I N F O R M A T I K

Cross-Monotonic Cost-Sharing Methods for Connected Facility Location Games

Stefano Leonardi
Guido Schäfer

MPI–I–2003–1–017 September 2003

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Authors' Addresses

First author:
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00196 Roma, Italy
Email: leon@dis.uniroma1.it

Second author:
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
Email: schaefer@mpi-sb.mpg.de

Acknowledgements

The second author was supported by DFG graduated studies program "Quality Guarantees for Computer Systems" (GK 623), Saarbrücken.

Abstract

We present cost sharing methods for connected facility location games that are cross-monotonic, competitive, and 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 for various NP-hard problems. We also provide a primal-dual cost sharing method for the connected facility location game with opening costs.

Keywords

game theory, approximation algorithms, connected facility location problem, cost sharing method, cross-monotonic

1 Introduction

We study the problem of sharing the cost of a jointly used service among a set of participating users. Suppose we are given a set U of (potential) users that want to utilize a common service, e.g., meals on wheels.[1] Each user $j \in U$ has a *utility* u_j, which corresponds to the price she is willing to pay for receiving the service. If j is asked to pay more than u_j, she prefers not to receive the service. For each set $Q \subseteq U$ of users, the cost of servicing all users in Q is denoted by $C(Q)$. 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 as the set of participating users increases. 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 [3]: 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 utility, stop. Otherwise, remove from Q all users whose cost shares are larger than their utilities, and repeat.

In this paper, we are interested in cost sharing methods that are cross-monotonic, and satisfy *competitiveness* and *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 $1/\varepsilon$ of the cost of the constructed solution is recovered: $\sum_{j\in Q} \xi_j(Q) \geq C(Q)/\varepsilon$. We call such a cost sharing method a $1/\varepsilon$-*approximate cost sharing method*.

Contributions of this Paper. In this paper, we present cross-monotonic cost sharing methods for the connected facility location game.

We present a novel idea of sharing the expected cost of a randomized algorithm. Recently, Gupta, Kumar and Roughgarden [1] 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. Besides some effort, we were not able to

[1] Another widely known example is the Internet.

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.

Moreover, we extend the recent result of Pál and Tardos [4] to the connected facility location game with opening costs: We give a $\frac{1}{30}$-approximate cross-monotonic cost sharing method for this problem.

2 Problem Definitions

In the *connected facility location problem* (*CFL*) we are given an undirected graph $G = (V, E)$ with non-negative edge costs c_e, 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, 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 cost, the connection cost, and M times the Steiner 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, i.e., $\mathcal{F} = V$, and all opening costs are zero: $f_i = 0$ 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 [1] 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.

We use SIMPLECFL(Q) to define a random cost share $\alpha_j(Q)$ for each user $j \in Q$, and prove that the *expected* cost shares, $\xi_j(Q) = \frac{1}{4}\mathbf{E}[\alpha_j(Q)]$, 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. We compute the minimum spanning tree on $G(F)$ by running Edmonds' primal-dual algorithm 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 [2] 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 SIMPLECFL(Q'), when the set of open facilities is $F' \subseteq Q'$. Assume that we add a demand k to Q', and 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 Q. Conditioned on the event that for 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 presence of k can only decrease the shortest path distance from j to 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 $O \subseteq Q'$,

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

□

3.3 Computability

The next lemma states that the cost share of a demand j can be computed efficiently if $j \notin F$.

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,

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

□

On the other hand, we are not able to efficiently compute j's cost share if $j \in F$.

3.4 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. [1] 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 [4] gave a 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 such 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. 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 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 [5], 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 user 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*, i.e., inclusion maximal subsets 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 user 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 user 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 users that contribute to the opening of i at time t_i.

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

$$\alpha_j(Q) = t_j,$$
$$\alpha'_j(Q) = t_j + \int_{t_j}^{t'_j} M/s_j(t)dt, \text{ and}$$
$$\alpha''_j(Q) = \min\left(\min_{i:j \in Q_i} t_i, \min_{i:j \notin Q_i} c(j,i)\right).$$

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

Proof. Pál and Tardos [4] proved that α, α', and α'' are cross-monotonic. □

4.2 Algorithm

We run the ghost process but take the following rules into account in order to decide which locations and facilities are 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 around open locations 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 the 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. Same as for Lemma 5. □

The following two lemmas and their proofs are the same as in [4].

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 $c(j,p) \leq 3\alpha_j$.*

Lemma 8. *The cost of the Steiner tree on the centers is at most $\sum_{j \in Q} 6\alpha'_j$.*

Lemma 9. *Let $j \in Q_i$ for some open facility i, then $t_i \leq 3\alpha''_j$. For each demand j that does not belong to any set Q_k with open facility k, there is an open facility i such that $c(j,i) \leq 3\alpha''_j$.*

Proof. Same as for Lemma 7. □

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

Proof. At time t_i, we have $f_i = \sum_{j \in Q_i}(t_i - c(j,i))$. The lemma follows from Lemma 9. □

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 with $j \in Q_i$, we have $c(j,i) \leq t_i \leq 3\alpha''_j$ by Lemma 9. Otherwise, if j is not contained in any set Q_k with open facility k, by Lemma 9 there exists an open facility i with $c(j,i) \leq 3\alpha''_j$. That is, for each $j \in Q_p$, there exists an open facility within distance at most $3\alpha''_j$. Since $i(p)$ is an open facility that is closest to p, we have

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

Hence,

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

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 an open facility i 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, $c(p,i(p)) \leq \min_{j \in L(p)}\{3\alpha_j + 3\alpha''_j\}$. We conclude that

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

□

Lemma 13. *The total cost of buying the shortest path between a center p and 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

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

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 [4]. 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 (cf. [4]). □

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

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

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

[4] M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of the Fourty-Fourth Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

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

INFORMATIK

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nürnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nürnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	Subsumption of concepts in $DL\ \mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-017	G. Schäfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schäfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems”
MPI-I-2003-1-015	A. Kovács	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schäfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint
MPI-I-2003-1-012	D. Fotakis, R. Pagh, P. Sanders, P. Spirakis	Space Efficient Hash Tables with Worst Case Constant Access Time

MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Soon to be published
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vöcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vöcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Träff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schäfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Träff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches
MPI-I-2001-1-006	T. Polzin, S. Vahdati	Partitioning Techniques for the Steiner Problem
MPI-I-2001-1-005	T. Polzin, S. Vahdati	On Steiner Trees and Minimum Spanning Trees in Hypergraphs