

VCG Redistribution with Gross Substitutes

Mingyu Guo

Department of Computer Science
University of Liverpool
Liverpool, Merseyside, UK
Mingyu.Guo@liverpool.ac.uk

Abstract

For the problem of allocating resources among multiple strategic agents, the well-known Vickrey-Clarke-Groves (VCG) mechanism is efficient, strategy-proof, and it never incurs a deficit. However, in general, under the VCG mechanism, payments flow out of the system of agents, which reduces the agents' utilities. VCG redistribution mechanisms aim to return as much of the VCG payments as possible back to the agents, without affecting the desirable properties of the VCG mechanism. Most previous research on VCG redistribution mechanisms has focused on settings with homogeneous items and/or settings with unit-demand agents. In this paper, we study VCG redistribution mechanisms in the more general setting of combinatorial auctions. We show that when the gross substitutes condition holds, we are able to design mechanisms that guarantee to redistribute a large fraction of the VCG payments.

Introduction

For the problem of allocating resources among multiple strategic agents, the well-known Vickrey-Clarke-Groves (VCG) mechanism (also known as the Clarke mechanism) satisfies the following properties:

- **Efficiency:** The allocation maximizes the agents' total valuation.
- **Strategy-proofness:** For any agent, reporting truthfully is always a dominant strategy no matter how the other agents report.
- **Individual Rationality:** Every agent's final utility is always non-negative.
- **Non-deficit:** No external subsidy is ever needed.

However, in general, under the VCG mechanism, payments flow out of the system of agents, which reduces the agents' utilities. One way to increase the agents' utilities is to redistribute (return) the VCG payments back to the agents. (Cavallo 2006) proposed the idea of strategy-proof redistribution, which is that as long as every agent's redistribution (amount received) is independent of her own type, then the agents' incentives are not affected – the combined mechanism (VCG plus redistribution) is still strategy-proof, which

also implies that the combined mechanism is still efficient. A *feasible* redistribution scheme is a strategy-proof redistribution scheme that also maintains individual rationality and the non-deficit constraint. We call the combination of the VCG mechanism and a feasible redistribution scheme a *VCG redistribution mechanism*.¹

(Cavallo 2006) proposed a specific VCG redistribution mechanism for general combinatorial auctions, which we call the Cavallo mechanism. The Cavallo mechanism generally performs well, in the sense that, for many problem settings, the Cavallo redistribution scheme can successfully redistribute most of the VCG payment for many type profiles. Inspired by the Cavallo mechanism, (Guo and Conitzer 2009) characterized a VCG redistribution mechanism that maximizes the worst-case redistribution fraction (fraction of total VCG payment redistributed in the worst case) in the setting of multi-unit auctions with non-increasing marginal values. Independently, in the more restricted setting of multi-unit auctions with unit demand, the same mechanism was characterized by (Moulin 2009) for a different design objective. Compared with settings with *homogeneous* items, it is much more difficult to design good redistribution schemes in settings with *heterogeneous* items. In the setting of heterogeneous-item auctions with unit demand, (Gujar and Narahari 2008) conjectured a redistribution scheme that maximizes the worst-case redistribution fraction. However, the feasibility of the conjectured scheme is unknown. Aside from the conjecture, the authors later showed in (Gujar and Narahari 2009) that, in the unit demand setting, the Cavallo redistribution scheme's worst-case redistribution fraction is at least $\frac{n-2m}{n}$, where n and m are the number of agents and the number of items, respectively.

In this paper, we continue the search for redistribution schemes with high worst-case redistribution fractions in settings with heterogeneous items. (Guo and Conitzer 2008b) showed that in general combinatorial auctions, all feasible redistribution schemes' worst-case redistribution fractions are 0. This implies that positive worst-case redistribution fractions are only possible in settings with certain restrictions. In this paper, we focus our attention to combinatorial auction settings where the gross substitutes condi-

¹VCG redistribution mechanisms are special cases of Groves mechanisms.

tion (Kelso and Crawford 1982) holds. For the objective of maximizing worst-case redistribution fraction, the setting studied in this paper is the most general so far.² (Both multi-unit auctions with non-increasing marginal values and heterogeneous-item auctions with unit demand are special cases of combinatorial auctions with gross substitutes condition.) We first show that in combinatorial auctions with gross substitutes condition, the Cavallo mechanism’s worst-case redistribution fraction is exactly $\frac{n-m-1}{n}$. Since our result is derived in a more general setting, it actually implies that in the more restricted setting of heterogeneous-item auctions with unit demand, the Cavallo mechanism’s worst-case redistribution fraction is also exactly $\frac{n-m-1}{n}$. That is, the lower bound $\frac{n-2m}{n}$ derived earlier in (Gujar and Narahari 2009) is now known to be not strict. Incidentally, our result also implies that in combinatorial auctions with gross substitutes condition, when $n = m + 2$, the Cavallo mechanism is worst-case optimal among all VCG redistribution mechanisms.³ Then, with the help of automated mechanism design, we solve for redistribution schemes that significantly outperform the Cavallo redistribution scheme in cases where n is much larger than m . For example, when there are 50 agents and 2 items, we find a redistribution scheme whose worst-case redistribution fraction is around 0.999999, while the Cavallo redistribution scheme’s worst-case redistribution fraction is only 0.94 (the worst-case fractions of waste differ by several magnitudes).

Problem Description

There are n agents ($I = \{1, 2, \dots, n\}$) and m ($m > 1$) heterogeneous items ($J = \{1, 2, \dots, m\}$). We only consider cases where $n > m + 1$.⁴

An agent has different valuations for different bundles of items. We use $v_i(B)$ to denote agent i ’s valuation for winning bundle B . We assume that $v_i(\emptyset) = 0$ for all i . We further assume *free disposal*: for any $B \supseteq B'$ and any i , we have $v_i(B) \geq v_i(B')$. An agent’s utility equals her valuation minus her payment.

An allocation of the items to the agents correspond to n disjoint bundles B_1, B_2, \dots, B_n , where B_i is the bundle

²Aside from the objective of this paper, other objectives have also been studied (Guo and Conitzer 2008a; de Clippel, Naroditskiy, and Greenwald 2009; Guo and Conitzer 2010b; 2008b; Apt et al. 2008). Most of the proposed mechanisms are only defined for settings with homogeneous items. One exception is (Guo and Conitzer 2008b), which gives two numerical algorithms for finding *undominated* redistribution schemes in general combinatorial auctions. Both numerical algorithms are based on discretization of the agents’ type spaces, which makes these two algorithms inapplicable to the worst-case objective discussed in this paper.

³(Guo and Conitzer 2009) showed that even in the more restricted setting of multi-unit auctions with unit demand, no feasible redistribution scheme’s worst-case redistribution fraction can be more than $\frac{n-m-1}{n}$ when $n = m + 2$.

⁴(Guo and Conitzer 2009) showed that in multi-unit auctions with non-increasing marginal values, when $n \leq m + 1$, the original VCG mechanism is worst-case optimal. Since the setting studied in this paper is more general, we have that in our setting, when $n \leq m + 1$, VCG is still worst-case optimal.

won by agent i . An efficient allocation is an allocation that maximizes $\sum_{i=1}^n v_i(B_i)$ (total valuation). Similarly, when we only consider a subset of agents S (removing the other agents from the auction), then an allocation of the items to the agents in S correspond to $|S|$ disjoint bundles, where B_i is the bundle won by agent i for $i \in S$. An efficient allocation of the items to the agents in S is then an allocation that maximizes $\sum_{i \in S} v_i(B_i)$. We use $U(S)$ to denote the total valuation of the agents in S when we allocate the items to them efficiently.

To avoid dealing with tie-breaking, we assume that the set of possible type profiles contains only those that are tie-free: for any set of agents S , the efficient allocation of the items to the agents in S is unique.⁵

The VCG mechanism allocates the items to the agents efficiently. If an agent does not win anything, then she pays nothing. If an agent wins something, then she pays “how much she hurts the other agents as a result of her presence”. Let X be the set of winners when we allocate the items to all the agents efficiently. For every winner $i \in X$, we denote her bundle by B_i . Winner i ’s VCG payment is defined to be $U(I - i) - (U(I) - v_i(B_i))$.⁶ The total VCG payment is then $\sum_{i \in X} (U(I - i) - U(I) + v_i(B_i)) = \sum_{i \in X} (U(I - i) - U(I)) + U(I) = \sum_{i \in X} U(I - i) - (|X| - 1)U(I)$. Since removing a loser does not change $U(I)$, we have that the total VCG payment can also be written as $\sum_{i \in X} U(I - i) - (|X| - 1)U(I) = \sum_{i \in X} U(I - i) + (n - |X|)U(I) - (n - 1)U(I) = \sum_{i \in I} U(I - i) - (n - 1)U(I)$. We use $VCG(I)$ to denote the total VCG payment when every agent participates in the auction. Similarly, we use $VCG(S)$ to denote the total VCG payment when we use the VCG mechanism to allocate the items to only the agents in S .

Our objective is to design a feasible redistribution scheme that always redistributes a large fraction of the total VCG payment even in the worst case.

Gross Substitutes

(Guo and Conitzer 2008b) showed that in general combinatorial auctions, when there are at least two items and at least three agents, the worst-case redistribution fraction of any feasible redistribution scheme is 0. Despite this negative result, we are able to achieve positive worst-case redistribution fraction in more restricted settings. (Guo and Conitzer 2009) and (Gujar and Narahari 2009) are two examples. These two papers studied multi-unit auctions with non-increasing marginal values and heterogeneous-item auctions with unit

⁵We cannot simply assume that a consistent tie-breaking rule exists, because we need the tie-breaking rule to satisfy Proposition 2, which is that, winners still win after we remove some other agents. Our conjecture is that such a tie-breaking rule exists, but we do not know how to construct one (without additional technical assumptions). Alternatively, we could drop this tie-free assumption. We allow all possible type profiles, but when we redistribute, we perturb any type profile that is not tie-free infinitesimally so that it becomes tie-free. This way we avoid dealing with tie-breaking, but certain mechanism properties may be violated infinitesimally due to perturbation.

⁶ $I - i$ is short for $I - \{i\}$.

demand, respectively. In this paper, we show that positive worst-case redistribution fraction is achievable in a more general setting — combinatorial auctions with gross substitutes condition.

The gross substitutes condition was first proposed in (Kelso and Crawford 1982). For completeness, we include its definition below. The terminology and notation are due to (Lehmann, Lehmann, and Nisan 2003).

We associate each item with a price. Under price vector $\vec{p} = (p_1, p_2, \dots, p_m)$, item j is priced at p_j . Agent i 's surplus of winning bundle B under price vector \vec{p} is $v_i(B) - \sum_{j \in B} p_j$. Agent i 's preferred bundles under \vec{p} are the bundles that maximize her surplus.

Definition 1. Agent i 's type satisfies the *gross substitutes* condition if for any item j , any two price vectors \vec{p} and \vec{q} , with $\vec{q} \geq \vec{p}$ (element-wise comparison) and $p_j = q_j$, we have that if item j is in a preferred bundle of agent i under \vec{p} , then there exists a preferred bundle of agent i under \vec{q} that also contains item j .

In words, agent i 's type satisfies the gross substitutes condition if her demand for an item does not decrease when the prices of the other items increase.

Combinatorial auctions with gross substitutes condition are combinatorial auctions in which every agent's type satisfies the gross substitutes condition. (Gul and Stacchetti 1999) listed several example settings of combinatorial auctions with gross substitutes condition. Among them are the aforementioned multi-unit auctions with non-increasing marginal values and heterogeneous-item auctions with unit demand. A third example is combinatorial auctions with additive valuations. Other example settings are omitted.

We use $G(B, S)$ to denote the total valuation of the agents in S , when we allocate the items in B to them efficiently. $G(J, S) = U(S)$. It was pointed out in (Yokoo, Sakurai, and Matsubara 2004) that when every agent's type satisfies the gross substitutes condition, the following two properties hold:

Submodularity in items: For any set of agents S , for any two item bundles B_1 and B_2 , we have $G(B_1, S) + G(B_2, S) \geq G(B_1 \cap B_2, S) + G(B_1 \cup B_2, S)$.

Submodularity in agents: For any two sets of agents S_1 and S_2 , we have $U(S_1) + U(S_2) \geq U(S_1 \cap S_2) + U(S_1 \cup S_2)$.

Submodularity in agents leads to the following two propositions, which are needed for deriving our results.

Proposition 1. *When every agent's type satisfies the gross substitutes condition, we have that for any set of agents S and any $i \in S$, $VCG(S) \geq VCG(S - i)$. That is, the VCG mechanism is revenue monotonic.*

Proof. If S contains only one agent, then $VCG(S) = VCG(S - i) = 0$. We then consider cases where $|S| \geq 2$. As illustrated earlier, $VCG(S) = \sum_{j \in S} U(S - j) - (|S| - 1)U(S)$ and $VCG(S - i) = \sum_{j \in S - i} U(S - i - j) - (|S| - 2)U(S - i)$. That is, we need to show $\sum_{j \in S} U(S - j) - (|S| - 1)U(S) \geq \sum_{j \in S - i} U(S - i - j) - (|S| - 2)U(S - i)$. We have $\sum_{j \in S} U(S - j) = \sum_{j \in S - i} U(S - j) + U(S - i)$. That is, we only need to show $\sum_{j \in S - i} U(S - j) - (|S| -$

$1)U(S) \geq \sum_{j \in S - i} U(S - i - j) - (|S| - 1)U(S - i)$. According to submodularity in agents, we have for any $j \in S - i$, $U(S - j) - U(S) \geq U(S - i - j) - U(S - i)$. That is, $\sum_{j \in S - i} U(S - j) - \sum_{j \in S - i} U(S) \geq \sum_{j \in S - i} U(S - i - j) - \sum_{j \in S - i} U(S - i)$. That is, $\sum_{j \in S - i} U(S - j) - (|S| - 1)U(S) \geq \sum_{j \in S - i} U(S - i - j) - (|S| - 1)U(S - i)$. \square

Proposition 2. *Let S be any set of agents. Given that the gross substitutes condition holds, if an agent $i \in S$ is a winner when we allocate the items to the agents in S according to the VCG mechanism, then for any $S' \subset S$ with $i \in S'$, i is still a winner when we allocate the items to the agents in S' according to the VCG mechanism.*

The above proposition says that a winner still wins after we remove some other agents from the auction. Lemma 1 of (Gujar and Narahari 2009) showed the same result for heterogeneous-item auctions with unit demand.

Proof. Let B_i be the bundle won by i when the set of participating agents is S . We have that $v_i(B_i) > G(J, S - i) - G(J - B_i, S - i)$ for the following reasons. $v_i(B_i)$ is i 's valuation for winning B_i . $G(J, S - i) - G(J - B_i, S - i)$ is i 's VCG payment for winning B_i . The first expression must be as large as the second one. Otherwise, i will not win B_i . If the two expressions are the same, then i may or may not win B_i , depending on how we break ties. Since we only consider type profiles that never result in ties, we have that $v_i(B_i)$ is strictly larger.

We only need to show that after we remove any agent $j \neq i$ from S , i is still a winner. After removing j , to win B_i , i needs to pay $G(J, S - i - j) - G(J - B_i, S - i - j)$. We prove that it is less than or equal to the original amount i needs to pay to win B_i . This implies that i is still a winner (she can at least win B_i to gain some positive utility, though her most preferable bundle may change). Let us construct an agent x , who is not in S . Let x 's valuation be additive, and she values every item in B_i extremely high, and she does not value the other items at all. The point is to make sure that x wins B_i when facing both agent set $S - i$ and agent set $S - i - j$. Let x 's valuation for B_i be X . As we mentioned, we need to prove $G(J, S - i - j) - G(J - B_i, S - i - j) \leq G(J, S - i) - G(J - B_i, S - i)$. We only need to prove $G(J, S - i - j) - G(J - B_i, S - i - j) - X \leq G(J, S - i) - G(J - B_i, S - i) - X$. That is, we only need to prove $U(S - i - j) - U(S - i - j + x) \leq U(S - i) - U(S - i + x)$. According to submodularity in agents, this inequality holds. \square

Cavallo Redistribution Scheme

In this section, we show that, in combinatorial auctions with gross substitutes condition, the worst-case redistribution fraction of the Cavallo redistribution scheme (Cavallo 2006) is exactly $\frac{n-m-1}{n}$.⁷

⁷As we mentioned earlier, we only consider cases where $n > m + 1$, because when $n \leq m + 1$, one optimal redistribution scheme for our objective is simply redistributing nothing.

In revenue monotonic settings, the Cavallo redistribution scheme is identical to the Bailey redistribution scheme (Bailey 1997). Under it, agent i 's redistribution is equal to $VCG(I - i)/n$ ($1/n$ times the total VCG payment after removing i herself). We see that i 's redistribution does not depend on her own type. Thus, the Cavallo scheme is strategy-proof and efficient. Individual rationality and the non-deficit constraint were proven to be satisfied in (Cavallo 2006).

First of all, (Guo and Conitzer 2009) showed that in the more restricted setting of multi-unit auctions with unit demand, the Cavallo redistribution scheme's worst-case redistribution fraction is exactly $\frac{n-m-1}{n}$. Since we are dealing with a more general setting, we automatically have that in our setting, the Cavallo redistribution scheme's worst-case redistribution fraction is at most $\frac{n-m-1}{n}$. Next, we show that the Cavallo redistribution scheme's worst-case redistribution fraction is also at least $\frac{n-m-1}{n}$.

Theorem 1. *In combinatorial auctions with gross substitutes condition, the Cavallo redistribution scheme's worst-case redistribution fraction is exactly $\frac{n-m-1}{n}$.*

Proof. Let X be the set of winners when we allocate the items to the agents using the VCG mechanism. The total VCG payment is the total payment from the agents in X . We use p_i to denote the VCG payment by winner $i \in X$. The total VCG payment is $\sum_{i \in X} p_i$.

$VCG(I - j)$ is the total VCG payment when we allocate the items to the agents other than j . When calculating $VCG(I - j)$ for $j \in I - X$, the winners are still those in X , but their payments may be different from the p_i . For $i \in X$ and $j \in I - X$, we use p_i^j to denote winner i 's payment when we allocate the items to the agents other than j . For $j \in I - X$, $VCG(I - j) = \sum_{i \in X} p_i^j$. The total redistribution received by the agents in $I - X$ are then $\frac{1}{n} \sum_{i \in X; j \in I - X} p_i^j$ (less than the total redistribution). To show that the total redistribution is at least $\frac{n-m-1}{n} VCG(I)$, it suffices to show $\frac{1}{n} \sum_{i \in X; j \in I - X} p_i^j \geq \frac{n-m-1}{n} \sum_{i \in X} p_i$. Then, to show the above inequality, it suffices to show that for all $i \in X$, $\sum_{j \in I - X} p_i^j \geq (n - m - 1)p_i$.

Let us consider an arbitrary winner i ($i \in X$). Let B_i be the bundle that i wins when we allocate all the items to the agents using the VCG mechanism. We have that $p_i = U(I - i) - (U(I) - v_i(B_i))$. Let X' be the set of winners when we allocate the items to the agents other than i . According to Proposition 2, we have $X - i \subseteq X'$ (the other winners still win when i is removed from the auction). Therefore, $X' \cup X = X' \cup \{i\}$. Since $|X'| \leq m$ (at most m winners), we have $|X' \cup X| \leq m + 1$. For any $j \notin X' \cup X$, when we allocate all the items to the agents other than j using the VCG mechanism, i still wins B_i , but her payment is now $p_i^j = U(I - i - j) - (U(I - j) - v_i(B_i))$. Since $j \notin X'$ (j is not a winner even after i is removed), we have $U(I - i - j) = U(I - i)$. Similarly, since $j \notin X$, we have $U(I - j) = U(I)$. That is, for all $j \notin X' \cup X$, $p_i^j = U(I - i) - (U(I) - v_i(B_i)) = p_i$. That is, $\sum_{j \in I - X} p_i^j \geq \sum_{j \notin X' \cup X} p_i^j = \sum_{j \notin X' \cup X} p_i \geq (n - m - 1)p_i$. This completes the proof. \square

(Gujar and Narahari 2009) showed that the Cavallo redistribution scheme's worst-case redistribution fraction is at least $\frac{n-2m}{n}$ in heterogeneous-item auctions with unit demand. Since our setting is more general, our theorem implies that Cavallo redistribution scheme's worst-case redistribution fraction is actually at least $\frac{n-m-1}{n}$ in heterogeneous-item auctions with unit demand. Then, since heterogeneous-item auctions with unit demand is more general than multi-unit auctions with unit demand, we also have that the Cavallo redistribution scheme's worst-case redistribution fraction is at most $\frac{n-m-1}{n}$ in heterogeneous-item auctions with unit demand. That is, Cavallo redistribution scheme's worst-case redistribution fraction is also exactly $\frac{n-m-1}{n}$ in heterogeneous-item auctions with unit demand.

Gadgets for Constructing Better Redistribution Schemes

In this section, we define a series of functions and show their properties. The motivation behind these functions may not be clear initially, but we will see later that they are useful gadgets for constructing better redistribution schemes.

Definition 2. For any set of agents S , we define the k -th winner set in S as follows:

The first winner set in S , denoted by $W(S, 1)$, is the set of winners in S when we use the VCG mechanism to allocate the items to only the agents in S .

For $k > 1$, the k -th winner set in S , denoted by $W(S, k)$, is the set of winners in S when use the VCG mechanism to allocate the items to the agents in $S - W(S, \leq k - 1)$. Here, $W(S, \leq k - 1) = \cup_{1 \leq j \leq k-1} W(S, j)$.

The meaning of the first winner set is clear. The second winner set is the set of new winners after we remove all the agents in the first winner set. Then, the third winner set is the set of new winners after we also remove all the agents in the second winner set. The k -th winner set is the set of new winners after we remove all the agents in the first $k - 1$ winner sets. The first several winner sets are non-empty. However, eventually, all agents are removed. That is, for large k , the k -th winner set is empty. We use k^S to denote the largest k satisfying that the k -th winner set is non-empty. S is divided into k^S disjoint winner sets.

We make the following observation. After we remove an arbitrary agent in the i -th winner set of S , the earlier winner sets (j -th winner set with $j < i$) do not change. The later winner sets may change, but they only change as follows: After removing one agent from the i -th winner set, the i -th winner set may be incomplete (some winner is gone). Some agents in the $(i + 1)$ -th winner set may be promoted to the i -th winner set. Only agents in the $(i + 1)$ -th winner set can possibly be promoted to fulfill the i -th winner set, according to Proposition 2. (If an agent is promoted to the i -th winner set, then consider removing all the agents in the first i winner sets, except the promoted agent. The promoted agent is now a winner, which means that she must be originally from the $(i + 1)$ -th winner set.) Next, the $(i + 1)$ -th winner set may be incomplete because some of its agents may have been promoted. That is, some agents in the $(i + 2)$ -th winner set may be promoted to the $(i + 1)$ -th winner set. Finally, the

$(k^S - 1)$ -th winner set may be incomplete, and some agents in the k^S -th winner set may be promoted to the $(k^S - 1)$ -th winner set. The observation is summarized in the following lemma.

Lemma 1. *Let S be any set with $k^S > 1$. For any $1 \leq i \leq k^S - 1$, for any $x \in W(S, i)$, we have the following:*

For $j < i$, $W(S - x, j) = W(S, j)$.

For j with $i \leq j \leq k^S - 1$, $W(S - x, j) \subseteq W(S, j) \cup W(S, j + 1)$.

For any j with $1 \leq j \leq k^S - 1$, $W(S - x, \leq j) \subseteq W(S, \leq j + 1)$ and $W(S - x, \leq j) \cup \{x\} \supseteq W(S, \leq j)$.

Next, we define another function R based on W .

Definition 3. For any set of agents S , we define function R as follows:

$R(S, 1) = VCG(S)$. That is, $R(S, 1)$ is just the total VCG payment when we use the VCG mechanism to allocate the items to the agents in S .

For $k \geq 2$, $R(S, k) = \frac{1}{km} (\sum_{j \in W(S, \leq k)} R(S - j, k - 1) + (km - |W(S, \leq k)|)R(S, k - 1))$.

The meaning of $R(S, 1)$ is clear. Here, we give some intuition on the meaning of $R(S, 2)$. When S is the set of participating agents, the total VCG payment is $VCG(S)$. Under the Cavallo mechanism, the redistribution received by agent $j \in S$ equals $\frac{1}{|S|} VCG(S - j)$. As was discussed in the proof of Theorem 1, the VCG payments only depend on the agents in the first two winner sets. That is, for any agent j not in $W(S, \leq 2)$, her Cavallo redistribution is exactly $\frac{1}{|S|}$ times the correct total VCG payment. That is, her redistribution is in some sense ‘‘correct’’. $|W(S, \leq 2)|$ is at most $2m$, since any winner set contains at most m winners. Therefore, there are (at most) $2m$ agents whose redistributions are ‘‘incorrect’’. $R(S, 1)$ is defined as the average of the redistributions received by these $2m$ agents. $R(S, 1)$ in some sense captures the ‘‘error’’ of the Cavallo mechanism. $R(S, k)$ for larger k are defined recursively.

R satisfies the following properties:

Lemma 2. *For any S and any k , if $S - W(S, \leq k + 1)$ is nonempty, then we have $R(S, k) = R(S - x, k)$ for any $x \in S - W(S, \leq k + 1)$.*

This lemma says that $R(S, k)$ only depends on the agents in the first $k + 1$ winner sets. Removing the other agents do not affect $R(S, k)$. Proof omitted due to space constraint.

Proposition 3. *For any $1 \leq k \leq \lfloor n/m \rfloor - 1$, we have $\sum_{i \in I} R(I - i, k) = (n - km - m)R(I, k) + (km + m)R(I, k + 1)$.*

We use one example to illustrate the usefulness of the above proposition. Let $n = 100$ and $m = 2$. We have $\sum_{i \in I} R(I - i, 1) = 96R(I, 1) + 4R(I, 2)$ (case $k = 1$) and $\sum_{i \in I} R(I - i, 2) = 94R(I, 2) + 6R(I, 3)$ (case $k = 2$). Let us consider the following strategy-proof redistribution scheme: agent i receives $\frac{1}{96}R(I - i, 1) - \frac{4}{96 \times 94}R(I - i, 2)$. The total redistribution is then $\frac{1}{96}(96R(I, 1) + 4R(I, 2)) - \frac{4}{96 \times 94}(94R(I, 2) + 6R(I, 3)) = R(I, 1) - \frac{4 \times 6}{96 \times 94}R(I, 3)$. We assume R is non-increasing in the second variable (we will prove next that this is indeed true). We have that every

agent’s redistribution is non-negative, and the total redistribution is at most $R(I, 1)$ and at least $(1 - \frac{4 \times 6}{96 \times 94})R(I, 1)$. We recall that $R(I, 1) = VCG(I)$. That is, this scheme satisfies the non-deficit constraint and has a worst-case redistribution fraction 0.99734. (For this pair of n and m , the Cavallo mechanism’s worst-case redistribution fraction is $\frac{100 - 2 - 1}{100} = 0.97$.)

Proof. Since $k + 1 \geq 2$, by definition of R , we have $(km + m)R(I, k + 1) = \sum_{j \in W(I, \leq k + 1)} R(I - j, k) + (km + m - |W(S, \leq k + 1)|)R(I, k)$. That is, we need to prove that $\sum_{j \in I} R(I - j, k) = (n - |W(S, \leq k + 1)|)R(I, k) + \sum_{j \in W(I, \leq k + 1)} R(I - j, k)$. The right-hand side can be rewritten as $\sum_{j \in I - W(I, \leq k + 1)} R(I, k) + \sum_{j \in W(I, \leq k + 1)} R(I - j, k)$.⁸ According to Lemma 2, when $j \in I - W(I, \leq k + 1)$, $R(I, k) = R(I - j, k)$. That is, the right-hand side is $\sum_{j \in I - W(I, \leq k + 1)} R(I - j, k) + \sum_{j \in W(I, \leq k + 1)} R(I - j, k)$, which is the same as the left-hand side. \square

Next, we prove that R is non-increasing in both variables.

Lemma 3. *For any S , any $x \in S$, and any $1 \leq k \leq k^S - 1$, we have $R(S, k) \geq R(S - x, k)$.*

Proof omitted due to space constraint.

Proposition 4. *For any S and any $2 \leq k \leq k^S$, $R(S, k) \leq R(S, k - 1)$.*

Proof. According to Lemma 3, $R(S, k) = \frac{1}{km} (\sum_{j \in W(S, \leq k)} R(S - j, k - 1) + (km - |W(S, \leq k)|)R(S, k - 1)) \leq \frac{1}{km} (\sum_{j \in W(S, \leq k)} R(S, k - 1) + (km - |W(S, \leq k)|)R(S, k - 1)) = R(S, k - 1)$. \square

Better Redistribution Scheme by AMD

As illustrated in the example in the previous section, we are able to construct better redistribution schemes based on R . Our construction is based on Automated Mechanism Design (AMD) (Conitzer and Sandholm 2002). Specifically, we focus our attention to a family of parameterized redistribution schemes. Then, by optimizing over the parameters, we solve for good redistribution schemes within the family of consideration. Similar approaches have been studied in (Likhodedov and Sandholm 2004; 2005; Guo and Conitzer 2010a).

We consider the following parameterized family of redistribution schemes (both the example redistribution scheme proposed in the previous section and the Cavallo redistribution scheme are special cases of this family): Agent i ’s redistribution equals $\sum_{1 \leq j \leq z} c_j R(I - i, j)$, where the c_j are the parameters, and $z = \lfloor n/m \rfloor - 1$.⁹

$R(I - i, j)$ does not depend on i ’s type. Therefore, it maintains strategy-proofness and efficiency. However, not

⁸This is true only when $I - W(I, \leq k + 1)$ is non-empty. To ensure this, we require that $(k + 1)m \leq n$ (the total size of the first $k + 1$ winner sets is at most $(k + 1)m$). That is, $k \leq \lfloor n/m \rfloor - 1$.

⁹The upper bound on j is to ensure that Proposition 3 holds.

all the c_j correspond to feasible redistribution schemes as we still need to make sure that individual rationality and the non-deficit constraint hold. The next lemma is helpful for finding the c_j that correspond to feasible schemes.

Lemma 4. (Guo and Conitzer 2009) *When the s_i and the x_i are independent, the following two sets of inequalities are equivalent:*

- 1) $s_1x_1 + s_2x_2 + \dots + s_ax_a \geq 0$ for all $x_1 \geq x_2 \geq \dots \geq x_a \geq 0$.
- 2) $s_1 + s_2 + \dots + s_b \geq 0$ for all $1 \leq b \leq a$.

For any i , we have $k^{I-i} \geq \lceil (n-1)/m \rceil \geq \lfloor n/m \rfloor - 1 = z$. According to Proposition 4, we have $R(I-i, j)$ is non-increasing in j for $1 \leq j \leq z$. That is, according to Lemma 4, the following is sufficient for ensuring the individual rationality constraint: For all $1 \leq b \leq z$, $\sum_{1 \leq t \leq b} c_t \geq 0$.

The total VCG payment is $VCG(I) = R(I, 1)$. The total redistribution under the above scheme is $\sum_{i \in I} \sum_{1 \leq j \leq z} c_j R(I-i, j) = \sum_{1 \leq j \leq z} c_j ((jm + m)R(I, j+1) + (n - jm - m)R(I, j)) = c_1(n - 2m)R(I, 1) + c_z(zm + m)R(I, z+1) + \sum_{2 \leq j \leq z} (c_{j-1}jm + c_j(n - jm - m))R(I, j)$. We introduce two additional constants $c_0 = c_{z+1} = 0$. With c_0 and c_{z+1} , the total redistribution can be rewritten as $\sum_{1 \leq j \leq z+1} (c_{j-1}jm + c_j(n - jm - m))R(I, j)$. We have $k^I \geq \lfloor n/m \rfloor = z + 1$. According to Proposition 4, we have that $R(I, j)$ is non-increasing in j for $1 \leq j \leq z + 1$. The following inequality ensures the non-deficit constraint: $R(I, 1) \geq \sum_{1 \leq j \leq z+1} (c_{j-1}jm + c_j(n - jm - m))R(I, j)$. The next inequality ensures that the worst-case redistribution fraction is at least α : $\alpha R(I, 1) \leq \sum_{1 \leq j \leq z+1} (c_{j-1}jm + c_j(n - jm - m))R(I, j)$. According to Lemma 4, to ensure $\sum_{1 \leq j \leq z+1} s_j R(I, j) \geq 0$, one sufficient condition is that for all $1 \leq b \leq z+1$, $\sum_{1 \leq t \leq b} s_t \geq 0$. That is, the non-deficit constraint and the worst-case redistribution constraint can be ensured by linear inequalities involving the c_j and α . Therefore, we are able to find a good assignment of the c_j (corresponding to a large α) using a linear program.¹⁰

Experimentally, when n/m is small, we can not find better schemes than the Cavallo redistribution scheme. When n/m is large, α^* (the worst-case redistribution fraction obtained based on the above technique) is higher than $\frac{n-m-1}{n}$ (the worst-case redistribution fraction of the Cavallo redistribution scheme). Below, we present several cases where α^* is higher.

	α^*	$\frac{n-m-1}{n}$
$n = 10, m = 2$	0.733333	0.7
$n = 50, m = 2$	0.999999	0.94
$n = 50, m = 4$	0.996028	0.9
$n = 100, m = 2$	1.000000 (rounded up)	0.97
$n = 100, m = 4$	0.999999	0.95

¹⁰The solution of the linear program is not necessarily an optimal assignment of the c_j , because, for example, the individual rationality constraint is ensured by enforcing a sufficient constraint, which usually is stronger.

References

- Apt, K.; Conitzer, V.; Guo, M.; and Markakis, E. 2008. Welfare undominated Groves mechanisms. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, 426–437.
- Bailey, M. J. 1997. The demand revealing process: to distribute the surplus. *Public Choice* 91:107–126.
- Cavallo, R. 2006. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 882–889.
- Conitzer, V., and Sandholm, T. 2002. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 103–110.
- de Clippel, G.; Naroditskiy, V.; and Greenwald, A. 2009. Destroy to save. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 207–214.
- Gujar, S., and Narahari, Y. 2008. Redistribution of VCG payments in assignment of heterogeneous objects. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, 438–445.
- Gujar, S., and Narahari, Y. 2009. Redistribution mechanisms for the assignment of heterogeneous objects. In *FAAMAS*.
- Gul, F., and Stacchetti, E. 1999. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory* 87:95–124.
- Guo, M., and Conitzer, V. 2008a. Better redistribution with inefficient allocation in multi-unit auctions with unit demand. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 210–219.
- Guo, M., and Conitzer, V. 2008b. Undominated VCG redistribution mechanisms. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 1039–1046.
- Guo, M., and Conitzer, V. 2009. Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior* 67(1):69–98.
- Guo, M., and Conitzer, V. 2010a. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1676–1679. NECTAR track.
- Guo, M., and Conitzer, V. 2010b. Optimal-in-expectation redistribution mechanisms. *Artificial Intelligence* 174(5-6):363–381.
- Kelso, A., and Crawford, V. 1982. Job matching, coalition formation, and gross substitutes. *Econometrica* 50:1483–1504.
- Lehmann, B.; Lehmann, D.; and Nisan, N. 2003. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*.
- Likhodedov, A., and Sandholm, T. 2004. Mechanism for optimally trading off revenue and efficiency in multi-unit auctions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*.
- Likhodedov, A., and Sandholm, T. 2005. Approximating revenue-maximizing combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.
- Moulin, H. 2009. Almost budget-balanced VCG mechanisms to assign multiple objects. *Journal of Economic Theory* 144(1):96–119.
- Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2004. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior* 46(1):174–188.