Strategyproof Cost-sharing Mechanisms for Set Cover and Facility Location Games

Nikhil R. Devanur
nikhil@cc.gatech.edu

Milena Mihail
mihail@cc.gatech.edu

Vijay V. Vazirani
vazirani@cc.gatech.edu

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

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Problem Complexity**]: General; G.0 [**Mathematics of Computing**]: General

General Terms

Algorithms, Theory

Keywords

Cost Sharing, Mechanism Design, Set Cover, Facility Location

1. INTRODUCTION

Achieving truth-revealing, also called strategyproofness or incentive compatibility, is fundamental to game theory. The seminal work of Vickery showed a way of achieving this for single item auctions. An extension of Vickery idea generales the framework of auctions to the design of strategyproof mechanisms. This goes under the name of Vickery-Clarke-Groves (VCG) mechanism, and is perhaps the single most influential idea in mechanism design.

VCG mechanisms inherently fit into the realm of optimization problems and hence give rise to inherent algorithmic issues – this insight was provided by the recent important paper of Nisan and Ronen. Indeed, they even dealt with situations in which the underlying optimization problems were **NP**-hard, by resorting to methods from the field of approximation algorithms.

Strategyproofness also plays an important role in cooperative game theory – when the cost of a shared resource is to be distributed among its users in such a way that revealing true utility is a dominant strategy of users. Much work has been done on obtaining strategyproof cost sharing mechanisms – for instance for the spanning tree game [1, 5, 6, 11, 12]. 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 [3, 4, 9, 16, 2], again using methods from approximation algorithms.

In this paper, we obtain strategyproof cost allocations for two fundamental games whose underlying optimization problems are **NP**-hard, the set cover game and the facility location game. For the latter game, this is made possible by new approximation algorithms for the underlying optimization problem using the technique of dual fitting [7]. In retrospect, the natural greedy algorithm for the set cover problem (see [17]) can also analyzed using this technique – we utilize this viewpoint for handling the set cover game. The facility location game was studied in [9, 4], 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, but cannot achieve group strategyproofness. More recently, Pal and Tardos [15] have announced a 3-approximately budget balanced cross-monotonic cost-sharing method for the facility location problem. This gives a group strategyproof mechanism for the facility location game that recovers $\frac{1}{3}^{\text{rd}}$ of the cost.

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 $c : T \to \mathbf{Q}^+$, 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

EC'03, June 9–12, 2003, San Diego, California, USA.

is the cost of the optimal sub collection 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. **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 maximises the profit for all possible values of others bids, it is called a *dominant strategy*) Here, truth telling is a dominant strategy.

2. **Budget Balance(BB):**

   (a) **Cost Recovery** $\sum_{i \in A} x_i \geq c(A)$, i.e., the cost of providing the service is recovered from all the bidders.

   (b) **Competitiveness** $\sum_{i \in A} x_i \not\geq c(A)$, i.e., no surplus is created. Because if any surplus is created then a competitor can deliver the message at a cheaper cost by reducing the surplus.

   The condition of *budget balance* consists of satisfying both, cost recovery and competitiveness, i.e., $\sum_{i \in A} x_i = c(A)$ (the set of bidders receiving the service pay exactly the total cost of T).
   Approximate Budget Balance: We relax the competitiveness condition to $\sum_{i \in A} x_i \leq \alpha c(A)$.

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

4. **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.

5. **Consumer Sovereignty(CS):** Every bidder is guaranteed to receive the service if she reports a high enough utility value. This forbids the trivial solution of not covering anyone. It also prevents the mechanism from being biased against any particular bidder.

6. **Polynomial Time(PT):** We also require that the mechanism run in time that is polynomial in the input size. This imposes a restriction that we can not solve an optimization problem when it is **NP**-Hard, unless **P=NP**. So we relax the notion and ask for an approximation, whenever applicable.

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 sub collection 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 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.

```
Input: An instance of the set cover problem, N, T, c :
       T → Q+ and the bids b_i, ∀i ∈ N
Output: The set of bidders to be served, A, and the
        prices charged to them, x_i ∀i ∈ A
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 j, C(S_j) = Σ_{i∈(N∩S_j)\A} x_i then
        A ← A ∪ (N ∩ S_j);
    end
end
```

Algorithm 1: Mechanism for set cover

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 [17]. 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 function to be minimized is

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

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

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

The dual of this LP is

$$\begin{array}{lll} \text{maximize} & \sum_{i \in N} x_i & \\ \text{subject to} & \sum_{i: i \in S_j} x_i \leq c(S_j), & \forall\ j = 1, 2, \ldots, k. \\ & x_i \geq 0, & \forall\ i \in N. \end{array}$$

The greedy set cover algorithm: 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 x_i's). Note, however, that $\{x_i\}_{i=1}^n$ is not feasible in the dual. The approximation guarantee follows from showing that $x'_i = x_i/H_n$, $(H_n = 1+1/2+1/3+\cdots+1/n)$ is feasible.

$$\begin{aligned} \sum_{j=1}^{k} c(S_j) y_j &= \sum_{i \in N} x_i \\ &= H_n . \sum_{i \in N} x'_i \\ &\leq H_n .(\text{Optimal solution to the LP}) \\ &\leq H_n .(\text{Optimal solution to the integer program}) \end{aligned}$$

The main observation here is that the dual variables 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. 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.

THEOREM 1. *The above mechanism returns a solution that is at most $O(\log n)$ times the optimum set cover of A.*

PROOF. We know that the greedy algorithm returns a solution within at most an $O(\log n)$ factor of the optimum (by the above dual fitting-based analysis). The result follows from the observation 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. □

Clearly, the mechanism is **BB**, and 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.

THEOREM 2. *The above mechanism is* **strategyproof**.

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

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

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 3 bidders , $\{1, 2, 3\}$ with $T = \{\{1, 2\}, \{2, 3\}, \{3, 1\}\}$ with the costs 2, 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 1 and 2 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 2 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 2 gets a higher profit, while bidder 3's profit does not go down. However, note that if we require that each bidder in the coalition should get a higher profit (*weakly group strategyproof*), then the above example does not work. In fact, the mechanism is weakly group strategyproof.

Moulin and Shenker [14] 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, $\sum_{i \in S} \xi(i, S) = C(S)$. 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, [14] give 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. Consider the previous example of 3 bidders. $C(\{1,2\}) + C(\{1,3\}) + C(\{3,2\}) < 2C(\{1,2,3\})$. So for some $i, \xi(i, \{i,j\}) < \xi(i, \{1,2,3\})$, which implies that ξ cannot be cross-monotonic.

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 j^{th} 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.) Our mechanism extends naturally to this game. Unfortunately it is *not* strategyproof. But, it has the property that underbidding is dominated, i.e., a bidder has no incentive to underbid (under all circumstances). The problem is that he may sometimes increase his profit by overbidding.

The mechanism is as before, except that it continues to offer to cover the bidders who have already been covered.

```
Input: An instance of the set cover problem, N, T, c :
       T → Q⁺ and the bids b_i, ∀i ∈ 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 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
```

Algorithm 2: Mechanism for set multi cover

As before, this mechanism can be seen as a modification of the greedy algorithm for the set multi cover problem (see [17]) where in addition to the usual set cover instance, each element is required to be covered a given number of times. The following is analogous to Theorem 1:

Theorem 3. *The above mechanism returns a solution that is at most $O(\log n)$ times the optimum set multi cover of A, where the multiplicity of the bidders refers to the coverage requirements.*

Why is this mechanism not strategyproof? Suppose that a bidder has utility 3 for getting covered each of the first two times. The smallest cost that he can be covered with for the first time is 4, and the cost for covering him the second time is 1. Then, by overbidding for the first, he can get covered twice at a cost of 5, and obtain a benefit of 1. If he had bid his true utility values, he would have not got covered at all, giving him zero benefit.

A strategy is dominant if it maximizes the profit in all the possible cases. In this mechanism, there is no dominant strategy. But underbidding is never the profit maximizing strategy. (If a user is getting covered for a value less than his utility, then it is always better to accept than reject.) We thus say, *underbidding is dominated*. Practically, this is a reasonable mechanism. Although overbidding may maximize the profit at times, the bidder does not know when, which means that if a bidder overbids, then there is a risk of him getting a negative benefit. So if a bidder is risk-averse, then his best strategy is to report the true utility.

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, $\phi : C \to I$ 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 [7, 13]. 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 x_S be an indicator variable denoting whether star S is picked and c_S denote the cost of star S.

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

The LP-relaxation of this program is:

$$\begin{array}{ll} \text{minimize} & \sum_{S\in\mathcal{S}} c_S x_S \\ \text{subject to} & \forall j \in C: \ \sum_{S: j\in S} x_S \geq 1 \\ & \forall S \in \mathcal{S}: \ x_S \geq 0 \end{array}$$

The dual program is:

$$\begin{array}{lll} \text{maximize} & & \sum_{j \in C} \alpha_j \\ \text{subject to} & \forall S \in \mathcal{S}: & \sum_{j \in S \cap C} \alpha_j \leq c_S \\ & \forall j \in C: & \alpha_j \geq 0 \end{array}$$

The main technique is to interpret the dual variables as cost shares of the cities. The mechanism uses the underlying greedy algorithm [7, 13], 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.

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 α_j, 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's cost share 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 [7, 13] with A as the set of cities instead of C. [7, 13] show that dividing the cost shares by 1.861 gives a dual feasible solution, and hence our mechanism gives a 1.861 factor approximation. This proves the theorem analogous to Theorem 1 Also the proof of strategyproofness is analogous to that Theorem 2.

We can use the improved algorithm of [8, 7] to get a better approximation ratio, but then the mechanism would not be budget balanced, 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 mechanism. Consider the example from [9]: a cycle on 6 vertices with 3 facilities and 3 cities alternating. The cost of each edge is 1, and the cost of each facility is 2.

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.

$A \leftarrow \emptyset$(connected cities) ;
$\forall\ j,\ x_j = 0$;
while $C \neq A$ **do**
    Raise all x_j's in $C \setminus A$ continuously at the same rate until one of the three events happens:
    if $x_j > b_j$ **then**
        $C \leftarrow C \setminus j$;
    end
    if *for some closed facility i, sum of offers to $i = f_i$* **then**
        Open facility i;
        Connect to i each city with no zero offer to i;
        $A \leftarrow A \cup \{$ cities connected to $i\}$;
    end
    if *For $j \in C \setminus A, i$ an open facility, $x_j = c_{ij}$* **then**
        $A \leftarrow A \cup j$;
        connect city j to facility i;
    end
end

Algorithm 3: Mechanism for Facility Location

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. $C(\emptyset) = 0$,
2. for any $S, T \subseteq N$, $C(S) + C(T) \geq C(S \cup T) + C(S \cap T)$.

The constraint 2 can also be replaced by

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

Jain and Vazirani ([10]) 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 ([14]) 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 $\sum_{i \in S} \xi(i, S) = C(S)$. Start with $x_i = 0$ for all $i \in S$. Say a set $S \subseteq N$ is tight if $C(S) = \sum_{i \in S} x_i$. 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. $\xi(i, S) := x_i$. [10] prove the following:

THEOREM 4. *At any time, there is a unique maximal tight set, and it can be found in polynomial time.*

THEOREM 5. *The cost sharing method ξ is cross-monotonic.*

4.2 MS Mechanism

Given an cross monotonic cost sharing method ξ, [14] give 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 continues the same way until everyone left can afford their cost shares.

```
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);
```

Algorithm 4: Mechanism $M(\xi)$

THEOREM 6. *For any cross monotonic cost sharing method* ξ *the mechanism* $M(\xi)$ *is* **BB**, *satisfies* **VP, NP, CS** *and is group strategyproof.*

4.3 Our mechanism

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.

```
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
```

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:

THEOREM 7. *The mechanism* M' *and* $M(\xi)$ *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. □

Note that our algorithm cannot be extended to the Steiner Tree game (considered in [9]) 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.

5. ACKNOWLEDGEMENTS

We would like to thank Aranyak Mehta and Amin Saberi for useful discussions.

6. REFERENCES

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

[2] A Fiat, A Goldberg, J Hartline, and A Karlin. Competitive Generalized Auctions. In *Proc. 34th Ann. ACM Symp. on Theory of Comp.*, 2002.

[3] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmission. In *Proc. 32nd Ann. ACM Symp. on Theory of Comp.*, 2000.

[4] M. Goemans and M. Sketella. Cooperative facility location games. In *Proc. 11th annual Symposium on Discrete Algorithm*, pages 76–85, 2000.

[5] D. Granot and G. Huberman. On minimum cost spanning tree games. *Mathematical Programming*, 21:1–18, 1981.

[6] D. Granot and G. Huberman. On the core and nucleolus of minimum cost spanning tree games. *Mathematical Programming*, 29:323–347, 1984.

[7] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. To appear in *Journal of ACM*.

[8] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problem. In *Proceedings of 34th ACM Symposium on Theory of Computing*, 2002.

[9] K. Jain and V. V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, 2001.

[10] K. Jain and V. V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, 2002.

[11] K. Kent and D. Skorin-Kapov. Population monotonic cost allocation on mst's. In *Operational Reerach Proceedings KOI*, pages 43–48, 1996.

[12] K. Kent and 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*, pages 463–468, 1997.

[13] M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. A Greedy Facility Location Algorithm Analyzed Using Dual Fitting. In *RANDOM-APPROX* 2001: 127-137

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

[15] M. Pal and E. Tardos. Strategy Proof Mechanisms via Primal-Dual Algorithms. *Manuscript*, 2002.

[16] T. Roughgarden and E. Tardos How Bad is Selfish Routing?. In *Proc. 41st IEEE Symp. on The Foundations of Comp. Sc.*, 2000.

[17] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2001.