Pricing Tree Access Networks with Connected Backbones

Vineet Goyal [*] Anupam Gupta [†] Stefano Leonardi [‡] R. Ravi [§]

Abstract

Consider the following *network subscription pricing* problem. We are given a graph $G = (V, E)$ with a root r, and potential customers are companies headquartered at r with locations at a subset of nodes. Every customer requires a network connecting its locations to r. The network provider can build this network with a combination of *backbone* edges (consisting of high capacity cables) that can route any subset of the customers, and *access* edges that can route only a single customer's traffic. The backbone edges cost M times that of the access edges. Our goal is to devise a *group-strategyproof* pricing mechanism for the network provider, i.e., one in which truth-telling is the optimal strategy for the customers, even in the presence of coalitions. We give a pricing mechanism that is 2-competitive and $O(1)$-budget-balanced.

As a means to obtaining this pricing mechanism, we present the first primal-dual 8-approximation algorithm for this problem. Since the two-stage Stochastic Steiner tree problem can be reduced to the underlying network design, we get a primal-dual algorithm for the stochastic problem as well. Finally, as a byproduct of our techniques, we also provide bounds on the inefficiency of our mechanism.

1 Introduction

Consider the following connected backbone for tree access network (CBTAN) design problem: given an undirected graph $G = (V, E)$ with metric costs $c(e)$ on the edges, and a root r, we want to build a network to connect a set of l possible *customers* U. The i^{th} customer is specified by a set $S_i \subseteq V$ of terminals. A solution to the problem is a set of connected backbone edges E_0 containing the root r and a set of access networks E_i one for each customer i such that $E_0 \cup E_i$ contains a Steiner tree connecting $S_i \cup \{r\}$ for all i. Backbone edges E_0 are a factor M costlier than the access edges E_i. The total cost to connect any subset $U' \subseteq U$ of customers is $Mc(E_0) + \sum_{i \in U'} c(E_i)$. Note that the objective of minimizing the total network cost naturally trades off backbone and access network costs.

The above problem is equivalent to the rooted two-stage stochastic Steiner tree problem (StocST) [IKMM04, GPRS04, GRS04] where the customers correspond to *scenarios* and the backbone network corresponds to the first-stage tree. In line with this analogy, we refer to customers as *scenarios*. Also, we refer to the nodes connected by the backbone edges E_0 to the root r as *backbone nodes* or *facilities*. Our problem generalizes the problem of network design for information networks defined by Hayrapetyan et al. [HST05] by imposing connection between facilities. The SROB network design problem [SK04, GKR03, PT03] can also derived from our problem if every scenario is a single terminal.

[*]Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: vineet@cmu.edu. Supported in part by NSF ITR grant CCR-0122581 (The ALADDIN project) and NSF grant CCF-0430751.

[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: anupamg@andrew.cmu.edu.

[‡]Dipartimento di Informatica e Sistemistica, University of Rome "La Sapienza", Via Salaria 113, 00198 Rome, Italy. Email: leon@dis.uniroma1.it. Part of this work was done while visiting the School of Computer Science at Carnegie Mellon University.

[§]Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: ravi@cmu.edu. Supported in part by NSF ITR grant CCR-0122581 (The ALADDIN project) and NSF grant CCF-043075.

In this paper, we are interested in a *game-theoretic extension* of the problem that is perhaps best described as a *subscription pricing* problem: suppose we have a set U of l customers, where customer i's ultimate goal is to connect the set S_i to the root r, and she derives a (privately-held) *utility* $u_i \geq 0$ from being connected to the root. We can sell *subscriptions* to the potential customers such that a subscription-holding customer is guaranteed connectivity of her set to the root. The goal is to price these subscriptions for potential customers in such a way that the sale of these subscriptions to some subset of customers yields enough money (up to constant factors) to pay for the cost of the network serving these subscription-holding customers.

Formally, we are interested in finding a *cost-sharing mechanism* that determines *group-strategyproof* subscription prices ξ_i for each customer i in U. Group-strategyproofness implies that reporting their true utility u_i as their bid should be a dominant strategy for each customer, and the customers should have no incentive to indulge in strategic behavior *even when they are allowed to collude*. The mechanism solicits bids $\{b_i\}_{i \in U}$ from all customers and commits to serve the customers in U iff $\xi_i \leq b_i, i \in U$. If there is a customer whose bid is less than its subscription price (as determined by the mechanism), the process is repeated after removing all the customers whose bid is lower than their subscription price.

Of course, making the subscriptions free would ensure truthfulness; to avoid such degenerate solutions, we would like to ensure other desirable properties. E.g., a mechanism is *budget balanced* if the actual cost $C(S)$ of servicing the customers in S is at most the sum of the subscription costs for the customers in S—i.e., we recoup our costs by selling the subscriptions. (In this paper, we focus on α-budget balance, where we only recover an $1/\alpha$ fraction of $C(S)$; hence budget-balance is the same as 1-budget-balance.) A mechanism is *competitive* if the sum of subscription costs to the customers in S does not exceed the cost of an optimal solution for S. A mechanism is called *efficient* if it selects a set S of customers that maximizes the *efficiency* $u(S) - C(S)$.

Classical results in economics [GKL76, Rob79] state that budget balance and efficiency cannot be simultaneously achieved by any mechanism. Moreover, Feigenbaum et al. [FPS01] recently showed that there is no group-strategyproof mechanism that always recovers a constant fraction of the maximum efficiency and a constant fraction of the incurred cost even for the simple fixed-tree multicast problem. In light of these impossibility results, previous work on mechanism design usually focused on a proper subset of the above desiderata. One class of such mechanisms are based on a framework of Moulin and Shenker [MS01], who show that given an α-budget balanced and *cross-monotonic cost sharing method* for the underlying problem, the natural *Moulin mechanism* [Mou99] satisfies both α-budget balance and group-strategyproofness. Cross-monotonicity imposes that the cost-share computed by the mechanism for each player only decreases if more players join the game. (Formal definitions are deferred to Section 2.) Moulin and Shenker's framework has recently been applied to game-theoretic variants of numerous classical optimization problems, and we will also seek cross-monotone cost-shares for our network design problem to solve the subscription pricing problem.

Contributions. Our main result is the following:

Theorem 1.1 *There is a cross-monotonic cost sharing scheme for* CBTAN *that is 2-competitive and 41-budget-balanced.*

As in several previous papers giving cost-shares, we derive these cost shares from a primal-dual algorithm for the CBTAN problem. However, rather surprising, no primal-dual algorithm was previously known for this problem. We obtain the first primal-dual algorithm for CBTAN and StocST that achieves an 8-approximation for these problems; due to space constraints, this algorithm is deferred to Appendix A.

The duals generated in such a primal-dual algorithm naturally give us cost-shares that are competitive and approximately budget-balanced. However, they are not cross-monotone and hence, we have to work harder to achieve this property. We are able to extend the results of Pál and Tardos on SROB network design [PT03] to obtain cross-monotone cost shares for this more general setting; the details of this process (and hence the proof of the above theorem) appear in Section 3.

Very recently, Roughgarden and Sundararajan [RS06] introduced an alternative measure of efficiency that can be approximated at the same time of budget balance. In Section 4, we present results on the approximated efficiency achieved by our mechanism.

Related Work Approximation algorithms for a variant of the CBTAN problem where the access network involves direct edges to the backbone nodes have been well-studied [RS99, GKK$^+$01, SK04, KM00]. Other variants where no connectivity is sought among the backbone nodes is studied in [AZ02].

The Stochastic Steiner Tree problem which is equivalent to CBTAN has been previously studied by [IKMM04, GPRS04, GRS04] and constant-factor approximations based on randomized selection strategies are known; however, no primal-dual algorithms were known for the problem.

See Moulin and Shenker [MS01] for a study of group-strategyproof mechanisms and how to use cross-monotonic cost sharing methods to design such mechanisms using the Moulin mechanism [Mou99]. This work has given game-theoretic variants of problems like fixed-tree multicast [AFK$^+$04, FKSS03, FPS01], submodular cost-sharing [MS01], Steiner trees [JV01, KSK96], facility location [PT03], single-source rent-or-buy network design [PT03, LS04, GST04], and Steiner forests [KLS05]. Lower bounds on the budget balance achievable by cross-monotonic cost shares are given in [IMM05, KLSvZ05].

2 Preliminaries on Cost Sharing Methods

A *cost sharing method* ξ for a problem Π is an algorithm that, given any subset $S \subseteq U$ of players demanding service, computes a solution for the set S; moreover, it computes a non-negative cost share $\xi_i(S)$ for each player $i \in S$. The following properties of cost-sharing methods will be useful.

Definition 2.1 *We say that the cost sharing method ξ is β-*budget balanced *if for every subset $S \subseteq U$,*

$$\frac{1}{\beta} \cdot C(S) \leq \sum_{i \in S} \xi_i(S) \leq C(S).$$

A cost sharing method ξ is cross-monotonic *[MS01] if for any two sets S and T such that $S \subseteq T$ and any player $i \in S$ we have $\xi_i(S) \geq \xi_i(T)$; i.e., the cost shares of a player never increase if more players enter the game.*

Moulin and Shenker [MS01] showed that, given a budget balanced and cross-monotonic cost sharing method ξ for the underlying problem, the following cost sharing mechanism $M(\xi)$ (henceforth known as the *Moulin mechanism*) satisfies budget-balance and group-strategyproofness: *initially, let $S \rightarrow U$. If for each player $i \in S$ the cost share $\xi_i(S)$ is at most her bid b_i, we stop. Otherwise, remove from S all players whose cost shares are larger than their bids, and repeat.* Eventually, let $\xi_i(S)$ be the costs that are charged to players in the final set S.

3 A cross-monotonic cost-sharing scheme for CBTAN

In this section, we develop a cross-monotonic cost-sharing method that is competitive and budget balanced for CBTAN. The algorithm in this section can be perceived a substantial extension of the one in Appendix A, where instead of running one primal-dual process, we run *an extra dual process* called, as in [PT03], the *ghost process*—this is a monotone process used to generate the cost shares; the heart of the argument is in relating the real and ghost processes to each other and arguing that the cost shares generated by the ghost process are enough to pay for the actual network created.

3.1 The Real and Ghost Processes

Note that if the we fix the set of connected backbone edges E_0, the access edges for each customer or scenario can be found relatively easily (say, by using a MST heuristic to connect to the backbone) to complete the solution. Thus, the problem essentially reduces to finding a low cost set of backbone edges such that there is a low cost of completion (set of access edges) for each scenario. Recall that the nodes connected to the root by the backbone edges are referred to as facilities or backbone nodes. Finding the set of backbone edges is equivalent to finding the set of facilities that are connected to the root by the backbone edges. The access edges of each scenario will form a Steiner forest on the terminals of the scenario, each tree of the forest containing at least one backbone node. Thus, we would consider the CBTAN problem as one of finding facilities that are connected to the root through the backbone edges.

We describe a *ghost process*, which is similar to the dual ascent schema of [AKR95], to construct the solution and cost shares for all the scenarios. It is similar in spirit to the idea of ghost process developed by Pál and Tardos in [PT03] for SROB, where, the ghost of each terminal j is a ball with center j and growing uniformly to infinity. However, unlike the mechanism for SROB where each scenario terminal has a direct connection to some open facility, our ghost process has to assign cost share for building Steiner tree connections to open facilities. This is done by integrating the ghost process with l separate dual ascent Steiner forest processes [AKR95]. For simplicity, we maintain $l+1$ different copies $(G_0, G_1, \ldots, G_l)$ of the graph G. Copies $G_1, \ldots, G_l$ correspond to the l scenarios and copy G_0 corresponds to the open facilities. Initially, every singleton terminal of scenario i is an active component in the copy G_i.

During the course of the algorithm, we would open some locations in G_0 as *tentative facilities* after M or more dual ascent processes in the other copies have made the location tight (in their respective dual packing constraints). Such a location is a feasible location to open a facility as it has clustered M different scenario demands. For simplicity, we assume that a facility can be opened anywhere along an edge. We can easily remove this assumption at no additional cost.

We open a *real facility* at a tentative location j only if there is no real open facility within a distance $4t_j$ from j, where t_j is the time when j was declared tentatively open. We define a corresponding ghost process in copy G_0 of the graph, where we run a dual ascent process on tentatively open facilities. Each tentatively open facility p becomes an active component in G_0 at the instant it is declared open, say t_p.

Definition 3.1 (Tentative Facility Moats) *We call the components in G_0 as the* tentative facility moats. *The ghost of a tentatively open facility p opened at time t_p is defined for any time $t \geq t_p$ as a ball $\mathcal{B}(p, t-t_p)$ of radius $t - t_p$ around vertex p. Tentative facility moats in G_0 are therefore the union of ghost components of different radii.*

Definition 3.2 ((Ghost) Scenario Moats) *In each scenario graph G_i, at any time t, we define a collection of active components also called* ghost scenario moats *or just* scenario moats. *Every terminal in each scenario has a ghost at any time $t \geq 0$, which is a ball $\mathcal{B}(v, t)$ of radius t around vertex v. As the time grows, a ghost of scenario i can eventually collide with (i) either another ghost of scenario i (in G_i), or (ii) a tentatively facility moat (in G_0), to merge into a single active component. The set of scenario moats of scenario i at time t is the set of disjoint active components of scenario i in G_i.*

Definition 3.3 (Dark and Lit Moats) *We call a scenario moat* dark *if it does not contain any open facility (tentative or real) and* lit *if it contains at least one tentatively or real open facility.*

Initially all the scenario moats are dark. The ghost process results in one of the following events:

Events in the Ghost Process

1. Two dark scenario moats C and C' intersect in some copy G_i of the graph. The two moats merge to form a new dark scenario moat $C \cup C'$.

2. For some location $j \in V$, at least M scenario moats of different scenarios (i.e. M moats in different copies of the graph) contain j for the first time.

   (a) Declare j as a tentatively open facility. The singleton terminal j becomes an active component in G_0.

   (b) All the scenario moats containing j are declared "lit".

   (c) If there is no real open facility within a distance $4t_j$ (t_j is the current time) from j, then open a real facility at j.

3. A dark scenario moat C intersects a lit moat C' in some copy G_i of the graph. The two moats merge to form a new lit moat.

4. Two lit scenario moats C and C' intersect in some copy G_i. The two moats merge to form a new lit moat.

5. A scenario moat C (dark or lit) of some scenario $i \neq 0$ intersects a tentative facility moat $\mathcal{F}$ in G_0. Declare the scenario moat C lit if it was dark and merge C with $\mathcal{F}$. Thus, the new lit moat in G_i is $C \cup \mathcal{F}$.

We continue this ghost process until every scenario moat contains the root. The ghost process described above lets us decide the cost shares for each terminal and also determines where to open real facilities.

Network Design Algorithm

- Build a Steiner forest E_i for each scenario i, that connects terminals in scenario i to closest real facilities (for each component).
- Build a Steiner tree (of backbone edges) over the real facilities connecting them to the root.

3.2 Defining the Cost Shares

We now describe the cost shares that are collected by terminals of all the scenarios during the ghost process. We assign two kinds of cost shares to every terminal: **(a)** one when it is a part of a dark scenario moat, and **(b)** another when it becomes a part of a lit scenario moat. Let us define the two cost shares for a terminal j in scenario i. Let $C_j(t)$ be the scenario moat containing j at time t and t_j^1 be the first time instant when j is contained in a lit scenario moat and let t_j^2 be the time when the moat containing j reaches the root. Thus, cost share for j till t_j^1 is defined as:

$$f_j^1 = \int_{t=0}^{t_j^1} \frac{1}{|C_j(t)|} dt$$

Here $|C_j(t)|$ denotes the number of terminals in the scenario moat $C_j(t)$ that divide up the cost share accumulated as dual by this growing moat. For $t \geq t_j^1$, j is in a lit moat $C_j(t)$.

Definition 3.4 *We say that the moat $C_j(t)$ contributes to a tentative facility moat $\mathcal{M}$ if there exists a terminal $k \in C_j(t)$ which is at a distance at most t from the moat $\mathcal{M}$.*

Note that $C_j(t)$ could possibly contribute to many facility moats. Suppose $C_j(t)$ contributes to moats $\mathcal{M}_1, \ldots, \mathcal{M}_l$ and let n_i be the number of different scenarios contributing to moat $\mathcal{M}_i$ at time t. Also, let $n_{C_j(t)} = \max_{i=1,2,\ldots,l} n_i$. The cost share for the terminal j is:

$$f_j^2 = \int_{t=t_j^1}^{t_j^2} \frac{M}{|C_j(t)| \cdot n_{C_j(t)}} dt \tag{3.1}$$

3.3 Properties of the Cost Shares

We need to prove that the cost shares defined above are competitive, cross-monotone and budget balanced. To prove competitiveness, we construct a feasible dual for the CBTAN problem from the cost shares of the terminals. Since, a feasible dual is a lower bound on the optimum cost, it proves that cost shares are competitive (approximately). The cross-monotonicity property follows from the description of ghost process. The crucial part is proving that cost shares are budget-balanced. In other words, the cost shares of the terminals can pay for the cost of the network constructed by our algorithm. Charging the cost of access networks (Steiner forest E_i) for each scenario to the cost shares collected by the terminals of that scenario is not very difficult and follows standard primal-dual arguments [AKR95]. However, proving that the total cost shares of all terminals are sufficient to pay for the Steiner tree over the real facilities is challenging and requires new ideas and charging techniques. In the following lemmas, we will prove the desired properties for the cost shares.

Lemma 3.5 *The cost shares ($f_j^1 + f_j^2$ of terminal j) defined by the dual ascent process are 2-competitive i.e. $\sum_{k=1}^{l} \sum_{j \in S_k} (f_j^1 + f_j^2) \leq 2OPT$, where OPT is the optimal cost of the network.*

Proof: We need to prove that the total cost shares of all the terminals is at most two times the optimal solution. We will show that half times the cost shares form a feasible dual. Consider a moat C at time t of scenario i. If C is a dark scenario moat at time t, the dual $\beta_{C,i}$ increases at a rate half, i.e. $\frac{d}{dt}\beta_{C,i} = \frac{1}{2}$. If C is a lit scenario moat at time t, then $\frac{d}{dt}\beta_{C,i} = \frac{M}{2n_{C(t)}}$. Here, we assume that there are locations at each point along every edge and dt is an infinitesimal amount of time. Clearly, the individual scenario constraint for edge packing is never violated. Consider the following dual constraint:

$$\sum_{i=1}^{l} \sum_{S: \phi \neq S \cap S_i \neq S_i, e \in \delta(S)} \beta_{S,i} \leq M \cdot c_e \quad \forall e \in E, \forall i = 1, \ldots, l \tag{3.2}$$

When an edge e is dark, i.e. no tentative facility has been opened on any location on e, each scenario moat collects cost share at a rate 1. Thus, total dual collected by moats which e crosses during the time it was dark is at most $\frac{(M-1)c_e}{2}$, because at most $M-1$ scenarios components can cross e while it is dark. When it becomes lit, the total dual collected by all moats that e crosses after this instant of time is at most $\frac{Mc_e}{2}$. Thus, the above constraint is not violated by the scaled dual. Thus, we have that $\sum_{j \in V} \frac{f_j^1 + f_j^2}{2} \leq$ OPT or that $\sum_{j \in V} (f_j^1 + f_j^2) \leq$ 2OPT. ■

Lemma 3.6 *The cost share $f_j^1 + f_j^2$ for any terminal j is cross-monotone.*

Proof: Suppose a new terminal j' is added in scenario k. The moats in the ghost process for the new instance are a superset of the moats in the original instance. Thus, for any other terminal j in scenario k, j collects cost share at a smaller rate in the new instance as compared to the original instance at any point of time. For a terminal j in scenario $k' \neq k$, the new terminal j' can only decrease the rate at which j collects cost share after it becomes a part of a lit moat. Thus, addition of a new terminal decreases the cost shares for all other terminals which implies $f_i^1 + f_i^2$ is a cross-monotone cost sharing function. ■

Budget balance. The proof of budget balance proceeds in two parts. In the first part, we prove that the cost shares f^1 of terminals are enough to connect terminals of a scenario to a real open facility. This is proved via a standard argument in the following lemma.

Lemma 3.7 *For any scenario i, we can build a Steiner forest over terminals in S_i such that each Steiner component is connected to some open facility and the cost of the Steiner forest is at most $8\sum_{j \in S_i} f_j^1$.*

Proof: Initially the ghost process for terminals of a scenario is exactly similar to the dual ascent process of the Steiner tree algorithm of [AKR95]. The cost share collected by each moat is exactly same as the dual collected in the dual ascent process as long as the moat is dark. Thus, consider the time instant (say time t) when a dark moat $\mathcal{M}$ becomes lit. At this instant, the cost shares of the terminals can pay for building a Steiner tree over them within a factor of 2 [AKR95]. The moat can become lit due to one of the following events:

1. Some location l in the moat is declared tentatively open at time t. In this case, either l becomes a real open facility or there is a real open facility within a distance $4t$ from l. Thus, the Steiner component can be connected to a real open facility by paying a cost at most $4t$. Since, the moat was dark till time t, the cumulative cost share collected by the terminals in the moat is at least t. Thus, cost of the Steiner component connecting all the terminals in the moat $\mathcal{M}$ to an open facility is at most $6\sum_{j \in \mathcal{M}} f_j^1$.

2. Moat $\mathcal{M}$ meets another lit moat of the same scenario. We can build a Steiner tree over the terminals in $\mathcal{M}$ as in the previous case. Moreover, in this case, there exists a terminal j' in the lit moat which is at a distance of at most $2t$ from some terminal in $\mathcal{M}$. Thus, terminals in $\mathcal{M}$ can get connected to an open facility by connecting to the terminal j'. The cost in this case is at most $4\sum_{j \in \mathcal{M}} f_j^1$.

3. Moat $\mathcal{M}$ meets a tentative facility moat. Thus, there exists a tentative facility l which is at a distance of at most $2t$ from some terminal in moat $\mathcal{M}$ which implies that the closest real open facility is at a distance of at most $6t$ from some terminal in $\mathcal{M}$. Thus, the cost of building a Steiner tree on the terminals in $\mathcal{M}$ and connecting them to a real open facility is at most $8\sum_{j \in \mathcal{M}} f_j^1$.

■

In the second part, we prove that the cost shares can pay for building a Steiner tree over the open facilities. This is more difficult part of the proof and is proved over the following series of lemmas. For the sake of the analysis, we consider the Steiner tree algorithm over real facilities being run in parallel to the ghost process.

Note that after a scenario moat becomes lit, it collects cost share at a rate that is less than 1. This may not be a sufficient to pay for Steiner connections between real facility moats, whose cost is M times the cost of the connection. In this case, however, we argue that the cost share collected by the scenario moats at a time $t' \leq t$ is sufficient to pay for the share requested by real facility moats at time $5t$.

We charge the cost of the Steiner connections between real facilities moats to a *merge tree* over the dark and the lit moats of each scenario. The merge tree is a virtual tree which we construct during the ghost process. Each edge e in the merge tree has an association fraction $f(e)$ which is decided during the ghost process. $f(e)$ is the fraction of the cost of e which can be paid by the cost shares of the terminals within a constant factor.

Merge Tree. To construct the merge tree for scenario i, we consider a slightly modified view of the ghost process in copy G_i of the graph corresponding to scenario i. Suppose a lit moat $\mathcal{M}$ intersects with a tentative facility moat F at time t in the ghost process. Recall that we merge $\mathcal{M}$ and F to form a new lit moat $\mathcal{M}$ in the ghost process. In the modified view, we call the tentative facility moat F at time t as a *hole* H in G_i.

Claim 3.8 *Consider a moat $\mathcal{M}$ in scenario i at time t. Any location $j \in \mathcal{M}$ is at a distance of at most t from a scenario terminal $v \in \mathcal{M}$ or a hole $H \subset \mathcal{M}$ created during the ghost process.*

The merge tree for scenario i is constructed as follows:

1. Suppose a moat $\mathcal{M}_1$ merges with a tentative facility moat F at time t at location j. There exists a terminal $v_1 \in \mathcal{M}_1$ or a hole $H_1 \subset \mathcal{M}_1$ which is at a distance t from j (wlog say v_1). In the merge tree $MT(i)$, we construct an edge e between v_1 and j. The fraction $f(e)$ associated with e is the rate at which $\mathcal{M}_1$ collects cost share at time t.

2. Suppose two moats $\mathcal{M}_1$ and $\mathcal{M}_2$ of scenario i merge at location j at time t. We can assume wlog that j is at a distance t from some terminal $v_1 \in \mathcal{M}_1$ and some hole $H_2 \subset \mathcal{M}_2$. In the merge tree $MT(i)$ we construct an edge e between v_1 and closest location $h \in H_2$. The fraction $f(e)$ associated with the edge is the maximum of the rates at which $\mathcal{M}_1$ and $\mathcal{M}_2$ collect cost shares at time t.

Lemma 3.9 *The total cost share collected by the terminals of a scenario i is at least a fraction $1/4$ of the total cost of $MT(i)$.*

Proof: Observe that the rate at which a scenario moat collects cost share decreases monotonically. The cost of the edge e joining $\mathcal{M}_1$ and $\mathcal{M}_2$ is at most $2t$. Let $r_1 (r_2)$ be the rate at which $\mathcal{M}_1 (\mathcal{M}_2)$ collects cost share at time t and suppose $r_1 > r_2$. Thus, using the argument in [AKR95] and charging the cost share collected by $\mathcal{M}_1$ twice we can build a virtual edge e at cost share r_1. The additional factor of 2 is lost because the cost of the tree is at most twice a feasible dual solution [AKR95]. ■

Recall that the dual ascent process for Steiner tree on the real open facilities continues in assumed to run in parallel to the ghost process. The following notation will be used in the remainder of the proof.

Definition 3.10 *The following components will be crucial to the following discussion:*

- ***Ghost component**: A tentative facility moat at time t and all the terminals of different scenarios which are within distance t of the moat.*
- ***Set of contributors** of real facility moat $\mathcal{M}_t$ at time t: set of scenario terminals which are within a distance of $\max\{t, t_p\}$ from a real facility p in moat $\mathcal{M}_t$ (where t_p is the opening time of facility p).*
- ***Real component**: a real facility moat $\mathcal{M}_t$ at time t and its set of contributors.*

The following lemma is a natural consequence of the condition for opening real facilities.

Lemma 3.11 *Any scenario terminal v is contained in one real component at any time.*

Proof: Suppose v is contained in real components R_1 and R_2 at time t. Thus, v is at a distance of at most $\max\{t, t_p\}$ from some real facility $p \in R_1$ and at a distance of at most $\max\{t, t_q\}$ from some real facility $q \in R_2$. Thus, $d(p, q) \leq \max\{t, t_p\} + \max\{t, t_q\}$. Clearly, $t < \max\{t_p, t_q\}$ since, otherwise $d(p, q) \leq 2t$ which implies R_1 and R_2 must have merged to form a single real component by time t. Assume wlog, $t_p \geq t_q$. Therefore, $d(p, q) \leq 2t_p$, which contradicts the fact that real facilities are opened at both p and q. ■

The next lemma, similar in spirit to the one in [PT03], helps in relating the cost shares collected by the set of contributors to the cost of the Steiner tree over open facilities.

Lemma 3.12 *The set of terminals contained in a ghost component at time t will be contained in the same real component at time $5t$.*

Proof: Consider a ghost component C at time t. There is a real facility r_f within a distance $4t$ of each tentatively open facility f in the ghost component C. Thus, all the terminals within a distance t of a tentatively open facility f are contained in the real component that contains the real facility r_f at time $5t$. Furthermore, we claim that all the real facilities which are close to some tentatively open facility in C are in the same real component at time $5t$. Let r_p, r_q be real facilities within a distance $4t$ from tentatively open facilities $p, q \in C$ respectively. We can find a path from p to q in C, say $\mathcal{P} = \{p, f_1, f_2, \ldots, f_k, q\}$ such that any two consecutive locations on the path are at a distance of at most $2t$. This is because p and q are in the same ghost component at time t. Also, since each of the locations $f_1, \ldots, f_k$ is a tentatively open facility, there exists a real facility r_{f_i} which is at a distance of at most $4t$ from f_i for $i = 1, \ldots, k$. Thus, in the sequence $r_p, r_{f_1}, \ldots, r_{f_k}, r_q$, any consecutive real facilities are at a distance of at most $10t$ which implies that all of them would be in the same real component at time $5t$. ■

We can now prove that the cost shares collected by the scenario moats are enough to pay for the Steiner tree over real facilities. Recall that for any real open facility p there is no other real open facility within a distance $4t_p$ from p, where t_p is the time when p is declared open in the ghost process. Let $\mathcal{M}_p$ denote the real component containing p. Until time t_p, facility p is the only open facility in $\mathcal{M}_p$. So, the terminals within a distance of t_p from p form the contributor set for $\mathcal{M}_p$ at time $t \leq t_p$. The following lemma proves that we can charge the cost of growing real components in the time interval $[0, t_p]$ to the first-stage cost shares of the set of contributors of the real component containing facility p at time t_p.

Lemma 3.13 *Consider a real open facility p and suppose $\mathcal{M}_p$ be the real component containing p till time t_p. The cost shares collected by the set of contributors of moat $\mathcal{M}_p$ can pay for its growth till time t_p.*

Proof: Real component $\mathcal{M}_p$ requires a cost share of $M \cdot t_p$ until time t_p. Note that there are terminals from at least M different scenarios in M_p. Thus, each contributing scenario needs to contribute a cost share of at most t_p towards $\mathcal{M}_p$. Let v be a contributing terminal in $\mathcal{M}_p$ and let $B(v, t_p)$ be a ball of radius t_p around v. $B(v, t_p)$ does not intersect any real component other than $\mathcal{M}_p$. Suppose $B(v, t_p)$ intersects a real component $R' \neq \mathcal{M}_p$ at time $t' < 2t_p$. Thus, there is a real facility $r \in R'$ at a distance of less than $3t_p$ from v. Also, p is at a distance of at most t_p from v. Thus, the distance between real facilities p and r is less than $4t_p$, a contradiction. At time $2t_p$, $B(v, t_p)$ is contained in $\mathcal{M}_p$, thus it does not intersect any other real component than $\mathcal{M}_p$.

For each contributing scenario i, consider a terminal $v_i \in \mathcal{M}_p$.

1. If the scenario moat containing v_i is lit before time t_p, then there exists a path $\mathcal{P}$ from v_i to a real facility $q \neq p$. Clearly, $q \notin B(v_i, t_p)$ which implies $\mathcal{P}$ crosses $B(v_i, t_p)$. The part of path $\mathcal{P}$ from v_i to the boundary of $B(v_i, t_p)$ has length t_p and we can charge the share of scenario i towards $\mathcal{M}_p$ to this part. Since, $B(v_i, t_p)$ does not intersect any other real component, we do not charge this part of path $\mathcal{P}$ again.

2. If the scenario moat SM_i containing v_i is dark until time t_p,

   (a) if there is a terminal $v' \in SM_i$ such that $v' \notin B(v_i, t_p)$, then there is a path $\mathcal{P}_{v_i, v'}$ from v_i to v'. We can charge the part of $\mathcal{P}_{v_i, v'}$ within $B(v_i, t_p)$ to pay the share t_p towards $\mathcal{M}_p$.

   (b) Assume there is no terminal in SM_i outside $B(v_i, t_p)$. The cost share collected by SM_i until time t_p is at least t_p. Thus, we can charge this cost share to pay towards growth of $\mathcal{M}_p$ until time t_p. Since, SM_i does not contain any terminal outside $B(v_i, t_p)$, the cost share collected by SM_i until t_p is not charged by any other real component.

■

The following lemma proves that the cost shares can continue to pay for the growth of real facility moats after the time facilities got opened in the ghost process.

Lemma 3.14 *Consider a real component R at time $5t$. Suppose $5t > t_p$ for all real facilities $p \in R$. The demanding rate of R at time $5t$ can be satisfied by the cost shares collected by its set of contributors at some time $t' \leq t$.*

Proof: Consider a real component R at time $5t > t_p, \forall p \in R$. There are at least M contributing scenarios in R. Let T_i^R be the terminals of scenario i in the set of contributors of R at time $5t$. If there is an active scenario moat $\mathcal{M}$ at time t in the ghost process of scenario i such that the terminals of $\mathcal{M}$ are contained in T_i^R, then (Lemma 3.12) the rate at which $\mathcal{M}$ collects cost shares at time t is at most the contribution requested from R to scenario i at time $5t$.

Thus, we can assume that every active scenario moat of scenario i at time t that contains terminals of T_i^R is not contained in R. Among the moats containing terminals of T_i^R, consider the moat $\mathcal{M}'$ that contained a terminal $u \in T_i^R$ and $v \notin T_i^R$ earliest, say at time t_f. There exist a path from u to v using the edges of merge tree and holes. Suppose, the path $\mathcal{P}(u, v) =< u = u_0, x_1, y_1, u_1, x_2, y_2, u_2, \ldots, x_s, y_s, u_s = v >$, where $u_i, i = 1, \ldots, s$ is a scenario terminal and x_i, y_i are locations on the boundary of hole $H_i \subset \mathcal{M}'$.

We claim that for all $j = 1, \ldots, s$, either both $x_j, y_j \in R$ or both $x_j, y_j \notin R$. This is because at time t_f both vertices x_j, y_j were part of some tentative facility moat and thus were contained in the same ghost component. Thus, at time $5t > 5t'$ both vertices must be contained in the same real component (due to Lemma 3.12). Thus, there must be an edge (u_{j-1}, x_j) or (y_j, u_j) that crosses R (wlog say (u_{j-1}, x_j)). Thus, we can charge the demand of R at time $5t$ from scenario i to the fractional cost of (u_{j-1}, x_j). The fraction $f(u_{j-1}, x_j)$ corresponding to the edge is greater than the demanding rate of R due to Lemma 3.12.

It is also clear that any two real components R and R' cannot load the same portion of an edge of the merge tree of scenario i. ■

The charge to cost shares for the Steiner forest on the scenario terminals is $8f^1$ (Lemma 3.7); the cost of the portion of the Steiner tree on the open facilities p charged until time t_p is $8f^1$ (Lemma 3.13). Finally, the remaining portion of the tree costs charge to $5 \cdot 4(f^1 + f^2) + 5(f^1 + f^2)$ by Lemma 3.14. This gives the following theorem.

Theorem 3.15 *The above cost-sharing scheme is 2-competitive, cross-monotone and 41-budget balanced.*

4 Cost Shares for CBTAN with Approximate Efficiency

In the previous section, we defined cost-shares for the CBTAN problem that were cross-monotonic, and (approximately) budget-balanced. In addition to these two properties, one may also want the cost-shares to give rise to Moulin mechanisms that result in high social welfare.

Definition 4.1 *Suppose each player $i \in U$ has a private utility u_i. For a set $S \subseteq U$, define $u(S) = \sum_{i \in S} u_i$. Define the* social cost $\Pi(S)$ *of a set $S \subseteq U$ as $\Pi(S) = u(U \setminus S) + C(S)$. The Moulin mechanism $M(\xi)$ is said to be* α-approximate *[RS06] if*

$$\Pi(S^M) \leq \alpha \cdot \Pi(S) \quad \forall S \subseteq U.$$

where S^M is the final set of players computed by the Moulin mechanism $M(\xi)$ on U.

We can also prove the following theorem:

Theorem 4.2 *There exist $O(1)$-budget-balanced cross-monotonic cost-shares which are also $O(\log^2 k)$-approximate; i.e., their inefficiency is at most $O(\log^2 k)$ times the inefficiency of any cost-sharing mechanism.*

Due to lack of space, all the details are given in Appendix B. This result extends the recent result of Roughgarden and Sundararajan [RS], who presented a cross-monotonic cost-sharing scheme for the Single-Source Rent-or-Buy (SSRoB) problem with an approximate efficiency of $O(\log^2 k)$. We end by noting that while we can define these cost-shares which are cross-monotone and even have a better budget balance factor than the cost shares defined in section 3, we do not yet have an efficient algorithm to compute these cost shares.

References

[AFK$^+$04] Aaron Archer, Joan Feigenbaum, Arvind Krishnamurthy, Rahul Sami, and Scott Shenker. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47(1):36–71, 2004.

[AKR95] Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995. (Preliminary version in *23rd STOC*, 1991).

[AZ02] Matthew Andrews and Lisa Zhang. Approximation algorithms for access network design. *Algorithmica*, 34(2):197–215, 2002. (Preliminary version in 39th FOCS, 1998.).

[Edm67] J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards*, B71:233–240, 1967.

[FKSS03] Joan Feigenbaum, Arvind Krishnamurthy, Rahul Sami, and Scott Shenker. Hardness results for multicast cost-sharing. *Theoretical Computer Science*, 304:215–236, 2003.

[FPS01] Joan Feigenbaum, Christos H. Papadimitriou, and Scott Shenker. Sharing the cost of multicast transmissions. *J. Comput. System Sci.*, 63(1):21–41, 2001. Special issue on internet algorithms.

[GKK$^+$01] Anupam Gupta, Amit Kumar, Jon Kleinberg, Rajeev Rastogi, and Bülent Yener. Provisioning a Virtual Private Network: A network design problem for multicommodity flow. In *Proceedings of the 33rd ACM Symposium on the Theory of Computing (STOC)*, pages 389–398, 2001.

[GKL76] J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.

[GKPR03] Anupam Gupta, Amit Kumar, Martin Pál, and Tim Roughgarden. Approximations via cost-sharing. In *Proceedings of the 44th Symposium on the Foundations of Computer Science (FOCS)*, pages 606–615, 2003.

[GKR03] Anupam Gupta, Amit Kumar, and Tim Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC)*, pages 365–372, 2003.

[GPRS04] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Boosted sampling: Approximation algorithms for stochastic optimization problems. In *Proceedings of the 36th ACM Symposium on the Theory of Computing (STOC)*, pages 417–426, 2004.

[GRS04] Anupam Gupta, R. Ravi, and Amitabh Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *Proceedings of the 45th Symposium on the Foundations of Computer Science (FOCS)*, pages 218–227, 2004.

[GST04] Anupam Gupta, Aravind Srinivasan, and Éva Tardos. Cost-sharing mechanisms for network design. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, volume 3122 of *Lecture Notes in Computer Science*, pages 139–150, 2004.

[GW95] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995. (Preliminary version in *5th SODA*, 1994).

[HST05] Ara Hayrapetyan, Chaitanya Swamy, and Éva Tardos. Network design for information networks. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 933–942, 2005.

[IKMM04] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 684–693, 2004.

[IMM05] N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost sharing schemes. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 602–611. ACM Press, 2005.

[JV01] Kamal Jain and Vijay Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 364–372, 2001.

[KLS05] Jochen Könemann, Stefano Leonardi, and Guido Schäfer. A group-strategyproof mechanism for Steiner forests. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 612–619. ACM Press, 2005.

[KLSvZ05] Jochen Könemann, Stefano Leonardi, Guido Schäfer, and Stefan van Zwam. From primal-dual to cost shares and back: a stronger LP relaxation for the Steiner forest problem. In *Automata, languages and programming*, volume 3580 of *Lecture Notes in Comput. Sci.*, pages 930–942. Springer, Berlin, 2005.

[KM00] David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41th Symposium on the Foundations of Computer Science (FOCS)*, pages 613–623, 2000.

[KSK96] Kathryn J. Kent and Darko Skorin-Kapov. Population monotonic cost allocations on MSTs. In *Proceedings of the 6th International Conference on Operational Research (Rovinj, 1996)*, pages 43–48. Croatian Oper. Res. Soc., Zagreb, 1996.

[LS04] Stefano Leonardi and Guido Schäfer. Cross-monotonic cost sharing methods for connected facility location games. *Theor. Comput. Sci.*, 326(1-3):431–442, 2004.

[Mou99] Hervé Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16:279–320, 1999.

[MS01] Hervé Moulin and Scott Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Econom. Theory*, 18(3):511–533, 2001.

[PT03] Martin Pál and Éva Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of the 44th Symposium on the Foundations of Computer Science (FOCS)*, pages 584–593, 2003.

[Rob79] K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.

[RS] T. Roughgarden and M. Sundararajan. Approximately efficient cost-sharing mechanisms. Unpublished manuscript.

[RS99] R. Ravi and F. S. Salman. Approximation algorithms for the traveling purchaser problem and its variants in network design. In *Algorithms—ESA '99 (Prague)*, volume 1643 of *Lecture Notes in Comput. Sci.*, pages 29–40. Springer, Berlin, 1999.

[RS06] T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *STOC*, 2006.

[SK04] Chaitanya Swamy and Amit Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[Vaz01] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

A A Primal-Dual Algorithm for CBTAN

A.1 LP formulation

Here, we present a primal dual algorithm for the CBTAN problem for which the LP relaxation can be formulated as follows.

$$\begin{array}{rrcll} \min & M\sum_{e\in E} c_e \cdot x_e^0 & + & \sum_{i=1}^{l}\sum_{e\in E} c_e \cdot x_e^i & \\ & \sum_{e\in\delta(S)}(x_e^i + x_e^0) & \geq & 1 & \forall S : S_i \subsetneq S, \forall i = 1,\ldots,l \\ & x_e^0, x_e^i & \geq & 0 & \forall e \in E, \forall i = 1,\ldots,l \end{array}$$

A set S *crosses* S_i if $S\cap S_i \notin \{\emptyset, S_i\}$, and we denote it by $S \odot S_i$. Then the dual of the above linear program is

$$\begin{array}{rrcll} \max & \sum_{i=1}^{l}\sum_{S:S\odot S_i}\beta_{S,i} & & & \\ & \sum_{S:S\odot S_i, e\in\delta(S)}\beta_{S,i} & \leq & c_e & \forall e\in E, \forall i = 1,\ldots,l \quad (1) \\ & \sum_{i=1}^{l}\sum_{S:S\odot S_i, e\in\delta(S)}\beta_{S,i} & \leq & M\cdot c_e & \forall e\in E, \forall i = 1,\ldots,l \quad (2) \end{array}$$

For simplicity, we assume that a facility can be opened anywhere along an edge. We show later how to remove this assumption incurring no additional cost.

A.2 The primal-dual algorithm

The algorithm runs in two phases. In the first phase, it identifies the locations to open facilities and connects the demand points to some open facility and in the second phase it builds a Steiner tree on the open facilities. The basic intuition behind the algorithm is simple: since the edges connecting the open facilities cost M times more than the edges connecting the demands to the facilities, the algorithm opens a facility only after it has succeeded in clustering demands from M different scenarios. This allows us to associate at least M different scenarios for each open facility and thus we can pay for the costlier connection between open facilities. Given this simple intuition, we need to flesh out the details, which we do in the following section.

A.2.1 The Algorithm: Phase I.

In this phase, we use the dual-ascent schema as in [AKR95, GW95]. For each scenario i, we maintain a separate Steiner forest E_i. We start with active components for each scenario as singleton terminals of that scenario. At time $t = 0$, all the variables $\beta_{S,i}$ are zero, and all the forests $E_i = \emptyset$. We start raising the dual values for active components uniformly until one of the following events happen.

i. **(Edge becomes tight.)** For some active component S of scenario i and edge $e \in \delta(S)$, the constraint (1) becomes tight. We include the edge e in the forest E_i for scenario i and update the active components, possibly merging two or more active components of scenario i. If the new component S' contains an open facility, we *freeze* the component S' (i.e., we do not increase the dual variables $\beta_{S',i}$ any further).

ii. **(M scenarios meet.)** For some location j, components from at least M different scenarios contain the location j. In this case, we *freeze* all the active components containing j and create a *tentative facility* at j (thereby adding j to a set F of tentative facilities).

We continue this process of raising duals of active components until all active components are frozen. We sometimes call these growing components *scenario components or moats*. At the end of Phase I, we are left with a subgraph E_i for each scenario i which connects each terminal in scenario i to some tentative facility in F, and let $\{\beta_{S,i}\}$ be the final dual solution. Phase I of the algorithm proceeds as follows.

I-1. **(Opening facilities.)** Let F be the set of tentative facilities and let t_j be the time when $j \in F$ was declared a tentative facility. We consider facilities in increasing order of opening times t_j, and declare some tentative facility j to be an *open (real) facility* if there is no open facility j' with $t_{j'} \leq t_j$ and distance $d(j, j') \leq 4t_j$. Let F' be the set of open facilities. We include root as an open facility in F'.

I-2. **(Removing Redundant edges.)** For every scenario i, we remove the redundant edges from the Steiner forest E_i; i.e. for each component $C \in E_i$, we select a minimal subgraph T that spans the terminals in C belonging to scenario i. Note that T may not contain an open facility at this point, so we let $j \in F$ be the tentative facility in the component C with the smallest t_j. We add a path connecting T to this facility j in the subgraph E_i and delete the rest of the edges of C.

I-3. **(Rerouting components.)** Since only a subset of the tentative facilities are opened, a component T from scenario i may not have any open facility in it, so we add a shortest path from T to its closest open facility to the edges in E_i.

The result of the above operations is a set F' of *open facilities* and a set of forests E_i (one for each scenario) such that each node in S_i is connected via edges in E_i to some open facility in F'. Before we describe Phase II, we present two simple lemmas that bound the cost of these forests E_i; proofs of these lemmas follow from standard primal-dual arguments, see e.g. [AKR95, GW95].

Lemma A.1 (Cost of Step I-2) *For any scenario i, let T be a component obtained after step **I**-2 in the above procedure. Then $c(T) \leq 2\sum_{S \subsetneq T} \beta_{S,i}$.*

Lemma A.2 (Cost of Step I-3) *Let T be a component in E_i after Step **I**-2, which happens to contain no open facility. Then the cost of connecting T to the closest open facility in F' is at most $4\sum_{S \subset T} \beta_{S,i}$*

Proof: Let the tree T be obtained by removing redundant edges from some component C; by construction, tree T contains a tentative facility j in C with the smallest t_j. Due to t_j being the smallest, each terminal $v \in C$ must be in an active component $S \subset T$ at any time instant $t < t_j$. Hence $\sum_{S: S \subset T} \beta_{S,i} \geq t_j$. Moreover, since j is not opened in Step **I**-1, there is an open facility within distance $4t_j$ of j, and hence the rerouting cost is bounded by $4t_j \leq 4\sum_{S \subset T} \beta_{S,i}$. ■

A.2.2 The Algorithm: Phase II.

We now go on to Phase II of the algorithm, where we build a Steiner tree connecting the open facilities. A little notation: each open facility $j \in F'$ has at least M different scenario components that contain j. Let $K_j \subseteq [l]$ be the set of scenarios whose Steiner components contain j. For each such scenario $i \in K_j$, consider the component C_j^i containing j, and the terminal $v_j^i \in C_j^i$ closest to j. Let $B_j^i = B(v_j^i, t_j)$ be a ball of radius t_j around the terminal v_j^i. Let $K_j' \subseteq K_j$ be the M scenarios in K_j in order of increasing distance $d(j, v_j^i)$ of the closest terminal v_j^i from the open facility j.

Lemma A.3 *Let j, j' be two facilities, with $i \in K_j'$ and $i' \in K_{j'}'$. Then B_j^i and $B_{j'}^{i'}$ are disjoint.*

Proof: Suppose $v \in B_j^i \cap B_{j'}^{i'}$. Thus, $d(v, v_j^i) \leq t_j$ and $d(v, v_{j'}^{i'}) \leq t_{j'}$. Also, $d(j, v_i^j) \leq t_j$ and $d(j', v_{j'}^{i'}) \leq t_j'$. Thus, the distance between facilities j and j' is at most $(d(j, v) + d(v, j')) \leq 2t_j + 2t_{j'} \leq 4\max\{t_j, t_{j'}\}$ which contradicts the rule of opening facilities. ■

The algorithm of Phase II is particularly simple.

II-1. **(Define Initial Moats.)** Define an *initial active moat* around the open facility j as $\mathcal{M}_j = B(j, t_j)$ where t_j is the time of opening of j. By Lemma A.3, the moats for two open facilities j and j' must be disjoint.

II-2. **(Connect Facilities.)** Starting with initial moats $\mathcal{M}_j$, run the Steiner tree algorithm of [AKR95, GW95] to connect all open facilities in F'.

The following lemma proves when we run the primal-dual algorithm in Step **II-2**, the duals β' we create actually form a feasible dual for the problem.

Lemma A.4 *The cost of Steiner tree T_F on the open facilities is at most $4\sum_{i=1}^{l}\sum_{S \subset V} \beta'_{S,i}$, where β' is a feasible dual.*

Proof: (Sketch) There are two components in the cost of the trees connecting the real facilities that must be accounted for - the portion of edges not inside the initial moats can be accounted for within a factor of two of the growing duals β' as in a standard primal dual argument; The portion of the tree within the initial moats is more problematic. Note that the initial moat around a real facility i that was opened at t_i contains at least M different scenario moats that have been accumulating duals for at least time t_i for a total dual of Mt_i. Since every pair i, j of real open facilities are pairwise $\max(4t_i, 4t_j)$ apart, each initial moat around i is active accumulating dual for at least another t_i time. We can charge the portion of edges of length t_i inside the initial moat i to the portion of the same edge along the first t_i time of growth around this initial moat, thus arguing that the cost of the full tree including the portions inside the initial moats is at most twice the cost of the portions outside them. (This charging idea is analogous to one in [RS99], and reused in [GKK$^+$01].) Since the portions outside have been argued to be at most twice the accumulated dual (which can be seen to be feasible), the overall cost of the tree is at most four times a feasible dual as claimed. ■

Theorem A.5 *The primal-dual algorithm for CBTAN is a 8-approximation algorithm.*

B Cost-Shares with Approximate Efficiency

In Section 3, we defined cost-shares for the CBTAN problem that were cross-monotonic, and (approximately) budget-balanced. In addition to these two properties, one may also want the cost-shares to give rise to Moulin mechanisms that result in high social welfare. Let us restate the theorem we want to prove:

Theorem 4.2 There exist $O(1)$-budget-balanced cross-monotonic cost-shares which are also $O(\log^2 k)$-approximate; i.e., their inefficiency is at most $O(\log^2 k)$ times the inefficiency of any cost-sharing mechanism.

The results of this section extend the recent result of Roughgarden and Sundararajan [RS], who presented a cross-monotonic cost-sharing scheme for the Single-Source Rent-or-Buy (SSRoB) problem with an approximate efficiency of $O(\log^2 k)$. (The fact that our result is an extension of the result for SSRoB follows from the fact that the SSRoB problem is the same as the CBTAN problem where the scenarios are singleton vertices; see, e.g., [IKMM04].) The cost-sharing scheme we will define is based on the "boosted sampling" technique proposed in [GPRS04] for StocST, which we now describe.

B.1 The Boost-and-Sample Framework

The Boost and Sample framework was proposed in [GPRS04] to obtain an algorithm for the two-stage stochastic version Stoc(Π) of some combinatorial optimization problem Π, given an algorithm for the original problem Π itself:

Algorithm B.1 *The* Boost-and-Sample *Algorithm:*

- ***B**-1.* Boosted Sampling*: Sample σ times from the set of scenarios to get sets of terminals $S_1, \ldots, S_\sigma$.*
- ***B**-2.* Building First Stage Solution*: Build an α-approximate solution for the clients $S = \bigcup_i S_i$.*
- ***B**-3.* Building recourse*: When actual future in the form of a set T of clients appears, augment the solution of Step **B**-2 to a feasible solution for T.*

A crucial role in the analysis of this technique is played by the notion of *strictness* [GKPR03, GPRS04]:

Definition B.2 *Given an α-approximation algorithm $\mathcal{A}$ for the problem* Π, *the function $\xi : 2^U \times U \to R_{\geq 0}$ is a β-*strict *cost sharing function if the following properties hold:*

1. *(**voluntary participation.**) For a set $S \subseteq U$, $\xi(S, j) > 0$ only for $j \in S$.*

2. *(**fairness**) For a set $S \subseteq U$, $\sum_{j \in S} \xi(S, j) \leq C(OPT(S))$.*

3. *(**strictness**) If $S' = S \cup T$, then $\sum_{j \in T} \xi(S', j) \geq (1/\beta) \times$ cost of* augmenting *the solution $\mathcal{A}(S)$ to a solution of S' at cost at most $\beta \sum_{j \in T} \xi(S', j)$.*

Theorem B.3 ([GPRS04]) *Given a combinatorial optimization problem Π that is sub-additive, let $\mathcal{A}$ be an α-approximation algorithm for its deterministic version with a β-strict cost-sharing function. Then* Boost-and-Sample *is an $(\alpha + \beta)$-approximation for* Stoc(Π)*.*

B.2 Defining the Cost Shares

Let us use the MST-heuristic to approximate the Steiner tree computed in Step **B**-2; i.e., we find a minimum spanning tree on the metric completion $G(S)$ of the vertex set S. (In the following, we abuse notation slightly and assume the root vertex r is part of S.) It is well-known that a minimum spanning tree on $G(S)$ is a 2-approximation of the optimal Steiner tree on S (see, e.g., [Vaz01]).

One way of computing the MST on a set of points is to bidirect every undirected edge, run Edmonds' primal-dual algorithm [Edm67] to compute a minimum cost arborescence, and then to simply discard the directions on edges. We can associate the standard notion of time with Edmonds' primal-dual algorithm on S, and at time t, use $s_j(t)$ to denote the number of vertices in the strongly connected component containing j. We define $\beta_j(t) = 1/s_j(t)$ if the component containing j does not contain the root, and $\beta_j(t) = 0$ otherwise. Jain and Vazirani [JV01] showed the cost shares $\alpha_j = \frac{1}{2} \int_0^\infty \beta_j(t) dt$ are cross-monotonic and 2-budget-balanced for the Steiner tree game. (We refer to these cost-shares as the *JV cost shares* in the following discussion.)

It is natural to consider the player j's *random* cost share (with respect to the random set S) as

$$\alpha_{S,j} = \begin{cases} \int_0^\infty \frac{1}{2} \beta_j(t) dt & \text{if } j \in S, \text{ and} \\ 0 & \text{if } j \notin S. \end{cases} \tag{B.3}$$

Then the cost-share $\xi(S, j)$ can be defined as

$$\xi(S, j) = E_S[\alpha_{S,j}].$$

Following the arguments presented in [LS04, GST04] for the SSRoB problem, one can infer that the cost shares ξ are competitive and cross-monotonic for the CBTAN problem. In order to prove that they are also budget balanced for CBTAN, we need to prove that they are both approximated and strict.

Lemma B.4 *The cost shares defined by Jain and Vazirani [JV01] for the Steiner tree problem are 2-approximated and 2-strict.*

Proof: JV cost shares are 2-approximated [JV01]. For proving 2-strictness, consider sets of terminals S, T, and $S' = S \cup T$ and the execution of the branching algorithm of Edmonds on S'. Let C be a strongly connected component $C \subseteq T$ that connects at time t to vertex v of a component C' that either contains the root r or a vertex of S. We know that the total cost share collected by the vertices of C till time t is sufficient to pay for at least half a spanning tree of C and half the cost of the edge connecting C to vertex v. For the base of the induction, the lemma is proved if $v = r$ or if $v \in S$. For the inductive step, if $v \in T$, assume that the total cost share collected by the vertices of $T \cap C'$ till time t is sufficient to pay for the connection to vertices of $S \cap C'$ or to root $r \in C'$. The claim then follows since vertex v will allow the vertices of C to connect to r or to a vertex of $S \cap C'$. This implies 2-strictness of JV cost-shares. ■

We finally prove that the cost-shares $\xi(S, j)$ are approximately efficient for CBTAN. Roughgarden and Sundararajan [RS06] proved that the Moulin mechanism $M(\xi)$ is $(\alpha + \beta)$-approximate and β-budget balanced if ξ is α-*summable* and β-budget balanced. The summability of a cost sharing method is defined as follows: Assume we are given an arbitrary permutation σ on the players in U and a subset $S \subseteq U$ of players. We assume that the players in S are ordered according to σ, i.e., $S = \{i_1, \ldots, i_{|S|}\}$ where $i_j \prec_\sigma i_k$ if and only if $1 \leq j < k \leq |S|$. We define $S_j \subseteq S$ as the (ordered) set of the first j players of S according to the order σ.

Definition B.5 *A cost sharing method ξ is α-*summable *if for every ordering σ and every subset $S \subseteq U$*

$$\sum_{j=1}^{|S|} \xi_{i_j}(S_j) \leq \alpha \cdot C(S). \tag{B.4}$$

where S_j is the set of the first j players, and i_j is the j^{th} player according to the ordering σ.

Lemma B.6 *Cost shares $\xi_(S, j)$ are $O(\log^2 k)$-summable for the CBTAN cost function.*

Proof: The expected cost of the solution provided by the Boost-and-Sample algorithm is related by a constant factor to the expected cost of a Steiner tree computed on a set $S = \bigcup_{i=1}^{\sigma} S_i$. The proof then reduce to the summability of the Steiner tree cost function, i.e., given a fix order σ of players in S,

$$\sum_{j=1}^{|S|} \xi_{i_j}(S_j) = O(\log^2 k) \cdot C_{ST}(S), \tag{B.5}$$

where S_j is the set of the first j players, i_j is the j^{th} player according to the ordering σ, and $C_{ST}(S)$ is the cost function of a Steiner tree connecting S to the root r. The claim the follows from the $O(\log^2 k)$-summability of JV's cost shares [RS06]. ■

We therefore conclude with

Theorem B.7 *Cost shares $\xi(S, j)$ are cross-monotonic, 4-approximated and $O(\log^2 k)$-approximately efficient for the CBTAN problem.*