Cost Sharing and Strategyproof Mechanisms for Set Cover Games

Xiang-Yang Li[⋆1], Zheng Sun[⋆⋆2], and WeiZhao Wang[1]

[1] Illinois Institute of Technology, Chicago, IL, USA, {lixian,wangwei4}@iit.edu
[2] Hong Kong Baptist University, Hong Kong, China, sunz@comp.hkbu.edu.hk

Abstract. We develop for set cover games several general cost-sharing methods that are approximately budget-balanced, core, and/or group-strategyproof. We first study the cost sharing for a single set cover game, which does not have a budget-balanced core. We show that there is no cost allocation method that can always recover more than $\frac{1}{\ln n}$ of the total cost if we require the cost sharing being a core. Here n is the number of all players to be served. We give an efficient cost allocation method that always recovers $\frac{1}{\ln d_{max}}$ of the total cost, where d_{max} is the maximum size of all sets. We then study the cost allocation scheme for all induced subgames. It is known that no cost sharing scheme can always recover more than $\frac{1}{n}$ of the total cost for every subset of players. We give an efficient cost sharing scheme that always recovers at least $\frac{1}{2n}$ of the total cost for every subset of players and furthermore, our scheme is cross-monotone. When the elements to be covered are selfish agents with privately known valuations, we present a strategyproof charging mechanism, under the assumption that all sets are simple sets, such that each element maximizes its profit when it reports its valuation truthfully; further, the total cost of the set cover is no more than $\ln d_{max}$ times that of an optimal solution. When the sets are selfish agents with privately known costs, we present a strategyproof payment mechanism in which each set maximizes its profit when it reports its cost truthfully. We also show how to *fairly* share the payments to all sets among the elements.

1 Introduction

Generalized Set Cover Problem Let $U = \{e_1, e_2, \cdots, e_n\}$ be a finite set, and let $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ be a collection of multisets (or *sets* for short) of U. For each $e_i \in U$ and each $S_j \in \mathcal{S}$, we denote the multiplicity of e_i in S_j by $k_{j,i}$. Each S_j is associated with a cost c_j. For any $\mathcal{X} \subseteq \mathcal{S}$, let $C(\mathcal{X})$ denote the total costs of the sets in $\mathcal{X}$, *i.e.*, $C(\mathcal{X}) = \sum_{S_j \in \mathcal{X}} c_j$. For a given $k > 0$ and a set of *element coverage requirements* $\{r_1, r_2, \cdots, r_n\}$, a k*-partial-cover* $\mathcal{C}$ is defined to be a subset $\{S_{j_1}, S_{j_2}, \cdots, S_{j_l}\}$ of $\mathcal{S}$ such that $\sum_{i=1}^{n} \min\{r_i, \sum_{t=1}^{l} k_{j_t,i}\} \geq k$. The *generalized set cover problem* is to compute an optimum k-partial-cover $\mathcal{C}_{opt}$ with the minimum cost $C(\mathcal{C}_{opt})$.

⋆ The research of the author was supported in part by NSF under Grant CCR-0311174. The research was conducted when the author was visiting Hong Kong Baptist University.

⋆⋆ The research of the author was supported in part by Grant FRG/03-04/II-21 and Grant RGC HKBU2107/04E.

This problem becomes the traditional multicover problem [1, 2] when we set $k = \sum_{i=1}^{n} r_i$ and $k_{j,i} = 1$ for all S_j and e_i, as each element e_i should be *fully* covered and each set S_j is a simple set. When we set $r_i = 1$, it becomes the traditional partial cover problem [3]. This problem is therefore a natural extension of the classic set cover problem by allowing partial cover, multiset, and element coverage requirement greater than 1. Accordingly, the greedy algorithm for this problem is a combination of the algorithms designed for partial cover and multicover [1–3].

Set Cover Game Consider the following scenario: a company can choose from a set of service providers $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ to provide services to a set of service receivers $U = \{e_1, e_2, \cdots, e_n\}$.

- With a fixed cost c_j, each service provider S_j can provide services to a fixed subset of service receivers.
- There may be a limit $k_{j,i}$ on the number of units of service that a service provider S_j can provide to a service receiver e_i. For example, each service provider may be a cargo company that is transporting goods to various cities (the service receivers), and the amount of goods that can be transported to a particular city daily is limited by the number of trains/trucks that are going to that city everyday.
- Each service receiver e_i may have a limit r_i on the number of units of service that it desires to receive (and is willing to pay for).
- There may be a limit k on the total number of units of service that the service providers shall provide to the service receivers.

The problem can be modeled by the generalized set cover problem defined previously. There may be different types of games according to various conditions:

1. Each service receiver e_i has to receive at least r_i units of service, and the costs incurred by the service providers will be shared by the service receivers.
2. Each service receiver e_i declares a bid $b_{i,r}$ for the r-th unit of service it shall receive, and is willing to pay for it only if the assigned cost is at most $b_{i,r}$.
3. Each service provider S_j declares a cost c_j, and is willing to provide the service only if the payment received is at least c_j.

There are different algorithmic issues for these games. For example, for Game 1, we shall define a cost allocation method so that every subset of service receivers feel that the total cost they need to pay is "fair" according to certain criteria. For Games 1 and 2, the cost allocation method, by charging service receivers, needs to recover (either entirely or a constant fraction of) the total cost of the chosen service providers. For Games 2 and 3, we need a mechanism (for determining costs charged to service receivers and payments paid to service providers) that can guarantee that the players are truthful with their declaration of bids/costs.

Our Results We first study how we share the cost of the selected service providers among the service receivers such that some fairness criteria are met. We present a cost sharing method that is $\frac{1}{\ln d_{max}}$-budget-balanced and core, where d_{max} is the maximum set size. The bound $\frac{1}{\ln d_{max}}$ is tight. We also present a cost sharing method that is $\frac{1}{2n}$-budget-balanced core and cross-monotone, which is almost the optimum [4].

We then design greedy set cover methods that are cognizant of the fact that the service providers or the service receivers are selfish and rational. By "selfish," we mean that they only care about their own benefits without consideration for the global performances or fairness issues. By "rational," we mean that when the methods of computing the output for the set cover game are instituted, they will always choose their actions to maximize their benefits. When the elements to be covered are selfish agents with privately known valuations, we present a strategyproof charging mechanism, under the assumption that all sets are simple sets, such that each element maximizes its profit when it reports its valuation truthfully; further, the total cost of the set cover is no more than $\ln d_{max}$ times that of an optimal solution for these selected service receivers and their coverage requirements. When the sets are selfish agents with privately known costs, we present a strategyproof payment mechanism in which each set maximizes its profit when it reports its cost truthfully. We also show how to *fairly* share the payments to all sets among the elements.

Paper Organization In Section 2, we give the exact definitions of fair cost sharing and mechanism design. In Section 3, we study how to fairly share the cost of the service providers among the covered service receivers when the receivers must receive the service. We show in Section 4 how to charge the cost of service providers to the selfish service receivers when each receiver has a valuation on the r-th cover received. We then show in Section 5 a strategyproof payment scheme to the selfish service providers when each has a privately known cost. We conclude our paper in Section 6.

2 Preliminaries and Prior Art

2.1 Preliminaries

Algorithm Mechanism Design (AMD) Assume that there are n agents. Each agent i, for $i \in \{1, \cdots, n\}$, has some *private* information t_i, called its *type*. All agents' types define a type vector $t = (t_1, t_2, \cdots, t_n)$. A mechanism defines, for each agent i, a set of strategies A_i. For each strategy vector $a = (a_1, \cdots, a_n)$, *i.e.*, agent i plays a strategy $a_i \in A_i$, the mechanism computes an *output* $o = \mathcal{O}(a)$ and a *payment* vector $\mathcal{P}(a) = (p_1, \cdots, p_n)$, where $p_i = \mathcal{P}_i(a)$ is the amount of money given to the participating agent i. Let $v_i(t_i, o)$ denote agent i's preferences to an output o and $u_i(t_i, o(a), p_i(a))$ denote its *utility* at the outcome (o, p) of the game. We assume that agents are *rational* and have quasi-linear utility functions. The utility function is *quasi-linear* if $u_i(t_i, o) = v_i(t_i, o) + p_i$. An agent is called *rational* if it always adopts its best strategy (called *dominant strategy*) that maximizes its utility regardless of what other agents do. A direct-revelation mechanism is *incentive compatible* (IC) if reporting valuation truthfully is a dominant strategy. Another common requirement in the literature for mechanism design is the so called *individual rationality* (IR): the agent's utility of participating in the output of the mechanism is not less than the utility of the agent if it did not participate at all. A mechanism is called *truthful* or *strategyproof* if it satisfies both IC and IR properties. To make the mechanism tractable, both methods $\mathcal{O}()$ and $\mathcal{P}()$ should be computable in polynomial time. A mechanism $M = (\mathcal{O}, \mathcal{P})$ is β-efficient if $\forall t$, $\sum_{i=1}^{n} v_i(t_i, \mathcal{O}(t)) \geq \beta \cdot \max_o \sum_{i=1}^{n} v_i(t_i, o)$. Obviously for the set cover game, we cannot design an $o(\ln n)$-efficient polynomial-time computable strategyproof mechanism unless $NP \subset DTIME(n^{\log \log n})$ [2].

Cost Sharing Consider a set U of n players. For a subset $S \subseteq U$ of players, let $C(S)$ be the cost of providing service to S. Here $C(S)$ could be the minimum cost, denoted by $\mathrm{OPT}(S)$, or the cost computed by some algorithm $\mathcal{A}$, denoted by $\mathcal{A}(S)$. We always assume that the cost function $C(S)$ is *cohesive*, *i.e.*, for any two disjoint subsets S_1 and S_2, $C(S_1 \cup S_2) \leq C(S_1) + C(S_2)$. A cost sharing scheme is simply a function $\xi(i, S)$ with $\xi(i, S) = 0$ for $i \notin S$, for every set $S \subseteq U$ of players. An obvious criterion is that the sharing method should be *fair*. While the definition of budget-balance is straightforward, defining fairness is more subtle: many fairness concepts were proposed in the literature, such as *max-min* [5], *min-max* [6], *core* and *bargaining set* [7]. Typically, the following three properties are required by a cost sharing scheme.

1. (**α-budget-balance**) For some given parameter $\alpha \leq 1$, $\alpha \cdot C(U) \leq \sum_{i \in U} \xi(i, U) \leq C(U)$. If $\alpha = 1$, we call the cost sharing scheme *budget-balanced*.
2. (**fairness under core**) For any subset $S \subseteq U$, $\sum_{i \in S} \xi(i, U) \leq \mathrm{OPT}(S)$.
3. (**Cross-monotonicity**) For any two subsets $S \subseteq T$ and $i \in S$, $\xi(i, S) \geq \xi(i, T)$.

When only the first two conditions are satisfied, we call the cost sharing scheme to be in the α-*core*. When all three conditions are met, we call the cost sharing scheme to be cross-monotone α-*core*. When each player i has a valuation v_i on getting the service, a mechanism should first decide the output of the game (who will get the service), and then decide what is the share of each selected player (what is the payment method). It is well-known that a cross-monotone cost sharing scheme implies a *group-strategyproof* mechanism [8]. Notice that the cross-monotone property is not the necessary condition for group-strategyproof. Naturally, several additional properties are required for a cost sharing scheme when every player has a valuation on getting the service.

1. (**Incentive Compatibility**) Assume that the valuation by player i on getting the service is v_i. Let $b = (b_1, b_2, \cdots, b_n)$ be the bidding vector of n players. Let $\mathcal{O}(b) = (o_1, o_2, \cdots, o_n)$ denote whether a player is selected to get the service or not and $\mathcal{P}(b)$ be the charge to player i, *i.e.*, the mechanism is $M = (\mathcal{O}(b), \mathcal{P}(b))$. It satisfies IC if every player maximizes its profit $v_i \cdot o_i - p_i$ when $b_i = v_i$.
2. (**No Positive Transfer**) For every player i, $p_i \geq 0$.
3. (**Individual Rationality**) For every player i, $v_i \cdot o_i - p_i \geq 0$.
4. (**Consumer Sovereignty**) Fix the bids of all other players, there exists a value τ_i such that player i is guaranteed to get the service when its bid is larger than τ_i.

2.2 Prior Arts on Cost Sharing and Algorithm Mechanism Design

Although the traditional set cover problem (without multisets and partial-cover requirement) can be viewed as a special case of multicast, several results were proposed specifically for set cover in selfish environment. Devanur *et al.* [9] studied, for the set cover game and facility location game, how the cost of shared resource is to be distributed among its users in such a way that revealing the true valuation is a dominant strategy for each user. Their cost sharing method is not in the *core* of the game. One of the open questions left in [9] is to design a strategyproof cost sharing method for multicover game in which the bidders might want to get covered multiple times. For facility location game, Pál and Tardos [10] gave a cost sharing method that can recover $\frac{1}{3}$ of the total cost, and recently, Immorlica *et al.* [4] showed that this is the best achievable upper bound for any cross-monotonic cost sharing method. Sharing the *cost* of the multicast structure among receivers was studied in [8, 11–16] so some fairness is accomplished.

3 Cost Sharing Among Unselfish Service Receivers

In this section, we study how to share the cost of the service providers among a given set of service receivers. For this scenario, it is difficult to find realistic examples where a partial cover is desired. Therefore, in the remainder of this section, we only consider the multiset multicover problem, *i.e.*, $k = \sum_{i=1}^{n} r_i$. However, the results presented here can easily be generalized to the partial cover case, should such a scenario arise.

3.1 α-Core

Given a subset of elements X, let $\mathrm{OPT}(X)$ denote the cost of an optimum cover $\mathcal{C}_{opt}(X)$ of X. This cost function clearly is *cohesive*: for every partition T_1, T_2, $\cdots$, T_t of U, $\mathrm{OPT}(U) \le \sum_{i=1}^{t} \mathrm{OPT}(T_i)$. A *cost allocation* for U is a n-dimensional vector $x = (x_1, x_2, \cdots, x_n)$ that specifies for each element $e_i \in U$ the share $x_i \ge 0$ of the total cost of serving U that e_i shall pay. Ideally, when the set of elements to be covered is fixed to be U, we want the cost allocation x to be budget-balanced and fair, *i.e.*, being in core. However, a simple example in [17] shows that there is no budget-balanced core for the set-cover game. We then relax the notion of budget-balance to the notion of α-budget-balance for some $\alpha \le 1$. See [17] for the proof of the achievable α-core.

Theorem 1. *For the set cover game,* no *cost allocation method is α-core for $\alpha > \frac{1}{\ln n}$.*

We then give a cost allocation method that can recover $\frac{1}{\ln d_{max}}$ of the total cost $\mathrm{OPT}(U)$ for a multiset multicover game, where $d_{max} = \max_{1 \le j \le m} |S_j|$. Without loss of generality, we assume that $d_{max} \le \sum_{i=1}^{n} r_i$. The basic approach of our cost allocation method is as follows. We first run the greedy Algorithm 1 to find a set cover $\mathcal{C}_{grd}$ with an approximation ratio of $\ln d_{max}$. Starting with $\mathcal{C}_{grd} = \emptyset$, the greedy algorithm adds to $\mathcal{C}_{grd}$ a set $S_{j_{t'}}$ at each round t'. After the s-th round, we define the *remaining required coverage* r'_i of an element e_i to be $r_i - \sum_{t'=1}^{s} k_{j_{t'},i}$. For any $S_j \notin \mathcal{C}_{grd}$, the *effective coverage* $k'_{j,i}$ of e_i by S_j is defined to be $\min\{k_{j,i}, r'_i\}$, the *value* v_j of S_j is defined to be $\sum_{i=1}^{n} k'_{j,i}$, and the *effective average cost* of S_j is defined to be $\frac{c_j}{v_j}$.

Algorithm 1 Greedy algorithm for multiset multicover problem.

```
C_grd ← ∅; r'_i ← r_i for each e_i.
while U ≠ ∅ do
    pick the set S_t' in S \ C_grd with the minimum effective average cost.
    C_grd ← C_grd ∪ {S_t'}.
    for all e_i ∈ U do
        r'_i ← max{0, r'_i − k_t',i}.
        if r'_i = 0 then U ← U \ {e_i}.
```

The greedy algorithm will select a set S_j with the least effective average cost. For any e_i and r such that $r_i - r'_i + 1 \leq r \leq r_i - r'_i + k'_{j,i}$, we let $\text{price}(i, r) = \frac{c_j}{v_j}$. Let $x'_i = \sum_{r=1}^{r_i} \text{price}(i, r)$ and $x_i = \frac{x'_i}{\ln d_{max}}$. We claim the following theorem (see [17]):

Theorem 2. *The above-defined cost allocation x is a $\frac{1}{\ln d_{max}}$-core.*

Recall that the core we defined, given a set of players U, required that $\sum_{e_i \in T} \xi(i, U)$ is at most the *optimum* cost of providing service to elements in T. For a set cover game, clearly it is NP-hard to find the optimum cost of covering T. Naturally, one may relax the α-core as follows: a cost sharing method $\xi(i, \cdot)$ is called a *relaxed* α-core if (1) $\alpha \cdot \mathcal{C}_{grd}(U) \leq \sum_{i \in U} \xi(i, U) \leq \mathcal{C}_{grd}(U)$; and (2) $\sum_{i \in T} \xi(i, U) \leq \mathcal{C}_{grd}(T)$ for every subset $T \subseteq U$. Even we relax the definition of the core to this, we can still prove in [17] that with the cost function computed by the greedy algorithm, there is no cost sharing method that is a relaxed α-core for $\alpha = \Omega(\frac{1}{\ln n})$.

3.2 Cross-monotone α-Core

Clearly, if a cost sharing scheme is cross-monotone α-core then every cost allocation method $\xi(\cdot, S)$ induced on a subset S of players is always α-core, but the reverse is not true. From Theorem 1, clearly *no* cost sharing scheme for the set cover game is cross-monotone α-core for $\alpha = \frac{1}{\ln n}$. Recently, it was claimed in [4] that for set cover game, there is *no* cross-monotone α-core cost sharing scheme for $\alpha = \frac{1}{n} + \epsilon$.

For generalized set cover games, we will present a cross-monotone cost sharing scheme $\xi(i, S)$ (see Algorithm 2) that can recover $\frac{1}{2n}$ of the total cost. We show an example in [17] that the bound $\frac{1}{2n}$ is tight for Algorithm 2. Further, the bound is tight, for set cover games without multisets (but still allowing multicover requirements): our cross-monotone cost sharing scheme $\xi(i, S)$ can recover $\frac{1}{n}$ of the total cost.

Algorithm 2 Cost sharing for multiset multicover game with elements T.

Set $\mathcal{C}_{\mathcal{A}} \leftarrow \emptyset$, $Y(i, j) = 0$ and $\zeta(i, j) = 0$ for $1 \leq i \leq n$ and $1 \leq j \leq m$. Here $Y(i, j)$ denotes how many cover requirements of element e_i are provided by set S_j, and $\zeta(i, j)$ denotes the fraction cost of set S_j shared by the element e_i.
for all element $e_i \in T$ **do**
  Set $r'_i \leftarrow r_i$;
  while $r'_i > 0$ **do**
    Find the set S_t with the minimum ratio $\min_{S_j \in \mathcal{S} - \mathcal{C}_{\mathcal{A}}} \frac{c_j}{\min(k_{j,i}, r'_i)}$;
    $Y(i, t) \leftarrow \min(k_{j,i}, r'_i)$; $r'_i \leftarrow r'_i - Y(i, t)$; and $\mathcal{C}_{\mathcal{A}} \leftarrow \mathcal{C}_{\mathcal{A}} \cup \{S_t\}$.
for all set S_j **do**
  if $\sum_{1 \leq i \leq n} Y(i, j) > 0$ **then** $\rho_j \leftarrow \frac{c_j}{\sum_{1 \leq i \leq n} Y(i, j)}$;
  for all element $e_i \in T$ **do**
    Set $\zeta(i, j) = Y(i, j) \cdot \rho_j$.
for all element $e_i \in T$ **do**
  Set $\xi'(i, T) = \sum_{1 \leq j \leq m} \zeta(i, j)$ and $\xi(i, T) = \frac{\sum_{1 \leq j \leq m} \zeta(i, j)}{2|T|}$.

Theorem 3. *The cost sharing scheme $\xi(\cdot,\cdot)$ is a cross-monotone $\frac{1}{2n}$-core and is cross-monotone $\frac{1}{n}$-core for set cover game when every set S_j is a simple set.*

4 Cost Sharing Among Selfish Service Receivers

In Section 3 we assumed that all elements (service receivers) are unselfish and all their coverage requirements are to be satisfied. In this section, we consider the problem of selecting service providers under the constraint of a collection of bids $B = B_1 \cup B_2 \cup \cdots \cup B_n$. Each B_i contains a series of bids $b_{i,1}, b_{i,2}, \cdots, b_{i,r_i}$, where $b_{i,r}$ denotes the declared price that element e_i is willing to pay for the r-th coverage (*i.e.*, the valuation of the r-th coverage). In this scenario, we may also consider partial cover, as the total number of units of service available may be limited by a constant k.

We assume that $b_{i,1} \geq b_{i,2} \geq \cdots \geq b_{i,r_i}$. This is often true in realistic situations: the marginal valuations are usually decreasing. A bid $b_{i,r}$ will be served (and the subsequent bid $b_{i,r+1}$ will be considered) only if $b_{i,r} \geq \text{price}(i,r)$, where $\text{price}(i,r)$ is the cost to be paid by e_i for its r-th coverage. Further, to guarantee that the mechanism is both strategyproof and budget-balanced, we assume that each set is a simple set.

We use a greedy algorithm (see Algorithm 3) similar to the one for the traditional set cover game [9]. Informally speaking, we start with $y = 0$, where y is the cost to be shared by each bid served. We raise y until there exists a set S_j whose cost can be sufficiently covered by the element copies in S_j, if each element copy needs to pay y. To adapt to the multicover scenario, we make the following changes:

- ⋆ For any set $S_j \notin \mathcal{C}_{grd}$ and any e_i, we define the *collection of alive bids* $B_i^{(j)}$ of e_i with respect to S_j to be $\{b_{i,r_i-r'_i+1}\}$ if $k'_{j,i} > 0$ (*i.e.*, $k'_{j,i} = 1$ since S_j is a simple set) and $b_{i,r_i-r'_i+1} \geq y$, or $\emptyset$ if otherwise. That is, if y is the cost to be paid for each bid served, $B_i^{(j)}$ contains the bid of e_i covered by S_j that can afford the cost (if any).
- ⋆ Define the value v_j of S_j as $\sum_{i=1}^{n} |B_i^{(j)}|$, and its effective average cost as $\frac{c_j}{v_j}$.

Algorithm 3 Cost sharing for multicover game with selfish receivers.

$\mathcal{C}_{grd}(B) \leftarrow \emptyset$; $A \leftarrow \emptyset$; $y \leftarrow 0$; $k' \leftarrow k$; $B' = \emptyset$;
while $A \neq U$ and $k' > 0$ **do**
  Raise y until one of the two events happens:
  if $B_i^{(j)} = \emptyset$ for all S_j **then** $U \leftarrow U \setminus \{e_i\}$;
  if $c_j \leq v_j \cdot y$ for some set S_j **then**
    $\mathcal{C}_{grd}(B) \leftarrow \mathcal{C}_{grd}(B) \bigcup \{S_j\}$; $k' \leftarrow k' - v_j$;
    for all element e_i with $B_i^{(j)} \neq \emptyset$ **do**
      $\text{price}(i, r_i - r'_i + 1) \leftarrow \frac{c_j}{v_j}$; $B' \leftarrow B' \cup \{b_{i,r_i-r'_i+1}\}$;
      $r'_i \leftarrow r'_i - 1$;
      if $r'_i = 0$ **then** $A \leftarrow A \bigcup \{e_i\}$;
    update all $B_i^{(j')}$ for all $S_{j'} \notin \mathcal{C}_{grd}$ and $e_i \in S_j \bigcap S_{j'}$;

When the algorithm terminates, B' contains all bids (of all elements) that are served. We prove the following theorem about this mechanism (see [17] for proof):

Theorem 4. *Algorithm 3 defines a strategyproof mechanism. Further, the total cost of the sets selected is no more than* $\ln d_{max}$ *times that of an optimal solution.*

In [9] multicover game was also considered. However, the algorithm used is different from ours and also they did not assume that the bids by the same element are non-increasing, and their mechanism is not strategyproof.

5 Selfish Service Providers

The underline assumption made so far in previous sections is that the service providers are truthful in revealing their costs of providing the service. In this section, we will address the scenario when service providers are selfish in revealing their costs.

5.1 Strategyproof Mechanism

We want to find a subset of agents D such that $\bigcup_{j \in D} S_j$ has r_i copies of element e_i for every element $e_i \in U$. Let $c = (c_1, c_2, \cdots, c_m)$. The social efficiency of the output D is $-\sum_{j \in D} c_j$, which is the objective function to be maximized. Clearly a VCG mechanism [18–20] can be applied if we can find the subset of $\mathcal{S}$ that satisfies the multicover requirement of elements in U with the minimum cost. Unfortunately this is NP-hard. Let $\mathcal{C}_{grd}(\mathcal{S}, c, U, r)$ be the sets selected from $\mathcal{S}$ (with cost specified by a cost vector $c = (c_1, \cdots, c_m)$) by the greedy algorithm to cover elements in U with cover requirement specified by a vector $r = (r_1, \cdots, r_n)$ (see Algorithm 1). We assume that the type of an agent is (S_j, c_j), *i.e.*, every service provider j could lie not only about its cost c_j but also about the elements it could cover. This problem now looks very similar to the combinatorial auction with single minded bidder studied in [21]. We show in [17] that the mechanism $M = (\mathcal{C}_{grd}, \mathcal{P}^{VCG})$ is not truthful, *i.e.*, use Algorithm 1 to find a set cover, and apply VCG mechanism to compute the payment to the selected agents: the payment to an agent j is 0 if $S_j \notin \mathcal{C}_{grd}$; otherwise, the payment to a set $S_j \in \mathcal{C}_{grd}$ is $\mathcal{P}_j^{VCG} = C(\mathcal{C}_{grd}(\mathcal{S} \setminus \{S_j\}, c|^j\infty, U, r)) - C(\mathcal{C}_{grd}(\mathcal{S}, c, U, r)) + c_j$. Here $C(\mathcal{X})$ is the total cost of the sets in $\mathcal{X} \subseteq \mathcal{S}$.

For the moment, we assume that agent j won't be able to lie about its element S_j. We will drop this assumption later. Clearly, the greedy set cover method presented in Algorithm 1 satisfies a monotone property: if a set S_j is selected with a cost c_j, then it is still selected with a cost less than c_j. Monotonicity guarantees that there exists a strategyproof mechanism for generalized set cover games using Algorithm 1 to compute its output. We then show how to compute the payment to each service provider efficiently. We assume that for any set S_j, if we remove S_j from $\mathcal{S}$, $\mathcal{S}$ still satisfies the coverage requirements of all elements in U. Otherwise, we call the set cover problem to be *monopoly*: the set S_j can charge an arbitrarily large cost in the monopoly game. The following presents our payment scheme for multiset multicover set cover problem.

We show in [17] that the mechanism $M = (\mathcal{C}_{grd}, \mathcal{P}^{grd})$ is strategyproof (when the agent j does not lie about the set S_j of elements it can cover) and the payment $\mathcal{P}_j^{grd}$

Algorithm 4 Strategyproof payment $\mathcal{P}_j^{grd}$ to service provider $S_j \in \mathcal{C}_{grd}$.

$\mathcal{C}_{grd} \leftarrow \emptyset$ and $s \leftarrow 1$;
$k' \leftarrow k$, $r'_i = r_i$ for each e_i;
while $k' > 0$ **do**
  pick the set $S_t \neq S_j$ in $\mathcal{S} \setminus \mathcal{C}_{grd}$ with the minimum effective average cost;
  Let v_t and v_j be the values of the sets S_t and S_j at this moment;
  $\kappa(j,s) \leftarrow \frac{v_j}{v_t} c_t$ and $s \leftarrow s+1$; $\mathcal{C}_{grd} \leftarrow \mathcal{C}_{grd} \cup \{S_t\}$; $k' \leftarrow k' - v_t$;
  for each e_i, $r'_i \leftarrow \max\{0, r'_i - k_{t,i}\}$;
$\mathcal{P}_j^{grd} = \max_{t=1}^{s-1} \kappa(j,t)$ is the payment to selfish service provider S_j.

is the minimum to the selfish service provider j among any strategyproof mechanism using Algorithm 1 as its output. We now consider the scenario when agent j can also lie about S_j. Assume that agent j cannot lie upward[3], *i.e.*, it can only report a $S'_j \subseteq S_j$. We argue that agent j will not lie about its elements S_j. Notice that the value $\kappa(j,s)$ computed for the s-th round is $\kappa(j,s) = \frac{v_j}{v_t} c_t = \frac{\sum_{1 \le i \le n} \min(r'_i, k_{j,i})}{\sum_{1 \le i \le n} \min(r'_i, k_{t,i})} c_t$. Obviously v_j cannot increase when agent j reports any set $S'_j \subseteq S_j$. Thus, falsely reporting a smaller set S'_j will not improve the payment of agent j.

Theorem 5. *Algorithm 1 and 4 together define a* $\ln d_{max}$*-efficient strategyproof mechanism* $M = (\mathcal{C}_{grd}, \mathcal{P}^{grd})$ *for multiset multicover set cover game.*

5.2 Sharing the Payment Fairly

In the previous subsection, we only define what is the payment to a selfish service provider S_j. A remaining question is how the payment should be charged fairly (under some subtle definitions) to encourage cooperation among service receivers. One natural way of defining fair payment sharing is to extend the fair cost sharing method. Consider a strategyproof mechanism $M = (\mathcal{O}, \mathcal{P})$. Let $\mathcal{P}(T)$ be the total payment to the selfish service providers when T is the set of service receivers to be covered. A payment sharing scheme is simply a function $\pi(i,T)$ such that $\pi(i,T) = 0$ for any element $e_i \notin T$. A payment sharing scheme is called α-*budget-balanced* if $\alpha \cdot \mathcal{P}(T) \le \sum_{e_i \in T} \pi(i,T) \le \mathcal{P}(T)$. A payment sharing scheme is said to be a *core* if $\sum_{e_i \in S} \pi(i,T) \le \mathcal{P}(S)$ for any subset $S \subset T$. A payment sharing scheme is said to be an α-*core* if it is α-budget-balanced and it is a core. For payment method $\mathcal{P}^{grd}$, we prove in [17] that

Theorem 6. *There is no* α*-core payment sharing scheme for* $\mathcal{P}^{grd}$ *if* $\alpha > \frac{1}{\ln n}$.

It is easy to show that if we share the payment to a service provider equally among all service receivers covered by this set, the scheme is not in the core of the game. We leave it as an open problem whether we can design an α-core payment sharing scheme for the payment $\mathcal{P}^{grd}$ with $\alpha = O(\frac{1}{\ln n})$.

[3] This can be achieved by imposing a large enough penalty if an agent could not provide the claimed service when it is selected.

In the next, we study the cross-monotone payment sharing scheme. A payment sharing scheme is said to be *cross-monotone* if $\pi(i,T) \leq \pi(i,S)$ for any two subsets $S \subset T$ and $i \in S$. A payment sharing scheme is said to be a *cross-monotone α-core* if it is α-budget-balanced and cross-monotone, and it is a core. We propose the following conjecture.

Conjecture 1 *For the strategyproof mechanism $M = (\mathcal{C}_{grd}, \mathcal{P}^{grd})$ of a set cover game, there is no payment sharing scheme $\pi(\cdot,\cdot)$ that is cross-monotone α-core for $\alpha = \frac{1}{n} + \epsilon$.*

In the remaining of this section we will present a cross-monotone budget-balanced payment sharing scheme for a strategyproof payment scheme of the set cover game. Our payment sharing scheme is coupled with the following *least cost set* mechanism $M = (\mathcal{C}_{lcs}, \mathcal{P}^{lcs})$. It uses the output called *least cost set* $\mathcal{C}_{lcs}$ (described in Algorithm 5): for each service receiver e_i, we find the service provider S_j with the least cost efficiency $\frac{c_j}{\min(r_i, k_{j,i})}$ to cover the element e_i. New cost efficient sets are found till the cover requirement of e_i is satisfied. The payment (described in Algorithm 6) to a set S_j is defined as $\mathcal{P}_j^{lcs} = \max_{e_i \in U} p_j^i$, where p_j^i is the largest cost that S_j can declare while S_j is still selected to cover e_i. If the set S_j is not selected to cover e_i, then $p_j^i = 0$.

Algorithm 5 Least cost set greedy for multiset multicover game.

Let $\mathcal{C}_{lcs} \leftarrow \emptyset$.
for all element $e_i \in T$ **do**
  Let $r'_i \leftarrow r_i$;
  while $r'_i > 0$ **do**
    Find the set S_t with the minimum ratio $\min_{S_j \in \mathcal{S} - \mathcal{C}_{lcs}} \frac{c_j}{\min(k_{j,i}, r'_i)}$;
    $r'_i \leftarrow r'_i - \min(k_{j,i}, r'_i)$; $\mathcal{C}_{lcs} \leftarrow \mathcal{C}_{lcs} \cup \{S_t\}$.

Algorithm 6 Compute the payment $\mathcal{P}_j^{lcs}$ to a set S_j in $\mathcal{C}_{lcs}$.

Let $\mathcal{C}_{lcs} \leftarrow \emptyset$, $p_j^i = 0$ for $1 \leq i \leq n$ and $s = 1$;
for all element $e_i \in T$ **do**
  Let $r'_i \leftarrow r_i$;
  while $r'_i > 0$ **do**
    Find the set $S_t \neq S_j$ with the minimum ratio $\min_{S_x \in \mathcal{S} - \mathcal{C}_{lcs} - \{S_j\}} \frac{c_x}{\min(k_{x,i}, r'_i)}$;
    $\kappa(j,i,s) = \frac{\min(k_{j,i}, r'_i)}{\min(k_{t,i}, r'_i)} c_t$; $r'_i \leftarrow r'_i - \min(k_{j,i}, r'_i)$; $\mathcal{C}_{lcs} \leftarrow \mathcal{C}_{lcs} \cup \{S_t\}$ and $s \leftarrow s + 1$;
  $p_j^i \leftarrow \max_{1 \leq x < s} \kappa(j,i,s)$;
$\mathcal{P}_j^{lcs} \leftarrow \max_{1 \leq i \leq n} p_j^i$;

Theorem 7. *The mechanism $M = (\mathcal{C}_{lcs}, \mathcal{P}^{lcs})$ is $\frac{1}{2n}$-efficient and strategyproof.*

We then study how we charge the service receivers so that a budget-balance is achieved and the charging scheme also is fair under some concepts. Notice that, given a subset of elements T, we can view the total payments $\mathcal{P}(T)$ to all service providers covering T as a "cost" to T. The payment computed by mechanism $M = (\mathcal{C}_{lcs}, \mathcal{P}^{lcs})$ clearly is cohesive. Then naturally, we could use the cost-sharing schemes studied before to share this special cost among elements. However, it is easy to show by example that the previous cost-sharing schemes (studied in Section 3) are not in the core and also not cross-monotone.

Roughly speaking, our payment sharing scheme works as follows. Notice that a final payment to a set S_j is the maximum of payments p_j^i by all elements. Since different elements may have different value of payment to set S_j, the final payment $\mathcal{P}_j^{lcs}$ should be shared *proportionally* to their values, not *equally* among them as cost-sharing.

Algorithm 7 Sharing MV cost $\mathcal{P}$ among receivers.

1: Initialize $\xi(i, U) = 0$ and $\zeta_j(i, U) = 0$. Here $\zeta_j(i, U)$ denotes the payment to set S_j shared by the element e_i when the set of elements is U.
2: **for all** $S_j \in \mathcal{S}$ **do**
3: For all elements e_i, we compute the payment p_j^i. Sort the payments p_j^i, $1 \le i \le n$, in an increasing order. Assume that $p_j^{\sigma(1)}$, $p_j^{\sigma(2)}$, $\cdots$, $p_j^{\sigma(n-1)}$, $p_j^{\sigma(n)}$ are the sorted list of payments in an incremental order.
4: For elements $e_{\sigma(1)}$, $\cdots$, $e_{\sigma(n)}$, let $\zeta_j(\sigma(i), U) \leftarrow \sum_{t=1}^{i} \frac{p_j^{\sigma(t)} - p_j^{\sigma(t-1)}}{n-t+1}$. Here we assume that $p_j^{\sigma(0)} = 0$. Update the payment sharing as follows: $\xi(i, U) = \xi(i, U) + \zeta_j(i, U)$ for each $e_i \in U$.
5: $\xi(i, U)$ is the final payment sharing of service receiver e_i.

Our payment sharing method described in Algorithm 7 applies to a more general cost function. A cost function $\mathcal{P}$ is said to be *maximum-view cost* (MV cost) if it is defined as $\mathcal{P}_j = \max_{e_i \in U} p_j^i$ where p_j^i is the *view* of the cost of set S_j by element e_i. Obviously, the traditional cost c is a MV cost function by setting $p_j^i = c_j$ for each element e_i. The payment function $\mathcal{P}^{lcs}$ is also a MV cost function.

A service receiver is called *free-rider* in a payment sharing scheme if its shared total payment is no more than $\frac{1}{n}$ of its total payment it has to pay if it acts alone. Notice that, when a service receiver acts alone, the same mechanism is applied to compute the payment to the service providers.

Theorem 8. *The payment sharing scheme described in Algorithm 7 is budget-balanced, cross-monotone, in the core and does not permit free-rider.*

6 Conclusion

We studied cost sharing and strategyproof mechanisms for various set cover games. We gave an efficient cost allocation method that always recovers $\frac{1}{\ln d_{max}}$ of the total cost, where d_{max} is the maximum size of all sets. We further gave an efficient cost sharing scheme that is $\frac{1}{2n}$-budget-balanced, core and cross-monotone. When the elements to be covered are selfish agents with privately known valuations, we presented a strategyproof charging mechanism. When the sets are selfish agents with privately known costs, we presented two strategyproof payment mechanisms in which each set maximizes its profit when it reports its cost truthfully. We also showed how to *fairly* share the payments to all sets among the elements.

There are a number of open questions left for future research. Are the bounds on the α-budget-balanced cost sharing schemes tight, although we proved that they are asymptotically tight? Consider the strategyproof mechanism $M = (\mathcal{C}_{grd}, \mathcal{P}^{grd})$. Is there a payment sharing method that is $\frac{1}{\ln n}$-core? Is there a payment sharing method that is cross-monotone $\frac{1}{n}$-core? Is this $\frac{1}{n}$ a tight lower bound?

References

1. Chvátal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operation Research **4** (1979) 233–235
2. Feige, U.: A threshold of for approximating set cover. Journal of the ACM **45** (1998) 634–652
3. Slavík, P.: Improved performance of the greedy algorithm for partial cover. Information Processing Letters **64** (1997) 251–254
4. Immorlica, N., Mahdian, M., Mirrokni, V.S.: Limitations of cross-monotonic cost-sharing schemes. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) To appear.
5. Marbach, P.: Priority service and max-min fairness. IEEE/ACM Transactions on Networking **11** (2003) 733–746
6. Radunović, B., Boudec, J.Y.L.: A unified framework for max-min and min-max fairness with applications. In: Proceedings of 40th Annual Allerton Conference on Communication, Control, and Computing. (2002)
7. Osborne, M.J., Rubinstein, A.: A course in game theory. The MIT Press (1994)
8. Moulin, H., Shenker, S.: Strategyproof sharing of submodular costs:budget balance versus efficiency. Economic Theory **18** (2001) 511–533
9. Devanur, N., Mihail, M., Vazirani, V.: Strategyproof cost-sharing mechanisms for set cover and facility location games. In: Proceedings of the 4th ACM Conference on Electronic Commerce. (2003) 108–114
10. Pál, M., Tardos, E.: Group strategyproof mechanisms via primal-dual algorithms. In: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science. (2003) 584–593
11. Feigenbaum, J., Papadimitriou, C.H., Shenker, S.: Sharing the cost of multicast transmissions. Journal of Computer and System Sciences **63** (2001) 21–41
12. Libman, L., Orda, A.: Atomic resource sharing in noncooperative networks. Telecommunication Systems **17** (2001) 385–409
13. Shenker, S., Clark, D., Estrin, D., Herzog, S.: Pricing in computer networks: Reshaping the research agenda. ACM Computer Communication Review **26** (1996) 19–43
14. Feigenbaum, J., Krishnamurthy, A., Sami, R., Shenker, S.: Hardness results for multicast cost sharing. Theoretical Computer Science **304** (2003) 215–236
15. Archer, A., Feigenbaum, J., Krishnamurthy, A., Sami, R., Shenker, S.: Approximation and collusion in multicast cost sharing. Games and Economic Behavior **47** (2004) 36–71
16. Herzog, S., Shenker, S., Estrin, D.: Sharing the cost of multicast trees: An axiomatic analysis. IEEE/ACM Transactions on Networking **5** (1997) 847–860
17. Li, X.Y., Sun, Z., Wang, W.: Cost sharing and strategyproof mechanisms for set cover games (2004) at http://www.cs.iit.edu/∼xli/publications-select.htm.
18. Vickrey, W.: Counterspeculation, auctions and competitive sealed tenders. Journal of Finance **16** (1961) 8–37
19. Clarke, E.H.: Multipart pricing of public goods. Public Choice **11** (1971) 17–33
20. Groves, T.: Incentives in teams. Econometrica **41** (1973) 617–631
21. Lehmann, D.J., Oćallaghan, L.I., Shoham, Y.: Truth revelation in approximately efficient combinatorial auctions. Journal of the ACM **49** (2002) 577–602