Preference elicitation/iterative mechanisms

Adapted from notes by Vincent Conitzer

“Much effort has been spent on developing algorithms for the hard problem of winner determination once bids have been received. Yet, preference elicitation has emerged as perhaps the key bottleneck in the real-world deployment of combinatorial auctions. Advanced clearing algorithms are worthless if one cannot simplify the bidding problem facing bidders.” ---David Parkes

Concerns:

– Communication cost of sending bids
– Cost to bidders to determine valuation on different bundles. Give an example where bundle valuation is difficult for the bidder.
– Agents may prefer NOT to reveal their valuation information
• When incremental elicitation is used, motivating bidders to answer queries truthfully is more difficult (than VCG, say), since elicitor queries may leak information to the bidder about the answers that other bidders have given.
  • Example: Suppose I give my valuation for \{a,b\} but I am never asked how I value \{a\}, what do I conclude?
  • Suppose the query is “Do you value \{a,b,c\} more than $100?” what do I conclude?
  • Example: Suppose I am asked about a bundle I hadn’t considered. How does that change my thinking?

**Elicitor mechanisms**

• Pull – querying (value, demand, order, bounding)
  – Difference query: what is the difference between the bundle values – may reveal less private information.
• Push – bidders can provide unsolicited information
• Bidders can refuse to answer (if too hard or other reasons?)
Constraint Network for each agent
- Mechanism for representing *incompletely* specified valuation functions
- One node for each bundle b. Labeled with the tightest range that elicitor can prove: \([LB_i(b), UB_i(b)]\) of bounds on valuation of bundle by i.
- Edge \((a\rightarrow b)\) if bundle a is preferable to bundle b. May be known without queries (due to free disposal).
- We want to infer information because of
  - answers to a query
  - structure of the constraint graph

How does that work?

How can you see free disposal?
Constraint Network for each agent
• An edge allows elicitor to infer $LB_i(a) \geq LB_i(b)$
• Edge allows inference $UB_i(a) \geq UB_i(b)$
• Can add edge $a \rightarrow b$ (a dominates b) if we can prove $v_i(a) \geq v_i(b)$
• Explicit representation is intractable as $2^m$ nodes

Terms:
• Collection: states which bundle $X_i$ bidder i receives.
• Allocation: feasible collection (no overlap)
• Pareto efficient
• Social Welfare maximizing: maximizes sum of valuations of those receiving bundle
• Certificate: a set of query-answer pairs that prove an allocation is optimal. Critical in knowing when to quit.
Terms

• A bidder’s utility function is **compressible** if it can be expressed using less than exponential bundle evaluations. For example, if it is linear.

• The **elicitation ratio** is the number of bundles elicited divided by the total number of bundles.

---

• Iterative CAs can address concerns about **privacy because bidders only need to reveal partial and indirect** information about their valuations.

• **Transparency is another practical concern in CAs:** *We want* bidders to be able to verify and validate the outcome of an auction.

• However, iterative CAs offer new opportunities to bidders for manipulation. **Explain.**

• The benefits of iterative auctions disappear when bidders choose to strategically delay bidding activity until the last rounds of an auction. **Activity rules** can be used to address this stalling and promote meaningful bidding during the early rounds of an auction.
Quasilinear (almost linear)
Used in a lot of settings (nlog n is considered quasilinear)
• We assume quasilinear utility:
  • \( u_i(S, p) = v_i(S) - p \)
  • (all supersets of S have value at least \( v_i(S) \))
• Linear prices: bundle is sum of individual prices
• Non-linear prices: not the sum of the individual prices
• non-anonymous prices: different bidders charged different amounts for same bundle

• Competitive equilibrium: maximizes the utility of every bidder and the seller
• The goods are substitutes condition requires that a bidder will continue to demand items that do not change in price as the price on other items increases.
Preference elicitation (elections)

“yes”

“yes”

“no”

“most preferred?”

“most preferred?”

Who wins?

Preference elicitation (elections)

“yes”

“yes”

“no”

“most preferred?”

wins
Preference elicitation (auction)

“v({A})?”

“What would you buy if the price for A is 30, the price for B is 20, the price for C is 20?”

“nothing”

“v({A,B,C}) < 70?”

“yes” “40”

“v({B, C})?”

“What would you buy if the price for A is 30, the price for B is 20, the price for C is 20?”

“nothing”

gets {A}, pays 30
gets {B,C}, pays 40
Unnecessary communication

• Combinatorial auctions, should in principle communicate a value for every single bundle!

• Much of this information will be irrelevant, e.g.:
  – Suppose each item has already received a bid >$1
  – Bidder 1 values the grand bundle, \( G \), of all items at
    \( v_1(G) = $1 \)
  – To find the optimal allocation, we need not know anything more about 1’s valuation function (assuming free disposal)
  – We may still need more detail on 1’s valuation function to compute Clarke payments…
  – … but not if each item has received two bids >$1

• Can we spare bidder 1 the burden of communicating (and figuring out) her whole valuation function?

Single-stage mechanisms

• If all agents must report their valuations (types) at the same time (e.g., sealed-bid), then almost no communication can be saved
  – if we do not know that other bidders have already placed high bids on items, we may need to know more about bidder 1’s valuation function
  – In this case, Can only save communication of information that is irrelevant regardless of what other agents report
    • E.g. if a bidder’s valuation is below the reserve price, it does not matter exactly where below the reserve price it is
    • E.g. a voter’s second-highest candidate under plurality rule

• Could still try to design the mechanism so that most information is (unconditionally) irrelevant
  – E.g. [Hyafil & Boutilier IJCAI 07]
Multistage mechanisms

- In a multistage (or iterative) mechanism,
  - bidders communicate something,
  - then find out something about what others communicated,
  - then communicate again, etc.
- After enough information has been communicated, the mechanism declares an outcome
- What multistage mechanisms have we seen already?

A (strange) example multistage auction

- Center can choose to hide information from agents, but only insofar as it is not implied by queries we ask of them
Elicitation algorithms

• Suppose agents always answer truthfully
• Design elicitation algorithm to minimize queries for given rule
• What is a good elicitation algorithm for Single Transferable Voting?
• What about Bucklin? Bucklin: First choice votes are first counted. If one candidate has a majority, that candidate wins. Otherwise the second choices are added to the first choices. Again, if a candidate with a majority vote is found, the winner is the candidate with the most votes accumulated. Lower rankings are added as needed.

An elicitation algorithm for the Bucklin voting rule based on binary search

[Conitzer & Sandholm 05]

• Alternatives: A B C D E F G H

• Top 4?  {A B C D}  {A B F G}  {A C E H}

• Top 2?  {A D}  {B F}  {C H}

• Top 3?  {A C D}  {B F G}  {C E H}

Agent only gives us the additional information we don’t know
How do we encourage truth telling?

• We know VCG encourages truth telling, but they required complete “up front” revelation of each bidder’s valuation.

• A bidder may condition her response on the precise sequence of queries asked by inferring the query is necessary due to answers given by others.

• Ascending auctions have some desirable attributes as far as truth telling.

• Proxy bidding also encourages truth telling.

Funky strategic phenomena in multistage mechanisms

• Suppose we sell two items A and B in parallel English auctions to bidders 1 and 2
  – Minimum bid increment of 1

• No complementarity/substitutability

• $v_1(A) = 30, v_1(B) = 20, v_2(A) = 20, v_2(B) = 30$, all of this is common knowledge

• 1’s strategy: “I will bid 1 on B and 0 on A, unless 2 starts bidding on B, in which case I will bid up to my true valuations for both.”

• 2’s strategy: “I will bid 1 on A and 0 on B, unless 1 starts bidding on A, in which case I will bid up to my true valuations for both.”

• What will happen?
Funky strategic phenomena in multistage mechanisms

- Suppose we sell two items A and B in parallel English auctions to bidders 1 and 2
  - Minimum bid increment of 1
- No complementarity/substitutability
- $v_1(A) = 30$, $v_1(B) = 20$, $v_2(A) = 20$, $v_2(B) = 30$, all of this is common knowledge
- 1's strategy: "I will bid 1 on B and 0 on A, unless 2 starts bidding on B, in which case I will bid up to my true valuations for both."
- 2's strategy: "I will bid 1 on A and 0 on B, unless 1 starts bidding on A, in which case I will bid up to my true valuations for both."
- This is an equilibrium!
  - Inefficient allocation (assignment is not to those who value it the most)
  - Self-enforcing collusion
  - Bidding truthfully (up to true valuation) is not a dominant strategy

Lower bounds on communication interesting recent research

- Communication complexity theory can be used to show lower bounds
  - "Any elicitation algorithm for rule r requires communication of at least N bits (in the worst case)"
- Voting [Conitzer & Sandholm 05]
  - Bucklin requires at least on the order of nm bits
  - STV requires at least on the order of n log m bits
    - Natural algorithm uses on the order of $n(\log m)^2$ bits
- Combinatorial auction winner determination requires exponentially many bits [Nisan & Segal 06]
  - … unless only a limited set of valuation functions is allowed
How is information about bidder valuations represented?

• Representation is tied to permitted query types. If only bounds queries are used, then upper and lower bounds on valuations must be maintained.
• One might use probabilistic representations to decide which queries are most likely to be useful.

Analogy

• What is the purpose of homework?
• Do I need weekly homework in order to assign a grade? Could I give one test, twohomeworks, and the final paper and have enough information to grade you?
• One professor used to make finals optional. Why? Is the information gained from scoring a final redundant?
• Same professor decided to make homework worth almost nothing as he figured people worked together so he couldn’t tell what they had done.
Fooling Set

• A set $H$ of input vectors is a **fooling set** for $f$ iff:
  1. $\forall (z_1, \ldots, z_n) \in H$, $f(z_1, \ldots, z_n) = f_0$.
  2. For every two distinct vectors $z, z'$ of $H \exists$ mix of coordinates s.t. image is $1-f_0$; e.g. $f(z_1,z_2',z_3',\ldots)=1-f_0$.

• **Lemma:** fooling set of size $m \Rightarrow$ lower bound of $\log(m)$ on communication complexity.

How do we know that we have found the **best** elicitation protocol for a mechanism?

• **Communication complexity theory:**
  - Agent $i$ holds input $x_i$, agents must communicate enough information to compute some $f(x_1, x_2, \ldots, x_n)$
  - Consider the tree of all possible communications:
  - Every input vector goes to some leaf
  - If $x_1, \ldots, x_n$ goes to same leaf as $x_1', \ldots, x_n'$ then so must any mix of them (e.g., $x_1, x_2', x_3, \ldots, x_n'$)
  - Only possible if $f$ is same in all $2^n$ cases
  - Suppose we have a fooling set of $t$ input vectors that all give the same function value $f_0$, but for any two of them, there is a mix that gives a different value
  - Then all vectors must go to different leaves $\Rightarrow$ tree depth must be $\geq \log(t)$
  - Also lower bound on **nondeterministic** communication complexity
    - With false positives or negatives allowed, depending on $f_0$
Communication Complexity: Example

• 2 players, each player holds 2 bits. Wish to determine whether all bits are 1.

• A “short circuit” evaluation.

iBundle: an ascending CA [Parkes & Ungar 00]

• Each round, each bidder i faces separate price \( p_i(S) \) for each bundle \( S \)
  – Note: different bidders may face different prices for the same bundle
  – Prices start at 0

• A bidder (is assumed to) bid \( p_i(S) \) on the bundle(s) \( S \) that maximize(s) her utility given the current prices, i.e., that maximize(s) \( v_i(S) - p_i(S) \) (straightforward bidding)
  – Bidder drops out if all bundles would give negative utility

• Winner determination problem is solved with these bids

• If some (active) bidder i did not win anything, that bidder’s prices are increased by \( \varepsilon \) on each of the bundles that she bid on (and supersets thereof), and we go to the next round

• Otherwise, we terminate with this allocation & these prices
Restricted valuations

• For (e.g.) combinatorial auctions, if we know that agents’ valuation functions lie in a restricted class of functions, then they may be easy to elicit.

• E.g. if we know that an agent’s valuation function is an OR of bundles of size at most 2, then all we need to ask a bidder for is his value of each bundle of size at most 2, to know the entire function.
  – $O(m^2)$ queries
  – So-called value queries

• Which classes of valuations can we elicit using only polynomially many queries?
  – … and what types of queries do we need?

• Closely related to query learning in machine learning.

Restricted valuations…

• Various restricted classes can be elicited using polynomially many value queries.
  – Read-once & toolbox valuations [Zinkevich, Blum, Sandholm EC 03]
  – Valuations with limited item interdependency [Conitzer, Sandholm, Santi AAAI 05]

• Other classes inherently require other types of query.
• E.g., demand query: “Which bundle would you buy given prices $p(S)$ on bundles?”
  – Could also just have prices on items
  – Compare iBundle ascending CA

• A value query can be simulated using polynomially many demand queries (even just with item prices), but not vice versa [Blumrosen & Nisan EC 05]

• Using (bundle-price) demand queries, XOR valuations can be elicited using $O(m^2 \# \text{terms})$ queries [Lahaie & Parkes EC 04]
• … but if only item-price demand queries (and value queries) are allowed, exponentially many queries are required [Blum et al. JMLR 04]
Hudson and Sandholm

- Allocating \( k \) items to \( n \) bidders
- Concern: how to even store the results they get
- Instead of storing the set of allocations, they repeatedly solve an integer program to compute the value of the highest allocation
- Store upper bounds and lower bounds for some bundles.