6.896: Topics in Algorithmic Game Theory

Lecture 21

Yang Cai

Overview

- Introduction to Frugal Mechanism Design
- Path Auctions
- Spanning Tree Auctions
- Generalization

Procurement

- The auctioneer is a ***buyer***, she wants to purchase goods or services.

- Agents are ***sellers***, who have costs for providing the good or service.

- The auctioneer's goal is to ***maximize the social welfare***.

What is the auctioneer's payment?

Single Good

- *Vickrey's auction.*

- *Payment = second cheapest price.*

Multiple Goods

In general, we might want to procure sets of goods that combine in useful ways, e.g. be a spanning tree of a graph.

We can use *VCG*!

- ***When does VCG never pay more than the cost of the second cheapest set of goods?***

- ***When does no incentive compatible mechanism achieve a total payment of at most the second cheapest set of goods?***

- ***If no mechanism can achieve that the total payment is at most the second cheapest price, what is the mechanism that guarantees the best worst-case approximation to it?***

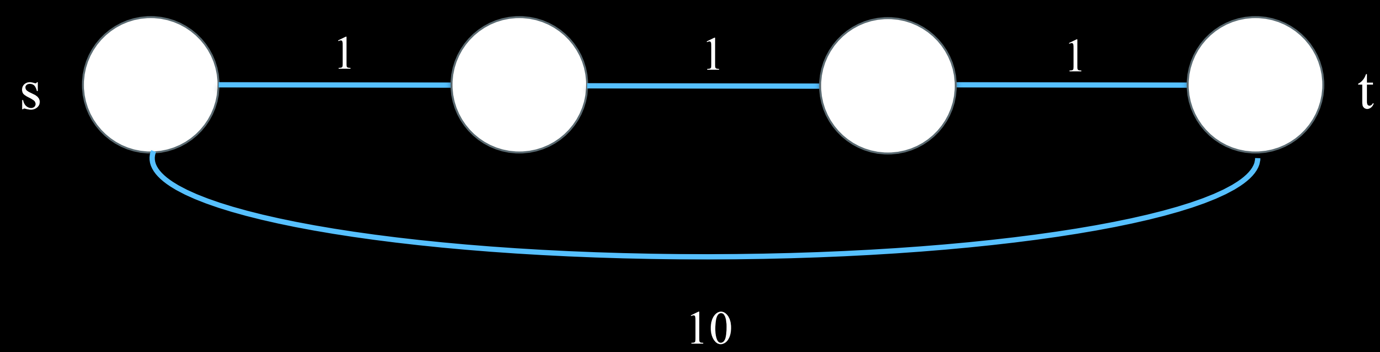
Paths & Spanning Trees

We consider the cost a buyer incurs in procuring a set of two paradigmatic systems: *paths* and *spanning trees*.

- ***Path auctions: Given a network, the auctioneer wants to buy a s-t path. Each edge is owned by a different agent. The auctioneer will try to buy the shortest path (maximize the social welfare).***

- ***Spanning tree auctions: Given a network, the auctioneer wants to buy a spanning tree. Each edge is owned by a different agent. The auctioneer will try to buy the minimum spanning tree (maximize the social welfare).***

Example 1.1 (path auction)



- shortest path = 3

VCG (with Clarke Pivot Rule)

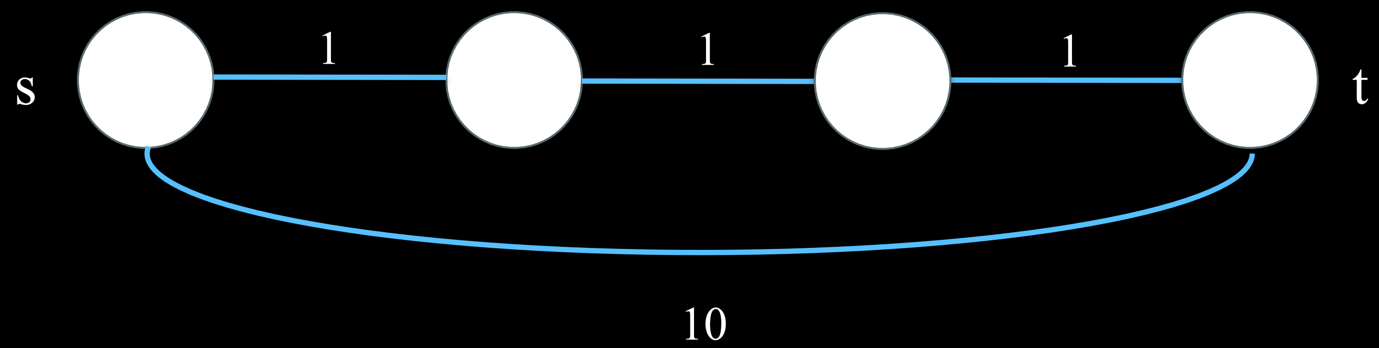
Def: A mechanism $(f, p_1, \ldots, p_n)$ is called a Vickrey-Clarke-Groves (VCG) mechanism if

(i) $f(v_1, \ldots, v_n) \in argmax_{a \in A} \sum_i v_i(a)$

(ii) Choose $h_i(v_{-i}) = max_{b \in A} \sum_{j \neq i} v_j(b)$ (Clarke Pivot Rule)

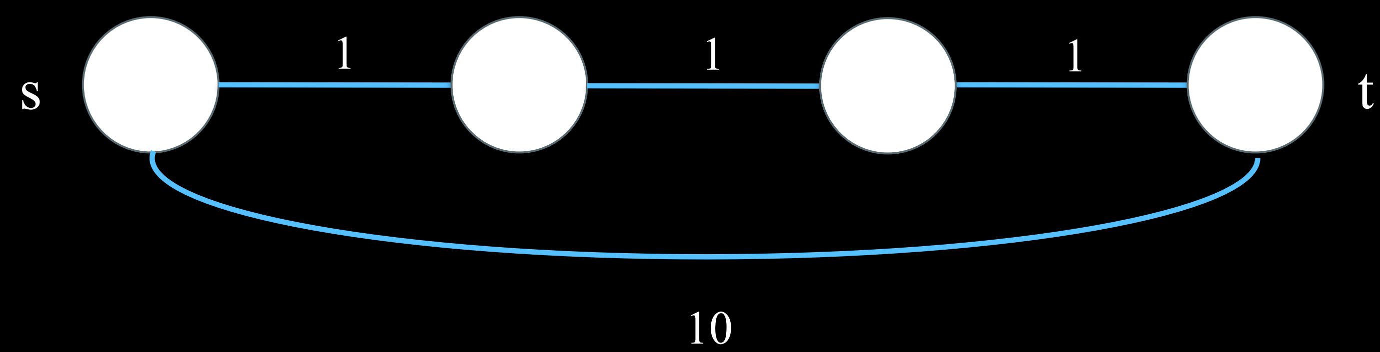
(iii) Payment $p_i(v_1, \ldots, v_n) = h_i(v_{-i}) - \sum_{j \neq i} v_j(f(v_1, \ldots, v_n))$

Example 1.1 (path auction)



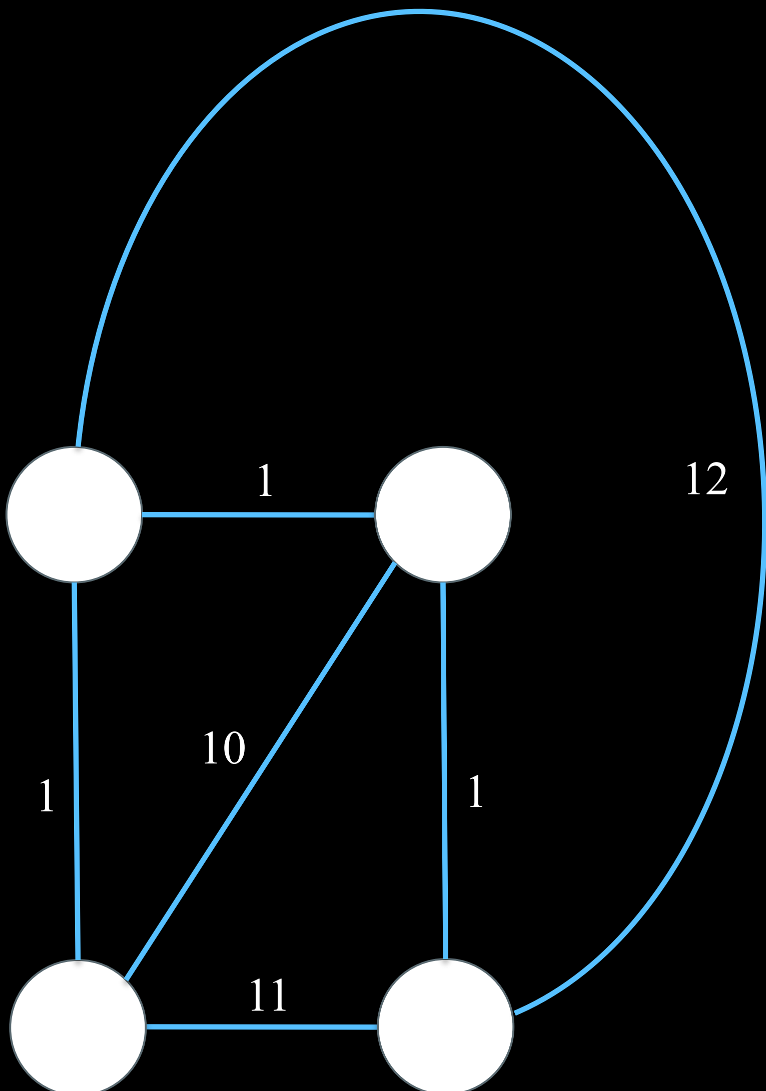
- shortest path = 10
- payment = 10 − (3 − 1) = 8

Example 1.1 (path auction)



- VCG payments = $[10 - (3 - 1)] \times 3 = 24$
- second cheapest path = 10
- overpayment ratio = $24 / 10$

Example 1.2 (spanning tree auction)



- VCG payments = 10 + 10 + 11 = 31
- second cheapest edge disjoint spanning tree = 10 +11+12 = 33
- overpayment ratio = 31 / 33

Frugal Mechanism Design

- ***The mechanism should minimize the total cost paid.***

- ***The mechanism should be frugal even in worst-case. (not the Bayesian setting)***

- ***In path auctions, VCG pays more than the second cheapest cost path. In spanning trees, it does not.***

Frugal Mechanism Design

Questions that we will explore:

- ***Does VCG on spanning trees never cost much more than the second cheapest (disjoint) spanning tree cost?***

- ***How bad can VCG on paths be in comparison to the second cheapest (disjoint) path cost?***

- ***If VCG on paths can be very bad, is there some other mechanism that does well?***

Path Auctions

Path Auction

We know VCG's payment may be more than the cost of the second cheapest path.

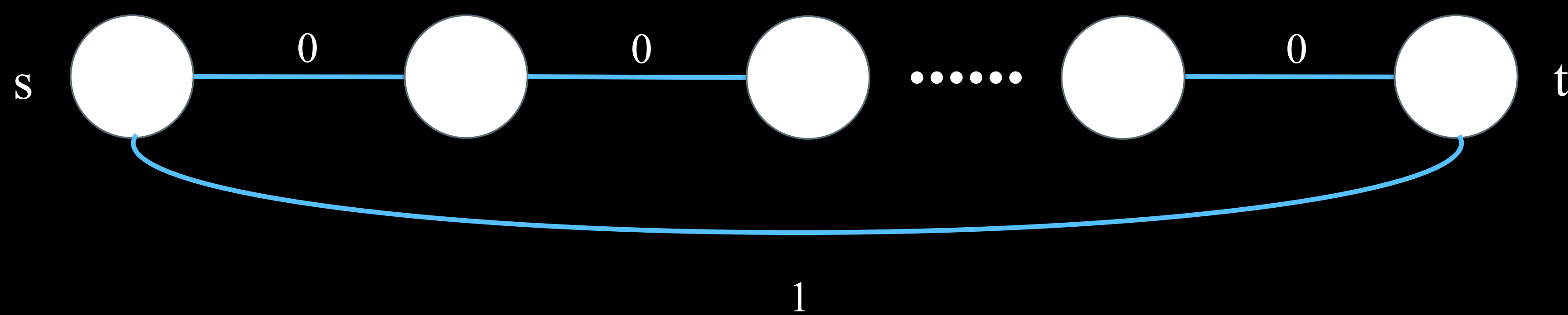
But how bad can VCG be?

As bad as one might imagine, could be a factor of $\Theta(n)$ more than the second cheapest path cost.

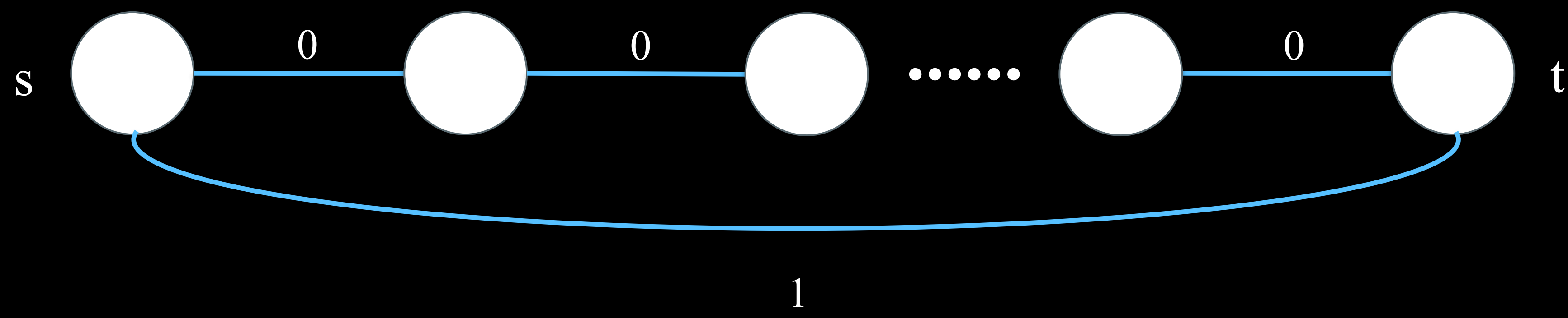
Path Auction

Proposition: There exists a graph G and edge valuation ***v*** where VCG pays a $\Theta(n)$ factor more than the cost of the second cheapest path.

Proof: Consider the following graph:



Proof (cont):



The VCG mechanism selects the top path (which has total cost zero). Each edge in the top path is paid 1. There are n-1 edges resulting in VCG payments totaling n-1. The second cheapest path cost is the bottom path with total cost 1. Therefore, the overpayment ratio is $\Theta(n)$.

Path Auction

Why does VCG has such poor performance?

- **Is it a flaw of the VCG?**

- **Is this worst-case *overpayment* an intrinsic property of any incentive compatible mechanism?**

Path Auction

Theorem: For any incentive compatible mechanism $\mathcal{M}$ and any graph G with two vertex disjoint s-t paths P and P', there is a valuation profile v such that $\mathcal{M}$ pays an $\Omega(\sqrt{|P||P'|})$ factor more than the cost of the second cheapest path.

Corollary: There exists a graph for which any incentive compatible mechanism has a worst-case $\Omega(n)$ factor overpayment.

Path Auction

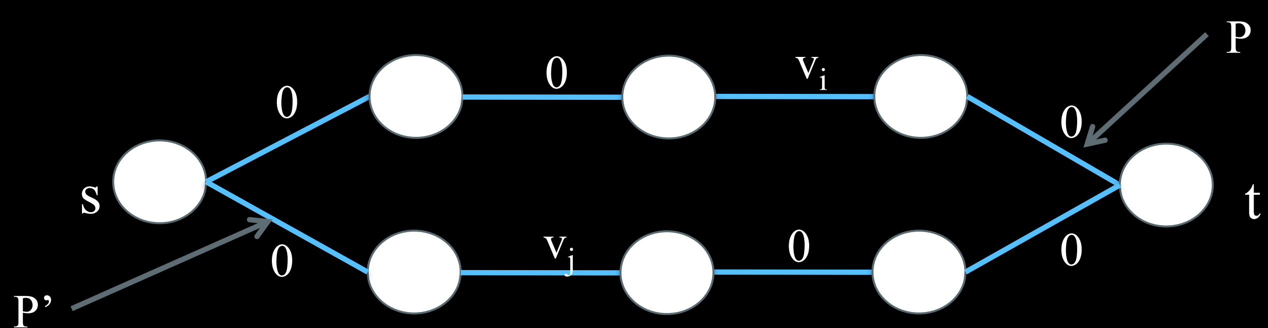
Theorem: For any incentive compatible mechanism $\mathcal{M}$ and any graph G with two vertex disjoint s-t paths P and P', there is a valuation profile v such that $\mathcal{M}$ pays an $\Omega(\sqrt{|P||P'|})$ factor more than the cost of the second cheapest path.

Proof: Let k= |P| and k' =|P'|. First we ignore all edges not in P or P' by setting their cost to infinity. Consider edge costs $\mathbf{V}^{(i,\ j)}$ of the following form.

Proof (cont):

$V^{(i,j)} \Rightarrow$

- the cost of the i-th edge of P is $v_i = 1/\sqrt{k}$,
- the cost of the j-th edge of P’ is $v_j = 1/\sqrt{k'}$, and
- all other edges cost zero.



Proof (cont):

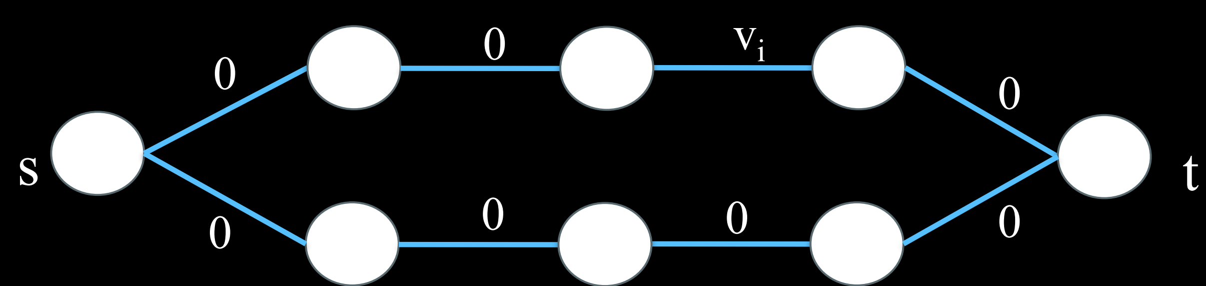
Notice that $\mathcal{M}$ on $\mathbf{V}^{(\mathbf{i},\,\mathbf{j})}$ must select either all edges in path P or all edges in path P' as winners. We define the directed bipartite graph G' = (P, P', E') on edges in path P and P'. For any pair of vertices (i, j) in the bipartite graph, there is either a directed edge (i, j) in E' denoting $\mathcal{M}$ on $\mathbf{V}^{(\mathbf{i},\,\mathbf{j})}$ selecting path P' (called "forward edges") or a directed edge (j, i) denoting $\mathcal{M}$ on $\mathbf{V}^{(\mathbf{i},\,\mathbf{j})}$ selecting path P (called "backwards edges").

Proof (cont):

Notice that the total number of edges in G' is kk'. WLOG, assume that there are more forward edges than backwards edges. G' has at least kk'/2 forward edges. Since there are k edges in path P, there must be one edge i with at least k'/2 forward edges. Let N(i) with $|N(i)| \geq k'/2$ represent the neighbors of i in the bipartite graph.

Proof (cont): Consider the valuation profile $\mathbf{V}^{(i,0)}$ of the following form

- the cost of the i-th edge of P is $v_i = 1/\sqrt{k}$, and
- all other edges cost zero.



Proof (cont):

Notice that by definition of N(i), for any j in N(i), $\mathcal{M}$ on $\mathbf{V}^{(\mathbf{i},\,\mathbf{j})}$ selects path P'. Since $\mathcal{M}$ is incentive compatible, its allocation rule must be monotone: if agent j is selected when bidding v_j, it must be selected when bidding 0 (WMON). Therefore, $\mathcal{M}$ selects P' on $\mathbf{V}^{(\mathbf{i},\,\mathbf{0})}$.

Proof (cont):

Also, for any j in N(i), the payment should be at least $1/\sqrt{k'}$. Since when the valuation profile is $\mathbf{V}^{(i,j)}$, the payment should be at least $1/\sqrt{k'}$. Otherwise, j will receive negative utility. By the direct characterization of incentive compatible mechanisms, we know when other bidders' valuations and the outcome are the same, the payment should also be the same. So payment for j is at least $1/\sqrt{k'}$ when the valuation profile is $\mathbf{V}^{(i,0)}$.

So on $\mathbf{V}^{(i,0)}$, the total payment of $\mathcal{M}$ is at least $N(i)\times 1/\sqrt{k'} \geq \sqrt{k'}/2$. Remember that the second cheapest path is P with cost $1/\sqrt{k}$. Therefore, the overpayment ratio is $\sqrt{k}k'/2$.

■

Path Auction

Remarks: 1. ***No incentive compatible mechanism is more frugal than VCG in worst-case.***

2. But it is possible to design mechanisms that are better than VCG on non-worst-case inputs

Spanning Tree Auctions

Spanning Tree Auction

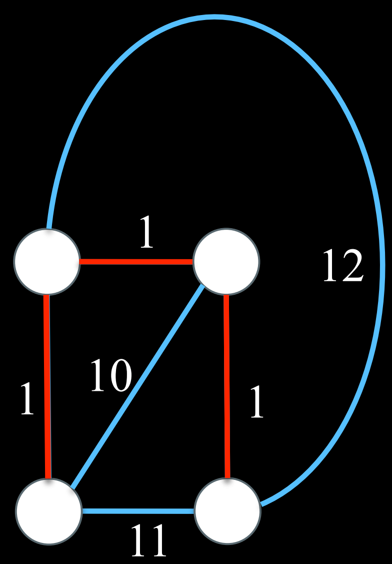
We will show that the overpayment of VCG for spanning trees is minimal.

Theorem: The total VCG cost for procuring a spanning tree is at most the cost of the second cheapest disjoint spanning tree.

Spanning Tree Auction

To prove this main theorem, we make the following definitions.

Definition: The ***replacement*** of e in a spanning tree T of a graph G=(V,E) are the edges $e' \in E$ that can replace e in the spanning tree T. I.e., $\{e' : T/\{e\} \cup \{e'\} is\ a\ spanning\ tree\}$. The ***cheapest replacement*** of e is the replacement with minimum cost.



- The MST is given by three edges with cost 1.
- The replacements of the left-most 1 in the MST are the edges with cost 10 and 11.
- The cheapest replacement is therefore the 10 edge

Spanning Tree Auction

Definition: The ***bipartite replacement graph*** for edge disjoint trees T_1 and T_2 is G'= (T_1, T_2, E') where $(e_1, e_2) \in E'$, if e_2 is a replacement of e_1 in T_1.

Remark: The neighbors N(e) of e that belongs to T_1 (respectively T_2) in the bipartite replacement graph are simply the replacements of e in T_1 (respectively T_2).

Proof Plan

1. The total VCG cost is at most the sum costs of the cheapest replacements of the MST edges.

2. If there is a ***perfect matching*** in the bipartite replacement graph for cheapest spanning tree T_1 and the second cheapest disjoint spanning tree T_2 then the total VCG cost is at most the cost of T_2.

3. There is a perfect matching in the bipartite replacement graph given T_1 and T_2.

VCG Payments and Cheapest replacements

Lemma: VCG pays each agent (edge) the cost of their cheapest replacement.

The proof of this lemma is based on the following basic facts about minimum spanning tree.

Fact 1: The cheapest edge across any cut is in the minimum spanning tree.

Fact 2: The most expensive edge in any cycle is not in any minimum spanning tree.

Proof :

Consider an edge e_1 in the MST T_1. Removal of this edge from T_1 partitions the graph into two sets A and B. The replacements for e_1 are precisely the edges that cross the A-B cut. Since e_1 is the only edge in the MST across the A-B cut, by Fact 1 it must be the cheapest edge across the cut. Let e_2 be the second cheapest edge across the A-B cut (and therefore e_1's cheapest replacement).

We claim that if we were to raise the cost of e_1 it would remain in the MST until it exceeds the cost of e_2 after which e_2 would replace it in the MST.

Proof (cont):

First, e_1 is in the MST when bidding less than e_2. This follows from Fact 1 as with such a bid, e_1 is still the cheapest edge across the A-B cut. Second, e_1 is not in the MST when bidding more than e_2. This follows because there is a cycle in $T_1 \cup \{e_2\}$ that contains e_1 and e_2. Since e_2 is not in the T_1 and all other edges in the cycle are, it must be that e_2 is the most expensive edge (by Fact 2). However, if e_1's cost is increased to be higher than that of e_2, e_1 would become the most expensive edge in the cycle. Fact 2 then implies that with such a cost e_1 could not be in the MST.

Proof (cont):

Now, we have proved the claim that if we were to raise the cost of e_1 it would remain in the MST until it exceeds the cost of e_2 after which e_2 would replace it in the MST. We still need to argue that the payment for e_1 is the cost of e_2, when e_1 is in the MST.

We know that the payment for e_1 will remain the same as long as e_1 is in the MST. But to guarantee that e_1 has positive utility, the payment should be higher than the cost. So the payment is at least as high as e_2's cost. But on the other hand, the payment should not exceed the cost of e_2. Otherwise, if e_1's cost is between e_2's cost and the payment, e_1 can increase his utility by misreporting his cost to be lower than e_2. Because, if he is truthful, he will not be in the MST and his utility will be 0, but if he misreports his cost to be smaller than e_2's, he will be in the MST and receive positive utility. Therefore, the payment is exactly the cost of e_2.

■

Bipartite Replacement Graph and VCG Payment

Lemma: For cheapest and second cheapest (disjoint) spanning trees T_1 and T_2, if there is a perfect matching in the bipartite replacement graph then the VCG payments sum to at most the cost of T_2.

Proof: Let M be a perfect matching in the bipartite replacement graph for T_1 and T_2. For e_1 in T_1 let $M(e_1)$ denote the edge e_2 in T_2 to which e_1 is matched in M. For e_1 in T_1, let $r(e_1)$ denote the cost of the cheapest replacement for e_1. And let $c(e)$ denote the cost of edge e. Notice that $r(e_1) \leq c(M(e_1))$.

$$\begin{aligned} VCG\ payments &= \sum_{e_1 \in T_1} r(e_1) \\ &\leq \sum_{e_1 \in T_1} c(M(e_1)) \\ &= \sum_{e_2 \in T_2} c(e_2) \end{aligned}$$

■

Perfect Matching

Lemma: The bipartite replacement graph for two edge disjoint spanning trees T_1 and T_2 has a perfect matching.

The proof follows from Hall's Theorem.

Definition: Let $N(v)$ denote the neighbors of a vertex v in a graph $G = (V, E)$. The neighbors of a set of vertices $S \subset V$ is the union of the neighbors of each vertex in the set, i.e., $N(S) = \bigcup_{v \in S} N(v)$.

Perfect Matching

Definition (Hall's condition): A bipartite graph $G = (A, B, E)$ satisfies Hall's condition if all subsets $S \subseteq A$ satisfy $|S| \leq |N(S)|$.

Theorem (Hall's Theorem): For a bipartite graph $G = (A, B, E)$, G has a perfect matching if and only if it satisfies Hall's condition.

Perfect Matching

Lemma: The bipartite replacement graph for two edge disjoint spanning trees T_1 and T_2 has a perfect matching.

We only need to argue that Hall's condition holds in the bipartite replacement graph for any T_1 and T_2.

Proof: Consider some subset $S_1 \subset T_1$. Let $k = |S_1|$. When we remove S_1 from T_1 the remaining tree edges do not span G. In particular there are exactly k+1 connected components. We can view these connected components as a "super-node" and S_1 as a spanning tree of these super-nodes. Let $S_2 \subset T_2$ be the set of edges from T_2 that connect any pair of super-nodes. We now make two arguments.

Proof (cont):

1. Any $e_2 \in S_2$ is a replacement for some $e_1 \in S_1$, i.e., $S_2 \subseteq N(S_1)$.

Consider any $e_2 \in S_2$. By definition, e_2 connects two super nodes. S_1 is a spanning tree of the super-nodes which implies that there is exactly one path in S_1 that connects them. The edge e_2 is a replacement for any edge e_1 in this path.

2. $|S_2| \geq k$.

Since T_2 spans the original graph and S_2 is precisely the set of edges from T_2 that are between super-nodes, S_2 must span the graph of super-nodes. There are k+1 super-nodes. Therefore, such a set of spanning edges must be of size at least k.

Combining the above two arguments: $|N(S_1)| \geq |S_2| \geq k = |S_1|$. Thus, Hall's condition holds for the bipartite replacement graph. Hall's Theorem then implies a perfect matching exists.

■

Spanning Tree Auction

The proof of the theorem follows from the three lemmas we showed above.

Theorem: The total VCG cost for procuring a spanning tree is at most the cost of the second cheapest disjoint spanning tree.

Generalizations

Generalizations

We can generalize our results for spanning trees to matroid set systems.

Matroids are set systems where analogs of Fact 1 and Fact 2 hold.

These facts imply a ***single-replacement*** property.

Generalizations

Besides matroids, is there any other set systems for which VCG overpayment is minimal? It turns out there is a very precise answer to this, but stating it requires moving beyond the framework discussed in this lecture. Instead we summarize.

Proposition: There is a very precise sense in which matroid set systems are the only set systems for which VCG has no overpayment.