Sharing the cost more efficiently: Improved Approximation for Multicommodity Rent-or-Buy

L. BECCHETTI
Università di Roma "La Sapienza"
and
J. KÖNEMANN
University of Waterloo
and
S. LEONARDI
Università di Roma "La Sapienza"
and
M. PÁL
Google Inc.

In the *multicommodity rent-or-buy* (MROB) network design problem we are given a network together with a set of k terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$. The goal is to provision the network so that a given amount of flow can be shipped between s_i and t_i for all $1 \leq i \leq k$ simultaneously. In order to provision the network one can either *rent* capacity on edges at some cost per unit of flow, or *buy* them at some larger fixed cost. Bought edges have no incremental, flow-dependent cost. The overall objective is to minimize the total provisioning cost.

Recently, Gupta et al. (Proceedings, IEEE Symp. Foundations of Computer Science, 2003) presented a 12-approximation for the MROB problem. Their algorithm chooses a subset of the terminal pairs in the graph at random and then buys the edges of an approximate Steiner forest for these pairs. This technique had previously been introduced Gupta et al. (Proceedings, ACM Symp. Theory of Computing, 2003) for the single sink rent-or-buy network design problem.

In this paper we give a 6.828-approximation for the MROB problem by refining the algorithm of Gupta et al. and simplifying their analysis. The improvement in our paper is based on a more careful adaptation and simplified analysis of the primal-dual algorithm for the Steiner forest problem due to Agrawal et al. (SIAM J. Comput., 1995). Our result significantly reduces the gap between the single-sink and multi-sink case.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complex-**

An extended abstract of this article appeared in *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 375–384.

Authors' addresses: L. Becchetti & S. Leonardi, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy. Email: {`becchett, leon`}`@dis.uniroma1.it`.

J. Könemann, Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Email: `jochen@uwaterloo.ca`.

M. Pál, Google, Inc., 1440 Broadway, 21st floor, New York, NY 10018, U.S.A. Email: `martin@palenica.com`.

ity]: Nonnumerical Algorithms and Problems; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Algorithms

Additional Key Words and Phrases: Approximation algorithms, cost-sharing, network design, Steiner forests

1. INTRODUCTION

In the *multi-commodity rent-or-buy problem* (MROB) we are given an undirected graph $G = (V, E)$, terminal pairs $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$, non-negative costs c_e for all edges $e \in E$, and a parameter $M \geq 0$. The goal is to select a set of *bought* edges F_b and a set of of *rented* edges F_r, respectively, such that for all $(s, t) \in R$, we can ship a given amount of flow from s to t using the edges in $F_b \cup F_r$. The cost of a bought edge $e \in F_b$ is $M \cdot c_e$. A rented edge $e \in F_r$ costs $c_e \cdot \lambda(F, e)$ where $\lambda(F, e)$ denotes the total flow traversing edge e. The aim is to find a feasible solution of smallest total cost.
The MROB problem generalizes the *single-commodity rent-or-buy* problem (SROB). Here we are again given an undirected network together with rental and buying costs on all edges $e \in E$ as before. We are also given a set of terminal vertices and a root vertex r. The goal is now to provision the network such that all terminals can send a specified amount of flow to the root vertex r simultaneously. A recent result of Gupta et al. [Gupta et al. 2003] gives a 3.55 approximation algorithm for the problem.

Awerbuch and Azar [Awerbuch and Azar 1997] and Bartal [Bartal 1998] were the first to give an $O(\log |V| \log \log |V|)$-approximation algorithm for the MROB problem. Later, Kumar, Gupta and Roughgarden [Kumar et al. 2002] gave the first constant approximation algorithm for the problem based on a primal-dual approach. More recently, Gupta et al. [Gupta et al. 2003] extended the techniques used by Gupta et al. [Gupta et al. 2003] and presented a 12-approximation for the MROB problem. Their work also uses the *cost-sharing* concept from game-theory (see, e.g., [Feigenbaum et al. 2001; Jain and Vazirani 2001; Pál and Tardos 2003]) in the analysis of the algorithm. Very recently, in [Fleischer et al. 2006], Fleischer et al. improved upon the results presented in this paper and obtained an elegant 5-approximation algorithm for the MRoB problem.

The minimum-cost Steiner tree and forest problems are closely related to both the MROB and SROB problems. In the more general Steiner forest problem, we are given an undirected graph $G = (V, E)$, non-negative costs c_e for all edges $e \in E$, and a set of terminal pairs $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$. The goal is to find a forest F of minimum total cost such for all $1 \leq i \leq k$, there is a tree $T \in F$ that contains both, s_i and t_i. It is well-known that the minimum-cost Steiner forest problem is NP-hard[Garey and Johnson 1979] and Max-SNP hard. On the positive side, Agrawal, Klein and Ravi [Agrawal et al. 1995] and later Goemans and Williamson [Goemans and Williamson 1995] gave a primal-dual 2-approximation for the problem.

The MROB algorithm from [Gupta et al. 2003] crucially relies on the primal-dual algorithm for Steiner forest of Agrawal et al. [Agrawal et al. 1995]. The algorithm in [Gupta et al. 2003] first picks a random subset of all terminal pairs $R_0 \subseteq R$ and then uses a modified primal-dual Steiner forest algorithm to compute a feasible Steiner forest F_0 for R_0. The algorithm buys all edges from F_0. Terminal pairs in $R \setminus R_0$ that are not connected in F_0 rent extra capacity in the cheapest possible way to establish connections.

The central feature of the modified primal-dual Steiner forest algorithm used in [Gupta et al. 2003] is *β-strictness*: The algorithm defines *cost-shares* χ_{st} for all terminal pairs (s, t) in R. Let the Steiner forest computed by the algorithm on input $R \setminus \{(s, t)\}$ be denoted by F_0 and let $G|F_0$ be the graph obtained from G by contracting F_0. The algorithm then guarantees that the cheapest way of connecting s to t in $G|F_0$ costs at most $\beta \cdot \chi_{st}$. Moreover, the sum over all cost-shares of terminal pairs is at most the cost of a minimum-cost Steiner forest for R.

The *prize* for a β-strict Steiner forest algorithm is a worse performance guarantee. Gupta et. al show that their Steiner forest modification returns a 6-approximate and 6-strict Steiner forest and this leads to a 12-approximate MROB-algorithm. In general, they show that any α-approximate and β-strict algorithm leads to an $(\alpha + \beta)$-approximation for the MROB problem.

We remark that cost-shares and the concept of β-strictness find application in the area of stochastic optimization as well. In [Gupta et al. 2004], Gupta et al. presented a general framework for *2-stage stochastic optimization with recourse*. The authors showed how to turn an α-approximate and β-strict approximation algorithm for a deterministic optimization problem into an $\alpha + \beta$-approximation algorithm for the corresponding stochastic version of the problem.

1.1 Our Contribution.

Our algorithm uses the cost-sharing framework proposed by Gupta et al. We prove the following main result:

THEOREM 1.1. *There is a polynomial-time $(2 + 2\gamma)$-approximate and $(2 + 1/\gamma)$-strict algorithm for the minimum-cost Steiner forest problem for any $\gamma \geq 1/2$.*

In [Gupta et al. 2003], Gupta et al. show the following main theorem:

THEOREM 1.2. *Suppose there is an α-approximate and β-strict algorithm for the Steiner forest problem. Then there exists an $(\alpha + \beta)$-approximation algorithm for the multicommodity rent-or-buy problem.*

Choosing $\gamma = \sqrt{1/2}$ in Theorem 1.1 together with Theorem 1.2 implies the following corollary:

COROLLARY 1.3. *There is a $(4 + 2\sqrt{2})$-approximate algorithm for the multicommodity rent-or-buy problem.*

The heart of our work is a new β-strict algorithm for the Steiner forest problem. Our Steiner forest algorithm has two main phases: The first phase runs the standard primal-dual Steiner forest algorithm from [Agrawal et al. 1995] and computes an approximate Steiner forest F' for a given set of terminal pairs R.

The goal of the second phase of the algorithm is to augment forest F' with additional edges in order to improve its connectivity. In order to do this, we first identify the terminal vertices in each tree T in F'. This yields a new graph G_+ that has a *super-vertex* for each tree in F'. We treat these super-vertices as terminals of another Steiner tree instance and run an adaptation of the primal-dual algorithm for the prize-collecting Steiner tree problem (see, e.g., [Goemans and Williamson 1995]) in order to obtain a forest F''. The final forest is obtained by adding the edges of F'' to the forest F'.

In a nutshell, the main insight leading to the improved performance guarantee of our algorithm in comparison to the result in [Gupta et al. 2003] is a more careful bound of the cost of the edges that are added to F' in the second phase. The two-phased combination of standard primal-dual Steiner forest algorithms is particularly helpful in our analysis as it produces a very structured dual much like existing primal-dual Steiner tree and forest algorithms do.

2. THE MINIMUM-COST STEINER FOREST PROBLEM

The first primal-dual algorithm for Steiner trees and forests were obtained by Agrawal, Klein, and Ravi [Agrawal et al. 1995]. Their algorithm was later extended to a larger class of network design problems by Goemans and Williamson [Goemans and Williamson 1995]. In this paper we deliberately choose the viewpoint taken in [Agrawal et al. 1995].

In the following we use `AKR` to refer to the algorithm in paper [Agrawal et al. 1995]. For notational convenience, we let $\mathcal{R}$ be the set of all terminals, i.e.

$$\mathcal{R} = \bigcup_{(s,t)\in R} \{s,t\}.$$

Similarly, for a set $U \subseteq V$, we use $\mathcal{R}[U]$ to denote the set of terminals that are contained in U.

The primal-dual algorithm `AKR` constructs both a feasible primal and a feasible dual solution for a linear programming formulation of the Steiner forest problem and its dual, respectively. A standard integer programming formulation for the Steiner forest problem has a binary variable x_e for all edges $e \in E$. Variable x_e has value 1 if edge e is part of the resulting forest. We let $\mathcal{U}$ contain exactly those subsets U of V that *separate* at least one terminal pair in R. In other words, $U \in \mathcal{U}$ iff there is $(s,t) \in R$ with $|\{s,t\} \cap U| = 1$. The sets in $\mathcal{U}$ are also referred to as *Steiner cuts*.

For a subset U of the vertices we also let $\delta(U)$ denote the set of those edges that have exactly one endpoint in U. We then obtain the following integer linear programming formulation for the Steiner forest problem:

$$\begin{aligned} \min \quad & \sum_{e\in E} c_e \cdot x_e && \text{(IP)} \\ \text{s.t.} \quad & \sum_{e\in\delta(U)} x_e \geq 1 \quad \forall U \in \mathcal{U} \\ & x \text{ integer} \end{aligned}$$

The linear programming dual of the LP-relaxation (LP) of (IP) has a variable y_U for all vertex sets $U \in \mathcal{U}$. There is a constraint for each edge $e \in E$ that limits the total dual assigned to sets $U \in \mathcal{U}$ that contain exactly one endpoint of e to be at most c_e.

$$\max \quad \sum_{U \in \mathcal{U}} y_U \tag{D}$$

$$\text{s.t.} \quad \sum_{U \in \mathcal{U}: e \in \delta(U)} y_U \leq c_e \quad \forall e \in E \tag{1}$$

$$y \geq 0$$

Algorithm AKR constructs a primal solution for (LP) and a dual solution for (D). The algorithm has two goals:

Compute a feasible solution for the given Steiner forest instance. The algorithm reduces the degree of infeasibility as it progresses.

Create a feasible dual solution of largest possible total value. AKR raises dual variables of certain subsets of vertices at all times. The final dual solution is maximal in the sense that no single set can be raised without violating a constraint of type (1).

We think of an execution of algorithm AKR as a process over time and let x^τ and y^τ be the primal incidence vector and feasible dual solution at time τ. We also use F^τ to denote the forest corresponding to x^τ. Initially, we let $x^0_e = 0$ for all $e \in E$ and $y^0_U = 0$ for all $U \in \mathcal{U}$. In the following we say that an edge $e \in E$ is *tight* if the corresponding constraint (1) holds with equality.

Assume that the forest F^τ at time τ is infeasible. A terminal vertex $v \in \mathcal{R}$ is *active* at time τ if v and its *mate* $\bar{v}$, i.e., $(v, \bar{v}) \in R$, are in different trees in the forest F^τ; v is *inactive* otherwise. We use $\bar{F}^\tau$ to denote the subgraph of G that is induced by the tight edges for dual y^τ. In order to avoid confusion between connected components in F^τ and those in $\bar{F}^\tau$, we reserve the term *moat* to refer to a connected component in $\bar{F}^\tau$. A moat U of $\bar{F}^\tau$ is active at time τ if U contains an active terminal; U is inactive otherwise. Let A^τ be the set of all active moats in $\bar{F}^\tau$ at time τ. AKR raises the dual variables for all sets in A^τ uniformly at all times $\tau \geq 0$.

Suppose now that two active moats U and U' *collide* at time τ in the execution of AKR. In other words, $\bar{F}^\tau$ contains a path P of tight edges between two active terminals $u \in U$ and $u' \in U'$. We then add path P to F^τ and continue. U and U' are part of the same moat of $\bar{F}^{\tau'}$ for $\tau' > \tau$.

Subsequently, we use $U^\tau(v)$ to refer to the moat in $\bar{F}^\tau$ that contains vertex $v \in V$ at time τ. Similarly, we let $U^\tau(C)$ denote the moat in $\bar{F}^\tau$ that contains the connected component $C \in F^\tau$ at time τ.

We define the age $\texttt{age}(s,t)$ of a terminal pair (s,t) as the first time in AKR where s and t are contained in the same moat, i.e. s and t are in the same connected component of $\bar{F}^\tau$ iff $\tau \geq \texttt{age}(s,t)$. We define the age of a terminal $v \in \mathcal{R}$ as the maximum age of any terminal pair that contains v, i.e.,

$$\texttt{age}(v) = \max_{(s,t) \in R, v \in \{s,t\}} \texttt{age}(s,t). \tag{2}$$

Let F be the final forest computed by AKR(R). For a connected component C of F, we then let $\texttt{age}(C) = \max_{v \in \mathcal{R}[C]} \texttt{age}(v)$ be the age of C. The following is the main theorem of [Agrawal et al. 1995]:

THEOREM 2.1. *Suppose that algorithm* AKR *outputs a forest F with connected components $\mathcal{C}$ and a feasible dual solution $\{y_U\}_{U\in\mathcal{U}}$. We then have*

$$c(F) \le 2\sum_{U\in\mathcal{U}} y_U - 2\sum_{C\in\mathcal{C}} \texttt{age}(C) \le \left(2-\frac{1}{k}\right)\cdot \texttt{opt}_R,$$

where $\texttt{opt}_R$ is the minimum cost of a Steiner forest for the given input instance with terminal set R.

In the discussion to follow, we may sometimes use primal-dual terminology to describe the properties of AKR. In particular, we may sometimes say that an active moat U of $\bar{F}^\tau$ or a terminal contained in U *loads* an edge $e \in E$. This means that AKR raises the dual y_U for moat U and $e \in \delta(U)$. Similarly, U loads a path P if U loads an edge on P. We also say that moat U *inflicts* load on edge e at time τ for some $\tau \ge 0$ if moat $U \in A^\tau$ loads e. Finally, we say that terminals $u, v \in \mathcal{R}$ *meet* at time $\tau \ge 0$ in a run of AKR if u and v are in the same moat $U \in A^{\tau'}$ iff $\tau' \ge \tau$.

3. A STRICT ALGORITHM FOR MINIMUM-COST STEINER FOREST

This section is split into three major parts. First we show how to compute the cost shares for each terminal pair $(s,t) \in R$. Subsequently we give our $(2+2\gamma)$-approximate and $(2+1/\gamma)$-strict algorithm for Steiner forests. The section ends with the strictness-analysis of the algorithm.

3.1 Computing cost-shares

We start by giving a precise definition of the strictness notion. For a forest F in G, let $G|F$ denote the graph resulting from contracting all trees of F. For vertices $u, v \in V$, we also let $c_G(u,v)$ denote the minimum-cost of any u,v-path in G. In [Gupta et al. 2003], Gupta et al. define the notion of *β-strict* algorithms for the minimum-cost Steiner forest problem.

Definition 3.1. An algorithm $\mathcal{A}$ for the Steiner forest problem is β-strict if it returns values χ_i for all $(s_i,t_i) \in R$ such that

(1) $\sum_{(s_i,t_i)\in R} \chi_i \le \texttt{opt}_R$, and
(2) $c_{G|F_i}(s_i,t_i) \le \beta\cdot\chi_i$ for all $(s_i,t_i)\in R$ where F_i is a Steiner forest for terminal pairs $R\setminus\{(s_i,t_i)\}$ returned by $\mathcal{A}$.

The algorithm to compute the cost shares χ_i for all terminal pairs $(s_i,t_i) \in R$ differs slightly from the one presented in [Gupta et al. 2003]. We run AKR on input graph G with terminal pairs R. For convenience, we use $\texttt{age}(i)$ as a short for $\texttt{age}(s_i,t_i)$.

For any time τ during the execution of the algorithm and for any active moat U, we arbitrarily designate a terminal $r_U \in \mathcal{R}[U]$ of maximum age as the *beneficiary* of U. Then define an indicator variable δ^i_τ for all terminal pairs (s_i,t_i) and for all times $\tau \ge 0$:

$$\delta^i_\tau = \begin{cases} 2 & : \text{ Both, } s_i \text{ and } t_i \text{ are beneficiaries at time } \tau < \texttt{age}(i) \\ 1 & : \text{ Exactly one of } s_i \text{ and } t_i \text{ is a beneficiary at time } \tau < \texttt{age}(i) \\ 0 & : \text{ otherwise.} \end{cases}$$

G	Graph for instance
c	Costs on edges
R	Set of terminal pairs
R_{st}	Set of all terminals except pair (s,t)
F	Steiner forest for instance defined by G, c and R
F_{st}	Steiner forest for instance defined by G, c and R_{st}
G_+	Graph obtained from G by identifying terminals in the same connected component of F_{st}
F_+	Forest in G_+ whose connectivity increases with γ
F_{st+}	Forest obtained from F_{st} by adding the edges of F_+
$\mathcal{C}_\alpha$	Connected components of F_α for $\alpha \in \{\emptyset, st, +, st+\}$
$U^\tau_\alpha(v)$	Moat containing terminal v at time τ in AKR run producing F_α for $\alpha \in \{\emptyset, st, +, st+\}$
$\mathtt{age}_\alpha(s', t')$	Age of terminal pair (s', t') at time τ in AKR run producing F_α for $\alpha \in \{\emptyset, st, +, st+\}$

Table I. Summary of notation used in the algorithm description.

The cost-share of terminal pair (s_i, t_i) is defined as

$$\chi_i = \int_0^{\tau^*} \delta^i_\tau \, d\tau \tag{3}$$

where τ^* is the time at which $\mathtt{AKR}(R)$ finishes. Notice that our definition implies that the total cost-share over all terminal pairs is equal to the objective function value of the computed dual solution.

3.2 Adding strictness: A modified Steiner forest algorithm

Fix a terminal pair $(s,t) \in R$ and let $R_{st} = R \setminus \{(s,t)\}$. Analogous to the definition of $\mathcal{R}$, we define $\mathcal{R}_{st}$ as $\bigcup_{(s',t') \in R_{st}} \{s', t'\}$. The new algorithm $\mathtt{AKR}_2$ first uses AKR to compute a feasible Steiner forest F_{st} for terminal set R_{st}. The second phase of the algorithm adds more paths to connect components of F_{st} that are *close* to each other. Selecting paths carefully in this second phase yields a Steiner forest F_{st+} whose cost is only a constant factor worse than that of F_{st} and that satisfies the necessary strictness properties. The notation used in the description of the algorithm is summarized in Table I.

In the following description of the algorithm $\mathtt{AKR}_2$ we use F^τ_α and $\bar{F}^\tau_\alpha$ in place of the corresponding F^τ and $\bar{F}^\tau$ from Section 2 when the final forest produced by the algorithm in discussion is F_α for $\alpha \in \{\emptyset, st, +, st+\}$. We also use y^α to refer to the dual solution computed at the same time and let $\mathtt{age}_\alpha(s,t)$ be the age of terminal pair $(s', t') \in R$ in the corresponding run of AKR. Finally, $\mathcal{C}_\alpha$ will be used to denote the set of connected components of forest F_α for all α.

We now describe the algorithm $\mathtt{AKR}_2$ in greater detail. The algorithm works on input R_{st} and has two phases:

Aerobic Phase. In this phase we first compute a feasible Steiner forest F_{st} for the set of terminal pairs in R_{st} using AKR.

We now use F_{st} to create a new graph $G_+ = (V_+, E_+)$ from the original graph G: For each connected component C of F_{st}, we identify the terminals in $\mathcal{R}_{st}[C]$. In other words, we replace the set $\mathcal{R}_{st}[C]$ by a new vertex v_C. Each edge $(v,u) \in \delta(\mathcal{R}_{st}[C])$ with $v \in \mathcal{R}_{st}[C]$ and $u \notin \mathcal{R}_{st}[C]$ is substituted by a new edge (v_C, u)

with cost c_{vu}. Observe that this process may introduce loops and parallel edges. Delete all loops and for each pair of neighboring vertices $u, v \in V_+$ keep an edge of minimum cost.

Anaerobic Phase. Define the *budget* $\mathtt{b}_v$ of a vertex $v \in V_+$ as

$$\mathtt{b}_v = \begin{cases} (2\gamma + 1) \cdot \mathtt{age}_{st}(C) & : \quad v = v_C \text{ for some } C \in \mathcal{C}_{st} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{4}$$

where γ is a parameter of value at least 1/2 that will be set later.

The goal of this phase is to compute a forest F_+ in G_+. The edges of F_+ are later added to the forest F_{st} in order to strengthen it. The algorithm we use to compute F_+ is similar to the *prize-collecting* Steiner tree algorithm introduced in [Goemans and Williamson 1995]. As with algorithm AKR presented in Section 2 the prize-collecting version described here starts with empty forests F_+^0 and $\bar{F}_+^0$. The initial budget of each vertex $v \in V_+$ is set to $\mathtt{b}_v$.

The crucial difference between this prize-collecting version of AKR and *vanilla* AKR as defined in Section 2 is the definition of the set A_+^τ of active moats at time $\tau \geq 0$. Initially A_+^0 is the set of all vertices $v \in V_+$ with positive budget $\mathtt{b}_C$. The algorithm then grows all moats in A_+^0 uniformly and decreases their budgets at the rate of growth. For $\tau \geq 0$ the set of active moats A_+^τ is the set of connected components of $\bar{F}_+^\tau$ with positive remaining budget.

As before, if two active moats U_1 and U_2 collide at some time $\tau \geq 0$ during the execution of the algorithm, we add a tight path in between them to the current forest F_+^τ. The budget of the resulting moat U in $\bar{F}_+^\tau$ is the sum of the budgets of moats U_1 and U_2.

The final forest F_{st+} is now obtained from F_{st} by adding the edges of F_+. Notice that, by choosing a larger value for γ in (4), we obtain a higher initial budget for all vertices in V_+ that correspond to connected components of F_{st}. As a consequence we therefore obtain a higher degree of connectivity in the forest F_+ and hence also in the forest F_{st+}. Choosing a value of at least 1/2 for γ will later enable us to lower-bound the time of growth of a connected component in the anaerobic phase of $\mathtt{AKR}_2$.

LEMMA 3.2. *The cost of the forest F_{st+} computed by $\mathtt{AKR}_2$ on terminal set R_{st} is at most*

$$(2 + 2\gamma) \cdot \mathtt{opt}_{R_{st}}$$

where $\mathtt{opt}_{R_{st}}$ is the cost of a minimum-cost feasible Steiner forest for terminal set R_{st}.

PROOF. Recall that whenever $\mathtt{AKR}(R_{st})$ grows a moat $U \in \mathcal{U}$ there is a terminal $r_U \in U$ of maximum age that is the beneficiary of this growth. For a connected component C of F_{st}, we then let

$$\mathcal{U}_C = \{U \in \mathcal{U} \, : \, r_U \in \mathcal{R}[C]\}$$

be the set of moats whose beneficiary is a terminal in C.

Observe that the age of a component $C \in \mathcal{C}_{st}$ is defined as the maximum age $\mathtt{age}_{st}(s', t')$ of any terminal pair (s', t') whose vertices are spanned by C (see (2)).

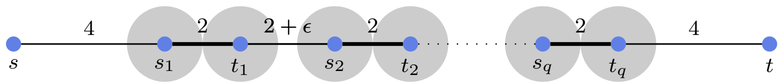

Fig. 1. The figure shows an example Steiner forest instance with $q+1$ terminal pairs. The label next to each of the edges is the cost of the edge. The thick edges in the figure depict the Steiner forest F_{st} computed by AKR for the instance with terminal pairs $R_{st} = \{(s_1, t_1), \ldots, (s_q, t_q)\}$. The grey disks in the figure represent the dual solution computed by $\texttt{AKR}(R_{st})$.

This implies that $\texttt{AKR}(R_{st})$ grows at least two active moats in $\mathcal{U}_C$ at all times prior to time $\mathsf{age}_{st}(C)$ and hence

$$2\gamma \cdot \mathsf{age}_{st}(C) \le \gamma \cdot \sum_{U \in \mathcal{U}_C} y_U^{st}$$

for all $C \in \mathcal{C}_{st}$. The argument used in the proof of Theorem 2.1 shows

$$c(F_+) \le 2 \cdot \sum_{U \subseteq V_+} y_U^+ = 2 \cdot \sum_{C \in \mathcal{C}_{st}} (2\gamma + 1) \cdot \mathsf{age}_{st}(C) \le$$
$$2\gamma \cdot \sum_{C \in \mathcal{C}_{st}} \sum_{U \in \mathcal{U}_C} y_U^{st} + 2 \cdot \sum_{C \in \mathcal{C}_{st}} \mathsf{age}_{st}(C).$$

Now observe that the sets $\{\mathcal{U}_C\}_{C \in \mathcal{C}_{st}}$ are pairwise disjoint and hence

$$c(F_+) \le 2\gamma \cdot \sum_{U \in \mathcal{U}} y_U^{st} + 2 \cdot \sum_{C \in \mathcal{C}_{st}} \mathsf{age}_{st}(C). \tag{5}$$

From Theorem 2.1 we know that

$$c(F_{st}) \le 2 \cdot \left(\sum_{U \in \mathcal{U}} y_U^{st} - \sum_{C \in \mathcal{C}_{st}} \mathsf{age}_{st}(C) \right) \tag{6}$$

and adding inequalities (5) and (6) gives

$$c(F_{st+}) = c(F_{st}) + c(F_+) \le (2 + 2\gamma) \cdot \sum_{U \in \mathcal{U}} y_U^{st}.$$

The lemma follows from weak duality and from the fact that y^{st} is feasible dual solution for (D) with respect to the Steiner forest instance induced by the terminal pairs in R_{st}. □

Much of the discussion in the remainder of this paper will try to formalize a relationship between the execution of AKR on terminal set R and that of $\texttt{AKR}_2$ on R_{st}. In our discussion of $\texttt{AKR}_2$ we will sometimes say that the algorithm grows a connected component $C \in F_{st}$ when we really mean that it grows the super-vertex v_C.

4. ANALYZING THE STRICTNESS OF ALGORITHM $\texttt{AKR}_2$

Before analyzing the strictness of Algorithm $\texttt{AKR}_2$, let us develop some intuition for this notion using some concrete examples.

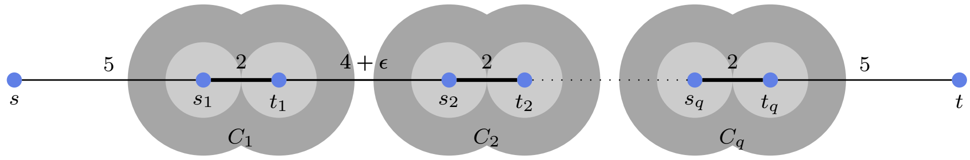

Fig. 2. This figure shows yet another instance of the Steiner forest problem. The thick edges in the figure show the feasible Steiner forest F_{st+} for terminal pairs $R_{st} = \{(s_1,t_1),\ldots,(s_q,t_q)\}$ output by $\mathtt{AKR}_2(R_{st})$. The light-grey, small disks show the dual computed by $\mathtt{AKR}(R_{st})$. The dark, big disks represent the dual computed by the anaerobic phase of $\mathtt{AKR}_2$.

4.1 Intuition

The first example in Figure 1 demonstrates that AKR together with the cost-shares defined in Section 3.1 is not $O(1)$-strict. The instance given in this example has terminal pairs

$$R = \{(s,t),(s_1,t_1),\ldots,(s_q,t_q)\}.$$

Edge-costs are depicted as labels on the edges. Recall that $R_{st} = R \setminus \{(s,t)\}$. The thick edges in the figure show the output forest F_{st} computed by $\mathtt{AKR}(R_{st})$ and the disks in the figure visualize the corresponding feasible dual computed by the algorithm.

Clearly, the forest output by $\mathtt{AKR}(R)$ is the whole path connecting s and t. The age $\mathtt{age}(i)$ of terminal pair (s_i,t_i) is 1 for all $1 \le i \le q$. The age of terminal pair (s,t) is

$$3 + \frac{q-1}{2}\cdot\epsilon \approx 3.$$

The cost-share χ_{st} of terminal pair (s,t) is therefore equal to $2\cdot\mathtt{age}(0) \approx 6$ which, in Figure 1, corresponds to the total cost of the segments on the s,t-path that are not covered by any of the grey disks.

Notice that the distance between s and t in the graph $G|F_{st}$ obtained from the original graph by contracting the edges of the forest F_{st} is $8+(q-1)(2+\epsilon)$. The strictness of AKR with the cost-shares from Section 3.1 is therefore at least

$$\frac{8+(q-1)(2+\epsilon)}{3+\frac{q-1}{2}\cdot\epsilon} \approx \frac{8}{3}+\frac{2}{3}\cdot(q-1)$$

which is an unbounded function of q.

The main insight taken from the example in Figure 1 is that large parts of the minimum-cost path connecting s and t in graph $G|F_{st}$ may be covered (or *hidden*) by grey disks. Terminal pair (s,t) does not obtain cost-share for these hidden segments, the shortest s,t-path in $G|F_{st}$ may, however, pass through these disks. Algorithm $\mathtt{AKR}_2$ deals with exactly this problem by growing the connected components of F_{st} further in the anaerobic phase. Indeed, it can be seen that the forest F_{st+} output by $\mathtt{AKR}_2$ for $\gamma \ge 1/2$ is the unique s_1,t_q-path in the instance shown in Figure 1. The cost of connecting s and t in the graph obtained from contracting the edges of this forest is 8.

Figure 2 shows yet another example on the same set of terminal pairs. Running

$\mathtt{AKR}$ on the instance with all terminal pairs yields $\mathtt{age}(i) = 1$ for $1 \le i \le q$ and

$$\mathtt{age}(0) = 4 + \frac{2+\epsilon}{2} \cdot (q-1) \approx 4 + (q-1).$$

The cost-share χ_{st} of (s,t) is thus roughly $8 + 2(q-1)$.

Let C_i be the connected component containing s_i and t_i in F_{st}. The dark grey disks in the Figure represent the duals computed during the anaerobic phase of $\mathtt{AKR}_2(R_{st})$ with $\gamma = 1/2$. The forest F_{st+} computed during this run is given by the thick edges in Figure 2. The cost of connecting s to t in $G|F_{st+}$ is

$$10 + (4+\epsilon) \cdot (q-1) \approx 10 + 4(q-1)$$

and this is less than $2\chi_{st}$.

As in the first example, the cost-share of (s,t) corresponds to the cost of those segments of the s,t-path in Figure 2 that are not covered by the light, small disks. Notice that the parameter γ in (7) controls the width of the dark grey ring centered at terminal $v \in \{s_1, t_1, s_2, t_2, \ldots, s_q, t_q\}$. In particular, choosing $\gamma = 1/2$ in the example above yields a budget of

$$\mathtt{b}_{C_i} = (2\gamma + 1)\mathtt{age}(C_i) = 2$$

for connected component C_i for all $1 \le i \le q$. In other words, the dark grey ring centered at s_i (or t_i) has width 1 in the example.

The total width of all grey rings is a lower-bound on the cost-share of terminal pair (s,t). We think of the dark-grey ring enclosing a component C_i as the part of the cost-share of χ_{st} that C_i *reserves* in order to pay for those segments on the s,t-path in $G|F_{st+}$ that are covered by the dark and light-grey rings around component C_i.

The intuition given in this section is clearly oversimplifying. In particular, the total width of all dark-grey rings may not be an accurate lower-bound on the cost-share of (s,t). There are two main reasons for this inaccuracy. First, the cost-share of (s,t) may not be $2 \cdot \mathtt{age}(s,t)$.

Definition 4.1. Let $(s,t), (s',t') \in R$ be two terminal pairs with $\mathtt{age}(s,t) \le \mathtt{age}(s',t')$. We say that terminal $v \in \{s,t\}$ *interferes* with $v' \in \{s',t'\}$ if there is an active moat $U \in A^\tau$ for some $0 \le \tau < \mathtt{age}(s,t)$ during the execution of $\mathtt{AKR}(R)$ that contains both v and v' and the beneficiary of U is v'.

Clearly, if the cost-share of (s,t) is smaller than $2 \cdot \mathtt{age}(s,t)$, then s and t must interfere with some other terminals. Secondly, even if the cost-share of (s,t) is $2 \cdot \mathtt{age}(s,t)$, χ_{st} may be much smaller than the total width of all dark-grey rings.

Definition 4.2. Consider the anaerobic phase of $\mathtt{AKR}_2$. A component $C \in \mathcal{C}_{st}$ *captures* a vertex $v \in V_+$ if, for some time $\tau \ge 0$, there is an active moat $U \in A_+^\tau$ that contains both the super-vertex v_C for C and v.

We also say that a connected component C of F_{st+} captures v if there is a connected component C' of F_{st} that captures v and $C' \subseteq C$.

Consider Figure 2. If C_i captures s then s is covered by the dark-grey moats cantered at s_i and t_i. Clearly, in this case, the total width of all dark-grey rings is not a good lower-bound on χ_{st}. We will deal with the above two problems in the next section.

4.2 Outline of analysis

Now consider a general instance of the Steiner forest problem with terminal set R and let (s,t) be an arbitrary pair in R. Recall that $R_{st} = R \setminus \{(s,t)\}$ and let F denote the forest computed by AKR on input R. As before we let F_{st+} be the forest computed by $\mathtt{AKR}_2$ on input R_{st}. As in Section 1 we use $G|F_{st+}$ to denote the graph obtained from G by contracting the connected components of forest F_{st+}. In order to prove that $\mathtt{AKR}_2$ is β-strict we need to show that

$$c_{G|F_{st+}}(s,t) \leq \beta \cdot \chi_{st}. \tag{7}$$

Let P_{st} be the unique s,t-path in the forest F. Notice that the path P_{st} may enter and leave a connected component of F_{st+} multiple times. We can then remove all loops and obtain a path P in $G|F_{st+}$ that enters and leaves each component of F_{st+} at most once.

The rough outline is as follows: The cost of P in $G|F_{st+}$ is at least $c_{G|F_{st+}}(s,t)$. We will show that

$$c_{G|F_{st+}}(P) \leq \beta \cdot \chi_{st}$$

where $c_{G|F_{st+}}(P)$ is the cost of path P in the graph $G|F_{st+}$ and this implies (7) since $c_{G|F_{st+}}(s,t) \leq c_{G|F_{st+}}(P)$.

In the following, we assume that $C_1, \ldots, C_p$ are the components of F_{st+} that contain vertices of P in order of non-decreasing distance from s. Since P is loopless in $G|F_{st+}$ it follows that each connected component of F_{st+} occurs at most once in this list. We also assume that s and t are not part of $\bigcup_{i=1}^{p} C_i$.

The next section presents a lower-bound of the cost-share χ_{st} of the cost-share of terminal pair (s,t). In particular we will relate this cost-share to the length of path P in $G|F_{st+}$.

4.3 A lower-bound of χ_{st}

In the following we let

$$\mathcal{U}_{st} = \{U \in \mathcal{U} \,:\, \{s,t\} \cap U = \emptyset\}$$

be the set of Steiner cuts in G that do not contain s or t. Recall that y denotes the dual solution computed by $\mathtt{AKR}(R)$. For an edge $e \in E$, we define the *residual cost* $\bar{c}_e$ as

$$\bar{c}_e = c_e - \sum_{U \in \mathcal{U}_{st}, e \in \delta(U)} y_U. \tag{8}$$

Using the examples in Section 4.1, the residual cost of an edge e on path P corresponds to the cost of the segment of e that is not hidden by light-grey disks. In other words, $\bar{c}_e$ is the potential cost-share that s and t gain from edge e.

For a connected component C_i of F_{st+} on path P, define P_i^s and P_i^t as the s, C_i-segment and the C_i, t-segment of P, respectively. Define an indicator variable $l_{i,u}^{\tau}$ for all times $\tau \geq 0$, for all $1 \leq i \leq p$ and for all $u \in \{s,t\}$, and let its value be 1 if there is an active moat $U \in \mathcal{U}_{st}$ at time τ in AKR that loads P_i^u, and whose beneficiary r_U is in C_i. Let $l_{i,u}^{\tau} = 0$ otherwise. Define $h_{i,s}$ and $h_{i,t}$ to be the cost of the two hidden segments of P inside C_i, i.e.

$$h_{i,u} = \int_0^{\tau^*} l^\tau_{i,u} d\tau \tag{9}$$

for $u \in \{s,t\}$ and let $h_i = \max\{h_{i,s}, h_{i,t}\}$. In Figure 2, $h_{i,s}$ and $h_{i,t}$ correspond to the radii of the light-grey disks centered at s_i and t_i, respectively. We also say that $h_{i,u}$ is the total load of active moats intersecting C_i on path P_i^u. We immediately obtain the following observation:

LEMMA 4.3. *Let $1 \leq i \leq p$ and $u \in \{s,t\}$ and assume that u meets the first terminal from C_i at time τ in $\texttt{AKR}(R)$. Then we must have $h_{i,u} \leq \tau$. In particular this means that $h_i \leq \texttt{age}(s,t)$ for all $1 \leq i \leq p$.*

PROOF. Observe that any active moat $U \in \mathcal{U}^{\tau'}$ for $\tau' \geq \tau$ that has a non-empty intersection with C_i^u and also intersects P_i^u must contain u. Therefore, U cannot be in $\mathcal{U}_{st}$ and hence $l^{\tau'}_{i,u} = 0$. $\square$

We can now express the cost of path P in $G|F_{st+}$ as

$$c_{G|F_{st+}}(P) = \bar{c}(P) + \sum_{i=1}^{p}(h_{i,s} + h_{i,t}). \tag{10}$$

The following main technical lemma relates χ_{st} to $c_{G|F_{st+}}(P)$. Its proof will be presented in Section 5.

LEMMA 4.4. *Let $\mathcal{I}$ be the set of indices of components on P that contain terminals that interfere with s or t, i.e.*

$$\mathcal{I} = \{i \in \{1,\ldots,p\} \,:\, \exists v' \in C_i \text{ that interferes with } \{s,t\}\}.$$

For $\gamma \geq 1/2$, we must have

$$\chi_{st} \geq \frac{1}{2} \cdot \texttt{sl} + \gamma \cdot \sum_{1 \leq i \leq p, i \notin \mathcal{I}} (h_{i,s} + h_{i,t}) \tag{11}$$

where the slack *in the residual cost $\bar{c}(P)$ is defined as*

$$\texttt{sl} = \max\left\{0, \bar{c}(P) + \left(\sum_{i\in\mathcal{I}}(h_{i,s}+h_{i,t})\right) - 2\gamma \cdot \sum_{1\leq i\leq p, i\notin\mathcal{I}}(h_{i,s}+h_{i,t})\right\}.$$

4.4 The strictness of $\texttt{AKR}_2$

Intuitively, Inequality (11) in Lemma 4.4 shows that terminal pair (s,t) obtains at least $\gamma \cdot (h_{i,s} + h_{i,t})$ units of cost-share for connected component C_i of forest F_{st+} for all $1 \leq i \leq p$, $i \notin \mathcal{I}$. Figure 3 shows such a connected component. As in Section 4.1, the dark grey rings represent (a lower-bound on) the component's growth in the anaerobic phase of $\texttt{AKR}_2(\widetilde{R})$. The rough idea of the following strictness proof is to use $\gamma \cdot (h_{i,s} + h_{i,t})$ of the cost-share χ_{st} for the stretch of P of length $(2\gamma+1)(h_{i,s}+h_{i,t})$ that is covered by dark and light grey disks in the figure.

LEMMA 4.5. *Algorithm $\texttt{AKR}_2$ is $(2 + 1/\gamma)$-strict for $\gamma \geq 1/2$.*

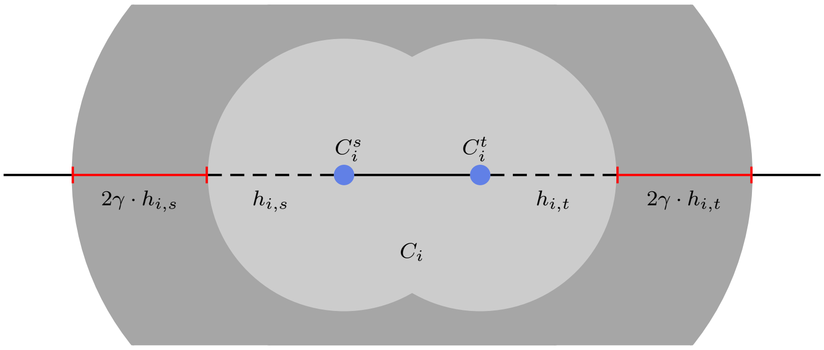

Fig. 3. Connected component C_i on path P together with its budget reservation on P.

PROOF. Using equation (10) and the definition of slack in Lemma 4.4 we obtain

$$\begin{aligned} c_{G|F_{st+}} &= \bar{c}(P) + \sum_{i=1}^{p}(h_{i,s} + h_{i,t}) \\ &\leq (2\gamma + 1)\left(\sum_{1 \leq i \leq p, i \notin \mathcal{I}} (h_{i,s} + h_{i,t})\right) + \mathtt{sl}. \end{aligned} \tag{12}$$

On the other hand Lemma 4.4 yields that the cost-share of terminal pair (s,t) is at least

$$\gamma \cdot \left(\sum_{1 \leq i \leq p, i \notin \mathcal{I}} (h_{i,s} + h_{i,t})\right) + \frac{\mathtt{sl}}{2}. \tag{13}$$

We will now prove the claimed strictness bound of $\mathtt{AKR}_2$ by comparing expressions (12) and (13) in a term-by-term fashion. First, we clearly have $\mathtt{sl} \leq (2+1/\gamma)\cdot(\mathtt{sl}/2)$ for $\gamma \geq 1/2$. Secondly, observe that

$$(2\gamma + 1)h_{i,u} = (2 + 1/\gamma) \cdot \gamma \cdot h_{i,u}$$

for all $1 \leq i \leq p, i \notin \mathcal{I}$ and for $u \in \{s,t\}$ and this finishes the strictness proof. □

Lemma 3.2 and Lemma 4.5 imply the following main theorem 1.1:

THEOREM. *There is a polynomial-time $(2 + 2\gamma)$-approximate and $(2+1/\gamma)$-strict algorithm for the minimum-cost Steiner forest problem for any $\gamma \geq 1/2$.*

5. A GENERAL LOWER-BOUND ON THE COST-SHARE χ_{ST}

In this section we will present the proof of Lemma 4.4. In order to do this, we will need to compare the execution of $\mathtt{AKR}$ on terminal set R with the anaerobic phase of $\mathtt{AKR}_2(R_{st})$.

5.1 Comparing $\mathtt{AKR}(R)$ and $\mathtt{AKR}_2(R_{st})$

The execution of $\mathtt{AKR}_2(R_{st})$ crucially depends on the budgets b_v of vertices in V_+. Thus, it also depends on $\mathtt{AKR}(R_{st})$. The following Lemma compares the ages of terminal pairs in $\mathtt{AKR}(R)$ and $\mathtt{AKR}(R_{st})$. Recall that $U^\tau(v)$ and $U^\tau_{st}(v)$ are the moats containing terminal v at time τ in $\mathtt{AKR}(R)$ and $\mathtt{AKR}(R_{st})$, respectively.

LEMMA 5.1. *For all* $\tau \leq \texttt{age}(s,t)$ *and for all terminals* $v \in \mathcal{R}_{st}$*:* $U^\tau_{st}(v) \subseteq U^\tau(v)$*. Moreover, if* $U^\tau(v) \cap \{s,t\} = \emptyset$*, then* $U^\tau_{st}(v) = U^\tau(v)$*.*

PROOF. We prove the lemma by induction over time τ. At time $\tau = 0$ we have $U^\tau_{st}(v) = U^\tau(v)$ for all $v \in \mathcal{R}_{st}$ and thus the induction hypothesis clearly holds.

Assume the induction hypothesis holds at time $0 \leq \tau < \texttt{age}(s,t)$. We will show that it remains true at time $\tau + \epsilon$ for any small $\epsilon > 0$.

Consider the case $U^\tau(v) \cap \{s,t\} = \emptyset$ and thus $U^\tau_{st}(v) = U^\tau(v)$. That is, $U^\tau_{st}(v)$ is active at time τ iff $U^\tau(v)$ is active at that time. Then $U^{\tau+\epsilon}_{st}(v) = U^{\tau+\epsilon}(v)$ if $U^{\tau+\epsilon}(v) \cap \{s,t\} = \emptyset$ and $U^{\tau+\epsilon}_{st}(v) \subseteq U^{\tau+\epsilon}(v)$ otherwise.

Now assume $U^\tau(v) \cap \{s,t\} \neq \emptyset$ and thus $U^\tau_{st}(v) \subseteq U^\tau(v)$. Clearly, $U^{\tau+\epsilon}(v) \cap \{s,t\} \neq \emptyset$. Since $\tau < \texttt{age}(s,t)$, terminals s and t are active at time τ and thus $U^\tau(v)$ is active at time τ. It follows that $U^{\tau+\epsilon}_{st}(v) \subseteq U^{\tau+\epsilon}(v)$. $\square$

COROLLARY 5.2. *Consider a terminal* $v \in \mathcal{R}_{st}$*. If* v *is active at time* $\tau < \texttt{age}(s,t)$ *in* $\texttt{AKR}(R)$ *then* v *must be active until time at least* τ *in* $\texttt{AKR}(R_{st})$*.*

Recall that V_+ is the set of vertices of graph G_+ used in the anaerobic phase of $\texttt{AKR}_2$. For a subset U of V_+, we let $\bar{U}$ be the set obtained from U by replacing each super vertex $v_C \in U$ by the set of terminals $\mathcal{R}[C]$ that are contained in C.

LEMMA 5.3. *Consider a terminal* $v \in \mathcal{R}_{st}$ *that is active at time* $0 \leq \tau < \texttt{age}(s,t)$ *in* $\texttt{AKR}(R)$*. Let* $C \in \mathcal{C}_{st}$ *be the connected component of* F_{st} *that contains* v *and let* $v_C \in V_+$ *be the corresponding super node in* G_+*. If* $U^\tau(v)$ *does not contain* s *or* t *then* $U^\tau(v) \subseteq \bar{U}^\tau_+(v)$*.*

PROOF. The proof is by induction on the time τ. The claim is true at time $\tau = 0$ as the set of active moats in $\texttt{AKR}(R)$ is the set of terminal vertices in $\mathcal{R}$.

Consider an active terminal pair $(v,u) \in R_{st}$ at time $0 \leq \tau < \texttt{age}(s,t)$ in $\texttt{AKR}(R)$. Lemma 5.1 shows that $U^\tau_{st}(v) = U^\tau(v)$ and hence v is active at time τ in $\texttt{AKR}(R_{st})$ as well. This implies that the age of (u,v) in $\texttt{AKR}(R_{st})$ is bigger than τ.

Let C be the connected component of forest F_{st} that contains u and v and let $v_C \in V_+$ be the corresponding super node in graph G_+. Using the induction hypothesis, we know that $U^\tau(v) \subseteq \bar{U}^\tau_+(v_C)$. The initial budget b_{v_C} of vertex v_C is at least

$$(2\gamma+1)\texttt{age}_{st}(u,v) > (2\gamma+1)\tau$$

as u and v are active at time τ in $\texttt{AKR}(R_{st})$. Therefore the moat $U^\tau_+(v_C)$ must have positive remaining budget at time τ in the anaerobic phase and it is active. The inductive hypothesis must therefore be true for terminal v at time $\tau + \epsilon$ for some small positive ϵ. $\square$

5.2 Observations: Interfering terminals

The observation presented in this section are corollaries of Lemma 5.3. They provide a useful characterization of the possible location of interfering terminals in the forest F_{st+}.

COROLLARY 5.4. *Let* u' *be a terminal that interferes with* $u \in \{s,t\}$ *and assume that* u *and* u' *meet at time* $\tau < \texttt{age}(s,t)$ *in* $\texttt{AKR}(R)$*. Let* $C' \in \mathcal{C}_{st}$ *be the connected component of* F_{st} *containing* u' *and let* $v_{C'} \in V_+$ *be the corresponding vertex of* G_+*.*

The total dual value assigned to moats that contain both u and $v_{C'}$ in the anaerobic phase of $\mathtt{AKR}_2(R_{st})$ must be at least $(2\gamma+1)\cdot \mathtt{age}(s,t) - 2\tau$, i.e.

$$\sum_{U \subseteq V_+, \{u, v_{C'}\} \subseteq U} y_U^+ \geq (2\gamma+1)\cdot \mathtt{age}(s,t) - 2\tau.$$

PROOF. Let $P_{u'}$ be the path that is added in $\mathtt{AKR}(R)$ when u and u' meet at time $\tau < \mathtt{age}(s,t)$. Lemma 5.3 shows that $U^\tau(v) \subseteq \bar{U}_+^\tau(v_{C'})$. Lemma 5.3 also implies that the total load on path $P_{u'}$ until time τ in the anaerobic phase of $\mathtt{AKR}_2(R_{st})$ is at least $c(P_{u'}) - \tau$.

Finally, as in the proof of Lemma 5.3 we observe that the remaining budget of vertex $v_{C'}$ at time τ in the anaerobic phase of $\mathtt{AKR}_2(R_{st})$ is at least

$$(2\gamma+1)\cdot \mathtt{age}(s,t) - \tau$$

which is at least τ by our choice of γ. This implies that $u \in U_+^{2\tau}(v_{C'})$ and the lemma follows. $\square$

The following corollary is implicit in the proof of Corollary 5.4.

COROLLARY 5.5. *Let u_1 and u_2 be terminals that interfere with $u \in \{s,t\}$. They both capture u by time $2\cdot \mathtt{age}(s,t)$ in the anaerobic phase of $\mathtt{AKR}_2(R_{st})$ and there must exist a connected component C in F_{st+} with $\{u_1, u_2\} \subseteq \mathcal{R}_{st}[C]$.*

PROOF. Let $\tau_i < \mathtt{age}(s,t)$ be the time at which u and u_i meet in $\mathtt{AKR}(R)$ for $i \in \{1,2\}$. Let $C_i \in \mathcal{C}_{st}$ be the connected component of F_{st} containing u_i for $i \in \{1,2\}$. The proof of Corollary 5.4 shows that u is contained in $U_+^{2\tau_i}(v_{C_i})$ for all $i \in \{1,2\}$. Our choice of $\gamma \geq 1/2$ in the budget definition (4) ensures that both u_1 and u_2 are active at time $2\cdot \mathtt{age}(s,t) \geq 2\tau_i$ for all $i \in \{1,2\}$ and hence, they must be in the same connected component of F_{st+}. $\square$

Let u' be a terminal that interferes with $u \in \{s,t\}$. We say that u' *is on* P if the path that is added when u and u' meet in $\mathtt{AKR}(R)$ is part of P.

COROLLARY 5.6. *Let u' be a terminal on P that interferes with $u \in \{s,t\}$ and let $C' \in \mathcal{C}_{st}$ be the connected component of F_{st} that contains u'. Then $\{s,t\} \subseteq U_+^{2\cdot \mathtt{age}(s,t)}(v_{C'})$ and there is a connected component C_m for $1 \leq m \leq p$ that contains all interfering terminals.*

PROOF. Corollary 5.2 implies that the initial budget of $v_{C'}$ is at least

$$(2\gamma+1)\mathtt{age}_{st}(u') \geq (2\gamma+1)\mathtt{age}(s,t)$$

and the right-hand side is at least $2\cdot \mathtt{age}(s,t)$ by our choice of γ.

As u' and u interfere, $U^\tau(u')$ intersects P for some $\tau < \mathtt{age}(s,t)$. Let P_s and P_t be the $s, U^\tau(u')$- and $U^\tau(u'), t$-segments of P, respectively. Lemma 5.3 implies that, for all $u \in \{s,t\}$, the total dual load on path P_u at time $\mathtt{age}(s,t)$ in the anaerobic phase of $\mathtt{AKR}_2(R_{st})$ is at least $c(P_u) - \mathtt{age}(s,t)$. Moat $U_+^{\mathtt{age}(s,t)}(v_{C'})$ has at least $\mathtt{age}(s,t)$ remaining budget. Lemma 5.3 shows that $\bar{U}_+^{\mathtt{age}(s,t)}(v_{C'})$ contains $U^{\mathtt{age}(s,t)}(u')$. Therefore we must have $\{s,t\} \subseteq U_+^{2\cdot \mathtt{age}(s,t)}(v_{C'})$.

Notice that this implies that u' captures s and t before time $2\cdot \mathtt{age}(s,t)$ in the anaerobic phase of $\mathtt{AKR}_2(R_{st})$. Corollary 5.5 implies that all interfering terminals for s and t must be contained in the same connected component of F_{st+}. $\square$

In the case of interfering terminals on P we will from now on use C_m to denote the connected component of F_{st+} that contains all interfering terminals.

5.3 Observations: Insufficient residual cost

Suppose that one or more connected components of the forest F_{st} capture s or t during the aerobic phase of $\texttt{AKR}_2(R_{st})$. In other words, the moats around these components grow beyond s or t in the anaerobic phase. Let C_r be such a connected component of F_{st} and assume that it captures $u \in \{s,t\}$. We can then show that the cost of path P_r^u in $G|F_{st+}$ is at least the total budget of all components on P_r^u excluding C_r itself.

LEMMA 5.7. *Let $u \in \{s,t\}$ and assume that C_r for $1 \leq r \leq p$ is a connected component of F_{st+} on P that captures u. Let $\mathcal{K}$ be the index set of connected components on P_r^u excluding C_r that capture u. Furthermore, let $\mathcal{M}$ be the set of indices of those components on P_r^u that do not capture u. We must have*

$$c_{G|F_{st+}}(P_r^u) \geq (2\gamma+1) \cdot \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}).$$

PROOF. For ease of notation and w.l.o.g. we now assume that $u = s$. We first consider components C_i with $i \in \mathcal{M}$. Such a component dies in the anaerobic phase of $\texttt{AKR}_2(R_{st})$ before any other component on P reaches it. In other words, the components in $\mathcal{M}$ exhaust their budget without capturing either s or t. The components $C_1, \ldots, C_q$ in the example in Figure 2 are such components; the area covered by the dark- and light-grey rings for these components is part of the s,t-path P. An application of Lemma 5.3 shows that the total dual load on path P from components C_i for $i \in \mathcal{M}$ is at least $(2\gamma+1)(h_{i,s} + h_{i,t})$.

For ease of notation renumber the components such that

$$\mathcal{K} = \{1, \ldots, r-1\}$$

and such that C_{i+1} captures s after C_i for all $1 \leq i < r$.

Consider component C_i for $1 \leq i < r$. C_i must be dead at the time τ at which C_{i+1} captures it in $\texttt{AKR}_2(R_{st})$ since otherwise C_i and C_{i+1} would be part of the same connected component of F_{st+}.

This has two consequences: first, component C_i loads the C_i, C_{i+1}-segment of P until its budget runs out. The argument in Lemma 5.3 shows that the dual load of C_i on the C_i, C_{i+1}-segment of P is at least

$$(2\gamma+1)\min\{h_{i,s}, h_{i,t}\}.$$

Second, the component C_i must be dead when it is captured by C_{i+1}. Thus, the load of C_{i+1} on the C_i, C_{i+1}-segment of P is at least

$$(2\gamma+1)\max\{h_{i,s}, h_{i,t}\}.$$

This means that the cost of path P_r^s in $G|F_{st+}$ is at least

$$(2\gamma+1) \cdot \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,t} + h_{i,s}).$$

The lemma follows. □

Let $\mathcal{L}_u$ be the set of indices of connected components that capture $u \in \{s,t\}$. We then define

$$\mathcal{L} = \{\max_{l \in \mathcal{L}_s} l, \min_{q \in \mathcal{L}_t} q\}.$$

For ease of notation we also define $\mathcal{K} = (\mathcal{L}_s \cup \mathcal{L}_t) \setminus \mathcal{L}$. Finally, we let $\mathcal{M}$ be the set of indices of connected components of F_{st+} on P that do not capture either s or t. Observe that this means that $\{l+1, \ldots, q-1\} \subseteq \mathcal{M}$ in the case where $\mathcal{L} = \{l, q\}$ with $1 \leq l < q \leq p$.

In the following corollary, we use $h_{\mathcal{L}}$ as a short for $h_{l,s} + h_{q,t}$ and let

$$h^i_{\mathcal{L}} = h_{l,t} + h_{q,s}$$

if $\mathcal{L} = \{l, q\}$ for $1 \leq l < q \leq p$ and $h^i_{\mathcal{L}} = 0$ otherwise.

Corollary 5.8. *With the notation defined above, we must have*

$$2\gamma \cdot \left(h^i_{\mathcal{L}} + \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}) \right) \leq \bar{c}(P) + h_{\mathcal{L}}.$$

Proof. Lemma 5.7 implies that

$$(2\gamma + 1) \cdot \left(h^i_{\mathcal{L}} + \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}) \right) \leq c_{G|F_{st+}}(P).$$

Subtracting $\sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t})$ on both sides yields

$$(2\gamma + 1) h^i_{\mathcal{L}} + 2\gamma \cdot \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}) \leq \bar{c}(P) + \sum_{i \in \mathcal{L}} (h_{i,s} + h_{i,t}). \tag{14}$$

Adding $h_{\mathcal{L}} - \sum_{i \in \mathcal{L}} (h_{i,s} + h_{i,t})$ to both sides of (14) finishes the proof. □

5.4 A general lower-bound for χ_{st}

We are now ready to give a proof of Lemma 4.4. We restate the lemma here for completeness.

Lemma. *Let $\mathcal{I}$ be the set of indices of components on P that contain terminals that interfere with s or t, i.e.*

$$\mathcal{I} = \{i \in \{1, \ldots, p\} \,:\, \exists v' \in C_i \text{ that interferes with } \{s,t\}\}.$$

For $\gamma \geq 1/2$, we must have

$$\chi_{st} \geq \frac{1}{2} \cdot \mathtt{sl} + \gamma \cdot \sum_{1 \leq i \leq p, i \notin \mathcal{I}} (h_{i,s} + h_{i,t})$$

where the slack *in the residual cost $\bar{c}(P)$ is defined as*

$$\mathtt{sl} = \max \left\{ 0, \bar{c}(P) + \left(\sum_{i \in \mathcal{I}} (h_{i,s} + h_{i,t}) \right) - 2\gamma \cdot \sum_{1 \leq i \leq p, i \notin \mathcal{I}} (h_{i,s} + h_{i,t}) \right\}.$$

Proof. We know from Corollary 5.6 that $\mathcal{I}$ is either empty or consists of index m only (in the case where there are interfering terminals on P). We subdivide the argument into two parts depending on the existence of interfering terminals that are on path P.

Interfering terminals on P. Corollary 5.6 shows that there exists an index $m \in \{1, \ldots, p\}$ such that C_m contains all terminals that interfere with s or t. Consider $u \in \{s, t\}$ and let $\tau_u \le \texttt{age}(s, t)$ be the time in $\texttt{AKR}(R)$ when u meets the first interfering terminal in $\mathcal{R}[C_m]$. Lemma 4.3 shows that

$$h_{m,u} \le \tau_u$$

for $u \in \{s, t\}$.

Observe that definition (8) implies that the residual cost of P is exactly $\tau_s + \tau_t$. Corollary 5.6 shows that there are no interfering terminals for s and t outside C_m. Hence,

$$\chi_{st} = \tau_s + \tau_t = \bar{c}(P) \ge h_{m,s} + h_{m,t}. \tag{15}$$

As in Corollary 5.8 we let $\mathcal{K}$ be the index set of components that capture s or t excluding m. We also let $\mathcal{M}$ be the set of indices of components on P that do not capture s or t. Corollary 5.8 implies that

$$\bar{c}(P) + h_{m,s} + h_{m,t} \ge 2\gamma \cdot \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}). \tag{16}$$

In the case of interfering terminals on P, the definition of slack reduces to

$$\texttt{sl} = \max \left\{ 0, \bar{c}(P) + (h_{m,s} + h_{m,t}) - 2\gamma \cdot \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}) \right\}$$

as $\mathcal{I} \cup \mathcal{M} = \{1, \ldots, m-1, m+1, \ldots, p\}$. In other words, $\texttt{sl}$ is precisely the slack in inequality (16). Hence, (15) and (16) imply

$$\chi_{st} \ge \frac{\bar{c}(P) + h_{m,s} + h_{m,t}}{2} = \frac{1}{2} \cdot \left(\texttt{sl} + 2\gamma \cdot \sum_{i \in \mathcal{K} \cup \mathcal{M}} (h_{i,s} + h_{i,t}) \right)$$

and this finishes the proof in the case of interfering terminals on P.

No interfering terminals on P. In the following we use v_s and v_t to denote terminals that interfere with s and t, respectively. Similarly, we let C_s and C_t be connected components of F_{st} that contain vertices v_s and v_t.

Corollary 5.5 shows that we need to consider only two cases: In the two-sided case, both s and t see interference from distinct terminals v_s and v_t. Notice that $C_s \ne C_t$ in this case since otherwise $C_s = C_t$ would be on P. In the one-sided case, only one of s and t sees interference.

[Case 1: *Two-sided interference*] Let τ_s and τ_t be the times when s meets v_s and when t meets v_t, respectively, in $\texttt{AKR}(R)$. Let P_{v_s} and P_{v_t} be the respective paths that $\texttt{AKR}$ adds at these times. Corollary 5.4 shows that the combined load from C_s and C_t on $\langle P_{v_s}, P, P_{v_t} \rangle$ is at least

$$(4\gamma + 2) \cdot \texttt{age}(s, t) \ge (2\gamma + 1) \cdot \bar{c}(P). \tag{17}$$

Define sets $\mathcal{L}_u$ for $u \in \{s, t\}$ as in Corollary 5.8 and consider set C_i for $i \in \mathcal{L}_u$. W.l.o.g. assume that u is the first vertex in $\{s, t\}$ that is captured by C_i. Now observe that u and C_i meet at a time $\tau \le \texttt{age}(s, t)$ in $\texttt{AKR}(R)$. An argument similar to that used in the proof of Corollary 5.4 therefore shows that C_i captures u by time $2 \cdot \texttt{age}(s,t)$ in the anaerobic phase. Corollary 5.5 shows that v_u captures u by time $2 \cdot \texttt{age}(s,t)$ as well. As C_u is not on P, this must mean that C_i is dead when C_u captures u.

Hence component C_i must have extended fully in the anaerobic phase of $\texttt{AKR}_2(R_{st})$ for all $1 \le i \le p$ before either C_s or C_t reach it in the anaerobic phase. A careful look at Corollary 5.4 shows that the load in (17) is inflicted before time $(2\gamma+1)\cdot\texttt{age}(s,t)$ in the anaerobic phase and thus, both C_s and C_t are active at this time.

Therefore the load in (17) has to be at most

$$c_{G|F_{st+}}(P)+2\tau_s+2\tau_t-(2\gamma+1)\cdot\sum_{i=1}^{p}(h_{i,s}+h_{i,t}) = \bar{c}(P)+2\tau_s+2\tau_t-2\gamma\cdot\sum_{i=1}^{p}(h_{i,s}+h_{i,t}).$$

Solving for $\tau_s + \tau_t$ gives

$$\tau_s + \tau_t \ge \frac{1}{2}\cdot\bar{c}(P) + \gamma\cdot\sum_{i=1}^{p}(h_{i,s}+h_{i,t}).$$

Observing that $\chi_{st} = \tau_s + \tau_t$ concludes the proof in Case 1.

[Case 2: *One-sided interference*] We assume, w.l.o.g., that there is no terminal v_t that interferes with t. As before let τ_s denote the time when s meets the first interfering terminal v_s in $\texttt{AKR}(R)$. Since t sees no interference in $\texttt{AKR}(R)$, we have

$$\chi_{st} = \texttt{age}(s,t) + \tau_s = \frac{1}{2}\cdot\bar{c}(P) + \tau_s. \tag{18}$$

We again let C_s be the connected component of F_{st} that captures s. Let C_i for $1 \le i \le p$ be a connected component of F_{st+} on path P. Component C_i must be dead when C_s captures it during the anaerobic phase of $\texttt{AKR}_2(R_{st})$ since otherwise C_s would be on path P as well. In other words, C_i must have finished its budget-growth phase by the time C_s reaches it in the anaerobic phase.

In the following we let $\mathcal{L} = \{l, q\}$ with $1 \le l \le q \le p$. Consider the case where $l < q$ and hence $\mathcal{L}$ contains exactly two indices. Observe that C_l captures s by time $2\cdot\texttt{age}(s,t)$ in this case. Otherwise C_l would also capture C_q and this contradicts the assumption $l \ne q$. Corollary 5.4 shows that C_s captures s by time $2\cdot\texttt{age}(s,t)$ as well. As before, this implies that C_l is dead by the time C_s captures s.

Lemma 5.7 shows that

$$(2\gamma+1)\cdot\sum_{i=q+1}^{p}(h_{i,s}+h_{i,t}) \le c_{G|F_{st+}}(P_q^t). \tag{19}$$

Let P_{v_s} be the path that is added in $\texttt{AKR}(R)$ when s and v_s meet and let $P' = \langle P_{v_s}, P_q^s\rangle$ be the concatenation of P_{v_s} and P_q^s.

Assume first that C_s captures C_q. This means that C_q is dead when the moats containing C_s and C_q meet in the anaerobic phase of $\texttt{AKR}_2(R_{st})$. Therefore, the dual load of C_s on path P' is at least $\texttt{b}_q$. The total load coming from super-vertices contained in sets $\{C_i\}_{1\le i\le q}$ and from C_s on path P' is bounded by $c_{G|F_{st+}}(P_q^s)+2\tau_s$. These observations imply

$$(2\gamma+1)\cdot\sum_{i=1}^{q}(h_{i,s}+h_{i,t}) \le c_{G|F_{st+}}(P_q^s) + 2\tau_s. \tag{20}$$

On the other hand assume that C_s does not capture C_q. C_q may still capture s but this must happen after C_s is dead and hence at a time later than

$$(2\gamma + 1) \cdot \mathtt{age}(v_s) \geq (2\gamma + 1) \cdot \mathtt{age}(s, t)$$

in the anaerobic phase. In other words, C_s and C_q are contained in different moats in $A_+^{(2\gamma+1)\mathtt{age}(s,t)}$. Hence

$$2 \cdot (2\gamma + 1) \cdot \mathtt{age}(s, t) + (2\gamma + 1) \cdot \sum_{1 \leq i < q} (h_{i,s} + h_{i,t}) \leq c_{G|F_{st+}}(P_q^s) + 2\tau_s.$$

Lemma 4.3 implies that $(h_{q,s} + h_{q,t}) \leq 2 \cdot \mathtt{age}(s, t)$ and the above inequality therefore yields

$$(2\gamma + 1) \cdot \sum_{1 \leq i \leq q} (h_{i,s} + h_{i,t}) \leq c_{G|F_{st+}}(P_q^s) + 2\tau_s. \tag{21}$$

Inequalities (19), (20), and (21) imply that $(2\gamma+1) \cdot \sum_{i=1}^{p} (h_{i,s} + h_{i,t}) \leq c_{G|F_{st+}} + 2\tau_s$ and hence

$$2\gamma \cdot \sum_{i=1}^{p} (h_{i,s} + h_{i,t}) \leq \bar{c}(P) + 2\tau_s.$$

It can be seen that (18) together with the definition of slack $\mathtt{sl}$ implies

$$\chi_{st} \geq \gamma \cdot \sum_{i=1}^{p} (h_{i,s} + h_{i,t}) + \frac{\mathtt{sl}}{2}$$

and the lemma follows. □

Acknowledgment. We thank R. Ravi for sharing his insights on primal-dual algorithms for Steiner forests and trees with us.

REFERENCES

AGRAWAL, A., KLEIN, P., AND RAVI, R. 1995. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM J. Comput. 24*, 440–456.

AWERBUCH, B. AND AZAR, Y. 1997. Buy-at-bulk network design. In *Proceedings, IEEE Symposium on Foundations of Computer Science*. 542–547.

BARTAL, Y. 1998. On approximating arbitrary metrics by tree metrics. In *Proceedings, ACM Symposium on Theory of Computing*. 161–168.

FEIGENBAUM, PAPADIMITRIOU, AND SHENKER. 2001. Sharing the cost of multicast transmissions. *JCSS: Journal of Computer and System Sciences 63*, 21–41.

FLEISCHER, L., KÖNEMANN, J., LEONARDI, S., AND SCHÄFER, G. 2006. Simple cost sharing schemes for multi-commodity rent-or-buy and stochastic Steiner tree. In *Proceedings, ACM Symposium on Theory of Computing*. 663–670.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability: A guide to the theory of NP-completeness.* W. H. Freeman and Company, San Francisco.

GOEMANS, M. X. AND WILLIAMSON, D. P. 1995. A general approximation technique for constrained forest problems. *SIAM J. Comput. 24*, 296–317.

GUPTA, A., KUMAR, A., PAL, M., AND ROUGHGARDEN, T. 2003. Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings, IEEE Symposium on Foundations of Computer Science*. 606–615.

GUPTA, A., KUMAR, A., AND ROUGHGARDEN, T. 2003. Simpler and better approximation algorithms for network design. In *Proceedings, ACM Symposium on Theory of Computing*. 365–372.

Gupta, A., Pál, M., Ravi, R., and Sinha, A. 2004. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings, ACM Symposium on Theory of Computing*. 417–426.

Jain, K. and Vazirani, V. V. 2001. Applications of approximation algorithms to cooperative games. In *Proceedings, ACM Symposium on Theory of Computing*. 364–372.

Kumar, A., Gupta, A., and Roughgarden, T. 2002. A constant-factor approximation algorithm for the multicommodity. In *Proceedings, IEEE Symposium on Foundations of Computer Science*. 333–344.

Pál, M. and Tardos, É. 2003. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings, IEEE Symposium on Foundations of Computer Science*.