Combinatorial auctions and the configuration LP

Uriel Feige

9 July, 2024

1 Introduction

Topics discussed in this lecture are covered in part in [5], chapter 11, and other references given at the end. Students are advised to consult these references.

The notes here were made for my own use, to remind me during the lecture what to cover, and to remind me in the future what was covered.

2 Combinatorial auctions

m indivisible items and n bidders.

Type (valuation function) of a bidder: a nonnegative monotone normalized set function.

Goals for allocation.

- *Economic efficiency.* We shall consider maximizing the social welfare of allocation: $\sum_{i=1}^{n} v_i(S_i)$.
- *Incentives.* It should be in the best interest of agents to reveal their true valuations, allowing the mechanism to truly maximize welfare. The allocation mechanism may charge payments. Bidders are assumed to have quasi-linear utility.
- *Complexity.* We want the mechanism to have polynomial communication and computation complexity.

Recall the VCG mechanism: pay the loss in welfare that you caused.

Complexity: 2^m in reporting v. Not practical for large m.

Single minded. Each player i has value $v_i > 0$ only for a specific set S_i.

Now reporting is easy. However, computing maximum welfare allocation is NP-hard.

3 Access to valuation functions

As the valuation functions might have exponential representation, we assume a query model. One may ask for a given class of valuations (additive, submodular, etc.) which types of queries (if any) suffice for computing an allocation that maximizes welfare.

The most basic query model is **value queries**.

Another query model that we will find very useful: **demand queries**. Given prices to items as a query, the reply is the bundle maximizes utility (breaking ties arbitrarily).

It is natural to expect an agent to be able to answer demand queries on her own valuation function. Or put differently, if agents do not know what they want to buy given the prices of items, it might not be realistic to expect an allocation mechanism to be able to find an allocation that maximizes welfare.

Answering demand queries might be NP-hard even for coverage valuation functions. Example: items are vertices of a graph, and the value of a set is the number of incident edges (vertex cover). If every vertex i (with degree d_i) is priced at $d_i - \frac{1}{2}$, the answer to the demand query is a maximum independent set.

One may imagine other types of queries (e.g., MMS queries, asking an agent her MMS partition), but we will focus on value queries and demand queries.

3.1 The Configuration LP

Often it is convenient to first solve a fractional relaxation of an integer problem, and then see how to round it to an integer solution. A very useful linear programming relaxation of the maximum welfare problem is the configuration LP (CLP). In an integer allocation, $x_{i,S}$ is a variable indicating whether the bunde received by agent i is S. In the configuration LP, $x_{i,S}$ may have fractional values.

maximize $\sum x_{i,S} v_i(S)$ (maximize welfare) subject to:

- $\sum_S x_{i,S} \leq 1$ for every agent i (an agent gets at most one bundle).
- $\sum_{i,S|j \in S} x_{i,S} \leq 1$ for every item j (an item is allocated at most once).
- $x_{i,S} \geq 0$.

For each agent, the solution gives a probability distribution over bundles. Under the joint (product) distribution, each item is allocated at most once in expectaton.

In general, it is useful to see the dual of an LP, and even more so in situations such as CLP, in which the number of variables is exponential, but the number of constraints is polynomial.

The dual:

minimize $\sum_{i \in N} u_i + \sum_{j \in M} p_j$ subject to:

- $u_i + \sum_{j \in S} p_j \geq v_i(S)$ for every i, S.

- $u_i, p_j \geq 0$.

Interpretation: p_j are item prices. u_i are utilities of the players from the optimal bundles for them. By complementary slackness, $x_{i,S} > 0$ only if the corresponding constraint in the dual is tight, meaning that S is a bundle most preferred by i at these prices. Also, if $u_i > 0$, player i gets a full bundle ($\sum_S x_{i,S} = 1$).

The dual of CLP can be solved using the ellipsoid algorithm. To implement the ellipsoid algorithm requires identifying whether a tentative solution has a violated constraint. If the dual has a violated constraint, demand queries can find it. Each agent is given the vector of currrent prices, and is asked to report S that maximizes $v_i(S) - \sum_{j \in S} p_j$. If this value is larger than u_i, this discovers a violated constraint. (The value $v_i(S)$ can be asked for using a value query.)

Given that the dual can be solved, standard techniques can be used in order to solve also the primal CLP. The basic idea is to keep in the primal only polynomially many of its $x_{i,S}$ variables – those whose corresponding constraint $u_i + \sum_{j \in S} p_j \geq v_i(S)$ in the dual is discovered to be tight during the run of the ellipoid algorithm. Further details are omitted.

3.2 Walrasian equiblirium

A Walrasian equilibrium is a vector of non-negative prices for the items and an allocation of all items, such that for each agent, the allocated bundle (which might be empty) is one that maximizes her utility given the prices. Stating it in economic terms, *the market clears*. In general, Walrasian equilibria need not exist (e.g, single minded bidders).

An integral solution to the configuration LP is a Walrasian equilibrium (with prices as in the dual solution). For general classes of valuations, the CLP need not have an integral solution. But for some interesting classes of valuations it does. This includes additive valuations (each item is priced as in a second price auction), and concave valuation functions over identical items. A class of valuations that is a sense characterizes Walrasian equilibrium is **Gross Substitutes** (GS). Given any set S and any three items not in S, there are two ways of partitioning the three items to a pair and a singleton in such a way that maximizes $v(S \cup \{a, b\}) + v(S \cup \{c\})$. Examples of GS valuations are additive, matroid rank functions (e.g., rank of a set of vectors), unit demand (the value of a set of items is that of the maximum valued item in the set), and more.

If a Walrasian equiblirium exists, it satisfies the complementary slackness conditions of the configuration LP, and hence it gives a pair of primal and dual solutions to CLP that match each other, and hence are optimal for CLP. This implies that a Walrasian equilibrium maximizes welfare. It is also envy free (given the prices).

It is in general not incentive compatible to report true valuations to a mechanism that computes a Walrasian equilibrium – pricing each item separately (as done in a Walrasian equilibrium) does not give VCG prices. Consider m identical items and two agents. $v_1(S) = |S|$ and $v_2(S) = \min\{m, 2|S|\}$. The Walrasian prices are 1 for each item. Reporting $v_1'(S) = \frac{|S|}{2}$, the prices drop.

3.3 Other classes of valuation functions

Recall that we assume that set functions (valuations) are normalized and monotone.

We are interested in identifying classes of valuations for which we can approximate the maximum welfare within a constant ratio in polynomial time. Having complements among items (as is the case for a single minded bidders) is an obstacle towards this. Consequently, one often focuses of classes of valuations that are "complement-free". A hierarchy of such valuations was proposed in [4]. At the bottom of this hierarchy we may place GS, for which welfare can be maximized exactly. The next levels are **submodular**, **XOS** (also referred to as *fractionally subadditive*), and **subadditive**.

Submodular functions are set functions that have the decreasing marginal value property.For example, if all items are identical, then they are concave. a

Marginal value of item x with respect to set S: $v(x|S) = v(x \cup S) - v(S)$.

Submodular: for every $S \subset T$ and $x \notin T$ it holds that $v(x|S) \geq v(x|T)$.

Equivalently, $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$.

Examples: GS, budget additive, coverage function (the last two classes are not GS). Entropy function of a set of random variables (linear if they are independent).

Additional classes.

Definition 3.1 *A set function is* subadditive *if $f(S)+f(T) \geq f(S \cup T)$ for disjoint S and T. A set function is fractionally subadditive (also referred to as XOS) if for every set T, and for every $S_1, \ldots, S_k$ and nonnegative coefficients $\alpha_1, \ldots, \alpha_k$, if $\sum \alpha_i S_i \geq T$ then $\sum \alpha_i f(S_i) \geq f(T)$.*

Fractional subadditive functions are subadditive. Submodular functions are fractionally subadditive. Consider an arbitrary order over the items of T, and sets S_i as in Definition 3.1. The marginal value of an item j in π is always at least as large as it is in any of the S_i (by submodularity). As the α_i are such that j is selected at least once, the proof follows by linearity of expectation.

XOS is defined as a maximum over additive valuations, and is known to be equivalent to fractionally subadditive [1].

The three classes are different, by example with three items in which each item has value 1, and the grand the bundle has value 2. If every pair of items has value 1 the function is subadditive, if the value is 4/3 it is also fractionally subadditive, and if the value is 3/2 it is also submodular.

3.4 Approximately maximizing welfare

Even if every valuation function involves only a constant number of items, welfare maximization with submodular valuation functions is APX-hard (meaning that there is some constant $\rho < 1$ such that achieving a ρ approximation is NP-hard). Reduction from 3-coloring a 5-regular graph. For some $\epsilon > 0$, it is NP-hard to distinguish between graph that are legally three colorable, and those for which every 3-coloring results in at least ϵm monochromatic edges.

Given a 5-regular graph, construct the following maximum welfare instance. Each edge is three items, one of each color. There are m edge players, where each edge player receives value 1 if she receives at least one of the three items associated with its edge. There are n vertex players, who want only items associated with edges incident with their vertices. Any $k \leq 4$ such items give them value k, five such items give value 5 if they have the same color and 9/2 otherwise, and six or more items give value 5. If the graph has a legal 3-coloring, the maximum welfare is m. If there is no 3-coloring, the maximum welfare is $(1 - \Omega(\epsilon))m$.

One the positive side, here are the approximation ratios that are known. For submodular valuation functions and using only value queries: a $1 - \frac{1}{e} \simeq 0.632$ ratio is achieved in [6]. It is NP-hard to do better in some cases (when the valuation function is simple enough to answer value queries, but too complex to answer demand queries). See [3].

Other approximation algorithms use the configuration LP, and hence use also demand queries, and not only value queries.

For submodular valuatons, there exists some fixed $1 - \frac{1}{e} < \delta < 1$ such that an approximation ratio of δ is achievable. See [2].

For XOS valuations, an approximation ratio of $1 - \frac{1}{e}$ is achievable and best possible.

For subadditive valuations, an approximation ratio of $\frac{1}{2}$ is achievable and best possible [1].

A standard way of rounding the solution of CLP to an integer solution is by a two phase scheme. In the first phase, each agent independently selects at most one bundle, with probability proportional to $x_{i,S}$. In expectation, the value of this tentative assignment equals the value of LP. However, the assignment is illegal, because the same item might be in the selected bundles of several agents. In the second phase, we use a *contention resolution* rule to decide which agent gets those items that are under contention. For XOS, the contention rule is such that for each item in the agent's tentative set, the probability that the agent keeps it is at least $1 - \frac{1}{e}$ (probability taken both over randomness of the contention resolution rule, and over choice of tentative sets of other agents). For subadditive valuations, the contention resolution rule is more complicated to explain. For submodular valuations, to improve over $1 - \frac{1}{e}$, a more complicated rounding was needed, in which each agent select two tentative sets rather than just one.

If the class of valuations is such that valuations have polynomial size representations and we can exactly maximize welfare, we can use the VCG mechanism (in polynomial time) to incentivize truthful reporting of valuations. If valuations do not have polynomial representations but we can use queries in order to maximize welfare (which is the case for GS, we note that answering demand queries for GS valuations can be done in polynomial time), then the implementation is not in dominant strategies, but is incentive compatible in some weaker sense. If we can only approximately maximize welfare, the VCG mechanism cannot be implemented. In particular, it is a major open question to determine there is an IC mechanisms that recovers a consant fraction of the welfare when valuations are submodular.

References

[1] Uriel Feige: On Maximizing Welfare When Utility Functions Are Subadditive. SIAM J. Comput. 39(1): 122–142 (2009).

[2] Uriel Feige, Jan Vondrak: The Submodular Welfare Problem with Demand Queries. Theory of Computing 6(1): 247–290 (2010).

[3] Subhash Khot, Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. In WINE, 2005.

[4] Benny Lehmann, Daniel J. Lehmann, Noam Nisan: Combinatorial auctions with decreasing marginal utilities. Games and Economic Behavior 55(2): 270–296 (2006).

[5] Noam Nisan, Tim Roughgarden, Eva Tardos and Vijay V. Vazirani (Editors), Algorithmic Game Theory, Cambridge University Press, 2007. (Chapter 11.)

[6] Jan Vondrak: Optimal approximation for the submodular welfare problem in the value oracle model. STOC 2008: 67–74.