Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games

Stefano Leonardi[*] *Guido Schäfer*[†]

July 19, 2004

Abstract

We present cost sharing methods for connected facility location games that are cross-monotonic and competitive and that recover a constant fraction of the cost of the constructed solution. The novelty of this paper is that we use randomized algorithms and that we share the expected cost among the participating users. As a consequence, our cost sharing methods are simple and achieve attractive approximation ratios. We also provide a primal-dual cost sharing method for the connected facility location game with opening costs.

1 Introduction

The problem of achieving truth-revealing or strategyproof mechanisms for sharing the cost of deploying a network infrastructure has recently received growing attention in computer science. In this work we are interested in the design of cost sharing mechanisms that would incite agents to cooperate to share the cost of the network facility and to reveal their true value for receiving the service, i.e., *group-strategyproof* mechanisms for which truthfulness is a dominant strategy for every user or coalition of users.

Suppose we are given a set U of (potential) users that want to utilize a common service. Each user $j \in U$ has a *utility* u_j, which corresponds to the price she is willing to pay for the service. If j is asked to pay more than u_j, she prefers to not receive the service. For each set $Q \subseteq U$ of users, let $C(Q)$ denote the cost of servicing all users in Q. The task is to design a *cost sharing mechanism*, i.e., an algorithm that determines (i) a set $Q \subseteq U$ of participating users that receive the service, and (ii) how to distribute the servicing cost $C(Q)$ among all users in Q such that each user $j \in Q$ is willing to pay her cost share, p_j. The *benefit* of a user j is $u_j - p_j$ if $j \in Q$, and zero if $j \notin Q$. We assume that each user is selfish and hence may misreport her utility so as to maximize her benefit. A cost sharing mechanism is *strategyproof* if each user has no incentive to misreport her true utility; it is said to be *group-strategyproof* if the same holds even if users collude.

Given a set Q of participating users, a *cost sharing method* ξ computes a cost share $\xi_j(Q)$ for each user $j \in Q$. We are particularly interested in cost sharing methods that are *cross-monotonic*, i.e., that have the property that the cost share of each individual user never increases

[*] Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Salaria 113, 00196 Roma, Italy. Email: `Stefano.Leonardi@dis.uniroma1.it`.

[†] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany. Email: `schaefer@mpi-sb.mpg.de`.

as the set of participating users grows. More formally, a cost sharing method ξ is cross-monotonic if it satisfies

$$\forall Q' \subseteq Q \subseteq U, \forall j \in Q', \quad \xi_j(Q') \geq \xi_j(Q).$$

The importance of cross-monotonic cost sharing methods is due to a result of Moulin and Shenker [6]: Let ξ be a cross-monotonic cost sharing method. Then, the following mechanism is group-strategyproof. Initialize $Q \leftarrow U$. If for each user $j \in Q$ the cost share $\xi_j(Q)$ is less than or equal to her reported utility, stop. Otherwise, remove from Q all users whose cost shares are larger than their utilities, and repeat.

It is well known that *competitiveness* and *cost recovery* are conflicting objectives for several games such as facility location and Steiner tree [5]. In this paper, we are interested in cost sharing methods that are cross-monotonic, and satisfy *competitiveness* and *approximate cost recovery*. Competitiveness requires that the participating users in Q are not charged more than the cost, $C^*(Q)$, of an optimal solution, i.e., $\sum_{j \in Q} \xi_j(Q) \leq C^*(Q)$. Cost recovery states that the total cost paid by the users covers the cost $C(Q)$ of the constructed solution, i.e., $\sum_{j \in Q} \xi_j(Q) \geq C(Q)$. Ideally, we may want to require that the constructed solution is optimal and therefore $\sum_{j \in Q} \xi_j(Q) = C^*(Q)$. However, we cannot enforce this condition if the underlying problem is NP-hard. We therefore relax the cost recovery condition and only require that a constant fraction $1/\lambda$, for some $\lambda \geq 1$, of the cost of the constructed solution is recovered: $\sum_{j \in Q} \xi_j(Q) \geq C(Q)/\lambda$. We call such a cost sharing method a *λ-approximate cost sharing method*.

Related Work. Cross-monotonic cost sharing mechanisms have been devised by Moulin and Shenker [6] when the optimal cost function is a submodular function of the set U. This is not the case for several network design problems such as Steiner tree, facility location, or rent-or-buy network design.

Jain and Vazirani [5] presented a cross-monotonic cost sharing method for the minimum spanning tree game and therefore a 2-approximate cost sharing method for the Steiner tree game. More recently, Devanur, Mihail and Vazirani [1] proposed strategyproof mechanisms for vertex cover and facility location games based on primal-dual algorithms. However, their algorithms are not group-strategyproof. In all these methods, the cost shares are closely related to a feasible dual solution generated by the algorithm and therefore competitiveness and approximate cost recovery are immediate consequences of the approximation guarantee achieved by the algorithm.

Very recently, Pàl and Tardos [7] proposed cross-monotonic cost sharing methods for facility location and single-source rent-or-buy network design. Their method is based on a novel idea of using primal-dual algorithms to obtain cross-monotonic cost sharing methods. Roughly speaking, the cost share is fixed when the user is connected to the network. However, an underlying *ghost* process continues to contribute to connect other users to the network. They present a 3-approximate cost sharing method for facility location and a 15-approximate cost sharing method for single-source rent-or-buy network design.

Among the approximation algorithms for single-source rent-or-buy developed in literature, we mention the primal-dual based 4.55-approximation of Swamy and Kumar [8] and the recent 3.55-approximation obtained through a novel and simple randomized algorithm proposed by Gupta, Kumar, and Roughgarden [3]. We will show how ideas of this last work can be turned into a cross-monotonic cost sharing method that recovers a larger fraction of the cost.

Our Contribution. In this paper, we present cross-monotonic cost sharing methods for connected facility location games. Our contribution is twofold.

We present a novel idea of sharing the expected cost of a randomized algorithm. Recently, Gupta, Kumar, and Roughgarden [3] gave a simple randomized algorithm for the single-source rent-or-buy problem. Using their algorithm, we define random cost shares and prove that the expected cost shares are cross-monotonic, competitive, and with high probability recover at least a $\frac{1}{4}(1+\varepsilon)^{-1}$-fraction of the constructed solution, where $\varepsilon > 0$ is an arbitrary constant.

Unfortunately, to compute the expected cost shares in polynomial time, it seems that one needs to derandomize the algorithm of Gupta et al. Despite some effort, we were not able to do so. However, we believe that the idea of sharing the expected cost will lead to attractive approximation ratios for cost sharing methods in the future. In a recent independent work by Gupta, Srinivasan, and Tardos [4] a similar idea is pursued to obtain a 4.5-approximate cross-monotonic cost sharing method for the single-source rent-or-buy problem. The authors show how to derandomize a version of the algorithm of Gupta et al. [3] at the expenses of a slight weakening of the approximation guarantee to compute cost shares in polynomial time.

Our second contribution is to extend the recent result of Pál and Tardos [7] to the connected facility location game with opening costs. In general, an algorithm for connected facility location consists of a first phase, in which users are grouped into clusters, with every cluster being represented by a location point, and a second phase, in which all locations are connected by a Steiner tree. The 15-approximate cross-monotonic cost sharing method of Pál and Tardos is restricted to the case in which locations can be opened at every point of the network and at zero cost. This is clearly not realistic in many applications in which only specific sites can host facilities and the cost of the network is formed by the individual costs of the opened facilities plus the cost of deploying a high bandwidth network infrastructure to connect all facilities. We give a 30-approximate cross-monotonic cost sharing method for this more general problem.

2 Problem Definition

In the *connected facility location problem* (*CFL*) we are given an undirected graph $G = (V, E)$ with non-negative edge costs $c : E \to \mathbb{R}^+$, a set $\mathcal{F} \subseteq V$ of potential facilities with opening cost f_i for each facility $i \in \mathcal{F}$, a set $D \subseteq V$ of demands (or agents, users), and a parameter $M > 1$. The goal is to open a subset $F \subseteq \mathcal{F}$ of facilities, to connect each demand $j \in D$ to the closest open facility $i(j) \in F$, and to build a Steiner tree T connecting all open facilities in F. The objective is to minimize the sum of the opening costs, the connection costs, and M times the Steiner tree cost, i.e.,

$$\sum_{i \in F} f_i + \sum_{j \in D} c(j, i(j)) + M \cdot c(T),$$

where $c(\cdot,\cdot)$ is the shortest path distance with respect to c, and $c(T)$ is the cost of the edges in the Steiner tree T. We may assume without loss of generality that a root node $r \in \mathcal{F}$, which is open in some optimal solution, is known in advance. (Otherwise, we could try all at most $|V|$ possibilities for r.)

In *rent-or-buy* network design problems an edge e can either be *bought* at cost $M \cdot c_e$, or *rented* at cost c_e; a bought edge can be used by an arbitrary number of paths, while a rented edge e costs c_e for each path that uses it.

The *single-source rent-or-buy problem* (*SSRB*) is a special case of *CFL*, where a facility can be opened at any node and all opening costs are zero, i.e., $\mathcal{F} = V$ and $f_i = 0$ for all $i \in \mathcal{F}$.

The problem then essentially reduces to establishing a minimum cost network such that each demand $j \in D$ is connected to the root r by a path.

3 Single-Source Rent-or-Buy Game

Gupta, Kumar, and Roughgarden [3] presented a randomized approximation algorithm for *SSRB*. For a given set D of demands, the algorithm works as follows:

SIMPLECFL(D):

1. Mark each demand $j \in D$ with probability $1/M$. Let $D' \subseteq D$ denote the set of marked demands.
2. Construct a ρ_{ST}-approximate Steiner tree T on $F = D' \cup \{r\}$.
3. Connect each demand $j \notin D'$ to its closest facility in F.

Gupta et al. prove that SIMPLECFL has an expected approximation ratio of $(2 + \rho_{\text{ST}})$, where ρ_{ST} denotes the approximation ratio of the Steiner tree algorithm used in Step 2.

For a given subset $Q \subseteq D$ of demands, we use SIMPLECFL(Q) to define a random cost share $\alpha_j(Q)$ for each demand $j \in Q$ and prove that the *expected* cost shares $\xi_j(Q)$, defined as $\xi_j(Q) = \frac{1}{4}\mathbf{E}[\alpha_j(Q)]$, are cross-monotonic. Moreover, we show that the expected cost shares are competitive and with high probability recover at least a $\frac{1}{4}(1+\varepsilon)^{-1}$-fraction of the cost of the constructed solution, where $\varepsilon > 0$ is an arbitrary constant.

3.1 Cost Shares

We approximate the Steiner tree in Step 2 by computing a minimum spanning tree on the metric completion of F, denoted by $G(F)$. It is known that a minimum spanning tree on $G(F)$ is a 2-approximation of the optimal Steiner tree on F, see, e.g., [9]. We compute the minimum spanning tree on $G(F)$ by running Edmonds' primal-dual algorithm [2] to compute a minimum branching on a graph $\vec{G}(F)$, which is obtained from $G(F)$ if for each edge in $G(F)$ we also add the reversed edge. Having computed a minimum branching on $\vec{G}(F)$, we obtain a minimum spanning tree on $G(F)$ by simply discarding directions of edges. We associate the standard notion of time with the primal-dual branching algorithm on $\vec{G}(F)$. At time t, let $s_j(t)$ 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 [5] showed that for the Steiner tree game the cost shares $\alpha_j = \int_0^\infty \beta_j(t)dt$ are cross-monotonic. We will exploit this fact later to prove cross-monotonicity for the expectation of the cost shares defined below.

We define j's random cost share with respect to Q as

$$\alpha_j(Q) = \begin{cases} M \cdot \int_0^\infty \beta_j(t)dt & \text{if } j \in F, \text{and} \\ c(j,F) & \text{if } j \notin F. \end{cases} \tag{1}$$

Here, $c(j,F)$ denotes the shortest path distance from j to a facility in F. Note that both $\beta_j(t)$ and $c(j,F)$ are random variables.

3.2 Cross-Monotonicity

We next prove that ξ, defined as $\xi_j(Q) = \frac{1}{4}\mathbf{E}[\alpha_j(Q)]$ for each $Q \subseteq D$, $j \in Q$, is cross-monotonic. Essentially, the argument is as follows. Let $Q' \subseteq D$ be a subset of demands. Consider the cost shares defined by $\textsc{SimpleCFL}(Q')$ and let $F' \subseteq Q'$ denote the set of open facilities. Assume that we add a demand k to Q'. Let the new set of demands be $Q = Q' \cup \{k\}$. We use $F \subseteq Q$ to denote the set of open facilities with respect to Q. Conditioned on the event that in $\textsc{SimpleCFL}(Q)$ the outcomes of the coin flips for demands in Q' are the same as before, we have two possibilities for F: (i) $F = F' \cup \{k\}$ (probability $1/M$), or (ii) $F = F'$ (probability $1 - 1/M$).

If k becomes part of F, the cross-monotonicity of the Steiner tree game implies that the cost share of each demand $j \in F'$ can only decrease. Moreover, the connection cost of each demand $j \in Q' \setminus F'$ can only decrease because of the additional option to connect to k.

If k does not become part of F, the cost share of each facility $j \in F'$ remains the same, and the cost share of each $j \in Q' \setminus F'$ can only decrease, since the shortest path distance from j to F can only decrease (via $k \in Q' \setminus F'$).

Lemma 1. *ξ is a cross-monotonic cost sharing method.*

Proof. Let $Q' \subset D$ be an arbitrary subset of demands, and let $Q = Q' \cup \{k\}$ for some $k \notin Q'$. It is sufficient to show that for each $j \in Q'$, $\xi_j(Q') \geq \xi_j(Q)$. Throughout the proof, let F' and F, respectively, denote the set of open facilities of Q' and Q. We have

$$\begin{aligned}\mathbf{E}[\alpha_j(Q)] &= \sum_{O \subseteq Q} \mathbf{E}[\alpha_j(Q) \,|\, F = O] \cdot \mathbf{P}[F = O] \\ &= \sum_{O \subseteq Q'} \Big(\mathbf{E}[\alpha_j(Q) \,|\, F = O] \cdot \mathbf{P}[F = O] + \mathbf{E}[\alpha_j(Q) \,|\, F = O \cup \{k\}] \cdot \mathbf{P}[F = O \cup \{k\}]\Big).\end{aligned}$$

From the discussion above, we know that for each $j \in Q'$ and for each $O \subseteq Q'$,

$$\begin{aligned}\mathbf{E}[\alpha_j(Q) \,|\, F = O] &\leq \mathbf{E}\big[\alpha_j(Q') \,|\, F' = O\big]\,, \text{ and} \\ \mathbf{E}[\alpha_j(Q) \,|\, F = O \cup \{k\}] &\leq \mathbf{E}\big[\alpha_j(Q') \,|\, F' = O\big]\,.\end{aligned}$$

Thus,

$$\mathbf{E}[\alpha_j(Q)] \leq \sum_{O \subseteq Q'} \mathbf{E}\big[\alpha_j(Q') \,|\, F' = O\big] \cdot \big(\mathbf{P}[F = O] + \mathbf{P}[F = O \cup \{k\}]\big).$$

The proof now follows from the observation that for each $O \subseteq Q'$,

$$\mathbf{P}[F = O] + \mathbf{P}[F = O \cup \{k\}] = \mathbf{P}\big[F' = O\big]\,.$$

□

The next lemma shows that the cost share of a demand $j \notin F$ can be computed efficiently. We are not able, however, to efficiently compute j's cost share if $j \in F$.

Lemma 2. *Let $Q \subseteq D$, and let $j \in Q$ be a demand. The expected connection cost of j with respect to Q can be computed in polynomial time.*

Proof. Consider the set $Q^- = Q \setminus \{j\}$ of all demands except j. Let $v_1, v_2, \ldots, v_l$, $l = |Q^-|$, denote the demands in Q^- ordered according to non-decreasing distances from j. Then,

$$\begin{aligned}\mathbf{E}[\alpha_j(Q) \,|\, j \notin F] &= \frac{1}{M}c(j, v_1) + \frac{1}{M}\left(1 - \frac{1}{M}\right)c(j, v_2) + \frac{1}{M}\left(1 - \frac{1}{M}\right)^2 c(j, v_3) + \ldots \\ &= \frac{1}{M}\sum_{i=1}^{l}\left(1 - \frac{1}{M}\right)^{i-1} c(j, v_i).\end{aligned}$$

□

3.3 Competitiveness and Cost Recovery

For a subset $Q \subseteq D$ of demands, let $C(Q)$ be a random variable denoting the cost of the solution of SIMPLECFL(Q). We use $C^*(Q)$ to denote the cost of an optimal solution for Q.

Lemma 3. *The cost shares $\xi_j(Q) = \frac{1}{4}\mathbf{E}[\alpha_j(Q)]$ are competitive and, for any constant $\varepsilon > 0$, with high probability recover at least a $\frac{1}{4}(1 + \varepsilon)^{-1}$-fraction of the cost of the constructed solution.*

Proof. From the analysis of Gupta et al. [3] we know that the expected cost $\mathbf{E}[C(Q)]$ of the solution is at most $4C^*(Q)$. Moreover, $\mathbf{E}[C(Q)] = \sum_{j \in Q} \mathbf{E}[\alpha_j(Q)]$. We conclude that

$$\sum_{j \in Q} \xi_j(Q) = \frac{1}{4}\mathbf{E}\left[\sum_{j \in Q} \alpha_j(Q)\right] \leq C^*(Q).$$

By Markov's inequality we have with probability at most $(1 + \varepsilon)^{-1}$, for any constant $\varepsilon > 0$, that $C(Q) > (1 + \varepsilon)\mathbf{E}[C(Q)]$. Thus, by rerunning the algorithm $\log(n)/\log(1 + \varepsilon)$ times, SIMPLECFL(Q) computes a solution such that with high probability

$$\sum_{j \in Q} \xi_j(Q) \geq \frac{1}{4}(1 + \varepsilon)^{-1} \cdot C(Q).$$

□

Theorem 1. *The cost shares $\xi_j(Q) = \frac{1}{4}\mathbf{E}[\alpha_j(Q)]$ are cross-monotonic, competitive, and, for any constant $\varepsilon > 0$, with high probability recover at least a $\frac{1}{4}(1 + \varepsilon)^{-1}$-fraction of the cost of the constructed solution.*

4 Connected Facility Location Game

Recently, Pál and Tardos [7] gave a 15-approximate cross-monotonic cost sharing method for the facility location problem and the single-source rent-or-buy problem. Their cute idea is to consider two processes: A “ghost” process to determine the cost shares and a “real” process constructing a solution to the problem. The ghost process is designed in a way that the cost shares are trivially cross-monotonic. The difficult part is to link the ghost process to the real process and to prove that at least a constant fraction of the cost of the computed solution is recovered.

We extend the result of Pál and Tardos to the connected facility location game with opening costs. Using similar ideas, we give a cross-monotonic cost sharing method that is competitive and recovers at least a $\frac{1}{30}$-fraction of the cost of the solution.

We make the simplifying assumption that the edges of G consist of a continuum of points. We use the term *location* to refer to both original vertices and intermediate points. Basically, the algorithm works as follows. We first form clusters around some locations such that each cluster contains at least M demands; call these locations *centers*. The clusters are then connected by building a Steiner tree T on the centers. Moreover, for each cluster we identify a facility that is opened and to which all demands in the cluster are assigned. Finally, we buy the shortest path from each center to the corresponding facility. Using the idea of Swamy and Kumar [8], we can transform the Steiner tree on the centers into a tree on original vertices of the graph without increasing the cost.

4.1 Ghost Process

For a given set $Q \subseteq D$ of demands, we run the following ghost process to determine three different cost shares, $\alpha_j(Q)$, $\alpha'_j(Q)$, and $\alpha''_j(Q)$, for each $j \in Q$. The final cost share of j will be a combination of the three.

We associate a notion of time with the process. For each demand j, we have a ghost ball $B_j(t)$, which is centered at j, and has radius equal to the current time t. When M or more balls intersect a common location p, we open p. We use t_p to refer to the time when p is opened and Q_p to denote the set of demands that are responsible for the opening of p, i.e., all demands j that satisfy $c(j,p) \leq t_p$. We say that the demands in Q_p form a *cluster*.

All locations open at time t form a set $\mathcal{C}(t)$. $\mathcal{C}(t)$ can be seen as being partitioned into *connected components*: A connected component C is an inclusion maximal subset of $\mathcal{C}(t)$ such that for any two locations p and q of C, all locations along the path from p to q are contained in C. Consider the evolution of $\mathcal{C}(t)$ over time. Initially, $\mathcal{C}(0)$ is empty. With increasing time, components start to appear. Each component C starts as a single location p. As time progresses, C grows uniformly like a ball with center p. Eventually, two or more components touch and are merged into a single component. Observe that the growing of components in the ghost process is very similar to the growing of components in the standard primal-dual algorithm for Steiner trees. The difference, however, is that in the ghost process new components may appear at arbitrary locations over time, while in the primal-dual algorithm we start to grow components from a fixed set of locations.

A demand j is connected to a component C of $\mathcal{C}(t)$ at time t, if $B_j(t) \cap C \neq \emptyset$. For each demand j, let t_j denote the time when j becomes connected to some component for the first time, and let t'_j denote the time when j becomes connected to a component that contains the root. For a component C of $\mathcal{C}(t)$, let $Q(C)$ denote the set of demands that are connected to C at time t. For a connected demand j, let $s_j(t)$ denote the maximum size $|Q(C)|$ over all components C of $\mathcal{C}(t)$ that j is connected to at time t.

At time t, the contribution of demand j to the opening cost of a facility i is $\max(0, t - c(j,i))$. If the total contribution towards a facility i equals the opening cost f_i, we open i. Let t_i denote the time when facility i is opened, and let Q_i denote the set of demands that contribute to the opening of i at time t_i.

For each demand $j \in Q$, we define three different cost shares:

$$\alpha_j(Q) = t_j,$$

$$\alpha'_j(Q) = t_j + M \cdot \int_{t_j}^{t'_j} \frac{1}{s_j(t)} dt, \text{ and}$$

$$\alpha''_j(Q) = \min\left(\min_{i:j \in Q_i} t_i, \min_{i:j \notin Q_i} c(j,i) \right).$$

Lemma 4. *α, α', and α'' are cross-monotonic cost sharing methods.*

Proof. (See also [7].)

Consider a set $Q' \subset D$ of demands and assume we add one additional demand k to Q'. By adding demand k, at any time t, the set $\mathcal{C}(t)$ can only become larger. Thus, the connection time t_j of a demand $j \in Q'$ can only become smaller. Moreover, since the number of demands that are connected to a component C of $\mathcal{C}(t)$ at time t can only become larger, $s_j(t)$ can only increase for a demand $j \in Q'$. Furthermore, by adding k to Q', the opening time t_i of any facility i can only become smaller. □

4.2 Algorithm

We run the ghost process but take the following rules into account in order to decide which locations and facilities are eventually opened.

- We open a location p at time t_p only if there is no other open location q with $c(p,q) \leq 2t_p$.
- We open a facility i at time t_i only if there is no other open facility k with $c(i,k) \leq 2t_i$.

Locations that are opened in the above process are called *centers*. Let F' denote the set of facilities that were opened. As will be seen below, these rules assure that (i) all clusters Q_p with center p are disjoint and (ii) all sets Q_i with $i \in F'$ are disjoint. The final solution is constructed as follows. For each open cluster Q_p, we determine a facility, $i(p)$, which is open and closest to center p; we say that $i(p)$ is the facility of cluster Q_p. Let F denote the set of all facilities $i(p)$ corresponding to open clusters Q_p, i.e., $F = \{i \in F' : i = i(p) \text{ for some center } p\}$. We assign all demands of a cluster Q_p to its facility $i(p)$. Demands that are not contained in any open cluster are assigned to the facility of their closest center. We build a Steiner tree on the centers and for each center p buy the shortest path from p to the facility $i(p)$ of the cluster. This makes sure that the facilities in F are connected. Observe that no demand is assigned to a facility in $F' \setminus F$. We therefore close all facilities in $F' \setminus F$.

4.3 Analysis

Lemma 5. *For any two centers p and q, Q_p and Q_q are disjoint.*

Proof. Assume otherwise, i.e., $Q_p \cap Q_q \neq \emptyset$. Let j be a demand that is contained in Q_p and in Q_q. Then, $c(j,p) \leq t_p$ and $c(j,q) \leq t_q$. Without loss of generality assume that $t_p \leq t_q$. When we are about to open q, we have $c(p,q) \leq c(j,p) + c(j,q) \leq t_p + t_q \leq 2t_q$, which is a contradiction, since then q would not be opened. □

Lemma 6. *For any two open facilities i and k, Q_i and Q_k are disjoint.*

Proof. Assuming that we open all facilities in F', we can prove the lemma analogously to Lemma 5. Since $F \subseteq F'$, the lemma also holds for facilities in F. □

Lemma 7. *Let $j \in Q_p$ for some center p. Then $t_p \leq 3\alpha_j$. For each demand j that does not belong to any open cluster, there is a center p such that $c(j, p) \leq 3\alpha_j$.*

Proof. (See also [7].)

Let $j \in Q_p$ for some center p and assume that $t_p > 3\alpha_j$. Let q be the location that defines α_j, i.e., $\alpha_j \geq t_q$ and $\alpha_j \geq c(j, q)$. If q is open, we have a contradiction, since then $c(p, q) \leq t_p + \alpha_j < 2t_p$. Assume that q is not open. Since q was not opened at time t_q, there must exist an open location q' such that $c(q, q') \leq 2t_q$. But then

$$c(p, q') \leq t_p + \alpha_j + 2t_q \leq t_p + 3\alpha_j < 2t_p,$$

a contradiction.

Next, assume that j is not contained in any open cluster. Let q be the location that defines α_j, i.e., $\alpha_j \geq t_q$ and $\alpha_j \geq c(j, q)$. Since q is not opened at time t_q, there must exist an open location p with $c(q, p) \leq 2t_q$. Thus, $c(j, p) \leq \alpha_j + 2t_q \leq 3\alpha_j$. □

Lemma 8. *Let $j \in Q_i$ for some facility $i \in F'$. Then $t_i \leq 3\alpha''_j$. For each demand j that does not belong to any set Q_k with facility $k \in F'$, there is a facility $i \in F'$ such that $c(j, i) \leq 3\alpha''_j$.*

Proof. Same as for Lemma 7. □

Lemma 9. *The cost of the Steiner tree on the centers is at most $\sum_{j \in Q} 6\alpha'_j$.*

Proof. The proof is given in [7]. □

Lemma 10. *The cost of opening facility $i \in F'$ is at most $\sum_{j \in Q_i}(3\alpha''_j - c(j, i))$.*

Proof. We have $f_i = \sum_{j \in Q_i}(t_i - c(j, i))$. The lemma follows from Lemma 8. □

Lemma 11. *The cost of connecting all demands of an open cluster Q_p to facility $i(p)$ is at most $\sum_{j \in Q_p} 6\alpha_j + 3\alpha''_j$.*

Proof. Let $j \in Q_p$. If there exists some open facility $i \in F'$ with $j \in Q_i$, we have $c(j, i) \leq t_i \leq 3\alpha''_j$ by Lemma 8. Otherwise, if j is not contained in any set Q_k with facility $k \in F'$, by Lemma 8 there exists an open facility $i \in F'$ with $c(j, i) \leq 3\alpha''_j$. That is, for each $j \in Q_p$, there exists a facility in F' within distance at most $3\alpha''_j$. Since we choose $i(p)$ from F' as the facility that is closest to p, we have

$$c(p, i(p)) \leq t_p + \min_{l \in Q_p}(3\alpha''_l).$$

Hence,

$$c(j, i(p)) \leq 2t_p + \min_{l \in Q_p}(3\alpha''_l) \leq 2t_p + 3\alpha''_j \leq 6\alpha_j + 3\alpha''_j,$$

where the last inequality follows from Lemma 7. □

Let $L \subseteq Q$ denote the set of demands that are not contained in any open cluster.

Lemma 12. *The cost of connecting all demands that are not contained in any open cluster to the facility of their closest center is at most $\sum_{j \in L} 6\alpha_j + 3\alpha''_j$.*

Proof. Let $j \in L$. By arguments similar to those in the proof of Lemma 11, we can identify a facility $i \in F'$ with $c(j,i) \leq 3\alpha''_j$. Moreover, due to Lemma 7 there is an open center p with $c(j,p) \leq 3\alpha_j$. Let $L(p)$ denote the demands in L whose closest center is p. Then, $c(p,i(p)) \leq \min_{l \in L(p)}(3\alpha_l + 3\alpha''_l)$. We conclude that

$$c(j,i(p)) \leq 3\alpha_j + \min_{l \in L(p)}(3\alpha_l + 3\alpha''_l) \leq 6\alpha_j + 3\alpha''_j.$$

□

Lemma 13. *The total cost of buying the shortest path between a center p and its facility $i(p)$ is at most $\sum_{j \in Q_p} 3\alpha_j + 3\alpha''_j$.*

Proof. Each cluster contains at least M demands. The cost of buying the shortest path between p and $i(p)$ is

$$M \cdot c(p,i(p)) \leq M \cdot (t_p + \min_{l \in Q_p}(3\alpha''_l)) \leq \sum_{j \in Q_p} t_p + 3\alpha''_j \leq \sum_{j \in Q_p} 3\alpha_j + 3\alpha''_j,$$

where the first inequality was already derived in the proof of Lemma 11 and the last inequality follows from Lemma 7. □

Lemma 14. *The cost of the solution constructed is at most $\sum_{j \in Q} 9\alpha_j + 6\alpha'_j + 9\alpha''_j$.*

Lemma 15. *Every feasible solution to the connected facility location problem on Q has cost at least* $\max\left(\sum_{j \in Q}\alpha_j, \frac{1}{2}\sum_{j \in Q}\alpha'_j, \sum_{j \in Q}\alpha''_j\right)$.

Proof. The lower bound proofs for $\sum_{j \in Q}\alpha_j$ and $\sum_{j \in Q}\alpha'_j$ are given in [7]. Moreover, $\sum_{j \in Q}\alpha''_j$ is a lower bound on the cost of a feasible solution for the facility location game even without the requirement that the open facilities are connected; see [7]. □

Theorem 2. *The cost sharing method ξ, which for each $Q \subseteq D$, $j \in Q$ is defined as $\xi_j(Q) = \frac{3}{10}\alpha_j(Q) + \frac{1}{5}\alpha'_j(Q) + \frac{3}{10}\alpha''_j(Q)$, is cross-monotonic, competitive, and recovers at least a $\frac{1}{30}$-fraction of the optimal cost.*

References

[1] N. R. Devanur, M. Mihail, and V. V. Vazirani, *Strategyproof cost-sharing mechanisms for set cover and facility location games*, Proceedings of the 4th ACM Conference on Electronic Commerce, 2003, pp. 108–114.

[2] J. Edmonds, *Optimum branchings*, Journal of Research of the National Bureau of Standards **71B** (1967), no. 4, 233–240.

[3] A. Gupta, A. Kumar, and T. Roughgarden, *Simpler and better approximation algorithms for network design*, Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, 2003, pp. 365–372.

[4] A. Gupta, A. Srinivasan, and É. Tardos, *Cost-sharing mechanisms for network design*, Proceedings of the Seventh International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2004, to appear.

[5] K. Jain and V. V. Vazirani, *Applications of approximation algorithms to cooperative games*, Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, 2001, pp. 364–372.

[6] H. Moulin and S. Shenker, *Strategyproof sharing of submodular costs: budget balance versus efficiency*, *http://www.aciri.org/shenker/cost.ps*, 1997.

[7] M. Pál and É. Tardos, *Group strategyproof mechanisms via primal-dual algorithms*, Proceedings of the Forty-Fourth Annual IEEE Symposium on Foundations of Computer Science, 2003, pp. 584–593.

[8] C. Swamy and A. Kumar, *Primal-dual algorithms for connected facility location problems*, Proceedings of the Fifth International Workshop on Approximation Algorithms for Combinatorial Optimization, 2002, pp. 256–270.

[9] V. V. Vazirani, *Approximation algorithms*, Springer, 2001.