

Average-Case Analysis of VCG with Approximate Resource Allocation Algorithms¹

Yevgeniy Vorobeychik^{a,1,*}, Yagil Engel^b

^a*Sandia National Laboratories, P.O. Box 969, Mailstop 9159, Livermore, CA 94551-0969*

^b*IBM Research, Haifa, Israel*

Abstract

The Vickrey-Clarke-Groves (VCG) mechanism offers a general technique for resource allocation with payments, ensuring allocative efficiency while eliciting truthful information about preferences. However, VCG relies on exact computation of an optimal allocation of resources, a problem which is often computationally intractable, and VCG that uses an approximate allocation algorithm no longer guarantees truthful revelation of preferences. We present a series of results for computing or approximating an upper bound on agent incentives to misreport their preferences. Our first key result is an incentive bound that uses information about *average* (not *worst-case*) performance of an algorithm, which we illustrate using combinatorial auction data. Our second result offers a simple sampling technique for amplifying the difficulty of computing a utility-improving lie. Finally, we offer a simulation-based technique for obtaining tighter, albeit empirical, bounds on incentives to misreport preferences. An important consequence of our analysis is an argument that using state-of-the-art algorithms for solving combinatorial allocation problems essentially eliminates agent incentives to lie.

Keywords: algorithmic mechanism design, VCG, average-case analysis

*Corresponding author

Email addresses: eug.vorobey@gmail.com (Yevgeniy Vorobeychik),
yagile@il.ibm.com (Yagil Engel)

¹phone: 1-773-562-0148, fax: 1-925-294-2234

¹Short preliminary versions of this work have previously appeared in the Workshop on Internet and Network Economics [21] and the International Conference on Autonomous Agents and Multiagent Systems [22].

1. Introduction

Mechanism design provides a useful practical paradigm for competitive resource allocation when agent preferences are uncertain. Perhaps of greatest practical significance has been the field of auction theory [9], and, in particular, the design of *combinatorial auctions* [2]. In a combinatorial auction, bidders are allowed to submit bids on all subsets of a given set of items.³ The auctioneer must then solve the *winner determination problem* (WDP), computing which subsets of the goods will be allocated to which bidders, with the objective of maximizing allocative efficiency.

Historically, the focus of mechanism design has been on engineering the incentives for participants to reveal their preferences truthfully, with computational aspects largely ignored. To this end, VCG has been advanced as one of the central schemes [15]. Computer Scientists have observed, however, that VCG coupled with approximate algorithms for WDP in combinatorial auctions fails to incentivize truthful revelation of preferences in most reasonable settings [17]. Since WDP is well known to be NP-Hard [12], and even hard to approximate [18], using VCG in practical combinatorial auction settings seems hopeless.

We argue that these hardness results may at times be unnecessarily pessimistic, and while the worst-case incentives to lie may exist in VCG-based mechanisms, *effective* incentives to lie may be negligible if the approximation algorithms used are very good in practice. Specifically, we present a series of results, both theoretical and experimental, which allow the designer to measure—and in some cases address—the severity of the incentive problem with VCG-based mechanisms. First, we offer general techniques to empirically assess incentive effects of specific algorithms based on *average-case* bounds. For example, if an algorithm can solve the allocation problem exactly in almost every instance, there are no incentives to deviate from truthfulness in the Bayes-Nash sense. We operationalize this bound in combinatorial auctions, illustrating how the use of a simulation-based model of bidder valuation distribution allows the designer to obtain precise probabilistic confidence bounds on agent incentives to lie. Our results provide some evidence that, at least in the VCG-based combinatorial auction setting, incentives of players to lie about their preferences are rather small. Furthermore, we show

³Items could be actual goods for sale, slots on a schedule, locations and times of banner ads displayed on a website, etc.

that the designer can use sampling to reduce the likelihood that any player will compute an improving deviation. Significantly, this is a *typical-case*, and not a *worst-case* result. Finally, we offer a simulation-based technique to obtain relatively tight, albeit approximate, bounds on incentives of agents to misreport their preference. We use this approach to offer some qualitative evidence that the incentives to lie in some combinatorial auction settings decrease with increasing problem size.

2. Related Work and Motivation

There has been considerable literature attempting to address the incentives to misreport preferences that emerge when approximate allocation algorithms are used together with the VCG price scheme, particularly in the context of combinatorial auctions. Here we give several representative examples. Sanghvi and Parkes [20] demonstrate that computing an improving deviation in VCG-based combinatorial auctions is NP-Hard, although this worst-case result is difficult to rely on in practice. Lavi and Swamy [10] present a truthful (in expectation) mechanism when the approximation algorithm bounds the integrality gap of LP relaxation, while Lehmann, O’Callaghan, and Shoham [13] and Mu’alem and Nisan [16] obtain general truthful mechanisms for combinatorial auctions when bidders are “single-minded” (i.e., each has positive value for exactly one bundle of items). Dobzinski, Nisan, and Shapira [5] present a framework for designing truthful approximation algorithms, and demonstrate instances with an asymptotically optimal worst-case bound for the general WDP. Nisan and Ronen [17] develop a mechanism called *second-chance*, in which players are not capable of computing a beneficial lie.

The extensive literature addressing the incentive problems of approximate WDP implicitly suggests that such problems are critical. Field practitioners of combinatorial auctions, however, seem to rarely, if ever, have come up against the worst-case complexity issues [4, 3]. Furthermore, the majority of combinatorial auction problems that have been studied in simulation can be solved very fast using modern algorithms [14, 18, 19], and, indeed, the general-purpose CPLEX integer programming tool is usually very effective [19].

There thus appears to be a gap between theory, which views the incentive problem of VCG-based mechanisms as severe, and practice, which ignores it almost entirely. We believe that one important reason for this gap is that

mechanisms that have been developed to be both truthful and to provide suitable approximation guarantees tend to be somewhat complicated to implement and difficult to communicate. Frequently, they randomize allocation, something that is difficult to operationalize because of fairness considerations (for example, Dobzinsky, Nisan, and Shapira [5] suggest withholding a random sample of bidders from consideration, an idea which some bidders may find disagreeable). Furthermore, worst-case guarantees provided tend to be so poor as to be of little practical consolation. Another important reason is that whereas literature to date has been motivated by the desire to prevent the worst-case performance, state-of-the-art algorithms (e.g., CPLEX) solve most realistic allocation problems (such as WDP) to optimality, and the worst case is rarely encountered in practice. Consequently, one may expect that in reality incentives to misreport preferences are often quite low. While practitioners appear to grasp this, they have, to date, no principled means to *verify* it. Our goal is to offer mechanism designers techniques to quantify (and, sometimes, to reduce) the incentives of bidders to lie, while capturing information about the *distribution* of the performance of their algorithm, rather than merely its worst-case performance.

A downside to our approach is that it is no longer distribution-free. Distribution-free mechanism design has clear advantages. First, we rarely truly know the distribution of agent preferences in practice, and attempts to glean information about it are likely to be gamed if the stakes are high enough. We also prefer not to assume that agents have common knowledge of such information either. Furthermore, economists have long been striving to reduce the common knowledge requirements in mechanism design [23]. Ultimately, a distribution-free guarantee is certainly most satisfying, as it relies on minimal assumptions. Many situations, however, do not lend themselves easily to simple, natural distribution-free mechanisms. Additionally, at times information about the distribution is available in some form. For example, combinatorial auctions are often run in high-stakes settings (such as wireless spectrum auctions), where there is considerable public information about the participants. Furthermore, a test suite has been developed for combinatorial auctions precisely with the goal of generating realistic problem instances [14]. Finally, there has already been much experience running large-scale combinatorial auctions, with state-of-the-art algorithms proving thus far quite capable [3]. All of this suggests that there may be many auction settings where much distributional information *is* available, and which therefore lend themselves well to the approaches we propose.

3. Preliminaries

In our setting, each player $i \in I$ submits to a central designer his utility function, as indexed by his type $t_i \in T_i$. Let O be the set of outcomes (e.g., feasible allocations), I be a set of n players, and let $T = T_1 \times \cdots \times T_n$ be the joint type set. Assume that O and T_i for all $i \in I$ are non-empty and compact. Let $F(\cdot)$ be a probability distribution over joint player types and let $u_i(t_i, o, p_i)$ be player utility functions where $o \in O$ typically depends on joint player report t and p_i is the payment received by the agent (which is negative when the agent is paying the designer). While we may hope that all players submit their types honestly, they may choose to lie, submitting some t'_i instead of t_i , and these lies could, in general, be a function of true type t_i .

We assume that agent utility functions are quasi-linear in payments p_i , that is $u_i(t_i, o, p_i) = v_i(t_i, o) + p_i$, where $v_i(t_i, o)$ is the underlying value that player i with type t_i has for outcome o ; we assume it to be continuous in both arguments. A mechanism is a function that chooses an outcome o and assigns the payments p_i for all players i given a joint report of types $t \in T$. Thus, we use $o(t)$ and $p_i(t)$ to indicate such choices as made by some specified mechanism.

A central aspect of mechanism design is the prediction of agent play for a given choice of a mechanism. Typically the role of such predictions is played by equilibrium concepts. We appeal to two such concepts below (defined with respect to direct revelation mechanisms, that is, mechanisms which attempt to truthfully elicit player preferences). Under a *dominant strategy equilibrium* each player is (weakly) best off reporting his true type *no matter what other players do*. Under a *Bayes-Nash equilibrium*, on the other hand, each player maximizes his expected utility by reporting his true type t_i , *assuming that all other players are honest*. Both equilibrium concepts admit natural notions of approximation: in an ϵ -dominant strategy equilibrium, a player can gain no more than ϵ by deviating, no matter what the opponents do, whereas an ϵ -Bayes-Nash equilibrium guarantees that the expected gain to any player from deviation is at most ϵ , with expectation taken with respect to the joint type distribution.

A useful measure of strategic stability is that of *game-theoretic regret*. While in general this measure can be defined for any joint strategy profile, we use it only to gauge the regret of truthful reporting. Hence, we use a

simpler definition, with $\tilde{\epsilon} = E_F[\epsilon(t)] = E_F[\max_i \epsilon_i(t_i)]$, where

$$\epsilon_i(t_i) = \max_{t'_i \in T_i} E_F[u_i(t_i, o(t'_i, t_{-i}), p_i(t'_i, t_{-i})) - u_i(t_i, o(t), p(t)) | t_i].$$

In words, it is the maximum expected benefit any player can obtain from reporting untruthfully.

A widely studied goal of mechanism design, and one we focus on here, is that of maximizing *social welfare*, or the sum of player valuations. Formally, define *social welfare* to be

$$V(t, o) = \sum_{i \in I} v_i(t_i, o),$$

where o is an outcome and t is a joint type profile. Let $o^* : T \rightarrow O$ denote the welfare optimal (efficient) outcome (allocation) and let

$$V^*(t) = \sum_{i \in I} v_i(t_i, o^*(t)) = \max_{o \in O} \sum_{i \in I} v_i(t_i, o)$$

be the maximum welfare achieved for a type profile t . Let $V^* = \max_{t \in T} V^*(t)$. It is well known that optimal allocation can be achieved as a truthful dominant strategy equilibrium by using Groves payments [15], with $p_i(t) = \sum_{j \neq i} v_j(t_j, o^*(t)) + h_i(t_{-i})$. Here h_i is any real-valued function of the types reported by other players; for simplicity of exposition, we set it to 0.⁴

Let $g : T \rightarrow O$ be an algorithm for computing an approximately efficient allocation.⁵ We say that $g(\cdot)$ is an α -approximation if $V^*(t) \leq \alpha V_g(t)$ for any $t \in T$. Since g may compute only a suboptimal allocation, we let $V_g(t)$ be welfare at the allocation $g(t)$, that is $V_g(t) = \sum_{i \in I} v_i(t_i, g(t))$. Define VCG-based payments by $p_i^g(t) = \sum_{j \neq i} v_j(t_j, g(t)) + h_i(t_{-i})$. Hence, the VCG-based mechanism selects an outcome according to g , and the players receive payments $p_i^g(t)$.

⁴VCG extends the Groves scheme by specifying $h_i(t_{-i})$ to guarantee *individual rationality*, that is, that every player obtains positive net value from participation. Since the subject of our inquiry is the incentive structure, setting $h_i(t_{-i})$ to 0 has no impact on any of the arguments we make.

⁵Our results can be extended rather directly to randomized allocation algorithms.

4. Connecting Approximation and Incentives

We begin our endeavor by exhibiting a simple bound that connects the approximation quality of an algorithm to the players' incentives to misreport preferences (both in the worst-case sense), due to Kothari, et al. [8].

Theorem 1. *Suppose that g is an α -approximation algorithm. Then truthful reporting is an ϵ -dominant strategy equilibrium for $\epsilon = \frac{\alpha-1}{\alpha}V^*$.*

While this bound is intuitive and easy to use in principle, in many interesting settings it is much too crude due to its worst-case nature. For example, the best known approximation ratio in combinatorial auctions is 2 (specifically, a greedy algorithm is a 2-approximation when player valuations are submodular [11]). From a practical perspective, the resulting bound of $\frac{V^*}{2}$ is hardly encouraging: if the incentives to lie are as high as 50% of welfare, then we can safely say that honesty would be remarkably altruistic.

Given this state of affairs in combinatorial auctions, one may expect practitioners to worry about incentives. Many real problems, however, are “easy” in that the optimal or nearly optimal allocation can be found extremely fast in practice. Thus, while an algorithm may prove very bad in the worst case, it may be quite effective in a typical case. Our goal now is to incorporate this “empirical” flavor into the analysis of incentives to lie.

To begin, suppose that, somehow, we have an approximation bound for g that is a *known function* of $\alpha(t)$ for all $t \in T$. In the most trivial case, it could be just a fixed α , reducing the setup to the worst-case analysis above. Alternatively, we may be able to split the set of type profiles into subsets T^1, T^2, \dots , and obtain much better uniform bounds on some of these subsets than the worst case analysis would allow. For example, perhaps we know that for some large subset of combinatorial auction problems we can compute exact or nearly exact optimal allocation quickly. In any case, presently we will see that we need not even construct $\alpha(t)$ for all possible type profiles, but can obtain probabilistic bounds based on a sample of a finite subset of these.

Our first key result echoes Theorem 1, although we must weaken the approximate equilibrium notion to Bayes-Nash.⁶

⁶The reason is that a player i is assumed to only know his own type t_i , and will thus only have a distribution over $\alpha(t)$ conditional on t_i .

Theorem 2. *Suppose that the algorithm g is an $\alpha(t)$ -approximation. Then a player i can gain at most $\epsilon_i(t_i)$ when others are playing truthfully, where*

$$\epsilon_i(t_i) = E_{t_{-i}} \left[\frac{\alpha(t) - 1}{\alpha(t)} V^*(t) | t_i \right].$$

The proofs of this and other results are in the appendix.⁷

Corollary 3. *Suppose that the algorithm g is an $\alpha(t)$ -approximation. Then truthful reporting constitutes an ϵ -Bayes-Nash equilibrium for*

$$\epsilon = n E_t \left[\frac{\alpha(t) - 1}{\alpha(t)} V^*(t) \right].$$

Proof. Here we measure the expected benefit to deviation using the corresponding game-theoretic regret.

$$\begin{aligned} E_t[\epsilon(t)] &= E_t[\max_i \epsilon_i(t_i)] \leq E_t[\sum_i \epsilon_i(t_i)] = \sum_i E_{t_i}[\epsilon_i(t_i)] \\ &= \sum_i E_t \left[\frac{\alpha(t) - 1}{\alpha(t)} V^*(t) \right] \\ &= n E_t \left[\frac{\alpha(t) - 1}{\alpha(t)} V^*(t) \right]. \end{aligned}$$

□

Corollary 3 gives us a bound on the incentives of any player to lie that can incorporate full information about the distribution of algorithmic performance $\alpha(t)$ induced by the distribution of instances (player preferences) F . Our use of the sum bound for the maximum, however, loses considerable tightness as compared to Theorem 1. Below we investigate, first theoretically (Section 5) and later empirically (Section 6), the situations in which our *average-case* bound has bite.

⁷While the proof of Theorem 2 is similar to that for the uniform bound (Theorem 1), note that we must be careful about the nature of information available to players (see the previous footnote). We also require the application of the following corollary to actually arrive at the incentive bound.

5. Illustration of the Non-Uniform Incentive Bound

To illustrate an application of the non-uniform bound on the incentives to lie (i.e., Corollary 3), suppose that the space of joint types T can be partitioned into “easy” and “hard” type profiles, that is, $T = \underline{T} \cup \bar{T}$. Let $\underline{\alpha} = \sup_{t \in \underline{T}} \alpha(t)$ and $\bar{\alpha} = \sup_{t \in \bar{T}} \alpha(t)$ and assume that $\underline{\alpha} \leq \bar{\alpha}$. For example, \underline{T} can be the set of combinatorial auction problem instances for which solutions are relatively easy to compute given state-of-the-art optimization tools (perhaps very good approximations, i.e., small $\underline{\alpha}$, can be obtained very fast for these instances). \bar{T} , on the other hand, can be the set of those instances for which the worst-case performance of an algorithm is realized. Then,

$$\begin{aligned}
 E_t \left[\frac{\alpha(t) - 1}{\alpha(t)} V^*(t) \right] &= \int_{\underline{T}} \frac{\alpha(t) - 1}{\alpha(t)} V^*(t) dF(t) + \int_{\bar{T}} \frac{\alpha(t) - 1}{\alpha(t)} V^*(t) dF(t) \\
 &\leq \frac{\underline{\alpha} - 1}{\underline{\alpha}} \int_{\underline{T}} V^*(t) dF(t) + \frac{\bar{\alpha} - 1}{\bar{\alpha}} \int_{\bar{T}} V^*(t) dF(t) \\
 &= \frac{\underline{\alpha} - 1}{\underline{\alpha}} E_t[V^*(t)] - \frac{\underline{\alpha} - 1}{\underline{\alpha}} \int_{\bar{T}} V^*(t) dF(t) \\
 &\quad + \frac{\bar{\alpha} - 1}{\bar{\alpha}} \int_{\bar{T}} V^*(t) dF(t) \\
 &\leq \frac{\underline{\alpha} - 1}{\underline{\alpha}} E_{t \sim F|\underline{T}}[V^*(t)] + \left(\frac{1}{\underline{\alpha}} - \frac{1}{\bar{\alpha}} \right) V^* F(\bar{T}),
 \end{aligned}$$

Note that since $\left(\frac{1}{\underline{\alpha}} - \frac{1}{\bar{\alpha}}\right) V^*$ is just a constant, as the probability measure of “hard” instances becomes small, the incentives for players to deviate approach $\frac{\underline{\alpha} - 1}{\underline{\alpha}} E_{t \sim F|\underline{T}}[V^*(t)]$. Hence the following corollary.

Corollary 4. *Suppose that $F(\bar{T}) = 0$. Then truthful reporting constitutes an ϵ -Bayes-Nash equilibrium for $\epsilon = n \frac{\underline{\alpha} - 1}{\underline{\alpha}} E_{t \sim F|\underline{T}}[V^*(t)]$.*

In the special case when $\underline{\alpha} = 1$ (that is, easy instances can be solved *exactly* in a relatively short time) as is the case in many combinatorial auction settings, and when $F(\bar{T}) = 0$, that is, when the probability of drawing a hard problem is 0, truthful reporting is a Bayes-Nash equilibrium. Hence the following direct corollary.

Corollary 5. *Suppose that $\underline{\alpha} = 1$ and $F(\bar{T}) = 0$. Then the strategy $s_i(t_i) = t_i$ —that is, truthfully reporting actual preferences—is a Bayes-Nash equilibrium under the allocation algorithm g .*

Proof. $E_t[\epsilon(t)] = nE_t\left[\frac{\alpha(t)-1}{\alpha(t)}V^*(t)\right] = n\frac{\alpha-1}{\alpha}E_t[V^*(t)] = 0.$ \square

Corollaries 4 and 5 are mainly conceptual observations. In practice, we can rarely know that, for example, $F(\bar{T}) = 0$. Nevertheless, as we demonstrate below, we can obtain practical probabilistic bounds based on Corollary 3.

6. Applying the Non-Uniform Incentive Bound

A key question that stems from the above analysis is how a mechanism designer would determine an incentive bound for his algorithm in practice. We would not, for example, want to require the designer to obtain a non-trivial $\alpha(t)$ for every $t \in T$. Rather, we offer the following empirical approach:

1. Obtain or construct a simulator that allows one to sample joint player types $t \in T$ according to F ;
2. Collect a set of K joint type samples t^1, \dots, t^K ;
3. For each t^k , compute $V_g(t^k)$ and $V^*(t^k)$ (or an upper bound $\bar{V}^*(t^k)$ on $V^*(t^k)$, which could be obtained, for example, using LP relaxation instead of solving the mixed integer program for this instance);
4. Compute $\alpha(t^k) = \frac{V^*(t^k)}{V_g(t^k)}$, let $\hat{Z}(t^k) = \frac{\alpha(t^k)-1}{\alpha(t^k)}V^*(t^k)$, and define $\hat{Z} = \frac{1}{K} \sum_{k=1}^K \hat{Z}(t^k)$;
5. Compute a probabilistic bound based on \hat{Z} .

The first step requires a designer to either obtain or construct a simulator. This seems rather demanding, but may be necessary to do for a high-stakes problem anyway. Moreover, in the case of combinatorial auctions, a state-of-the-art simulator to generate realistic problem instances is already publicly available [14].

For the last step, we have a few options. A most general option would be to use a distribution-free bound (e.g., Hoeffding inequality), but these tend to be very loose. Instead, we assume that \hat{Z} is Normally distributed (an assumption that is justified by the Central Limit Theorem when K is large; Central Limit Theorem applies here since $\hat{Z}(t^k)$ are i.i.d. and our assumptions of continuity of $v_i(\cdot)$ and compactness of T imply that the variance of $\hat{Z}(t^k)$ is finite). Suppose we use $s^2(\hat{Z}(t^k))/K$ (where $s^2(\cdot)$ is the sample variance) as an estimate of the variance of \hat{Z} . Then,

$$E_t\left[\frac{\alpha(t)-1}{\alpha(t)}V^*(t)\right] \leq \hat{Z} + z_\delta \sqrt{\frac{s^2(\hat{Z}(t^k))}{K}} \quad (1)$$

with probability at least $1 - \delta$, where z_δ is the value of Normal distribution at $1 - \delta$.

6.1. Example: Combinatorial Auctions

To illustrate the techniques introduced above, we now offer an incentive analysis of combinatorial auctions based on auction instances (in our notation, t^k) generated by CATS [14]. Since the absolute values of the bounds are not very meaningful, we give them as fractions of V^* . While V^* is actually unknown, note that $\hat{Z}/V^* \leq \hat{Z}/\max_k V^*(t^k)$, so below we report $\hat{Z}' = \hat{Z}/\max_k V^*(t^k)$. Additionally, CATS generates a set of bids, but does not specify the number of players (which could therefore be arbitrary). Consequently, we ultimately report bounds as multiples of nV^* .

The data set we used is composed of (a) a set of samples with 1000 bids on 144 goods ($1K - 144$), (b) a set with 1000 bids on 256 goods ($1K - 256$), (c) a set with 2000 bids on 64 goods ($2K - 64$), and (d) a set with varying problem sizes (varsize). Each set contains 5000 samples, 500 for each of 10 different distributions. The data include the result obtained by CPLEX which ran to optimality, the results obtained by CASS [6] after about 7500 seconds for $1K - 144$ and $1K - 256$, or 44000 seconds for the other datasets, and, for the dataset $1K - 256$, the result obtained by the Gonen-Lehmann (GL) algorithm [7].⁸

We computed the bound on the incentives of agents to lie for each dataset, as well as for the union set. For each one we include the data for all CATS distributions except “arbitrary”. For $g(t)$ we used the following combination: we used the result returned by CPLEX for a sampled profile t^k if it was obtained in at most S seconds; otherwise the result returned by CASS was used. We varied the time limit S between 500 and 60000 seconds (about 16.6 hours). The longer time limits are reasonable for high volume auctions in which a lot of money is at stake.⁹

In Figure 1 (left), we show the resulting bound for each dataset as a function of the time limit, which allows us to quantify the tradeoff between the amount of time given to the algorithm and regret (incentives for players

⁸A small fraction of the problems for which CPLEX reported an optimal result are not in fact optimal, because the results by CASS and GL are higher. On these problems we use the maximum value of the three algorithms as the “true” optimum.

⁹Since the data are relatively old, our bounds are likely excessively pessimistic.

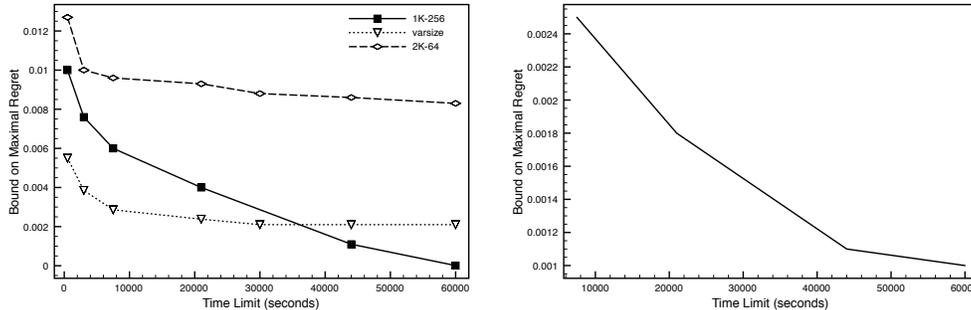


Figure 1: Upper bound on regret, as a fraction of nV^* , *left*: for several data set sizes, and *right*: for the union of all data.

to lie).¹⁰ The bounds are computed as explained above, with confidence level $1 - \delta = 0.95$. The chart for the dataset of $1K - 144$ is omitted because the \hat{Z} is zero (all instances were quickly solved to optimality). In this case, we can obtain an upper bound of 0.0006 on the proportion of suboptimally solved instances (giving a regret bound of $0.0006nV^*$) with 0.95 confidence using the Clopper-Pearson bound [1].

The results for the union data set are shown in Figure 1, right. Observe that for all our results, with the possible exception of $2K - 64$, the bound on incentives to lie is quite low, far lower than $\frac{V^*}{2}$ suggested by the uniform (worst-case) analysis. Thus, if the number of players is not too large, there do not appear to be significant incentives for bidders to lie.¹¹

7. Computing a Better Response

Our average-case analysis of incentives to deviate assumes that mechanism participants have unlimited computational power. However, if we find the allocation problem fundamentally hard to solve for the designer, it is likely that it is no less difficult for players to profitably deviate. Indeed, Sanghvi and Parkes [20] show that computing a better response is NP-Hard

¹⁰The raw data for the plots in Figure 1 is available publicly at <http://vorobeychik.com/downloads.html>.

¹¹When the number of players is very large, the incentives to lie tend to become negligible as the market becomes close to competitive, since each bidder is then unlikely to influence the outcome.

for a combinatorial allocation problem under certain assumptions on g . This worst-case result, however, is unsatisfying in our setting, since as long as the players can *often* (that is, on many actual problem instances) compute a better response, they will try to do so. Insofar as we can bound their gains tightly, this would not be a concern. But for the case that our bounds leave enough room for gains, we wish to have a general purpose technique to make deviations relatively challenging to compute.

In this section we suggest a very simple sampling technique which allows us to amplify complexity of the deviation problem *on average*, under some assumptions on the algorithmic capabilities of the mechanism participants.

Our first result reflects an assumption that the designer can construct a belief (prior) distribution over the algorithms which would be used (independently) by each player. Our results below are then with respect to the randomized “pseudo-algorithm” induced by this distribution. In its simplest form, it may be that the problem is well-enough understood that state-of-the-art algorithms for computing improving deviations are readily available, so both the designer and the players would simply utilize the best of these. Under this assumption, consider the following sampling algorithm:

1. For each player i , draw L samples t'_i from the belief distribution over the players’ algorithms (we can think of these as sample deviations); let $T'_i = \{t_i, t'_i{}^1, \dots, t'_i{}^L\}$, where t_i is the actual reported type of i ;
2. Define $g'_i(t) \in \arg \max_{o=g(t'_i, t_{-i}) | t'_i \in T'_i} \sum_{j \in I} v_j(t_j, g(t'_i, t_{-i}))$ for each i ;
3. Define $g'(t) \in \arg \max_{o=g'_i(t) | i \in I} \sum_{j \in I} v_j(t_j, g_i(t))$.

Observe that this enhanced algorithm can only improve social welfare.

Theorem 6. *Given $g'(t)$ as the allocation mechanism, the probability (with respect to the belief distribution) that some player can compute an improving deviation is at most $\frac{n}{L+1}$.*¹²

A direct consequence of Theorem 6 is that incentives of any player to lie can be made arbitrarily small as long as there is some non-zero cost to performing better-response computations.¹³

¹²This result is only meaningful when $n < L + 1$. However, note that L is under the designer’s control. We assume that it is cheap for the designer to take such samples; see the next footnote.

¹³We are ignoring the cost of such computation to the designer. Our assumption is that

While the assumption that algorithmic capabilities of players are predictable is often reasonable, we may wish to make a stronger statement. For example, perhaps sampling deviations using some very simple heuristic (e.g., uniform random search), while entirely different from what players may in fact do, nevertheless makes profitable deviations increasingly difficult for them to compute. As the following theorem suggests, the above result does, indeed, generalize (in a somewhat weaker form) to a very large class of sampling distributions and player deviation algorithms.

Formally, let $G(u)$ be the distribution function of player utilities induced by the *designer's* search process (e.g., uniform sampling from the type space), whereas $H(u)$ is the distribution function of player i 's utilities induced by the player's search.

Theorem 7. *Let $U_1 = \{u | G(u) = 1\}$ and suppose that $H(U_1) = 0$. Then $\lim_{L \rightarrow \infty} \int_{\mathbb{R}} G(u)^L dH(u) = 0$.*

The interpretation is that as long as the players do not have a positive probability of reaching a utility that is better than *any* that the designer *can possibly* attain, the designer can use random sampling to effectively eliminate incentives to lie. In essence, in order to gain from lying the players need to know significantly more about the problem than the designer.

8. Simulation-Based Analysis of Incentives to Lie

The theoretical bounds above are in an important sense rather crude. Specifically, recall that we arrived at Corollary 3 by bounding $E_t[\max_i \epsilon_i(t_i)]$ with $\sum_i E_{t_i}[\epsilon_i(t_i)]$. That the sum can be a very loose upper bound is easy to see with a simple illustration. Consider a collection of i.i.d. random variables distributed uniformly on a unit interval. Clearly, the expectation of the maximum of these cannot exceed 1. On the other hand, if we have n such variables, the sum of expectations gives us an upper bound of $n/2$, or a factor of $O(n)$ larger than the quantity it bounds! While the incentives to lie may often be small enough for the loose upper bound of Corollary 3 to nevertheless give meaningful results, a negative conclusion based on it need not imply that all hope is lost. Rather, we present in this section a simulation-based scheme

the designer's computational capacity far surpasses that of any agent. This assumption is not needed for the next result, since there the designer can use a very low-cost algorithm, such as uniform random search.

that bounds more tightly, albeit approximately, the incentives of agents to lie for any given approximation algorithm combined with a VCG-based scheme. A further benefit of a fully simulation-based approach is that it allows us to combine very naturally the upper bound on regret with a prior distribution on the algorithms that agents may use to compute a utility-improving lie. We illustrate our simulation-based technique in the context of combinatorial auctions using a greedy approximation algorithm [11]. Since the greedy algorithm, while highly effective when valuation distribution is submodular, can be quite poor when complementarities are significant, we should not draw conclusions based on *absolute* regret values presented; rather, we offer some qualitative insights that are suggestive of a broader pattern.

Consider the following algorithm for simulation-based regret approximation.

1. Generate M random type profile t according to F ;
2. For each sampled type profile t , generate L random deviations t'_i for each player i , drawn according to a randomized distribution G ;¹⁴
3. Compute approximate regret for each player i ,

$$\hat{\epsilon}_i(t_i) = \max_{l=1, \dots, L} u_i(t'_i, g(t)) - u_i(t_i, g(t));$$

4. Compute approximate regret $\hat{\epsilon}(t) = \max_i \hat{\epsilon}_i$;
5. Compute average regret $\hat{\epsilon} = \frac{1}{M} \hat{\epsilon}(t^m)$.

For the results reported below, we chose M to be 40-100 and let $L = 100$.¹⁵ The results that we report are actually (approximate) upper bounds on the incentives to deviate because (a) we consider maximum gain for *every type profile* sampled, rather than for each player given the distribution of other player types, and (b) we report the fraction of utility relative to the greedy, rather than optimal, allocation (i.e., we report $\hat{\epsilon} / \max_{m=1, \dots, M} V_g(t^m)$).

Our first set of simulations considers two kinds of valuations: randomly generated submodular valuations and valuations in which marginal values

¹⁴These could be drawn, for example, uniformly randomly, or using a prior distribution over player algorithms. In our experiments, we drew deviations according to the marginal distribution F_i of i 's types.

¹⁵100 randomly chosen deviations appears sufficiently large that increasing it does not have a significant impact on our results. The code and results of this analysis can be downloaded from <http://vorobeychik.com/downloads.html>.

of items are generated uniformly randomly on the unit interval. We present these results in Figure 2, where we separately vary the number of players and items in the auction. As we can observe, in spite of the rather weak worst-

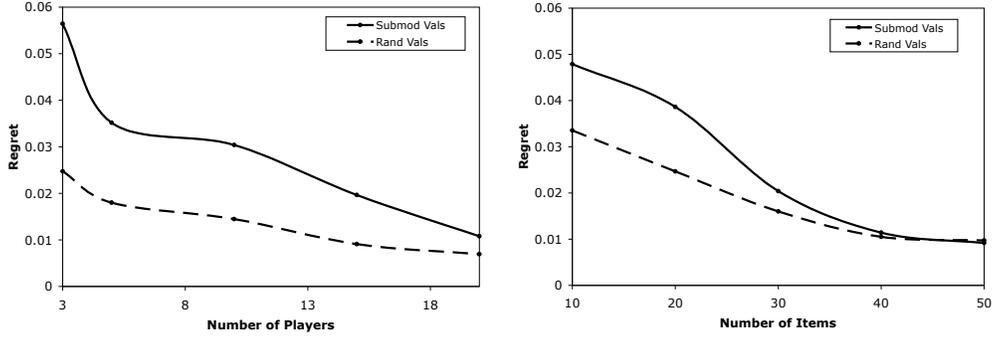


Figure 2: Average game-theoretic regret when greedy algorithm is used to approximate WDP of a combinatorial auction with submodular valuations. *Left*: as a function of the number of players; the number of items is fixed at 20. *Right*: as a function of the number of items; the number of players is fixed at 5.

case guarantees on the performance of the greedy algorithm in the case of submodular valuations, the average regret tends to be low. Additionally, regret tends to *decrease* with increasing complexity of the allocation problem. In order to assess the robustness of the latter phenomenon to different distributions of valuations, we use the CATS tool [14].

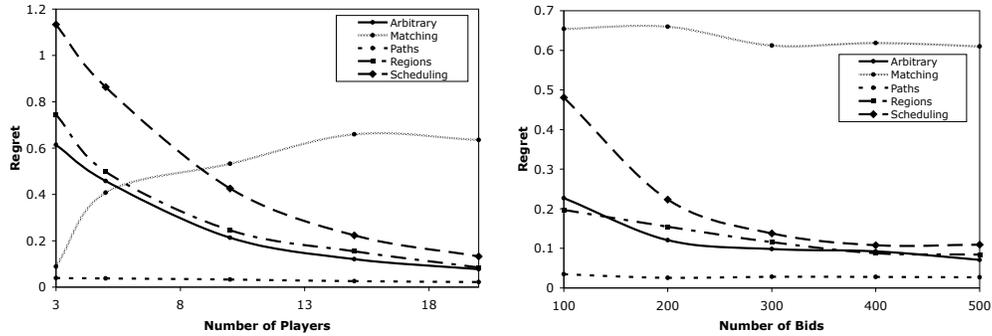


Figure 3: Average game-theoretic regret in combinatorial auctions with submodular and random valuations. *Left*: as a function of the number of players; the number of bids is 20. *Right*: as a function of the number of submitted bids; the number of players is 15.

The results for several CATS distributions are shown in Figure 3.¹⁶ While incentives to lie can be relatively large, in all but one case the incentives to deviate from truthful reporting decrease with increasing problem complexity, suggesting that this observation is rather robust.

9. Conclusion

We presented a series of results that allow construction of average-case bounds on agent incentives to lie about their preferences for VCG-based mechanisms. Conceptually, this deviates from the more traditional worst-case analysis which often fails to provide meaningful bounds. Practically, we introduce a simple method for assessing incentive properties of specific approximation algorithms, and even heuristics that lack formal approximation guarantees, in the context of economic resource allocation problems. We illustrate the resulting empirical incentive analysis for a specific approximation algorithm in the context of several combinatorial auction problems. Our results here suggest that using state-of-the-art algorithms for solving combinatorial allocation problems essentially eliminates agent incentives to misreport their preferences. In addition, we show that even if incentives to lie about true player types are significant, the designer can use sampling to make it unlikely that any player will compute a utility-improving lie. This provides a typical-case complement to an already known worst-case hardness result. Our final contribution is a fully simulation-based method for approximating tighter bounds on incentives of agents to misreport their preferences. As an illustration of the power of this method, we demonstrate that incentives to lie decrease with increasing problem size when the designer uses a greedy approximation algorithm.

References

- [1] C. Clopper, E. Pearson, The use of confidence or fiducial limits illustrated in the case of the binomial, *Biometrika* 26 (1934) 404–413.
- [2] P. Cramton, Y. Shoham, R. Steinberg (Eds.), *Combinatorial Auctions*, MIT Press, 2006.

¹⁶For details about the distributions see Leyton-Brown and Shoham [14].

- [3] R.W. Day, P. Cramton, The quadratic core-selecting payment rule for combinatorial auctions, 2010,. Working paper.
- [4] R.W. Day, S. Raghavan, Fair payments for efficient allocations in public sector combinatorial auctions, *Management Science* 53 (2007) 1389–1406.
- [5] S. Dobzinski, N. Nisan, M. Shapira, Truthful randomized mechanisms for combinatorial auctions, in: 38th ACM Symposium on Theory of Computing, pp. 644–652.
- [6] Y. Fujishjima, K. Leyton-Brown, Y. Shoham, Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches, in: International Joint Conference on Artificial Intelligence, pp. 548–553.
- [7] R. Gonen, D. Lehmann, in: Electronic Market Design Workshop.
- [8] A. Kothari, D.C. Parkes, S. Suri, Approximately-strategyproof and tractable multi-unit auctions, *Decision Support Systems* 39 (2005) 105–121.
- [9] V. Krishna, *Auction Theory*, Academic Press, 2002.
- [10] R. Lavi, C. Swamy, Truthful and near-optimal mechanism design via linear programming, in: 46th IEEE Symp. on Foundations of Comp. Sc., pp. 595–604.
- [11] B. Lehmann, D. Lehmann, N. Nisan, Combinatorial auctions with decreasing marginal utilities, *Games and Economic Behavior* 55 (2006) 270–296.
- [12] D. Lehmann, R. Müller, T. Sandholm, The winner determination problem, in: [2], pp. 297–318.
- [13] D. Lehmann, L.I. O’Callaghan, Y. Shoham (????).
- [14] K. Leyton-Brown, Y. Shoham, A test suite for combinatorial auction, in: [2], pp. 451–478.
- [15] A. Mas-Colell, M.D. Whinston, J.R. Green, ????

- [16] A. Mu’alem, N. Nisan, in: AAAI.
- [17] N. Nisan, A. Ronen, Computationally feasible VCG mechanisms, *Journal of Artificial Intelligence Research* 29 (2007) 19–47.
- [18] T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions, *Artificial Intelligence* 135 (2002) 1–54.
- [19] T. Sandholm, S. Suri, A. Gilpin, D. Levine, CABOB: A fast optimal algorithm for winner determination in combinatorial auctions, *Management Science* 51 (2005) 374–390.
- [20] S. Sanghvi, D.C. Parkes, Hard-to-Manipulate Combinatorial Auctions, Technical Report, Harvard University, 2004.
- [21] Y. Vorobeychik, Y. Engel, Average-case analysis of incentives under approximate allocation algorithms, in: *Sixth International Workshop on Internet and Network Economics*, pp. 251–258.
- [22] Y. Vorobeychik, Y. Engel, Incentive analysis of approximately efficient allocation algorithms, in: *Ninth International Conference on Autonomous Agents and Multiagent Systems*, pp. 1479–1480.
- [23] R. Wilson, Game-theoretic analyses of trading processes, in: T. Bewley (Ed.), *Advances in Economic Theory: Fifth World Congress*, Cambridge University Press, Cambridge, U.K., 1987, pp. 33–70.

Appendix A. Welfare Properties of VCG-based Mechanisms

It is generally assumed that incentives to lie are undesirable, in part because they result in greater uncertainty about outcomes. However, such incentives would pose a substantially lesser problem if they are aligned with social utility. Note that under VCG-based payments any unilateral deviation that improves player i ’s utility is also welfare improving. Specifically, observe that

$$\begin{aligned}
 u_i(t_i, g(t'_i, t_{-i}), p_i(t'_i, t_{-i})) &= \sum_j v_j(t_j, g(t'_i, t_{-i})) > u_i(t_i, g(t), p_i(t_i, t_{-i})) \\
 &= \sum_j v_j(t_j, g(t)).
 \end{aligned}$$

However, group deviations may in general lead to welfare loss. A key question, then, is whether there necessarily exists a welfare improving Bayes-Nash equilibrium strategy profile. We now show that the answer is, in general, negative.

Example 1. Consider the following combinatorial auction setting. We have two players (1 and 2) and two items (1 and 2). As is standard, assume that $v_1(\emptyset) = v_2(\emptyset) = 0$ and consider the following value functions:

$$\begin{aligned} v_1(\{1, 2\}) &= 10, & v_1(\{1\}) &= v_1(\{2\}) = 4 \\ v_2(\{1, 2\}) &= 5, & v_2(\{1\}) &= v_2(\{2\}) = 2. \end{aligned}$$

Define v'_1 and v'_2 to be:

$$\begin{aligned} v'_1(\{1, 2\}) &= 2, & v'_1(\{1\}) &= v'_1(\{2\}) = 0 \\ v'_2(\{1, 2\}) &= 2, & v'_2(\{1\}) &= v'_2(\{2\}) = 0 \end{aligned}$$

and suppose that the algorithm g allocates the items as follows:

- $g(v_1, v_2)$ assigns good 1 to player 1 and good 2 to player 2 (for a total welfare of 6)
- $g(v'_1, v_2)$ assigns both goods to player 1 (to yield the optimal welfare of 10)
- $g(v'_1, v'_2)$ assigns both goods to player 2 (to yield total welfare of 5)
- $g(v''_1, v'_2)$ assigns both goods to player 2 for any v''_1
- $g(v'_1, v''_2)$ assigns both goods to player 1 for all v''_2 except $v''_2 = v'_2$.
- $g(v''_1, v''_2)$ assigns both goods to player 2 for all v''_1 and v''_2 , with the exception of the cases outlined above.

Now, for the computation of player utilities below, ignore the h_i payment term as it does not affect the players' incentives. Observe that in this example, (v_1, v_2) is not an equilibrium, since the utility to each players is 6, whereas player 1 could obtain 10 by deviating to v'_1 , which would yield the utility of 2 for the second player. Furthermore, (v'_1, v_2) is not an equilibrium either, since player 2 could now gain by deviating to v'_2 , obtaining the utility of 5, which would give player 1 the utility of 2. The profile (v'_1, v'_2) is, however, an

equilibrium, and yields lower welfare than the truthful profile. Additionally, any pure or mixed strategy profile with support on v_1'' and v_2'' that are not the special cases described above will yield the same welfare as (v_1', v_2') .

To see that no mixed strategy equilibrium with any support will do the job, note that we can only increase welfare by having player 2 play v_2 as a part of the support. Without loss of generality, let's look at the restricted game with player 1 choosing between actions v_1 and v_1' and player 2 choosing between v_2 and v_2' . Suppose that player 2 plays v_2 with probability α and v_2' with probability $1 - \alpha$. Then the utility of player 1 from playing v_1 is $4\alpha + 2$, while his utility from playing v_1' is $8\alpha + 2$, and the two are only equal when $\alpha = 0$, that is, when player 2 always selects v_2' . In this case, all profiles yield welfare of 5. \square

We now formally state the negative result demonstrated by the above example.

Proposition 8. *Let P be a combinatorial allocation problem. Then there exists an allocation algorithm g and player valuation functions v_i with $v_i(\emptyset) = 0$ for all players i such that every Bayes-Nash equilibrium yields strictly lower welfare than the strategy profile in which all players report their valuations truthfully.*

For the purposes of the above example, we had to construct a rather bizarre outcome function $g(t)$. An open question is whether some typical approximation algorithms have properties which do ensure that at least one equilibrium (or, ideally, all equilibria) is (are) welfare improving as compared to truthful reporting.

Appendix B. Proofs

Appendix B.1. Proof of Theorem 2

Let $t_i^* = \arg \max_{t_i} u_i(t_i, t_{-i})$. The most that the player can gain from deviating to t_i^* is

$$\begin{aligned}
& E_{t_{-i}} \left[\sum_j v_j(t_j, g(t_i^*, t_{-i})) - \sum_j v_j(t_j, g(t)) | t_i \right] \\
& \leq E_{t_{-i}} \left[\sum_j v_j(t_j, o^*(t_i^*, t_{-i})) - \sum_j v_j(t_j, g(t)) | t_i \right] \\
& \leq E_{t_{-i}} \left[\sum_j v_j(t_j, o^*(t_i^*, t_{-i})) - \frac{1}{\alpha(t)} \sum_j v_j(t_j, o^*(t)) | t_i \right] \\
& \leq E_{t_{-i}} \left[\sum_j v_j(t_j, o^*(t)) - \frac{1}{\alpha(t)} \sum_j v_j(t_j, o^*(t)) | t_i \right] \\
& \leq E_{t_{-i}} \left[\frac{\alpha(t) - 1}{\alpha(t)} V^*(t) | t_i \right].
\end{aligned}$$

Appendix B.2. Proof of Theorem 6

To prove this theorem, we first need the following lemma.

Lemma 9. *Let t_i be the type of player i and $t'_i \neq t_i$ be his deviation, and suppose that t'_i is sampled from T_i uniformly randomly. Then for any $q \geq 0$*

$$\Pr \left\{ \sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq q \right\} = \Pr \left\{ \sum_i v_i(t_i, g(t'_i, t_{-i})) \geq q \right\}.$$

Proof. Observe, first, that for any q, z and any $t'_i \neq t_i$,

$$\begin{aligned}
& \Pr \{ v_i(t_i, g(t'_i, t_{-i})) \geq q | v_i(t'_i, g(t'_i, t_{-i})) \geq z \} \\
& = \Pr \{ v_i(t_i, g(t'_i, t_{-i})) \geq q \},
\end{aligned}$$

since the value function itself is fixed. Consequently, letting

$$V_{-i}(t'_i) = \sum_{j \neq i} v_j(t_j, g(t'_i, t_{-i})),$$

we get

$$\begin{aligned}
& \Pr\{v_i(t_i, g(t'_i, t_{-i})) + V_{-i}(t'_i) \geq q | v_i(t'_i, g(t'_i, t_{-i})) + V_{-i}(t'_i) \geq z\} \\
&= E_{V_{-i}(t'_i)=w}[\Pr\{v_i(t_i, g(t'_i, t_{-i})) \geq q - w | v_i(t'_i, g(t'_i, t_{-i})) \geq z - w\}] \\
&= E_{V_{-i}(t'_i)=w}[\Pr\{v_i(t_i, g(t'_i, t_{-i})) \geq q - w\}] \\
&= \Pr\{v_i(t_i, g(t'_i, t_{-i})) + V_{-i}(t'_i) \geq q\}.
\end{aligned}$$

Now, observe that, given the definition of $g'()$,

$$\begin{aligned}
& \Pr\left\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq q\right\} \\
&= \Pr\left\{\sum_i v_i(t_i, g(t'_i, t_{-i})) \geq q | v_i(t'_i, g(t'_i, t_{-i})) + V_{-i}(t'_i) \geq z\right\},
\end{aligned}$$

and, consequently,

$$\Pr\left\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq q\right\} = \Pr\left\{\sum_i v_i(t_i, g(t'_i, t_{-i})) \geq q\right\}.$$

But, since both t'_i and t_i^* are drawn uniformly randomly,

$$\Pr\left\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq q\right\} = \Pr\left\{\sum_i v_i(t_i, g(t'_i, t_{-i})) \geq q\right\}.$$

□

We are now ready to prove the theorem. First, consider a select player i .

$$\begin{aligned}
& \Pr\{u_i(t_i, g'(t'_i, t_{-i}), p_i(t'_i, t_{-i})) > u_i(t_i, g'(t), p(t))\} \\
&\leq \Pr\{u_i(t_i, g'(t'_i, t_{-i}), p_i(t'_i, t_{-i})) \geq u_i(t_i, g'(t), p(t))\} \\
&= \Pr\left\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq \sum_i v_i(t_i, g'(t))\right\}.
\end{aligned}$$

First, consider the quantity $\Pr\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq q\}$ for some fixed q . By Lemma 9,

$$\Pr\left\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq q\right\} = \Pr\left\{\sum_i v_i(t_i, g(t'_i, t_{-i})) \geq q\right\}.$$

Hence,

$$\begin{aligned} & \Pr\left\{\sum_i v_i(t_i, g'(t'_i, t_{-i})) \geq \sum_i v_i(t_i, g'(t))\right\} \\ &= \Pr\left\{\sum_i v_i(t_i, g(t'_i, t_{-i})) \geq \sum_i v_i(t_i, g'(t))\right\}. \end{aligned}$$

Let $V'(t) = \sum_i v_i(t_i, g(t'_i, t_{-i}))$. By conditioning, we have

$$\begin{aligned} & \Pr\left\{\sum_i v_i(t_i, g(t'_i, t_{-i})) \geq \sum_i v_i(t_i, g'(t))\right\} \\ &= E_{V'(t)}[\Pr\left\{\sum_i v_i(t_i, g'(t)) \leq z \mid V'(t) = z\right\}] \\ &\leq E_{V'(t)}[\Pr\left\{\max_{t''_i \in T_i \setminus t_i} \sum_i v_i(t_i, g(t''_i, t_{-i})) \leq z \mid V'(t) = z\right\}] \\ &= E_{V'(t)}[\Pr\left\{\sum_i v_i(t_i, g(t''_i, t_{-i})) \leq z \mid V'(t) = z\right\}^L]. \end{aligned}$$

Letting $F(z)$ be the distribution over t'_i (it is identical for t''_i by construction), we have

$$\begin{aligned} & E_{V'(t)}[\Pr\left\{\sum_i v_i(t_i, g(t''_i, t_{-i})) \leq z \mid V'(t) = z\right\}^L] \\ &= \int_{\mathbb{R}} F(z)^L dF(z) = \frac{1}{L+1}. \end{aligned}$$

To conclude the proof, we only need to apply the union bound to obtain $\frac{n}{L+1}$ as the upper bound over all the players.

Appendix B.3. Proof of Theorem 7

First, observe that by the Dominated Convergence Theorem,

$$\lim_{L \rightarrow \infty} \int_{\mathbb{R}} G(u)^L dH(u) = \int_{\mathbb{R}} \lim_{L \rightarrow \infty} G(u)^L dH(u),$$

since both $G(u)$ and $H(u)$ are probability densities. Now,

$$\begin{aligned} \int_{\mathbb{R}} \lim_{L \rightarrow \infty} G(u)^L dH(u) &= \int_{\overline{U}_1} \lim_{L \rightarrow \infty} G(u)^L dH(u) \\ &\quad + \int_{U_1} \lim_{L \rightarrow \infty} G(u)^L dH(u) \\ &= \int_{\overline{U}_1} \lim_{L \rightarrow \infty} G(u)^L dH(u). \end{aligned}$$

Since for every $u \in \overline{U_1}$, $\lim_{L \rightarrow \infty} G(u)^L = 0$,

$$\int_{\mathbb{R}} \lim_{L \rightarrow \infty} G(u)^L dH(u) = \int_{\overline{U_1}} \lim_{L \rightarrow \infty} G(u)^L dH(u) = 0.$$