Destroy to Save

Geoffroy de Clippel[1] Victor Naroditskiy[2] Maria Polukarov[2]
Amy Greenwald[3] Nicholas R. Jennings[2]

Abstract

We study the problem of allocating m identical items among $n > m$ agents, where the items are jointly owned by the agents rather than by an auctioneer. Each agent desires exactly one item and has a private value for consuming it. We assume quasi–linear utilities and focus on dominant–strategy implementation. The key issue is that in the absence of an auctioneer who can absorb payments collected from the agents, the payments must be burnt to support dominant–strategy implementation.

Existing mechanisms for this setting modify the classic VCG mechanism by redistributing as much of the payments as possible back to the agents while still satisfying incentive constraints. This approach guarantees allocative efficiency, but in some cases a large percentage of social welfare is lost. In this paper, we provide a mechanism that is not allocatively efficient but is instead guaranteed to achieve at least 80% of the social welfare as $n \to \infty$. Moreover, in the extreme case of $m = n - 1$ where VCG–based mechanisms provide zero welfare, the percentage of social welfare maintained by our mechanism approaches 100% as $n \to \infty$.

1. Introduction

Suppose six city–dwelling roommates jointly own a car that seats five people. They decide to take a trip to the countryside. While they all would like to

[1]Department of Economics, Brown University, Providence, RI, USA. Financial support from the National Science Foundation (grant SES-0851210) and the Deutsche Bank through the Institute for Advanced Study is gratefully acknowledged.

[2]School of Electronics and Computer Science, University of Southampton, United Kingdom. The authors are supported by the ALADDIN project, jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1).

[3]Department of Computer Science, Brown University, Providence, RI, USA

go, there is not room for all of them in the car. Some really need the fresh country air while others would not really mind staying home. The roommates do not necessarily know one another's desires, but each of them knows her own true value of getting out of the city. How should they decide who gets to go?

We study a class of resource allocation problems, in which n agents jointly own $m < n$ identical items that they wish to distribute among themselves. We assume each agent wants exactly one item, and has a *private value* for that item. As further examples, consider the allocation of free tickets for a sporting event among club members, or seats on an overbooked airplane.

We assume that the agents can make monetary payments, and further that they have *quasi–linear* utilities. These assumptions allow to provide incentives for the agents to reveal their private values truthfully. In the presence of a participant possessing no private information (e.g., an auctioneer), monetary payments can be absorbed by him, thus achieving budget balance. In our setting however, all agents have private information, and any collected payments need to be burnt in order to maintain truthful reporting. Burning money is undesirable as it decreases social welfare. Therefore, it is important to design mechanisms that ensure a high level of social welfare while maintaining the incentives.

In this work, we focus on *strategy–proof* mechanisms, which require that it be a dominant strategy for each agent to report her value truthfully. This requirement is less permissive, but more robust than Bayesian implementation. In particular, agents are more likely to play a dominant strategy than a strategy that is optimal only when other agents play their part of the truthful equilibrium. Perhaps even more importantly, dominant–strategy implementation does not require any assumptions about the distribution of the agents' values (or their beliefs about those values), nor their attitude towards risk.

As for measuring the appeal of various strategy–proof mechanisms, we choose the strictest metric: worst–case performance. More specifically, if one fixes the agents' profile of values, one can compute the ratio of the social welfare realized by the mechanism over the maximal social welfare that could be achieved, should these values be known. Since these values are not known, nor is their probabilistic distribution, the appeal of a strategy–proof mechanism is measured by the minimum of this ratio over all possible value profiles. We call this ratio a *social welfare ratio*[4], and we use it to determine a mechanism's worst–case (i.e., guaranteed) level of social welfare: reaching a level $\alpha \in [0, 1]$ means that a mechanism realizes at least a proportion α of the maximal social welfare for *every* possible profile of the agents' values.

In addition to strategy–proofness, we also impose the following natural constraints: 1) *feasibility*—no more than m items can be allocated, and monetary deficits are not allowed (i.e., no external subsidy), 2) *individual rationality*—each agent's total utility is nonnegative, and 3) *anonymity*—the allocation and payment decision applied to each agent does not depend on her identity. The question we are interested in can now be stated formally:

> *Find a mechanism that maximizes the worst–case social welfare ratio among all those that are strategy–proof, feasible, individually rational, and anonymous.*

Recently, two sets of authors (Moulin (2009) and Guo and Conitzer (2009)) solved the above question under the additional constraint that the items be allocated to the m agents who value them most. Their solution characterized the best mechanism within the class of VCG[5] mechanisms, which has received special attention in the economics literature because members of this class admit a simple functional form (cf. characterization by Green and Laffont (1979)). A VCG mechanism guarantees an efficient allocation of all m items, but not necessarily a good level of "overall" efficiency (as measured for instance by the worst–case social welfare ratio), because allocative efficiency may come at the cost of "burning" quite a bit of money to meet the incentive constraints, whereas it may be better in terms of overall efficiency to destroy some items in order to burn less money. Indeed it is. It is not difficult to check that it is impossible to guarantee a strictly positive worst–case social welfare ratio using a VCG mechanism when $m = n - 1$. On the other hand, applying the best VCG mechanism after destroying even one item would secure a strictly positive ratio.

Still, applying a VCG mechanism after destroying some fixed number of items is not the best way to optimize overall efficiency. We show that *contingent destruction* mechanisms, which make destruction decisions based on the

[4]The same measure is used in (Moulin, 2009; Guo and Conitzer, 2009).

[5]VCG stands for Vickrey, Clarke, and Groves, who independently defined and studied some of these mechanisms in various contexts.

values agents report, perform better. In fact, all strategy–proof mechanisms rely on *contingent payments* as well as contingent destruction: an agent receives an item if and only if her reported value is larger than a threshold value, which in general depends on other agents' reports. We refer to such mechanisms as *threshold mechanisms*.

The question of finding the best threshold mechanism is more complicated than the question of finding the best VCG mechanism. The VCG mechanism's allocation function is constant once the agents' values are ordered (the agents with the m highest values receive the items). In this restricted context, Guo and Conitzer (2009) and Moulin (2009) calculate the optimal payment function. In contrast, allocation functions in the broader class of threshold mechanisms are not constant.

We discuss related literature before describing our results. Enhancing VCG mechanisms with payment redistribution has been studied in various settings. Bailey (1997) proposes a way to redistribute some of the VCG tax in a public good domain. Cavallo (2006) designs a redistribution mechanism for single–item allocation problems, and provides a characterization of redistribution mechanisms for more general allocation problems. As already mentioned, Guo and Conitzer (2009) and Moulin (2009) independently discover the optimal VCG redistribution mechanism for the allocation domain studied here. Guo and Conitzer (2010b) derive a linear redistribution VCG mechanism to maximize the expected social welfare when the distribution of agents' values is known. Porter et al. (2004) study the problem of allocating undesirable goods (e.g., tasks) to agents in a fair manner. In the model where a single item (task) needs to be allocated, the mechanisms by Porter, Bailey, and Cavallo coincide.[6] In other contexts, Hartline and Roughgarden (2008) study welfare–maximizing mechanisms in settings where positive transfers to the agents are not allowed. The tradeoff between efficiency and budget balance in cost sharing scenarios is discussed by Moulin and Shenker (2001).

Most related to our work is the work by Guo and Conitzer (2008). Starting from the observation that destroying items might save enough money to be socially beneficial, they limit attention to allocation rules where the number of items destroyed is independent of the agents' reports. In order for the social benefit to be more significant, they allow for that number to be deter-

[6] A detailed discussion of relationship between these mechanisms appears in (Guo and Conitzer, 2009).

mined randomly.[7] Instead, we observe that significant gains can be achieved via deterministic mechanisms provided one uses *contingent* destruction rules. Furthermore, Guo an Conitzer require feasibility only in expectation, and need to assume that the agents are risk neutral. Our deterministic analysis allows to dispense with these assumptions. If one is willing to use lotteries, then it may be of interest to combine the insights from our two papers, making Guo and Conitzer's random variables depend on reported values.

Our approach is designed to achieve the right balance between tractability, and showing that one can obtain a significant improvement in overall welfare if one does not rely on the technical convenience of VCG mechanisms. Even a very minor departure from VCG—destroying at most one item—allows for drastic savings in problem instances with many agents and items. Specifically, this paper presents a mechanism that guarantees an asymptotic ratio of at least 0.8 as $n \to \infty$. Importantly, the ratio guarantee improves as the number of items increases. Perhaps most striking is the case where $m = n - 1$. As already pointed out, VCG mechanisms do not provide any strictly positive ratio in this case. Further, applying the best VCG mechanism after destroying a fixed number of items does not guarantee a ratio larger than 1/2 (see numerical computations in Guo and Conitzer (2008)). In contrast, our method of *contingent destruction* guarantees the ratio of $1 - \frac{2}{n^2-n}$, which rapidly approaches 1 as n increases. Finally, an additional advantage of our mechanism is that it has a much simpler analytical form than the optimal VCG mechanism.

This paper unfolds as follows. Section 2 formally states the problem we are studying. A computational method of searching for an optimal mechanism in a restricted setting is proposed in Section 3. Numerical results for this setting are presented in Section 4. Based on the numerical results, we discern the analytical form of a general mechanism in Section 5.

[7] If one is ready to use lotteries, then observe that other natural mechanisms come to mind. Faltings (2005), for instance, proposes a mechanism that picks an agent at random, and makes him the recipient of the VCG payments. This mechanism applies to domains more general than our allocation domain and achieves budget balance. However, if one applies this mechanism to our allocation domain, one sees that the resulting allocation is not efficient (unless the chosen recipient happens to value the item less than those who are allocated an item).

2. Definitions

An *allocation problem* is a triple $\langle n, m, v \rangle$, where n is the number of agents, $m < n$ is the number of (identical) items available to allocate, and $v \in \mathbb{R}_+^n$ represents the agents' satisfaction from consuming one item (agents do not care for consuming multiple units). We restrict attention to value profiles v such that $v_1 \geq v_2 \geq \ldots \geq v_n \geq 0$. This is without loss of generality since our problem involves only anonymous mechanisms. Monetary compensations are possible, and utilities are quasi–linear. The space of possible value profiles is then $V = \{v \in \mathbb{R}_+^n \mid v_1 \geq v_2 \geq \ldots \geq v_n \geq 0\}$. An *allocation* is a pair $(\mathbf{f}, \mathbf{t}) \in \{0,1\}^n \times \mathbb{R}^n$, where $\mathbf{f}_i = 1$ if and only if agent i gets one item, and $\mathbf{t}_i$ represents the amount of money that agent i receives (this number can be negative, of course, in which case agent i pays that amount). Hence, the total utility of agent i when implementing the allocation $(\mathbf{f}, \mathbf{t})$ is $\mathbf{f}_i v_i + \mathbf{t}_i$, if her value for an item is v_i. A *mechanism* is a pair of functions $f : \mathbb{R}_+^n \to \{0,1\}^n$ and $t : \mathbb{R}_+^n \to \mathbb{R}^n$. Thus, it determines an allocation for each possible report from the agents regarding their value for an item. We slightly abuse the notation and define $f_i(v) = \left(f(v)\right)_i$ and $t_i(v) = \left(t(v)\right)_i$. The vector $v_{-i} \in \mathbb{R}^{n-1}$ denotes values of the agents other than agent i and the vector v can be written as (v_i, v_{-i}). We focus on mechanisms that satisfy the following constraints:

- *Feasibility*: no more than m items should be allocated, and the sum of payments to the agents should be less than or equal to zero, for all value profiles v.

$$\sum_{i=1}^{n} f_i(v) \leq m \text{ and } \sum_{i=1}^{n} t_i(v) \leq 0 \quad \forall v \in V$$

- *Strategy–proofness*: It is a dominant strategy for each agent to report her value truthfully.

$$f_i(v_i, v_{-i})v_i + t_i(v_i, v_{-i}) \geq f_i(v_i', v_{-i})v_i + t_i(v_i', v_{-i}) \quad \forall v \in V, i, v_i'$$

- *Individual Rationality*: It is in each agent's interest to participate in the mechanism, for all value profiles v.

$$f_i(v)v_i + t_i(v) \geq 0 \quad \forall v \in V, i$$

We now define the index that we use to measure the overall efficiency of a mechanism (f, t) that is implemented truthfully (an equivalent index was used in (Moulin, 2009; Guo and Conitzer, 2009, 2008)). If the true value profile is v, then the social welfare realized by the mechanism is equal to $\sum_{i=1}^{n}[v_i f_i(v) + t_i(v)]$. This number is less interesting than knowing how far it is from the first–best solution, i.e. the maximal welfare one could achieve if the agents' values were known. In order to have an index that is unit–free (i.e. homogenous of degree zero), it is natural to consider a ratio. Finally, since the agents' values are not known, nor their probabilistic distribution, it is natural to consider the worst–case index. To summarize, the index that we use to measure the performance of a mechanism (f, t) that is truthfully implemented is given by the following ratio:

$$\min_{v \in V} \frac{\sum_{i=1}^{n}[f_i(v)v_i + t_i(v)]}{\max_{f' \in \mathcal{F}(m)} \sum_{i=1}^{n} f_i' v_i},$$

where $\mathcal{F}(m) = \{f' \in \{0,1\}^n | \sum_{i=1}^{n} f_i' \leq m\}$. Finding a mechanism whose ratio is α means that a proportion α of the maximal social welfare is achieved, *independently* of what the true values are. Following the convention $v_1 \geq v_2 \geq \ldots \geq v_n \geq 0$, the denominator becomes $\sum_{i=1}^{m} v_i$ and we write the ratio as

$$\min_{v \in V} \frac{\sum_{i=1}^{n}[f_i(v)v_i + t_i(v)]}{\sum_{i=1}^{m} v_i}$$

The formal content of the question stated in the introduction can thus be summarized by the following optimization problem:

$$\max_{(f,t)} \quad \min_{v \in V} \frac{\sum_{i=1}^{n}[f_i(v)v_i + t_i(v)]}{\sum_{i=1}^{m} v_i} \tag{1}$$

$$\sum_{i=1}^{n} f_i(v) \leq m \quad \forall v \in V \tag{2}$$

$$\sum_{i=1}^{n} t_i(v) \leq 0 \quad \forall v \in V \tag{3}$$

$$f_i(v_i, v_{-i})v_i + t_i(v_i, v_{-i}) \geq f_i(v_i', v_{-i})v_i + t_i(v_i', v_{-i}) \quad \forall v \in V, i, v_i' \tag{4}$$

$$f_i(v)v_i + t_i(v) \geq 0 \quad \forall v \in V, i \tag{5}$$

There is a full characterization of dominant–strategy implementable mechanisms in settings where agent's private information is a single number $v_i \in \mathbb{R}$.

Theorem 1 (e.g., see Nisan et al. (2007) p. 229). *A mechanism* (f, t) *is implementable in dominant strategies if and only if for each agent* i*:* (i) f_i *is monotone*[8] *in* v_i*;* (ii) $t_i(v) = h(v_{-i}) - \tau(v_{-i})$ *if* $f_i(v) = 1$ *(i.e., i is allocated) and* $t_i(v) = h(v_{-i})$ *otherwise, where* $\tau(v_{-i}) = \sup_{v_i | f_i(v_i, v_{-i})=0} v_i$ *defines the* threshold.[9,10]

These mechanisms are easy to interpret. Each agent faces a personalized *price* (the threshold τ) that is determined by the reports of the other agents. She gets the good if and only if her reported value is larger than this price, and must pay it in exchange. The collected money can be redistributed to some extent to the agents via the *rebate* function h. The VCG mechanisms form a special case, where i's price is the m^{th} largest component of v_{-i}.

The allocation function f is determined by the threshold function, while the payment function t is determined by the threshold and rebate functions. Thus, we can restate the optimization problem (1)–(5) using these functions.

$$
\boxed{
\begin{array}{l}
\max\limits_{(h,\tau)} \quad \min\limits_{v \in V} \dfrac{\sum_{i | v_i \geq \tau(v_{-i})} (v_i - \tau(v_{-i})) + \sum_{i=1}^{n} h(v_{-i})}{\sum_{i=1}^{m} v_i} \\
\#\{i | v_i \geq \tau(v_{-i})\} \leq m \quad \forall v \in V \\
\displaystyle\sum_{i=1}^{n} h(v_{-i}) \leq \sum_{i | v_i \geq \tau(v_{-i})} \tau(v_{-i}) \quad \forall v \in V \\
h(v_{-i}) \geq 0 \quad \forall v \in V, i
\end{array}
}
$$

The first constraint is the feasibility constraint with respect to the items being allocated[11], while the second constraint is the feasibility constraint

[8] Monotonicity of f_i in v_i means that if an agent is allocated when she reports v_i, she is also allocated when she reports $v'_i \geq v_i$.

[9] An agent i is allocated if and only if her report is above the threshold $\tau(v_{-i})$.

[10] As our focus is on anonymous mechanisms, the theorem was adapted to payment functions h (rather than h_i) that do not vary from agent to agent.

[11] For notational convenience from now on we restrict our attention to profiles v where all components are distinct. This restriction is introduced without loss of generality as we

with respect to money (the sum of all rebates should be no more than the amount collected from the agents that get an item). The third constraint is the individual rationality constraint (remember that an agent's value must be larger than the threshold when she gets an item, and so this constraint is trivially satisfied for her as well). The strategy–proofness constraint is no longer needed as all threshold mechanisms are strategy–proof.

We now propose a last formulation of our optimization problem. We remove the minimization over v by introducing a variable $r \in \mathbb{R}$ denoting the best ratio that holds for any profile of values. The resulting optimization program appears in Figure 1.

$$\max_{(h,\tau),r} r \tag{6}$$

$$\sum_{i|v_i \geq \tau(v_{-i})} (v_i - \tau(v_{-i})) + \sum_{i=1}^{n} h(v_{-i}) \geq r \sum_{i=1}^{m} v_i \quad \forall v \in V \tag{7}$$

$$\#\{i|v_i \geq \tau(v_{-i})\} \leq m \quad \forall v \in V \tag{8}$$

$$\sum_{i=1}^{n} h(v_{-i}) \leq \sum_{i|v_i \geq \tau(v_{-i})} \tau(v_{-i}) \quad \forall v \in V \tag{9}$$

$$h(v_{-i}) \geq 0 \quad \forall v \in V, i \tag{10}$$

Figure 1: The problem of finding an optimal mechanism stated as an optimization problem.

A solution to the mathematical program above is the optimal mechanism for the allocation domain. However, the program is hard to solve for different reasons. Firstly, maximization is over arbitrary functions h and τ, and there is little hope in optimizing over the space of arbitrary functions. Secondly,

---

can extend the mechanism to all value profiles by using uniform lotteries to break ties, as is usually done in papers on auctions. Suppose for instance that agent i should receive an item, and that more than m other agents have the same value as i. Anonymity would then come in conflict with feasibility. A uniform lottery can then be used to determine which subset of agents receives an item, among all those that have the same value. Even so, the way agents react to risk is irrelevant because all the outcomes of the lottery are equivalent in terms of utility. Specifically, the lottery is between receiving an item worth v_i at a price p_i and receiving a rebate h_i such that $h_i = v_i - p_i$.

the program has an infinite number of constraints as the set of possible value profiles $v \in V$ is infinite. We address these problems in the next section where we make assumptions about the form of the functions h and τ, show that it is sufficient to consider a finite number of constraints, and solve the resulting problem computationally.

3. A Simpler Problem

There are infinitely many constraints in our optimization problem, since they are indexed by the value profiles v. Our technique for dealing with this difficulty comes from an observation that linear constraints are satisfied throughout a convex polytope if and only if they are satisfied at its extreme points (see Observation 1). To apply the observation we need to identify regions where the constrains are linear. In order to do this, we restrict attention to the threshold and rebate functions for which we can partition the space of value profiles into such regions. Letting $w \in \mathbb{R}^{n-1}$ refer to a profile of agents' values with one agent excluded we make two assumptions.

Assumption 1. *The threshold function is of the form $\tau(w) = \max(kw_p, w_m)$, with $k \in [0, 1]$ and $p \in \{1, 2, \ldots, (m-1)\}$.*

Adding w_m–component inside the max operator guarantees that no more than m items are allocated, as required by the feasibility constraint. The p parameter determines how many items are guaranteed to be allocated, independently of the agents' reports. The parameter k controls how large the values of agents $p+1$ through m must be for them to be allocated. Taking $k = 0$ brings us back to VCG mechanisms, while setting of $k = 1$ means that items $p+1, \ldots, m$ are always destroyed.

Similarly, we focus on rebate functions that are linear on convex subsets of the set of value profiles. Here too we face a trade–off between tractability and generality. A finer partition allows more flexibility in the rebates, but also increases the number of constraints in the linear program to be solved. The simplest choice would be to choose functions that are linear on the whole cone characterized by $v_1 \geq v_2 \geq \ldots \geq v_n \geq 0$. Focusing on VCG mechanisms, Moulin (2009) and Guo and Conitzer (2009) proved that the optimal rebate function actually falls in that class. This is not the case anymore when considering non–VCG mechanisms, as evidenced by our numerical results. Another simple choice would be to take rebate functions that are linear on the regions where the number of items being allocated is constant. This choice is not permitted, though, because these regions depend on the values of all the agents, while the rebate function can depend only on the values of the agents different from the one receiving the rebate. We decided to choose the closest match, imposing a condition that mimics the definition of these regions while using only the right values.

Assumption 2. *The rebate function h is linear in values on two regions:*

$$h(w; a, b) = \begin{cases} aw & \textit{if } kw_p \geq w_m \\ bw & \textit{otherwise} \end{cases}$$

where $a, b \in \mathbb{R}^{n-1}$.

The approach we follow to optimize over the class of mechanisms satisfying these assumptions is summarized in Figure 2. We consider threshold functions satisfying Assumption 1 with k coming from a finite set of constants between 0 and 1 and p taking any value between 1 and $m - 1$. For each threshold function specified by k and p, we compute the rebate function that guarantees the highest social welfare among rebate functions satisfying Assumption 2. We then select the threshold and the corresponding rebate functions that achieved the highest welfare among the ones we considered. The key step is computation of optimal rebates, which is the focus of the rest of this section.

For the allocation problem with m items and n agents:

1. let K denote a finite set of values for the parameter k
2. for each threshold function τ_{kp} given by $k \in K$ and $p \in \{1, 2, \ldots, m-1\}$
   - find the optimal rebate function h_{kp} that satisfies Assumption 2
3. choose the mechanism (τ_{kp}, h_{kp}) with the highest welfare

Figure 2: Computational search for a welfare-maximizing mechanism.

To tackle computation of optimal rebates, we first characterize the regions where the rebate function is linear and the number of allocated items is

$i \in \{1 \dots p\}$

$$f_i = 1,\ t_i(v) = -\max(kv_{p+1}, v_{m+1}) + \begin{cases} av_{-i} & \text{if } kv_{p+1} \geq v_{m+1} \\ bv_{-i} & \text{otherwise} \end{cases}$$

$i \in \{(p+1) \dots m\}$

$$\text{if } v_i \geq kv_p\text{: } f_i = 1,\ t_i(v) = -\max(kv_p, v_{m+1}) + \begin{cases} av_{-i} & \text{if } kv_p \geq v_{m+1} \\ bv_{-i} & \text{otherwise} \end{cases}$$

$$\text{otherwise: } f_i = 0,\ t_i = \begin{cases} av_{-i} & \text{if } kv_p \geq v_{m+1} \\ bv_{-i} & \text{otherwise} \end{cases}$$

$i \in \{(m+1) \dots n\}$

$$f_i = 0,\ t_i = \begin{cases} av_{-i} & \text{if } kv_p \geq v_m \\ bv_{-i} & \text{otherwise} \end{cases}$$

Figure 3: Mechanism satisfying Assumptions 1 and 2.

constant. By the definition of the threshold function $\tau = \max(kw_p, w_m)$, there are $m - p + 1$ possible allocations (the first p agents get the items, the first $p + 1$ agents get the items, ..., the first m agents get the items) determined by the position of kv_p among $v_p \dots v_m$. The rebate function $h(w)$ is resolved to one of the two linear functions (aw or bw) when the position of kv_p relative to v_m and v_{m+1} and the position of kv_{p+1} relative to v_{m+1} are determined (see Figure 3). Below, we partition the space of values into regions where allocation is determined and payment is resolved to either aw or bw. A region $V_{j,\cdot}$ positions kv_p between v_j and v_{j+1} determining allocation and payment for some of the agents. Payment for the rest of the agents is determined by the second subscript which specifies whether kv_{p+1} is above ($>$) or below ($<$) v_{m+1}.

$$\forall j \in \{p \dots m+1\}$$
$$V_{j,>} = \{v \in V | v_1 \geq \dots \geq v_p \geq \dots \geq v_j \geq kv_p \geq v_{j+1} \geq \dots \geq v_m \geq \dots \geq v_n \text{ AND } kv_{p+1} \geq v_{m+1}\}$$
$$V_{j,<} = \{v \in V | v_1 \geq \dots \geq v_p \geq \dots \geq v_j \geq kv_p \geq v_{j+1} \geq \dots \geq v_m \geq \dots \geq v_n \text{ AND } kv_{p+1} \leq v_{m+1}\}$$

The collection of regions above partitions the space $\{v \in V|\ v_1 \geq v_2 \geq \dots \geq v_n \geq 0\}$. We group constraints by region and restate the optimization problem (see Figure 4). Notice that on each region the constraints are of the form $dv \geq 0$ for some $d \in \mathbb{R}^n$, which means that they are satisfied at λv ($\forall \lambda > 0$) as soon as they are satisfied at v. Hence we can assume without loss of generality that $v_1 = 1$ and focus on polytopes of vectors $(v_2, \ldots, v_n) \in \mathbb{R}^{n-1}$. A polytope $V_{j,>}$ (symmetrically $V_{j,<}$) is given by $(n+2)$ inequalities:

$$1 \geq v_2,\ v_2 \geq v_3,\ \cdots,\ v_j \geq kv_p,\ kv_p \geq v_{j+1},\ \cdots,\ v_{n-1} \geq v_n,\ v_n \geq 0$$
$$kv_{p+1} \geq v_{m+1}$$

The following observation tells us that it is enough to restrict attention to the extreme points of each of these polytopes.

Observation 1. *For any coefficients $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, a linear constraint $av \geq b$ holds at all $v \in P$ of a polytope $P \subset \mathbb{R}^n$ if and only if it holds at the points $v \in$ ExtremePoints(P), where ExtremePoints(P) denotes the set of extreme points of polytope P.*

Thus, making sure the constraints hold at the extreme points of $V_{j,\cdot}$ guarantees that the constraints hold everywhere on $V_{j,\cdot}$. Now the linear program in Figure 4 can be solved by enforcing constraints only at the extreme points of each polytope $V_{j,\cdot}$.

Example As an example consider the allocation problem with $n = 3$, $m = 2$ and the threshold function specified by $k = .5$ and $p = 1$: $\tau(w) = \max(.5w_1, w_2)$. The threshold function for agent 1 is $\max(.5v_2, v_3) < v_1$ (see Figure 3). So agent 1 is always allocated an item. The threshold for agent 2 is $\max(.5v_1, v_3)$. Agent 2 is allocated an item only when $v_2 \geq .5v_1$. Agent 3 is never allocated an item as the threshold for agent 3 is $\max(.5v_1, v_2) > v_3$.

The rebate function is linear when in addition to allocation, the position of $.5v_1$ and $.5v_2$ relative to v_3 is determined. Taking $v_1 = 1$ we can represent this on a 2-dimensional graph (Figure 5). The space is divided into 5 regions, with each region having a linear rebate function and a fixed allocation. To make sure the constraints hold for all $\{v \in V |\ v_1 \geq v_2 \geq v_3\}$, we just need to enforce each region's constraints at its extreme points. For example, the extreme points of the right bottom region after adding $v_1 = 1$ as the first component are (1,.5,0), (1,.5,.25), (1,1,.5), (1,1,0).

$$
\begin{aligned}
&\max_{a,b\in\mathbb{R}^{n-1},r\in\mathbb{R}} r\\
&\forall j \in \{p \dots m+1\},\ v \in V_{j,>}\\
&\qquad \sum_{i=1}^{j} v_i - \sum_{i=1}^{p} kv_{p+1} - \sum_{i=p+1}^{j} kv_p + \sum_{i=1}^{n} h(v_{-i};a,b) \geq r \sum_{i=1}^{m} v_i\\
&\qquad \sum_{i=1}^{n} h(v_{-i};a,b) \leq \sum_{i=1}^{p} kv_{p+1} + \sum_{i=p+1}^{j} kv_p\\
&\qquad h(v_{-i};a,b) \geq 0 \quad \forall i\\
&\forall j \in \{p \dots m\},\ v \in V_{j,<}\\
&\qquad \sum_{i=1}^{j} v_i - \sum_{i=1}^{p} v_{m+1} - \sum_{i=p+1}^{j} kv_p + \sum_{i=1}^{n} h(v_{-i};a,b) \geq r \sum_{i=1}^{m} v_i\\
&\qquad \sum_{i=1}^{n} h(v_{-i};a,b) \leq \sum_{i=1}^{p} v_{m+1} + \sum_{i=p+1}^{j} kv_p\\
&\qquad h(v_{-i};a,b) \geq 0 \quad \forall i
\end{aligned}
$$

Figure 4: Linear program with constraints grouped by regions $V_{j,>}$ and $V_{j,<}$.

4. Numerical Results

We find mechanisms for different values of n and m using the computational procedure described in Figure 2. The class of threshold functions we consider is given by all pairs (k, p) where k takes values in $\{0, .025, .05, \dots, .975\}$ and p in $\{1, 2, \dots, m-1\}$. We used CPLEX 11.2.0 as a linear program solver.

Figure 6 illustrates the results we generate for each setting of n, m, and p. The value for the parameter k is varied along the horizontal axis. For each value of k, the corresponding threshold function is $\tau = \max(kw_p, w_m)$, and we can solve the linear program in Figure 4 to find an optimal rebate function h_{kp}. The social welfare ratio of each mechanism (τ_{kp}, h_{kp}) is plotted for the corresponding k value. We refer to the resulting graph as a *performance curve*. We scan the values of k for the one that has the highest ratio. Figure 6 illustrates a typical graph one can construct given n, m, and p as input. This graph is for the allocation problem with $n = 6$ agents and $m = 5$ items when considering thresholds that destroy at most one item ($p = 4$). The best ratio

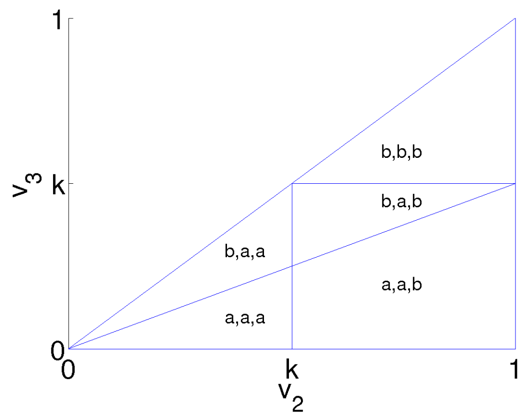


Figure 5: ($v_1 = 1$) Regions where the number of allocated items remains constant and the rebate function is linear for 3 agents and 2 items. Each region is labeled with the coefficients used in the rebate function for each agent, e.g. (b,a,a) means that the rebate functions for agents 1,2, and 3 are bv_{-1}, av_{-2}, and av_{-3} respectively. One item is allocated to the left of the vertical line $v_2 = k$ and two items to the right.

is achieved at $k = .175$. Notice that the shape of the curve suggests there is only one peak. We try other values of k around .175 to find the peak at $k = \frac{1}{6}$. In all of our results we noticed that the performance curve as a function of k is single-peaked. This observation makes it easy to search for the optimal k computationally.

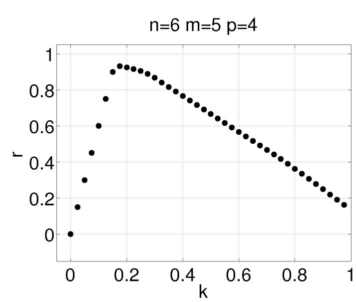


Figure 6: Performance curve for the allocation problem with 6 agents and 5 items when at least 4 items are allocated ($p = 4$).

For any fixed values of n and m we found that a mechanism with p set to $m - 1$ achieves the highest ratio. This setting of p means that at most one

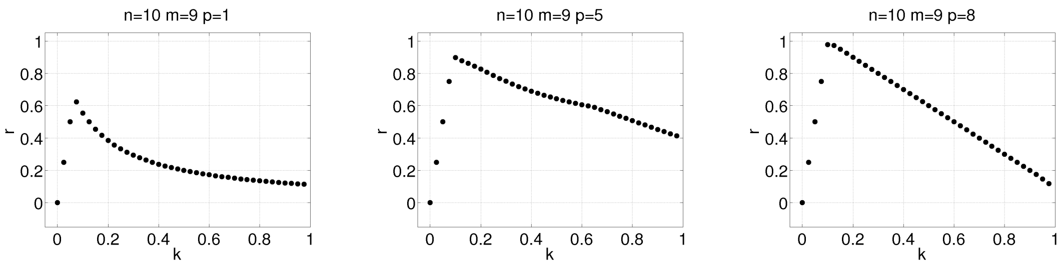


Figure 7: Performance curves for the allocation problem with 10 agents and 9 agents for different values of p

item is destroyed. This result is consistent with the one obtained by Guo and Conitzer Guo and Conitzer (2008) for randomized VCG mechanisms. They find that the best mechanism randomizes between destroying one item and not destroying any items. The performance curves for different values of p are shown in Figure 7. Notice that the highest ratio is obtained on the graph for $p = m - 1 = 8$ (at $k = .1$).

The mechanisms we find provide the most improvement when the number of items is close to the number of agents. Our ratio gets closer to the VCG ratio as the number of items gets smaller and approximately around $m = \frac{n}{2}$ the ratios and the mechanisms coincide. Figure 8 shows this trend for 10 agents and varying number of items. Also plotted are the ratios achieved by the best VCG mechanism as well as the ratio achieved by the mechanism that first destroys a fixed number of items and then applies an optimal VCG mechanism (see deterministic burning mechanism in Guo and Conitzer (2008)). All mechanisms coincide when the number of items is 4 or fewer.

5. Analytical Results

After analyzing mechanisms obtained numerically for various values of n and m, we noticed a pattern and derived a simple mechanism parameterized by n and m. We show that this mechanism, termed *SimpleDestroy* (SD), achieves the ratio of at least $1 - \frac{\binom{n-m+1}{2}}{\binom{n}{2}}$, which for any $m \geq .555n$ is at least 0.8 asymptotically; moreover, when m is close to n, the ratio approaches 1.

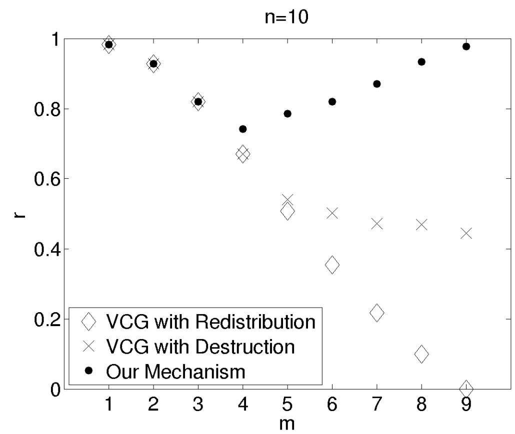


Figure 8: Performance of various mechanisms as a function of the number of items.

5.1. The SimpleDestroy mechanism

The SD mechanism is defined in Figure 9. Note that it satisfies Assumptions 1 and 2 and is equivalently determined by the parameters $p = m - 1$, $k = \frac{n-m}{n}$ and the coefficients $a = (0, \dots, 0)$, $b = (0, \dots, 0, -k, 1)$.

Intuitively, the mechanism is specified in two cases based on the amount of payment collected from the allocated agents before rebates. If the collected amount is small relative to the social welfare, then there are no rebates. This case occurs when the threshold is a fraction of the value of the last agent who is guaranteed to be allocated: $\tau(w) = kw_{m-1}$. On the other hand, when the threshold is above kw_{m-1}, the mechanism makes sure enough redistribution occurs if a significant portion of welfare is collected.

$$\tau(w) = \max\{kw_{m-1}\,;\ w_m\}$$

$$h(w) = \begin{cases} 0 & \text{if } \ \tau(w) = kw_{m-1} \\ w_m - kw_{m-1} & \text{if } \ \tau(w) = w_m \end{cases}$$

$$\text{where } k = \frac{n-m}{n}\,.$$

Figure 9: The SimpleDestroy Mechanism.

The next theorem proves that the SD mechanism is valid and shows good performance.

Theorem 2. *The SimpleDestroy mechanism is individually rational, subsidy–free, and achieves the ratio of at least* $1 - \frac{\binom{n-m+1}{2}}{\binom{n}{2}}$.

Proof First note, that the rebate function is nonnegative, as its value is different from 0 only if $w_m > \frac{n-m}{n} w_{m-1}$, in which case $h(w) = w_m - \frac{n-m}{n} w_{m-1} > 0$. So, individual rationality always holds.

The threshold function $\tau(w) = \max\left\{\frac{n-m}{n} w_{m-1}\,;\; w_m\right\}$ allows two allocations: agents $1 \ldots (m-1)$ are allocated when $v_m < \frac{n-m}{n} v_{m-1}$ and agents $1 \ldots m$ are allocated otherwise. We prove for each case separately.

(I) Assume $v_m \geq \frac{n-m}{n} v_{m-1}$, that is, m items are allocated. This also defines the threshold and the rebates to agents $i = m+1, \ldots, n$ for which $(v_{-i})_m = v_m$ and $(v_{-i})_{m-1} = v_{m-1}$, and hence $\tau(v_{-i}) = v_m$ and $h(v_{-i}) = v_m - \frac{n-m}{n} v_{m-1}$. For agents $i = 1, \ldots, m$, there are three possibilities as follows.

(Ia) If $v_{m+1} \geq \frac{n-m}{n} v_{m-1}$, then also $v_{m+1} \geq \frac{n-m}{n} v_m$, and thus for all $i = 1, \ldots, m$ we have $\tau(v_{-i}) = v_{m+1}$. The rebates are given by $h(v_{-m}) = v_{m+1} - \frac{n-m}{n} v_{m-1}$ and $h(v_{-j}) = v_{m+1} - \frac{n-m}{n} v_m$ for $j = 1, \ldots, m-1$. Thereby, in this case we have

$$\begin{aligned}
&\sum_{i=1}^{n} h(v_{-i}) - \sum_{i=1}^{m} \tau(v_{-i}) \\
&= \left[(m-1)\left(v_{m+1} - \frac{n-m}{n} v_m\right) + \left(v_{m+1} - \frac{n-m}{n} v_{m-1}\right)\right. \\
&\left. +(n-m)\left(v_m - \frac{n-m}{n} v_{m-1}\right)\right] - m v_{m+1} \\
&= \frac{(n-m)(n-m+1)}{n}\left(v_m - v_{m-1}\right) \quad \leq \quad 0,
\end{aligned}$$

and so the SD mechanism is subsidy-free. The ratio is bounded as follows:

$$\begin{aligned}
r^{\mathrm{SD}}(n,m) &= \frac{\sum_{i=1}^m v_i + \sum_{i=1}^n h(v_{-i}) - \sum_{i=1}^m \tau(v_{-i})}{\sum_{i=1}^m v_i} \\
&= \frac{\sum_{i=1}^m v_i + \frac{(n-m)(n-m+1)}{n}\left(v_m - v_{m-1}\right)}{\sum_{i=1}^m v_i} \\
&= 1 + \frac{(n-m)(n-m+1)\left(v_m - v_{m-1}\right)}{n\sum_{i=1}^m v_i} \\
&\geq 1 + \frac{(n-m)(n-m+1)\left(\frac{n-m}{n}v_{m-1} - v_{m-1}\right)}{n\sum_{i=1}^m v_i} \\
&= 1 - \frac{(n-m)(n-m+1)m v_{m-1}}{n^2\sum_{i=1}^m v_i} \geq 1 - \frac{(n-m)(n-m+1)m}{n^2\left[(m-1)+\frac{n-m}{n}\right]} \\
&= 1 - \frac{(n-m)(n-m+1)}{n(n-1)} = 1 - \frac{\binom{n-m+1}{2}}{\binom{n}{2}},
\end{aligned}$$

where the inequalities follow from $v_m \geq \frac{n-m}{n}v_{m-1}$ and $v_j \geq v_{m-1} \geq 0$ for $j = 1, \ldots, m-1$.

(Ib) If $\frac{n-m}{n}v_{m-1} > v_{m+1} \geq \frac{n-m}{n}v_m$, then $\tau(v_{-m}) = \frac{n-m}{n}v_{m-1}$ and $h(v_{-m}) = 0$. For agents $i = 1, \ldots, m-1$ we have $\tau(v_{-i}) = v_{m+1}$ and $h(v_{-j}) = v_{m+1} - \frac{n-m}{n}v_m$, as before. In this case,

$$\begin{aligned}
&\sum_{i=1}^n h(v_{-i}) - \sum_{i=1}^m \tau(v_{-i}) \\
&= \left[(m-1)\left(v_{m+1} - \frac{n-m}{n}v_m\right) + (n-m)\left(v_m - \frac{n-m}{n}v_{m-1}\right)\right] \\
&- \left[(m-1)v_{m+1} + \frac{n-m}{n}v_{m-1}\right] \\
&= \frac{(n-m)(n-m+1)}{n}\left(v_m - v_{m-1}\right) \quad \leq \quad 0,
\end{aligned}$$

as required by the no–subsidy constraint. The bound on the ratio is achieved in the same way as in the previous case.

(Ic) Finally, if $\frac{n-m}{n}v_{m-1} \geq \frac{n-m}{n}v_m \geq v_{m+1}$ (with at least one inequality being strict), then all agents $i = 1, \ldots, m$ get zero rebates, and the thresholds are $\tau(v_{-m}) = \frac{n-m}{n}v_{m-1}$ and $\tau(v_{-j}) = \frac{n-m}{n}v_m$ for

$j = 1, \ldots, m-1$. Now,

$$\begin{aligned}
&\sum_{i=1}^{n} h(v_{-i}) - \sum_{i=1}^{m} \tau(v_{-i}) \\
&= (n-m)\left(v_m - \frac{n-m}{n} v_{m-1}\right) - \left[(m-1)\frac{n-m}{n} v_m + \frac{n-m}{n} v_{m-1}\right] \\
&= \frac{(n-m)(n-m+1)}{n}\left(v_m - v_{m-1}\right) \quad \leq \quad 0,
\end{aligned}$$

so the no–subsidy holds. The ratio bound follows as in the previous case.

(II) Assume $v_m < \frac{n-m}{n} v_{m-1}$, that is, $m-1$ items are allocated. This also implies $v_{m+1} < \frac{n-m}{n} v_{m-1}$, and so for all agents $i = m, \ldots, n$ we have $\tau(v_{-i}) = \frac{n-m}{n} v_{m-1}$ and $h(v_{-i}) = 0$. For agents $i = 1, \ldots, m-1$, there are two possibilities as follows.

(IIa) If $v_{m+1} \geq \frac{n-m}{n} v_m$, then $\tau(v_{-i}) = v_{m+1}$ and $h(v_{-i}) = v_{m+1} - \frac{n-m}{n} v_m$ for all $i = 1, \ldots, m-1$. Thus, in this case we have

$$\begin{aligned}
&\sum_{i=1}^{n} h(v_{-i}) - \sum_{i=1}^{m-1} \tau(v_{-i}) \\
&= (m-1)\left(v_{m+1} - \frac{n-m}{n} v_m\right) - (m-1)v_{m+1} = -\frac{(m-1)(n-m)}{n} v_m \leq 0,
\end{aligned}$$

and so the SD mechanism is subsidy-free. The ratio is bounded as follows:

$$\begin{aligned}
r^{\mathrm{SD}}(n,m) &= \frac{\sum_{i=1}^{m-1} v_i + \sum_{i=1}^{n} h(v_{-i}) - \sum_{i=1}^{m-1} \tau(v_{-i})}{\sum_{i=1}^{m} v_i} \\
&= \frac{\sum_{i=1}^{m-1} v_i - \frac{(m-1)(n-m)}{n} v_m}{\sum_{i=1}^{m} v_i} = \frac{\sum_{i=1}^{m} v_i - \frac{(m-1)(n-m)}{n} v_m - v_m}{\sum_{i=1}^{m} v_i} \\
&= 1 - \frac{m(n-m+1)v_m}{n\sum_{i=1}^{m} v_i} > 1 - \frac{m(n-m+1)}{n\left((m-1)\frac{n}{n-m} + 1\right)} \\
&= 1 - \frac{(n-m)(n-m+1)}{n(n-1)} = 1 - \frac{\binom{n-m+1}{2}}{\binom{n}{2}},
\end{aligned}$$

where the inequality is implied by $v_i \geq v_{m-1} > \frac{n}{n-m} v_m \geq 0$ for $i = 1, \ldots, m-1$.

(IIb) If $v_{m+1} < \frac{n-m}{n} v_m$, then $\tau(v_{-i}) = \frac{n-m}{n} v_m$ and $h(v_{-i}) = 0$ for all $i = 1, \ldots, m-1$. In this case,

$$\sum_{i=1}^{n} h(v_{-i}) - \sum_{i=1}^{m-1} \tau(v_{-i})$$
$$= 0 - (m-1)\frac{n-m}{n} v_m = -\frac{(m-1)(n-m)}{n} v_m \quad \leq \quad 0,$$

so the no–subsidy holds, and the ratio bound follows as before.

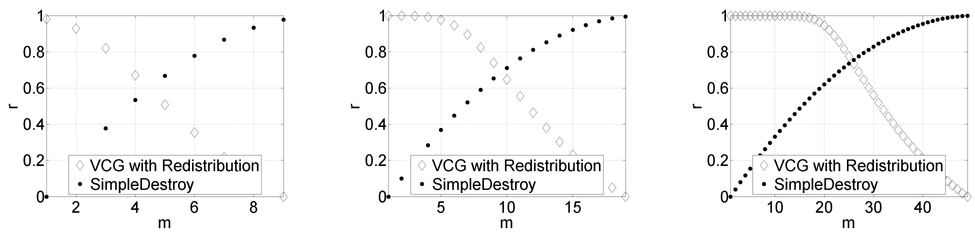
The proof is now complete. □

Recall that the ratio of the optimal VCG mechanism is close to 1 when the number of items is small. However, it is not difficult to check that it decreases as the number of items increases, and reaches 0 once the number of items is as high as it can be: one fewer than the number of agents. Indeed, this ratio, derived by Moulin (2009) and Guo and Conitzer (2009), is

$$r^{\mathrm{VCG}}(n,m) = 1 - \frac{\binom{n-1}{m}}{\sum_{j=m}^{n-1} \binom{n-1}{j}}$$

On the other hand, as shown above, the ratio of SimpleDestroy increases with the number of items. For each n, there exists a unique integer $m^* < n$ such that SimpleDestroy overtakes the optimal VCG mechanism, whenever there are at least m^* items to allocate. While there is no simple closed-form expression of m^* as a function of n, it is simple enough to compute it numerically for specific values of n, using the expressions of r^{VCG} and r^{SD}. We observe that m^* seems to always fall in the neighborhood of 1/2 (see Figure 10). SimpleDestroy performs best when the number of items is the largest, i.e. when the optimal VCG mechanism performs the worst. This motivates the definition of the *hybrid mechanism*, which uses the optimal VCG mechanism when the number of items is less than m^* and SimpleDestroy otherwise.

We now study the limit case when both m and n are large. The purpose of this asymptotic analysis is to better understand how SimpleDestroy and VCG compare, and in which circumstances to employ one or the other in the hybrid mechanism. Consider two increasing sequences (n_q) and (m_q) of positive integers such that $m_q < n_q$ for all q, the number $n_q \to \infty$ when $q \to \infty$, and the sequence $(\frac{m_q}{n_q})$ converges to some $\alpha \in]0,1[$ (thus to be interpreted as the maximal percentage of the population that could receive an item).



(a) $n = 10$ (b) $n = 20$ (c) $n = 50$

Figure 10: Performance of optimal VCG with Redistribution and SimpleDestroy mechanisms.

Proposition 1.

1.
$$r^{VCG}(\alpha) := \lim_{q\to\infty} r^{VCG}(n_q, m_q) = \begin{cases} 1 & \alpha \le 1/2 \\ \frac{1-\alpha}{\alpha} & \alpha > 1/2 \end{cases}$$
(see Moulin (2009), Theorem 3).
2. $r^{SD}(\alpha) := \lim_{q\to\infty} r^{SD}(n_q, m_q) = 2\alpha - \alpha^2$.
3. *The asymptotically best hybrid mechanism is obtained by choosing* $m^* \sim 0.555n$. *It guarantees a ratio of at least* 0.8 *for any* α.

Proof Note that

$$r^{\mathrm{SD}}(n, m) = 1 - \frac{(n-m)(n-m+1)}{n(n-1)} = \frac{2(m-1)}{n-1} - \frac{m(m-1)}{n(n-1)}$$

Hence,

$$\lim_{q\to\infty} r^{\mathrm{SD}}(n_q, m_q) = \lim_{q\to\infty} \left(\frac{2(m_q-1)}{n_q-1} - \frac{m_q(m_q-1)}{n_q(n_q-1)} \right) = 2\alpha - \alpha^2$$

The asymptotically best hybrid mechanism and its associated guaranteed ratio are then obtained by solving the equation $\frac{1-m^*}{m^*} = 2m^* - (m^*)^2$. $\square$

Notice that $r^{\mathrm{SD}}(\cdot)$ is non-negative and monotonically increasing for α between 0 and 1; moreover, it is concave (quadratic in α), and so the ratio becomes quickly higher for relatively low α's. The function $r^{\mathrm{VCG}}(\cdot)$, on the other hand, is monotonically decreasing and convex when $\alpha > 1/2$, implying that the ratio becomes quickly lower for α's larger than 1/2. For instance, if there are enough items to serve 75% of the population, then SimpleDestroy asymptotically guarantees the ratio of $\frac{15}{16}$, while the optimal VCG mechanism asymptotically guarantees the ratio of only 1/3.

6. Discussion

Allocative efficiency of VCG mechanisms comes in conflict with social welfare in a basic allocation model we consider—a lot of money may need to be burnt to maintain strategy-proofness, especially when the proportion of allocated agents is high. In particular, the social welfare is lost completely when there are enough items for all but one agent, and all agents have the same value for consuming an item. In this case, the VCG payment equals the value, and no redistribution is possible (Moulin, 2009; Guo and Conitzer, 2009).

It turns out that a small departure from efficiency allows to recover most of the loss. Specifically, this paper presents the SimpleDestroy mechanism that sometimes does not allocate to the last agent who would be allocated under an efficient mechanism, and guarantees a high level of social welfare when the number of items is at least half the number of agents. In contrast to efficient mechanisms, the welfare guaranteed by this mechanism increases with the percentage of allocated agents, and rapidly goes to 1 as the number of items approaches the number agents. Furthermore, it follows that a hybrid mechanism applying VCG for $m < .555n$ and SimpleDestroy otherwise, guarantees a high level of social welfare (asymptotic ratio of at least 0.8) for all allocation instances.

Our results are guided by an algorithmic procedure that exploits linearity inherent in the model. Specifically, we restrict attention to a class of mechanisms where optimization can be performed via linear programming, and numerically find mechanisms which are optimal within this restricted class.[12] In many cases, these mechanisms guarantee social welfare which is close to the total social welfare, and thus no significant improvement is possible. However, they are not generally optimal. For instance, for the problem with 3 agents and 2 items we were able to find the following provably optimal

[12]Similar methodology was used in (Guo and Conitzer, 2010a) to derive optimal payments when free items are efficiently allocated. More generally, algorithmic approaches to mechanism design problems are considered in (Conitzer and Sandholm, 2002; Guo and Conitzer, 2010a).

mechanism[13]:

$$\tau(w) = \max\left(\frac{1}{4}(w^1 + w^2), w^2\right)$$

$$h(w) = \begin{cases} \frac{2}{32}w^2 & \text{if } \frac{1}{9}w^1 \geq w^2 \\ -\frac{10}{32}w^1 + \frac{11}{32}w^2 & \text{if } \frac{1}{3}w^1 \geq w^2 \geq \frac{1}{9}w^1 \\ -\frac{4}{32}w^1 + \frac{20}{32}w^2 & \text{otherwise} \end{cases}$$

This mechanism guarantees the ratio of .75, which is higher than what can be obtained for 3 agents and 2 items by the mechanisms within the restricted class we considered. The challenge is then to find a general mechanism that is provably optimal for any number of items and agents. A few simpler questions will probably need to be answered along the way: Is the natural property that if an agent with value v_i is allocated than all agents with values above v_i are also allocated, consistent with an optimal mechanism? Is destroying more than one item ever beneficial? Another interesting open question is whether a broader class of mechanisms would provide a significant *asymptotic* improvement for general problem instances.

References

Bailey, M.J., 1997. The demand revealing process: To distribute the surplus. Public Choice 91, 107–26.

Cavallo, R., 2006. Optimal decision-making with minimal waste: Strategyproof redistribution of vcg payments, in: AAMAS'06, Hakodate, Japan.

Conitzer, V., Sandholm, T., 2002. Complexity of mechanism design, in: UAI, pp. 103–110.

Faltings, B., 2005. A Budget-balanced, Incentive-compatible Scheme for Social Choice, in: Faratin, P., Rodriguez-Aguilar, J.A. (Eds.), Agent-Mediated Electronic Commerce VI. Springer. volume 3435, pp. 30–43.

Green, J.R., Laffont, J.J., 1979. Incentives in public decision-making. North Holland, New York.

[13]The optimality proof of this mechanism can be found in Naroditskiy (2009).

Guo, M., Conitzer, V., 2008. Better redistribution with inefficient allocation in multi-unit auctions with unit demand, in: EC'08, pp. 210–219.

Guo, M., Conitzer, V., 2009. Worst-case optimal redistribution of vcg payments in multi-unit auctions. Games and Economic Behavior 67, 69 – 98.

Guo, M., Conitzer, V., 2010a. Computationally feasible automated mechanism design: General approach and case studies, in: AAAI.

Guo, M., Conitzer, V., 2010b. Optimal-in-expectation redistribution mechanisms. Artif. Intell. 174, 363–381.

Hartline, J.D., Roughgarden, T., 2008. Optimal mechanism design and money burning, in: STOC, pp. 75–84.

Moulin, H., 2009. Almost budget-balanced vcg mechanisms to assign multiple objects. Journal of Economic Theory 144, 96–119.

Moulin, H., Shenker, S., 2001. Strategyproof sharing of submodular costs: budget balance versus efficiency. Economic Theory 18, 511–533.

Naroditskiy, V., 2009. Select Problems at the Intersection of Computer Science and Economics. Ph.D. thesis. Brown University.

Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V., 2007. Algorithmic Game Theory. Cambridge University Press, New York, NY, USA.

Porter, R., Shoham, Y., Tennenholtz, M., 2004. Fair imposition. Journal of Economic Theory 118, 209 – 228.