Cost-Constrained Minimum-Delay Multicasting (Extended Abstract)

Satoshi Tayu, Turki Ghazi Al-Mutairi, and Shuichi Ueno

Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo 152-8552-S3-57, Japan {tayu, ueno}@lab.ss.titech.ac.jp

Abstract. We consider a problem of cost-constrained minimum-delay multicasting in a network, which is to find a Steiner tree spanning the source and destination nodes such that the maximum total delay along a path from the source node to a destination node is minimized, while the sum of link costs in the tree is bounded by a constant. The problem is \mathcal{NP} -hard even if the network is series-parallel. We present a fully polynomial time approximation scheme for the problem if the network is series-parallel.

1 Introduction

The multicasting is the simultaneous transmission of data from a source node to multiple destination nodes in a network. The multicasting involves the generation of a multicast tree, which is a Steiner tree spanning the source and destination nodes. The performance of multicasting is determined by both the cost of the multicast tree and the maximum delay between the source node and a destination node in the tree. Therefore, constructing efficient multicasting is formulated as a bicriteria Steiner tree problem.

In connection with the problem, the following problem has been considered in the literature [1, 2, 3, 4, 5]. The delay-constrained minimum cost multicast tree problem (DCMCMT) is to construct a multicast tree such that the cost of the tree is minimized while the delay between the source node and a destination node in the tree is bounded by a constant integer. DCMCMT is \mathcal{NP} -hard since it reduces to the Steiner tree problem, which is well-known to be \mathcal{NP} -hard. Chen and Xue proposed a fully polynomial time approximation scheme (FPTAS) for DCMCMT if the number of destination nodes is bounded by a constant [1], while many heuristic algorithms have been proposed in [2, 3, 4, 5]. We present a pseudo-polynomial time algorithm for DCMCMT if the network is series-parallel.

We also consider the following problem, which is another variant of the problem of constructing efficient multicasting. The cost-constrained minimum delay multicast tree problem (CCMDMT) is to construct a multicast tree such that the maximum delay between the source node and a destination node in the tree is minimized while the cost of the tree is bounded by a constant integer. CCMDMT is \mathcal{NP} -hard since it reduces to the cost-constrained shortest path problem (CCSP) which is known to be \mathcal{NP} -hard [6]. In fact, CCMDMT is \mathcal{NP} -hard even for series-parallel networks, since CCSP is \mathcal{NP} -hard for series-parallel networks as mentioned by Chen and Xue [7]. We present in this paper a pseudo-polynomial time algorithm and an FPTAS for CCMDMT if the network is series-parallel. This paper is the first to consider CCMDMT, as far as the authors know.

Due to space limitations most proofs are omitted in the extended abstract and will appear in the final version of the paper.

2 Problems

We consider a connected graph G with vertex set V(G) and edge set E(G). Each edge e is assigned a cost $\gamma(e)$ and a delay $\delta(e)$ which are assumed to be nonnegative integers. The cost of a subgraph H of G, denoted by $\gamma(H)$, is defined as $\gamma(H) = \sum_{e \in E(H)} \gamma(e)$. The delay of a path P in G, denoted by $\delta(P)$, is defined as $\delta(P) = \sum_{e \in E(P)} \delta(e)$. A vertex s is designated as the source and a set D of vertices is designated as the destinations. A tree T is called a multicast tree if $\{s\} \cup D \subseteq V(T)$. The delay of a multicast tree T, denoted by $\delta(T)$, is defined as $\delta(T) = \max\{\delta(P(s,d)) | d \in D, P(s,d): (s,d)$ -path in $T\}$. Let Γ and Δ be positive integers. The delay-constrained minimum cost multicast tree problem (DCMCMT) is to construct a multicast tree T such that $\delta(T) \leq \Delta$ and $\gamma(T)$ is minimized, while the cost-constrained minimum delay multicast tree problem (CCMDMT) is to construct a multicast tree T such that $\gamma(T) \leq \Gamma$ and $\delta(T)$ is minimized.

3 Pseudo-Polynomial Time Algorithms

A graph is said to be *series-parallel* if it contains no subdivision of K_4 as a subgraph. A maximal series-parallel graph is called a 2-*tree*. The 2-trees can be defined recursively as follows: (1) K_2 is a 2-tree on two vertices; (2) Given a 2-tree on n vertices ($n \ge 2$), a graph obtained from G by adding a new vertex adjacent to the ends of an edge of G is a 2-tree on n + 1 vertices. A 2-tree on $n \ge 2$ vertices has 2n - 3 edges by definition.

In this section, we will show an $\mathcal{O}(n\Delta^3)$ time algorithm and an $\mathcal{O}(n^4\delta_{\max}^3)$ time algorithm to solve DCMCMT and CCMDMT, respectively, for a seriesparallel graph G with n vertices, where $\delta_{\max} = \max\{\delta(e)|e \in E(G)\}$. We use methods similar to those used in [7]. We first augment a connected series-parallel graph with n vertices to a 2-tree on n vertices using a linear time algorithm presented in [8]. Each added edge has infinite cost and delay so that the added edges are never chosen in an optimal multicast tree. We next find an optimal multicast tree in the 2-tree. The algorithms are based on the dynamic programming.

3.1 Preliminaries

Let G be a 2-tree and $C_3(G)$ be the set of triangles of G. A tree T_G is defined as follows: $V(T_G) = E(G) \cup C_3(G)$; for any $e \in E(G)$ and $\nabla \in C_3(G)$, $(e, \nabla) \in$ $E(T_G)$ if and only if $e \in E(\nabla)$. T_G thus defined is indeed a tree since G is a 2-tree. T_G is considered as a rooted tree with root r, where r is an edge incident to s in G.

Let $\mathcal{C}(p)$ be the set of all children of $p \in E(G)$ in T_G . Notice that a child of p is a triangle in G. Let $\mathcal{D}(\nabla)$ be the set of triangles which are descendants of $\nabla \in C_3(G)$ in T_G . For $\mathcal{C}'(p) \subseteq \mathcal{C}(p), G[p, \mathcal{C}'(p)]$ is a subgraph of G induced by the edges of triangles in $\bigcup_{\nabla \in \mathcal{C}'(p)} \mathcal{D}(\nabla)$ together with edge p.

Let \prec be a partial order on V(G) satisfying the following conditions:

- $s \prec v$ for all $v \in V(G)$;
- If ∇ is a triangle with $V(\nabla) = \{x, y, z\}$, and edge (x, z) is the parent of ∇ with $x \prec z$, then $x \prec y$ and $y \prec z$.

Such an order can be constructed recursively from the root of T_G as follows: First, we define $s \prec v$ for edge r = (s, v). For every edge p = (x, z) with $x \prec z$, if p has a child triangle C, we define $x \prec y$ and $y \prec z$ for vertex $y \in V(C) \setminus \{x, z\}$. We continue this process until \prec is defined on every pair of endvertices of an edge. Then the transitive reflexive closure of \prec is the desired partial order.

For any edge p = (x, y) with $x \prec y$ and $\mathcal{C}'(p) \subseteq \mathcal{C}(p)$, $H_{\bullet\bullet}^{[p,\mathcal{C}'(p)]}$, $H_{\bullet\circ}^{[p,\mathcal{C}'(p)]}$, $H_{\bullet\circ}^{[p,\mathcal{C}'(p)]}$, and $H_{\bullet\circ}^{[p,\mathcal{C}'(p)]}$ are subgraphs of $G[p,\mathcal{C}'(p)]$ such that each subgraph contains the vertices (destinations) in $D \cap V(G[p, \mathcal{C}'(p)])$ and;

- $H^{[p,\mathcal{C}'(p)]}_{\bullet\bullet}$ is a tree including both x and y,
- $H^{[p,\mathcal{C}'(p)]}_{\bullet\circ}$ is a tree with $x \in V(H^{[p,\mathcal{C}'(p)]}_{\bullet\circ})$ and $y \notin V(H^{[p,\mathcal{C}'(p)]}_{\bullet\circ})$, $H^{[p,\mathcal{C}'(p)]}_{\circ}$ is a tree with $x \notin V(H^{[p,\mathcal{C}'(p)]}_{\circ\circ})$ and $y \in V(H^{[p,\mathcal{C}'(p)]}_{\circ\circ})$,
- $H_{\bullet\bullet}^{[p,\mathcal{C}'(p)]}$ consists of vertex-disjoint two trees $T_x^{[p,\mathcal{C}'(p)]}$ and $T_y^{[p,\mathcal{C}'(p)]}$ such that $x \in V(T_x^{[p,\mathcal{C}'(p)]})$ and $y \in V(T_u^{[p,\mathcal{C}'(p)]})$.

Finally, let $\mathbb{S}_{\Delta} = \{-\infty, 0, 1, \dots, \Delta\}.$

3.2**Functions**

Let p = (x, y) be an edge with $x \prec y$ and $\mathcal{C}'(p) \subseteq \mathcal{C}(p)$.

 $W_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_x, \xi_{xy})$ is the minimum cost of a tree $H_{\bullet\bullet}^{[p, \mathcal{C}'(p)]}$ in $G[p, \mathcal{C}'(p)]$ such that $\max \{\delta(x,d) | d \in D \cap V(G[p,\mathcal{C}'(p)])\} \leq \xi_x$ and $\delta(x,y) \leq \xi_{xy}$, where $\delta(u, v)$ is the delay of (u, v)-path in tree $H^{[p, \mathcal{C}'(p)]}_{\bullet, \bullet}$, and $\xi_x, \xi_{xy} \in \mathbb{S}_{\Delta}$.

 $\overline{W}_{\bullet\bullet}(p,\mathcal{C}'(p);\xi_y,\xi_{xy})$ is the minimum cost of a tree $H^{[p,\mathcal{C}'(p)]}_{\bullet\bullet}$ in $G[p,\mathcal{C}'(p)]$ such that $\max \{\delta(y,d) | d \in D \cap V(G[p,\mathcal{C}'(p)])\} \leq \xi_y$ and $\delta(x,y) \leq \xi_{xy}$, where $\delta(u, v)$ is the delay of (u, v)-path in tree $H^{[p, \mathcal{C}'(p)]}_{\bullet}$, and $\xi_y, \xi_{xy} \in \mathbb{S}_{\Delta}$.

 $W_{\bullet\circ}(p, \mathcal{C}'(p); \xi_x)$ is the minimum cost of a tree $H_{\bullet\circ}^{[p, \mathcal{C}'(p)]}$ in $G[p, \mathcal{C}'(p)]$ such that $\max \{\delta(x,d) | d \in D \cap V(G[p,\mathcal{C}'(p)])\} \leq \xi_x$, where $\delta(x,d)$ is the delay of (x, d)-path in tree $H^{[p, \mathcal{C}'(p)]}_{\bullet \circ}$, and $\xi_x \in \mathbb{S}_{\Delta}$.

 $W_{\circ \bullet}(p, \mathcal{C}'(p); \xi_y)$ is the minimum cost of a tree $H^{[p, \mathcal{C}'(p)]}_{\circ \bullet}$ in $G[p, \mathcal{C}'(p)]$ such that $\max \{\delta(y,d) | d \in D \cap V(G[p,\mathcal{C}'(p)])\} \leq \xi_y$, where $\delta(x,d)$ is the delay of (x, d)-path in tree $H^{[p, \mathcal{C}'(p)]}_{\circ \bullet}$, and $\xi_y \in \mathbb{S}_{\Delta}$.

 $W_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_x, \xi_y)$ is the minimum cost of a forest $H_{\bullet\bullet}^{[p, \mathcal{C}'(p)]}$ in $G[p, \mathcal{C}'(p)]$ such that $\max\{\delta(x, d)|d \in D \cap V(T_x^{[p, \mathcal{C}'(p)]})\} \leq \xi_x$, and $\max\{\delta(y, d)|d \in D \cap V(T_y^{[p, \mathcal{C}'(p)]})\} \leq \xi_y$, where $\delta(x, d)$ is the delay of (x, d)-path in tree $T_x^{[p, \mathcal{C}'(p)]}$ and $\delta(y, d)$ is the delay of (y, d)-path in tree $T_y^{[p, \mathcal{C}'(p)]}$, and $\xi_x, \xi_y \in \mathbb{S}_\Delta$.

 $N_{\circ\circ}(p)$ is defined to be 0 if $G[p, \mathcal{C}(p)]$ has no destination and ∞ otherwise.

For an edge $p = (x, y) \in E(G)$ with $x \prec y$ and $\mathcal{C}'(p) \subseteq \mathcal{C}(p)$, the table $\mathcal{W}(p, \mathcal{C}'(p))$ for p and $\mathcal{C}'(p)$ is the list of values of $W_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_x, \xi_{xy})$, $\overline{W}_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_y, \xi_{xy}), W_{\bullet\circ}(p, \mathcal{C}'(p); \xi_x), W_{\circ\bullet}(p, \mathcal{C}'(p); \xi_y)$, and $W_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_x, \xi_y)$ for every $\xi_x, \xi_y, \xi_{xy} \in \mathbb{S}_\Delta$.

The following is immediate from the definition of functions above.

Lemma 1. For any $\xi \in \{0, 1, ..., \Delta\}$, $\min\{W_{\bullet\circ}(r, \mathcal{C}(r); \xi), W_{\bullet\bullet}(r, \mathcal{C}(r); \xi, \Delta)\}$ is the minimum cost of a multicast tree T of G with $\delta(T) \leq \xi$, where $r = (s, y) \in E(G)$ is the root of T_G .

3.3 Basic Algorithm $BA(G, s, D, \gamma, \delta, \Delta)$

We describe in this subsection a basic algorithm $BA(G, s, D, \gamma, \delta, \Delta)$ which computes $\mathcal{W}(r, \mathcal{C}(r))$ for a 2-tree G with n vertices in $\mathcal{O}(n\Delta^3)$ time.

 $BA(G, s, D, \gamma, \delta, \Delta)$ first computes T_G and chooses an edge incident with s in G as the root of T_G .

Then, $BA(G, s, D, \gamma, \delta, \Delta)$ recursively computes the functions defined in Section 3.2. We distinguish three cases.

Case 1 : $C'(p) = \emptyset$.

For every
$$p = (x, y) \in E(G)$$
 with $x \prec y$, and $\xi_x, \xi_y, \xi_{xy} \in \mathbb{S}_\Delta$,

$$W_{\bullet\bullet}(p,\emptyset;\xi_x,\xi_{xy}) = \begin{cases} \gamma(p) \text{ if the following conditions are satisfied:} \\ (i) \text{ if } y \in D \text{ then } \xi_x \ge \delta(p); \quad (ii) \text{ if } x \in D \text{ then } \xi_x \ge 0; \\ (iii) \xi_{xy} \ge \delta(p), \\ \infty \text{ otherwise.} \end{cases}$$

$$\overline{W}_{\bullet\bullet}(p,\emptyset;\xi_y,\xi_{xy}) = \begin{cases} \gamma(p) \text{ if the following conditions are satisfied:} \\ (i) \text{ if } y \in D \text{ then } \xi_y \ge 0; \quad (ii) \text{ if } x \in D \text{ then } \xi_y \ge \delta(p); \\ (iii) \xi_{xy} \ge \delta(p), \\ \infty \text{ otherwise.} \end{cases}$$

$$W_{\bullet\circ}(p,\emptyset;\xi_x) = \begin{cases} 0 \text{ if } y \notin D, \text{ and if } x \in D \text{ then } \xi_x \ge 0, \\ \infty \text{ otherwise.} \end{cases}$$

$$W_{\circ\bullet}(p,\emptyset;\xi_y) = \begin{cases} 0 \text{ if } x \notin D, \text{ and if } y \in D \text{ then } \xi_y \ge 0, \\ \infty \text{ otherwise.} \end{cases}$$

$$W_{\bullet\bullet}(p,\emptyset;\xi_x,\xi_y) = \begin{cases} 0 \text{ if } x \notin D, \text{ and if } y \in D \text{ then } \xi_y \ge 0, \\ \infty \text{ otherwise.} \end{cases}$$

$$W_{\bullet\bullet}(p,\emptyset;\xi_x,\xi_y) = \begin{cases} 0 \text{ if the following conditions are satisfied:} \\ (i) \text{ if } x \in D \text{ then } \xi_x \ge 0, \\ \infty \text{ otherwise.} \end{cases}$$

Case 2 : $\mathcal{C}'(p) = \{\nabla\}$ for some $\nabla \in \mathcal{C}(p)$. For every $p = (x, z) \in E(G)$ with $\mathcal{C}(p) \neq \emptyset$ and $x \prec z$, for every $\nabla \in \mathcal{C}(p)$ with $V(\nabla) = \{x, y, z\}, E(\nabla) = \{p = (x, z), q = (x, y), t = (y, z)\}, \text{ and } x \prec y \prec z, \text{ and for every } \xi_x, \xi_y, \xi_{xy} \in \mathbb{S}_\Delta$, the functions are computed as follows:

$$\begin{split} & W_{\bullet\bullet}(p, \{\nabla\}; \xi_{x}, \xi_{xz}) = \\ & \min \left\{ \gamma(p) + W_{\bullet\circ}(q, \mathcal{C}(q); \xi'_{x}) + W_{\circ\bullet}(t, \mathcal{C}(t); \xi''_{x}) \mid \xi_{xz} \ge \delta(p), \\ & \xi_{x} \ge \max\{\xi'_{x}, \delta(p) + \xi''_{x}\}, (\xi'_{x}, \xi'_{xy}) \in \mathbb{S}^{2}_{\Delta} \right\}, \\ & \min \left\{ \gamma(p) + W_{\bullet\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi'_{xy}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{y}, \xi''_{x}) \mid \xi_{xz} \ge \delta(p), \\ & \xi_{x} \ge \max\{\xi'_{x}, \xi'_{xy} + \xi''_{y}, \delta(p) + \xi''_{x}\}, (\xi'_{x}, \xi'_{xy}, \xi''_{y}, \xi''_{x}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ \gamma(p) + W_{\bullet\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi'_{y}) + \overline{W}_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{yz}) \mid \xi_{xz} \ge \delta(p), \\ & \xi_{x} \ge \max\{\xi'_{x}, \delta(p) + \xi''_{x}\}, \delta(p) + \xi''_{yz} + \xi'_{y}\}, (\xi'_{x}, \xi'_{x}, \xi''_{x}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ W_{\bullet\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi'_{xy}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{y}, \xi''_{x}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \overline{W}_{\bullet\bullet}(p, \{\nabla\}; \xi_{z}, \xi_{zz}) = \\ & \min \left\{ \gamma(p) + W_{\bullet\circ}(q, \mathcal{C}(q); \xi'_{x}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \mid \xi_{xz} \ge \delta(p), \\ & \xi_{x} \ge \max\{\xi'_{x}, \delta(p) + \xi''_{x}\}, (\xi'_{x}, \xi''_{x}) \in \mathbb{S}^{2}_{\Delta} \right\}, \\ & \min \left\{ \gamma(p) + \overline{W}_{\bullet\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi'_{xy}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \mid \xi_{xz} \ge \delta(p), \\ & \xi_{x} \ge \max\{\xi'_{x}, \delta(p) + \xi''_{x}\}, (\xi') + \overline{\xi}'_{x}, \xi''_{xy}, \xi''_{y}, \xi''_{x}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ \gamma(p) + W_{\bullet\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi'_{xy}) + \overline{W}_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \mid \xi_{xz} \ge \delta(p), \\ & \xi_{x} \ge \max\{\xi'_{x}, \delta(p) + \xi''_{x}\}, (\xi') + \xi''_{x}, \xi''_{y}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ \overline{W}_{\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi'_{xy}) + \overline{W}_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{y}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ \overline{W}_{\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi''_{xy}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ \overline{W}_{\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi''_{xy}) + \overline{W}_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ \overline{W}_{\bullet}(q, \mathcal{C}(q); \xi'_{x}, \xi''_{xy}) + \overline{W}_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & \min \left\{ W_{\bullet\bullet}(q, \mathcal{C}(q); \xi''_{x}, \xi''_{xy}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{xy}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ & W_{\bullet\bullet}(p, \{\nabla\}; \xi_{x}) = \min \left\{ \begin{array}{c} \min \left\{ W_{\bullet\bullet}(q, \mathcal{C}(q); \xi''_{x}, \xi''_{xy}) + W_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_{x}, \xi''_{xy}) \in \mathbb{S}^{4}_{\Delta} \right\}$$

$$\min \left\{ \begin{array}{l} \min \left\{ \mathcal{W}_{\bullet\bullet}(q, \mathcal{C}(q), \zeta_x, \zeta_{xy}) + \mathcal{W}_{\bullet\bullet}(t, \mathcal{C}(t), \zeta_y, \zeta_z) \mid \\ \xi_x \ge \max\{\xi'_x, \xi'_{xy} + \xi''_y\}, \xi_z \ge \xi''_x, (\xi'_x, \xi'_{xy}, \xi''_y, \xi''_z) \in \mathbb{S}^4_{\Delta} \right\}, \\ \min \left\{ \mathcal{W}_{\bullet\bullet}(q, \mathcal{C}(q); \xi'_x, \xi'_y) + \overline{\mathcal{W}}_{\bullet\bullet}(t, \mathcal{C}(t); \xi''_x, \xi''_{yz}) \mid \xi_x \ge \xi'_x, \\ \xi_z \ge \max\{\xi''_x, \xi''_{yz} + \xi'_y\}, (\xi'_x, \xi'_{xy}, \xi''_x, \xi''_{yz}) \in \mathbb{S}^4_{\Delta} \right\} \end{array} \right\}.$$
(5)

 $\textbf{Case 3:} \ \mathcal{C}'(p) = \mathcal{C}''(p) \cup \{\nabla\} \text{ for some } \mathcal{C}''(p) \subseteq \mathcal{C}(p) \text{ and } \nabla \in \mathcal{C}(p) - \mathcal{C}''(p).$

For every $p = (x, y) \in E(G)$, $\mathcal{C}'(p) \subseteq \mathcal{C}(p)$, $\nabla \in \mathcal{C}'(p)$, and $\xi_x, \xi_y, \xi_{xy} \in \mathbb{S}_\Delta$, the functions are computed as follows:

$$W_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_x, \xi_{xy}) = \\ \min \left\{ \begin{aligned} \min \left\{ W_{\bullet\bullet}(p, \mathcal{C}''(p); \xi'_x, \xi'_{xy}) + W_{\bullet\bullet}(p, \{\nabla\}; \xi''_x, \xi''_y) \mid \xi_{xy} \ge \xi'_{xy}, \\ \xi_x \ge \max\{\xi'_x, \xi''_x, \xi''_y + \xi'_{xy}\}, (\xi'_x, \xi'_{xy}, \xi''_x, \xi''_y) \in \mathbb{S}^4_{\Delta} \right\}, \\ \min \left\{ W_{\bullet\bullet}(p, \mathcal{C}''(p); \xi'_x, \xi'_y) + W_{\bullet\bullet}(p, \{\nabla\}; \xi''_x, \xi''_{xy}) \mid \xi_{xy} \ge \xi''_{xy}, \\ \xi_x \ge \max\{\xi'_x, \xi''_x, \xi''_y + \xi''_{xy}\}, (\xi'_x, \xi'_y, \xi''_x, \xi''_{xy}) \in \mathbb{S}^4_{\Delta} \right\} \end{aligned} \right\},$$
(6)

$$W_{\bullet\bullet}(p, \mathcal{C}'(p); \xi_{y}, \xi_{xy}) = \\ \min \left\{ \begin{array}{l} \min \left\{ \overline{W}_{\bullet\bullet}(p, \mathcal{C}''(p); \xi'_{y}, \xi'_{xy}) + W_{\bullet\bullet}(p, \{\nabla\}; \xi''_{x}, \xi''_{y}) \mid \xi_{xy} \geq \xi'_{xy}, \\ \xi_{y} \geq \max\{\xi'_{y}, \xi''_{y}, \xi''_{x} + \xi'_{xy}\}, (\xi'_{y}, \xi'_{xy}, \xi''_{x}, \xi''_{y}) \in \mathbb{S}^{4}_{\Delta} \right\}, \\ \min \left\{ W_{\bullet\bullet}(p, \mathcal{C}''(p); \xi'_{x}, \xi'_{y}) + \overline{W}_{\bullet\bullet}(p, \{\nabla\}; \xi''_{y}, \xi''_{xy}) \mid \xi_{xy} \geq \xi''_{xy}, \\ \xi_{y} \geq \max\{\xi'_{y}, \xi''_{y}, \xi''_{x} + \xi''_{xy}\}, (\xi'_{y}, \xi''_{y}, \xi''_{xy}, \xi''_{xy}) \in \mathbb{S}^{4}_{\Delta} \right\},$$
(7)

$$W_{\bullet\circ}(p,\mathcal{C}'(p);\xi_x) = W_{\bullet\circ}(p,\mathcal{C}''(p);\xi_x) + W_{\bullet\circ}(p,\{\nabla\};\xi_x),\tag{8}$$

$$W_{\circ\bullet}(p, \mathcal{C}'(p); \xi_y) = W_{\circ\bullet}(p, \mathcal{C}''(p); \xi_y) + W_{\circ\bullet}(p, \{\nabla\}; \xi_y), \tag{9}$$

$$W_{\bullet\bullet}(p,\mathcal{C}'(p);\xi_x,\xi_y) = W_{\bullet\bullet}(p,\mathcal{C}''(p);\xi_x,\xi_y) + W_{\bullet\bullet}(p,\{\nabla\};\xi_x,\xi_y).$$
(10)

The computation of the tables for functions proceeds as follows. We first compute $\mathcal{W}(p, \mathcal{C}(p)) = \mathcal{W}(p, \emptyset)$ for every leaf p of T_G as in Case 1 above.

For every triangle ∇ with parent p and children q and t, $\mathcal{W}(p, \{\nabla\})$ is computed using tables $\mathcal{W}(q, \mathcal{C}(q))$ and $\mathcal{W}(t, \mathcal{C}(t))$ as in Case 2.

For every $p \in E(G)$ with $\mathcal{C}(p) = \{\nabla_1, \nabla_2, \dots, \nabla_{|\mathcal{C}(p)|}\}, \mathcal{W}(p, \mathcal{C}(p))$ is computed as follows. Let $\mathcal{C}^{(i)}(p) = \{\nabla_1, \nabla_2, \dots, \nabla_i\}$ for $1 \leq i \leq |\mathcal{C}(p)|$. $\mathcal{W}(p, \mathcal{C}^{(i)}(p))$ is computed using $\mathcal{W}(p, \mathcal{C}^{(i-1)}(p))$ and $\mathcal{W}(p, \{\nabla_i\})$ as in Case 3 for $2 \leq i \leq |\mathcal{C}(p)|$.

Finally, $BA(G, s, D, \gamma, \delta, \Delta)$ outputs $\mathcal{W}(r, \mathcal{C}(r))$.

3.4 Analysis of $BA(G, s, D, \gamma, \delta, \Delta)$

We use the following lemmas to prove Theorem 1 below. The proofs of the lemmas are omitted in the extended abstract.

Lemma 2. BA $(G, s, D, \gamma, \delta, \Delta)$ computes W(r, C(r)), correctly.

Lemma 3. $\mathcal{W}(p, \emptyset)$ is computed in $\mathcal{O}(\Delta^2)$ time for any leaf p of T_G .

Lemma 4. Let ∇ be a triangle with parent p and children q and t. Given $\mathcal{W}(q, \mathcal{C}(q))$ and $\mathcal{W}(t, \mathcal{C}(t)), \mathcal{W}(p, \{\nabla\})$ is computed in $\mathcal{O}(\Delta^3)$ time. \Box

Lemma 5. Let $p \in E(G)$, $C(p) = \{\nabla_1, \nabla_2, \dots, \nabla_{|\mathcal{C}(p)|}\}$, and $\mathcal{C}^{(i)}(p) = \{\nabla_1, \nabla_2, \dots, \nabla_i\}$ for $1 \leq i \leq |\mathcal{C}(p)|$. Given $\mathcal{W}(p, \mathcal{C}^{(i-1)}(p))$ and $\mathcal{W}(p, \{\nabla_i\})$, $\mathcal{W}(p, \mathcal{C}^{(i)}(p))$ is computed in $\mathcal{O}(\Delta^3)$ time for $2 \leq i \leq |\mathcal{C}(p)|$.

Theorem 1. For a 2-tree G on n vertices, $BA(G, s, D, \gamma, \delta, \Delta)$ computes W(r, C(r)) in $O(n\Delta^3)$ time.

Proof. The tables $\mathcal{W}(p, \emptyset)$ for all leaves p can be computed in $\mathcal{O}(n\Delta^2)$ time by Lemma 3. Since the number of triangles is $\mathcal{O}(n)$, the tables $\mathcal{W}(p, \{\nabla\})$ for all triangles ∇ can be computed in $\mathcal{O}(n\Delta^3)$ time by Lemma 4. By Lemma 5, $\mathcal{W}(p, \mathcal{C}(p))$ can be computed in $\mathcal{O}(|\mathcal{C}(p)|\Delta^3)$ time. Since $\sum_{p \in E(G)} |\mathcal{C}(p)| = \mathcal{O}(n)$, the tables $\mathcal{W}(p, \mathcal{C}(p))$ for all edges p can be computed in $\mathcal{O}(n\Delta^3)$ time. It follows that $BA(G, s, D, \gamma, \delta, \Delta)$ computes $\mathcal{W}(r, \mathcal{C}(r))$ in $\mathcal{O}(n\Delta^3)$ time by Lemma 2. \Box

By Lemma 1 and Theorem 1, $BA(G, s, D, \gamma, \delta, \Delta)$ computes the minimum cost of a multicast tree with delay at most ξ for any $\xi \in \{0, 1, \ldots, \Delta\}$. If we perform some bookkeeping operations such as recording how the minimum was achieved during the computation of the tables for functions, we can construct a delay-constrained minimum cost multicast tree in the same time complexity. Thus, we have the following.

Corollary 1. Given a 2-tree G on n vertices, s, D, γ , δ , Δ , and an integer ξ , $0 \leq \xi \leq \Delta$, a minimum cost multicast tree T with $\delta(T) \leq \xi$ can be constructed in $\mathcal{O}(n\Delta^3)$ time.

We denote by $MT(G, s, D, \gamma, \delta, \Delta, \xi)$ such an $\mathcal{O}(n\Delta^3)$ time algorithm constructing a minimum cost multicast tree T with $\delta(T) \leq \xi$ for a given 2-tree G, $s, D, \gamma, \delta, \Delta$, and an integer $\xi, 0 \leq \xi \leq \Delta$.

3.5 Pseudo-Polynomial Time Algorithm for DCMCMT

Given a connected series-parallel graph G' with $\cos t$ and delay functions γ' and δ' , we denote by $Ext(G', \delta', \gamma')$ a linear time procedure for augmenting G' to a 2-tree G with V(G) = V(G') [8], and extending γ' and δ' to γ and δ , respectively, by defining $\gamma(e) = \infty$ and $\delta(e) = \infty$ for each $e \in E(G) - E(G')$, and $\gamma(e) = \gamma'(e)$ and $\delta(e) = \delta'(e)$ for each $e \in E(G')$. Then, it is easy to see that Algorithm 1 shown in Fig. 1. solves DCMCMT for series-parallel graphs, and we have the following by Theorem 1.

Theorem 2. For a series-parallel graph G with n vertices and a positive integer Δ , Algorithm 1 solves DCM-CMT in $\mathcal{O}(n\Delta^3)$ time. \Box

Input a series-parallel graph $G', s \in V(G'), D \subseteq V(G'), \gamma' : E(G') \to \mathbb{N}, \delta' : E(G') \to \mathbb{N}, \Delta \in \mathbb{Z}^+.$

Output a minimum cost multicast tree T with delay at most Δ .

begin

 $\begin{array}{c} \operatorname{Ext}(G',\gamma',\delta');\\ \operatorname{BA}(G,s,D,\gamma,\delta,\Delta);\\ \operatorname{MT}(G,s,D,\gamma,\delta,\Delta,\Delta);\\ \operatorname{if}\gamma(T)<\infty\\ \operatorname{return}T;\\ \operatorname{else}\\ \operatorname{return}"\operatorname{NO"};\\ \operatorname{endif}\\ \operatorname{end}\end{array}$

Fig. 1. Algorithm 1

3.6 Pseudo-Polynomial Time Algorithm for CCMDMT

Given a cost bound Γ and the table $\mathcal{W}(r, \mathcal{C}(r))$ for functions, we denote by MIN_DELAY($\Gamma, \mathcal{W}(r, \mathcal{C}(r))$) a linear time procedure for computing the minimum ξ satisfying $\min\{W_{\bullet\bullet}(r, \mathcal{C}(r); \xi, \Delta), W_{\bullet\circ}(r, \mathcal{C}(r); \xi)\} \leq \Gamma$ if exists. It returns ∞ if there exists no such ξ .

Since the number of edges of multicast tree is at most n-1, the maximum delay of a multicast tree is at most $(n-1)\delta_{\max}$, where $\delta_{\max} = \max_{e \in E(G')} \delta'(e)$. Thus, it is easy to see that Algorithm 2 shown in Fig. 2 is a pseudo-polynomial time algorithm for CCMDMT, and we have the following by Theorem 1.

Theorem 3. For a series-parallel graph G with n vertices and a nonnegative integer Γ , Algorithm 2 solves CCMDMT in $\mathcal{O}(n^4 \delta_{\max}{}^3)$ time if $\delta_{\max} \geq 1$. **Input** a series-parallel graph G', $s \in V(G'), D \subseteq V(G'),$ $\gamma': E(G') \to \mathbb{N}, \ \delta': E(G') \to \mathbb{N},$ $\Gamma \in \mathbb{Z}^+$. **Output** a minimum delay multicast tree T with cost at most Γ . begin $\delta_{\max} := \max_{e \in E(G')} \delta(e);$ $\Delta' := (n-1)\delta_{\max};$ $\operatorname{Ext}(G', \gamma', \delta');$ $BA(G, s, D, \gamma, \delta, \Delta');$ MIN_DELAY($\Gamma, \mathcal{W}(r, \mathcal{C}(r))$); if $\xi < \infty$ $MT(G, s, D, \gamma, \delta, \Delta, \xi);$ return T; else return "NO"; endif end

Fig. 2. Algorithm 2

4 FPTAS for CCMDMT

We use standard techniques [7, 6, 9, 10] to turn $BA(G, s, D, \gamma, \delta, \Delta)$ into an FP-TAS for CCMDMT. We show in Section 4.1 a pair of upper and lower bounds U and L for the minimum delay of a cost constrained multicast tree such that $U/L \leq n - 1$. For any $\varepsilon > 0$, we show in Section 4.2 a $(1 + \varepsilon)$ -approximation algorithm for CCMDMT. The algorithm runs in $\mathcal{O}(n^7/\varepsilon^3)$ time, provided that we have a pair of upper and lower bounds U and L for the delay of a cost constrained multicast tree such that $U/L = \mathcal{O}(n)$. It follows that we have an FPTAS for CCMDMT.

4.1 Upper and Lower Bounds for Minimum Delay

We use a technique similar to [7]. Let $\nu_1 < \nu_2 < \cdots < \nu_k$ be different edge delays, and γ_j be the cost function defined as $\gamma_j(e) = \gamma(e)$ if $\delta(e) \leq \nu_j$, and $\gamma_j(e) = \infty$ otherwise. Let T_j be a minimum cost multicast tree of G for γ_j , and J be the minimum j such that $\gamma_j(T_j) \leq \Gamma$.

By the definition of J, the minimum delay of a cost constrained multicast tree is at least ν_J and at most $(n-1)\nu_J$. Since such J and also T_J can be computed in $\mathcal{O}(n \log n)$ time [7], we have the following. **Theorem 4.** A pair of upper and lower bounds U and L for the minimum delay of a cost constrained multicast tree satisfying U/L = n - 1 can be computed in $\mathcal{O}(n \log n)$ time. Moreover, a multicast tree T_J with cost at most Γ and delay at most U can also be computed in $\mathcal{O}(n \log n)$ time.

Given a 2-tree G with source s and destinations D, cost and delay functions γ and δ , and a positive integer Γ , we denote by COMP_UL($G, s, D, \gamma, \delta, \Gamma$) an $\mathcal{O}(n \log n)$ time procedure for computing upper and lower bounds U and L with $U/L \leq n-1$.

4.2 FPTAS for CCMDMT

For any $\alpha > 0$, let δ_{α} be a delay function defined as $\delta_{\alpha}(e) = \lfloor \alpha \delta(e) \rfloor$ for any $e \in E(G)$. Let T_{α} be a minimum delay multicast tree with cost at most Γ for δ_{α} and $\text{OPT}(\delta_{\alpha}) = \delta_{\alpha}(T_{\alpha})$. Notice that T_1 is a minimum delay multicast tree with cost at most Γ for $\delta = \delta_1$. We denote by P_{α} a maximum delay path in T_{α} for δ_{α} .

By the definition of δ_{α} , we have $\delta(e) \geq \delta_{\alpha}(e)/\alpha$ and

$$\delta(e) < (\delta_{\alpha}(e) + 1)/\alpha \tag{11}$$

for any $e \in E(G)$. If we denote by P'_1 a maximum delay path of T_1 for δ_{α} , $OPT(\delta) = \sum_{e \in E(P_1)} \delta(e) \ge \sum_{e \in E(P'_1)} \delta(e) \ge \sum_{e \in E(P'_1)} \delta_{\alpha}(e)/\alpha \ge \sum_{e \in E(P_{\alpha})} \delta_{\alpha}(e)/\alpha$, where the second inequality follows from $\delta(e) \ge \delta_{\alpha}(e)/\alpha$. Thus, we have

$$OPT(\delta) \ge \delta_{\alpha}(T_{\alpha})/\alpha$$

= $OPT(\delta_{\alpha})/\alpha$. (12)
Moreover, if we set α = $(n - 1)/\varepsilon L$, and denote by P'_{α} a maximum delay path in T_{α} for δ , we have

$$\delta(T_{\alpha}) = \sum_{e \in E(P'_{\alpha})} \delta(e)$$

$$< \frac{1}{\alpha} \sum_{e \in E(P'_{\alpha})} (\delta_{\alpha}(e) + 1) \quad (13)$$

$$\leq \frac{1}{\alpha} |E(P'_{\alpha})| + \frac{1}{\alpha} \sum_{e \in E(P'_{\alpha})} \delta_{\alpha}(e)$$

$$\leq \frac{n-1}{\alpha} + \frac{1}{\alpha} \sum_{e \in E(P_{\alpha})} \delta_{\alpha}(e)$$

$$= \varepsilon L + \frac{1}{\alpha} \text{OPT}(\delta_{\alpha})$$

$$\leq \varepsilon L + \text{OPT}(\delta) \quad (14)$$

$$\leq (1 + \varepsilon) \text{OPT}(\delta),$$

Input a series-parallel graph G', $s \in V(G'), D \subseteq V(G'),$ $\gamma' : E(G') \to \mathbb{N}, \ \delta' : E(G') \to \mathbb{N},$ $\Gamma \in \mathbb{Z}^+, \varepsilon > 0.$

Output a multicast tree T with cost at most Γ and delay at most $(1 + \varepsilon)$ OPT (δ') .

begin

$$\begin{split} & \operatorname{Ext}(G',\gamma',\delta');\\ & \operatorname{COMP_UL}(G,s,D,\gamma,\delta,\Gamma);\\ & \alpha := (n-1)/\varepsilon L;\\ & \delta_{\alpha}(e) := \lfloor \alpha \delta(e) \rfloor \; \forall e \in E(G);\\ & \Delta_{\alpha} := \alpha U;\\ & \operatorname{BA}(G,s,D,\gamma,\delta_{\alpha},\Delta_{\alpha});\\ & \operatorname{MIN_DELAY}(\Gamma,\mathcal{W}(r,\mathcal{C}(r)));\\ & \operatorname{MT}(G,s,D,\gamma,\delta_{\alpha},\Delta_{\alpha},\xi);\\ & \mathbf{return}\;T;\\ & \mathbf{end} \end{split}$$

Fig. 3. Algorithm 3

where inequality (13) and (14) follow from (11) and (12), respectively.

Thus, we conclude that Algorithm 3 shown in Fig. 3 is an FPTAS for CCMDMT. Since $\Delta_{\alpha} = (n-1)U/\varepsilon L = \mathcal{O}(n^2/\varepsilon)$, we have the following by Theorem 2.

Theorem 5. For a series-parallel graph G with n vertices and a non-negative integer Δ , Algorithm 3 computes a $(1 + \varepsilon)$ -approximate solution for CCMDMT in $\mathcal{O}(n \log n + n^7/\varepsilon^3)$ time.

Finally, it should be noted that our method to obtain FPTAS for CCMDMT cannot apply to DCMCMT in a straightforward way, since Δ can be exponentially large.

References

- 1. Chan, G., Xue, G.: *k*-pair delay constrained minimum cost routing in undirected networks. ACM-SIAM Symposium on Discrete Algorithms (2001) 230–231
- Kompella, V., Pasquale, J., Polyzos, G.: Multicast routing for multimedia communication. IEEE/ACM Transactions on Networking 1 (1993) 286–292
- Parsa, M., Zhu, Q., G.-L.-Aceves, J.J.: An iterative algorithm for delay-constrained minimum-cost multicast. IEEE/ACM Trans. Networking 6 (2001) 213–219
- Sriram, R., Manimaran, G., Murthy, C.: Algorithms for delay-constrained low-cost multicast tree construction. Computer Communications 21 (1998) 1693–1706
- Youssef, H., A.-Mulhem, A., Sait, S., Tahir, M.: QoS-driven multicast tree generation using tabu search. Computer Communications 25 (2002) 1140–1149
- 6. Hassin, R.: Approximation schemes for the restricted shortest path problem. Mathematics of Operations Research **17** (1992) 36–42
- Chen, G., Xue, G.: A PTAS for weight constrained steiner trees in series-parallel graphs. Theoretical Computer Science 304 (2003) 237–247
- Wald, J., Colbourn, C.: Steiner trees, partial 2-trees and minimum IFI networks. Networks 13 (1983) 159–167
- 9. Lorenz, D., Raz, D.: A simple efficient approximation scheme for the restricted path problem. Operations Research Letters **28** (2001) 213–219
- Warburton, A.: Approximation of pareto optima in multiple-objective shortest path problems. Operations Research 35 (1987) 70–79