Game Theory and Algorithms*
Lecture 12: VCG Part 2: Fractional VCG

April 12, 2011

Summary: We finish up our introduction of the VCG mechanism, giving the proof of correctness, an example, some variants and some critiques. Then we talk about *fractional VCG*, which is one way of coping with the NP-completeness of VCG as it applies to combinatorial auctions.

1 VCG, Part 2

In the previous lecture we left the following fact to be proven later.

Lemma 1. *In a strategic game, if $\widehat{a}_i$ is weakly dominant for player i, and we alter the payoffs u_i for player i by*

$$u'_i(a) := u_i(a) + h(a_{-i})$$

where $h(a_{-i})$ is any function depending on a_{-i} but not on a_i, then $\widehat{a}_i$ is still weakly dominant in the new game.

Proof.

$$\begin{aligned}
\widehat{a}_i \text{ weakly dominant in the old game} &\Leftrightarrow \forall a_i, \forall a_{-i} : u_i(\widehat{a}_i, a_{-i}) \geq u_i(a_i, a_{-i}) \\
&\Leftrightarrow \forall a_i, \forall a_{-i} : u_i(\widehat{a}_i, a_{-i}) + h(a_{-i}) \geq u_i(a_i, a_{-i}) + h(a_{-i}) \\
&\Leftrightarrow \forall a_i, \forall a_{-i} : u'_i(\widehat{a}_i, a_{-i}) \geq u'_i(a_i, a_{-i}) \\
&\Leftrightarrow \widehat{a}_i \text{ weakly dominant in the new game.}
\end{aligned}$$

□

Recall VCG chooses $\mathfrak{a}^*$ which maximizes the social welfare $\sum_{i=1}^n \mathrm{bid}_i(\mathfrak{a}^*)$, and charges each player i the "damage" $\max_{\mathfrak{a}\in\mathfrak{A}} \sum_{j\neq i} \mathrm{bid}_i(\mathfrak{a}) - \sum_{j\neq i} \mathrm{bid}_i(\mathfrak{a}^*)$. Then the last part of the proof of VCG's truthfulness goes as follows.

Theorem 2. *In VCG, bidding your true value is weakly dominant.*

* Lecture Notes for a course given by David Pritchard at EPFL, Lausanne.

Proof. The utility of player i is $\text{value}_i(\mathfrak{a}^*)$ minus the payment of player i. The first term in player i's payment doesn't depend on player i's action (bid) so by Lemma 1, we can ignore that term. This leaves

$$u_i \cong \text{value}_i(\mathfrak{a}^*) + \sum_{j \neq i} \text{bid}_j(\mathfrak{a}^*). \tag{1}$$

The crucial point is that for player i, this is completely determined by several factors he cannot control $(\text{value}_i, \text{bid}_{-i})$ and only one factor he can indirectly control, $\mathfrak{a}^*$. Moreover, submitting $\text{bid}_i = \text{value}_i$ ensures that the optimal $\mathfrak{a}^*$ (the one maximizing (1)) is chosen, by the way that VCG chooses $\mathfrak{a}^*$. So $\text{bid}_i = \text{value}_i$ is indeed weakly dominant. □

This completes the proof that VCG is truthful. We also can verify two other naturally desirable properties of VCG.

Claim 3. *VCG as described above has each* $\text{payment}_i \geq 0$ *— the auctioneer does not pay players.*

Proof. The payment

$$\max_{\mathfrak{a} \in \mathfrak{A}} \sum_{j \neq i} \text{bid}_i(\mathfrak{a}) - \sum_{j \neq i} \text{bid}_i(\mathfrak{a}^*)$$

is nonnegative since $\mathfrak{a}^*$ is a possible choice of $\mathfrak{a}$. □

Exercise. If all bids are nonnegative, then each player has $\text{payment}_i \leq \text{bid}_i(\mathfrak{a}^*)$.

1.1 Example: Multiunit Auctions

Let us consider the case that we have k identical copies of an item to auction away, at most one to each player. The auctioneer must choose the k players which will win: the set $\mathfrak{A}$ of alternatives is

$$\mathfrak{A} = \{\text{all } k\text{-element subsets of } \{1, 2, \ldots, n\}\}.$$

We assume/require that each player's valuation/bid depends only on whether they win the item, and is 0 when they lose the item. E.g. for bids this means that instead of functions each player submits a single number b_i indicating their own bid for a single copy of the item which implicitly defines

$$\text{bid}_i(\mathfrak{a}) = \begin{cases} b_i, & i \in \mathfrak{a}; \\ 0, & i \notin \mathfrak{a}. \end{cases}$$

We will only analyze this under the assumption that each $b_i \geq 0$. This implies that players who do not win get 0 (since players pay a nonnegative amount but no more than their bid, and each player bids 0 on alternatives where they lose).

The VCG outcome $\mathfrak{a}^*$ is the one which maximizes

$$\sum_{i=1}^{n} \text{bid}_i(\mathfrak{a}^*) = \sum_{i \in \mathfrak{a}^*} b_i,$$

which means that the k largest bids will be chosen.

Payments for winners: the payment charged to a player i who wins a copy of the item ($i \in \mathfrak{a}^*$) is

$$\underbrace{\max_{\mathfrak{a}} \sum_{j \in \mathfrak{a}, j \neq i} b_j}_{\text{sum of top } k \text{ bids among players } \neq i} - \underbrace{\sum_{j \in \mathfrak{a}^*, j \neq i} b_j}_{\text{sum of top } k-1 \text{ bids among players } \neq i}$$

and this equals the $(k+1)$st largest bid.

1.2 Variants

Different "pivots". The VCG proof of truthfulness goes through even using any different "pivot function" h_{-i} in Lemma 1.

Non-uniqueness. VCG is not the only mechanism satisfying

- Truthfulness
- Payments are nonnegative
- Payments do not exceed bids.

An example which we will return to next class is adding a *reserve price* to an auction. Namely, consider a single-item auction (we studied this many weeks ago). We find that VCG is equivalent to a second-price auction. But here is another auction with all three properties. For a real number r modify the auction so that (i) if no bids are greater than r, nobody wins the item, and (ii) if only a single bidder bids above r, they win and pay r. We'll return to this idea next class to maximize auctioneer revenue in certain settings.

In certain situations a complete list of all mechanisms satisfying truthfulness and other basic properties is known. Knowing all such mechanisms is important since it tells you exactly how much flexibility the mechanism designer has, e.g. if they want other application-specific properties. Roberts' theorem gives one characterization:

Theorem 4. *Assume players can submit arbitrary bid functions $\mathfrak{A} \to \mathbb{R}$ and there are at least 3 players. Any truthful mechanism is VCG, with pivots, with generalized reserve prices and with player weights (an "affine maximizer").*

We intentionally skip some details, which appear in Section 9.5 of Nisan et al., but *player weights* resembles the following observation: the auctioneer can simply ignore some player and always charge them 0.

1.3 Critiques and Caveats

As mentioned above, in some applications VCG is not the unique truthful mechanism, so some other truthful mechanism might be better in some ways. Also if we have a different solution concept than truthfulness, for example *randomized* mechanisms which have to be *truthful in expectation*, even more alternatives to VCG become possible.

VCG is not *group-strategyproof*. We mentioned an example of this last lecture in the case of auctioning a single item to 2 players: if they collaborate (by dropping the loser's bid to 0) they can increase the winner's utility without hurting the loser's utility.

A major problem with applying VCG to auctions is that the problem of computing $\max_{\mathfrak{a}\in\mathfrak{A}} \sum_i \mathrm{bid}_i(\mathfrak{a})$ can be NP-complete, i.e. computationally complex. (In fact, the first bump is that specifying a bid for every subset of items the naive way takes an exponentially large bid, but this a general issue in combinatorial auctions that has nothing to do with VCG.) The NP-completeness happens even if we restrict the bids to have polynomial size; we compare and contrast two examples.

Example 5. Take an auction with n players and m items, and suppose each player is allowed to win a maximum of 1 item. Thus each player's bid consists of m real numbers, one per item.

If we want to run VCG on this auction, we need to compute a valid allocation of items to players (at most one item per player) maximizing the social welfare. This is identical to the *maximum-weight bipartite matching problem*, which is solvable in polynomial time, and which we introduce by example. Draw a graph with n player nodes on the left, and m item nodes on the right. Draw an edge between player i and item j with weight equal to player i's bid for item j. We want to find a set of edges (assignment) of maximum total weight, so that every node has degree at most 1 (at most one player per item and at most one item per player).

Computing the payments can be modelled by a similar max-weight bipartite matching calculation.

Example 6 (Pair auctions)**.** Take an auction with n players and m items, and this time suppose each player is allowed to win a maximum of 2 items. Thus each players' bid consists of $\binom{m}{2} + m$ real numbers, one per pair and individual item.

This time, determining the optimal valid allocation of items is NP-complete since it encompasses the *max-weight 3D matching* problem, which is: given three disjoint ground sets, and weighted triples consisting each of one item per ground set, find the maximum-weight collection of triples which are pairwise disjoint. (To formally prove the reduction, we take a hard 3D matching instance and let one ground set be the players and the other two ground sets become items in the auction.)

One might try using an *approximation algorithm* to approximately optimize the social welfare. For some $0 < \alpha \leq 1$, an *α-approximation algorithm* always computes an alternative/allocation from $\mathfrak{A}$ whose social welfare is at least α times the maximum possible social welfare. The superficial approach is to have "approximate-VCG" return the $\mathfrak{a}$ specified by the approximation algorithm, and again use the approximation algorithm to compute prices. The problem is that this approach is not truthful.

2 Fractional VCG

We spend the rest of today's lecture talking about an approach to ameliorate the NP-completeness which occurs above, when we apply VCG to combinatorial auctions. The main result can be stated as:

Theorem 7 (Lavi and Swamy, 2005 [3]). *As long as we have an* LP-relative *α-approximation algorithm to maximize the social welfare, we get a mechanism which is*

- *Truthful in expectation, and*
- *α-approximately optimizes the social welfare in expectation.*

The mechanism in this theorem will use VCG as a subroutine. It applies to pair auctions, in which case it is known that we can take $\alpha = \frac{3}{7}$ [2].

These new terms have the following meaning:

- *Truthful in expectation* means the mechanism is randomized, and bidding the true valuation always maximizes the *expected* utility of each player, no matter what the other players do.
- *α-approximately optimizes the social welfare in expectation* means that the expected value of the social welfare of the alternative (allocation) $\mathfrak{a}$ it chooses is at least α times the optimal one.
- *LP-relative approximation* is a more complex idea taken from the field of approximation algorithms. It relates to the mantra of LP-based approximation algorithms: even though optimizing over a certain discrete set may be NP-complete, it is often possible to "enlarge" the set to a continuous one while making it optimizable in polynomial time via linear programming. Then, an "LP-relative α-approximation algorithm" is one which computes an integral solution which is at least α times the LP optimum. This implies it is an α-approximation algorithm since

$$\text{OUTPUT} \geq \alpha \cdot \text{LP-OPT} \geq \alpha \cdot \text{ACTUAL-OPT}.$$

Theorem 7 is interesting since for problems like pair auctions, it gives an auction which is truthful (in a weaker sense than before) but still VCG-like. Approximately optimizing the social welfare is important since we want to preclude useless truthful mechanisms like "allocate no items and charge no payments."

2.1 Fractional Allocations

A *fractional allocation* has a variable $x_{i,S}$ for each player i and each set S of items that player i could win. These are variables between 0 and 1. We think of $x_{i,S} = 1$ as meaning that player i wins item set S.

Definition 8. A fractional allocation is *feasible* if for each player i, $\sum_{i \in S} x_{i,S} = 1$, and for each item j, $\sum_i \sum_{S:j \in S} x_{i,S} \leq 1$.

Observe that if a fractional allocation is feasible and also has integral values, it corresponds to a valid allocation of items to players.

Definition 9. The *social welfare* of a fractional allocation is $\sum_i \sum_S x_{i,S} \text{bid}_i(S)$.

Again, in the special case of an allocation with integral values, this gives the allocation's ocial welfare (like what we studied before in VCG).

Observation 10. *Assume that for each player i there are at most polynomially many sets S they could win. Then the problem of finding a feasible fractional allocation with maximal social welfare is just a polynomial-size linear program.*[1]

This observation is part of the general mantra of LP-based approximation algorithms: even though optimizing over a certain discrete set may be NP-complete, it is often possible to "enlarge" the set to a continuous one while making it optimizable in polynomial time via linear programming.

2.2 Approximate Decompositions

Next, we want to make a connection between fractional allocations and probability distributions over integral allocations.

Definition 11. For a feasible fractional allocation x, a *decomposition of αx* is a collection of pairs $(\lambda_j, \mathfrak{a}_j)$ such that

1. The λ_j are nonnegative and sum to 1.
2. Each $\mathfrak{a}_j$ is a valid allocation of items to users; let $\mathfrak{a}_j(i)$ denote the set of items allocated to player i.
3. For each player i and each set S,

$$\sum \{\lambda_j \mid \mathfrak{a}_j(i) = S\} = \alpha \cdot x_{i,S}.$$

In other words, a decomposition is a probability distribution over (integral) allocations, so that for each player i the overall probability they get S is $\alpha \cdot x_{i,S}$.

The key technical component in the Lavi-Swamy approach is the following:

Theorem 12. *If we have an* LP-relative *α-approximation algorithm to find the (integral) social optimal allocation, then we can compute a decomposition of αx for the optimal fractional allocation x. Moreover, the number of pairs is polynomial.*

This builds on a similar result of Carr-Vempala [1], which in turns relies on the ellipsoid algorithm. This family of results is very useful in LP-based approximation! We will just assume Theorem 12 is true, and then show how this gives the randomized mechanism. We remark that the converse of Theorem 12 is also true — and much easier to prove.

[1]The Lavi-Swamy approach also works in settings where players can win a super-polynomial number of sets if we assume that bids are given by a *demand oracle* that we can query polynomially many times.

2.3 The Mechanism

1. We compute the optimal fractional allocation x.
2. Using Theorem 12 we obtain a decomposition $\{(\lambda_j, \mathfrak{a}_j)\}_j$ of $\alpha \cdot x$.
3. With each probability λ_j, we allocate the items according to $\mathfrak{a}_j$.
4. Charge each player i the "fractional damage" they cause:

$$\text{payment}_i = \underbrace{\alpha \cdot (\text{maximum social welfare } \textit{for players} \neq i \text{ among } \textit{fractional} \text{ allocations})}_{\text{(deterministic)}} - \underbrace{\text{social welfare of players} \neq i \text{ under the chosen outcome } \mathfrak{a}_j}_{\text{(randomized)}}.$$

The equality comes from the definition of *decomposition.* As before, the expected payment_i is nonnegative and does not exceed $\text{bid}_i(\mathfrak{a}_j)$, in expectation.

Finally, to prove that this mechanism is truthful in expectation, we proceed more or less like we did before, but carefully. First, we can ignore the deterministic term in the payment, since it does not depend on player i. Then, the modified expected utility for player i becomes

$$\sum_j \lambda_j(\text{value}_i(\mathfrak{a}_j(i)) + \sum_{k \neq i} \text{bid}_k(\mathfrak{a}_j(k)))$$
$$\underbrace{=}_{\text{(by definition of decomposition)}} \alpha \left(\sum_S x_{i,S} \cdot \text{value}_i(S) + \sum_{k \neq i} \sum_S x_{k,S} \cdot \text{bid}_k(S) \right).$$

Player i can only affect this indirectly, by altering the algorithm's choice of x. Moreover, if $\text{bid}_i = \text{value}_i$ then this is α times the optimal fractional social welfare, which is what the mechanism seeks to maximize. Thus bidding truthfully is weakly dominant in expectation.

A closing remark: the Hartline-Lucier paper on the course webpage is another example where approximation algorithms (average-case ones) are methodically transformed into mechanisms.

References

[1] R. D. Carr and S. Vempala. Randomized metarounding. *Random Struct. Algorithms*, 20(3):343–352, 2002. Preliminary version in STOC 2000.

[2] Y. H. Chan and L. C. Lau. On linear and semidefinite programming relaxations for hypergraph matching. In *Proc. 21st SODA*, pages 1500–1511, 2010.

[3] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proc. 46th FOCS*, pages 595–604, 2005.