Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Increasing VCG Revenue by Decreasing the Quality of Items

Mingyu Guo
mingyu.guo@adelaide.edu.au
School of Computer Science
University of Adelaide, Australia

Argyrios Deligkas and **Rahul Savani**
{A.Deligkas, rahul.savani}@liverpool.ac.uk
Department of Computer Science
University of Liverpool, UK

Abstract

The VCG mechanism is the standard method to incentivize bidders in combinatorial auctions to bid truthfully. Under the VCG mechanism, the auctioneer can sometimes increase revenue by "burning" items. We study this phenomenon in a setting where items are described by a number of attributes. The value of an attribute corresponds to a quality level, and bidders' valuations are non-decreasing in the quality levels. In addition to burning items, we allow the auctioneer to present some of the attributes as lower quality than they actually are. We consider the following two revenue maximization problems under VCG: finding an optimal way to mark down items by reducing their quality levels, and finding an optimal set of items to burn. We study the effect of the following parameters on the computational complexity of these two problems: the number of attributes, the number of quality levels per attribute, and the complexity of the bidders' valuation functions. Bidders have unit demand, so VCG's outcome can be computed in polynomial time, and the valuation functions we consider are step functions that are non-decreasing with the quality levels. We prove that both problems are NP-hard even in the following three simple settings: a) four attributes, arbitrarily many quality levels per attribute, and single-step valuation functions, b) arbitrarily many attributes, two quality levels per attribute, and single-step valuation functions, and c) one attribute, arbitrarily many quality-levels, and multi-step valuation functions. For the case where items have only one attribute, and every bidder has a single-step valuation (zero below some quality threshold), we show that both problems can be solved in polynomial-time using a dynamic programming approach. For this case, we also quantify how much better marking down is than item burning, and we compare the revenue of both approaches with computational experiments.

Introduction

Combinatorial auctions allow agents to bid on bundles of items. They are a key paradigm for resource allocation in multiagent systems. The *Vickrey-Clarke-Groves* (VCG) mechanism (Vickrey 1961; Clarke 1971; Groves 1973) is the canonical method that is used to incentivize agents to bid truthfully. While VCG maximizes social welfare, it is well-known that it can be deficient with respect to revenue (Conitzer and Sandholm 2006; Rothkopf 2007).

It is easy to see that the auctioneer can increase the revenue of VCG by destroying ("burning") items. A closely related idea is the deliberate damage of goods in order to create price discrimination. Deneckere and McAfee (1996) observed this phenomenon in practice, and studied several theoretical settings where it increases social welfare. They observed that some companies choose to remove some functions from goods in order to achieve higher revenue. We investigate a similar option that auctioneers have, to represent items as a lower quality than they actually are.

In our setting, items are described by a number of attributes. The value of an attribute can be thought of as a quality level. The quality levels induce a partial order over items. As an example, consider a cloud computing services provider, who offers a number of different specifications for virtual machines to clients. Each virtual machine has a number of characteristics like the number of cores, speed, memory, and storage space. Clients have (different) valuation functions that depend on their requirements and the characteristics of virtual machines, though it is reasonable to assume that all of them value better characteristics.

We explore marking down the quality of items, and the special case of completely destroying items, from an auctioneer's perspective. Formally, we study the following two revenue maximization problems under VCG: finding an optimal way to mark down items by reducing their quality levels, and finding an optimal set of items to burn. We study the effect of the following parameters on the computational complexity of these two problems: the number of attributes, the number of quality levels per attribute, and the complexity of the bidders' valuation functions. We provide positive and negative results. The negative results hold even when bidders have unit demand (in which case the allocation and payments of VCG can be computed in polynomial time).

The valuation functions we consider are step functions that are non-decreasing with the quality levels. Every agent has its own valuation function. As a special case, we consider *single-step* valuation functions, where there is a fixed value for any item whose quality is above a certain threshold, where this threshold comprises minimum quality levels for every attribute, and for all items that do no meet this threshold, the value is zero.

We prove that both problems are NP-hard even in the following three simple settings:

a) four attributes, arbitrarily many quality levels per attribute, and single-step valuation functions,

b) arbitrarily many attributes, two quality levels per attribute (low and high), and single-step valuation functions, and

c) one attribute, arbitrarily many quality-levels, and multi-step valuation functions.

For the case that items have only one attribute, and every bidder has a single-step valuation, we show that both problems can be solved in polynomial-time using a dynamic programming approach. We also quantify how much better marking down is than item burning, and compare the revenue of both approaches empirically.

Other related work. Several authors have studied how the auctioneer can increase her revenue using the information asymmetry by revealing only partial information about the items, using signaling (Emek et al. 2012; Miltersen and Sheffet 2012; Guo and Deligkas 2013). Both item burning and marking down the quality of items, can be seen as another way the auctioneer can take advantage of the information asymmetry between her and the bidders. Other approaches have been studied in order to increase the revenue in combinatorial auctions: bundling items together (Palfrey 1983; Ghosh, Nazerzadeh, and Sundararajan 2007)[1], generalizing the VCG to virtual valuation combinatorial auctions (VVCAs), affine maximizer combinatorial auctions (Likhodedov and Sandholm 2004; 2005), and mixed bundling auctions with reserve prices (Tang and Sandholm 2012). Another way for the auctioneer to increase the revenue of VCG is by removing bidders from the auction, as noted by Ausubel and Milgrom (2011) and Rastegari, Condon, and Leyton-Brown (2011).

The idea of burning items in VCG auctions has been studied from other perspectives. For example, item burning and money burning with payments redistribution have been used to increase the social surplus for the bidders (de Clippel, Naroditskiy, and Greenwald 2009; Guo and Conitzer 2008).

Model Description

We study the allocation of m items among n agents using the VCG mechanism. Every item is characterized by k numerical attributes that represent its quality. We use $(q_1, q_2, \dots, q_k)$ to denote an item whose i-th quality attribute equals q_i. Higher attribute value corresponds to higher quality. We define the following partial order over the items: Given two items $(q_1, q_2, \dots, q_k)$ and $(q'_1, q'_2, \dots, q'_k)$, if $q_i \geq q'_i$ for all i and $q_i > q'_i$ for some i, then we say the first item is “better”. Wlog, we assume the attribute values are non-negative integers. An agent’s valuation for an item only depends on the item’s quality. That is, if two items share the same quality attributes, then an agent’s valuations for them are the same. Given two items, if they are comparable based on the above partial order, then every agent’s valuation for the better item is at least as high. We use $v_i(q_1, q_2, \dots, q_k)$ to denote agent i’s valuation for an item with quality $(q_1, q_2, \dots, q_k)$. We assume every agent’s valuation for the worst quality equals 0, i.e., $v_i(0, 0, \dots, 0) = 0$ for all i. A large portion of our paper deals with a restricted agent valuation model. Agent i is called a *simple agent* if her valuation function is a single-step function: there exists a threshold $(q_1, q_2, \dots, q_k)$ and a value $c > 0$ such that

$$v_i(q'_1, q'_2, \dots, q'_k) = \begin{cases} c & \forall j, q'_j \geq q_j \\ 0 & \text{Otherwise} \end{cases}$$

We will use $c|(q_1, q_2, \dots, q_k)$ to denote (the valuation of) such an agent. We assume the agents have unit demand, which implies that VCG’s outcome can be computed in polynomial time.

In this paper we focus on the VCG mechanism in the full information setting, where the auctioneer knows agents’ valuation functions. The auctioneer has the ability to burn items (remove some items from the auction), and mark down items (present some item attributes as lower quality than they actually are). Marking down is more general than burning: an item with quality $(q_1, q_2, \dots, q_k)$ can be marked down to $(0, 0, \dots, 0)$, which is equivalent to burning this item.

The following example shows that both burning and marking down may increase VCG revenue, and marking down may result in higher revenue than mere burning.

Example 1. For example, suppose there is only one attribute ($k = 1$), thus the quality vector can be written as a single value. There are two items, both with quality 2, which is the highest quality. There are three agents. Agent 1 is a general agent, whose valuation function is as follows: $v_1(0) = 0; v_1(1) = 1; v_1(2) = 100$. Agent 2 and 3 are both simple agents. Agent 2’s valuation is $200|2$ (her valuation is 200 if the quality is at least 2, and her valuation is 0 otherwise). Agent 3’s valuation is $2|1$ (her valuation is 2 if the quality is at least 1, and her valuation is 0 otherwise).

- Original VCG revenue: Agent 1 and 2 each wins one item and pays 2. The original VCG revenue is 4.
- Optimal burning: The auctioneer burns one copy of the items. As a result, agent 2 becomes the only winner. He pays 100, and the VCG revenue increases to 100.
- Optimal marking down: Instead of burning, the auctioneer marks down one copy of the items to quality 1. As a result, agent 2 wins an item of quality 2 and agent 3 wins an item of quality 1. Agent 2 pays 100 and agent 3 pays 1. That is, after marking down, the VCG revenue increases to 101.

Given items’ qualities and agents’ valuations, we study the following revenue maximization problems under VCG:

- Optimal burning: find an optimal set of items to burn.
- Optimal marking down: find an optimal way to mark down items by reducing their quality levels.

We show that both are NP-hard in restricted settings. We give complete proofs for two cases to illustrate our different types of reduction (Theorems 1 and 4). The details of other reductions are omitted due to space constraints. For our polynomial-time algorithms for the single-attribute case, we give complete details and proofs.

[1] See Kroer and Sandholm (2013) for a detailed study of bundling and VCG.

General Valuation Functions

In this section, we show that if we allow multi-step valuation functions (non-simple agents), then both optimal burning and optimal marking down are NP-hard. This holds even if there is only one quality attribute. The results are based on reductions from the NP-hard *minimum dominating set* problem (Garey and Johnson 1979): given a graph, find a smallest subset of nodes so that every node not in the subset is adjacent to at least one node in the subset.

Theorem 1. *For general agents, optimal burning is NP-hard. This holds even if items have only one attribute.*

Proof. We consider an arbitrary graph with z nodes. We construct the following auction scenario. The numbers of agents, items, and possible quality vectors are all polynomial in z. The items are characterized by only one quality attribute. For the constructed scenario, the problem of optimal burning is equivalent to the problem of finding a minimum dominating set in the graph. This suffices to show that optimal burning is NP-hard.

We create $2z$ items. Item i has quality i (i from 1 to $2z$).

We first create an agent whose valuation function v is as follows. This agent is out bid by the other agents. The purpose of constructing him is to ensure that item i's VCG price is at least $v(i)$.

$$v(i) = 4^{i-1}\epsilon \quad 1 \le i \le z \tag{1}$$

$$v(z+i) = i \quad 1 \le i \le z \tag{2}$$

We then create the following z agents a_1 to a_z with valuations according to (1) and (2) except that agent a_i's value for the i th and $(z+i)$th items are:

$$v_{a_i}(i) = 3(4^{i-1}\epsilon)$$
$$v_{a_i}(z+i) = i + 4^{i-1}\epsilon + \beta$$

Let $i \leftrightarrow j$ denote that node i and j are adjacent in the graph. We create z agents b_1 to b_z with valuations according to (1) and (2) except that agent b_i's value for the $(z+i)$th item and any items $(z+j)$ s.t. $i \leftrightarrow j$ are:

$$v_{b_i}(z+i) = i + 4^{i-1}\epsilon + \frac{\beta}{2}$$
$$v_{b_i}(z+j) = j + \beta \quad \forall j \text{ satisfying } i \leftrightarrow j$$

ϵ and β ($0 < \beta < \epsilon$) are set to be small enough so that all the above valuation functions are increasing.

Under VCG, for $1 \le i \le z$, a_i wins item i and b_i wins item $z+i$. The agents' total payment is $\sum_{1 \le i \le 2z} v(i)$, which is above $1 + 2 + \ldots + z$.

We can set ϵ small enough, so that if we burn any item between $z+1$ to $2z$, the agents' total valuation is smaller than $1 + 2 + \ldots + z$. We do so, which ensures that we never want to burn any item between $z+1$ and $2z$.

The problem is then to choose a subset from items 1 to z to burn. Let S be the subset. Let $\bar{S}$ be $\{1, 2, \ldots, z\} - S$. We analyze the revenue change after S is burned. We lose $\sum_{i \in S} v(i)$, which was the revenue for selling the burned items. For item i in $\bar{S}$, there is no change in terms of revenue. It is still won by a_i, who still pays $v(i)$. For item $z+i$ with $i \in S$, a_i now wins it instead of b_i. The VCG price changes from $v(z+i)$ to $v_{b_i}(z+i)$. For these items, the total increase in revenue is $\sum_{i \in S}(4^{i-1}\epsilon + \frac{\beta}{2}) = \sum_{i \in S} v(i) + |S|\frac{\beta}{2}$. Combining all revenue changes so far, we have gained $|S|\frac{\beta}{2}$ by burning. Finally, for items $z+i$ with $i \notin S$, b_i still wins it. If there exists $j \in S$ with $i \leftrightarrow j$, then b_j is a loser and b_j's valuation for $z+i$ is $i+\beta$. Hence, b_i pays i if there is no such b_j and pays $i+\beta$ otherwise. That is, b_i pays β in extra if $i \leftrightarrow j$ for some $j \in S$. In summary, for every item burned, we gain $\frac{\beta}{2}$ extra revenue, and for every item not burned and adjacent to a burned item (the corresponding nodes are adjacent in the graph), we gain β extra revenue. Under the optimal burning scheme, the set of items burned S must correspond to a minimum dominating set of the graph. □

Theorem 2. *For general agents, optimal marking down is NP-hard. This holds even if items have only one attribute.*

Given the above NP-hardness results, from now on we focus only on simple agents.

Multi-Attribute Setting

In this section, we show that even if we only consider simple agents, and there are multiple attributes (*e.g.*, at least 4 attributes), then both optimal burning and optimal marking down are NP-hard. The results are based on reductions from the NP-hard *monotone one-in-three 3SAT* problem (Schaefer 1978). Monotone means that the literals are just variables, never negations. One-in-three means that the determination problem is to see whether there is an assignment so that for each clause, exactly one literal is true.

Theorem 3. *For simple agents, optimal burning is NP-hard. This holds if the number of quality attributes is at least* 4.

Theorem 4. *For simple agents, optimal marking down is NP-hard. This holds if the number of quality attributes is at least* 4.

Proof. We consider an arbitrary monotone one-in-three 3SAT instance with z variables. We construct the following auction scenario. The numbers of agents, items, and possible quality vectors are all polynomial in z. The items are characterized by 4 quality attributes. For the constructed scenario, the problem of optimal marking down is equivalent to the problem of finding a satisfying assignment of the monotone one-in-three 3SAT instance. This suffices to show that optimal marking down is NP-hard.

Let the variables be x_1 to x_z. Let c be the number of clauses. c is less than z^3 if we ignore duplicate clauses.

We construct the following items. For $1 \le i \le z$ and $1 \le j \le c$, if x_i appears in clause j, then we create one copy of $(i, z-i, j, c-j)$ (denoted by item A_{ij}). For $1 \le i \le z$, we create one copy of $(i, z-i, 0, 0)$ (denoted by item B_i).

We construct the following agents. We use z_i to denote the number of appearances of x_i in the 3SAT instance. For $1 \le i \le z$, we create $z_i + 2$ agents. We name this set of agents S_i. The quality requirement for all is $(i, z-i, 0, 0)$ and their valuations are $1, 1, \frac{1}{2}, \ldots, \frac{1}{z_i}, \frac{1+z_i\epsilon}{z_i+1}$, respectively. We choose δ that is smaller than $\frac{1}{z}$. We set $\epsilon < \delta$, and make sure that $\frac{1}{z_i} > \frac{1+z_i\epsilon}{z_i+1}$. For $1 \leq j \leq c$, we create two agents both with type $\delta|(0,0,j,c-j)$.

Let us first analyze the VCG revenue before marking down. Given i, the total number of items better than or equal to $(i, z-i, 0, 0)$ is z_i+1. They are won by the highest z_i+1 agents in S_i. Everyone pays $\frac{1+z_i\epsilon}{z_i+1}$. The total revenue from these items is then $1+z_i\epsilon$. Summing over all i, the total VCG revenue is $z+3c\epsilon$.

There is no point marking down any of the B_i. Given i, if we mark down any attribute of B_i, then no agent wants the item. As a result, there is one less winner among S_i, which never increases the revenue. Let the variables in clause j be i_1, i_2, and i_3. There are three items better than $(0,0,j,c-j)$, which are A_{i_1j}, A_{i_2j}, and A_{i_3j}. We should mark down exactly one of them to $(0,0,j,c-j)$, which leads to δ extra revenue (only two agents bid δ for this quality). At most we gain $c\delta$ extra revenue (δ for each clause).

We assume we marked down k items ($k \leq c$) and gained $k\delta$ extra revenue. One thing we have not yet mentioned is that marking down comes at a cost. Given i, if any A_{ij} is marked down, then we lose $z_i\epsilon$ revenue. On the other hand, given i, if all of A_{ij} have been marked down, then we still just lose $z_i\epsilon$ revenue. That is, at the minimum, we lose ϵ for every item marked down. The net gain is then $k(\delta-\epsilon)$. This expression is maximized when $k=c$. That is, the maximum revenue gain by marking down is $c(\delta-\epsilon)$. This maximum revenue is achievable if the marking down scheme satisfies that 1) for every clause j, we mark down exactly one A_{ij}; 2) for every i, we either mark down all of A_{ij} or mark down none. Thus, the maximum revenue is achievable if and only if the 3SAT instance is satisfiable. □

The above NP-hardness proofs are based on scenarios with 4 attributes. By modifying the scenarios slightly, we may derive another set of NP-hardness results. That is, even if all quality attributes are *binary* (an attribute value is 0 or 1), if the number of attributes is linear in $log(nm)$, then both optimal burning and optimal marking down are NP-hard.

Corollary 1. *For simple agents and binary attributes, optimal burning is NP-hard.*

Corollary 2. *For simple agents and binary attributes, optimal marking down is NP-hard.*

The Single-Attribute Setting

In this section, we focus on simple agents and the case of one quality attribute. In this setting, both optimal burning and optimal marking down can be solved in polynomial-time using a dynamic programming approach. We then quantify how much better marking down is than item burning in this setting, and conclude with computational experiments.

For single-attribute settings, every pair of items are comparable. Let H be the best item's quality. That is, the items' qualities are all from 0 to H. Here, it is wlog to assume $H \leq m$ ($H = m$ if all items have different qualities). We use $(m_1, m_2, \ldots, m_H)$ to denote the set of all items, where m_i is the number of items of quality i.

Here, an agent's valuation function is characterized by an valuation and a quality threshold. The winner determination problem under VCG can be solved using the following greedy algorithm. We loop from quality 1 to H (worst to best). In the i-th round, we consider all items of quality i. We allocate the items to the highest unallocated agents who accept quality i (whose thresholds are at most i).

```
Start with an empty queue $Q = \{\}$;
for $i$ from $1$ to $H$ do
    $Q := Q + S_i$, where $S_i$ is the set of agents whose
    thresholds are equal to $i$;
    Each of the $m_i$ highest agents in $Q$ wins an item of
    quality $i$ (or every agent in $Q$ wins if $|Q| \leq m_i$);
    Remove all winners from $Q$;
end
```

Algorithm 1: Winner Determination

For presentation purpose, we assume that all agents' valuations are different. That is, given two different agents $v|q$ and $v'|q'$, we must have $v \neq v'$. This assumption is not restrictive for the following reasons. Given a fixed set of items, the VCG revenue is continuous in the agents' valuations. The optimal VCG revenue after burning / marking down is therefore also continuous in the agents' valuations (it is the maximum over a finite number of ways to burn / mark down). Given a set of agents, if there exist ties among the valuations, then we may simply perturb the valuations infinitesimally to get rid of ties. We solve the optimal burning / marking down problem by working on the perturbed valuations. The resulting optimal revenue is infinitesimally close to the original optimal revenue. Therefore, the corresponding burning / marking down scheme is infinitesimally close to optimality.

Throughout the section, we will stick to the VCG outcome produced by the above winner determination algorithm, which is unique given the no-tie assumption. We introduce the following notation, which will be used frequently in this section. Let W_i be the set of agents allocated in the round i. Let Q_i be the set of losers after round i (some of them may win in later rounds). Let $Top(Q_i)$ be the agent with the highest valuation in Q_i. If Q_i is empty, then we define $Top(Q_i)$ to be the dummy agent $0|0$. We use $v_{Top(Q_i)}$ to denote $Top(Q_i)$'s valuation. An agent is called a loser if he does not get allocated at the end of the algorithm.

Lemma 1. *If $Top(Q_i)$ does not win, then every agent in W_i pays $v_{Top(Q_i)}$.*

Proof. Let a be an agent in W_i. If a does not exist, then $Top(Q_i)$ wins an item in round i. The other winners stay the same with or without a. Therefore, every agent in W_i pays the value of $Top(Q_i)$. □

Lemma 2. *If $Top(Q_i)$ wins (which can only happen in rounds later than i), then every agent in W_i has the same payment as $Top(Q_i)$.*

Proof. We use $U^*(A, S)$ to represent the optimal social welfare when we allocate items in A to agents in S. Let a be an agent in W_i and let v_a be his valuation. We use $\{Item \leq x\}$ to represent all items of quality up to x. Similarly, we use $\{Agent \leq x\}$ to represent all agents of thresholds up to x. a's VCG payment equals

$$\begin{aligned} &U^*(\{Item \leq i\}, \{Agent \leq i\}) - v_a + v_{Top(Q_i)} \\ &+U^*(\{Item > i\}, \{Agent > i\} \cup Q_i - Top(Q_i)) \\ &-U^*(\{Item \leq i\}, \{Agent \leq i\}) + v_a \\ &-U^*(\{Item > i\}, \{Agent > i\} \cup Q_i) \\ &= v_{Top(Q_i)} + U^*(\{Item > i\}, \{Agent > i\} \cup Q_i - Top(Q_i)) \\ &-U^*(\{Item > i\}, \{Agent > i\} \cup Q_i). \end{aligned}$$

$Top(Q_i)$'s VCG payment can be simplified to the same expression. □

The above lemmas show that items of the same quality share the same VCG price. Next, we show that the VCG price is nondecreasing in the quality.

Lemma 3. *Agents in W_i pay $v_{Top(Q_j)}$, where j is the minimum value satisfying $j \geq i$ and $Top(Q_j)$ is a loser.*

Lemma 3 directly follows from Lemma 1 and Lemma 2.

Proposition 1. *The prices of the items are nondecreasing in the qualities. That is, if $i > i'$, then agents in W_i pay the same or more, compared to agents in $W_{i'}$.*

Proof. By Lemma 3, we have $j \geq j'$, agents in W_i pay $v_{Top(Q_j)}$ and agents in $W_{i'}$ pay $v_{Top(Q_{j'})}$. Since both $Top(Q_j)$ and $Top(Q_{j'})$ are losers, we have Q_j is a superset of $Q_{j'}$. Therefore, $v_{Top(Q_j)} \geq v_{Top(Q_{j'})}$. □

Now we are ready to describe a dynamic programming approach for solving the optimal burning and the optimal marking down problems. Due to space constraint, we will only present the solution of the optimal marking down problem. The solution of the optimal burning problem is similar.

We define $MD(\bar{t}, \bar{p}, \bar{d})$ to be the optimal VCG revenue after marking down, under the following extra conditions:

- Only agents whose thresholds are at least $\bar{t}$ are allowed in the auction. The other agents are considered removed. $\bar{t}$ is from 1 to H. The function returns 0 if $\bar{t} = H + 1$.
- Based on Lemma 3, we know that the VCG price of an item must be either equal to a certain agent's valuation, or the price is 0. Here, we require that the VCG price of any item is at least equal to the $\bar{p}$-th highest valuation from the agents. $\bar{p}$ is from 1 to $n + 1$. (We define the $(n + 1)$-th highest valuation to be 0.) The function returns 0 if $\bar{p} = 1$.
- Only items with the $\bar{d}$ highest quality are in the auction. The other items are considered burned. $\bar{d}$ is from 1 to m. The function returns 0 if $\bar{d} = 0$.

$MD(1, n + 1, m)$ is then the optimal VCG revenue after marking down.

The original item vector is $(m_1, m_2, \ldots, m_H)$ (there are m_i items of quality i). Let the new item vector after optimal marking down be $(m'_1, m'_2, \ldots, m'_H)$ (after marking down, there are m'_i items of quality i). It should be noted that $\sum m'_i$ may be smaller than $\sum m_i$, since some items may have been burned (marked down to quality 0).

Suppose that after marking down, the minimum price of any item is equal to the p-th highest valuation from the agents. Let v be this valuation. Let t be the highest quality of the item whose VCG price is v. For now, we assume that we know v, p, and t.

By Proposition 1, we have that all items with quality up to t must have price v. The revenue gained by selling items with quality up to t is then $v \sum_{1 \leq i \leq t} m'_i$. We recall that Q_t is the set of unallocated agents after round t (here we are talking about the winner determination problem involving the marked down items). By Lemma 2, $Top(Q_t)$ must be a loser. If $Top(Q_t)$ is a winner, then he wins an item of quality higher than t, and he pays the same price v, which contradicts with our assumptions. Since $Top(Q_t)$ is a loser, by Lemma 1, we have $v_{Top(Q_t)} = v$. This implies that from round 1 to t, the agents allocated are exactly the set of agents whose valuations are higher than v (such an agent must be allocated before the end of round t, otherwise he should be placed higher than $Top(Q_t)$; also, no agents whose valuations are lower than v can be allocated, as such an agent cannot afford to pay v under VCG). As a result, the revenue gained by selling items with quality up to t can be rewritten as $vC(t, v)$, where $C(t, v)$ is the number of agents whose valuations are higher than v and whose thresholds are at most t. Given t and v, this part of the revenue can be easily calculated.

We then consider the revenue gained by selling items with quality higher than t. The set of items under discussion are $(0, \ldots, 0, m'_{t+1}, \ldots, m'_H)$. The total number is then $d = \sum_{t+1 \leq i \leq H} m'_i$. These d items can be interpreted as the results of marking down, from the items with the d highest original quality. It is wlog to assume that the item with the i-th highest quality (counting duplicates) among the marked down items is marked down from the item with the i-th highest quality among the original items. Furthermore, since for items with quality higher than t, the VCG price is by assumption higher than v, we have that the VCG payments of the items with quality higher than t are only determined by agents whose thresholds are at least $t+1$ (all items with quality higher than t are sold at prices higher than v, so all agents in Q_t do not win and their valuations are too low to effect the VCG payments). Therefore, the revenue gained by selling items with quality higher than t is $MD(t + 1, p - 1, d)$.

The overall revenue after marking down is then $vC(t, v) + MD(t + 1, p - 1, d)$. We notice that $MD(t + 1, p - 1, d)$ is nondecreasing in d. We want to maximize d. We recall that the items with the d highest original quality are marked down to $(0, \ldots, 0, m'_{t+1}, \ldots, m'_H)$, and the items with the $m - d$ lowest original quality are marked down to $(m'_1, \ldots, m'_t, 0, \ldots, 0)$. We need to ensure that it is possible to mark down the items with the $m - d$ lowest original quality to $(m'_1, \ldots, m'_t, 0, \ldots, 0)$, so that the prices for items with quality up to t are all v, and the number of winners is exactly $C(t, v)$. Let x_i $(1 \leq i \leq t)$ be the number of agents whose valuations are greater than v and whose thresholds are equal to i. Let g be the threshold of $Top(Q_t)$. Let h be the smallest i so that x_i is not empty. By Lemma 1 and Lemma 2, for all $h \leq i \leq t$, $Top(Q_i)$ is either a winner or the same as $Top(Q_t)$. As a result, for $h \leq i < g$ (if such i exists), we need to ensure that $Top(Q_i)$ is a winner. For $\max\{g, h\} \leq i < t$, $Top(Q_i)$ may be the same as

$Top(Q_t)$. Finally, the number of winners and the number of items should match. Mathematically, the above translates to

$$\begin{array}{ll} m'_i = 0 & 1 \le i < h \\ \sum_{1\le j\le i} m'_j < \sum_{1\le j\le i} x_j & h \le i < g \\ \sum_{1\le j\le i} m'_j \le \sum_{1\le j\le i} x_j & \max\{g,h\} \le i < t \\ \sum_{1\le j\le i} m'_j = \sum_{1\le j\le i} x_j & i = t \end{array}$$

We need to mark down the items with the $m - d$ lowest original quality to $(m'_1, \dots, m'_t, 0, \dots, 0)$, so that the above inequalities hold. Meanwhile, we want to minimize $m - d$ (maximize d). Algorithm 2 shows how this can be done.

```
Let $m'_i = 0$ for $i < h$;
for $i$ from $h$ to $g - 1$ do
    if $\sum_{1\le j<i} m'_j + m_i < \sum_{1\le j\le i} x_j$ then
        Let $m'_i = m_i$;
    else
        Let $m'_i = \sum_{1\le j\le i} x_j - 1 - \sum_{1\le j<i} m'_j$;
    end
end
for $i$ from $\max\{g,h\}$ to $t - 1$ do
    if $\sum_{1\le j<i} m'_j + m_i \le \sum_{1\le j\le i} x_j$ then
        Let $m'_i = m_i$;
    else
        Let $m'_i = \sum_{1\le j\le i} x_j - \sum_{1\le j<i} m'_j$;
    end
end
Assign $m'_t$ a value so that $\sum_{1\le j\le t} m'_j = \sum_{1\le j\le t} x_j$;
```

Algorithm 2: Marking Down to Maximize d

If $m'_i = m_i$, then we do not mark down any items of quality i. If $m'_i < m_i$, then we need to burn some of the items of quality i. If $m'_i > m_i$ (this can only happen when $i = t$), we need to mark down $m'_t - m_t$ items of higher quality to quality t. The last part determines the maximum value of d:[2]

$$d^* = \begin{cases} \sum_{t+1\le i\le n} m_i - (m'_t - m_t) & m'_t > m_t \\ \sum_{t+1\le i\le n} m_i & m'_t \le m_t \end{cases}$$

Combining the two parts of revenue, the optimal revenue after marking down can be calculated as follows:

$$MD(1, n+1, m) = vC(t, v) + MD(t+1, p-1, d^*)$$

The above is on the basis that we know the values of t, p, and v. For actual calculation, we need to maximize over all possible values of t and p (v is determined by p). Actual calculation is done as follows.

$$MD(\bar{t}, \bar{p}, \bar{d}) =$$

$$\max_{\bar{t}\le t\le H, 1\le p\le \bar{p}} \{vC(t,v) + MD(t+1, p-1, d^*) + F(t,p)\}$$

Here, we also added an extra term $F(t,p)$, which serves as a sanity check. That is, $F(t,p) = 0$ if the pair of t and p is a valid pair. Otherwise, $F(t,p) = -\infty$. t and p form a valid pair if it is possible to mark down the items so that the minimum VCG price is equal to the p-th highest valuation from the agents, and the highest quality of the cheapest item is t. F checks three things: 1) whether the agent with the p-th highest valuation has a threshold that is less than or equal to t; 2) whether there exists at least one agent who can afford the p-th highest valuation and has a threshold that is less than or equal to t; and 3) whether d^* is non-negative.

The number of possible parameter sets for MD is polynomial in m and n (we recall that $H \le m$). Given a parameter set, MD's value is a maximum over a polynomial number of choices. The calculation of d^* and F both take polynomial time. Therefore, the overall process takes polynomial time.

We then quantify the ratio between the optimal VCG revenue after burning and the optimal VCG revenue after marking down.[3]

Theorem 5. *For simple agents and single-attribute settings, the ratio between the optimal VCG revenue after burning and the optimal VCG revenue after marking down is between $1/H$ and 1, where H is the highest possible quality. The above bounds are tight.*

The above Theorem only says that in extreme cases, marking down can lead to H times more revenue. The following numerical experiment suggests that *on average*, the revenue after marking down is only slightly higher than the revenue after burning. On the other hand, both marking down and burning significantly improves upon the original VCG. The experimental setup is as follows: $m = H = 10$. The items' qualities and the agents' quality requirements are drawn *i.i.d.* from $\{1, 2, \dots, H\}$. The agents' valuations are drawn *i.i.d.* from $U(0,1)$. We average over 100 cases. The ratio between the optimal revenue after marking down and the original VCG revenue is 14.912 ($n = 5$), 3.228 ($n = 10$), 1.170 ($n = 20$), and 1.029 ($n = 40$). The ratio between the optimal revenue after burning and the original VCG revenue is 14.901 ($n = 5$), 3.208 ($n = 10$), 1.152 ($n = 20$), and 1.018 ($n = 40$). It is expected that when n/m becomes large, marking down and burning become less effective. For large n/m, generally every item is highly contested, making marking down and burning undesirable.

Open problems

An open problem is to decide if the optimal marking down/burning problems are tractable for the cases with simple agents and items with two or three attributes. It seems that neither our reduction nor our algorithm can be extended to these cases, thus new techniques should be proposed in order to answer these questions. Another interesting direction is to study the case with single attribute items and simple agents under the Bayesian setting.

[2] If we only allow burning, then d^* is always just $\sum_{t+1\le i\le n} m_i$. We still run Algorithm 2 to determine the values of the m'_i, as it mostly just involves burning. If $m'_t > m_t$, then m'_t cannot be achieved by burning. So for burning, the sanity check function F also needs to ensure $m'_t \le m_t$.

[3] As a comparison, the worst-case ratio between the original VCG revenue and the revenue after burning (marking down) is 0. There exist scenarios where the original VCG revenue is 0, and after burning (marking down), the VCG revenue becomes positive.

Acknowledgments

When conducting this work, the second author was supported by Microsoft Research through its PhD Scholarship Programme.

References

Ausubel, L. M., and Milgrom, P. 2006. The lovely but lonely vickrey auction. In *Combinatorial Auctions, chapter 1*. MIT Press.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* 11(1).

Conitzer, V., and Sandholm, T. 2006. Failures of the vcg mechanism in combinatorial auctions and exchanges. In *AAMAS*, 521–528.

de Clippel, G.; Naroditskiy, V.; and Greenwald, A. 2009. Destroy to save. In *ACM Conference on Electronic Commerce*, 207–214.

Deneckere, R. J., and McAfee, P. R. 1996. Damaged goods. *Journal of Economics and Management Strategy* 5(2):149–174.

Emek, Y.; Feldman, M.; Gamzu, I.; Leme, R. P.; and Tennenholtz, M. 2012. Signaling schemes for revenue maximization. In *ACM Conference on Electronic Commerce*, 514–531.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Ghosh, A.; Nazerzadeh, H.; and Sundararajan, M. 2007. Computing optimal bundles for sponsored search. In *WINE*, 576–583.

Groves, T. 1973. Incentives in teams. *Econometrica* 41(4):617–31.

Guo, M., and Conitzer, V. 2008. Better redistribution with inefficient allocation in multi-unit auctions with unit demand. In *ACM Conference on Electronic Commerce*, 210–219.

Guo, M., and Deligkas, A. 2013. Revenue maximization via hiding item attributes. In *IJCAI*.

Kroer, C., and Sandholm, T. 2013. Computational bundling for auctions. Technical Report CMU-CS-13-111, Carnegie Mellon University, School of Computer Science.

Likhodedov, A., and Sandholm, T. 2004. Methods for boosting revenue in combinatorial auctions. In *AAAI*, 232–237.

Likhodedov, A., and Sandholm, T. 2005. Approximating revenue-maximizing combinatorial auctions. In *AAAI*, 267–274.

Miltersen, P. B., and Sheffet, O. 2012. Send mixed signals: earn more, work less. In *ACM Conference on Electronic Commerce*, 234–247.

Palfrey, T. R. 1983. Bundling decisions by a multiproduct monopolist with incomplete information. *Econometrica* 51(2):pp. 463–483.

Rastegari, B.; Condon, A.; and Leyton-Brown, K. 2011. Revenue monotonicity in deterministic, dominant-strategy combinatorial auctions. *Artificial Intelligence* 175(2):441 – 456.

Rothkopf, M. H. 2007. Thirteen reasons why the vickrey-clarke-groves process is not practical. *Oper. Res.* 55(2):191–197.

Schaefer, T. J. 1978. The complexity of satisfiability problems. In *STOC*, 216–226.

Tang, P., and Sandholm, T. 2012. Mixed-bundling auctions with reserve prices. In *AAMAS*, 729–736.

Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16(1):8–37.