Honor Among Thieves — Collusion in Multi-Unit Auctions

Yoram Bachrach
Microsoft Research
Cambridge UK
yobach@microsoft.com

ABSTRACT

We consider collusion in multi-unit auctions where the allocation and payments are determined using the VCG mechanism. We show how collusion can increase the utility of the colluders, characterize the optimal collusion and show it can easily be computed in polynomial time. We then analyze the colluders' coalition from a cooperative game theoretic perspective. We show that the collusion game is a convex game, so it always has a non-empty core, which contains the Shapley value. We show how to find core imputations and compute the Shapley value, and thus show that in this setting the colluders can always share the gain from their manipulation in a stable and fair way. This shows that this domain is extremely vulnerable to collusion.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Collusion, Cooperative Game Theory, Bid Rigging, Shapley Value, Core, Weber Set

1. INTRODUCTION

Collusion is an agreement between two or more agents to limit competition by manipulating or defrauding in order to obtain an unfair advantage [26]. Such manipulations include agreements to divide the market, set prices or limit production or bids. For example, in oligopoly where there are few firms producing a certain good, the decision of a few firms to limit production can significantly affect the market as a whole. Cartels are a special case of explicit collusion, where firms coordinate prices. Collusion which is not overt is called *tacit collusion*. Certain types of collusion are illegal in many countries, due to competition law. However, many types of tacit collusion are hard to detect. This paper studies one form of collusion called bid rigging, where participants in an auction change their bids in an attempt to lower prices. Specifically, we consider the domain of multi-unit auctions under the Vickrey-Clarke-Groves (VCG) mechanism [1].

In a multi-unit domain there are multiple *identical* items to be allocated. Each agent thus only cares about the *number* of items she receives. We assume free disposal, so agents always value obtaining more items at least as much as obtaining less items. Agents can thus express their preferences as a valuation function mapping the number of items they obtain to their valuation for this quantity of items.

Given the above valuation functions, a central mechanism can easily allocate the items in a way that maximizes social welfare. The center can ask agents to specify the maximal amount they would be willing to pay to obtain various quantities of items. These specifications are the agents' bids.

One major problem that plagues such domains is the fact that the valuation functions are private information of the agents. Agents may bid strategically, by misreporting their valuation functions in order to achieve a better outcome for themselves. The VCG payment scheme is the canonical method for incentivizing the agents to bid truthfully (i.e. reveal their true valuation function).

Although VCG has many desirable properties, it is known to be susceptible to collusion. Although any single agent is incentivised to truthfully reveal her valuation function, *several* agents may agree to misreport their valuations in a *coordinated* way, and split the gains from this manipulation.

In this paper we show how agents can collude in VCG multi-unit auctions. We provide a simple polynomial algorithm for finding the optimal way to collude, given a specific coalition of colluders. We then analyze how the colluders can split the gains from such a manipulation using cooperative game theory, by modeling the situation as a coalitional game we call the *collusion game*. We show that this game is a convex game, so it has a non-empty core that contains the Shapley value. This means the coalition of colluders can form *stable* agreements, in a way that guarantees no bickering among the colluders. In fact, the colluders can even split the gains in a fair manner, based on the contribution of each colluder to the coalition. We also provide algorithms to compute these colluder utility distributions. These disturbing results indicate that VCG multi-unit domains are

Cite as: Honor Among Thieves — Collusion in Multi-Unit Auctions, Yoram Bachrach, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 617-624

[1]See Section 5 and [21] for more information regarding the origins and properties of the VCG mechanism.

extremely vulnerable to collusion.

The paper proceeds as follows. Section 2 contains some preliminary definitions and notation. Section 3 discusses possible ways of collusion in multi-unit VCG auctions. In Section 4 we analyze the colluders coalition using tools from cooperative game theory. Section 5 contains references for important related work. We conclude in Section 6.

2. PRELIMINARIES

We consider collusion in multi-unit auction under the VCG mechanism. We begin with a brief review of VCG.

We have a set N of n agents, $\{1, 2, \dots, \}$. The mechanism needs to choose one of a set of possible alternatives K. Each agent reports a type $\theta_i \in \Theta_i$, representing the agent's preferences over the different alternatives in K. Each agent has a different valuation of the mechanism's chosen alternative $k \in K$, $v_i(k, \theta_i)$. The mechanism chooses the outcome according to a choice rule $k : \Theta_1 \times ... \times \Theta_n \to K$. Each agent is also required to make a payment p_i to the mechanism. The mechanism chooses the payment of each agent according to a payment rule $t_i : \Theta_1 \times ... \times \Theta_n \to \mathbb{R}$. If the agents have quasi-linear utility functions, then the agents have utility $u_i(k, p_i, \theta_i) = v_i(k, \theta_i) - p_i$. An agent might not report her true type, but has to always choose a type to report to the mechanism. Thus, agent i reports a type $\theta'_i = s_i(\theta_i)$, according to its own strategy s_i. In Groves mechanisms, the mechanism's choice rule given the reported types $\theta' = (\theta'_1, ..., \theta'_n)$ maximizes the sum of the agents' utilities, according to their reported types, as seen in Equation 1.

Equation 1 Groves Allocation

$$k^*(\theta') = arg \max_{k \in K} \sum_i v_i(k, \theta'_i) \qquad (1)$$

The payment rule in Groves mechanisms is given in Equation 2. where $h_i : \Theta_{-i} \to \mathbb{R}$ may be any function that only

Equation 2 Groves Payments

$$t_i(\theta') = h_i(\theta'_{-i}) - \sum_{j \neq i} v_j(k^*, \theta'_j) \qquad (2)$$

depends on the reported types of agents other than i. A special case of Groves mechanisms is that of the VCG mechanism, given in Equation 3.

Equation 3 VCG Payments

$$h_i(\theta'_{-i}) = \sum_{j \neq i} v_j(k^*_{-i}(\theta'_{-i}), \theta'_j) \qquad (3)$$

Our analysis of collusion is based on coalitional game theory. A transferable utility coalitional game is composed of a set of n agents, N, and a characteristic function mapping any subset (coalition) of the agents to a real value $v : 2^N \to \mathbb{R}$, indicating the total utility these agents achieve together. We denote the set of all the agents except i as $N_{-i} = N \setminus \{i\}$. A coalitional game is *increasing* if for all coalitions $C' \subset C$ we have $v(C') \leq v(C)$, and is *super-additive* when for all *disjoint* coalitions $A, B \subset N$ we have $v(A) + v(B) \leq v(A \cup B)$. In super-additive games, it is always worthwhile for two subcoalitions to merge, so eventually the *grand coalition* containing all the agents will form.

The characteristic function only defines the gains a coalition can achieve, but does not define how these gains are distributed among the agents who formed the coalition. An *imputation* $(p_1, \dots, p_n)$ is a division of the gains of the grand coalition among the agents, where $p_i \in \mathbb{R}$, such that $\sum_{i=1}^n p_i = v(N)$. We call p_i the payoff of agent i, and denote the payoff of a coalition C as $p(C) = \sum_{i \in C} p_i$. Obviously, a key issue is choosing the appropriate imputation for the game. Cooperative Game theory offers several answers to this question.

A basic requirement for a good imputation is *individual rationality*, which states that for any agent $i \in N$, we have that $p_i \geq v(\{i\})$—otherwise, some agent is incentivized to work alone. Similarly, we say a coalition B *blocks* the payoff vector $(p_1, \dots, p_n)$ if $p(B) < v(B)$, since the members of B can split from the original coalition, derive the gains of $v(B)$ in the game, give each member $i \in B$ its previous gains p_i—and still some utility remains, so each member can get more utility. If a blocked payoff vector is chosen, the coalition is somewhat unstable. The most prominent solution concept focusing on such stability is that of the core [13].

Definition 1. *The core of a coalitional game is the set of all imputations $(p_1, \dots, p_n)$ that are not blocked by any coalition, so that for any coalition C, we have the following equation: $p(C) \geq v(C)$.*

Another solution concept is the Shapley value [23] which defines a *single* value division. The Shapley value focuses on *fairness*, rather than stability. The Shapley value fulfills several important fairness axioms [23, 29] and has been used to fairly share gains or costs. The Shapley value of an agent depends on its marginal contribution to possible coalition permutations. We denote by π a permutation (ordering) of the agents, and by Π the set of all possible such permutations. Given permutation $\pi \in \Pi = (i_1, \dots, i_n)$, the marginal worth vector $m^\pi[v] \in \mathbb{R}^n$ is defined as $m^\pi_{i_1} = v(\{i_1\})$ and for $k > 1$ as $m^\pi_{i_k}[v] = v(\{i_1, i_2, \dots, i_k\}) - v(\{i_1, i_2, \dots, i_{k-1}\})$. The convex hull of all the marginal vectors is called the *Weber Set*. Weber showed [28] that the Weber set of any game contains its core. The Shapley value is the centroid of the marginal vectors.

Definition 2. *The Shapley value is the payoff vector:*

$$\phi[v] = \frac{1}{n!} \sum_{\pi \in \Pi} m^\pi[v]$$

A specific type of super-additive games are convex games.

Definition 3. *A game is convex if for any $A, B \subseteq I$ we have $v(A \cup B) \geq v(A) + v(B) - v(A \cap B)$.*

For convex games it is known [24, 15] that the core is always non-empty, and that the Weber Set is identical to the core. The Shapley value is a convex combination of the marginal vectors, so it lies in the Weber Set. Thus, in convex games, the Shapley value lies in the core.

3. COLLUSION IN VCG MULTI-UNIT AUCTIONS

Consider a multi-unit auction, where the auctioneer offers to sell t identical items to n bidders, $\{1, \dots, n\}$. Each bidder i has a certain valuation to any number of items she receives, given by a function $v_i : \mathbb{N} \cup \{0\}) \to \mathbb{R}$ which maps the number of units to the utility to that bidder. We denote the marginal value of the j'th item to agent i as $m_i(j) = v_i(j) - v_i(j-1)$. A typical assumption that we also adopt is that of free disposal, so for any $k > j$ we have $v_i(k) > v_i(j)$. Another typical assumption is that $v_i(0) = 0$. Throughout this work we will assume that the *marginal utility of items for each bidder is decreasing*, so for any agent i and number of items j, $v_i(j+2) - v_i(j+1) \leq v_i(j+1) - v_i(j)$. This assumption is critical in our analysis.

Since all the items are identical, it is possible to define an allocation of items to agents in terms of the number of items each user receives. Thus, an allocation is simply vector of quantities $q = (q_1, \dots, q_n)$ such that $\sum_{i=1}^{n} q_i = t$. Given an allocation q, we denote the total utility the allocation q generates as $f(q) = \sum_{i=1}^{n} v_i(q_i)$. It is easy to see that the total utility can be expressed as the following sum of marginal utilities: $f(q) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} m_i(j)$.

3.1 VCG in Multi-Unit Auctions

Given the functions $v_1, \dots, v_n$ of the bidders, it is possible to compute the optimal allocation of items to the bidders, that maximizes $f(q)$. However, the v_i functions are typically private information of the bidders. Typically in such allocation settings, in order to make sure the bidders truthfully reveal their preferences, the Vickrey-Clark-Groves mechanism is used. As discussed in Section 2, the VCG prices make truthful revelation the dominant strategy for the agents, which results in the optimal allocation of the items. Any mechanism for allocating the items must receive as input the agents' valuation functions $v_1, \dots, v_n$. A possible representation for a function is the list of valuations $v_i(1), \dots, v_i(t)$, and another equivalent representation is the list of marginal values of this function $m_i(1), \dots, m_i(t)$. Since there are only t items to allocate, there is no need to provide valuations for more than t items. Before we discuss collusion, we first describe how the optimal allocation can be computed. We consider the greedy algorithm that assigns an item to the agent with the current highest marginal utility for an item.

Algorithm 1 VCG Allocation

1. For $i = 1$ to n do $l_i \leftarrow 1$ (Initialize locations)
2. For $i = 1$ to n do $q_i \leftarrow 0$ (Initialize quantities)
3. For $j = 1$ to t do:
   (a) $x \leftarrow argmax_i m_i(l_i)$ (Next highest)
   (b) $l_x \leftarrow l_x + 1$ (Move to next marginal)
   (c) $q_x \leftarrow q_x + 1$ (Assign item)

Proposition 1. *The greedy assignment algorithm results in the optimal assignment, maximizing $f(q)$.*

Proof. We note that given an optimal allocation q_x for x items, due to the diminishing marginal utility functions, assigning an additional item to the highest marginal agent yields an optimal allocation q_{x+1} for $x+1$ items. The optimal allocation for a single item is assigning it to the agent with the highest marginal. A simple induction on the number of assigned items completes the proof. □

It is also possible to express the above algorithm in the following way. The algorithm obtains a list of marginal utilities for each of the agent. It then sorts these lists, from highest to lowest, but keeps track of the agent from which the marginal originated from. It then takes the first t items in the sorted lists, and assigns items to the agents from which these marginals originated from.

We now consider a VCG based mechanism for allocating the t items. Such a mechanism receives the functions v_i, reported by each of the agents, which allows computing the marginal utility functions m_i for the agents. Each agent i has the marginals list $m_i = (m_i(1), \dots, m_i(t))$. Many of the results in this paper rely on manipulation of marginal lists, so we introduce some notation. Given two marginal lists, $m_i = (m_i(1), \dots, m_i(t))$ and $m_j = (m_j(1), \dots, m_j(t))$ we denote the concatenated list

$$(m_i, m_j) = (m_i(1), \dots, m_i(t), m_j(1), \dots, m_j(t))$$

Typically, we sort the marginal lists from the highest to lowest. Given a marginal list m, we denote the sorted list as m_s. Given two marginal lists m_i, m_j, we can concatenate them and sort the concatenated list to obtain $(m_i, m_j)_s$. Since all orderings of the elements in the original lists result in the same sorted concatenated list, we can simply denote $(m_i, m_j)_s = (m_i \cup m_j)_s$. Given a subset C of agents, we denote the sorted set of marginals of all the agents as $m_s^C = (\cup_{i \in C} m_i)_s$ (i.e. the marginal list generated by concatenating all the marginal lists and sorting the resulting list). We denote the sorted list or marginals of all the users *except* user j as $m_s^{-j} = (\cup_{i \in C \setminus \{j\}} m_i)_s$. Slightly abusing notation, we extend this to coalitions, denoting the sorted marginals of all the agents that are *not* in C as $m_s^{-C} = (\cup_{i \notin C} m_i)_s$

Observation 1 (Alternative Greedy Algorithm). *An alternative presentation of the greedy algorithm is as follows. It obtains the marginals list m_i for each user i, computes m^N, sorts it into to m_s^N, and uses the first t marginals. Thus, if $m_s^N = (a_1, a_2, \dots, a_{nt})$, the social welfare obtained by the optimal allocation of t items to the users N is $\sum_{i=1}^{t} a_i$. We now compute the VCG payments using Equations 2 and 3. To compute the allocation, the algorithm maintains the* origin *agent of each marginal, and assigns a unit to the first t such origins.*

Observation 2 (Computing VCG Prices). *Consider the case where agent i obtained q_i items in the optimal allocation for all the agents. First consider the optimal assignment when agent i is not present, as required by Equation 3. Denote sorted list of marginals for all the users except i as $m_s^{-i} = (b_1, b_2, \dots b_{(n-1)t})$. We note that the optimal allocation for $N \setminus \{i\}$ has a social welfare of $\sum_{i=1}^{t} b_i$. Now consider the optimal assignment when agent i is present, as required by Equation 2. When agent i is present, the algorithm uses the full marginals list $m_s^N = (a_1, a_2, \dots, a_{nt})$. Since user i was allocated q_i items, the $t - q_i$ first items in m_s^{-i} also occur in the first t items in m_s^N, and the next q_i items do not occur in m_s^N. We refer to these q_i items that do not occur in the first t items of m_s^N as the* payment marginals *for i.*

The VCG payment for i is the difference between Equation 2 and Equation 3, and is thus simply q_i marginals in m_s^{-i} starting from position $t - q_i$.

EXAMPLE 1 (VCG PRICES). *Consider $m_1 = (10, 8, 3)$, $m_2 = (9, 7, 5)$, $m_3 = (6, 4, 2)$, and $t = 5$. We compute the VCG payment of agent 1.*

In example 1 we have $m_s^N = (10, 9, 8, 7, 6, 5, 4, 3, 2)$, and $m_s^{-1} = (9, 7, 6, 5, 4, 2)$. Thus in the optimal allocation for $N \setminus \{1\}$ the welfare of $N \setminus \{1\}$ is $9+7+6+5+4$, and in the optimal allocation for N the welfare of $N \setminus \{1\}$ is $9+7+6$. User 1 obtains $q_1 = 2$ items, so the VCG payment p_1 for her can be computed by considering m_s^{-1}, skipping the first $t - q_1 = 5 - 2 = 3$ items, and summing the next q_1 items, resulting in a payment of $5 + 4 = 9$.

3.2 How to Collude In VCG Multi-Unit Auctions

We begin with an example for collusion in a VCG Multi-Unit Auctions.

EXAMPLE 2 (SIMPLE COLLUSION SCHEME). *Consider 4 agents, with the following marginals: $m_1 = (10, 8, 8)$, $m_2 = (9, 1, 1)$, $m_3 = (9, 1, 1)$, $m_4 = (1, 1, 1)$, and $t = 3$. The optimal allocation assigns one item to each of agents 1, 2, 3. Agent 1 gets an item of valuation $v_1 = 10$ and a payment of $p_1 = 1$, so her utility is $u_1 = 10 - 1 = 9$. Similarly, $v_2 = 9, p_2 = 8, u_2 = 9 - 8 = 1$ and $v_3 = 9, p_3 = 8, u_3 = 9 - 8 = 1$.*

Note that the payment agents 2 and 3 make is due to the second marginal of agent 1, $m_1(2) = 8$, and that the payment agent 1 makes is due to the second marginal of agent 2, $m_2(2) = 1$. Assume agents 1, 2, 3 trust each other and know each other's marginals, and consider the case where they collude. They can misreport their marginals the following way: $m'_1 = (10, 0, 0), m'_2(9, 0, 0), m'_3 = (9, 0, 0)$.

In this case, agent 4 is not a part of the colluders' coalition, and truthfully reports her marginals: $m'_4 = (1, 1, 1)$. Note that under these declarations of the agents, the allocation does not change, and each of agents 1, 2, 3 get one item. However, the VCG payments are very different: $p'_1 = 1, p'_2 = 1, p'_3 = 1$. Thus $u'_1 = 10-1 = 9, u'_2 = 9-1 = 8, u'_3 = 9-1 = 8$, so agent 1 obtains exactly the same utility as before, and agents 2, 3 each increase their utility from 1 to 8.

One might claim that since agent 1 did not increase her utility, she might not be willing to collude. However, agents 2, 3 can *compensate* agent 1 via a monetary transfer [2]. Using such transfers allow the colluders to cooperate and share the spoils in various ways.

We denote by q_i^* the quantity of items agent i obtains under truthful revelation. In example 2 each colluder keeps her first q_i^* original marginals (for which they obtain items under truthful decelerations) and sets the reminder marginals to 0. We refer to this as the *simple collusion scheme*. While such a manipulation scheme never harms the colluders, we now show that sometimes a more sophisticated manipulation is required, where both monetary and item transfers are necessary.

EXAMPLE 3 (STRONGER COLLUSION SCHEME). *Consider $m_1 = (8, 1, 1, 1)$, $m_2 = (9, 1, 1, 1)$, $m_3 = (10, 4, 3, 2)$, where $t = 4$ and agents $C = \{1, 2\}$ consider colluding. Under truthful revelation, agents 1 and 2 get a single item each, and agent 3 gets two items, and the payments are $p_1 = 3, p_2 = 3, p_3 = 2$, and thus the colluder coalition C gets $1+1 = 2$ items, and pays $p(C) = 3+3 = 6$. Under the simple collusion scheme of example 2, the following marginals can be declared: $m'_1 = (8, 0, 0, 0)$, $m'_2(9, 0, 0, 0)$, $m'_3 = m_3 = (10, 4, 3, 2)$, resulting with the same allocation, with different payments $p'_1 = 3, p'_2 = 3, p'_3 = 0$. Thus, the coalition C gets 2 items, and pays $p'(C) = 3 + 3 = 6$, just as before, so this manipulation does not benefit the colluders (although the non-colluding member does benefit from it). Now consider the manipulation where the colluders designate agent 1 as a proxy who attempts to obtain the items for* all *the colluders. In this case a possible manipulation is declaring the following marginals: $m'_1 = (9, 8, 0, 0)$, $m'_2(0, 0, 0, 0)$, $m'_3 = m_3 = (10, 4, 3, 2)$.*

The above manipulation results in agent 1 getting 2 items, agent 2 getting 0 items and agent 3 getting 2 items. As a whole, the colluders $C = \{1, 2\}$ obtains $2 + 0 = 2$ items, the same as $1 + 1 = 2$ items it got under truthful revelation. The payments change: $p'_1 = 3+2 = 5, p'_2 = 0, p'_3 = 0$, so the total payment of the colluders is $p'(C) = 5 + 0 = 5$.

Thus using this manipulation, the coalition obtains exactly the same number of items it gets under truthful revelation, but with lower payments. All the items are allocated to the proxy agent, who also makes all the payment. However, it is of course possible for the colluders to reallocate the items among themselves exactly as under truthful revelation, and since their total payment has dropped, they can use monetary transfers so that each colluder pays slightly less than it would under truthful revelation. In this example, the coalition of colluders has to make both item transfers and monetary transfers to increase the utility.

We now explicitly define the above manipulation scheme by presenting an algorithm to compute it, and then show that this scheme is indeed optimal for the colluders.

The above manipulation operates by designating a single agent from the coalition of colluders as the proxy agent, and making sure the proxy agent obtains the number of items allocated to the coalition under truthful revelation. All the marginals of the non-proxy colluders are set to 0. We denote the colluders $C = \{1, 2, \ldots, r\}$ with marginal lists $m_1, \ldots, m_r$, and the non-colluders $\{r+1, r+2, \ldots, n\}$ with marginals $m_{r+1}, \ldots, m_n$. Each marginal list m_i has t values $m_i(1), \ldots, m_i(t)$. We continue to denote the list of all the marginals of an agent subset X as m^X and this list sorted as m_s^X. We denote the marginal functions reported by the colluders as $m^{C'} = (m'_1, \ldots, m'_r)$. The following algorithm computes the manipulation.

THEOREM 2 (OPTIMAL COLLUSION). *Algorithm 2 computes the optimal manipulation for the colluder coalition C.*

PROOF. Denote q_i^* the quantity of items a colluder $i \in C$ obtains under truthful revelation. We call the manipulation scheme of Example 2, where each colluder keeps her first q_i^* original marginals and sets the reminder marginals to 0, the simple collusion scheme. It is easy to see that the simple collusion scheme never harms the colluders, as they obtain the same quantities of items for smaller payments. We also note that Algorithm 2 obtains the same quantity of items for the

[2]VCG relies on such monetary transfers, so obviously we are already operating in a quasi-linear preferences domain anyway.

Algorithm 2 Manipulation algorithm

1. Compute q_C, the number of items allocated to the members of C under truthful revelation.
2. Compute m_s^C the sorted marginals list for the entire colluder coalition
3. Designate agent 1 as the proxy agent, and construct her marginal list as the first q_C values of m_s^C and 0 for any marginal beyond that point: for any $i \leq q_C$ have $m_1(i) = m_s^C(i)$, and for any $i > q_C$ have $m_1(i) = 0$.
4. The marginals for all the non-proxy colluders $j \in \{2, 3, \dots, c\}$ are always 0: $m_j(l) = 0$ for any l.

coalition q_C. We now note that the payments under the manipulation of Algorithm 2 are at most the payments of the simple collusion scheme. Consider merging two marginal declarations $m'_x = (m_x(1), \dots, m_x(q_x^*), 0, 0, \dots)$ and $m'_y = (m_y(1), \dots, m_y(q_y^*), 0, 0, \dots)$ by sorting the following vector $v = m_x(1), \dots, m_x(q_x^*), m_y(1), \dots, m_y(q_y^*)$ into v_s and declaring $m''_x = (v_s(1), \dots, v_s(q_x^* + q_y^*), 0, \dots)$ and $m''_y = (0, 0, \dots)$. Similarly to Example 1 under m'_x, m'_y, agent x would have to pay q_x^* marginals of m_s^{-x} starting at location $t - q_x^*$, and y would have to pay q_y^* marginals of m_s^{-y} starting at location $t - q_y^*$. Under m''_x, m''_y agent y pays nothing, and x pays $q_x^* + q_y^*$ marginals of $m_s^{-\{x,y\}}$ starting at location $t - q_x^* - q_y^*$. We note that both options include the same *number* of marginals, but the second option includes smaller marginals, so agents x, y are better off using m''_x, m''_y, which obtains them the same items for a lower cost (as a coalition). Applying the same argument repeatedly shows that the manipulation of Algorithm 2 is never worse to the colluders than the simle collusion scheme.

Note that m_s^C indicates the utility the coalition of colluders as a whole obtains from obtaining any quantity q_C of items. This utility for the coalition is obtained when they allocate the items optimally, according to the greedy allocation algorithm. A coalition of colluders can either have more than one agent who declares non-zero marginals, or have a single agent who declares non-zero marginals. Due to the above argument, it is always useful to merge two marginals list to a single agent, so the optimal strategy for the colluders is to have only *one* agent who declares non-zero marginals.

Consider a *single* agent with a marginals list m_s^C, entering a multi-unit VCG auction. Due to the VCG truthfulness, a dominant strategy for this agent is to truthfully declare m_s^C. Since it is never better for such an agent to "split" its marginals across several false identities, the optimal strategy for the colluders is choosing a single proxy agent. However, the optimal strategy for a single proxy agent is truthful revelation, so the optimal strategy for the colluders is using a single proxy agent who declares m_s^C. Algorithm 2 uses exactly this manipulation, so it computes the optimal manipulation, that maximizes the total utility of the colluders $\sum_{i \in C} u_i$. Since running the VCG auction requiers polynomial time, the algorithm runs in polynomial time. □

Algorithm 2 provides a polynomial time algorithm to compute a manipulation for the colluders, and Theorem 2 shows it is an optimal manipulation in terms of the sum of utilities of the colluders. However, after performing the manipulation, all of the items are owned by the proxy agent, who also makes all the payments [3]. The colluders must distribute the items and make inter-coalitional monetary payments. In other words, the collusion allows the coalition to generate more utility *as a group*, and the colluders must still determine how to allocate this utility. The following section models this situation as a cooperative game, and analyzes the possible outcomes.

4. THE COLLUSION GAME

Consider a multi-unit auction of t items, with n agents. We examine a certain subset $C \subseteq N$, who may decide to collude. Under truthful revelation the VCG mechanism results in the allocation $q^t = (q_1^t, \dots, q_n^t)$ [4], and payments $p_1^t, \dots, p_n^t$, so the coalition C as a whole obtains a certain utility:

$$u^t(C) = \sum_{i \in C} u_i = \sum_{i \in C} v_i(q_i^t) - p_i^t$$

If the agents in C decide to collude, they can form a coalition and use either the simple manipulation scheme of example 2 or the *optimal manipulation* of Algorithm 2, changing the allocation to $q^* = (q_1^*, \dots, q_n^*)$ and the payments to $p_1^*, \dots, p_n^*$. The simple manipulation scheme does not change the allocation, so $q_i^* = q_i$ for any i, but reduces the payments to coalition members so $p_i^* \leq p_i^t$ for any $i \in C$. Under the optimal collusion, both the allocation and payments may change. As seen in section 3.2 and Theorem 2, under optimal collusion, the total number of items C receives does not change, i.e $q^t(C) = q^*(C)$ where $q^t(C) = \sum_{i \in C} q_i^t$ and $q^*(C) = \sum_{i \in C} q_i^*$. However, the total payment made by the coalition drops, i.e. $p^t(C) \geq p^*(C)$ where $p^t(C) = \sum_{i \in C} p_i^t$ and $p^*(C) = \sum_{i \in C} p_i^*$. As seen in example 3, after performing the manipulation, the proxy agent distributes the items similarly to their distribution under the truthful declarations, which results in each colluder i obtaining the same amout of items they would get under truthfull reports, q_i^t. The proxy agent is the only colluding agent who has a non-zero payment to the VCG mechanism. The colluders must then make intra-coalition monetary transfers, which determine their utility. Thus, the coalition, *as a whole*, generates the following utility to its members

$$u^*(C) = \sum_{i \in C} v_i(q_i) - \sum_{i \in C} p_i^*$$

We now define a coalitional game, based on the total utility a coalition of colluders generates to its members.

Definition 4 (The Collusion Game.). *Given a VCG multi-unit auction of t items for agents $N = \{1, 2, \dots, n\}$ with marginal functions $m_1, \dots, m_n$, we define the value $v(C)$ of a coalition $C \subseteq N$ as [5]:*

$$v(C) = u^*(C)$$

[3]Proxies in collusion are related to false-name proofness. In domains where a single proxy agent does worse than the agents it represents, a single agent with marginals similar to the proxy can do better by splitting up its marginals across false identities.

[4]The subscript t stands for truthful.

[5]In this definition v maps a coalition of colluders to the utility they achieve, so v denotes the characteristic function, not to be confused with $v_i(q_i)$ which is a valuation of a certain number of items.

Since the colluders must trust each other to perform the manipulation, the coalition C might be restricted to only a certain subset of the agents [6].

Once the coalitional game is defined, we can focus on determining how the total utility generated by the colluders can be shared amongst the members. We note that monetary transfers allow any distribution of the utility among the members, regardless of how the items are allocated. Coalitional game theory focuses on notions such as fairness and stability. At a first glance, one might hope that the above game would have an empty core. If the core is empty, no stable coalition of colluders can form, and the colluders would endlessly bicker regarding the monetary transfers.

Unfortunately, we show that the above game always has a non-empty core, and that a simple polynomial algorithm allows finding a stable utility distribution (core imputation). Furthermore, regarding *fairness*, we show that the Shapley value, which is widely considered a "fair" solution, is also in the core. Thus, the colluders can share their gains from the manipulation in a stable and fair manner [7], making collusion a very significant problem in such auctions. Our results are based on showing the above defined game is convex.

THEOREM 3 (CONVEXITY OF THE COLLUSION GAME). *The collusion game defined above is convex, for any multi-unit auction domain.*

PROOF. We first note that by definition the collusion game is increasing (monotone), so if $C' \subseteq C$ then $v(C') = u^*(C') \leq u^*(C) = v(C)$. For increasing games, it is known that the game is convex (under Definition 3) iff for any $C' \subseteq C$ and any agent j we have $v(C' \cup \{j\}) - v(C') \leq v(C \cup \{j\}) - v(C)$. This means that a game is convex iff an agent adds to the utility of a coalition at least as much as it adds to a coalition contained in it. We show that this condition holds for the collusion game. The definition of the collusion game defines the value of a coalition is the utility it achieves under the optimal manipulation $v(C) = u^*(C)$.

We now show convexity when the collusion game is defined using the optimal manipulation. We now decompose the value agent j adds to a coalition C, $\Delta_i^C = v(C \cup \{j\}) - v(C)$. We denote the amount paid by agent i when the agents in coalition M optimally collude as p_i^{M*}. By Definition 4 of the collusion game above,

$$\begin{aligned}
\Delta_i^C &= \\
&= v(C \cup \{i\}) - v(C) \\
&= u^*(C \cup \{i\}) - u^*(C) \\
&= \sum_{i \in C \cup \{j\}} v_i(q_i) - \sum_{i \in C \cup \{j\}} p_i^{C \cup \{j\}*} - \sum_{i \in C} v_i(q_i) + \sum_{i \in C} p_i^{C*} \\
&= v_j(q_j) + (\sum_{i \in C} p_i^{C*} - \sum_{i \in C \cup \{j\}} p_i^{C \cup \{j\}*})
\end{aligned}$$

To show that $\Delta_j^{C'} \leq \Delta_j^C$ we thus must show that $v_j(q_j) + (\sum_{i \in C'} p_i^{C'*} - \sum_{i \in C' \cup \{j\}} p_i^{C' \cup \{j\}*}) \leq v_j(q_j) + (\sum_{i \in C} p_i^{C*} - \sum_{i \in C \cup \{j\}} p_i^{C \cup \{j\}*})$

[6] In the unlikely case where *all* the agents trust each other enough to collude, they obtain all the items for a zero cost.
[7] We mean "fair" for the colluders, of course. Collusion is very *unfair* for the auctioneer. Hence the title "Honor among thieves".

An equivalent condition that we can show is the following condition:

$$(\sum_{i \in C'} p_i^{C'*} - \sum_{i \in C' \cup \{j\}} p_i^{C' \cup \{j\}*}) \leq (\sum_{i \in C} p_i^{C*} - \sum_{i \in C \cup \{j\}} p_i^{C \cup \{j\}*})$$

Yet another equivalent condition is that:

$$\sum_{i \in C' \cup \{j\}} p_i^{C' \cup \{j\}*}) - \sum_{i \in C'} p_i^{C'*} \geq \sum_{i \in C \cup \{j\}} p_i^{C \cup \{j\}*}) - \sum_{i \in C} p_i^{C*}$$

In other words, we must show that the payment C' incurs for adding j (under the optimal manipulation), is greater than the payment C incurs for adding j (again, under the optimal manipulation). Due to theorem 2, under the optimal manipulation only the proxy actually has a non-zero payment.

Thus, $\sum_{i \in C'} p_i^{C'*}$ simply contains the single payment of the proxy agent $p_x^{C'*}$, and other zero payments. The proxy obtains $q^t(C')$ items, so according to Observation 2 $p_{C'}^x$ is the sum of $q^t(C')$ elements from $m_s^{-C'}$ starting from position $t - q^t(C')$. We note the set of all agents N is composed of C', D, agent i and the non-colluders $H = N \setminus C \cup \{i\}$ (the honest agents). The first $t - q^t(C')$ marginals of $m_s^{-C'}$ contain the first $q^t(i)$ elements of m_s^i, the first $q^t(H)$ elements of m_s^H and the first $q^t(D)$ elements of m_s^D. Given a set $X \subseteq N$, we can consider the *remaining* marginals, for which no items were allocated, denoted w_s^X where $w_s^X(j) = m_s^X(j + q^t(X))$. Similarly to the notation for marginals m_s^X, we denote the concatenaed and sorted remaining marginal lists as $w_s^C = (\cup_{i \in C} w^i)_s$. Using this notation, we have the payment of the proxy of C', $p_{C'}^x$, is simply the sum of the first $q^t(C')$ marginals in $(w^D \cup w^i \cup w^H)_s$. Similarly, when i joins the colluder coalition C', the payment of the proxy agent for coalition $C' \cup \{i\}$ is the sum of the first $q^t(C' \cup \{i\}) = q^t(C) + q_i^t$ marginals in $(w^D \cup w^H)_s$. The same analysis for $C = C' \cup D$ and $C \cup \{i\}$ shows that p_C^x is the sum of the first $q^t(C) = q^t(C) + q^t(D)$ marginals in $(w^i \cup w^H)_s$, and that $p_{C \cup \{i\}}^x$ is the sum of the first $q^t(C \cup \{i\}) = q^t(C) + q^t(D) + q_i^t$ marginals in w_s^H.

Note w_s^H is a vector than can be derived by taking some elements in $(w^i \cup w^H)_s$ and erasing some elements (those of w_s^i), and replacing them with zero marginals appearing on the tail. Similarly, $(w^i \cup w^H)_s$ can be derived by "sparsing out" $(w^D \cup w^i \cup w^H)_s$ etc. Given the definitions of the above payments, we can see that $p_{C' \cup \{i\}}^x - p_{C'}^x \geq p_{C \cup \{i\}}^x - p_C^x$, as the right hand side sums marginals for a sparser vector. Since $p_{C' \cup \{i\}}^x - p_{C'}^x \geq p_{C \cup \{i\}}^x - p_C^x$ we have $\sum_{i \in C' \cup \{j\}} p_i^{C' \cup \{j\}*}) - \sum_{i \in C'} p_i^{C'*} \geq \sum_{i \in C \cup \{j\}} p_i^{C \cup \{j\}*}) - \sum_{i \in C} p_i^{C*}$ for any agent j, so the collusion game is convex. □

The convexity of the collusion game has several important implications regarding the ways in which the coalition members can share the utility they derive from the collusion. The collusion causes the prices paid to significantly drop. Item transfers and monetary transfers allow the colluders to share the excess utility in any way they desire, and convexity guarantees there exist *stable* utility distributions. Under unstable utility distributions, the colluders' coalition is likely to disintegrate due to arguments among the colluders. However, convexity guarantees the colluders could find a way to distribute the gains so no subset of the colluders would benefit from leaving the colluders' coalition.

The colluders may also want to share the excess utility in a fair manner, using the Shapley value. A general game might not even have one stable payoff division, as the core may be empty. Even if there do exist stable allocations, the Shapley value may be an unstable allocation. Unfortunately, in the multi-unit VCG auction, a stable allocation for the colluders always exists, and the fair allocation is also stable.

COROLLARY 1. *The collusion game always has a non-empty core, which contains the Shapley value.*

PROOF. Due to Theorem 3, the collusion game is convex. As discussed in Section 2, convex games have a non-empty core, which coincides with the Weber set. The Shapley value is in the Weber set, and thus lies in the core. □

One final barrier that may make it harder to collude is computational complexity. Although Corollary 1 guarantees the colluders a fair and stable allocation, it might be hard to compute. We show that the colluders can always find a stable allocation in polynomial time. We also show that when all the marginal functions are identical, except for a bounded number of agents, the Shapley value can also be computed in polynomial time.

THEOREM 4. *Computing an imputation in the core of the collusion game can be done in polynomial time.*

PROOF. Given a permutation of the agents $\pi = (i_1, \ldots, i_n)$, we show the marginal contribution vector $m^\pi(v)$ (see definitions in Section 2) can be computed in polynomial time. Due to Algorithms 1 and 2 we can compute, in polynomial time, the optimal collusion manipulation for any coalition C, the VCG payments under the collusion, and the the value $v(C) = u^*(C)$ of any coalition in the collusion game.

Thus given π we can easily compute the marginal contribution vector $m^\pi_{i_1} = v(\{i_1\})$ and $m^\pi_{i_k}(v) = v(\{i_1, i_2, \ldots, i_k\}) - v(\{i_1, i_2, \ldots, i_{k-1}\})$ for any $k > 1$. The Weber set is, by definition, the convex hull of the marginal contribution vectors $m^\pi(v)$ (for any $\pi \in \Pi$). Thus, any such marginal contribution vector is in the Weber set.

The core in the collusion game, as a convex game, coincides with the Weber set, so any such marginal contribution vector is in the core. Thus, we only need to arbitrarily select a permutation π, compute $m^\pi(v)$, and the result would be an imputation in the core. □

We note that any permutation π results in a marginal contribution vector that is in the core. However, there is an exponential number of such possible vectors, one for each permutation $\pi \in \Pi$.

The Shapley value is the centroid of this exponentially size set of vectors, so the naive algorithm for computing it requires exponential time. Consider a multi-unit auction with agents N, where all of the marginal functions m_i are identical (so $m_i = m^*$), except for a *constant* number b of agents, $a_1, \ldots, a_b$ who have a different marginal function. We call such a domain a b-bounded collusion domain.

THEOREM 5. *In a b-bounded collusion domain, the Shapley value can be computed in polynomial time, and testing whether a given imputation is in the core can also be performed in polynomial time.*

PROOF. We show that for b bounded collusion domains, there is a polynomially bounded number of *different* marginal contribution vectors. In such domains, all agents except $B = \{a_1, \ldots, a_b\}$ have the same marginal function (i.e. all the agents in $N \setminus B$ have the same marginal function). Any two permutations of the agents that differ only in the ordering of the agents in $N \setminus B$ result in the same marginal contribution vector. Thus there are only $b!\binom{n}{b} = \frac{n!}{(n-b)!} = (n-b+1) \cdot (n-b+2) \ldots \cdot n \leq n^b$ different permutations to consider (we first choose where the non-standard agents are placed, and then order them within these locations).

Thus there are less than n^b distinct marginal contribution vectors, so all the different marginal contribution vectors can be computed in polynomial time. Each such "representative" vector occurs an equal number of times, so averaging all these vectors gives the Shapley value.

Given an imputation $p = (p_1, \ldots, p_n)$ we can test whether it is in the core by checking if it is in the convex hull of the marginal contribution vectors. There are at most n^b such distinct vectors, so we can construct a simple polynomially sized linear program that tests whether p is in the convex hull of these vectors. We can test the feasibility of the linear program in polynomial time, and thus test whether this is a core imputation in polynomial time. □

5. RELATED WORK

Auctions are a commonly used mechanism for selling or allocating goods. A key problems that such mechanisms face is that bidders may not bid truthfully. Such bid-shading is common in many auctions. Under such strategic behavior, even a mechanism that attempt to maximize social welfare based on the information it is given may reach a sub-optimal allocation, as it is given incorrect information. By using a proper payment rule, it is possible to incetivise the bidders to truthfully report their valuations. The most prominent method for achieving this is the VCG mechanism [27, 9, 14]. A detailed introduction to VCG and its properties is contained in [21].

Despite its advantages, VCG has many shortcomings [1], including vulnerability to collusion [20]. Collusion and anti-competitive behavior occur in many domains, and many of its forms are illegal [20]. Avoiding collusion has been a key issue in huge auctions, such as the famous FCC auction [10]. A study of collusion in school milk contracts in Florida and Texas [22] considers first price auctions, and shows that the cartels can operate either by dividing the market or using side-payments. Our analysis is also based on side-payments, and [22] indicates such collusion occurs in practice, and is a major problem in real-world auctions.

An analysis of bidding rings in weak cartels (without side-payments) and strong cartels (with side-payments) appears in [19]. To our knowledge, this is the first paper to study collusion in multi-unit auctions under VCG prices from a *cooperative* game theoretic perspective. Our analysis is based on solution concepts from cooperative game theory. We have considered the core and the Shapley value. The core was introduced in [13] as a solution focusing on stability. The Shapley value was introduced in [23], and has been used for distributing the gains of cooperation in a *fair* manner.

The Shapley value [23] and the similar Banzhaf index [8] were used to measure power in decision making bodies [25]. They were also used to find the importance of various nodes and links for high network reliability [7, 6, 2]. Our "optimal collusion" is somewhat similar to merging and splitting weights in weighted voting games [4, 3]. The Shapley value and other power indices are typically hard to compute [17, 18, 11, 12, 7], so our result for computing the Shapley value in a restricted version of the collusion game is especially interesting. Even for more general collusion games, the colluders can *approximate* [16, 5, 17] the Shapley value in order to share their gains.

6. CONCLUSION

We have analyzed collusion in multi-unit auctions under VCG payments, and shown that such a domain is extremely valunerable to collusion. The colluders can easily find the optimal manipulation, and split the gains from the collusion in a stable and fair manner. Our results indicate the colluders can not only find a stable way to distribute the gains, they can even use a distribution that fairly allocates each colluder a share of the utility reflecting her contribution to the colluders' utility. Due to Theorem 5, in many domains the colluders can even use a simple polynomial algorithm to compute this utility distribution. These results indicate that many factors operate in favour of the colluders in this domain, so significant counter-measures must be used to stop collusion. We now present several questions that remain open for future research.

First, we have only examined VCG multi-unit auctions. Similar analysis can be done for other auctions, such as combinatorial auctions or the GSP (Generalized Second Price) auction common in sponsored search auctions. We believe collusion can have a strong effects in sponsored seach, with a significant market impact. Second, We have assumed diminishing marginal valuations of additional items, and it may be interesting to analyze domains where this does not hold. Finally, we have assumed the colluders completely trust each other. It may be interesting to model situations where the trust is only partial. For example, colluders may not desire collusion schemes that require item transfers or large monetary transfers. Such distrust may help stop collusion.

7. REFERENCES

[1] L. M. Ausubel and P. Milgrom. The lovely but lonely vickrey auction. In *Combinatorial Auctions*, chapter 1, pages 17–40. 2006.

[2] H. Aziz, O. Lachish, M. Paterson, and R. Savani. Power Indices in Spanning Connectivity Games. In *Proceedings of the 5th International Conference on Algorithmic Aspects in Information and Management*, page 67. Springer-Verlag, 2009.

[3] H. Aziz and M. Paterson. False name manipulations in weighted voting games: splitting, merging and annexation. In *AAMAS*, pages 409–416, 2009.

[4] Y. Bachrach and E. Elkind. Divide and conquer: False-name manipulations in weighted voting games. In *AAMAS*, pages 975–982, 2008.

[5] Y. Bachrach, E. Markakis, A. D. Procaccia, J. S. Rosenschein, and A. Saberi. Approximating power indices. In *AAMAS (2)*, pages 943–950, 2008.

[6] Y. Bachrach, J. Rosenschein, and E. Porat. Power and stability in connectivity games. In *AAMAS*, pages 999–1006, 2008.

[7] Y. Bachrach and J. S. Rosenschein. Computing the Banzhaf power index in network flow games. In *AAMAS*, pages 323–329, Honolulu, Hawaii, May 2007.

[8] J. F. Banzhaf. Weighted voting doesn't work: a mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[9] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.

[10] P. Cramton and J. A. Schwartz. Collusive bidding: Lessons from the FCC spectrum auctions. *Journal of Regulatory Economics*, 17(3):229–252, 2000.

[11] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[12] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. In *AAMAS*, 2008.

[13] D. B. Gillies. *Some theorems on n-person games*. PhD thesis, Princeton University, 1953.

[14] T. Groves. Incentives in teams. *Econometrica*, 41:617–663, 1973.

[15] T. Ichiishi. Super-modularity: Applications to convex games and to the greedy algorithm for lp. *Journal of Economic Theory*, 25(2):283–286, October 1981.

[16] I. Mann and L. S. Shapley. Values of large games, IV: Evaluating the electoral college by Monte-Carlo techniques. Technical report, The Rand Corporation, Santa Monica, CA, 1960.

[17] Y. Matsui and T. Matsui. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan*, 43, 2000.

[18] Y. Matsui and T. Matsui. NP-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science*, 263(1–2):305–310, 2001.

[19] P. R. Mcafee and J. Mcmillan. Bidding rings. *The American Economic Review*, 82(3):579–599, 1992.

[20] P. Milgrom. *Putting Auction Theory To Work*. Cambridge University Press, 2004.

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

[22] M. Pesendorfer. A study of collusion in first-price auctions. *Review of Economic Studies*, 67(3):381–411, July 2000.

[23] L. S. Shapley. A value for n-person games. *Contrib. to the Theory of Games*, pages 31–40, 1953.

[24] L. S. Shapley. Cores of convex games. *Internat. J. Game Theory*, 1:12–26, 1971.

[25] L. S. Shapley and M. Shubik. A method for evaluating the distribution of power in a committee system. *American Political Science Review*, 48:787–792, 1954.

[26] A. Sullivan and S. Sheffrin. *Economics: principles in action*. Prentice Hall, 2003.

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

[28] R. J. Weber. Probabilistic values for games. Cowles Foundation Discussion Papers 471R, Cowles Foundation, Yale University, 1977.

[29] E. Winter. The Shapley Value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002.