Mechanism Design For Set Cover Games When Elements Are Agents

Zheng Sun*[1], Xiang-Yang Li**[2], WeiZhao Wang[2], and Xiaowen Chu***[1]

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

Abstract. In this paper we study the set cover games when the elements are selfish agents. In this case, each element has a privately known valuation of receiving the service from the sets, *i.e.*, being covered by some set. Each set is assumed to have a fixed cost. We develop several approximately efficient strategyproof mechanisms, each of which decides, after soliciting the declared bids by all elements, which elements will be covered, which sets will provide the coverage to these selected elements, and how much each element will be charged. For single-cover set cover games, we present a mechanism that is at least $\frac{1}{d_{\max}}$-efficient, *i.e.*, the total valuation of all selected elements is at least $\frac{1}{d_{\max}}$ fraction of the total valuation produced by any mechanism. Here $d_{\max}$ is the maximum size of the sets. For multi-cover set cover games, we present a budget-balanced strategyproof mechanism that is $\frac{1}{d_{\max} H_{d_{\max}}}$-efficient under reasonable assumptions. Here H_n is the harmonic function. For set cover games when both sets and elements are selfish agents, we show that a cross-monotonic *payment*-sharing scheme does not necessarily induce a strategyproof mechanism. This is a sharp contrast to the well-known fact that a cross-monotonic *cost*-sharing scheme always induces a strategyproof mechanism.

1 Introduction

In the past, an indispensable and implicit assumption on algorithm design for interconnected computers has been that all participating computers (called *agents*) are cooperative; they will behave exactly as instructed. This assumption is being shattered by the emergence of the Internet, as it provides a platform for distributed computing with agents belonging to independent and self-interested organizations, who may diverge from the prescribed algorithm to maximize their own benefits. This gives rise to a new challenge that demands the study of *algorithmic mechanism design*, the sub-field of algorithm design under the assumption that all agents are *selfish* and yet *rational*.

* The research of the author was supported in part by Grant FRG/03-04/II-21 and Grant RGC HKBU2107/04E.
** The research of the author was supported in part by NSF under Grant CCR-0311174.
*** The research of the author was supported in part by Grant RGC HKBU2159/04E.

This work focuses on developing new mechanisms for strategic games that can be formulated as different variants of the set cover problem, with three main objectives. First of all, each mechanism has to be *strategyproof*; that is, it shall provide incentives (such as payments made by service-receiving agents, or *service receivers*, to service-providing agents, or *service providers*) to ensure that agents are truthful and cooperative. Secondly, the outcome achieves an approximation of the optimal one with respect to the specified objective function. Thirdly, it is desirable that the mechanism is approximately *budget-balanced* so that the service receivers pay at least a bounded fraction of the total cost incurred by the service providers.

1.1 Set Cover Games

A *set cover game* can be generally defined as the following. Let $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ be a collection of multisets (or *sets* for short) of a universal set $U = \{e_1, e_2, \cdots, e_n\}$. Element e_i is specified with an *element coverage requirement* r_i (*i.e.*, it desires to be covered r_i times). The multiplicity of an element e_i in a set S_j is denoted by $k_{j,i}$. Let $d_{\max}$ be the maximum size of the sets in $\mathcal{S}$, *i.e.*, $d_{\max} = \max_j \sum_i k_{j,i}$. Each S_j is associated with a cost c_j. For any $\mathcal{X} \subseteq \mathcal{S}$, let $c(\mathcal{X})$ denote the total cost $\sum_{S_j \in \mathcal{X}} c_j$ of the sets in $\mathcal{X}$.

Many practical problems can be reasonably formulated as a set cover game defined above. For example, consider the following scenario: a business 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.
- ⋆ 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).

The *outcome* of the game is a *cover* $\mathcal{C}$, which is a subset of $\mathcal{S}$. The mechanism of the game is to determine an optimal (or approximately optimal) outcome of the game, according to a pre-defined objective function. For example, for set cover games where the sets are considered to be selfish agents whose total cost is to be minimized [1], the mechanism needs to compute a cover $\mathcal{C}_{opt}$ with the (approximately) minimum cost $c(\mathcal{C}_{opt})$, among all covers $\mathcal{C}$ such that $\sum_{S_j \in \mathcal{C}} k_{j,i} \geq r_i$.

There may be different variants of games according to various conditions (with different objective functions):

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.

1.2 Our Results and Organization of Paper

We design greedy set cover methods that are aware of the fact that the service receivers and/or the service providers are *selfish* (*i.e.*, they only care about their own benefits without consideration for the global performances or fairness issues) and *rational* (*i.e.*, they will always choose their actions to maximize their benefits). The study of selfish and rational agents participating in a cooperative or non-cooperative game is central to game theory. Two fundamental concepts in game theory are *Nash Equilibrium* and *dominant strategy*. Assume that there are n players. Given a set of actions $a = (a_1, a_2, \cdots, a_n)$, where player i chooses the action a_i, let $u(a) = (u_1(a), u_2(a), \cdots, u_n(a))$ be the payoffs vector: $u_i(a)$ is the payoff (or called profit, benefit) to the player i. An action vector a is called a *Nash Equilibrium* if no player can unilaterally switch its action to improve its benefit when the actions of other players are fixed. An action a_i is called a *dominant strategy* for player i if it maximizes its payoff regardless of the actions chosen by other players.

When the elements to be covered are selfish agents with privately known valuations, we first show that the strategyproof mechanism designed by a straightforward application of cross-monotonic cost-sharing scheme is not α-efficient for any $\alpha > 0$. We then present a strategyproof charging mechanism such that the total valuation of the elements covered is at least $\frac{1}{d_{\max}}$ times that of an optimal solution. This mechanism, however, may have free-riders: some elements do not have to pay at all and is still covered. We continue to present a strategyproof mechanism without free-riders and it is at least $\frac{1}{d_{\max} \ln d_{\max}}$-efficient. When the sets are also selfish agents with privately known costs, we show that the cross-monotonic *payment*-sharing scheme does not induce a strategyproof mechanism: a set could lie its cost downward to improve its utility. This is a sharp contrast to the theorem proved in [2] that a cross-monotonic *cost*-sharing scheme implies a strategyproof mechanism for selfish elements. The positive side is that the mechanism is still strategyproof for elements, *i.e.*, no element can lie about its bids to improve its utility.

The rest of the paper is organized as follows. In Section 2, we review the definitions of strategyproof mechanism and cost-sharing scheme. We also review the previous results on mechanism design and cost-sharing that are related to this paper. In Section 3, we give some general properties of the set cover games when the elements (also called service receivers) are selfish. In Section 4, we present a strategyproof mechanism for selfish receivers and we prove that the mechanism is approximately efficient. We then extend our results for the case that the elements may have multiple coverage requirement in Section 5. In Section 6, we study the scenario when both the service providers (*i.e.*, sets) and the service receivers (*i.e.*, elements) are selfish agents. We show that a cross-monotonic payment-sharing scheme does not induce a strategyproof mechanism. We conclude our paper in Section 7 with the discussion of some future works.

2 Preliminaries and Prior Art

A standard economic model for the design and analysis of scenarios in which the participants act according to their own self-interests is as follows. Assume that there are n agents $\{1, 2, \cdots, i, \cdots, n\}$, and each agent i has some *private* information t_i, called its *type*. For example, an agent could be a bidder in an auction, and its type is its valuation of an auctioned item. For direct-revelation mechanisms, the strategy of each agent i is to declare its type, although it may choose to report a carefully designed lie to influence the outcome of the game to its liking. For any vector $t = (t_1, t_2, \cdots, t_n)$ of reported types, the mechanism computes an output o as well as a payment p_i for each agent i. For each possible output o, agent i's preference is defined by a valuation function $v_i(t_i, o)$. The utility of agent i for the outcome of the game is defined to be $u_i = v_i(t_i, o) + p_i$.

An agent is called *rational*, if it always picks the the dominant strategy. A mechanism is *incentive compatible* (IC) if reporting its type truthfully is a dominant strategy for every agent. Another very common requirement in the literature for mechanism design is *individual rationality*: 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.

A classical result in mechanism design is the Vickrey-Clarke-Groves (VCG) mechanism by Vickrey [3], Clarke [4], and Groves [5]. The VCG mechanism applies to maximization problems where the objective function $g(o, t)$ is simply the sum of all agents' valuations. A VCG mechanism is always truthful [5], and is the only truthful implementation, under mild assumptions, to maximize the total valuation [6]. Although the family of VCG mechanisms is powerful, it has its limitations. To use a VCG mechanism, we have to compute the exact solution that maximizes the total valuation of all agents. This makes the mechanism computationally intractable for many optimization problems.

While designing feasible mechanisms for set cover games, we aim to achieve the following objectives, which are sometimes at odds with each other and thus require proper tradeoffs.

- ⋆ **Economic Efficiency** To make mechanisms tractable, we have to adopt approximation algorithms that compute only approximately optimal outcomes. We say that a mechanism is α-efficient if its output achieves a total valuation that is no less than α times the optimal total valuation. For VCG mechanisms, replacing the exact algorithm with an approximation algorithm usually destroys incentive compatibility [7]. In this case, we shall design new mechanisms that preserve incentive compatibility.
- ⋆ **Budget Balance** Frequently, a game involves a set of agents (service receivers) who are willing to pay for receiving services, and the mechanism needs to decide, based on the valuations of the services reported by all agents, the subset S of agents who shall receive services and how much they are charged. Let $C(S)$ be the total cost incurred by providing services to all agents in S. If $\xi_i(S)$ is the cost charged to each agent $i \in S$, we say that the cost-sharing method is budget-balanced if $\sum_{i \in S} \xi_i(S) = C(S)$. It has been proved to be im-

possible to achieve both budget balance and efficiency [2]. Thus, we may seek a β-budget-balanced cost-sharing method such that $\sum_{i \in S} \xi_i(S) \geq \beta \cdot C(S)$, for some $0 < \beta < 1$.

- ⋆ **Fair Cost-Sharing** Budget balance is only a measure of how good the cost-sharing method is from a global point of view. We also need to address how individual agent would view the cost-sharing method; we need to make the method fair, encouraging agents to participate. Besides the well accepted measures such as *group strategyproofness* (*i.e.*, for any group of agents who collude in revealing their valuations, if no member is made worse off, then no member is made better off) and *cross-monotonicity* (*i.e.*, the cost share of an agent should not go up if more players require the service), we also consider a less-studied measure, called *fairness under core* (*i.e.*, the cost shares paid by any subset of agents should not exceed the minimum cost of providing the service to them alone, hence they have no incentives to secede), which is derived from game theory concepts [8].
- ⋆ **No Positive Transfers (NPT)** The cost shares are non-negative.
- ⋆ **Voluntary Participation (VP)** The utility of each agent is guaranteed to be non-negative if an element reports its bid truthfully.
- ⋆ **Consumer Sovereignty (CS)** When an agent's bid is large enough, and others' bids are fixed, the agent will get the service.

Devanur *et al.* [9] studied the strategyproof cost-sharing mechanisms for set cover games, with elements considered to be selfish agents. In a game of this type, each element will declare its bid indicating its valuation of being covered, and the mechanism uses the greedy algorithm [10] to compute a cover with an approximately minimum total cost. Li *et al.* [1] extended this work by providing a strategyproof cost-sharing mechanism for multi-cover games. They also designed several cost-sharing schemes to fairly distribute the costs of the selected sets to the elements covered, for the case that both sets and elements are unselfish (*i.e.*, the will declare their costs/bids truthfully). The case of set cover games where sets are considered as selfish agents was also considered. Immorlica *et al.* [11] provided bounds on approximate budget balance for cross-monotone cost-sharing scheme for the set cover games.

3 Selfish Service Receivers

Typically, the objective function of a game is defined to be the total valuation of the agents selected by the outcome of the game. In set cover games, when sets are considered to be agents (*e.g.*, [1]), maximizing the total valuation of all selected agents is equivalent to minimizing the total cost of all selected sets. However, if the elements are considered to be agents, the objective becomes to maximize the total valuation of all elements (*i.e.*, the sum of all bids covered). Correspondingly, we need to solve the following optimization problem:

Problem 1. Each element e_i is associated with a *coverage requirement* r_i as well as a set of bids $B_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,r_i}\}$ such that $b_{i,1} \geq b_{i,2} \geq \cdots \geq b_{i,r_i}$. An *assignment* $\mathcal{C}$ is defined as the following:

(i) $\mathcal{C} \subseteq \mathcal{S}$;
(ii) a bid $b_{i,r}$ can be assigned to at most one set $S_{\pi(i,r)} \in \mathcal{C}$;
(iii) For any $S_j \in \mathcal{C}$, the *assigned value* $\nu_j(\mathcal{C}) = \sum_{\pi(i,r)=j} b_{i,r}$ is no less than c_j (S_j is "affordable");
(iv) $\kappa_{j,i} \leq k_{j,i}$, where $\kappa_{j,i}$ is the number of bids of e_i assigned to S_j;
(v) if the number γ_i of assigned bids of e_i is less than r_i, then the assigned bids must be the first γ_i bids (with the greatest bid values) of e_i.

The *total value* $V(\mathcal{C}) = \sum_{S_j \in \mathcal{C}} \nu_j(\mathcal{C})$ is the sum of all assigned bids in $\mathcal{C}$. The problem is to find an assignment with the maximum total value.

This problem is NP-hard. In fact, the weighted set packing problem, which is NP-complete, can be viewed as a special case of this problem, with $r_i = 1$ and $b_{i,1} = 1$ for each e_i and $c_j = |S_j|$ for each S_j. Therefore, the VCG mechanism cannot be used here if polynomial-time computability is required. In the rest of the paper, we concentrate on designing approximately efficient and polynomial-time computable mechanisms.

All our methods follow a round-based greedy approach: in each round t, we select some set S_{j_t} to cover some elements. After the s-th round, we define the *remaining required coverage* r'_i of an element e_i to be $r_i - \sum_{t'=1}^{s} \kappa_{j_{t'},i}$. 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 *effective value* (or *value* for short) v_j of S_j is therefore $\sum_{i=1}^{n} \sum_{r=1}^{k'_{j,i}} b_{i,r_i - r'_i + r}$ and it is *affordable* after s-th round if $v_j \geq c_j$.

One scheme is to select a set S_j as long as it is still affordable, and assign all appropriate bids to S_j. However, in this case an element may find it profitable to lie about its bid, as we will show in Section 4. An alternative scheme is to pick a set only if it is *individually affordable*, as defined as the following:

Definition 1. *A set S_j is individually affordable by d bids if it contains at least d bids each with a value no less than $\frac{c_j}{d}$, for some $d > 0$.*

Consequently, only the d largest bids are assigned to S_j, for the maximum d such that S_j is individually affordable by d bids. Notice that here an implicit assumption is that each set S_j can selectively provide coverage to a subset of elements contained by S_j. This is to prevent anybody from taking "free rides." The *modified value* $\tilde{v}_j$ of S_j is defined to be the total value of these bids.

In essence, this scheme is contradictory to our objective of maximizing total valuation. We throw away bids that can otherwise be assigned (without incurring any extra cost) to a set. Further, we may discard an affordable set with a value much greater than its cost (see the following lemma). However, to achieve strategyproofness while avoiding free riders, it is somewhat another form of "price of anarchy."

The following lemma gives upper bounds on the total value lost by enforcing individually affordable sets (see Appendix B for the proof):

Lemma 1. *For any set $S_j \in \mathcal{S}$,*
1.1) if S_j is individually affordable, the modified value $\tilde{v}_j$ is no less than $\frac{1}{\ln d_{\max}}$ fraction of its value v_j;

1.2) if S_j is not individually affordable, its value is no more than $\ln d_{\max}$ *times the cost c_j of S_j.*

The bound is tight, as we can have a set with a cost of $1+\epsilon$, and with $d_{\max}$ bids $\frac{1}{d_{\max}}, \frac{1}{d_{\max}-1}, \cdots, \frac{1}{2}, 1$.

4 Single Cover Games

In this section we first study the case where each element only needs to be covered once, *i.e.*, $r_i = 1$ for each $e_i \in U$. This corresponds to the traditional set cover problem.

An obvious solution to designing a strategyproof mechanism for single-cover set cover games is to use a cross-monotone cost-sharing scheme based on a theorem proved in [2]: a cross-monotone cost-sharing scheme implies a group-strategyproof mechanism when the cost function is submodular, non-negative, and non-decreasing. A cost function C is submodular if $C(T_1)+C(T_2) \geq C(T_1 \cup T_2) + C(T_1 \cap T_2)$ for any T_1, T_2. A cost function C is non-decreasing if $C(T_1) \leq C(T_2)$ for any $T_1 \subseteq T_2$. For set cover games, it is not difficult to show by example that the following cost functions are *not* submodular: the cost $c(\mathcal{C}_{opt})$ defined by the optimal cover $\mathcal{C}_{opt}$ of a set of elements, and the cost defined by the traditional greedy method (*i.e.*, in every round we select the set S_j with the minimum ratio of cost c_j over the number of elements covered by S_j and not covered by sets selected before)[3]. Even if a cost function is submodular, sometimes it may be NP-hard to compute this cost, and thus we cannot use this cost function to design a strategyproof mechanism. It was shown in [1] that there is a cost function that is indeed submodular: for each element $e_i \in T$, we select the set S_j with the least cost that covers e_i. Let $\mathcal{C}_{lcs}(T)$ be all sets selected as above to cover a set of elements T. Then $c(\mathcal{C}_{lcs})$ is submodular, non-decreasing, and non-negative. Notice that, if it is a multi-cover set cover game, each set S_j is only eligible to cover an element e_i $k_{j,i}$ times.

Given the cost function $c(\mathcal{C}_{lcs})$, it was shown in [1] that the cost-sharing method $\xi_i(T)$, defined as $\xi_i(T) = \sum_{S_j \in \mathcal{C}_{lcs}(T)} \frac{\kappa_{j,i} \cdot c_j}{\sum_a \kappa_{j,a}}$, is budget-balanced, cross-monotone and a $\frac{1}{2n}$-core. Here $\kappa_{j,i}$ is the number of bids of e_i assigned to S_j. For a single-cover set cover game, based on the method described in [2], given the single bid b_i by each element e_i, we can define a mechanism $M(\xi)$ as follows.

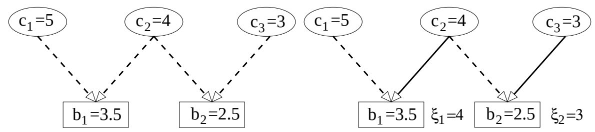
The following theorem is directly implied by the result in [2].

Theorem 1. *The mechanism $M(\xi)$ is group-strategyproof, budget-balanced, and meets NPT, CS, and VP.*

However, this mechanism is not *efficient* at all: we will show by example that it is possible that the total valuation achieved by this mechanism is 0 while the maximum total valuation achieved is a positive number. In other words, this mechanism cannot be α-efficient for any $\alpha > 0$. Figure 1 illustrates such

Algorithm 1 Mechanism for single cover games via cost-sharing.

$S^0 = U$; $t = 0$;
repeat
  $S^{t+1} = \{e_i \mid b_i \geq \xi_i(S^t)\}$; $t = t + 1$;
until $S^{t-1} = S^t$
The output of mechanism $M(\xi)$ is $\tilde{U}(\xi, b) = S^t$,
The charge by $M(\xi)$ to an element e_i is $\xi_i(\tilde{U}(\xi, b))$.



(a) original game (b) LCS output

Fig. 1. An example that the mechanism $M(\xi)$ is not efficient. In all figures, sets are represented by ovals while elements are represented by rectangles. A dashed link (with arrow) between an oval and a rectangle denotes that the set contains one copy of the element. A solid link (with arrow) between an oval and a rectangle denotes that the set is selected to cover the element.

an example. It is easy to show that no element will be selected by mechanism $M(\xi)$. On the other hand, if we choose S_2 to cover elements $\{e_1, e_2\}$ and charge each elements $\frac{1}{2} \cdot c_2 = 2$, each element has a positive utility and the game has its maximum total valuation $3.5 + 2.5 = 6$.

Next, in Algorithm 2, we describe a new greedy algorithm that computes for a single cover game an approximately optimal assignment $\mathcal{C}_{grd}$. Starting with $\mathcal{C}_{grd} = \emptyset$, in each round t' the algorithm adds to $\mathcal{C}_{grd}$ a set $S_{j_{t'}}$ with the maximum effective value.

The following theorem establishes an approximation bound for the algorithm (see Appendix B for the proof).

Theorem 2. *Algorithm 2 computes an assignment* $\mathcal{C}_{grd}$ *with a total value* $V(\mathcal{C}_{grd}) \geq \frac{1}{d_{\max}} \cdot V(\mathcal{C}_{opt})$.

It is easy to show that the above bound is tight.

Next we show how to compute the payment charged to each element in a strategyproof mechanism. In many problems, the total payment is often less than the total cost incurred by the service providers in order to guarantee strategyproofness and therefore the mechanisms are not budget-balanced. However, it is important to note that even in this scenario we still want to guarantee that the total valuation of the service receivers covered by any particular service provider is no less than the cost of this service provider; otherwise it is not worthwhile to select this service provider in terms of the social efficiency. When the mechanism

[3] Notice that the greedy method we will present later is different from this traditional greedy set cover method.

Algorithm 2 Greedy algorithm for single cover games.

$\mathcal{C}_{grd} \leftarrow \emptyset$.
for all $S_j \in \mathcal{S}$ **do**
    compute effective value v_j.
while $\mathcal{S} \neq \emptyset$ **do**
    pick set S_t in $\mathcal{S}$ with the maximum effective value v_t.
    $\mathcal{C}_{grd} \leftarrow \mathcal{C}_{grd} \cup \{S_t\}$, $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_t\}$.
    for all $e_i \in S_t$ **do**
        $\pi(i,1) \leftarrow t$.
        remove e_i from all $S_j \in \mathcal{S}$.
    for all $S_j \in \mathcal{S}$ **do**
        update effective value v_j.
        if $v_j < c_j$ **then**
            $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_j\}$.

runs into deficit, it is traditionally assumed that there is an outsider banker (*e.g.*, the government) who will subsidize the costs of the sets.

The payment $p_{i,1}$ of each bid $b_{i,1}$ can be decided according to Algorithm 3 using a round-based approach. Algorithm 3 examines all possible cases that an element e_i can lie about its bid $b_{i,1}$ while still ensuring that $b_{i,1}$ is assigned to a set in $\mathcal{C}_{grd}$, and charge e_i the minimum bid value in all these cases.

Intuitively, Algorithm 3 runs Algorithm 2 without the participation of e_i (*i.e.*, not including $b_{i,1}$ when evaluating the value v_j of each $S_j \ni e_i$). As e_i is "watching" the set selection process, every time a set S_t is picked, it would record for each set $S_j \ni e_i$, how much it needs to raise its bid $b_{i,1}$ so that S_j can beat S_t in this round (so that S_j is selected and consequently $b_{i,1}$ is assigned), as shown in Line 16 of Algorithm 3.

Just like in Algorithm 2, we maintain a priority queue containing all sets using their values as keys, so that in each round we can extract the set with the maximum value. However, when a set $S_j \ni e_i$ becomes unaffordable (because of losing bids to sets already picked), we need to handle it differently. In this case, e_i has to raise its bid $b_{i,1}$ at least to $c_j - v_j$; otherwise S_j will still not be qualified to be selected. To beat a set S_t being picked, e_i might have to raise its bid even further, a situation already handled in Line 16. On the other hand, with a value equal to c_j, it may already be sufficient for S_j to get picked; in this case, e_i does not need to report a bid more than $c_j - (v_j - b_{i,1})$. This is handled in Line 12.

We have the following theorem on the above cost-sharing mechanism:

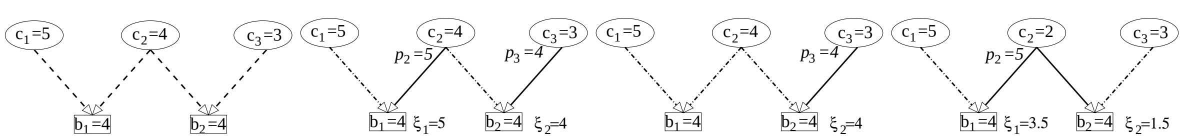
Theorem 3. *The cost-sharing mechanism defined in Algorithm 3 is strategyproof.*

Proof. It is easy to show that Algorithm 3 actually computes the minimum bid that an agent can report such that it is still selected in the outcome if it is originally selected. Since the set-cover game is a demand game and Algorithm 2 satisfies a certain monotone property defined in [12], the result proved in [12] implies that Algorithm 3 is strategyproof. ◻

Algorithm 3 Computing payment $p_{i,1}$ of e_i in single cover games.

$p_{i,1} \leftarrow +\infty$; $\mathcal{S}' \leftarrow \emptyset$; $\mathcal{S}'' \leftarrow \emptyset$; $\mathcal{C}_{grd} \leftarrow \emptyset$.
for all $S_j \in \mathcal{S}$ **do**
  compute value v_j.
  if $e_i \in S_j$ **then**
    $w_j \leftarrow \max\{v_j - b_{i,1}, c_j\}$, $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S_j\}$.
  else
    $w_j \leftarrow v_j$, $\mathcal{S}'' \leftarrow \mathcal{S}'' \cup \{S_j\}$.
while $\mathcal{S}' \neq \emptyset$ **do**
  pick set S_t in $\mathcal{S}' \cup \mathcal{S}''$ with the maximum w_t.
  if $S_t \in \mathcal{S}'$ **then**
    $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{S_t\}$.
    $p_{i,1} \leftarrow \min\{p_{i,1}, w_t - (v_t - b_{i,1})\}$.
  else
    $\mathcal{S}'' \leftarrow \mathcal{S}'' \setminus \{S_t\}$; $\mathcal{C}_{grd} \leftarrow \mathcal{C}_{grd} \cup \{S_t\}$.
    for all $S_j \in \mathcal{S}'$ **do**
      $p_{i,1} \leftarrow \min\{p_{i,1}, v_t - (v_j - b_{i,1})\}$.
    for all $e_x \in S_t$ **do**
      remove e_x from all $S_j \in \mathcal{S}' \cup \mathcal{S}''$.
    for all $S_j \in \mathcal{S}' \cup \in \mathcal{S}''$ **do**
      update v_j and w_j.
      if $S_j \in \mathcal{S}''$ **and** $v_j < c_j$ **then**
        $\mathcal{S}'' \leftarrow \mathcal{S}'' \setminus \{S_j\}$.
      if $S_j \in \mathcal{S}'$ **and** $v_j + p_{i,1} < c_j$ **then**
        $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{S_j\}$.

Notice that Algorithm 2 and Algorithm 3 together may produce an output such that the payment by a certain element is 0. For example, see the set cover game illustrated by Figure 2 (a). It is easy to show that, according to Algorithm 3, the payments by both elements e_1 and e_2 are 0 since each element can lie its bid to as low as 0 and still get covered.



(a) sets-elements (b) LCS output (c) selected elements (d) output if S_2 lies

Fig. 2. An example that a set can lie its cost to improve its utility when LCS is used as output.

To avoid this zero payment problem, we use a slightly different algorithm to determine the outcome of the game. Our modified greedy method (described in Algorithm 4) instead only selects individually affordable sets. When a set S_j is added into $\mathcal{C}_{grd}$, the algorithm only assigns to S_j the largest d bids, such that S_j is individually affordable with d bids, for the maximum such d.

Algorithm 4 Improved greedy algorithm for single cover games.

$\mathcal{C}_{grd} \leftarrow \emptyset$.
for all $S_j \in \mathcal{S}$ **do**
    compute the modified value $\tilde{v}_j$.
while $\mathcal{S} \neq \emptyset$ **do**
    pick set S_t in $\mathcal{S}$ with the maximum modified value $\tilde{v}_t$.
    $\mathcal{C}_{grd} \leftarrow \mathcal{C}_{grd} \cup \{S_t\}$, $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_t\}$.
    $d_t \leftarrow$ the largest d such that the set S_t is individually affordable by d largest unsatisfied bids.
    for all $e_i \in S_t$ **do**
        if $b_{i,1}$ is one of the largest d_t unsatisfied bids in S_t **then**
            $\pi(i,1) \leftarrow t$
            remove e_i from all $S_j \in \mathcal{S}$.
    for all $S_j \in \mathcal{S}$ **do**
        update the modified value $\tilde{v}_j$.
        if $\tilde{v}_j < c_j$ **then**
            $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_j\}$.

Obviously, Algorithm 4 satisfies the monotone property defined in [12]: when an element e_i was selected with a bid $b_{i,1}$, then it will always be selected with a bid $\overline{b}_{i,1} > b_{i,1}$. This monotone property implies that there is always a strategyproof mechanism using Algorithm 4 to compute its output. It is easy to show that Algorithm 4 is a round-based greedy method that satisfies the cross-independence property defined in [12]. Thus, the payment to each element can always be computed in polynomial time.

We have the following theorem on the approximate efficiency of the modified greedy algorithm:

Theorem 4. *When only individually affordable sets are allowed to be picked, the assignment $\mathcal{C}_{grd}$ computed by Algorithm 4 has a total value*

4.1) no less than $\frac{1}{d_{\max}} \cdot V(\mathcal{C}_{opt})$, if the optimal assignment $\mathcal{C}_{opt}$ also allows only individually affordable sets;

4.2) no less than $\frac{1}{2d_{\max}} \cdot V(\mathcal{C}_{opt})$, if the optimal assignment $\mathcal{C}_{opt}$ allows sets that are not individually affordable, but all sets in $\mathcal{S}$ are individually affordable initially.

To compute the payment of each element e_i for its assigned bid, we use an algorithm similar to Algorithm 3. The only differences are:

⋆ in Line 3 we compute the modified values for the sets;

⋆ in Line 5 we replace $v_j - b_{i,1}$ by the modified value of S_j after bid $b_{i,1}$ is removed (or, equivalently, $b_{i,1}$ is considered to be 0);

- ⋆ in Line 12, e_i has to raise its bid $b_{i,1}$ not only to make sure that S_t is individually affordable, but also to let $b_{i,1}$ become one of the largest d bids in S_t, for some $d > 0$ such that S_t is individually affordable by d bids;
- ⋆ in Line 16, e_i has to raise its bid $b_{i,1}$ to make sure that the modified value of a certain set $S_j \ni e_i$ is larger than the modified value of the set S_t currently being selected.

Furthermore when an element e_i is covered by a set that is individually affordable by d elements, then the bid of e_i cannot be less than c_j/d for some S_j, which is not necessarily the set covering e_i. Thus, we know the payment by element e_i is at least c_j/d, which prevents free-riders.

5 Multi-cover Games

Theorem 2 and Theorem 4 can easily be extended to the case of multi-cover. However, when it comes to computing payments, there is a problem: in the multi-cover case, an element can lie in different ways, and it may not be of its best interest if it achieves the maximum utility in the first bid (or the last bid). In that case, how can we compute payments efficiently?

In this section we study the multi-cover games. To overcome the computational complexity of computing payments, we need to instead use a different greedy algorithm to compute the outcome of the game. This algorithm is the same as Algorithm 3 of [1]. For the completeness of our presentation, we include the algorithm (with minor notational changes, and listed as Algorithm 5) in Appendix Section A.

In [1] it is shown that this mechanism produces an outcome with a total cost no more than $\ln d_{\max}$ times the total cost of an optimal outcome. In the following we show that the outcome is also approximately efficient with respect to the total valuation of the assigned (covered) bids (see Appendix Section B for the proof).

Theorem 5. *Algorithm 5 (Algorithm 3 of [1]) defines a budget-balanced and strategyproof mechanism. Further, it is $\frac{1}{d_{\max} H_{d_{\max}}}$-efficient, if all sets are individually affordable initially.*

The above bound is tight. We may construct an example with $d_{\max}^2$ elements, $e_{1,1},\ e_{1,2}, \cdots,\ e_{d_{\max},d_{\max}}$, and $d_{\max}+1$ sets, $S_j = \{e_{j,1},\ e_{j,2}, \cdots,\ e_{j,d_{\max}}\}$ for $1 \le j \le d_{\max}$ and $S_{d_{\max}+1} = \{e_{1,1},\ e_{2,1}, \cdots,\ e_{d_{\max},1}\}$. The bid for each element $e_{u,v}$ is $1+\epsilon$ if $v = 1$ or $\frac{d_{\max}}{d_{\max}-v+1}$ if $v > 1$, and the cost of each set S_j is $d_{\max}+\epsilon$ if $j \le d_{\max}$ or $\frac{d_{\max}}{2}$ if $j = d_{\max}+1$. Obviously all these sets are individually affordable at the beginning. Algorithm 5 picks set $S_{d_{\max}+1}$ first, because its average shared cost, which is $\frac{1}{2}$, is the smallest among all sets. However, once $S_{d_{\max}+1}$ is added into $\mathcal{C}_{grd}$, none of the remaining sets is individually affordable, and thus the algorithm terminates with an assignment with a value $d_{\max} \cdot (1+\epsilon)$. The optimal assignment is to select sets $S_1, S_2, \cdots, S_{d_{\max}}$, with a total value of $d_{\max}^2 \cdot (H_{d_{\max}} + \epsilon)$.

6 Selfish Service Providers and Receivers

So far, we assume that the cost of each set is publicly known or each set will truthfully declare its cost. In practice, it is possible that each set could also be a selfish agent that will maximize its own benefit, *i.e.*, it will provide the service only if it receives a payment by some elements (not necessarily the elements covered by itself) large enough to cover its cost. In [1], Li *et al.* designed several truthful payment schemes to selfish sets such that each set maximizes its utility when it truthfully declares its cost and the covered elements will pay whatever a charge computed by the mechanism. They also designed a payment sharing scheme that is budget-balanced and in the core.

To complete the study, in this section, we study the scenario when both the sets and the elements are individual selfish agents: each set S_j has a privately known cost c_j, while each element e_i has a privately known 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 $b_{i,r}$. It is well-known that a cross-monotone *cost* sharing scheme implies a strategyproof mechanism [2]. Unfortunately, since the sets are selfish agents, it is impossible to design any cost-sharing scheme here, and the best we can do is to design some payment sharing scheme. It was shown in [13] that a cross-monotone payment sharing scheme does *not* necessarily induce a strategyproof mechanism by using multicast as a running example: a relay node could lie its cost upward or downward to improve its utility.

Given a subset of elements $T \subseteq U$ and their coverage requirement r_i for $e_i \in T$, a collection of multisets $\mathcal{S}$, and each set $S_j \in \mathcal{S}$ with cost c_j, let M_S be a strategyproof mechanism that will determine which sets from $\mathcal{S}$ will be selected to provide the coverage to *all* elements T, and the payment p_j to each set S_j. We assume that the mechanism is normalized: the payment to a unselected set S_j is always 0. Based on two monotonic output methods, the traditional greedy set cover method (denoted as GRD) and the least cost set method (denoted as LCS) for each element, Li *et al.* [1] designed two strategyproof mechanisms for set cover games.

Let $E(S_j, c, T, M_S)$ be the set of elements covered by S_j in the output of M_S. In the remaining of the paper, we assume that the mechanism M_S satisfies the property that if a set S_j increases its cost then the set of elements covered by S_j in the output of M_S will *not* increase, *i.e.*, $E(S_j, c|^j d, T, M_S) \subseteq E(S_j, c, T, M_S)$ for $d > c_j$. This property is satisfied by all methods currently known for set cover games.

Let $\xi_{i,j}(T)$ be the shared payment by element e_i for its jth copy when the set of elements to be covered is T, given a strategyproof payment scheme M_S to all sets. Following the method described in [2], given the set U of n elements and their bids $B_1, \cdots, B_n$ we can compute the outcome $\tilde{U}(\xi, B)$ as the limit of the following inclusion monotonic sequence: $S^0 = U$; $S^{t+1} = \{e_i \mid b_{i,j} \geq \xi_{i,j}(S^t)\}$. Notice that here we have to recompute the payments to all sets, and thus the shared payments by all elements, when the set of elements to be covered changed from S^t to S^{t+1}. In other words, we define a mechanism $M_E(\xi)$ associated with the payment sharing method ξ as follows: the set of elements to be covered is

$\tilde{U}(\xi, B)$, the charge to element e_i is $\xi_{i,j}(\tilde{U}(\xi, B))$ if $e_i \in \tilde{U}(\xi, B)$; otherwise its charge is 0. Based on the strategyproof mechanism using LCS as output for set cover games, Li *et al.* [1] designed a payment sharing mechanism that is budget-balanced, cross-monotone, and in the core.

In the remaining of the paper, we assume that for the payment-sharing mechanism ξ, the payment p_j to the set S_j is only shared among the elements, *i.e.*, $E(S_j, c, T, M_S)$, covered by S_j. This property is satisfied by the payment-sharing methods studied in [1] for set cover games.

For the set cover games, we prove the following theorem (see Appendix Section B):

Theorem 6. *For set cover games with selfish sets and elements, a strategyproof mechanism M_S to sets and a cross-monotone payment sharing scheme ξ imply that in mechanism M_E each set S_j cannot improve its utility by lying upward its cost.*

Unfortunately, for set cover games, we show that a strategyproof mechanism M_S to sets and a cross-monotone payment sharing scheme ξ do *not* induce a strategyproof mechanism M_E for each element. Figure 2 illustrates such an example when LCS is used as the output, a set s_j can lie its cost downward to improve its utility from 0 to $p_j - c_j$. A similar example can be constructed when the traditional greedy method is used as the output. When set S_2 is truthful, although LCS will select it to cover element e_1 with payment $p_2 = 5$, but the corresponding sharing by e_1 is $\xi_1 = 5$, which is larger then its bid $b_{1,1} = 4$. Consequently, set S_2 will not be selected and element e_1 will not be covered (see Figure 2 (c)). On the other hand, if S_2 lies its cost downward to $\bar{c}_2 = 2$, its payment is still $p_2 = 5$, but now, since it covers elements e_1 and e_2, the shared payments by e_1 and e_2 become $\xi_1 = 3.5$ and $\xi_2 = 1.5$. Thus, the set S_2 becomes affordable by elements e_1 and e_2.

We leave it as future work to study whether there exists a strategyproof mechanism to select selfish sets to cover selfish elements using the combination of a strategyproof mechanism for sets, and a good payment-sharing method for elements. Notice that since this is still a binary-demand game [12], any truthful mechanism must use an output method that is monotone for both the sets and the elements: when a selected set decreases its cost, it will still be selected to provide service; when a selected receiver increases its bid, it will still be selected to receive service.

7 Conclusion

Strategyproof mechanism design has attracted a significant amount of attentions recently in several research communities. In this paper, we focused the set cover games when the elements are selfish agents with privately known valuations of being covered. We presented several (approximately budget-balanced) strategyproof mechanisms that are approximately efficient, which are summarized in Table 1. When the service providers (*i.e.* sets) are also selfish, we show that a cross-monotonic *payment*-sharing scheme does not necessarily induce a strategyproof mechanism. This is a sharp contrast to the well-known fact [2] that a cross-monotonic *cost*-sharing scheme always implies a strategyproof mechanism.

Table 1. Summary of mechanisms presented in this paper.

Mechanism	Efficiency	Budget-Balance	Truthful
Alg 1	0	1	Group-Strategyproof
Alg (2, 3)	$\frac{1}{d_{\max}}$	0	Strategyproof
Alg 4	$\frac{1}{2d_{\max}}$	> 0	Strategyproof
Alg 5	$\frac{1}{d_{\max} \cdot H_{d_{\max}}}$	1	Strategyproof

This paper does not intend to solve all problems related to designing strategyproof mechanisms for set cover games. There are several interesting and also important problems left open for future works.

1. Whether the approximation bounds of efficiency given by several strategyproof mechanisms are tight? Notice that we showed that these bounds are tight for these mechanisms presented here. It is unknown whether there exist some other mechanisms with asymptotically better approximation bounds on efficiency.
2. It is well-known that there is no mechanism that is both efficient and budget-balanced. Then what is the best possible tradeoffs between the efficiency and the budget-balance. It there any bound on $\alpha \cdot \beta$ for an α-efficient and β-budget-balanced mechanisms for set cover games? We know for sure that $\frac{1}{d_{\max} \cdot H_{d_{\max}}} \leq \alpha \cdot \beta < 1$ when the original optimal solution only admits individually affordable sets.
3. What are the necessary and/or sufficient conditions for a strategyproof mechanism M_S for selfish sets and a payment sharing scheme ξ such that the induced mechanism M_E discussed in Section 6 is strategyproof?
4. The last question is, when both the providers and the elements are selfish agents, to design a strategyproof mechanism (not necessarily using the approach discussed in Section 6) that is approximately efficient. Remember that the total efficiency of an output of this game now becomes the total valuation of selected to-be-covered elements minus the total cost of the selected sets that cover these elements.

References

1. Li, X.Y., Sun, Z., Wang, W.: Cost sharing and strategyproof mechanisms for set cover games. In: Proceedings of the 22nd International Symposium on Theoretical Aspects of Computer Science. Volume 3404 of Lecture Notes in Computer Science. (2005) 218–230
2. Moulin, H., Shenker, S.: Strategyproof sharing of submodular costs: budget balance versus efficiency. Economic Theory **18** (2001) 511–533

3. Vickrey, W.: Counterspeculation, auctions and competitive sealed tenders. Journal of Finance **16** (1961) 8–37
4. Clarke, E.H.: Multipart pricing of public goods. Public Choice **11** (1971) 17–33
5. Groves, T.: Incentives in teams. Econometrica **41** (1973) 617–631
6. Green, J., Laffont, J.J.: Characterization of satisfactory mechanisms for the revelation of preferences for public goods. Econometrica **45** (1977) 427–438
7. Nisan, N., Ronen, A.: Computationally feasible VCG mechanisms. In: Proceedings of the 2nd ACM Conference on Electronic Commerce. (2000) 242–252
8. Osborne, M.J., Rubinstein, A.: A course in game theory. The MIT Press (1994)
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. Chvátal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operation Research **4** (1979) 233–235
11. 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)
12. Kao, M.Y., Li, X.Y., Wang, W.: Toward truthful mechanisms for binary demand games: A general framework. In: Proceedings of the 6th ACM Conference on Electronic Commerce. (2005) To appear.
13. Wang, W., Li, X.Y., Sun, Z., Wang, Y.: Design multicast protocols for non-cooperative networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Communication Society (INFOCOM). (2005) To appear.

A Strategyproof Charging Mechanism for Multi-cover Games

Algorithm 5 Strategyproof charging mechanism for multi-cover games.

$\mathcal{C}_{grd} \leftarrow \emptyset$.
for all $e_i \in U$ **do**
  $r'_i \leftarrow r_i$.
while $\mathcal{S} \neq \emptyset$ **do**
  pick an individually affordable set $S_t \in \mathcal{S}$ (by d bids) with the smallest average cost $\frac{c_t}{d}$.
  $\mathcal{C}_{grd} \leftarrow \mathcal{C}_{grd} \cup \{S_t\}$, $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_t\}$.
  for all element $e_i \in S_t$ with $b_{r_i - r'_i + 1} \geq \frac{c_t}{d}$ **do**
    $p_{i, r_i - r'_i + 1} \leftarrow \frac{c_t}{d}$.
    $\pi(i, r_i - r'_i + 1) \leftarrow t$.
    $r'_i \leftarrow r'_i - 1$.
  for all $S_j \in \mathcal{S}$ **do**
    update value $\tilde{v}_j$.
    if $\tilde{v}_j < c_j$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_j\}$.

B Proofs

Lemma 1 *For any set $S_j \in \mathcal{S}$,*
1.1) if S_j is individually affordable, the modified value $\tilde{v}_j$ is no less than $\frac{1}{\ln d_{\max}}$ fraction of its value v_j;
1.2) if S_j is not individually affordable, its value is no more than $\ln d_{\max}$ times the cost c_j of S_j.

Proof. Let $b_1, b_2, \cdots, b_x$ be the bids currently contained in S_j. Without loss of generality, we assume that $b_1 \leq b_2 \leq \cdots \leq b_x$. If S_j is individually affordable by d bids but not by $d+1$ bids, we have the following inequalities: 1) $b_r < \frac{c_j}{x+1-r}, \forall r \leq x - d$; 2) $b_r \geq \frac{c_j}{d}, \forall r > x - d$.

Obviously, $c_j \leq d \cdot b_{x-d+1} \leq \sum_{r=x-d+1}^{x} b_r = \tilde{v}_j$. Therefore, we have

$$
\begin{aligned}
v_j = \sum_{r=1}^{x-d} b_r + \sum_{r=x-d+1}^{x} b_r &< \sum_{r=1}^{x-d} \frac{c_j}{x+1-r} + \tilde{v}_j \\
&\leq (\ln x - 1) \cdot c_j + \tilde{v}_j \leq \ln x \cdot \tilde{v}_j \leq \ln d_{\max} \cdot \tilde{v}_j .
\end{aligned}
$$

This proves Lemma 1.1.

Lemma 1.2 can be proved similarly. □

Theorem 2 *Algorithm 2 computes an assignment $\mathcal{C}_{grd}$ with a total value $V(\mathcal{C}_{grd}) \geq \frac{1}{d_{\max}} \cdot V(\mathcal{C}_{opt})$.*

PROOF. Let S_k be a set selected by $\mathcal{C}_{opt}$ (with some assigned bids). Before Algorithm 2 adds any set to $\mathcal{C}_{grd}$, S_k is affordable. When the algorithm finishes, no more set in $\mathcal{S} \setminus \mathcal{C}_{grd}$ is affordable, and therefore at least one bid assigned to S_k in $\mathcal{C}_{opt}$ must have been assigned to a set in $\mathcal{C}_{grd}$ (which could be S_k itself).

Let S_{a_k} be the first set in $\mathcal{C}_{grd}$ that takes bid(s) assigned to S_k in $\mathcal{C}_{opt}$. Consider the current value v_{a_k} of S_{a_k} and the current value v_k of S_k right before S_{a_k} is added into $\mathcal{C}_{grd}$. Since till now no set in $\mathcal{C}_{grd}$ has taken a bid assigned to S_k in $\mathcal{C}_{opt}$, v_k should be no less than the assigned value $\nu_k(\mathcal{C}_{opt})$ of S_k in $\mathcal{C}_{opt}$. However, by the greedy nature of our algorithm, $v_k \le v_{a_k}$. Therefore, we have $\nu_k(\mathcal{C}_{opt}) \le v_k \le v_{a_k} = \nu_{a_k}(\mathcal{C}_{grd})$, as we assign all existing bids in S_{a_k} to it when we add S_{a_k} into $\mathcal{C}_{grd}$ (see Line 8 of Algorithm 2).

This way, we can "charge" the assigned value $\nu_k(\mathcal{C}_{opt})$ of each set $S_k \in \mathcal{C}_{opt}$ to a set $S_{a_k} \in \mathcal{C}_{grd}$ with at least the same assigned value. Since each set in $\mathcal{C}_{grd}$ can only take bids assigned to at most $d_{\max}$ sets in $\mathcal{C}_{opt}$ (and hence be charged at most $d_{\max}$ times), the total value $V(\mathcal{C}_{opt})$ of $\mathcal{C}_{opt}$ is no less than $d_{\max}$ times the total value $V(\mathcal{C}_{grd})$ of $\mathcal{C}_{grd}$. □

Theorem 4 *When only individually affordable sets are allowed to be picked, the assignment $\mathcal{C}_{grd}$ computed by Algorithm 4 has a total value*

4.1) no less than $\frac{1}{d_{\max}} \cdot V(\mathcal{C}_{opt})$, if the optimal assignment $\mathcal{C}_{opt}$ also allows only individually affordable sets;

4.2) no less than $\frac{1}{2d_{\max}} \cdot V(\mathcal{C}_{opt})$, if the optimal assignment $\mathcal{C}_{opt}$ allows sets that are not individually affordable, but all sets in $\mathcal{S}$ are individually affordable initially.

PROOF. The proof of Theorem 4.1 is similar to that of Theorem 2 and thus is omitted. Here we prove Theorem 4.2. Let S_k be a set selected by $\mathcal{C}_{opt}$, and let $b_1, b_2, \cdots, b_x$ be all bids initially contained in S_k (not necessarily assigned to S_k in $\mathcal{C}_{opt}$). Since S_k is individually affordable at the beginning, there exists a d such that: 1) $b_r < \frac{c_k}{x+1-r}, \forall r \le x-d$; 2) $b_r \ge \frac{c_k}{d}, \forall r > x-d$. Therefore, the value of S_k is $v_k = \sum_{r=1}^{x} b_r$, and the modified value of S_k is $\tilde{v}_k = \sum_{r=x-d+1}^{x} b_r$.

Again, after the greedy algorithm finishes, S_k must have at least one of the bids $b_{x-d+1}, b_{x-d+2}, \cdots, b_x$ assigned to a set added into $\mathcal{C}_{grd}$. Let S_{a_k} be the first set chosen by $\mathcal{C}_{grd}$ that takes a bid b_y, where $x-d+1 \le y \le x$. Then, at the moment S_{a_k} is selected, we have $\tilde{v}_k \le \tilde{v}_{a_k} = \nu_{a_k}(\mathcal{C}_{grd})$ due to the nature of the greedy algorithm. Further, since $b_y \ge \frac{c_k}{d}$, we have

$$\sum_{r=1}^{x-d} b_r < c_k \cdot (\sum_{r=1}^{x-d} \frac{1}{x+1-r}) \le b_y \cdot d \cdot (\sum_{r=1}^{x-d} \frac{1}{x+1-r}) = b_y \cdot d \cdot (H_x - H_d)$$
$$< b_y \cdot d \cdot (1 + \ln x - \ln d) \le b_y \cdot x \le b_y \cdot d_{\max}.$$

Here H_x is the harmonic function, *i.e.*, $H_x = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{x}$. Therefore, to bound the value v_k of each $S_k \in \mathcal{C}_{opt}$, we split it into two different parts and bound them separately: the part $\tilde{v}_k$ is no more than $\nu_{a_k}(\mathcal{C}_{grd})$, and the remaining part $v_k - \tilde{v}_k$ (which is exactly the sum of the bids $b_1, \cdots, b_{x-d}$) is no more than $d_{\max} \cdot b_y$.

Now consider each set $S_q \in \mathcal{C}_{grd}$. It is assigned with no more than $d_{\max}$ bids that are assigned to different sets in $\mathcal{C}_{opt}$, and therefore may be charged at most $d_{\max}$ times for its assigned value $\nu_{a_k}(\mathcal{C}_{grd})$. Further, for each of its assigned bids b_z, S_q may be charged for $d_{\max}$ times b_z, if b_z is assigned to a set in $\mathcal{C}_{opt}$. Therefore, in total each $S_q \in \mathcal{C}_{grd}$ is charged for at most $2d_{\max}$ times its assigned value, implying that

$$2d_{\max} \cdot V(\mathcal{C}_{grd}) = 2d_{\max} \cdot \sum_{S_q \in \mathcal{C}_{grd}} \tilde{v}_q \geq \sum_{S_k \in \mathcal{C}_{opt}} v_k \geq \sum_{S_k \in \mathcal{C}_{opt}} \nu_k(\mathcal{C}_{opt}) = V(\mathcal{C}_{opt}).$$

This finishes the proof. □

Theorem 5 *Algorithm 5 (Algorithm 3 of [1]) defines a budget-balanced and strategyproof mechanism. Further, it is* $\frac{1}{d_{\max} H_{d_{\max}}}$*-efficient, if all sets are individually affordable initially.*

Proof. The budget-balance part is obvious. The proof for strategyproofness is the same as in [1]. In the following we prove that this mechanism is $\frac{1}{d_{\max} H_{d_{\max}}}$ efficiency. Let S_k be a set in $\mathcal{C}_{opt}$. When Algorithm 5 finishes, at least one bid assigned to S_k in $\mathcal{C}_{opt}$ must have been assigned to a set in $\mathcal{C}_{grd}$. Otherwise, due to the monotonicity of the bids, for each element e_i, the currently available bids should be no less than the ones assigned to S_k in $\mathcal{C}_{opt}$. This implies that S_k is still individually affordable, a contradiction.

Let $B_k = \{b_1, b_2, \cdots, b_x\}$ be all x bids assigned to S_k in $\mathcal{C}_{opt}$ and without loss of generality let $B'_k = \{b_1, b_2, \cdots, b_y\}$ be the subset of B_k containing all bids already assigned to sets in $\mathcal{C}_{grd}$ right after S_k becomes individually unaffordable. For our convenience, we assume that $b_1 \leq b_2 \leq \cdots \leq b_y$. Clearly, $b_{y+1}, b_{y+2}, \cdots, b_x$ cannot make S_k individually affordable. On the other hand, we claim that b_y belongs to a subset of S_k that makes S_k individually affordable; otherwise, losing B'_k will not make S_k individually unaffordable. Hence, we have 1) $\sum_{r=y+1}^{x} b_r < \sum_{r=y+1}^{x} \frac{c_k}{x-r+1}$, 2) $b_y \geq \frac{c_k}{d_{\max}}$. Therefore,

$$\begin{aligned}
\nu_k(\mathcal{C}_{opt}) &= \sum_{r=1}^{y} b_r + \sum_{r=y+1}^{x} b_r \leq \sum_{r=1}^{y} b_y + \sum_{r=y+1}^{x} \frac{c_k}{x-r+1} \\
&\leq b_y \cdot \sum_{r=1}^{y} \frac{d_{\max}}{x-r+1} + d_{\max} \cdot b_y \cdot \sum_{r=y+1}^{x} \frac{1}{x-r+1} \\
&\leq d_{\max} \cdot b_y \cdot \sum_{r=1}^{x} \frac{1}{x-r+1} \leq d_{\max} \cdot H_{d_{\max}} \cdot b_y
\end{aligned}$$

Let S_{a_k} be the set in $\mathcal{C}_{grd}$ that is assigned with bid b_y. Then we can "charge" S_{a_k} for $d_{\max} \cdot H_{d_{\max}}$ times b_y. Therefore, the value $V(\mathcal{C}_{opt})$ of $\mathcal{C}_{opt}$ is no more than $d_{\max} \cdot H_{d_{\max}}$ times the value $V(\mathcal{C}_{grd})$ of $\mathcal{C}_{grd}$.

This finishes the proof. □

Theorem 6 *For set cover games with selfish sets and elements, a strategyproof mechanism M_S to sets and a cross-monotone payment sharing scheme ξ imply* *that in mechanism M_E each set S_j cannot improve its utility by lying upward its cost.*

PROOF. At current moment, for the sake of simplicity, we assume that any set does not change its declared cost. Thus, the payment to each set will not change. Since the payment sharing scheme is cross-monotone, any group of elements cannot change their bids to increase the utility of some elements without decreasing the utility of some other elements in this group.

We then show that each set indeed does not have incentives to lie about its cost *upward*. Notice that since the payment scheme to each set is strategyproof by assumption, any set cannot lie about its cost to increase its payment when it is selected to cover some elements. In other words, its utility cannot be increased as long as it is selected in the final outcome. Consequently, the only scenario that a selfish set S_j may increase its utility is that (1) it is selected to cover some elements initially when it declares its true cost c_j and each element e_i is assumed to have an infinitely large bid $b_{i,j}$; [4] (2) it is not selected if it declares its true cost c_j because the corresponding charges to some elements are not affordable, *i.e.*, larger than the bids of elements; [5] and (3) it will be selected if it declares a false cost $\bar{c}_j$, *i.e.*, the corresponding charges will be no more than the bids of elements. We will show that this is impossible if $\bar{c}_j > c_j$.

Assume that the declared costs of all sets other than S_j are fixed and the declared bids of all elements are fixed. Let p_j be the payment to set S_j. Since the payment scheme to sets are strategyproof, p_j is independent of its declared cost. If the set S_j lies its cost upward as $\bar{c}_j > c_j$, then the set of elements that will be covered by S_j is only a subset of the elements previously covered by S_j. Since the payment p_j to S_j is only shared among elements $E(S_j, c|^j\bar{c}_j, T, M_S)$, the cross-monotonicity of the payment-sharing method ξ implies that the shared payment of each element e_i in $E(S_j, c|^j\bar{c}_j, T, M_S)$ is *not* smaller than its shared payment if S_j did not lie its cost. Remember that the set S_j is not affordable when it reports its cost c_j, *i.e.*, the total amount of bids of elements in $E(S_j, c, T, M_S)$ for their copy covered by S_j is less than p_j. Consequently, the set S_j is still not affordable when it reports its cost as $\bar{c}_j > c_j$. This finishes the proof. ◻

[4] We need this condition because otherwise its payment will always be no more than its cost from the strategyproof property. Notice that when it is not selected its utility is 0.

[5] This condition makes sure that it does have incentives to lie. Otherwise its payment will be fixed when it is selected.