Available online at www.sciencedirect.com
SCIENCE DIRECT®
Decision Support Systems 39 (2005) 11–22
Decision Support Systems
www.elsevier.com/locate/dsw

Strategyproof cost-sharing mechanisms for set cover and facility location games

Nikhil R. Devanur*, Milena Mihail, Vijay V. Vazirani

College of Computing, Georgia Institute of Technology. 801, Atlantic Dr., Atlanta, GA 30332, United States

Available online 23 September 2004

Abstract

Strategyproof cost-sharing mechanisms, lying in the core, that recover $1/\alpha$ fraction of the cost, are presented for the set cover and facility location games: $\alpha=O(\log n)$ for the former and 1:861 for the latter. Our mechanisms utilize approximation algorithms for these problems based on the method of dual-fitting.

Keywords: Internet; Cooperative games; Cost-sharing; Core; Strategyproof; Duality

1. Introduction

Achieving truth-revealing, also called strategyproofness or incentive compatibility, is fundamental to mechanism design. In a cost sharing mechanism, the goal is to distribute the cost of a shared resource among its users in such a way that revealing true utility is a dominant strategy of users. Other considerations include budget balance: that the users are not charged in excess of the incurring cost, at the same time recovering as much of the cost as possible. Ideally, one would like to recover all the cost, but this may not always be possible. Instead, the mechanism is required to recover a $1/\alpha$ fraction of the cost, with α being a measure of how good the mechanism is.

In this paper, we consider cost functions that are defined as optimization problems. For instance, in the set cover (facility location) game, the cost is defined by the optimum solution to a set cover (facility location) problem. The set cover problem is a versatile optimization problem and can be used to model many situations. It is one of the fundamental problems in optimization and approximation algorithms. The facility location problem has been widely studied in Operations Research and also in the context of network design.

A recent paper of Nisan and Ronen [17] and one by Lehmann et al. [9] also considered cost functions defined as optimization problems, though in the setting of auctions, rather than cost sharing. Indeed, they even dealt with situations in which the underlying optimization problems were NP-hard, by

* Corresponding author.
E-mail addresses: nikhil@cc.gatech.edu (N.R. Devanur), mihail@cc.gatech.edu (M. Mihail), vazirani@cc.gatech.edu (V.V. Vazirani).

0167-9236/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2004.08.004

resorting to methods from the field of approximation algorithms. Much work has been done on obtaining strategyproof cost sharing mechanisms—for instance for the spanning tree game [1,5,6,13,14]. Once again, the underlying optimization problems of some of the interesting games are NP-hard, and strategyproof cost allocation for several such games have been studied in Refs. [2–4,8,19], again using methods from approximation algorithms.

Most of the work related to cost sharing mechanisms in general concentrate on achieving the much harder task of group-strategyproofness (i.e., the agents have no incentive to lie even if collusions are allowed). Moulin and Shenker [16] showed that one way to get a group-strategyproof cost sharing mechanism is to construct a cross-monotone cost sharing method. They also showed that it is, in fact, the only way. This severely restricts the class of group-strategyproof mechanisms. In this paper, we show that relaxing this condition to individual strategyproofness results in simpler and better mechanisms.

Another important concept in the framework of cooperative game theory is that of a core. An allocation is in the core if it ensures that no subset of agents have an incentive to secede, i.e., no subset of the agents is charged more than the stand alone cost of serving that subset. Indeed, a budget balanced group-strategyproof mechanism is always in the core; the same is not true for a strategyproof mechanism. Hence, we impose this condition additionally. However, we consider a weaker version of core that ensures that no subset of users that actually obtain the shared resource have an incentive to secede, as opposed to the standard definition that ensures the same for all possible subsets.

In this paper, we obtain strategyproof cost allocations in the core that recover $1/\alpha$ fraction of the cost, for two fundamental games whose underlying optimization problems are NP-hard, the set cover game and the facility location game. For the former, $\alpha=O(\log n)$, and for the latter, $\alpha=1:861$. For the latter game, this is made possible by new approximation algorithms for the underlying optimization problem using the technique of dual fitting [11]. In retrospect, the natural greedy algorithm for the set cover problem (see Ref. [20]) can also be analyzed using this technique. We utilize this viewpoint for handling the set cover game. The facility location game was studied in Ref. [4,8], who left the open problem of obtaining a group-strategyproof mechanism based on a constant factor approximation algorithm. Our paper partially answers this question. We give a strategyproof mechanism. Subsequently, Pal and Tardos [18] gave a cross-monotonic cost-sharing method for the facility location problem that recovers one-third of the cost. This gives a group-strategyproof mechanism for the facility location game that recovers one-third of the cost. However, recently, Immorlica et al. [7] have shown that no cross-monotone cost sharing method for the facility location game can recover more than one-third of the cost. They also show a similar bound of $1=n$ for the set cover game. In light of these results, our paper shows that relaxing group-strategyproofness to strategyproofness indeed gives mechanisms that recover a larger fraction of the cost.

In fact, our technique seems to be quite general. Towards this, we show that our technique also extends to a game defined by a variation of the set cover problem, called the set multicover problem (under certain assumptions on the utilities). Our technique also speeds up the Moulin-Shenker mechanism for submodular cost functions, using the cost sharing method defined by Jain and Vazirani [11] by a factor of n.

1.1. Organization

In Section 2, we formally define the set cover game and give the mechanism. We also extend the mechanism to the set multicover game. In Section 3, we do the same for the facility location game. Section 4 shows how our technique yields essentially the same mechanism as that of Ref. [11] for submodular cost functions.

2. The set cover cost sharing game

Let N be a set of bidders. For each coalition $S \subseteq N$ the cost of providing a service to the bidders in S is $C(S)$. How do you share $C(N)$ among all the bidders?

Definition 1. Given the set of bids, $b_1, b_2, \ldots, b_n$ a cost sharing mechanism computes

(1) the set of successful bidders, A, who are provided the service, and
(2) the amount charged to each bidder, $x_1, x_2, \ldots, x_n$.

An important consideration is the representation of the costs. A natural and interesting (to the computer scientist) case is when the cost is given by a solution to an optimization problem.

Definition 2 (Set cover problem). Given a universal set U, and a collection of subsets of U, $T=\{S_1, S_2, \ldots, S_k\}$, and a cost function c: $T \rightarrow Q^+$, find a minimum cost sub collection of T that covers all the elements of U.

Given an instance of the set cover problem over the universal set N, the cost of providing the service to a coalition S is the cost of the optimal subcollection of T that covers all the elements in S. In order to make the output meaningful, we impose the following well-known constraints on the mechanism:

(1) *Polynomial time (PT)*: we require that the mechanism run in time that is polynomial in the input size.

(2) *Strategyproofness*: each bidder i has a privately known utility value, u_i, which is the maximum he is willing to pay for the service. If he is charged x_i, then his profit, or benefit is $u_i - x_i$. Assume that the bidders are selfish, i.e. they are only interested in maximizing their benefit, and nothing else. We say that a mechanism is strategyproof, if for each bidder i, his profit is maximized by bidding u_i, for all choices of bids for other bidders. (When a strategy is such that it maximizes the profit for all possible values of others bids, it is called a dominant strategy.) In case of a strategyproof mechanism, truth telling is a dominant strategy.

(3) *Budget balance (BB)*:

(a) *Costrecovery* $\Sigma_{i \in A} x_i \geq C(A)$, i.e., the cost of providing the service is recovered from all the bidders.
(b) *Competitiveness* $\Sigma_{i \in A} x_i \leq C(A)$, i.e., no surplus is created. Because if any surplus is created then a competitor can provide the service at a cheaper cost by reducing the surplus.

The condition of budget balance consists of satisfying both, cost recovery and competitiveness, i.e., $\Sigma_{i \in A} x_i = C(A)$ (the set of bidders receiving the service pay exactly the total cost of T).

Approximate budget balance: we obtain the notion of α-approximate budget balance by relaxing the cost recovery condition to $\Sigma_{i \in A} x_i \geq (1/\alpha) C(A)$.

(4) *Core*: an allocation is said to be in the core, if for every subset of users in A, the total amount charged to the users in the subset is no more than the cost of serving the subset alone, i.e.,

$$\forall S \subseteq A, \sum_{i \in S} x_i \leq C(S).$$

Note that this definition is weaker than the standard definition of core. This definition only forbids coalitions that are subsets of A (the set of users that obtain the service), where as the standard definition forbids all coalitions.

(5) *No positive transfer (NPT)*: for each bidder i, $x_i \geq 0$, i.e., bidders will not be paid for receiving a service.

(6) *Voluntary participation (VP)*: if $i \notin A$ then $x_i = 0$ and if $i \in A$ then $x_i \leq b_i$, i.e., only those bidders will pay who will receive the service. There is no "entrance fee" for the mechanism. Moreover, they will never be asked to pay more than their reported utilities. In other words, each bidder has the option to not receive the service, and if so, derives a benefit of 0.

(7) *Consumer sovereignty (CS)*: every bidder is guaranteed to receive the service if she reports a high enough utility value. Moreover, this value should be independent of the bids of other bidders. This condition forbids the trivial solution of not covering anyone. It also prevents the mechanism from being biased against any particular bidder.

To summarize, given a set of bidders, N, a collection T of subsets of N, and the bids of the bidders, a set cover cost sharing mechanism computes the set of successful bidders, A, the price charged to each bidder, and a subcollection of T that covers A, such that all the above constraints are satisfied.

2.1. The mechanism

A rule of thumb for designing strategyproof mechanisms is that the amount charged to bidder i should be independent of his bid b_i. The bid b_i only decides whether the bidder gets covered or not. The main idea of the mechanism is to try to cover the bidders with as little cost as possible.

Start with a target cost share of zero for all bidders. Raise the cost shares of all the bidder at the same rate. As soon as someone's cost share exceeds his bid, he can be discarded from further considerations. And as soon as some bidders (who have not already been covered) are collectively able to pay for a $1/H_n$ (H_n=1+1/2+1/3+ · ·+1/ n) fraction of the cost of a set in T, pick that set to be in the cover. These bidder get covered at their current cost shares. Continue to raise the cost shares of others, until everyone either gets covered, or is discarded.

Algorithm 1 (*Mechanism for set cover*).

Input: An instance of the set cover problem, $N, T, c : T \rightarrow \mathbf{Q}^+$ and the bids $b_i, \forall i \in N$
Output: The set of bidders to be served, A, and the prices charged to them, $x_i \forall i \in A$

$A \leftarrow \emptyset$;
$\forall\ i,\ x_i = 0$;
while $A \neq N$ **do**
  Raise all x_i's in $N \setminus A$ continuously at the same rate until one of the two events happens:
  if $x_i > b_i$ **then**
    $N \leftarrow N \setminus i$;
  end
  if *for some* j, $C(S_j) = \left(\sum_{i \in (N \cap S_j) \setminus A} x_i\right) \times H_n$ **then**
    $A \leftarrow A \cup (N \cap S_j)$;
  end
end

We digress here and give some background on the dual fitting-based analysis for the greedy set cover algorithm. We briefly describe here what is explained in more detail in Ref. [20]. The connection between the above mechanism and dual fitting will be clear shortly.

Formulate the set cover as an integer program as follows: let y_j be a 0/1 variable that denotes whether the set $S_j \in T$ is picked in the cover or not. So the objective is

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{k} c(S_j) y_j \\ \text{subject to} & \sum_{j: i \in S_j} y_j \geq 1, \quad \forall\ i \in N. \\ & y_j \in \{0, 1\}, \quad \forall\ j = 1, 2, \ldots, k \end{array} .$$

The LP-relaxation of this integer program is obtained by letting the variables y_j's take any value between 0 and 1.

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{k} c(S_j) y_j \\ \text{subject to} & \sum_{j: i \in S_j} y_j \geq 1, \quad \forall\ i \in N. \\ & y_j \geq 0, \quad \forall\ j = 1, 2, \ldots, k \end{array} .$$

The dual of this LP is

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in N} \alpha_i \\ \text{subject to} \quad & \sum_{i: i \in S_j} \alpha_i \le c(S_j), \quad \forall\, j = 1, 2, \ldots, k.. \\ & \alpha_i \ge 0, \quad \forall\, i \in N \end{aligned}$$

The greedy set cover algorithm: (see Ref. [20] for a detailed description). Iteratively pick the most cost effective set and remove the covered elements until all the elements are covered. Cost effectiveness of a set is defined as the average cost at which it covers new elements, i.e. the cost of that set divided by the number of uncovered elements in it. Also whenever a set S_j is picked, set the x_i of all new elements covered by S_j to be the cost effectiveness of S_j. So the greedy algorithm not only gives a solution to the LP, but also a solution to the dual. The objective function value of the primal solution is equal to that of the dual solution (because we are just re-distributing the cost of picking a set to the α_i' s). Note, however, that $\{\alpha_i\}_{i=1}^{n}$ is not feasible in the dual. The approximation guarantee follows from showing that $x_i = \alpha_i / H_n$ is feasible.

$$\sum_{j=1}^{k} c(S_j) y_j = \sum_{i \in N} \alpha_i \tag{1}$$

$$\sum_{j=1}^{k} c(S_j) y_j = H_n \sum_{i \in N} x_i \tag{2}$$

$$\sum_{j=1}^{k} c(S_j) y_j \le H_n \text{ (Optimal solution to the LP)} \tag{3}$$

$$\sum_{j=1}^{k} c(S_j) y_j \le H_n \text{ (Optimal solution to the integer program)} \tag{4}$$

The main observation here is that the dual variables x_i can also be interpreted as cost shares. The following implementation shows that the mechanism can be seen as a modification of the greedy set cover algorithm. Find the most cost effective set in T. The cost of that set is divided equally among the users in that set (that are not yet covered) and scaled down by a factor of H_n. If everyone in that set can afford their cost shares, then pick that set and continue. Otherwise, discard all those who cannot afford their cost shares and continue with the rest. Also, whenever cost effectiveness is calculated, the bidders who are already covered and those who are discarded are ignored.

Lemma 1. *The above mechanism returns a solution that is at most O(log n) times the optimum set cover of A. Also it is O(log n)-approximate budget balanced.*

Proof. From (in)equalities (1), (2), (3), (4), it is enough to prove that the cover returned by the mechanism is exactly the same as the one that the greedy set cover algorithm would return with A as the universal set.

Consider the first set picked by the greedy algorithm, say S. It is also the first set picked by the mechanism. Suppose not, and the mechanism picked some other set first. Since all the bidders in that set would also be there in A, it is more cost effective than S, a contradiction. Similarly, all the sets picked by the greedy algorithm are exactly the ones picked by the mechanism. Hence, they return identical solutions. □

Lemma 2. *The mechanism returns a core allocation.*

Proof. Given any subset $S \subseteq A$, consider the LP and its dual restricted to the set S. Since $\{x_i\}_{i \in N}$ (where $x_i = \alpha_i / H_n$) is feasible for the dual corresponding to A, $\{x_i\}_{i \in S}$ is feasible for the dual corresponding to S. Hence, $\sum_{i \in S} x_i \le C(S)$. □

Clearly, the mechanism satisfies VP, NPT and CS. Intuitively, why is the mechanism strategyproof? Note the similarity of the mechanism with an English auction, where the bidders incrementally bid for a single item until all but one drop out. Here too, the cost shares of the bidders are always increasing. And once a set is picked, the cost shares of the bidders in that set are frozen. Also note that the greedy algorithm is such that if some bidders are dropped mid-way, one does not have to start all the way from the beginning.

Lemma 3. *The above mechanism is strategyproof.*

Proof. The mechanism does only comparison operations on the bids of bidders. Suppose that an bidder bids a value b_i less than u_i. If the bidder does not get covered, then there is nothing to prove. If the bidder gets covered, then it means that all the comparisons returned the bid as the higher value. Now if the bidder had bid u_i, then the result would have been the same, and hence the bidder would have got covered at the same charge.

Suppose an bidder bids a value b_i greater than u_i. Again, there is nothing to prove if the bidder does not get covered. If the bidder gets covered, at a charge greater than u_i, then the benefit to the bidder is negative, and hence, by VP, it is better to bid u_i. If the bidder gets covered at a charge less than u_i, then it is again indistinguishable from the case where the bidder bids u_i, and hence he would have got covered at the same price. □

To summarize,

Theorem 4. *The mechanism defined by Algorithm 1 is strategyproof, O(log n)-approximate BB, satisfies VP, NPT, CS, and is in the core.*

In strategyproofness, we assume that the bidders do not collude. In fact, there is a stronger notion which says that even if a set of bidders collude, the dominant strategy of all the bidders is to bid their true utility value.

Definition 3. Suppose that a coalition of bidders misreport their utility such that the profit of each bidder in the coalition does not decrease, but some bidder gets a higher profit. Call it a successful coalition. A mechanism is group-strategyproof if no coalition is successful.

The above mechanism is not group-strategyproof. Consider the example of three bidders, {1,2,3} with T={{1,2},{2,3},{3,1}} with the costs $2+\epsilon$, 2 and $2+\epsilon$, respectively. The true utilities of 1, 2 and 3 are 2, 2 and 1, respectively. If all the bidders bid their true utilities, then bidders 2 and 3 get the service at cost 1 each. However, the benefit to bidder 3 is zero, since his cost share equals his true utility. Now if bidders 1 and 3 collude, and bidder 3 reports his utility as $1-\epsilon$, then bidders 1 and 2 get the service at a cost $1+\epsilon/2$. Hence, bidder 1 gets a higher benefit, while bidder 3's benefit does not go down. However, note that, if we require that each bidder in the coalition should get a strictly higher benefit (weakly group-strategyproof), then the above example does not work. In fact, it is easy to see that the mechanism is weakly group-strategyproof. It is implicit in the proof of strategyproofness that a bidder never gets a higher benefit by lying, no matter what the other bidders bid. However, he may get the same benefit by lying, while helping some other bidder get a higher benefit.

Moulin and Shenker [16] give a general method to get a group strategyproof cost sharing mechanism, conditioned on the existence of a certain kind of method to distribute the cost.

Definition 4. A cost sharing method is a function ξ, which, given a subset of bidders S, distributes the cost of serving the subset among all the bidders in it. That is, $\Sigma_{i \notin S}\xi(i,S)=C(S)$. And $i \notin S \Rightarrow \xi(i,S)=0$.

Definition 5. A cost sharing method is cross-monotonic if for all $S \subseteq S'$, $\xi(i,S) \geq \xi(i,S')$ $\forall i \in S$.

Given any cross-monotonic cost sharing method, Ref. [16] gives a budget-balanced group-strategyproof mechanism that satisfies NPT, VP and CS. In fact, all the known group-strategyproof mechanisms are Moulin-Shenker mechanisms. The set cover problem does not always admit a cross-monotone cost sharing method. In fact, Ref. [7] shows that no cross-monotone cost sharing method can always recover $\Omega(1/n)$ fraction of the cost.

2.2. Variation of set cover

So far, we assumed that the bidders want to get covered only once. Here, we consider the generalization that the bidders might want to get covered multiple times. Assume that each bidder i has a utility u_{ij} for getting covered the jth time. The bidders submit bids for getting covered the first time, as before. Every time a bidder is covered, he submits a bid for getting covered again. This bid may be different than the ones submitted before.

Note that this does not fit the prototype of the cost sharing mechanism defined in Definition 1. In particular, the bidders' utilities are multidimensional in this game. Consumer sovereignty now means that for each i, the bidder is guaranteed to be covered the ith time if he bids high enough. Also, the definition of core needs to be changed: let bidder $i \in A$ be covered r_i times by the mechanism. An allocation is in the core, if the total amount charged to a subset S of users in A is no more than the cost of covering each bidder $i \in S$, r_i times.

Our mechanism extends naturally to this game. Unfortunately, it is strategyproof only under the extra assumption of decreasing marginal utilities. That is, for each bidder i and for all $j' \geq j \geq 1$, $u_{ij} \geq u_{ij'}$. The mechanism is as before, except that it continues to offer to cover the bidders who have already been covered.

As before, this mechanism can be seen as a modification of the greedy algorithm for the set multicover problem (see Ref. [20]) where in addition to the usual set cover instance, each element is required to be covered a given number of times.

Note that it is possible that a bidder may lie about his utility for the first time in the hope that he gets covered twice and thus gain a positive benefit. However, the additional cost share for getting covered each subsequent time is only increasing. Hence, under the assumption of decreasing marginal utilities, we get that the mechanism is strategyproof.

The following is analogous to Theorem 4:

Theorem 5. *The mechanism defined by Algorithm 2 is O(log n)-approximate BB, satisfies VP, NPT, CS, and is in the core, under the assumption of decreasing marginal utilities.*

Algorithm 2 (*Mechanism for set multicover*).

Input: An instance of the set cover problem, $N, T, c : T \rightarrow \mathbf{Q}^+$ and the bids $b_i, \forall i \in N$
Output: The (multi)set of bidders to be served, A, and the prices charged to them

```
A ← ∅ ;
∀ i, x_i = 0;
while N ≠ ∅ do
    Raise all x_i's in N \ A continuously at the same rate until one of the two
    events happens:
    if x_i > b_i then
        N ← N \ i;
    end
    if for some j, c(S_j) = (Σ_{i∈(N∩S_j)} x_i) × H_n then
        A ← A ∪ (N ∩ S_j);
        prices charged to them for the current cover is x_i;
        x_i ← 0, ∀ i ∈ (N ∩ S_j);
        b_i ← new bids;
    end
end
```

3. Facility location

Definition 6 (Metric uncapacitated facility location): F is a set of facilities and C is a set of cities. Each facility i has an opening cost f_i and the cost of connecting a facility i with a city j is c_{ij}. The connection costs satisfy the triangle inequality. The problem is to find a subset of facilities to open, $I \subseteq F$, and a way to connect each city to an open facility, ϕ: $C \rightarrow I$ such that the total cost of opening the facilities and connecting cities to open facilities is minimized.

In the cost sharing problem, the cities are the bidders. The mechanism is given F, C, $\{f_i\}_{i \in F}$, $\{c_{ij}\}_{i \in F,\ j \in C}$ as above, along with the bids, $\{b_j\}_{j \in C}$. It is required to compute

(1) A set of facilities to open,
(2) The set of cities to be connected to each facility,
(3) The amount to be charged to each city that gets connected.

As in the set cover problem, the mechanism uses the underlying greedy algorithm in Refs. [11,15]. Consider the following IP-formulation of the facility location problem. Let us say that a star is a facility with several cities, (i,C') where $i \in F$, $C' \subseteq C$. The cost of a star (i,C') is $f_i + \sum_{j \in C'} c_{ij}$. The facility location problem is equivalent to picking a minimum cost of collection of stars such that each city is in at least one star. Let y_S be an indicator variable denoting whether star S is picked and c_S denote the cost of star S.

$$\begin{aligned} &\text{minimize} && \sum_{S \in \mathcal{S}} c_S y_S \\ &\text{subject to } \forall\, j \in C : && \sum_{S : j \in S} y_S \geq 1 \\ &\qquad \forall\, S \in \mathcal{S} : && y_S \in \{0,1\} \end{aligned}$$

The LP-relaxation of this program is:

$$\begin{aligned} &\text{minimize} && \sum_{S \in \mathcal{S}} c_S y_S \\ &\text{subject to } \forall\, j \in C : && \sum_{S : j \in S} y_S \geq 1 \\ &\qquad \forall\, S \in \mathcal{S} : && y_S \geq 0 \end{aligned}$$

The dual program is:

$$\begin{aligned} &\text{minimize} && \sum_{j \in C} \alpha_j \\ &\text{subject to } \forall\, S \in \mathcal{S} : && \sum_{j \in S \cap C} \alpha_j \leq c_S \\ &\qquad \forall\, j \in C : && \alpha_j \geq 0 \end{aligned}$$

The main technique is to interpret the dual variables as cost shares of the cities. The mechanism uses the underlying greedy algorithm [11,15], which raises the dual variables greedily until a primal feasible solution is obtained whose cost is equal to that of the duals. The approximation guarantee is then obtained using dual fitting, that is by showing that dividing the dual variables by a constant (here, 1.861) gives a feasible dual.

3.1. The mechanism

Unlike in the set cover problem, the cost share has to be accounted for connection costs as well as the facility opening costs. For an unconnected city j, if the cost share is $x_j = \alpha_j / 1.861$, then it offers $\max(0,\ \alpha_j - c_{ij})$ to the opening cost of a closed facility i, i.e., the money left over (if any) after paying for the connection cost. As before, start with a cost share of zero for all the cities, and raise the cost shares of all the unconnected cities at the same rate, until one of the following happens:

(1) If some city's cost share goes beyond its bid, then discard the city from all further considerations.
(2) If for some closed facility i, the total offer it gets is equal to the opening cost, then the facility i is opened, and every city j that has a non-zero offer to i is connected to i.
(3) If some unconnected city j, α_j is equal to its connection cost to an already opened facility i, then connect city j to facility i.

Continue doing this until all the cities are either connected or discarded. As in the set cover problem, if A is the set of cities served, then the resulting choice of facilities and connections is the same as that obtained by running the algorithm of Refs. [11,15] with A as the set of cities instead of C. Refs. [8,15] show that dividing the α_j's by 1.861 gives a dual feasible solution. This proves the lemmas analogous to Lemmas 1 and 2. Also the proof of strategyproofness is analogous to that of Lemma 3.

Theorem 6. *The mechanism defined by Algorithm 3 is strategyproof, 1.861-approximate BB, satisfies VP, NPT, CS, and is in the core.*

Remark 7. We can use the improved algorithm of Refs. [8,9] to get a better approximation ratio (of 1.61), but then the mechanism would not be in the core, since in that algorithm, the dual variables may pay for more than the primal objective function.

Like the set cover, the facility location also does not admit a cross-monotonic cost sharing method. Ref. [18] shows a method that recovers one-third fraction of the costs and Ref. [7] shows a matching lower bound.

4. Submodular cost functions

In this section, we deviate from the previous model and assume that the cost function is given by an oracle. We consider those cost functions that follow the economies of scale:

Definition 7. A cost function is submodular if

(1) $C(\emptyset)=0$,

Input: An instance of the facility location problem, $F, C, \{f_i\}_{i\in F}, \{c_{ij}\}_{i\in F, j\in C}$ and the bids, $\{b_j\}_{j\in C}$.
Output: The facilities to open, the cities to be connected to each facility and the amount charged to each city.

```
A ← ∅(connected cities) ;
∀ j, x_j = 0, α_j := 1.861 x_j;
while C ≠ A do
    Raise all x_j's in C \ A continuously at the same rate until one of the
    three events happens:
    if x_j > b_j then
        C ← C \ j;
    end
    Define offer(i, j) := max{0, α_j − c_ij};
    if for some closed facility i, Σ_j offer(i, j) = f_i then
        Open facility i;
        Connect to i each city with non zero offer to i;
        A ← A ∪ { cities connected to i};
    end
    if For j ∈ C \ A, i an open facility, α_j = c_ij then
        A ← A ∪ j;
        connect city j to facility i;
    end
end
```

(2) for any S, $T \subseteq N$, $C(S)+C(T) \geq C(S \cup T)+C(S \cap T)$.

Algorithm 3 (*Mechanism for facility location*).

The constraint 2 can also be replaced by

$$\forall S \subseteq T \subseteq N, \forall iT, C(S+i) - C(S) \geq C(T+i) - C(T).$$

Jain and Vazirani [11] give a primal-dual type algorithm (JV algorithm) to compute a cross-monotonic cost sharing method for submodular cost functions. This combined with the mechanism (MS mechanism) of Moulin and Shenker ([16]) gives a group strategyproof cost sharing mechanism when the cost is submodular. Our mechanism extends to this game and, in fact, gives the same output. Moreover, it is a factor $n(:=|N|)$ faster.

4.1. JV algorithm

Recall that, given any $S \subseteq N$, we need to compute $\xi(i,S) \geq 0$ for all $i \in S$ such that $\Sigma_{i \in S} \xi(i,S)=C(S)$. Start with $x_i=0$ for all $i \in S$. Say a set $S \subseteq N$ is tight if $C(S)=\Sigma_{i \in S} x_i$. Raise the cost shares x_i of bidders all at the same rate. Whenever a set goes tight, freeze the cost shares of all bidders in that set. Continue raising the cost shares of others until all the cost shares are frozen. $\xi(i,S):=x_i$. Ref. [11] proves the following:

Theorem 8. *At any time, there is a unique maximal tight set and it can be found in polynomial time.*

Theorem 9. *The cost sharing method ξ is cross-monotonic.*

4.2. MS mechanism

Given an cross-monotonic cost sharing method ξ, Ref. [16] gives a group-strategyproof cost sharing mechanism $M(\xi)$. The mechanism tries to serve all the bidders as a first step, by using ξ to determine the prices. If someone is not able to pay, i.e., his bid is less than the price, then the mechanism drops him and tries to serve the remaining bidders. It

```
A ← N ;
repeat
  | if ξ(i, A) > b_i then
  |   | A ← A \ i;
  | end
until ∀ i ∈ A, b_i ≥ ξ(i, A);
x_i ← ξ(i, A);
```

continues the same way until everyone left can afford their cost shares.

Algorithm 4 (*Mechanism $M(\xi)$*).

Theorem 10. *For any cross-monotonic cost sharing method ξ the mechanism $M(\xi)$ is BB, satisfies VP, NP, CS, and is group strategyproof* [16].

4.3. Our mechanism

```
A ← ∅ ;
∀ i, x_i = 0;
while A ≠ N do
    Raise all x_i's in N \ A continuously at the same rate until one of the two
    events happens:
    if x_i > b_i then
        N ← N \ i;
    end
    if for some S ⊂ N, C(S) = Σ_{i∈S} x_i then
        A ← A ∪ S;
    end
end
```

Our mechanism is similar to the one for set cover. Raise the cost shares uniformly. If a set goes tight, then we freeze their cost shares, and if a bidder's cost share exceeds his bid, then we drop him, and continue with the rest.

Algorithm 5 (*Mechanism M′ for submodular cost functions*).

It is easy to see that our mechanism is a factor n faster. The properties of BB, VP, NPT, CS and group-strategyproofness follow from the following theorem:

Lemma 11. *The mechanism M′ and M(ξ) give identical output.*

Proof. Since in mechanism $M(\xi)$ the order in which the bidders are dropped (in case there are many) does not matter, we drop them in the same order as M'. This is possible if, whenever we drop a bidder in M', we are allowed to drop him in $M(\xi)$ as well. If a bidder is dropped in M' then with the current N his bid is less than $\xi(i,N)$, and hence can be dropped in $M(\xi)$. So the set of users served is the same in the two mechanisms.

Note that by cross-monotonicity, the cost-shares of each bidder (not already dropped) in $M(\xi)$ is always increasing. Now suppose a set S goes tight at some time t_S in M'. By the algorithm, we know that at this point all the bidders in S can afford their cost shares, and that it does not go tight at any time before that. So it goes tight at the same time in all the runs of ξ. Hence, the cost shares of all the bidders are the same in the two mechanisms. □

Theorem 12. *The mechanism M′ is group-strategy-proof, BB, satisfies VP, NPT, CS, and is in the core and makes O(n) iterations.*

Note that our algorithm cannot be extended to the Steiner Tree game (considered in Ref. [10]) in which the cost is given as a solution to a problem of Steiner network. Intuitively, it is because the solution is changing each time ξ is run, whereas, in the set cover case, the solution given by the greedy algorithm remains the same.

Acknowledgments

We would like to thank Aranyak Mehta and Amin Saberi for useful discussions. We would also like to thank the anonymous referees for several useful comments.

References

[1] C. Bird, On cost allocation for a spanning tree: a game theoretic approach, Networks 6 (1976) 335–350.

[2] J. Feigenbaum, C. Papadimitriou, S. Shenker, Sharing the Cost of Multicast Transmission, In Proc. 32nd Ann. ACM Symp. on Theory of Comp., 2000.
[3] A Fiat, A Goldberg, J Hartline, A Karlin, Competitive Generalized Auctions, In Proc. 34th Ann. ACM Symp. on Theory of Comp., 2002.
[4] M. Goemans, M. Skutella, Cooperative Facility Location Games, In Proc. 11th annual Symposium on Discrete Algorithm, 2000, pp. 76–85.
[5] D. Granot, G. Huberman, On minimum cost spanning tree games, Mathematical Programming 21 (1981) 1–18.
[6] D. Granot, G. Huberman, On the core and nucleolus of minimum cost spanning tree games, Mathematical Programming 29 (1984) 323–347.
[7] N. Immorlica, M. Mahdian, V.S. Mirrokni, Lower bounds for cost sharing and group-strategyproof mechanisms, Manuscript (2004).
[8] K. Jain, V.V. Vazirani, Applications of Approximation Algorithms to Cooperative Games, In Proceedings of 33rd ACM Symposium on Theory of Computing, 2001.
[9] K. Jain, V.V. Vazirani, Equitable Cost Allocations Via Primal-Dual-Type Algorithms, In Proceedings of 33rd ACM Symposium on Theory of Computing, 2002.
[10] K. Jain, M. Mahdian, A. Saberi, A new greedy approach for facility location problem, In Proceedings of 34th ACM Symposium on Theory of Computing, 2002.
[11] K. Jain, M. Mahdian, E. Markakis, A. Saberi, V. Vazirani, Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP, To appear in Journal of the ACM.
[12] D. Lehmann, L.I. O'Callaghan, Y. Shoham, Truth Revelation in Approximately Efficient Combinatorial Auctions, Journal of the ACM 49 (5) (2002 September) 577–602.
[13] K. Kent, D. Skorin-Kapov, Population Monotonic Cost Allocation on Mst's, Operational Research Proceedings KOI, 1996, pp. 43–48.
[14] K. Kent, D. Skorin-Kapov, Distance Monotonic Stable Cost Allocation Scheme for the Minimum Cost Spanning Tree Network, In Proceedings of the 19th Conference Information Technology Interfaces ITI, 1997, pp. 463–468.
[15] M. Mahdian, E. Markakis, A. Saberi, V. Vazirani, A Greedy Facility Location Algorithm Analyzed Using Dual Fitting, RANDOM-APPROX, 2001, pp. 127–137.
[16] H. Moulin, S. Shenker, Strategyproof sharing of submodular costs: budget balance versus efficiency, Economic Theory 18 (2001) 511–533.
[17] N. Nisan, A. Ronen, Algorithmic Mechanism Design, In Proc. 31st ACM Symp. on Theory of Computing, 1999, pp. 129–140.
[18] M. Pal, E. Tardos, Strategyproof Mechanisms Via Primal-Dual Algorithms, In Proc. FOCS, 2003.
[19] T. Roughgarden, E. Tardos, How Bad is Selfish Routing? In Proc. 41st IEEE Symp. on The Foundations of Comp. Sc., 2000, pp. 93–102.
[20] V.V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin, 2001.

Nikhil Devanur got his undergraduate degree in Computer Science and Engineering from the Indian Institute of Technology, Bombay, in 2001. He then joined the College of Computing at Georgia Tech. He was the College's Dean's Fellow during 2001–2002. His research has been focussed on efficient computation of market equilibria, and algorithmic aspects of game theory, arising in the context of the Internet, and other huge networks.
In joint work with some of his co-authors, he gave the first polynomial time algorithm for computing market equilibrium when buyers have linear utility functions. He is expected to graduate with a PhD in 2006 under the guidance of Dr. Vijay Vazirani.

Milena Mihail holds a Dimploma in Electrical Engineering form the National Technical University of Athens and a PhD in Computer Science from Harvard University. She has been a member of the technical staff and a manager at Bell Communications Research and has been an Associate Professor at Georgia Tech since 1999.

Vijay Vazirani got his Bachelor's degree in Computer Science from MIT in 1979, and his PhD from U.C. Berkeley in 1983. He is currently a Professor of Computer Science at Georgia Tech. His research career has been centered around the design of algorithms, together with work on complexity theory, cryptography, coding theory, and game theory. During the first 10 years of his career, he made seminal contributions to the classical maximum matching problem which has historically played a central role in the development of the theory of algorithms. In the 1990s, he had much influence on the theory of approximation algorithms through work on several of its fundamental problems. In 2001, he published a book on this topic with Springer-Verlag. Over the last few years, he has been working in the nascent area of algorithmic game theory which addresses new game theoretic/computational issues arising from the Internet.