COMP323 – Introduction to Computational Game Theory

Auctions and Sponsored Search: Special Topics

Paul G. Spirakis

Department of Computer Science
University of Liverpool

Outline

1. Dynamic aspects of sponsored search
2. Combinatorial auctions

Sponsored search

Recall:

- Web search engines monetize their service by auctioning off advertising space next to their standard algorithmic search results, a practice known as sponsored search.
- Offline, the price for advertising is typically set by negotiation or posted price.
- Online, much advertising is sold via auction.

$\Rightarrow$ We will formally model and analyze various mechanisms used in this domain and study potential improvements.

Existing models and mechanisms

In sponsored search mechanisms:

1. The advertisers specify a list of pairs of keywords and bids as well as a total maximum daily or weekly budget.
2. Every time a user searches for a keyword, an auction takes place among the set of interested advertisers who have not exhausted their budgets.

Existing models and mechanisms

Focusing on a single auction:

- Let n be the number of bidders and $m < n$ the number of slots.
- The search engine estimates the clickthrough rate a_{ij}, the probability that a user will click on the ith slot when it is occupied by bidder j.
- It is usually presumed for all j that $a_{ij} \geq a_{i+1,j}$ for $i = 1, \ldots, m-1$ (this allows for more refined equilibrium analyses).
- The search engine also assigns a weight w_j to each advertiser j (thought of as a relevance or quality metric).

Existing models and mechanisms

- If agent j bids b_j, his corresponding score is $s_j = w_j b_j$.
- The search engine allocates slots in decreasing order of scores, so that the agent with highest score is ranked first, and so on.
- We assume that agents are numbered so that agent j obtains slot j.
- An agent pays per click the lowest bid necessary to retain his position, so that the agent in slot j pays s_{j+1}/w_j.
- This weighted bid ranking mechanism includes the two most prominent keyword auction designs that have been used in practice:
  1. rank by bid mechanism ($w_j = 1$);
  2. rank by revenue mechanism ($w_j = a_{1j}$).

Dynamic aspects

- Sponsored search auctions are repeated with great frequency.
- The online nature of the auctions in sponsored search complicates the computation of an efficient allocation.
- A proper model would be a repeated game of incomplete information.
- The set of equilibria of such games is quite rich and complicated.
- We mention two phenomena that arise in this setting:
  1. bid rotation and
  2. vindictive bidding.

Dynamic aspects

Bid rotation

Bid rotation:

- Competing bidders take turns at winning the auction.
- This means that bidders take turns at occupying the top slot.
- If bidders are short lived, this is unlikely to be a problem, if not, this will lower the auctioneer's revenue.

Dynamic aspects

Vindictive bidding

Vindictive bidding:

- In the GSP auction one's bid determines the payment of the bidder in the slot above and not one's own.
- Therefore one can increase the payment of the bidder in the slot above by raising one's bid without affecting one's own payment.
- This may be beneficial if the bidder in the slot above is a competitor with a limited budget for advertising.
- In a dynamic environment this encourages a bidder to constantly adjust their bids so as to inflict or avoid damage upon or from their competitor.

The online allocation problem

In the online allocation model:

- The search engine receives the bids of advertisers and their maximum budget for a certain period (e.g., a day).
- As users search for these keywords during the day, the search engine assigns their advertisement space to advertisers and charges them the value of their bid for the impression of the advertisement.
- For simplicity of notation we assume that each page has only one slot for advertisements.
- The objective is to maximize total revenue while respecting the budget constraint of the bidders.

Note:

- Bidders pay their bid which is counter to practice.
- Budget constraints, a real-world feature, apply across a set of keywords.

The online allocation problem

Formally:

- Let n be the number of advertisers and m the number of keywords.
- Suppose that advertiser j has a bid of b_{ij} for keyword i and a total budget of B_j.
- In this context, it is reasonable to assume that bids are small compared to budgets, i.e., $b_{ij} << B_j$.

If the search engine has an accurate estimate of r_i, the number of people searching for keyword i for all $1 \leq i \leq m$, then it is easy to approximate the optimal allocation using a simple linear program.

LP for online allocation

Let x_{ij} be the total number of queries on keyword i allocated to bidder j. The LP is

$$\begin{array}{lll} \max & \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij} & \\ \text{s. t.} & \sum_{j=1}^{n} x_{ij} \leq r_i & 1 \leq i \leq m \\ & \sum_{i=1}^{m} b_{ij} x_{ij} \leq B_j & 1 \leq j \leq n \\ & x_{ij} \geq 0 & 1 \leq i \leq m,\ 1 \leq j \leq n \end{array}$$

Dual LP for online allocation

The dual LP is

$$\begin{aligned} \min \quad & \textstyle\sum_{j=1}^{n} B_j \beta_j + \sum_{i=1}^{m} r_i \alpha_i & \\ \text{s. t.} \quad & \alpha_i + b_{ij}\beta_j \geq b_{ij} & 1 \leq i \leq m,\ 1 \leq j \leq n \\ & \beta_j \geq 0 & 1 \leq j \leq n \\ & \alpha_i \geq 0 & 1 \leq i \leq m \end{aligned}$$

Optimal solution for online allocation

Complementary slackness

- By complementary slackness, in an optimal solution, advertiser j is assigned to keyword i if $(1-\beta_j)b_{ij} = \max_k(1-\beta_k)b_{ik}$.
- Using this property, the search engine can use the solution of the dual linear program to find the optimum allocation:

Every time a user searches for keyword i, the search engine allocates its corresponding advertisement space to the bidder j with the highest $b_{ij}(1-\beta_j)$.

In other words, the bid of advertiser j will be scaled down by $1-\beta_j$.

Optimal solution for online allocation

- β_j represents the rate of change of the optimal objective function value of the LP for a sufficiently small change in the right-hand side of the corresponding constraint.
- If advertiser j's budget were to increase by Δ, the optimal objective function value would increase by $\beta_j \Delta$. It is the opportunity cost of consuming j's budget.
- Hence, if we allocate keyword i to agent j now we obtain an immediate "payoff" of b_{ij}.
- This consumes b_{ij} of the budget, which imposes an opportunity cost of $\beta_j b_{ij}$.
- Therefore, it makes sense in the optimal solution to assign keyword i to j provided $b_{ij} - \beta_j b_{ij} > 0$.

Dynamic procedures for online allocation

- In practice, a good estimate of the frequencies of all search queries is unavailable.
- Queries arrive sequentially; the search engine must instantly decide to allocate slots without knowledge of the future.
- A dynamic procedure is needed for allocating bidders to keywords that are queried.

We will describe one such dynamic procedure and analyze its performance within the competitive ratio framework:

- We compare the revenue achieved by a dynamic procedure that does not know the r_i's in advance, with the revenue that could be achieved knowing the r_i's in advance.
- The revenue in the second case is given by the optimal objective function value of the LP for online allocation.

A greedy dynamic procedure

The obvious dynamic procedure to consider is a greedy one:

Greedy dynamic procedure for online allocation

Among the bidders whose budgets are not exhausted, allocate the query to the one with the highest bid.

This greedy procedure:

- is equivalent to setting all β_j's to 0;
- is not guaranteed to find the optimal solution.

Sub-optimality of the greedy procedure

A simple example with two bidders and two keywords in which the revenue of the greedy algorithm is half the optimum:

- Each of two bidders has a budget of \$2.
- Assume $b_{11} = 2, b_{12} = 2 - \epsilon$, $b_{21} = 2$, and $b_{22} = \epsilon$.
- If query 1 arrives before query 2, it will be assigned to bidder 1. Then bidder 1's budget is exhausted.
- When query 2 arrives, it is assigned to bidder 2. This produces an objective function value of $2 + \epsilon$.
- The optimal solution would assign query 2 to bidder 1 and query 1 to bidder 2, yielding an objective function value of 4.

The problem with the greedy algorithm is that, unlike the solution to the LP, it ignores the opportunity cost of assigning a query to a bidder.

Sub-optimality of the greedy procedure

- One can prove that the revenue of greedy algorithm is at least half of the optimum revenue for any instance.
- In the standard terminology of online algorithms, the competitive ratio of greedy algorithm is $1/2$.

  Q: Can one do better in terms of competitive ratio?

  A: Yes: by trying to dynamically estimate the opportunity cost, i.e., the β_j's, of assigning a query to a bidder.
  - This has the effect of spreading the bidders expenditures over time (budget smoothing).
  - It is a feature that some search engines offer their advertisers.

An adaptive algorithm

- The following modification of the greedy algorithm adaptively updates the β_j's as a function of the bidders spent budget. Let

$$\phi(x) = 1 - e^{x-1} \ .$$

- The algorithm sets $\beta_j = 1 - \phi(f_j)$, where f_j is the fraction of the budget of bidder j, which has been spent.

Adaptive algorithm for online allocation

Every time a query i arrives, allocate its advertisement space to the bidder j, who maximizes $b_{ij}\phi(f_j)$, where f_j is the fraction of the bidder j's budget which has been spent so far.

The competitive ratio of the adaptive algorithm

- The revenue of this algorithm is at least $1 - 1/e$ of the optimum revenue.
- It is also possible to prove that no deterministic or randomized algorithm can achieve a better competitive ratio.

Theorem

The competitive ratio of the adaptive algorithm is $1 - 1/e$.

The competitive ratio of the adaptive algorithm

Proof.

- Let k be a sufficiently large number used for discretizing the budgets of the bidders.
- An advertiser is of type j if she has spent within $\left(\frac{j-1}{k}, \frac{j}{k}\right]$ fraction of her budget so far.
- Let s_j be the total budget of type j bidders.
- For $i = 0, 1, \ldots, k$, define w_i to be the amount of money spent by all the bidders from the interval $\left(\frac{i-1}{k}, \frac{i}{k}\right]$ of their budgets.
- Also define the discrete version of function ϕ:

$$\Phi(s) = 1 - \left(1 - \frac{1}{k}\right)^{k-s} .$$

- When $k \to \infty$, then $\Phi(s) \to \phi(s/k)$.

The competitive ratio of the adaptive algorithm

Proof (continued).

- Let OPT be the solution of the optimal off-line algorithm (i.e., the solution of the LP).
- For simplicity, assume that the optimal algorithm spends all of the budget of the bidders.
- Consider the time that query q arrives. Suppose that OPT allocates q to a bidder of current type t, whose type at the end of the algorithm will be t'.
- Let b_{opt} and b_{alg} be the amount of money that OPT and the algorithm get from bidders for q.
- Let i be the type of the bidder that the algorithm allocates the query. We have

$$\Phi(t')b_{opt} \leq \Phi(t)b_{opt} \leq \Phi(i)b_{alg} \ .$$

The competitive ratio of the adaptive algorithm

Proof (continued).

- $\Phi(t')b_{opt} \leq \Phi(t)b_{opt} \leq \Phi(i)b_{alg}$.
- Summing over all queries, it follows that at the end of the algorithm,

$$\sum_{i=0}^{k} \Phi(i)s_i \leq \sum_{i=0}^{k} \Phi(i)w_i \ .$$

- By definition $w_i \leq \frac{1}{k}\sum_{j=i}^{k} s_j$ so

$$\sum_{i=0}^{k} \Phi(i)s_i \leq \frac{1}{k}\sum_{i=0}^{k} \Phi(i)\sum_{j=i}^{k} s_j \ .$$

The competitive ratio of the adaptive algorithm

Proof (continued).

- Changing the order of the sums and computing the sum of the geometric series, it follows that

$$\left(\Phi(0) - O\left(\frac{1}{k}\right)\right)\sum_{i=0}^{k} s_i \leq \sum_{i=0}^{k} \frac{i}{k} s_i \ .$$

- As $k \to \infty$ the left-hand side tends to $(1 - 1/e)\mathrm{OPT}$.
- The right-hand side is equal to the revenue of the algorithm.

So the competitive ratio follows. □

Open questions on sponsored search

The role of the budget constraints:

- In many cases they do not appear to be hard constraints as bidders frequently adjust them.
- A bidder can also "expand" their budget simply by lowering their bid and paying less per click.
- They provide a convenient way to express other desires, e.g., limiting one's exposure or spreading one's advertising over a longer period.

Need for richer bidding models:

- Allow bidders to express decreasing marginal value for clicks.
- Allow for distinct values for traffic from certain geographic regions, demographic profiles, etc.

Open questions on sponsored search

Advertiser payments:

When advertiser payments are based on user clicks, search engines must invest in the task of detecting and ignoring

- robot clicks;
- spam clicks;
- clicks from an advertiser trying to impose costs on their competitor;
- clicks from an affiliate who actually benefits monetarily from additional clicks.

Need for alternate pricing conventions:

- The most compelling is **pay per action or conversion**: the advertiser pays only if a click results in a sale, for example.
- This raises new incentive issues associated with tracking sales.

Open questions on sponsored search

Monopoly search engine:

- Current models assume a monopoly search engine with a static user base: works if switching costs for advertisers and users were high.
- But switching costs for advertisers are low: many work with both Google and Yahoo! simultaneously, or work with third-party search engine marketers.
- Switching costs for users are zero: to patronize a different search engine, users need merely type a new address into their web browser.

Need to study keyword auctions in competition with each other:

- Firms should focus less on extracting the maximum revenue from advertisers possible and more on attracting and retaining advertisers.
- Search engines must make trade-off decisions between maximizing current revenue and attracting and retaining users in the long term.

Open questions on sponsored search

Affiliations:

- The major search engines syndicate their advertisements to affiliate search engines and content providers.
- E.g., Google, through its AdSense program, syndicates advertisements to AOL, MySpace, and thousands of other Web sites.
- The introduction of affiliates greatly complicates the semantics of bidding and allocation.

Open questions on sponsored search

Clickthrough rates:

- They are assumed to be given.
- In practice, they are learned over time and can depend on a variety of factors:
  - bidder identity;
  - advertisement identity and content;
  - user characteristics (demographics, location, history);
  - page context (other advertisements and algorithmic search results).

Need for learning clickthrough rates:

- Explore/exploit trade-off: the auctioneer can exploit known high-rate advertisements, or explore new advertisements or infrequently shown advertisements to uncover even higher-rate advertisements.
- The auctioneer's rate estimate may differ from the bidder's estimate: the auctioneer usually has more contextual information to learn from.

Open questions on sponsored search

Focus on the advertiser:

- We have focused on the auctioneer's mechanism design problem.
- The advertiser's bidding optimization problem is also challenging.
- It is currently the focus of a great deal of commercial and research activity.

Combinatorial auctions

In a combinatorial auction (CA)

- a large number of items are auctioned concurrently
- bidders are allowed to express preferences on bundles of items.

Interesting case: the valuation of a given set of resources is different from the sum of valuations of each resource separately.

- E.g., when we have a set of complementary products: each product alone is useless but the group has a significantly larger value (left and right shoes),
- or when the opposite takes place (tickets for a movie: no use of two tickets if you are going alone).

$\Rightarrow$ Preferable to selling each item separately when there are dependencies between the different items.

Combinatorial auctions

Auction of a single item:

- a single indivisible resource, and two (or more) players desire using it: who should get it?

Combinatorial auction:

- multiple resources are involved: how do I allocate a collection of interrelated resources?
- interrelations of the different resources may be combinatorially complex, and handling them requires effective handling of this complexity.

Problem statement

Valuations

Formally:

- There is a set of m indivisible items that are concurrently auctioned among n bidders.
- Bidders have preferences regarding subsets (bundles) of items: every bidder i has a valuation function v_i describing her preferences in monetary terms:

Definition

A valuation v is a real-valued function such that for each subset S of items, $v(S)$ is the value that bidder i obtains if he receives this bundle of items.

A valuation must be

- monotone: for $S \subseteq T$ we have that $v(S) \leq v(T)$, and
- normalized: $v(\emptyset) = 0$.

Problem statement

Valuations

The point of defining valuation functions is that the value of a bundle of items need not be equal to the sum of the values of the items in it.

For sets S and T, $S \cap T = \emptyset$, we say that

- S and T are complements to each other if

$$v(S \cup T) > v(S) + v(T) \ ,$$

- S and T are substitutes if

$$v(S \cup T) < v(S) + v(T) \ .$$

Problem statement

Valuations

The point of defining valuation functions is that the value of a bundle of items need not be equal to the sum of the values of the items in it.

For sets S and T, $S \cap T = \emptyset$, we say that

- S and T are complements to each other if

$$v(S \cup T) > v(S) + v(T) \ ,$$

- S and T are substitutes if

$$v(S \cup T) < v(S) + v(T) \ .$$

Two assumptions about bidder preferences:

1. they are quasi-linear in the money: if bidder i wins bundle S and pays a price of p for it then his utility is $v_i(S) - p$.
2. there are no externalities: a bidder only cares about the item that he receives and not about how the other items are allocated among the other bidders.

Problem statement

Allocations

Definition

- An allocation of the items among the bidders is $S_1, \ldots, S_n$ where $S_i \cap S_j = \emptyset$ for every $i \neq j$.
- The social welfare obtained by an allocation is $\sum_i v_i(S_i)$.
- A socially efficient allocation is an allocation with maximum social welfare among all allocations.

Goals:

- To design a mechanism that will find the socially efficient allocation, given that valuation function v_i of bidder i is private information (unknown to the auctioneer or to the other bidders).
- To design a mechanism where this is found in equilibrium.
- To design combinatorial auctions that maximize the auctioneer's revenue.

Problem statement

Difficulties

There are multiple difficulties that we need to address:

Computational complexity: The allocation problem is computationally hard (NP-complete) even for simple special cases.

Representation and communication: Valuation functions are exponential size objects since they specify a value for each bundle.

- How can we even represent them?
- How do we transfer enough information to the auctioneer so that a reasonable allocation can be found?

Strategies: How can we analyze the strategic behavior of the bidders?

An example

Consider a communication network:

- Supply of multiple "connection requests", each requesting a path between two specified nodes in the network.
- A price is offered for such a path.
- Assume each network edge must be fully allocated to one of the requests (edge-disjoint paths allocated to the requests).
- Which requests should be fulfilled, and which paths be allocated?

We can view this as a combinatorial auction:

- The items sold are the edges of the network.
- The players are the different requests.
- The valuation of a request gives the offered price for any bundle of edges that contains a path between the required nodes, and 0 for all other bundles.

Single-minded bidders

We focus on players with very simple valuation functions which we call single-minded bidders:

- interested only in a single specified bundle of items;
- get a specified scalar value if they get this whole bundle (or any superset) and get zero value for any other bundle.

Definition

A valuation v is called single minded if there exists a bundle of items S^* and a value $v^* \in \mathbb{R}_+$ such that $v(S) = v^*$ for all $S \supseteq S^*$, and $v(S) = 0$ for all other S. A single-minded bid is the pair (S^*, v^*).

Note:

- Single-minded valuations are very simply represented.
- We assume as common knowledge that all bidders are single minded.

Single-minded bidders

The allocation problem

- Recall: an allocation gives disjoint sets of items S_i to each bidder i, and aims to maximize the social welfare $\sum_i v_i(S_i)$.
- In the case of single-minded bidders whose bids are given by (S_i^*, v_i^*), an optimal allocation can allocate to every bidder either exactly the bundle she desires ($S_i = S_i^*$) or nothing at all ($S_i = \emptyset$).

Definition (The allocation problem among single-minded bidders)

INPUT: (S_i^*, v_i^*) for each bidder $i = 1, \ldots, n$.

OUTPUT: A subset of winning bids $W \subseteq \{1, \ldots, n\}$ such that for every $i \neq j \in W$, $S_i^* \cap S_j^* = \emptyset$ (i.e., the winners are compatible with each other) with maximum social welfare $\sum_{i \in W} v_i^*$.

Single-minded bidders

Complexity of an allocation

The allocation problem among single-minded bidders is NP-complete.

- We will show this by reduction from the INDEPENDENT-SET problem.
- An independent set of a graph is a subset of the vertices that have no edge between any two of them.

Definition (INDEPENDENT SET problem)

Given an undirected graph $G = (V, E)$ and a number k, does G have an independent set of size k?

Single-minded bidders

Complexity of an allocation

Proposition

The allocation problem among single-minded bidders is NP-hard. More precisely, the decision problem of whether the optimal allocation has social welfare of at least k (where k is an additional part of the input) is NP-complete.

Proof. Given an INDEPENDENT-SET instance, we will build an allocation problem from it as follows:

- The set of items will be E, the set of edges in the graph.
- We have a player for each vertex in the graph.
- For vertex $i \in V$ we have the desired bundle of i be the set of adjacent edges $S_i^* = \{e \in E \mid i \in e\}$, and the value be $v_i^* = 1$.

Single-minded bidders

Complexity of an allocation

Proof (continued).

- A set W of winners satisfies $S_i^* \cap S_j^* = \emptyset$ for every $i \neq j \in W$ if and only if the set of vertices corresponding to W is an independent set in the original graph G.
- The social welfare obtained by W is exactly the size of this independent set.
- It follows that an independent set of size at least k exists if and only if the social welfare of the optimal allocation is at least k, and this concludes the NP-hardness proof.
- The fact that the problem of whether the optimal allocation has social welfare at least k is in NP is trivial as the optimal allocation can be guessed and then the social welfare can be calculated routinely. $\square$

Single-minded bidders

Dealing with NP-hardness

As usual, when a computational problem is shown to be NP-complete, there are three approaches for the next step.

1. Approximation: find an allocation that is approximately optimal.
2. Special cases: identify special cases that can be solved efficiently.
3. Heuristics: find algorithms that run reasonably fast and produce optimal (or near-optimal) results on most natural input instances.

We will discuss each in turn.

Single-minded bidders

Approximation

Definition

An allocation $S_1, \ldots, S_n$ is a c-approximation of the socially optimal allocation $T_1, \ldots, T_n$ iff $\frac{\sum_i v_i(T_i)}{\sum_i v_i(S_i)} \leq c$.

- Approximating the maximum independent set to within a factor of $n^{1-\epsilon}$ (for any fixed $\epsilon > 0$) is NP-complete.
- In our reduction the social welfare was exactly equal to the independent set size, so we get the same hardness here.
- In terms of the number of items m (m is the number of edges, n is the number of vertices, $m \leq n^2$), we get:

Proposition

Approximating the optimal allocation among single-minded bidders to within a factor better than $m^{1/2-\epsilon}$ is NP-hard.

Single-minded bidders

Special cases

Special cases solved efficiently:

- **Case 1:** Each bidder desires a bundle of at most two items ($|S_i^*| \leq 2$). This case is seen to be an instance of the weighted matching problem (in general nonbipartite graphs) which is known to be efficiently solvable.
- **Case 2:** The items are arranged in a linear order and each desired bundle is for a continuous segment of items. This case can be solved efficiently using dynamic programming.

Single-minded bidders

Heuristics

Heuristics:

The allocation problem can be stated as an integer programming problem.

- Known heuristics for solving integer programs can be applied.
- Many of these heuristics rely on the linear programming relaxation.
- Most allocation problems with up to hundreds of items can be practically solved optimally, and even problems with thousands or tens of thousands of items can be practically approximately solved quite well.

Incentive-compatible approximation mechanisms

Goals:

- We would like to optimize the social welfare as much as possible.
- The approach we take is the standard one of mechanism design: incentive compatibility.
- I.e., An allocation algorithm and payment functions such that each player always prefers reporting her private information (the true bids) truthfully to the auctioneer rather than any potential lie.
- This would ensure that the allocation algorithm at least works with the true information.
- We also wish everything to be efficiently computable.

Incentive-compatible approximation mechanism

Definition

- Let V_{sm} denote the set of all single-minded bids on m items.
- Let A be the set of all allocations of the m items between n players.

Definition

A mechanism for single-minded bidders is composed of an allocation mechanism $f : (V_{sm})^n \to A$ and payment functions $p_i : (V_{sm})^n \to \mathbb{R}$.

- The mechanism is computationally efficient if f and all p_i can be computed in polynomial time.
- The mechanism is incentive compatible (in dominant strategies) if for every i, and every $v_1, \ldots, v_n$, $v_i' \in V_{sm}$, we have that

$$v_i(a) - p_i(v_i, v_{-i}) \geq v_i(a') - p_i(v_i', v_{-i}) \ ,$$

where $a = f(v_i, v_{-i})$, $a' = f(v_i', v_{-i})$, and $v_i(a) = v_i$ if i wins in a and zero otherwise.

Incentive-compatible approximation mechanism

Approaches

Main difficulty: the clash between the requirements of incentive compatibility and that of computational efficiency.

- Solution if we leave aside the requirement of computational efficiency: take the socially efficient allocation and let the payments be the VCG payments.
- These payments charge each bidder her externality, i.e., the amount by which her allocated bundle reduced the total reported value of the bundles allocated to others.
- This would be incentive compatible, and would give the exactly optimal allocation.
- However, exact optimization of the social welfare is computationally intractable.

Incentive-compatible approximation mechanism

Approaches

Thus, when we return to the requirement of computational efficiency, exact optimization is impossible.

- Use of "VCG-like" mechanisms: take the best approximation algorithm you can find for the problem (no better than $O(\sqrt{m})$-approximation) and attempt using the same idea of charging each bidder her externality according to the allocation algorithm used.
- This would not be incentive compatible!
- VCG-like payments lead to incentive compatibility if but only if the social welfare is exactly optimized by the allocation rule.

The greedy mechanism for single-minded bidders

Thus we need to find another type of non-VCG mechanisms.

- In general settings, almost no incentive compatible mechanisms are known beyond VCG.
- The single-minded setting is "almost single-dimensional": private values are composed of a single scalar and the desired bundle.
- We will describe a computationally efficient, incentive-compatible mechanism, which provides a $\sqrt{m}$ approximation guarantee (as good as theoretically possible in polynomial time).

The greedy mechanism for single-minded bidders

The greedy mechanism for single-minded bidders

Initialization: Re-order the bids so that $v_1^*/\sqrt{|S_1^*|} \geq \cdots \geq v_n^*/\sqrt{|S_n^*|}$.
$W \leftarrow \emptyset$.

For $i = 1$ to n: if $S_i^* \cap \left(\cup_{j \in W} S_j^*\right) = \emptyset$ then $W \leftarrow W \cup \{i\}$.

Output: **Allocation:** The set of winners is W.

Payments: For each $i \in W$, $p_i = v_j^*/\sqrt{|S_j^*|/|S_i^*|}$, where j is the smallest index such that $S_i^* \cap S_j^* \neq \emptyset$, and for all $k < j$, $k \neq i$, $S_k^* \cap S_j^* = \emptyset$ (if no such j exists then $p_i = 0$).

The greedy mechanism for single-minded bidders

- This mechanism greedily takes winners in an order determined by the value of the expression $v_1^*/\sqrt{|S_1^*|}$.
- This expression was taken as to optimize the approximation ratio obtained theoretically.
- The intuition behind the choice of j for defining the payments is that this is the bidder who lost exactly because of i: if i had not participated in the auction, j would have won.

The greedy mechanism for single-minded bidders

- This mechanism greedily takes winners in an order determined by the value of the expression $v_1^*/\sqrt{|S_1^*|}$.
- This expression was taken as to optimize the approximation ratio obtained theoretically.
- The intuition behind the choice of j for defining the payments is that this is the bidder who lost exactly because of i: if i had not participated in the auction, j would have won.

Theorem

The greedy mechanism is efficiently computable, incentive compatible, and produces a $\sqrt{m}$-approximation of the optimal social welfare.

Proof. Computational efficiency is obvious; we will next show incentive compatibility and the approximation performance.

The greedy mechanism for single-minded bidders

Proof of incentive-compatibility

Lemma

A mechanism for single-minded bidders in which losers pay 0 is incentive compatible if and only if it satisfies the following two conditions:

1. *Monotonicity: A bidder who wins with bid (S_i^*, v_i^*) keeps winning for any $v_i' > v_i^*$ and for any $S_i' \subset S_i^*$ (for any fixed setting of the others).*
2. *Critical Payment: A bidder who wins pays the minimum value needed for winning: the infimum of all values v_i' such that (S_i^*, v_i') still wins.*

Our mechanism satisfies these two properties.

- Monotonicity: increasing v_i^* or decreasing S_i^* can only move bidder i up in the greedy order, making it easier to win.
- Critical payment: i wins as long as she appears in the greedy order before j. The payment is exactly the value at which the transition between i being before and after j in the greedy order happens.

The greedy mechanism for single-minded bidders

Proof of incentive-compatibility

Under the given conditions, a truthful bidder will never receive negative utility:

- if she looses her utility is zero (losers pay zero),
- if she wins her value must be at least the critical value, which exactly equals his payment.

A bidder can never improve his utility by reporting some bid (S', v') instead of her true values (S, v).

- If (S', v') is a losing bid or if S' does not contain S, then clearly reporting (S, v) can only help.
- Therefore we will assume that (S', v') is a winning bid and that $S' \supseteq S$.

The greedy mechanism for single-minded bidders

Proof of incentive-compatibility

We next show that the bidder will never be worse off by reporting (S, v') rather than (S', v').

- Denote the bidder's payment for the bid (S', v') by p', and for the bid (S, v') by p.
- For every $x < p$, bidding (S, x) will lose since p is a critical value.
- By monotonicity, (S', x) will also be a losing bid for every $x < p$, and therefore the critical value p' is at least p.
- It follows that by bidding (S, v') instead of (S', v') the bidder still wins and her payment will not increase.

The greedy mechanism for single-minded bidders

Proof of incentive-compatibility

It is left to show that bidding (S, v) is no worse than the winning bid (S, v'):

- Assume first that (S, v) is a winning bid with a payment (critical value) $\tilde{p}$.
  - If $v' \geq \tilde{p}$, the bidder still wins with the same payment, thus misreporting her value would not be beneficial.
  - If $v' < \tilde{p}$ the bidder will lose, gaining zero utility, and she will not be better off.
- If (S, v) is a losing bid, v must be smaller than the corresponding critical value, so the payment for any winning bid (S, v') will be greater than v, making this deviation non-profitable.

$\square$

The greedy mechanism for single-minded bidders

Proof of approximation performance

The approximation guarantee is ensured by the following lemma.

Lemma

Let OPT be an allocation (i.e., set of winners) with maximum value of $\sum_{i \in OPT} v_i^*$, and let W be the output of the algorithm. Then

$$\sum_{i \in OPT} v_i^* \leq \sqrt{m} \sum_{i \in W} v_i^* \ .$$

Proof.

- For each $i \in W$ let $OPT_i = \{j \in OPT, j \geq i \mid S_i^* \cap S_j^* \neq \emptyset\}$ be the set of elements in OPT that did not enter W because of i.
- Clearly $OPT \subseteq \cup_{i \in W} OPT_i$ and the lemma will follow if we prove that for every $i \in W$, $\sum_{j \in OPT_i} v_j^* \leq \sqrt{m} v_i^*$.

The greedy mechanism for single-minded bidders

Proof of approximation performance

Proof (continued).

- Every $j \in OPT_i$ appeared after i in the greedy order and thus $v_j^* \leq \frac{v_i^* \sqrt{|S_j^*|}}{\sqrt{S_i^*}}$.
- Summing over all $j \in OPT_i$,

$$\sum_{j \in OPT_i} v_j^* \leq \frac{v_i^*}{\sqrt{|S_i^*|}} \sum_{j \in OPT_i} \sqrt{S_j^*} \ .$$

- Using the Cauchy-Schwarz inequality, we can bound

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|OPT_i|} \sqrt{\sum_{j \in OPT_i} |S_j|} \ .$$

The greedy mechanism for single-minded bidders

Proof of approximation performance

Proof (continued).

- Every S_j^* for $j \in OPT_i$ intersects S_i^*.
- Since OPT is an allocation, these intersections must all be disjoint, and thus $|OPT_i| \leq |S_i^*|$.
- Since OPT is an allocation, $\sum_{j \in OPT_i} |S_j| \leq m$.
- We thus get

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} \leq \sqrt{|S_i^*|}\sqrt{m} \ .$$

- Using the inequality in the previous slide we get

$$\sum_{j \in OPT_i} v_j^* \leq \sqrt{m} v_i^* \ ,$$

as needed. $\square$.

Combinatorial auctions with general valuations

Combinatorial auctions with general valuations:

- We will study a linear-programming relaxation of the winner-determination problem.
- We will define the economic notion of a competitive equilibrium with item prices (or Walrasian equilibrium).
- We will describe a strong connection between them.

The integer program

The winner determination problem in combinatorial auctions can be formulated by an integer program (IP):

$$\begin{array}{rll} \text{maximize} & \sum_{i \in N, S \subseteq M} x_{i,S} \cdot v_i(S) & \\ \text{s. t.} & \sum_{i \in N, S: j \in S} x_{i,S} \leq 1 & \forall j \in M \\ & \sum_{S \subseteq M} x_{i,S} \leq 1 & \forall i \in N \\ & x_{i,S} \in \{0,1\} & \forall i \in N, S \subseteq M \ . \end{array}$$

- Each variable $x_{i,S}$ equals 1 if bidder i receives the bundle S, and zero otherwise.
- The objective function is therefore maximizing social welfare.
- Conditions ensure that
  1. each item is allocated to at most one bidder, and
  2. each player is allocated at most one bundle.

The LP relaxation

Linear program (LP) relaxation:

$$\begin{array}{rll} \text{maximize} & \sum_{i \in N, S \subseteq M} x_{i,S} \cdot v_i(S) & \\ \text{s. t.} & \sum_{i \in N, S: j \in S} x_{i,S} \leq 1 & \forall j \in M \\ & \sum_{S \subseteq M} x_{i,S} \leq 1 & \forall i \in N \\ & x_{i,S} \geq 0 & \forall i \in N, S \subseteq M \ . \end{array}$$

- Solutions to the LP can be intuitively viewed as fractional allocations that would be allowed if items were divisible.
- The LP has exponentially (in m) many variables but, assuming reasonable access to the valuations, the it can be solved efficiently.

The dual LP

The dual LP:

$$\begin{array}{rll} \text{minimize} & \sum_{i \in N} u_i + \sum_{j \in M} p_j & \\ \text{s. t.} & u_i + \sum_{j \in S} p_j \geq v_i(S) & \forall i \in N, S \subseteq M \\ & u_i \geq 0, p_j \geq 0 & \forall i \in N, j \in M \ . \end{array}$$

The usage of the notations p_j and u_i is intentional: at the optimal solution, these dual variables can be interpreted as the prices of the items and the utilities of the bidders.

Walrasian equilibrium

Demands

A fundamental notion in economic theory is the notion of a competitive equilibrium: a set of prices where the market clears, i.e., the demand equals the supply.

- Given a set of prices, the demand of each bidder is the bundle that maximizes her utility.
- Linear pricing rule: a price per each item is available, and the price of each bundle is the sum of the prices of the items in this bundle.

Definition

For a given bidder valuation v_i and given item prices $p_1, \ldots, p_m$, a bundle T is called a demand of bidder i if for every other bundle $S \subseteq M$ we have that

$$v_i(S) - \sum_{j \in S} p_j \leq v_i(T) - \sum_{j \in T} p_j \ .$$

Walrasian equilibrium

A Walrasian equilibrium is a set of "market-clearing" prices where

- every bidder receives a bundle in his demand set, and
- unallocated items have zero prices.

Definition

A set of nonnegative prices $p_1^*, \ldots, p_m^*$ and an allocation $S_1^*, \ldots, S_m^*$ of the items is a Walrasian equilibrium if for every player i, S_i^* is a demand of bidder i at prices $p_1^*, \ldots, p_m^*$ and for any item j that is not allocated (i.e., $j \notin \cup_{i=1}^n = S_i^*$) we have $p_j^* = 0$.

The First Welfare Theorem

- Walrasian equilibria, if they exist, are economically efficient: they necessarily obtain the optimal welfare.
- If a Walrasian equilibrium exists, then the optimal solution to the LP relaxation will be integral.

Theorem (The First Welfare Theorem)

Let $p_1^*, \ldots, p_m^*$ and $S_1^*, \ldots, S_n^*$ be a Walrasian equilibrium. Then the allocation $S_1^*, \ldots, S_n^*$ maximizes social welfare. Moreover, it even maximizes social welfare over all fractional allocations. I.e., let $\{x_{i,S}^*\}_{i,S}$ be a feasible solution to the LP relaxation. Then,

$$\sum_{i=1}^{n} v_i(S_i^*) \geq \sum_{i \in N, S \subseteq M} x_{i,S}^* v_i(S) \ .$$

The First Welfare Theorem

Proof. In a Walrasian equilibrium, each bidder receives his demand: for every bidder i and every bundle S,

$$v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq v_i(S) - \sum_{j \in S} p_j^* \quad .$$

The fractional solution is feasible to the LP relaxation: for every bidder i, $\sum_S x_{i,S}^* \leq 1$, and therefore

$$v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq \sum_{S \subseteq M} x_{i,S}^* \left(v_i(S) - \sum_{j \in S} p_j^* \right) \quad .$$

The theorem will follow from summing over all bidders.

The First Welfare Theorem

Proof (continued). We can show that

$$\sum_{i \in N} \sum_{j \in S_i^*} p_j^* \geq \sum_{i \in N, S \subseteq M} x_{i,S}^* \sum_{j \in S} p_j^* \ .$$

- The left-hand side equals $\sum_{j=1}^{m} p_j^*$ since $S_1^*, \ldots, S_n^*$ is an allocation and the prices of unallocated items in a Walrasian equilibrium are zero.
- The right-hand side is at most $\sum_{j=1}^{m} p_j^*$ since the coefficient of every price p_j^* is at most 1 (by the constraint of the LP).

□

Inexistence of Walrasian equilibrium

A simple class of valuations for which no Walrasian equilibrium exists:

- There are two players, Alice and Bob, and two items $\{a, b\}$.
- Alice has a value of 2 for every nonempty set of items, and Bob has a value of 3 for the whole bundle $\{a, b\}$, and 0 for any of the singletons.
- The optimal allocation will clearly allocate both items to Bob.
- Therefore, Alice must demand the empty set in any Walrasian equilibrium.
- Both prices will be at least 2; otherwise, Alice will demand a singleton.
- Hence, the price of the whole bundle will be at least 4, Bob will not demand this bundle, and consequently, no Walrasian equilibrium exists for these players.

The Second Welfare Theorem

The existence of an integral optimum to the linear programming relaxation is also a sufficient condition for the existence of a Walrasian equilibrium:

Theorem (The Second Welfare Theorem)

If an integral optimal solution exists for the LP relaxation, then a Walrasian equilibrium whose allocation is the given solution also exists.

Proof.

- An optimal integral solution for the LP relaxation defines a feasible efficient allocation $S_1^*, \ldots, S_n^*$.
- Consider an optimal solution $p_1^*, \ldots, p_n^*, u_1^*, \ldots, u_n^*$ to the dual.
- We will show that $S_1^*, \ldots, S_n^*, p_1^*, \ldots, p_n^*$ is a Walrasian equilibrium.

The Second Welfare Theorem

Proof (continued).

- Complementary slackness conditions are necessary and sufficient conditions for optimality of solutions to the primal and dual LPs.
- For every player i for which $x_{i,S_i^*} > 0$ (i.e., $x_{i,S_i^*} = 1$), we have that

$$u_i^* = v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \ .$$

- For any other bundle S we get

$$v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq v_i(S) - \sum_{j \in S} p_j^* \ .$$

- The complementary slackness conditions also imply that for every item j for which $\sum_{i \in N, S: j \in S} x_{i,S} < 1$, which for integral solutions means that item j is unallocated, then necessarily $p_j^* = 0$. □

Existence of Walrasian equilibrium

The two welfare theorems:

Theorem (The First Welfare Theorem)

If a Walrasian equilibrium exists, then the optimal solution to the LP relaxation will be integral.

Theorem (The Second Welfare Theorem)

If an integral optimal solution exists for the LP relaxation, then a Walrasian equilibrium whose allocation is the given solution also exists.

show that the existence of a Walrasian equilibrium is equivalent to having a zero integrality gap:

Corollary

A Walrasian equilibrium exists in a combinatorial auction environment iff the corresponding LP relaxation admits an integral optimal solution.

Bidding languages

Representation of bids in combinatorial auctions:

- We are looking for representations of valuations: bidders encode their valuation and send it to the auctioneer.
- The auctioneer takes all valuations (bids) received determine the allocation.
- We will consider indirect, iterative ways of transferring information to the auctioneer.
- Specifying a valuation in a combinatorial auction of m items requires providing a value for each of the possible $2^m - 1$ nonempty subsets.
- A naive representation would thus require $2^m - 1$ real numbers to represent each possible valuation.
  $\Rightarrow$ completely impractical even for about two or three dozen items.

Bidding languages

- The representation problem seems to be the bottleneck in practice.
- We will thus be looking for languages that allow succinct representations of valuations.
- We will call these **bidding languages** reflecting their intended usage.
- Due to information-theoretic reasons it will never be possible to encode all possible valuations succinctly.
- Our interest will be in succinctly representing interesting or important ones.
- Trade-off: expressiveness vs. simplicity.
  - A language to express succinctly as many "naturally occurring" valuations as possible.
  - A language to be as simple as possible, both for humans to express and for programs to work with.

Elements of representation: atoms, OR, and XOR

The common bidding languages construct their bids from combinations of simple atomic bids.

- The usual atoms in such schemes are the single-minded bids (S, p) meaning an offer of p monetary units for the bundle S of items.
- Formally, the valuation represented by (S, p) is one where $v(T) = p$ for every $T \supseteq S$, and $v(T) = 0$ for all other T.

Intuitively, bids can be combined by simply offering them together. There are two possible semantics for an offer of several bids:

- The bids are totally independent, allowing any subset of them to be fulfilled (OR bids).
- The bids are mutually exclusive and allows only one of them to be fulfilled (XOR bids).

Elements of representation: atoms, OR, and XOR

Examples:

- Consider the valuations represented by

$$(\{a,b\},3) \text{ XOR } (\{c,d\},5) \quad \text{and} \quad (\{a,b\},3) \text{ OR } (\{c,d\},5) \ .$$

- Each of them values the bundle $\{a,c\}$ at 0 (since no atomic bid is satisfied) and values the bundle $\{a,b\}$ at 3.
- The difference is in the bundle $\{a,b,c,d\}$, which is valued at 5 by the XOR bid (according to the best atomic bid satisfied), but is valued at 8 by the OR bid.
- For another example, consider the bid

$$(\{a,b\},3) \text{ OR } (\{a,c\},5) \ .$$

Here, the bundle $\{a,b,c\}$ is valued at 5 since both atomic bids cannot be satisfied together.

OR and XOR bids

Formally:

- Both OR and XOR bids are composed of a collection of pairs (S_i, p_i), where
  - each S_i is a subset of the items;
  - p_i is the maximum price that he is willing to pay for that subset.
- For the valuation $v = (S_1, p_1)$ XOR $\cdots$ XOR (S_k, p_k), the value of $v(S)$ is defined to be $\max_{i:S_i \subseteq S} p_i$.
- For the valuation $v = (S_1, p_1)$ OR $\cdots$ OR (S_k, p_k), the value of $v(S)$ is defined to be the maximum over all possible "valid collections" W, of the value of $\sum_{i \in W} p_i$, where W is a valid collection of pairs if for all $i \neq j \in W, S_i \cap S_j = \emptyset$.

OR and XOR bids

- XOR bids can represent every valuation v: just XOR the atomic bids $(S, v(S))$ for all bundles S.
- OR bids can represent only superadditive bids (i.e., bids such that, for any two disjoint sets S, T, $v(S \cup T) \geq v(S) + v(T)$):
  - The atoms giving the value $v(S)$ are disjoint from those giving the value $v(T)$, and they will be added together for $v(S \cup T)$.
  - All superadditive valuations can indeed be represented by OR bids by ORing the atomic bids $(S, v(S))$ for all bundles S.

Additive and unit-demand valuations

We are interested in the size of the representation, defined to be simply the number of atomic bids in it.

Definition

A valuation is called additive if $v(S) = \sum_{j \in S} v(\{j\})$ for all S. A valuation is called unit-demand if $v(S) = \max_{j \in S} v(\{j\})$ for all S.

- An additive valuation is directly represented by an OR bid:

$$(\{1\}, p_1) \text{ OR } (\{2\}, p_2) \text{ OR } \cdots \text{ OR } (\{m\}, p_m) \ .$$

- A unit-demand valuation is directly represented by an XOR bid:

$$(\{1\}, p_1) \text{ XOR } (\{2\}, p_2) \text{ XOR } \cdots \text{ XOR } (\{m\}, p_m) \ .$$

Additive and unit-demand valuations

Definition

A valuation is called additive if $v(S) = \sum_{j \in S} v(\{j\})$ for all S. A valuation is called unit-demand if $v(S) = \max_{j \in S} v(\{j\})$ for all S.

- Additive valuations can be represented by XOR bids, but this may take exponential size:
  - Atomic bids for all $2^m - 1$ possible bundles will be needed whenever $p_j > 0$ for all j.
  - This is because an atomic bid is required for every bundle S with $v(S)$ strictly larger than that of all its strict subsets, which is the case here for all S.
- On the other hand, nontrivial unit-demand valuations are never superadditive and thus cannot be represented at all by OR bids.

Combinations of OR and XOR

- None of the OR and XOR bidding languages is expressive enough to succinctly represent many simple valuations.
- A natural attempt is to combine the power of OR and XOR bids.
- The most general way is to define OR and XOR as operations on valuations.

Definition

Let v and u be valuations, then (v XOR u) and (v OR u) are valuations and are defined as follows:

$$\begin{aligned}(v \text{ XOR } u)(S) &= \max(v(S), u(S)) \\ (v \text{ OR } u)(S) &= max_{R,T \subseteq S, R \cap T = \emptyset}(v(R) + u(T)) \ .\end{aligned}$$

Combinations of OR and XOR

- A general OR/XOR bid will be given by an arbitrary expression involving the OR and XOR operations over atomic bids.
- For instance, the bid

$$((\{a, b\}, 3) \text{ XOR } (\{c\}, 2)) \text{ OR } (\{d\}, 5)$$

values the bundle $\{a, b, c\}$ at 3, but the bundle $\{a, b, d\}$ at 8.

Downward sloping symmetric valuations

Downward sloping symmetric valuations demonstrate the added power we can get from such OR/XOR combinations.

Definition

A valuation is called symmetric if $v(S)$ depends only on $|S|$. A symmetric valuation is called downward sloping if it can be represented as $v(S) = \sum_{j=1}^{|S|} p_j$, with $p_1 \geq p_2 \geq \cdots \geq p_m \geq 0$.

It is easy to verify that every downward sloping valuation with $p_1 > p_2 > \cdots > p_m > 0$

1. requires XOR bids of size $2^m - 1$ and
2. cannot be represented at all by OR bids.

Downward sloping symmetric valuations

Lemma

OR-of-XORs bids can express any downward sloping symmetric valuation on m items in size m^2.

Proof.

- For each $j = 1, \ldots, m$ we will have a clause that offers p_j for any single item.
- Such a clause is a simple XOR-bid, and the m different clauses are all connected by an OR.
- Since the p_j's are decreasing, we are assured that the first allocated item will be taken from the first clause, the second item from the second clause, etc.

□

Dummy items

- General OR/XOR formulae seem very complicated and dealing with them algorithmically would appear to be quite difficult.
- A generalization of the language makes things simple: The main idea is to allow XORs to be represented by ORs.
- This is done by allowing the bidders to introduce dummy items into the bids.
- These items will have no intrinsic value to any of the participants, but they will be indirectly used to express XOR constraints.
- The idea is that an XOR bid (S_1, p_1) XOR (S_2, p_2) can be represented as $(S_1 \cup \{d\}, p_1)$ OR $(S_2 \cup \{d\}, p_2)$, where d is a dummy item.

Dummy items

Formally:

- Each bidder i has its own set of dummy items D_i, which only he can bid on.
- An OR* bid by bidder i is an OR bid on the augmented set of items $M \cup D_i$.
- The value that an OR* bid gives to a bundle $S \subseteq M$ is the value given by the OR bid to $S \cup D_i$.
- Thus, for example, for the set of items $M = \{a, b, c\}$, the OR* bid

$$(\{a, d\}, 1) \text{ OR } (\{b, d\}, 1) \text{ OR } (\{c\}, 1) \ ,$$

where d is a dummy item, is equivalent to

$$((\{a\}, 1) \text{ XOR } (\{b\}, 1)) \text{ OR } (\{c\}, 1) \ .$$

Dummy items

Despite its apparent simplicity, this language can simulate general OR/XOR formulae.

Theorem

Any valuation that can be represented by OR/XOR formula of size s can be represented by OR bids of size s, using at most s^2 dummy items.*

Sketch of proof:

- We can prove by induction on the formula structure that a formula of size s can be represented by an OR* bid with s atomic bids.
- We can then show that each atomic bid in the resulting OR* bid can be modified so as not to include more than s dummy items in it. □

Bottom line: we have identified a simple, powerful language, that is as easily handled by allocation algorithms as are the single-minded bids.

Further reading

- Noam Nisan, Tim Roughgarden, Éva Tardos, and Vizay Vazirani (eds): Algorithmic Game Theory, **Chapters 11 & 28**. Cambridge University Press, 2007.