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

Luca Becchetti* Jochen Könemann† Stefano Leonardi* M. Pál‡

Abstract

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 $1 \le i \le k$ 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. [8] 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 has previously been introduced [9] 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, Klein and Ravi [1]. Our result significantly reduces the gap between the single-sink [9] and multi-sink case.

1 Introduction

In the *multi-commodity rent-or-buy problem* (MROB) we are given an undirected graph $G = (V, E)$, terminal pairs $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, non-negative costs c_e for all edges $e \in E$, and a parameter $M \ge 0$. 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 nodes and a root node r. The goal is now to provision the network such that all terminals can send a specified amount of flow to the root node r simultaneously. A recent result of Gupta et al. [9] gives a 3.55 approximation algorithm for the problem.

Awerbuch and Azar [2] and Bartal [3] were the first to give an $O(\log |V| \log\log |V|)$-approximation algorithm for the MROB problem. Later, Kumar, Gupta and Roughgarden [13] give the first constant approximation algorithm for the problem based on a primal-dual approach. A more recent result by Gupta et al. [8] builds on the techniques used by Gupta et al. [9] for the single-commodity rent-or-buy problem and obtains a 12-approximation for the MROB problem. Their work also uses the *cost-sharing* concept from game-theory (see, e.g., [5, 10, 14]) in the analysis of the algorithm.

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 $R = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. The goal is to find a forest F of minimum total cost such for all $1 \le i \le k$, 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[6] and Max-SNP hard. On the positive side, Agrawal, Klein and Ravi [1] and later Goemans and Williamson [7] give a primal-dual 2-approximation for the problem.

The MROB algorithm from [8] crucially relies on the primal-dual algorithm for Steiner forest of Agrawal

*Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy. Email: {leon,becchett}@dis.uniroma1.it.

†Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Email: jochen@uwaterloo.ca. This work was done while being on leave at the Dipartimento di Informatica e Sistemistica at Università di Roma "La Sapienza", Italy.

‡Department of Computer Science, Cornell University, Ithaca, NY 14853, USA. Email: mpal@cs.cornell.edu.

et al. [1]. The algorithm in [8] 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 $R \backslash R^0$ 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 [8] 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.

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

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

In [8], 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 $\beta = 2 + \sqrt{2}$ in Theorem 1.1 together with Theorem 1.2 implies the following corollary:

COROLLARY 1.1. *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 [1] and computes an approximate Steiner forest F' for a given set of terminal pairs R.

The second phase identifies the terminal nodes in each tree T in F'. The newly created *super-node* is treated as a terminal of another Steiner tree instance. We then run a *budgeted* version of the primal-dual algorithm for Steiner trees to obtain a final Steiner forest. Here, we borrow ideas from earlier work on *prize-collecting* variants of the Steiner tree problem (see, e.g., [7]).

The benefit of our method is two-fold: First, we combine existing primal-dual algorithms in a black-box fashion as opposed to modifying technical details of an existing method. This leads to a much simplified algorithm and more intuitive analysis. Second, since we use standard primal-dual algorithms for the Steiner forest and Steiner tree problems we inherit some their nice properties. Most notably, our dual solutions are laminar.

Organization of this paper. The next section recaps the primal-dual Steiner forest algorithm from [1] since our methods and its analysis strongly relies on it. Subsequently, we present in Section 3 our β-strict Steiner forest. Section 4.1 has a complete analysis of a 5 approximation for a special case of the algorithm. The analysis of the general case and the main ingredients of the proof of Theorem 1.1 are given in Section 4.2. Due to space limitations we defer proofs of certain technical lemmas to the full version of the paper[4].

2 The Minimum-cost Steiner forest problem

We present the primal-dual algorithm (subsequently referred to as `AKR`) for the Steiner forest problem due to Agrawal, Klein, and Ravi [1]. The algorithm 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 $|\{s,t\} \cap U| = 1$. For a subset U of the nodes 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:

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

For a pair of nodes $u, v \in R$, let c_{uv} be the minimum cost of any u, v-path in G. It can be shown (see [11, 12]) that the following linear program is equivalent to the dual of the LP relaxation (LP) of (IP):

$$
\begin{aligned}
&\text{(D)} \qquad \max \quad \sum_{U \subseteq R} y_U \\
&\text{(2.1)} \qquad \text{s.t} \quad \sum_{U \subseteq R : |\{u,v\} \cap U| = 1} y_U \leq c_{uv} \quad \forall u, v \in R \\
&\qquad\qquad\qquad y \geq 0
\end{aligned}
$$

In our presentation, we let AKR construct a primal solution for (LP) and a dual solution for (D).

We think of an execution of Algorithm AKR as a process over time and let x^t and y^t be the primal incidence vector and dual feasible solution at time t. We also use F^t to denote the forest corresponding to x^t. The algorithm now starts with $x_e^0 = 0$ for all $e \in E$ and $y_U^0 = 0$ for all $U \in \mathcal{U}$.

Assume that the forest F^t at time t is infeasible. For a connected component C of F^t, we use $R[C]$ to denote the set of terminal nodes in C. We say that a connected component C of F^t is *active* if $R[C] \in \mathcal{U}$. The algorithm raises the dual variables corresponding to all active connected components of F^t simultaneously until a constraint of type (2.1) is satisfied with equality. Suppose that this happens for terminals $u, v \in R$ and also assume that $u \in C_u$ and $v \in C_v$ for connected components C_u and C_v of F^t. We then add a u, v-path of smallest total cost to F^t and continue.

The algorithm terminates at the earliest time t^* when F^{t^*} is a feasible Steiner forest. A proof of the following main result from [1] can also be found in [4].

THEOREM 2.1. *Suppose that algorithm* AKR *stops at time t^*. We then must have that*

$$c(F^{t^*}) \leq 2 \cdot \sum_{U \subseteq R} y_U^{t^*}.$$

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 $(2+2/(\beta-2))$-approximate and β-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 [8], Gupta et al. define the notion of *β-strict* algorithms for the minimum-cost Steiner forest problem.

DEFINITION 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. *$\sum_{(s_i,t_i) \in R} \chi_i \leq c(F^*)$ where F^* is a feasible Steiner forest for R of minimum total cost, and*
2. *$c_{G|F_i}(s_i, t_i) \leq \beta \cdot \chi_i$ 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 [8]. We run AKR on input graph G with terminal pairs R. Let $\mathtt{age}_i$ be the time at which s_i and t_i meet during this execution.

For an active component U at some time t during the execution of AKR(R) we pick a distinct terminal $r \in R[U]$ of maximum age and declare it the *beneficiary* of U. We then define an indicator variable δ_t^i for all terminal pairs (s_i, t_i) and for all times $t \geq 0$:

$$
\delta_t^i = \begin{cases} 2 & : \text{ Both, } s_i \text{ and } t_i \text{ are beneficiaries at time } t < \mathtt{age}_i \\ 1 & : \text{ Exactly one of } s_i \text{ and } t_i \text{ is a beneficiary at time } t < \mathtt{age}_i \\ 0 & : \text{ otherwise.} \end{cases}
$$

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

$$\chi_i = \int_0^{t^*} \delta_t^i \, dt. \tag{3.2}$$

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 $R^0 = R \setminus \{(s, t)\}$. The new algorithm $\mathtt{AKR}_2$ first uses AKR to compute a feasible Steiner forest F' for terminal set R^0. The second phase of the algorithm adds more paths to connect components of F' that are *close* to each other. Selecting paths carefully in this second phase yields a Steiner forest F^0 whose cost is only a constant factor worse than that of F' and that satisfies the necessary strictness properties.

We now describe the algorithm $\mathtt{AKR}_2$ in greater detail. The algorithm works on input R^0 and has two phases:

[**Aerobic Phase**] In this phase we execute AKR on terminal set R^0. This produces a forest F' that is feasible for R^0 and a corresponding dual solution $\{y'_U\}_{U \subseteq R^0}$. We let $\mathcal{C}'$ be the set of connected components of F' and define $\mathcal{U}'$ to be the set of subsets of R^0 that receive positive dual in $\texttt{AKR}(R^0)$, i.e.

$$\mathcal{U}' = \{U \subseteq R^0 \,:\, y'_U > 0\}.$$

We now use F' to create a new graph G' from the original graph G: For each connected component C of F', we identify the terminals in $R^0[C]$. In other words, we replace the set $R^0[C]$ by a new vertex C. Each edge $(u, v) \in \delta(R^0[C])$ with $u \in R^0[C]$ and $v \notin R^0[C]$ is substituted by a new edge (C, v) with cost c_{uv}. Finally, we delete all edges $e \in E$ that have both end-points in $R^0[C]$. The graph G' contains a *super-node* C for each non-trivial connected component $C \in \mathcal{C}'$.

[**Anaerobic Phase**] Recall that whenever $\texttt{AKR}(R^0)$ 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', we then let

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

be the set of moats whose beneficiary is a terminal in C. The set $\{U'_C\}_{C \in \mathcal{C}'}$ is a partition of $\mathcal{U}'$.

For a node $C \in \mathcal{C}'$ let $\texttt{age}_C$ denote the maximum age among the terminal pairs in $R^0[C]$. Then define the *budget* $\texttt{b}_C$ of node $C \in \mathcal{C}'$ as

$$\texttt{b}_C = \texttt{age}_C + \gamma \cdot \sum_{U \in \mathcal{U}'_C} y'_U \tag{3.3}$$

for a parameter $\gamma \geq 1/2$. For nodes $v \in V[G'] \setminus \mathcal{C}'$ we let $\texttt{b}_v = 0$.

We now run a budgeted version of the Steiner tree algorithm that bears resemblance to the prize-collecting Steiner tree algorithm from [7]: Say a connected component of the current forest is *active* if it has remaining budget. At any point during the algorithm we then raise the dual variables of all active connected components in the current forest. We decrease the budget of these components at the rate at which their duals grow.

Two possible events can occur:

Merge A path connecting two active connected components C_1 and C_2 in the current forest becomes tight. In this case, add the edges of the path to the current forest and by this create a new connected component C. The budget of this new component C is the sum of the remaining budgets of C_1 and C_2.

Death A connected component runs out of budget in the growth phase. In this case the component simply dies and we continue growing those components that have positive remaining budget.

We let F'' be the forest in G' computed during the anaerobic phase and let $\{y''_U\}_{U \subseteq V[G']}$ be the corresponding dual solution. We obtain the final forest F^0 from F'' by replacing each super-node $v \in \mathcal{C}'$ by the corresponding connected component in F'.

The proof of the following lemma is a direct consequence of Theorem 2.1.

LEMMA 3.1. *The cost of the forest F^0 computed by $\texttt{AKR}_2$ on terminal set R^0 is at most*

$$(2 + 2\gamma) \cdot \texttt{opt}_{R^0}$$

where $\texttt{opt}_{R^0}$ is the cost of a minimum-cost feasible Steiner forest for terminal set R^0.

4 Analyzing the strictness of Algorithm $\texttt{AKR}_2$

We focus on terminal pair $(s, t) \in R$. Recall that $R^0 = R \setminus \{(s, t)\}$ and let F denote the forest computed by AKR on input R. As before we let F^0 be the forest computed by $\texttt{AKR}_2$ on input R^0. As in Section 1 we use $G|F^0$ to denote the graph obtained from G by contracting the connected components of forest F^0. In order to prove that $\texttt{AKR}_2$ is β-strict we need to show that

$$c_{G|F^0}(s, t) \leq \beta \cdot \chi_{st}. \tag{4.4}$$

Let P_{st} be the unique s,t-path in the forest F. Notice that P_{st} may enter and leave a given connected component of F^0 more than once. In this case we obtain a new s,t-path P in $G|F^0$ from P_{st} by deleting such loops.

The rough outline is as follows: The cost of P in $G|F^0$ is at least $c_{G|F^0}(s, t)$. We will show that $c_{G|F^0}(P) \leq \beta \cdot \chi_{st}$ where $c_{G|F^0}(P)$ is the cost of path P in the graph $G|F^0$ and this implies (4.4) since $c_{G|F^0}(s, t) \leq c_{G|F^0}(P)$.

We let $C_1, \ldots, C_p$ be the connected components of F^0 that P touches in that order. Since P is loop-less in $G|F^0$ it follows that each connected component of F^0 occurs at most once in this list. We also assume that s and t are not part of $\bigcup_{i=1}^{p} C_i$. Finally, let p_m be the point on P where the active moats containing s and t meet during the execution of $\texttt{AKR}(R)$.

Recall that we use $\{y_U\}_{U \subset R}$ to denote the dual solution computed by $\texttt{AKR}(R)$. We then define the *residual cost* $\widetilde{c}_e$ of edge $e \in E$ as

$$\widetilde{c}_e = c_e - \sum_{U \subseteq R^0, e \in \delta(U)} y_U. \tag{4.5}$$

The residual cost of edge e is the part of c_e that does not feel dual load from subsets of R^0 in $\texttt{AKR}(R)$. Therefore, roughly speaking, s and t gather $\widetilde{c}_e$ units of cost-share while traversing edge e.

We can now express $c_{G|F^0}(P)$ as the sum of residual and *hidden* costs of P. Formally, for a connected component C_i of F^0 that is on path P, we let P_i^s and P_i^t be the s, C_i-segment and the C_i, t-segment of P, respectively. Let $\mathcal{C}'[C]$ be the set of connected components of forest F' that are contained in a connected component C of F^0. We then let $C_i^s, C_i^t \in \mathcal{C}'[C_i]$ be the first and last connected components on the s,t-path P_{st} in G'.

We then define $h_{i,s}$ and $h_{i,t}$ to be the cost of the two hidden segments of P inside C_i, i.e.

$$h_{i,u} = \sum_{U \subseteq R^0, U \cap C_i \neq \emptyset} |\delta(U) \cap P_i^u| \cdot y_U \tag{4.6}$$

for $u \in \{s,t\}$ and let $h_i = \max\{h_{i,s}, h_{i,t}\}$. The cost of path P in $G|F^0$ can now be expressed as

$$c_{G|F^0}(P) = \widetilde{c}(P) + \sum_{i=1}^{p} (h_{i,s} + h_{i,t}). \tag{4.7}$$

In the following, we use $\mathtt{age}_{s',t'}$ and $\mathtt{age}^0_{s',t'}$ to denote the time at which the terminals of $(s',t') \in R^0$ meet during the execution of $\mathtt{AKR}(R)$ and $\mathtt{AKR}(R^0)$, respectively. We extend this notion to sets $C \subseteq V$ by letting $\mathtt{age}_C = \max_{(s',t') \in R^0[C]} \mathtt{age}_{s',t'}$.

Consider two connected components $C_1, C_2 \in \mathcal{C}'$. We say that C_1 encloses C_2 in $\mathtt{AKR}(R^0)$ if there is a set $U \in \mathcal{U}'_{C_1}$ that contains C_2. In other words, there is a point in time (the *time of enclosure*) during $\mathtt{AKR}(R^0)$ at which an active moat containing C_1 grows across a dead moat containing C_2.

For $1 \leq i \leq p$ and for $u \in \{s,t\}$, we let $\overline{C_i^u} \in \mathcal{C}'[C_i]$ be the connected component in C_i that encloses C_i^u latest ($\overline{C_i^u} = C_i^u$ if C_i^u is not enclosed by any other connected component in C_i). Intuitively, the budget-growth of component $\overline{C_i^u}$ along path P_i^u reserves parts of the residual cost of P_i^u which are later used to pay for the segments of P_i^u that are hidden within C_i.

For ease of notation we define the *excess budget*

$$\mathtt{b}^0_{i,u} = 2\gamma \cdot \sum_{U \in \mathcal{U}'_{\overline{C_i^u}}} y'_U \tag{4.8}$$

and let $\mathtt{b}_{i,u} = \mathtt{b}^0_{i,u} + h_{i,u}$. We also use $\mathtt{b}_i$ for the maximum of $\mathtt{b}_{i,s}$ and $\mathtt{b}_{i,t}$.

Lemma 4.1. *Let $1 \leq i \leq p$ and $u \in \{s,t\}$ and assume that u meets the first terminal from C_i at time T in $\mathtt{AKR}(R)$. Then we must have $h_{i,u} \leq \min\{T, \mathtt{age}^0_{\overline{C_i^u}}\}$. In particular this means that $h_i \leq \mathtt{age}_{s,t}$ for all $1 \leq i \leq p$.*

Proof. First, consider the case where $T \leq \mathtt{age}^0_{\overline{C_i^u}}$. Let $U \subseteq R$ be an active moat in $\mathtt{AKR}(R)$ at time $T' \geq T$ with $\delta(U) \cap P_i^u \neq \emptyset$. In this case u must clearly also be in U and hence the dual assigned to U does not contribute to $h_{i,u}$. At any time prior to T there exists at most one active moat loading P_i^u that intersects C_i and hence $h_{i,u} \leq T$.

Now assume that $T > \mathtt{age}^0_{\overline{C_i^u}}$. This means that u meets C_i only after time $\mathtt{age}^0_{\overline{C_i^u}}$ and the moat containing $\overline{C_i^u}$ is dead at this point. Since $\overline{C_i^u}$ encloses C_i^u we know that C_i^u must also be dead at time $\mathtt{age}^0_{\overline{C_i^u}}$. Moreover, as before, at any time $t \in [0, \mathtt{age}^0_{\overline{C_i^u}}]$ there is at most one moat loading P_i^u in $\mathtt{AKR}(R)$. Therefore we must have $h_{i,u} \leq \mathtt{age}^0_{\overline{C_i^u}}$.

The lemma follows since $T \leq \mathtt{age}_{s,t}$.

Lemma 4.2. *For all connected components C_i on P and for $u \in \{s,t\}$ we must have $\mathtt{b}^0_{i,u} \geq 2\gamma h_{i,u}$.*

Proof. Observe that $\mathtt{AKR}(R^0)$ grows at least two moats that are contained in $\overline{C_i^u}$ at all times $t \in [0, \mathtt{age}^0_{C_i^u}]$. Therefore we must have

$$\mathtt{b}^0_{i,u} = \gamma \cdot \sum_{U \in \mathcal{U}'_{\overline{C_i^u}}} y'_U \geq 2\gamma \cdot \mathtt{age}^0_{\overline{C_i^u}}.$$

An application of Lemma 4.1 finishes the proof.

We define a useful interference notion that is needed throughout the rest of this paper.

Definition 2. *Let $(s',t') \in R$ be a terminal pair with $\mathtt{age}_{s,t} \leq \mathtt{age}_{s',t'}$. We say that terminal $v' \in \{s',t'\}$* interferes *with $v \in \{s,t\}$ if v and v' meet before time $\mathtt{age}_{s,t}$ in $\mathtt{AKR}(R)$. Formally v' interferes with v if there is a set $U \subseteq R$ such that $\{v,v'\} \subseteq U$ and $y_U > 0$.*

Recall that we use $\mathcal{C}'$ to denote the set of connected components in the forest F' produced by the aerobic phase of $\mathtt{AKR}_2(R^0)$.

Definition 3. *A component $C \in \mathcal{C}'$* captures *a node v if a moat containing C reaches v in the anaerobic phase of $\mathtt{AKR}_2(R^0)$. We also say that a connected component C of F^0 captures v if there is a connected component C' of F' that captures v and $C' \subseteq C$.*

4.1 The strictness of $\mathtt{AKR}_2$: A simple case As a warm up for the reader, we prove the strictness result under the following assumption:

Assumption 1. *There are no interfering terminals for terminal pair (s,t) and none of the components in $\{\overline{C_i^u}\}_{u \in \{s,t\}, 1 \leq i \leq p}$ captures s or t.*

We will argue that Assumption 1 implies that the amount of cost-share recovered by s and t is at least the

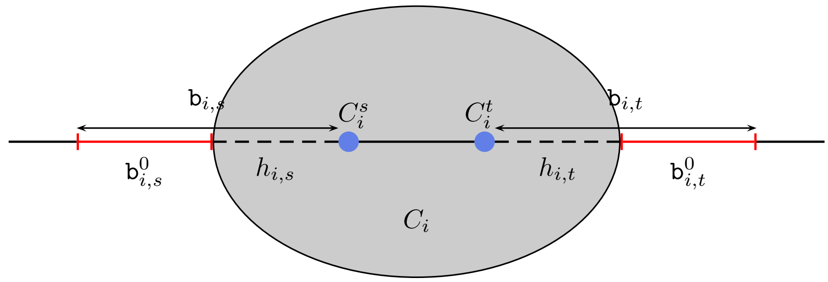

Figure 1: Connected component C_i on path P together with its budget reservation on P. The budget growth $\mathtt{b}_i^u$ of component C_i^u is split into $h_{i,u}$ and $\mathtt{b}^0_{i,u}$.

residual cost $\widetilde{c}(P)$ of path P. This in turn will enable us to prove that the algorithm is 5-strict in this case.

LEMMA 4.3. *Consider two connected components $C_1, C_2 \in \mathcal{C}'$ such that C_1 encloses C_2 at time T in* $\mathtt{AKR}(R^0)$*. Then C_1 also encloses C_2 by time T in the anaerobic phase of* $\mathtt{AKR}_2(R^0)$.

Proof. Since C_1 encloses C_2 at time T in $\mathtt{AKR}(R^0)$, there must exist terminals $s_1 \in R^0[C_1]$ and $s_2 \in R^0[C_2]$ and a tight path P_{12} connecting them at time T in $\mathtt{AKR}(R^0)$. By definition, the budget $\mathtt{b}_{C_j}$ of component C_j for $j \in \{1,2\}$ is at least the maximum age of any terminal in C_j. Therefore, path P_{12} must also be tight in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ at time T.

The Lemma implies that the connected components in $\mathcal{C}'(C_i)$ inflict at least $h_{i,u}$ units of dual on P_i^u by time $\mathtt{age}_{\overline{C_i^u}}$ in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ for all $1 \le i \le p$ and for $u \in \{s,t\}$. Since the remaining budget of component $\overline{C}_i^u$ at this point is $\mathtt{b}^0_{i,u}$ it follows that the load on P_i^u coming from C_i in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ is at least

$$\mathtt{b}_{i,u} = h_{i,u} + \mathtt{b}^0_{i,u} \ge (2\gamma+1)h_{i,u}$$

where the inequality follows from Lemma 4.2. Assumption 1 implies that

(4.9)
$$(2\gamma+1)\sum_{i=1}^{p}(h_{i,s}+h_{i,t}) \le \sum_{i=1}^{p}(\mathtt{b}_{i,s}+\mathtt{b}_{i,t}) \le c_{G|F^0}(P).$$

Let χ^1_{st} denote the cost-share of terminal pair (s,t) given Assumption 1. We now show that χ^1_{st} is at least equal to the residual cost of P.

LEMMA 4.4. *The cost share χ^1_{st} of terminal pair (s,t) is at least the residual cost $\widetilde{c}(P)$ of the s,t-path P in* $\mathtt{AKR}(R)$.

Proof. Let $U \subseteq R$ be an active moat in the execution of $\mathtt{AKR}(R)$ and let $u \in \{s,t\} \cap U$. By Assumption 1, u must be the beneficiary of U. Hence, the total cost-share collected by $\{s,t\}$ is exactly

$$\sum_{U \subseteq R, |\{s,t\}\cap U|=1} y_U = c_{G|F^0}(P) - \sum_{U \subseteq R^0} |\delta(U)\cap P| \cdot y_U$$

and the right-hand side of this equality is $\widetilde{c}(P)$.

In the anaerobic phase of $\mathtt{AKR}_2$, the super-node $C_i^u \in \mathcal{C}'$ extends along P_i^u for all $1 \le i \le p$ and for $u \in \{s,t\}$. This way, component C_i^u *reserves* $\mathtt{b}^0_{i,u}$ units of the residual cost $\widetilde{c}(P)$ of path P. The total amount of residual cost reserved for component i is therefore $\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t}$ and Lemma 4.4 shows that this translates into at least the same amount of cost-share.

We use this cost-share to *pay* for those segments of P in $G|F^0$ that feel dual load from C_i in the anaerobic phase of $\mathtt{AKR}_2(R^0)$. Specifically, showing β-strictness amounts to proving

(4.10)
$$\mathtt{b}_{i,u} = \mathtt{b}^0_{i,u} + h_{i,u} \le \beta \cdot \mathtt{b}^0_{i,u}$$

for all $1 \le i \le p$ and for $u \in \{s,t\}$. Remember that the size of $\mathtt{b}_{i,u}$ is controlled by the parameter γ in (3.3).

THEOREM 4.1. *For any $\beta \ge 2$ there is a polynomial-time $(2+1/(\beta-1))$-approximate and β-strict algorithm for the minimum-cost Steiner forest problem under Assumption 1.*

Proof. Define the *slack* $\mathtt{sl}$ of path P as the amount of residual cost that is not needed for budget-reservation in $\mathtt{AKR}_2$: $\mathtt{sl} = \widetilde{c}(P) - \sum_{i=1}^{p}(\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t})$. Equation (4.7) then shows that the cost $c_{G|F^0}(P)$ of path P is $\sum_{1\le i\le p}(\mathtt{b}^0_{i,s} + h_{i,s} + \mathtt{b}^0_{i,t} + h_{i,t}) + \mathtt{sl}$.

On the other hand we know from Assumption 1 that none of the components on path P capture s and t and hence equation (4.9) holds. The definition of residual cost together with Lemma 4.4 imply

$$\chi^1_{s,t} \geq \widetilde{c}(P) = \left(\sum_{i=1}^{p} \mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t} \right) + \mathtt{sl}.$$

Therefore showing $\mathtt{b}^0_{i,u} + h_{i,u} \leq \beta \mathtt{b}^0_{i,u}$ for all $1 \leq i \leq p$ and for $u \in \{s,t\}$ suffices to prove β-strictness. Equivalently we need to show $\mathtt{b}^0_{i,u} \geq 1/(\beta-1) \cdot h_{i,u}$ for all $1 \leq i \leq p$ and for $u \in \{s,t\}$. By Lemma 4.2, this is true for $\gamma \geq 1/2(\beta-1)$ and our final choice of $\beta = 2$ implies that $\gamma \geq 1/2$ as wanted.

Choosing $\beta = 2$ in Theorem 4.1 together with [8] yields:

COROLLARY 4.1. *There is a 5-approximate algorithm for the multicommodity rent-or-buy problem under Assumption 1*

4.2 The strictness of $\mathtt{AKR}_2$: The general case

The intuitive outline given above does not suffice to analyze the strictness of $\mathtt{AKR}_2$ in general. The problem is two-fold: First, there maybe components on P that capture s or t and hence (4.9) may not hold. Second, there may exist terminals that interfere with $\{s,t\}$.

In order to present a general relation between χ_{st} and the residual cost of P we need to handle the problem of insufficient residual cost. The presence of interfering terminals further complicates matters. We start with a few useful observations whose proofs are reported in [4].

4.2.1 Observations: Old terminals In this section we prove a few structural properties of the forest F^0 computed by $\mathtt{AKR}_2(R^0)$ pertaining to the location of terminal pairs (s',t') that interfere with (s,t). Recall that y'' is the dual solution computed by $\mathtt{AKR}_2(R^0)$ in the anaerobic phase. In the following we say that a connected component C of F^0 interferes with $u \in \{s,t\}$ if there is a terminal $u' \in R^0[C]$ that interferes with u.

LEMMA 4.5. *Let u' be a terminal that interferes with $u \in \{s,t\}$ and assume that u meets u' at time $T < \mathtt{age}_{s,t}$ in $\mathtt{AKR}(R)$. Let $C' \in \mathcal{C}'$ be the super-node in G' containing u'. The total dual value assigned to moats that contain both u and C' in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ must be at least $(2\gamma+1)\cdot \mathtt{age}_{st} - 2T$, i.e.*

$$\sum_{U \subseteq V[G'], \{u,u'\} \subseteq U} y''_U \geq (2\gamma+1)\cdot \mathtt{age}_{st} - 2T.$$

The following corollary is a consequence of Lemma 4.5.

COROLLARY 4.2. *Let u_1 and u_2 be terminals that interfere with $u \in \{s,t\}$. They both reach u by time $2\cdot \mathtt{age}_{s,t}$ in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ and there must exist a connected component C in F^0 with $\{u_1,u_2\} \subseteq R^0[C]$.*

Let u' be a terminal that interferes with $u \in \{s,t\}$. We say that u' *is on* P if u and u' meet in $\mathtt{AKR}(R)$ at some point p on P. Recall that p_m is the point on P where the moats of s and t collide in $\mathtt{AKR}(R)$.

LEMMA 4.6. *Let u' be a terminal on P that interferes with $u \in \{s,t\}$. Also let C' be the connected component of F' containing u'. In this case C' captures both s and t by time $2\cdot \mathtt{age}_{s,t}$ in the anaerobic phase of $\mathtt{AKR}_2(R^0)$. Moreover, there must be a connected component C_m for $1 \leq m \leq p$ that contains all interfering terminals.*

In the case of interfering terminals on P we will from now on use C_m to denote the connected component of F^0 that contains all interfering terminals.

LEMMA 4.7. *Let $u \in \{s,t\}$ and assume that C_r is a connected component of F^0 that captures u before time $(2\gamma+1)\cdot \mathtt{age}_{s,t}$ in $\mathtt{AKR}_2(R^0)$. Let u' be a terminal that interferes with u and let C' be its connected component in F'. Moreover let T be the time where u and u' meet in $\mathtt{AKR}(R)$ and let T' be the time when C' captures u in the anaerobic phase of $\mathtt{AKR}_2(R^0)$. We then must have either $u' \in C_r$ or $2T \geq T' \geq \mathtt{b}_r \geq h_r$.*

4.2.2 Observations: Insufficient residual cost Suppose that one or more connected components of the forest F' at the end of the aerobic phase of $\mathtt{AKR}_2(R^0)$ do not find enough space on P to reserve their portion of budget. In other words, they grow beyond s or t in the anaerobic phase. Let C_r be such a connected component of F' and assume that it captures $u \in \{s,t\}$. We can then show that the cost of path P^u_r in $G|F^0$ is at least the total budget of all components on P^u_r excluding C_r itself.

LEMMA 4.8. *Let $u \in \{s,t\}$ and assume that C_r for $1 \leq r \leq p$ is a connected component of F^0 on P that captures u. Let $\mathcal{C}$ be the index set of connected components on P^u_r excluding C_r that capture u. Furthermore, let $\mathcal{M}$ be the set of indices of those components on P^u_r that do not capture u. We must have*

$$c_{G|F^0}(P^u_r) \geq \sum_{i \in \mathcal{C}\cup\mathcal{M}} (\mathtt{b}_{i,s} + \mathtt{b}_{i,t}).$$

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 $\mathcal{C} = (\mathcal{L}_s \cup \mathcal{L}_t) \setminus \mathcal{L}$. Finally, we let $\mathcal{M}$ be the set of indices of connected components of F^0 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 $1 \le l < q \le p$.

COROLLARY 4.3. *Define* $\mathtt{b}^0_{\mathcal{L}} = \mathtt{b}^0_{l,t} + \mathtt{b}^0_{q,s}$ *if* $\mathcal{L} = \{l,q\}$ *for some* $1 \le l < q \le p$. *Otherwise let* $\mathtt{b}^0_{\mathcal{L}} = 0$. *Also let* $h_{\mathcal{L}} = h_{l,s} + h_{q,t}$. *We then must have*

$$\mathtt{b}^0_{\mathcal{L}} + \sum_{i \in \mathcal{C} \cup \mathcal{M}} (\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t}) \le \widetilde{c}(P) + h_{\mathcal{L}}.$$

4.2.3 A general lower-bound for χ_{st} We are finally ready to present the general lower bound on χ_{st}.

LEMMA 4.9. *Let $\mathcal{I}$ be the set of indices of components on P that contain terminals that interfere with s or t, i.e.*

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

Also define $\overline{\mathtt{b}}^0_{i,u} = \min\{\mathtt{b}^0_{i,u}, 2/(\beta-2) \cdot h_{i,u}\}$ *and let* $\overline{\mathtt{b}}_{i,u} = \overline{\mathtt{b}}^0_{i,u} + h_{i,u}$ *for all* $1 \le i \le p$ *and for* $u \in \{s,t\}$. *We must have*

$$\chi_{st} \ge \frac{1}{2} \cdot \left(\mathtt{sl} + \sum_{1 \le i \le p, i \notin \mathcal{I}} (\overline{\mathtt{b}}^0_{i,s} + \overline{\mathtt{b}}^0_{i,t}) \right) \tag{4.11}$$

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

$$\mathtt{sl} = \max\{0, \widetilde{c}(P) + \sum_{i \in \mathcal{I}} (h_{i,s} + h_{i,t}) - \sum_{1 \le i \le p, i \notin \mathcal{I}} (\overline{\mathtt{b}}^0_{i,s} + \overline{\mathtt{b}}^0_{i,t})\}.$$

Proof. We know from Lemma 4.6 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. Lemma 4.6 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 $T_u \le \mathtt{age}_{s,t}$ be the time in $\mathtt{AKR}(R)$ when u meets the first interfering component $C \in \mathcal{C}'[C_m]$. Lemma 4.1 shows that

$$h_{m,u} \le T_u \tag{4.12}$$

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

Let p be the point on P^u_m where u and C_m meet in $\mathtt{AKR}(R)$ and use P_{up} and P_{pm} to denote the u,p-segment and the p, C_m-segment of P^u_m, respectively. Definition (4.5) implies that the residual cost of P_{pm} is 0. We therefore obtain

$$\widetilde{c}(P^u_m) = \widetilde{c}(\langle P_{up}, P_{pm} \rangle) = \widetilde{c}(P_{up}) = h_{m,u}.$$

Together with (4.12) this implies that

$$\chi_{st} \ge T_s + T_t \ge h_{m,s} + h_{m,t} = \widetilde{c}(P^s_m) + \widetilde{c}(P^t_m) = \widetilde{c}(P). \tag{4.13}$$

As in Corollary 4.3 we let $\mathcal{C}$ be the index set of components that capture either s or t excluding m. We also let $\mathcal{M}$ be the set of indices of components on P that do not capture s and t. Corollary 4.3 implies that

$$\widetilde{c}(P) + h_{m,s} + h_{m,t} \ge \sum_{i \in \mathcal{C} \cup \mathcal{M}} (\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t}) \ge \sum_{i \in \mathcal{C} \cup \mathcal{M}} (\overline{\mathtt{b}}^0_{i,s} + \overline{\mathtt{b}}^0_{i,t}).$$

The definition of $\mathtt{sl}$ together with (4.13) imply

$$\chi_{st} \ge \frac{\widetilde{c}(P) + h_{m,s} + h_{m,t}}{2} = \frac{1}{2} \cdot \left(\mathtt{sl} + \sum_{i \in \mathcal{C}} (\overline{\mathtt{b}}^0_{i,s} + \overline{\mathtt{b}}^0_{i,t}) \right)$$

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' that contain vertices v_s and v_t.

We observe that the cost-share collected by $\{s,t\}$ is smallest if there are interfering terminals. Corollary 4.2 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 $C_s \neq C_t$ 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 from older terminal pairs.

[Case 1: *Two-sided interference*] Let T_s be the time when s meets v_s in $\mathtt{AKR}(R)$ and define T_t analogously for t and v_t. Let P_{v_s} and P_{v_t} be the paths that are added in $\mathtt{AKR(R)}$ when v_s and s meet and when v_t and t meet, respectively. Lemma 4.5 shows that the combined load from C_s and C_t on $\langle P_{v_s}, P, P_{v_t} \rangle$ is at least

$$(4\gamma + 2) \cdot \mathtt{age}_{s,t} \ge (2\gamma + 1) \cdot \widetilde{c}(P). \tag{4.14}$$

Define sets $\mathcal{L}_u$ for $u \in \{s,t\}$ as in Corollary 4.3 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. Then C_i captures u by time $2 \cdot \mathtt{age}_{s,t}$ in the anaerobic phase: Let $\mathcal{C}$ contain the indices of all sets on P^u_i excluding i. Notice that all components C_j for $j \in \mathcal{C}$ must be dead by the time C_i captures u. Hence, the maximal load that C_i can inflict on P_i^u is bounded by

$$c_{G|F^0}(P_i^u) - \sum_{j\in\mathcal{C}}(\mathtt{b}_{j,s} + \mathtt{b}_{j,t}) \leq \widetilde{c}(P_i^u) + h_{i,u} \leq 2\cdot \mathtt{age}_{s,t}.$$

This shows that C_i captures u by time $2\cdot\mathtt{age}_{s,t}$ in the anaerobic phase. Let $C \in \{C_s, C_t\}$ be the first set to reach C_i in the anaerobic phase of $\mathtt{AKR}_2(R^0)$. The above argument shows that the collision of C_i and C must happen before time $(2\gamma+1)\cdot\mathtt{age}_{s,t}$. It follows that C_i must be dead at this time since otherwise C would be on path P.

Hence component C_i must have extended fully in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ for all $1 \leq i \leq p$ before either C_s or C_t reach it in the anaerobic phase. A careful look at Lemma 4.5 shows that the load in (4.14) is inflicted before time $(2\gamma+1)\cdot\mathtt{age}_{s,t}$ in the anaerobic phase and thus, both C_s and C_t are active at this time.

Therefore the load in (4.14) has to be at most

$$c_{G|F^0}(P) + 2T_s + 2T_t - \sum_{i=1}^{p}(\mathtt{b}_{i,s} + \mathtt{b}_{i,t}) = \widetilde{c}(P) + 2T_s + 2T_t - \sum_{i=1}^{p}(\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t}).$$

Solving for $T_s + T_t$ gives

$$T_s + T_t \geq \frac{1}{2}\cdot\widetilde{c}(P) + \frac{1}{2}\cdot\sum_{i=1}^{p}(\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t}).$$

Now observe that $\chi_{st} \geq T_s + T_t$ and hence

$$\chi_{st} \geq \frac{1}{2}\cdot\widetilde{c}(P) + \frac{1}{2}\cdot\sum_{i=1}^{p}(\mathtt{b}^0_{i,s} + \mathtt{b}^0_{i,t}).$$

This 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 T_s denote the time when s meets the first interfering terminal v_s in $\mathtt{AKR}(R)$. Since t sees no interference in $\mathtt{AKR}(R)$, the proof of Lemma 4.4 implies that

$$\chi_{st} \geq \mathtt{age}_{st} + T_s = \frac{1}{2}\cdot\widetilde{c}(P) + T_s. \tag{4.15}$$

We again let C_s be the connected component of F' that captures s. Let C_i for $1 \leq i \leq p$ be a connected component of F^0 on path P. Component C_i must be dead when C_s captures it during the anaerobic phase of $\mathtt{AKR}_2(R^0)$ 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 $1 \leq l \leq q \leq p$. Consider the case where $l < q$ and hence $\mathcal{L}$ contains exactly two indices. Observe that C_l captures s by time $2\cdot\mathtt{age}_{s,t}$ in this case. Otherwise C_l would also capture C_q and this contradicts the assumption $l \neq q$. The budget of C_s is at least $(2\gamma+1)\cdot\mathtt{age}_{s,t} \geq 2\cdot\mathtt{age}_{s,t}$ and therefore C_s reaches s by time $2T_s \leq 2\cdot\mathtt{age}_{s,t}$ as well. This means that C_s captures C_l and C_l must be dead at that time.

Lemma 4.8 implies that

$$\sum_{i=q+1}^{p}(\overline{\mathtt{b}}_{i,s} + \overline{\mathtt{b}}_{i,t}) \leq c_{G|F^0}(P_q^t). \tag{4.16}$$

Let P_{v_s} be the path that is added in $\mathtt{AKR}(R)$ 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 C_s meets it in $\mathtt{AKR}_2(R^0)$ and therefore, C_s inflicts at least $\mathtt{b}_q$ units of budget on path P'. The total load coming from super-nodes contained in sets $\{C_i\}_{1\leq i\leq q}$ and from C_s on path P' is bounded by $c_{G|F^0}(P_q^s) + 2T_s$. These observations imply

$$\sum_{i=1}^{q}(\overline{\mathtt{b}}_{i,s} + \overline{\mathtt{b}}_{i,t}) \leq c_{G|F^0}(P_q^s) + 2T_s. \tag{4.17}$$

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

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

in the anaerobic phase. In other words, C_s and C_q do not touch at time $(2\gamma+1)\cdot\mathtt{age}_{s,t}$ in the anaerobic phase of $\mathtt{AKR}_2(R^0)$ and hence

$$2\cdot(2\gamma+1)\cdot\mathtt{age}_{s,t} + \sum_{1\leq i<q}(\overline{\mathtt{b}}_{i,s} + \overline{\mathtt{b}}_{i,t}) \leq c_{G|F^0}(P_q^s) + 2T_s. \tag{4.18}$$

For $u \in \{s,t\}$, the definition of $\overline{\mathtt{b}}_{q,u}$, our choice of $\gamma \geq 1/(\beta-2)$ in Theorem 1.1, and Lemma 4.1 imply that

$$\overline{\mathtt{b}}_{q,u} \leq (2\gamma+1)\cdot h_{q,u} \leq (2\gamma+1)\cdot\mathtt{age}_{s,t}.$$

Together with (4.18) we then obtain

$$\sum_{1\leq i\leq q}(\overline{\mathtt{b}}_{i,s} + \overline{\mathtt{b}}_{i,t}) \leq c_{G|F^0}(P_q^s) + 2T_s. \tag{4.19}$$

Inequalities (4.16), (4.17), and (4.19) imply that $\sum_{i=1}^{p}(\overline{\mathtt{b}}_{i,s} + \overline{\mathtt{b}}_{i,t}) \leq c_{G|F^0} + 2T_s$ and hence

$$\sum_{i=1}^{p}(\overline{\mathtt{b}}^0_{i,s} + \overline{\mathtt{b}}^0_{i,t}) \leq \widetilde{c}(P) + 2T_s.$$

It can be seen that (4.15) together with the definition of slack $\mathtt{sl}$ implies

$$\chi_{st} \geq \frac{1}{2} \cdot \left(\sum_{i=1}^{p} (\overline{\mathtt{b}}_{i,s}^{0} + \overline{\mathtt{b}}_{i,t}^{0}) \right) + \frac{\mathtt{sl}}{2}$$

and the lemma follows.

Equation (4.11) in Lemma 4.9 shows that we obtain $(\overline{\mathtt{b}}_{i,s}^{0} + \overline{\mathtt{b}}_{i,t}^{0})$ units of cost-share for each component C_i with $i \in \{1, \ldots, p\} \setminus \mathcal{I}$. For each such $i \in \{1, \ldots, p\} \setminus \mathcal{I}$ we are going to use this amount of cost-share to pay for a stretch of length $\overline{\mathtt{b}}_{i,s}^{0} + h_{i,s} + \overline{\mathtt{b}}_{i,t}^{0} + h_{i,t}$ along path P. In particular, this way we pay for a total of

$$\sum_{1 \leq i \leq p, i \notin \mathcal{I}} (\overline{\mathtt{b}}_{i,s}^{0} + \overline{\mathtt{b}}_{i,t}^{0}) \tag{4.20}$$

of the residual cost of path P. The slack $\mathtt{sl}$ in Lemma 4.9 is the difference between the residual cost of P and (4.20). A negative difference indicates that all of the residual cost is paid for by $\sum_{1 \leq i \leq p, i \notin \mathcal{I}} (\overline{\mathtt{b}}_{i,s}^{0} + \overline{\mathtt{b}}_{i,t}^{0})$ and we therefore define the slack to be 0 in this case. We are now ready to prove Theorem 1.1 which we restate for completeness.

THEOREM 4.2. *For any $\beta \geq 2$ there is a polynomial-time $(2 + 2/(\beta - 2))$-approximate and β-strict algorithm for the minimum-cost Steiner forest problem.*

Proof. We assume that there exist terminals that interfere with $\{s, t\}$. Notice that this assumption is w.l.o.g. since the presence of interfering terminals can only lower the cost-share χ_{st}. Now recall the definition of slack in Lemma 4.9 and observe that the cost $c_{G|F^0}(P)$ of path P is at most

$$\left(\sum_{1 \leq i \leq p, i \notin \mathcal{I}} (\overline{\mathtt{b}}_{i,s}^{0} + h_{i,s} + \overline{\mathtt{b}}_{i,t}^{0} + h_{i,t}) \right) + \mathtt{sl}.$$

On the other hand Lemma 4.9 yields that the cost-share collected by (s, t) is at least

$$\frac{1}{2} \cdot \left(\sum_{1 \leq i \leq p, i \notin \mathcal{I}} (\overline{\mathtt{b}}_{i,s}^{0} + \overline{\mathtt{b}}_{i,t}^{0}) \right) + \frac{\mathtt{sl}}{2}.$$

We clearly have $\mathtt{sl} \leq \beta \cdot (\mathtt{sl}/2)$ as $\beta \geq 2$. In order to complete the proof it suffices to show $\overline{\mathtt{b}}_{i,u}^{0} + h_{i,u} \leq \frac{\beta}{2} \cdot \overline{\mathtt{b}}_{i,u}^{0}$ for all $1 \leq i \leq p, i \notin \mathcal{I}$ and for $u \in \{s, t\}$. Equivalently we need to have $\overline{\mathtt{b}}_{i,u}^{0} \geq 2/(\beta - 2) \cdot h_{i,u}$. This follows from the definition of $\overline{\mathtt{b}}_{i,u}^{0}$ in Lemma 4.9 and from Lemma 4.2 with $\gamma \geq 1/(\beta - 2)$. Our final choice of $\beta = 2 + \sqrt{2}$ also ensures that $\gamma \geq 1/2$ as wanted.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM J. Comput.*, 24:440–456, 1995.

[2] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 542–547, 20–22 October 1997.

[3] Yair Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings, ACM Symposium on Theory of Computing*, pages 161–168, May 23–26 1998.

[4] L. Becchetti, J. Könemann, S. Leonardi, and M. Pál. Sharing the cost more efficiently: Improved approximation for multicommodity rent-or-buy. See `www.math.uwaterloo.ca/~jochen/docs/mrob-jv.pdf`, 2004.

[5] Feigenbaum, Papadimitriou, and Shenker. Sharing the cost of multicast transmissions. *JCSS: Journal of Computer and System Sciences*, 63, 2001.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness.* W. H. Freeman and Company, San Francisco, 1979.

[7] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24:296–317, 1995.

[8] A. Gupta, A. Kumar, M. Pal, and T. Roughgarden. Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, 2003.

[9] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings, ACM Symposium on Theory of Computing*, pages 365–372, 2003.

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

[11] J. Könemann. *Approximation Algorithms for Minimum-Cost Low-Degree Subgraphs.* PhD thesis, Carnegie Mellon University, 2003.

[12] J. Könemann and R. Ravi. Quasi-polynomial time approximation algorithm for low-degree minimum-cost steiner trees. In *Proceedings of Foundations of Software Technology and Theoretical Computer Science.*, 2003.

[13] Amit Kumar, Anupam Gupta, and Tim Roughgarden. A constant-factor approximation algorithm for the multicommodity. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 333–344, 2002.

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