Theoretical Computer Science 412 (2011) 240–245

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Size-constrained tree partitioning: Approximating the multicast k-tree routing problem

Zhipeng Cai, Randy Goebel, Guohui Lin *

Department of Computing Science, University of Alberta. Edmonton, Alberta T6G 2E8, Canada

ARTICLE INFO

Keywords:
Capacitated multicast tree routing
Approximation algorithm
Tree partitioning

ABSTRACT

In the multicast k-tree routing problem, a data copy is sent from the source node to at most k destination nodes in every transmission. The goal is to minimize the total cost of sending data to all destination nodes, which is measured as the sum of the costs of all routing trees. This problem was formulated out of optical networking and has applications in general multicasting. Several approximation algorithms, with increasing performance, have been proposed in the last several years; the most recent ones rely heavily on a *tree partitioning* technique. In this paper, we present a further improved approximation algorithm along the line. The algorithm has a worst-case performance ratio of $\frac{5}{4}\rho + \frac{3}{2}$, where ρ denotes the best approximation ratio for the Steiner minimum tree problem. The proofs of the technical routing lemmas also provide some insights into why such a performance ratio could be the best possible that one can get using this tree partitioning technique.

1. Introduction

Multicast is a point-to-multipoint communication in which a source node sends data to multiple destinations [2,19,13,11,18]. In computer and communication networks supporting multimedia applications, such as news feed and video distribution, multicast is an important service. Implementing multicast on local area networks (LANs) is easy because nodes connected to a LAN usually communicate over a broadcast network. In contrast, implementing multicast on wide area networks (WANs) is still quite challenging [20,9], because nodes connected to a WAN typically communicate via a switched/routed network. Basically, to perform multicast in WANs, the source node and all the destination nodes must be interconnected. So, finding a multicast routing in a WAN is equivalent to finding a multicast tree T in the network such that T spans the source node and all the destination nodes. The objective of the routing is to minimize the *cost* of T, which is defined to be the total weight of edges in T.

In certain networks such as wavelength-division multiplexing (WDM) optical networks with limited light-splitting capabilities, during each transmission, only a limited number of destination nodes can be assigned to receive the data copies sent from the source node. A routing model for such networks, called the *multi-tree model* [14,9,10,12], has been introduced in the literature. Under this model, we are interested in the problem of finding a collection of routing trees such that each tree spans the source node and a limited number of destination nodes that are assigned to receive data copies, and every destination node must be designated to receive a data copy in one of the routing trees. We call this problem the *capacitated multicast routing problem*. In particular, when the number of destination nodes in each routing tree is limited to a pre-specified number k, we call it the *multicast k-tree routing* (kMTR) problem. Correspondingly, a feasible routing solution is called a *k-tree routing*. Compared with the traditional multicast routing model without the capacity constraint — the *Steiner*

* Corresponding address: Department of Computing Science, University of Alberta, Athabasca Hall 2-21, Edmonton, Albert T6G 2E8, Canada. Tel.: +1 780 492 3737; fax: +1 780 492 1071.

E-mail addresses: zhipeng@cs.ualberta.ca (Z. Cai), goebel@cs.ualberta.ca (R. Goebel), ghlin@cs.ualberta.ca (G. Lin).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2009.05.031

minimum tree (SMT) problem, which allows any number of receivers in the routing tree, this simpler model makes multicast easier and more efficient to be implemented, at the expense of increasing the total routing cost.

We next formally define the kMTR problem. For a graph G, we denote its node set by $V(G)$. The underlying communication network is modeled as a triple (G, s, D), where G is a simple, undirected, and edge-weighted complete graph, $s \in V(G)$ is the *source* node, and $D = \{d_1, d_2, \ldots, d_n\} \subseteq V(G) - \{s\}$ is the set of *destination* nodes. The weight of each edge e in G, denoted by $w(e)$, is non-negative and represents the routing cost of e. The *weight* (or *cost*, used interchangeably) of a subgraph T of G, denoted by $w(T)$, is the total weight of edges in T. Let k be a given positive integer. The kMTR problem asks for a *minimum-weight k-tree routing*, that is, a partition of D into disjoint sets $D_1, D_2, \ldots, D_\ell$, such that each D_i contains no more than k destination nodes, and a Steiner tree T_i spanning the source node s and the destination nodes in D_i for $i = 1, 2, \ldots, \ell$, such that $\sum_{i=1}^{\ell} w(T_i)$ is minimized. It is worth pointing out that the union of all these routing trees does not necessarily remain as a tree, since some destination nodes in D_i could appear as *Steiner* nodes for purely routing purpose in T_j for $j \neq i$ (i.e., they are not counted as destination nodes towards D_j). In the following, we assume without loss of generality that the edge weight function (the shortest path metric) satisfies triangle inequality.

When $k \geq |D|$, kMTR reduces to the well-known SMT problem. The SMT problem is NP-hard, and its current best approximation ratio is $\rho \approx 1.55$ [8,17]. (We reserve ρ to denote this best approximation ratio throughout the paper.) On the other hand, when $k = 1$ or 2, kMTR can be solved efficiently [9,10]. The algorithmically most interesting case is $3 \leq k < |D|$, where kMTR differs from the SMT problem yet remains NP-hard [3,15].

Let $\{T_1^*, T_2^*, \ldots, T_m^*\}$ be the set of trees in an optimal k-tree routing. Let $w(T_j^*)$ denote the weight of tree T_j^*, the sum of the weights of edges in T_j^*. Let $R^* = \sum_{j=1}^{m} w(T_j^*)$ be the weight of this optimal k-tree routing. Since every destination node d_i in tree T_j^* satisfies $w(s, d_i) \leq w(T_j^*)$, due to non-negative edge weights, we have

$$\sum_{i=1}^{n} w(s, d_i) \leq k \times R^*. \tag{1.1}$$

Let T be a tree in (G, s, D) containing a subset D_T of destination nodes. For ease of presentation, we define the *size* of tree T to be $|D_T|$, though T might contain other non-destination nodes. Tree T can be used in a feasible k-tree routing to route as many as k destination nodes in D_T. For doing so, the incurred routing cost will be the weight of tree T, $w(T)$, plus a *connection cost*, $c(T)$, which is measured as the minimum weight of edges between the source node s and all nodes in $V(T)$. It follows that, if source node s is in $V(T)$, then $c(T) = 0$; in the other cases, we can always have

$$c(T) \leq \min_{d \in D_T} w(s, d) \leq \frac{1}{|D_T|} \sum_{d \in D_T} w(s, d). \tag{1.2}$$

As noted by Jothi and Raghavachari [12], an algorithm presented by Altinkemer and Gavish [1] about 20 years ago for a slightly different problem serves as a $(2\rho + 1)$-approximation algorithm for kMTR. Hu and his colleagues [9,10] were probably the first to study the kMTR problem, and they presented an approximation algorithm starting with a Hamiltonian cycle on $s \cup D$ obtained by the $\frac{3}{2}$-approximation algorithm for the metric *Traveling Salesman Problem* (TSP) [7], partitioning it into segments each containing exactly k destination nodes (except one segment), and then connecting these segments to source node s separately via the minimum weight edge to form a k-tree routing. Note that the weight of an optimal TSP tour on $s \cup D$ is upper bounded by $2R^*$. By Eqs. (1.1) and (1.2), the connection cost for this k-tree routing is bounded from above by $\frac{1}{k} \sum_{i=1}^{n} w(s, d_i) \leq R^*$. Therefore, the weight of this k-tree routing is at most $(\frac{3}{2} \times 2 + 1)R^* = 4R^*$, and their algorithm is a 4-approximation, an improvement over the $(2\rho + 1)$-approximation [1,12].

Lin [15] proposed to start with a good Steiner tree on $s \cup D$, obtained by any currently the best approximation algorithms for the SMT problem, partition it into small trees of size at most k without duplicating any edges (and thus not increase the weight of the Steiner tree), and then connect each such obtained tree to the source via the minimum weight edge. He demonstrated that in a top-down fashion the Steiner tree can be partitioned into a collection of feasible routing trees, such that each tree has size in $(\frac{1}{6}k, k]$ and its connection cost is less than or equal to the minimum edge weight of a distinct set of at least $\frac{5}{12}k$ source-to-destination edges. Equivalently speaking, on average, each routing tree has size at least $\frac{5}{12}k$. By Eqs. (1.1) and (1.2), the connection cost for this k-tree routing is bounded from above by $\frac{12}{5} \times \frac{1}{k} \sum_{i=1}^{n} w(s, d_i) \leq 2.4R^*$. Thus it is an improved $(\rho + 2.4)$-approximation algorithm for kMTR.

Subsequently, two groups of researchers [3,6,12] independently designed $(\rho + 2)$-approximation algorithms for kMTR. Cai and his colleagues [3,6] continued the study from [9,10,15]; Jothi and Raghavachari [12] were directed from [1] to consider a variant of kMTR in which the destination nodes have varying integral amounts of request and no destination nodes can be used as Steiner points to assist the routing. When the given network is completely connected, Jothi and Raghavachari [12] designed an approximation algorithm for the variant. This approximation algorithm turns out to be a $(\rho+2)$-approximation algorithm for kMTR. The two $(\rho+2)$-approximation algorithms are surprisingly similar in the design nature, in that both algorithms start with a Steiner tree on $s \cup D$, partition it into feasible routing trees without duplicating any edges, and then connect each such obtained tree to the source via the minimum weight edge. The difference between them is that the algorithm by Cai et al. partitions the tree in a top-down fashion to guarantee that, equivalently speaking, each tree has size in the range $[\frac{1}{2}k, k]$, while the algorithm by Jothi and Raghavachari cuts iteratively from the Steiner tree a

Table 1
A historical record of the approximation algorithms for kMTR.

Year	References	Performance ratio
1988	[1]	$2\rho + 1 = 4.10$
2004	[9,10]	4
2004/2005	[3,15]	$\rho + 2.4 = 3.95$
2004/2005	[3,12,6]	$\rho + 2 = 3.55$
2008	[16]	$\frac{4}{3}\rho + \frac{3}{2} = 3.5667$
2008	[5]	$\frac{5}{4}\rho + \frac{8}{5} = 3.5375$
2009	[4]	$\frac{5}{4}\rho + \frac{\sqrt{2089}+77}{80} = 3.4713$
2009	This paper	$\frac{5}{4}\rho + \frac{3}{2} = 3.4375$

subtree of size in the range $[\frac{1}{2}k, k]$. It follows from Eqs. (1.1) and (1.2) that the connection costs of both k-tree routings are at most $2R^*$, implying that the two algorithms are both $(\rho + 2)$-approximations for kMTR.

All the above approximation algorithms [1,9,10,3,6,12] show that the total cost of a k-tree routing consists of two components: the weight of the initial infeasible solution subgraph (such as a Hamiltonian cycle or a Steiner tree) and the connection cost depending on the size range of the achieved routing trees using Eqs. (1.1) and (1.2). These two components are seemingly independent but actually closely related to each other. Efforts have been invested in developing better tree partitioning schemes without increasing the total weight of the routing trees too much, compared with the weight of the initial infeasible solution subgraph. For example, Morsy and Nagamochi [16] presented a tree partitioning scheme that can, roughly speaking, guarantee a lower bound of $\frac{2}{3}k$ on the size of the routing trees at a cost of $\frac{1}{3}$ the weight of the starting Steiner tree. This gives a new approximation algorithm with a worst-case performance ratio of $(\frac{4}{3}\rho + \frac{3}{2})$. Unfortunately, this is not an improvement over the $(\rho + 2)$-approximation algorithms unless $\rho < 1.5$ [16]. Cai et al. [5] were able to do better. Last year, they presented at COCOA 2008 a slightly different but better tree partitioning scheme to guarantee a lower bound of $\frac{5}{8}k$ on the size of the routing trees at the expense of $\frac{1}{4}$ the weight of the starting Steiner tree. Their algorithm is thus a $(\frac{5}{4}\rho + \frac{8}{5})$-approximation, a genuine improvement over the $(\rho + 2)$-algorithms, given that $\rho \approx 1.55$. Most recently, Cai et al. [4] further improved their tree partitioning scheme to guarantee a better lower bound of $\frac{80}{\sqrt{2089}+77}k$ on the size of the routing trees, giving rise to a $(\frac{5}{4}\rho + \frac{\sqrt{2089}+77}{80})$-approximation algorithm for kMTR.

In this paper, we show that at the same expense of $\frac{1}{4}$ the weight of the starting Steiner tree, the lower bound of $\frac{2}{3}k$ on the size of the routing trees can be guaranteed. This results in a $(\frac{5}{4}\rho + \frac{3}{2})$-approximation algorithm for the kMTR problem. On one hand, this algorithm outperforms the previous best by 0.0338; on the other hand, it beats the $(\frac{4}{3}\rho + \frac{3}{2})$-approximation algorithm by Morsy and Nagamochi [16] for any possible value of ρ. Table 1 contains a historical record of the approximation algorithms for kMTR, to the best of our knowledge. Nevertheless, it is worth pointing out that our new design and analysis presented here are nothing but a more careful case analysis, yet we doubt that any further improvement is achievable along this tree partitioning line of research, if no new techniques are introduced.

2. A $(\frac{5}{4}\rho + \frac{3}{2})$-approximation algorithm for kMTR

We want to point out that the readers should also read our preceding works [5,4] for a full understanding of the following algorithm.

Following previous design, we first apply the best known ρ-approximation algorithm for the SMT problem to obtain a Steiner tree T^0 on $\{s\} \cup D$ in the underlying network (G, s, D). As discussed earlier, $w(T^0) \leq \rho R^*$, where R^* denotes the weight of an optimal k-tree routing. Root tree T^0 at source s and denote it as T_s^0. One may use $\frac{n}{k}$ copies of tree T_s^0 to form a k-tree routing, which is apparently very expensive. In what follows, we use T_v to denote the rooted subtree at v in T_s^0, and D_v denotes the associated destination node set of T_v (i.e., $D_v = D \cap T_v$). Also, for a child u of v in T_v, the subtree T_u together with edge (v, u) is called the *branch rooted at* v *and containing* u. In the algorithm to be presented, we will iteratively cut from T_s^0 a rooted subtree T_r of certain size if $|T_s^0| > \frac{4}{3}k$. This cutting process does not duplicate any edge and thus would not increase the tree weight. Nonetheless, (at most) one node might need to be duplicated for connectivity purpose. We then show that, using tree T_r, the destination nodes in D_r can be routed at a cost

$$\leq \frac{5}{4}w(T_r) + \frac{3}{2} \times \frac{1}{k}\sum_{d \in D_r} w(s, d). \tag{2.1}$$

Given the rooted tree T_s^0, for any internal node v, we have the destination node set D_v for the rooted subtree T_v. For ease of presentation, we say v has size $|D_v|$. We assume the non-trivial case that source s has size $> \frac{4}{3}k$. At each iteration, the cutting process examines T_s^0 in a bottom-up fashion. First of all, for any u_1, u_2 being two children of v, if the sum of their sizes is $\leq k$, then the corresponding two branches are merged into one (via a copy of v, say v', and a dummy edge (v, v') of cost 0). If the process locates a node r of size in the range $[\frac{2}{3}k, k]$, it cuts T_r off the tree and the destination nodes in D_r are routed by Lemma 2.2 with the routing cost bounded by Eq. (2.1); otherwise, define a node as *extremely big* if its size is $>k$ but none of its children has size $>k$. It follows that every extremely big node v has at least two children (of size $<\frac{2}{3}k$).

If there exists an extremely big node r having more than two children, the process cuts off exactly three branches rooted at r, together with a copy of r, to form a three-branch subtree T_r. From the fact that the total size of every pair of two subtrees is greater than k, one concludes that the size of T_r is in the open interval $(\frac{3}{2}k, 2k)$. The destination nodes in D_r are routed by Lemma 2.4 with the routing cost accordingly bounded by Eq. (2.1). When every extremely big node has only two children, the process searches for a node of size in the range $[\frac{4}{3}k, 2k]$.

If such a node is found, say u, then u will have exactly one descendent, say r, which is extremely big. The process cuts T_u off the Steiner tree and re-roots it at r, subsequently denoted as T_r. Two cases are distinguished depending on the size of D_r, the set of destination nodes in T_r. If $|D_r| \geq \frac{3}{2}k$, then these destination nodes are routed again by Lemma 2.4 with the routing cost accordingly bounded by Eq. (2.1); if $|D_r| < \frac{3}{2}k$, then the destination nodes are routed by Lemma 2.3 with the routing cost accordingly bounded by Eq. (2.1). The remaining case is that every node has size in the range $(0, \frac{2}{3}k] \cup (k, \frac{4}{3}k) \cup (2k, n]$, and every extremely big node has exactly two children.

Now, for any node v of size $>2k$, if none of its children has size $>2k$, then v is called *extremely huge*. Clearly, an extremely huge node must have at least two children, each of which has size in the range $(0, \frac{2}{3}k] \cup (k, \frac{4}{3}k)$. Note that if there are two children of size $\leq\frac{2}{3}k$, we can merge the two corresponding branches into one two-branch subtree (via a copy of v, say v', which becomes an extremely big node, and a dummy edge (v, v') of cost 0). It follows that we may assume without loss of generality that there is at most one child of size $\leq\frac{2}{3}k$. Consequently, there are at least two children of size in the open interval $(k, \frac{4}{3}k)$. The process locates an extremely huge node r, and cuts off exactly two of its branches of size $>k$, together with a copy of r, to form a two-branch subtree T_r. The number of destination nodes in this subtree T_r, $|D_r|$, is thus in the range $(2k, \frac{8}{3}k)$. Two cases are distinguished depending on the actual size of D_r: If $|D_r| \leq \frac{5}{2}k$, then these destination nodes are routed by Lemma 2.5 with the routing cost accordingly bounded by Eq. (2.1); if $|D_r| > \frac{5}{2}k$, then the destination nodes are routed by Lemma 2.6 with the routing cost accordingly bounded by Eq. (2.1).

Finally, when no subtree can be cut out of the base Steiner tree, still denoted as T_s^0, we conclude that the size of T_s^0 is in the range $(0, \frac{2}{3}k] \cup (k, \frac{4}{3}k)$. In the former case, this residual tree is taken as a routing tree to route the destination nodes therein, with routing cost $w(T_s^0)$; in the latter case, the residual tree is split into two routing trees to route the destination nodes therein, with the total routing cost $\leq w(T_s^0) + \frac{1}{k}\sum_{d\in D_s} w(s, d)$ [5]. Summing up, the main result is the following:

Theorem 2.1. *kMTR ($k \geq 3$) admits a $(\frac{5}{4}\rho + \frac{3}{2})$-approximation algorithm, where ρ is the currently best performance ratio for approximating the* Steiner minimum tree *problem.*

Proof. Notice that whenever a subtree T_r is cut out of the base Steiner tree T_s^0, we do not increase the weight of the trees, though we might need to duplicate a certain (Steiner or destination) node for connectivity purposes. The total routing cost for T_r, as proven in the technical Lemmas 2.2–2.6, is upper bounded by Eq. (2.1): $\frac{5}{4}w(T_r) + \frac{3}{2} \times \frac{1}{k}\sum_{d\in D_r} w(s, d)$. The total routing cost for the residual Steiner tree is also upper bounded by Eq. (2.1). Therefore, the total routing cost for the output k-tree routing is $R \leq \frac{5}{4}w(T_s^0) + \frac{3}{2} \times \frac{1}{k}\sum_{d\in D} w(s, d) \leq \frac{5}{4}w(T_s^0) + \frac{3}{2}R^*$, where the last inequality follows from Eq. (1.1). Since $w(T_s^0) \leq \rho R^*$, we have $R \leq (\frac{5}{4}\rho + \frac{3}{2})R^*$. □

2.1. Technical lemmas

Lemma 2.2 ([5]). *Given a Steiner tree T_r such that*

- $\frac{2}{3}k \leq |D_r| \leq k$,

the routing cost for T_r is $\leq w(T_r) + \frac{3}{2} \times \frac{1}{k}\sum_{d\in D_r} w(s, d)$.

The main contribution in this work is the routing design and analysis presented in the proof of the following Lemma 2.3. Previous designs can only guarantee upper bounds of $\frac{5}{4}w(T_r) + \frac{8}{5} \times \frac{1}{k}\sum_{d\in D_r} w(s, d)$ [5] and $\frac{5}{4}w(T_r) + \frac{\sqrt{2089}+77}{80} \times \frac{1}{k}\sum_{d\in D_r} w(s, d)$ [4], respectively, on the total routing cost, while our new design ensures an improved upper bound of $\frac{5}{4}w(T_r) + \frac{3}{2} \times \frac{1}{k}\sum_{d\in D_r} w(s, d)$. This eventually leads to Theorem 2.1.

Lemma 2.3. *Given a Steiner tree T_r such that*

- $\frac{4}{3}k \leq |D_r| < \frac{3}{2}k$;
- *root node r has exactly three child nodes v_1, v_2, v_3; and*
- $|D_{v_1}| < \frac{2}{3}k$, $|D_{v_2}| < \frac{2}{3}k$, *and* $|D_{v_1}| + |D_{v_2}| > k$.

It is always possible to partition T_r into two subtrees of size $\leq k$, such that the total routing cost for these subtrees is $\leq\frac{5}{4}w(T_r) + \frac{3}{2} \times \frac{1}{k}\sum_{d\in D_r} w(s, d)$.

Proof. By the conditions in the lemma, $|D_{v_3}| < \frac{1}{2}k$. Without loss of generality, we assume that $|D_{v_1}| \le |D_{v_2}|$. Then, $|D_{v_2}| > \frac{1}{2}k$. For each $i \in \{1, 2, 3\}$, let B_i be the branch rooted at r and containing v_i. We distinguish two cases as follows.

Case 1: [4] $|D_{v_3}| + |D_{v_2}| \le k$. In this case, $|D_{v_3}| + |D_{v_1}| \le k$. Among the nodes in D_r, we find the $\frac{2}{3}k$ closest nodes to s, and form them into a set C. Similarly, among the nodes in D_r, we find the $\frac{2}{3}k$ farthest nodes from s, and form them into a set F. Since $|D_r| \ge \frac{4}{3}k$, $F \cap C = \emptyset$. Moreover, since $|D_{v_i}| < \frac{2}{3}k$ for each $i \in \{1, 2, 3\}$, there are at least two indices $i \in \{1, 2, 3\}$ such that $(D_{v_i}) \cap C \ne \emptyset$. If $(D_{v_3}) \cap C = \emptyset$, then we set $X_1 = B_1$ and construct X_2 by initializing it as the union of B_2 and B_3 and then subtract r. Otherwise, we find an index $i \in \{1, 2\}$ with $(D_{v_i}) \cap C \ne \emptyset$, set $X_1 = B_i$ and construct X_2 by initializing it as the union of B_j and B_3 and then subtract r, where j is the other index in $\{1, 2\} - \{i\}$. In any case, $|D_r \cap X_1| \le k$, $|D_r \cap X_2| \le k$, $(D_r \cap X_1) \cap C \ne \emptyset$, and $(D_r \cap X_2) \cap C \ne \emptyset$. Obviously, one of $D_r \cap X_1$ and $D_r \cap X_2$ contains d', which is the closest destination node to s among the nodes in D_r. We assume that $D_r \cap X_1$ contains d'; the other case is symmetric. Then, $c(X_1) \le w(s, d') \le \frac{3}{2} \times \frac{1}{k} \sum_{d \in C} w(s, d)$. Moreover, since $(D_r \cap X_2) \cap C \ne \emptyset$, $c(X_2) \le w(s, d'')$, where d'' is the farthest destination node from s among the nodes in C. Furthermore, since $C \cap F = \emptyset$, $w(s, d'') \le w(s, d''')$, where d''' is the closest destination node to s among the nodes in F. Thus, $c(X_2) \le w(s, d''') \le \frac{3}{2} \times \frac{1}{k} \sum_{d \in F} w(s, d)$. Therefore, $c(X_1) + c(X_2) \le \frac{3}{2} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$. Consequently, the total routing cost of X_1 and X_2 is at most $w(T_r) + \frac{3}{2} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$, and the lemma is proved.

Case 2: $|D_{v_3}| + |D_{v_2}| > k$. We assume that $|D_{v_2}| = (\frac{1}{2} + p)k$. Note that, since $|D_{v_1}| + |D_{v_2}| > k$, we have $|D_{v_1}| > (\frac{1}{2} - p)k$; likewise, we have $|D_{v_3}| > (\frac{1}{2} - p)k$. These two together imply that $|D_{v_1}| + |D_{v_3}| \in ((1 - 2p)k, (1 - p)k)$. Therefore, in the first routing option, we set $X_1 = B_2$ and construct X_2 by initializing it as the union of B_1 and B_3 and then subtract r. It follows that the connection cost $c(X_1) \le \frac{1}{\frac{1}{2}+p} \times \frac{1}{k} \sum_{d \in D_{v_2}} w(s, d)$, and the connection cost $c(X_2) \le \frac{1}{1-2p} \times \frac{1}{k} \sum_{d \in D_{v_1} \cup D_{v_3}} w(s, d)$. Since $p \in (0, \frac{1}{6})$, the total routing cost of X_1 and X_2 is $w_1 \le w(T_r) + \frac{1}{\frac{1}{2}+p} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$.

Next, among the three branches, we select the one having the minimum weight and duplicate it. Assume without loss of generality that B_1 has the minimum weight. Then, $w(B_1) \le \frac{1}{3} w(T_r)$. We create two routing trees out of this tree: one is the union of one copy of branch B_1 and branch B_2 and the other the union of the other copy of branch B_1 and branch B_3, and then subtract r. The destination nodes routed by the first routing tree include the ones in D_{v_2} and a subset of D_{v_1}, such that their number is exactly $\frac{1}{2}|D_r|$, which is $>(\frac{3}{4} - \frac{1}{2}p)k$; the destination nodes routed by the second routing tree include the ones in D_{v_3} and the remainder of D_{v_1}. It follows that the total routing cost of the second routing option is $w_2 \le \frac{4}{3} w(T_r) + \frac{1}{\frac{3}{4} - \frac{1}{2}p} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$.

Note that $\min\{w_1, w_2\} \le \frac{1}{4} w_1 + \frac{3}{4} w_2 = \frac{5}{4} w(T_r) + \left(\frac{1}{4(\frac{1}{2}+p)} + \frac{3}{4(\frac{3}{4} - \frac{1}{2}p)} \right) \times \sum_{d \in D_r} w(s, d) = \frac{5}{4} w(T_r) + \frac{9+10p}{(2+4p)(3-2p)} \sum_{d \in D_r} w(s, d)$. From $0 < p < \frac{1}{6}$, we conclude that $\min\{w_1, w_2\} < \frac{5}{4} w(T_r) + \frac{3}{2} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$, since $\frac{9+10p}{(2+4p)(3-2p)} - \frac{3}{2} = \frac{p(6p-1)}{(1+2p)(3-2p)} < 0$. Therefore, in this case, choosing the better option between the two proves the lemma. □

Lemma 2.4 ([5]). *Given a Steiner tree T_r such that*

- $\frac{3}{2}k < |D_r| \le 2k$;
- *root node r has exactly three child nodes v_1, v_2, v_3; and*
- $|D_{v_1}| < \frac{2}{3}k$, $|D_{v_2}| < \frac{2}{3}k$, *and* $|D_{v_1}| + |D_{v_2}| > k$.

It is always possible to partition T_r into two or three subtrees of size $\le k$, such that the total routing cost for these subtrees is $\le \frac{5}{4} w(T_r) + \frac{3}{2} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$.

Lemma 2.5 ([5]). *Given a Steiner tree T_r such that*

- $2k < |D_r| \le \frac{5}{2}k$;
- *root node r has exactly two child nodes v_1, v_2, and $k < |D_{v_1}|, |D_{v_2}| < \frac{4}{3}k$;*
- *for $i = 1, 2$, within T_{v_i}, there is a node u_i which has exactly two child nodes x_{i1} and x_{i2};*
- *for $i = 1, 2$, $|D_{x_{i1}}| < \frac{2}{3}k$, $|D_{x_{i2}}| < \frac{2}{3}k$, and $|D_{x_{i1}}| + |D_{x_{i2}}| > k$.*

It is always possible to partition T_r into three subtrees of size $\le k$, such that the total routing cost for these subtrees is $\le \frac{5}{4} w(T_r) + \frac{3}{2} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$.

Lemma 2.6 ([4]). *Given a Steiner tree T_r such that*

- $\frac{5}{2}k < |D_r| < \frac{8}{3}k$;
- *root node r has exactly two child nodes v_1, v_2, and $k < |D_{v_1}|, |D_{v_2}| < \frac{4}{3}k$;*
- *for $i = 1, 2$, within T_{v_i}, there is a node u_i which has exactly two child nodes x_{i1}, x_{i2};*
- *for $i = 1, 2$, $|D_{x_{i1}}| < \frac{2}{3}k$, $|D_{x_{i2}}| < \frac{2}{3}k$, and $|D_{x_{i1}}| + |D_{x_{i2}}| > k$.*

It is always possible to partition T_r into three or four subtrees of size $\le k$, such that the total routing cost for these subtrees is $\le \frac{5}{4} w(T_r) + \frac{3}{2} \times \frac{1}{k} \sum_{d \in D_r} w(s, d)$.

3. Conclusions

We have presented a $(\frac{5}{4}\rho + \frac{3}{2})$-approximation algorithm for kMTR. This performance ratio was targeted in several previous works [16,5,4], but never achieved. Since every technical lemma takes time that is linear in the number of destination nodes in the subtree under consideration, provided that for every destination node its distance to the source node s has been pre-computed, the running time of the complete algorithm is dominated by the running time of the ρ-approximation algorithm for the Steiner minimum tree problem, whose complexity could be prohibitively high [17]. For the kMTR problem, we conjecture that this ratio of $(\frac{5}{4}\rho + \frac{3}{2})$ is the best possible by this line of tree partitioning based approximation algorithms. To design better approximations, certain new techniques other than tree partitioning might need to be introduced.

Acknowledgements

The first author was supported by AICML, ALIDF, and NSERC. The second author was supported by AICML, iCORE, and NSERC. The third author was supported by ALIDF, CFI, and NSERC.

References

[1] K. Altinkemer, B. Gavish, Heuristics with constant error guarantees for the design of tree networks, Management Science 34 (1988) 331–341.
[2] K. Bharath-Kumar, J.M. Jaffe, Routing to multiple destinations in computer networks, IEEE Transaction on Communications 31 (1983) 343–351.
[3] Z. Cai, Improved algorithms for multicast routing and binary fingerprint vector clustering, Master's Thesis, Department of Computing Science, University of Alberta, June 16, 2004.
[4] Z. Cai, Z.-Z. Chen, G.-H. Lin, A 3.4713-approximation algorithm for the capacitated multicast tree routing problem, Theoretical Computer Science (2009) doi:10.1016/j.tcs.2009.05.013.
[5] Z. Cai, Z.-Z. Chen, G.-H. Lin, L. Wang, An improved approximation algorithm for the capacitated multicast tree routing problem, in: Proceedings of the Second Annual International Conference on Combinatorial Optimization and Applications, COCOA 2008, in: LNCS, vol. 5165, 2008, pp. 286–295.
[6] Z. Cai, G.-H. Lin, G. Xue, Improved approximation algorithms for the capacitated multicast routing problem, in: Proceedings of the Eleventh International Computing and Combinatorics Conference, COCOON 2005, in: LNCS, vol. 3593, 2005, pp. 136–145.
[7] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
[8] C. Gröpl, S. Hougardy, T. Nierhoff, H.J. Prömel, Approximation algorithms for the Steiner tree problem in graphs, in: D.-Z. Du, X. Cheng (Eds.), Steiner Trees in Industries, Kluwer Academic Publishers, 2001, pp. 235–279.
[9] J. Gu, X.D. Hu, X. Jia, M.-H. Zhang, Routing algorithm for multicast under multi-tree model in optical networks, Theoretical Computer Science 314 (2004) 293–301.
[10] X.-D. Hu, T.-P. Shuai, X. Jia, M.-H. Zhang, Multicast routing and wavelength assignment in WDM networks with limited drop-offs, in: Proceedings of IEEE INFOCOM 2004, 2004.
[11] C. Huitema, Routing in the Internet, Prentice Hall PTR, 2000.
[12] R. Jothi, B. Raghavachari, Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design, in: Proceedings of the 31st International Colloquium on Automata, Languages and Programming, ICALP 2004, in: LNCS, vol. 3142, 2004, pp. 805–818.
[13] F. Kuo, W. Effelsberg, J.J. Garcia-Luna-Aceves, Multimedia Communications: Protocols and Applications, Prentice Hall, Inc., 1998.
[14] R. Libeskind-Hadas, Efficient collective communication in WDM networks with a power budget, in: Proceedings of the Ninth IEEE Conference on Computer Communications and Networks, ICCCN 2000, 2000, pp. 612–616.
[15] G.-H. Lin, An improved approximation algorithm for multicast k-tree routing, Journal of Combinatorial Optimization 9 (2005) 349–356.
[16] E. Morsy, H. Nagamochi, An improved approximation algorithm for capacitated multicast routings in networks, Theoretical Computer Science 390 (2008) 81–91.
[17] G. Robins, A.Z. Zelikovsky, Improved Steiner tree approximation in graphs, in: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2000, 2000, pp. 770–779.
[18] L.H. Sahasrabuddhe, B. Mukherjee, Multicast routing algorithms and protocols: A tutorial, IEEE Networks 14 (2000) 90–102.
[19] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, IEEE Journal on Selected Areas in Communications 14 (1996) 1228–1234.
[20] X. Zhang, J. Wei, C. Qiao, Constrained multicast routing in WDM networks with sparse light splitting, in: Proceedings of IEEE INFOCOM 2000, 2000, pp. 1781–1790.