

Lecture 15

Lecturer: Aaron Roth

Scribe: Aaron Roth

Auction Design in Single Parameter Domains

Last lecture, we saw the VCG mechanism, which has a tremendous number of nice features, and achieves them all in a very general setting. However, the VCG mechanism was particular to maximizing *social welfare*: $\sum_i v_i(a)$. What if we want to design an auction to maximize some other objective? What can we do?

One thing we can do is (slightly) generalize VCG to maximize any *affine* objective function:

$$\sum_{i=1}^n \alpha_i v_i(a) + \beta(a).$$

You will prove this generalization on the homework.

What else can we do? In simple settings, as we will see, we can completely characterize the set of objective functions we can optimize truthfully.

Definition 1 (Single Parameter Domain) A single parameter domain *with a set of alternatives* A is defined by a public value summarization function:

$$w_i : A \rightarrow \mathbb{R}$$

such that agent i 's valuation function is parameterized by a real number $v_i \in \mathbb{R}$, and values outcome a at $v_i \cdot w_i(a)$

i.e. single parameter domains are simple settings in which an agent's valuation can be described by a single real number. Or in other words, it is already known what an agent's *relative preferences* are over outcome – the only thing unspecified is the strength of their preferences.

Many things are single parameter domains. For example:

1. Single item auctions. We already know that agent i prefers to win the item than to lose it – all that needs to be specified is how much agent i values the item. Here:

$$w_i(a) = \begin{cases} 1, & a = i; \\ 0, & \text{otherwise.} \end{cases}$$

2. Buying a path in a network: In this problem, agents correspond to edges in a network, and will experience some cost if they are used. The mechanism would like to buy service from a set of agents that form a path in the network, to optimize some objective (minimize social cost, maximize throughput, etc.) Here an alternative a is a set of edges and:

$$w_e(a) = \begin{cases} 1, & e \in A; \\ 0, & \text{otherwise.} \end{cases}$$

3. Job Scheduling: In this problem, the agents correspond to machines i , each of whom has a different cost c_i for running one unit of computation. Jobs j have different sizes ℓ_j (i.e. a job that would cost machine i $\ell_j \cdot c_i$ to run), and the task is to allocate jobs to machines to optimize some objective. We write $a_{ij} = 1$ if job j is allocated to machine i . Then:

$$w_i(a) = \sum_j a_{ij} \ell_j$$

4. Online Advertising: Each alternative a allocates a set of advertising slots, where $a_{ij} = 1$ if slot j is allocated to advertiser i . Each advertiser has some utility v_i for each unique viewer who sees his ad. He may receive many advertising slots, but the same viewer might see multiple slots. His total utility is proportional to the number of viewers who see his ad. Let E_j be the set of viewers who see slot j . Here:

$$w_i(a) = \left| \bigcup_{j: x_{ij}=1} E_j \right|$$

We will be interested in *monotone* choice rules: i.e. choice rules so that agents can only *increase* their value for the final allocation by raising their bid.

Definition 2 (Monotone Choice Rule) A choice rule X for a single parameter domain is *monotone-non-decreasing* in v_i if for all $v_{-i} \in \mathbb{R}^{n-1}$, and for every $v'_i \geq v_i$:

$$w_i(X(v_i, v_{-i})) \leq w_i(X(v'_i, v_{-i}))$$

For example, in a single item auction, this constraint requires that if an agent wins the item at bid v_i , he must also win the item at all higher bids $v'_i > v_i$.

We will prove that an allocation rule can be made truthful (by pairing it with an appropriate payment rule) if and only if it is monotone.

Theorem 3 A mechanism defined in a single parameter domain can be made truthful if and only if $X(v)$ is monotone non-decreasing for all v_i . In this case, it can be made truthful by using payment rule:

$$P(v)_i = v_i w_i(a^*) - \int_0^{v_i} w_i(X(z, v_{-i})) dz$$

where $a^* = X(v)$.

Proof To simplify notation, fix some agent i and v_{-i} , write v for v_i , and write $y(v)$ for $w_i(X(v, v_{-i}))$. (i.e. in a single item auction, we now write $y(v) = 1$ if i is allocated at bid v , and 0 otherwise).

First we show the backwards direction – assuming $X(v)$ is monotone non-decreasing and the payment rule is as given, we show that the auction is truthful. What we need to show is:

$$v \cdot y(v) - P(v)_i \geq v' \cdot y(v') - P(v')_i$$

for all v' . Plugging in the definition of the payment rule, we have to show:

$$v \cdot y(v) - v \cdot y(v) + \int_0^v y(z) dz \geq v' y(v') - v' y(v') + \int_0^{v'} y(z) dz$$

Which is equivalent to showing:

$$\int_0^v y(z) dz \geq \int_0^{v'} y(z) dz - (v' - v)y(v') \tag{1}$$

To show this, consider two cases:

1. **Case 1:** $v' > v$. In this case, equation 1 becomes:

$$\int_v^{v'} y(z) dz \leq (v' - v)y(v')$$

But this is true by monotonicity. We know that $y(v') \geq y(z)$ for all $z \leq v'$, and so:

$$\int_v^{v'} y(z) dz \leq \int_v^{v'} y(v') dz = (v' - v)y(v')$$

This is also easy to see by drawing a picture.

2. **Case 2:** $v' < v$. In this case, equation 1 becomes:

$$\int_{v'}^v y(z) dz \geq (v - v')y(v')$$

Again, this follows from monotonicity since we know that $y(v') \leq y(z)$ for all $z \geq v'$. Hence, we have:

$$\int_{v'}^v y(z) dz \geq \int_{v'}^v y(v') dz = (v - v')y(v')$$

Once again, a picture helps too.

It remains to prove the forwards direction. We must show that if we are given a truthful mechanism, its allocation rule must be monotone. Suppose we are given an arbitrary truthful mechanism over a single parameter domain. Fix any $v' > v$. By truthfulness, we must have:

$$v \cdot y(v) - P(v)_i \geq v \cdot y(v') - P(v')_i$$

since a bidder with valuation v cannot benefit by misreporting value v' . However, we also know that a bidder with valuation v' cannot benefit by misreporting v , and so we also have:

$$v' \cdot y(v') - P(v')_i \geq v' \cdot y(v) - P(v)_i$$

Adding these two inequalities, we get:

$$v \cdot y(v) + v' \cdot y(v') \geq v \cdot y(v') + v' \cdot y(v)$$

Rearranging, we get:

$$(v' - v)y(v') \geq (v' - v)y(v)$$

Since $v' - v > 0$, we can divide to obtain:

$$y(v') \geq y(v).$$

So the allocation rule must be monotone! ■