots, u_k$ denote the neighbors of vertex κ in T such that $v = u_h$ and $|T(u_i, \kappa)| \geq |T(u_{i+1}, \kappa)|$ for $1 \leq i \leq k-1$. Then, since $|T(v, \kappa)| \neq S$, we have $v \neq u_1$ and $|T(u_1, \kappa)| > |T(v, \kappa)|$. Hence, we have

$$\begin{aligned} b(\kappa, T) &= b\big(v, T(v, \kappa)\big) + \big|T(v, \kappa)\big| \cdot \big(h \cdot \alpha + w(\kappa, v)\big) \\ &\quad + b\big(\kappa, T(\kappa, v)\big) + \sum_{i=h+1}^{k} \big|T(u_i, \kappa)\big| \cdot \alpha \end{aligned}$$

$$
\begin{aligned}
&= b\big(v, T(v,\kappa)\big) + b\big(\kappa, T(\kappa,v)\big) + \big|T(v,\kappa)\big| \cdot w(\kappa,v) \\
&\quad + \left(h \cdot \big|T(v,\kappa)\big| + \sum_{i=h+1}^{k} \big|T(u_i,\kappa)\big|\right) \cdot \alpha.
\end{aligned}
$$

Since $|T(u_1,\kappa)| > |T(v,\kappa)|$ and $|T(u_i,\kappa)| \ge |T(v,\kappa)|$ for $1 < i < h$, one can also verify that $b(v,T) > b(\kappa,T)$. Thus, we have $v \notin BM(T)$.

Sufficiency. Finally, we prove that (1) $|T(\kappa,v)| = |T(v,\kappa)|$ or (2) $|T(v,\kappa)| = S$ and $w(v,\kappa) = 0$ implies v is a broadcast median of T. We prove this by showing $b(v,T) \le b(\kappa,T)$. Once again, since $|T(\kappa,v)| \ge |T(v,\kappa)|$, we have $b(v,T) = b(\kappa, T(\kappa,v)) + b(v, T(v,\kappa)) + |T(\kappa,v)| \cdot w(\kappa,v) + (n-1)\cdot\alpha$ by Lemma 3. Suppose $u_1, u_2, \ldots, u_k$ are the neighbors of vertex κ in T such that $|T(u_i,\kappa)| \ge |T(u_{i+1},\kappa)|$ for $1 \le i \le k-1$. In either Case (1) or (2), we have $u_1 = v$. Then, according to Lemma 2, one can see that

$$
\begin{aligned}
b(\kappa,T) &= b\big(v, T(v,\kappa)\big) + \big|T(v,\kappa)\big| \cdot \big(\alpha + w(\kappa,v)\big) \\
&\quad + b\big(\kappa, T(\kappa,v)\big) + \big(\big|T(\kappa,v)\big| - 1\big) \cdot \alpha \\
&= b\big(v, T(v,\kappa)\big) + b\big(\kappa, T(\kappa,v)\big) + \big|T(v,\kappa)\big| \cdot w(\kappa,v) \\
&\quad + (n-1) \cdot \alpha.
\end{aligned}
$$

Since either $|T(v,\kappa)| = |T(\kappa,v)|$ or $w(v,\kappa) = 0$, one can verify that $b(v,T) = b(\kappa,T)$ as desired. It follows that we have $v \in BM(T)$. □

Lemmas 4, 5, and 7 give us the correctness of Algorithm 1. In the following, we show that the algorithm can be implemented in $O(n)$ time. We first observe that Steps 1, 2, 12, and 13 take $O(n)$ time. The while-loop executes $n-1$ iterations, in which Steps 4–6 and 8–9 together take $O(n)$ time in total. The only problem is how to obtain the sorted sequence of neighbors of v according to their labels $num(\cdot)$ in Step 7. Here, an $O(n)$-time preprocessing is proposed that obtains for each vertex $v \in V(T)$ a sorted sequence of the values $|T(u_1,v)|, |T(u_2,v)|, \ldots, |T(u_k,v)|$. Thus, the sorted sequence required in Step 7 can be directly derived from the pre-computed sequence of v. The preprocessing is done as follows. Let T be rooted at an arbitrary vertex. By a depth-first traversal, for each v, the size $S(v)$ of the subtree rooted at v can be computed in $O(n)$ time. Suppose that u is the parent of v. It can be see that $|T(v,u)| = S(v)$ and $|T(u,v)| = n - S(v)$. Thus, for every v, the sizes of all subtrees connecting to v can be computed in total $O(n)$ time. Then, these values are collected together and sorted by a counting sort. The sorting takes $O(n)$ time, since there are only $O(n)$ values and each of them is an integer bounded by n. Finally, each of these values is reassigned back to its corresponding vertex in $O(n)$ time by scanning the sorted sequence in order. In such way, each vertex obtains its sorted sequence $|T(u_1,v)|, |T(u_2,v)|, \ldots, |T(u_k,v)|$ in total $O(n)$ time. This leads to the fact that the complexity of the algorithm is $O(n)$. Then we have the main result of this paper.

Theorem 2 *Given a weighted tree T, Algorithm 1 computes the set of broadcast medians $BM(T)$ and determines the overall delay $b(T)$ in $O(n)$ time.*

4 Conclusion and future work

We remark that it is easy to modify Algorithm 1 to compute $b(v, T)$ for any given vertex $v \in V(T)$ in $O(n)$ time by excluding v from the target of vertex removal. In fact, based on the properties developed in this paper, we can even design a two-phase dynamic programming algorithm that computes $b(v, T)$ for all vertices $v \in V(T)$ in total $O(n)$ time. Let T be rooted at an arbitrary vertex. The first phase uses a bottom-up computation to compute $b(v, T(u, v))$ for each vertex v and every child u of v in T. Then, the second phase uses a top-down computation to compute $b(v, T(p(v), v))$ for each v and its parent $p(v)$ in T. Thereafter, combining with the sorting preprocessing mentioned previously, we can compute $b(v, T)$ for each v individually. It is not difficult to see that the algorithm takes total $O(n)$ time, which serves as another linear time solution of the broadcast median problem. We choose to present Algorithm 1 in this paper instead since it is more interesting.

In the following, we propose some related open problems. As shown in Sect. 2, computing the overall delay of a vertex in a general graph is NP-hard. It would be of interest to design approximation algorithms for obtaining low overall delay. Furthermore, in certain applications, the broadcasting operation may be asked for both low broadcasting time and low overall delay simultaneously. So we can consider the bi-criteria broadcasting problems, such as minimizing the broadcasting time with bounded overall delay or minimizing the overall delay with bounded broadcasting time. Designing bi-criteria broadcasting algorithms would be a great challenge.

References

Bar-Noy A, Kipnis S (1994) Designing broadcasting algorithms in the postal model for message-passing systems. Math Syst Theory 27(5):431–452

Bar-Noy A, Kipnis S (1997) Multiple message broadcasting in the postal model. Networks 29(1):1–10

Bar-Noy A, Guha S, Naor J, Schieber B (2000) Message multicasting in heterogeneous networks. SIAM J Comput 30(2):347–358

Farley AM (1980) Broadcast time in communication networks. SIAM J Appl Math 39(2):385–390

Hedetniemi SM, Hedetniemi ST, Liestman AL (1988) A survey of gossiping and broadcasting in communication networks. Networks 18(4):319–349

Karp RM (1972) Reducibility among combinatorial problems. In: Complexity of computer computations, Proc sympos, IBM Thomas J Watson Res Center, Yorktown Heights, NY, 1972. Plenum, New York, pp 85–103

Khuller S, Kim YA (2007) Broadcasting in heterogeneous networks. Algorithmica 48(1):1–21

Khuller S, Kim YA, Wan YC (2006) On generalized gossiping and broadcasting. J Algorithms 59(2):81–106

Khuller S, Kim YA, Wan YC (2008) Broadcasting on networks of workstations. Algorithmica 52(3):1–21

Koh JM, Tcha DW (1991) Information dissemination in trees with nonuniform edge transmission times. IEEE Trans Comput 40(10):1174–1177

Lee HM, Chang GJ (1992) Set to set broadcasting in communication networks. Discrete Appl Math 40(3):411–421

Richards D, Liestman AL (1988) Generalizations of broadcasting and gossiping. Networks 18(2):125–138

Slater PJ, Cockayne EJ, Hedetniemi ST (1981) Information dissemination in trees. SIAM J Comput 10(4):692–701

Su YH, Lin CC, Lee DT (2010) Broadcasting in heterogeneous tree networks. In: Proceedings of the 16th annual international conference on computing and combinatorics, COCOON'10. Springer, Berlin, pp 368–377