Advanced Algorithms - Lecture	December 18, 2005
Recitation 8	
Lecturer: Shahar Dobzinski	*Scribe: Or Sattath*

1 Finishing The Mincut Problem

In the previous recitation we saw the min-multicut problem. We tried to find a solution where the given graph is a tree. definitions: $depth(v)$ = The distance from the root (the root is arbitrary). $lca(v_1, v_2)$ = the vertex with the minimum depth that is in the path between v_1 and v_2 (there is only one path because G is a tree).

The Algorithm
Last time we saw the following algorithm (see the full details in the previous recitation).

1. INIT:
   $f := 0$
   $D = \emptyset$
2. FLOW: for each vertex v in a non increasing depth order (i.e. the deepest leaf is first) s.t. $v = lca(s_i, t_i)$, flow in a greedy manner between s_i to t_i.
   add all the saturated edges to D.
3. BACKWARD REMOVAL: remove all the edges in a reversed order to the one inserted - if it's still a legal multi-cut.

We still had to show the following lemma:

Lemma 1 *let (s_i, t_i) a couple with a flow that isn't 0, and let $v = lca(s_i, t_i)$. Then, at most one edge is chosen in the path $s_i \hookrightarrow v$ (and also in $t_i \hookrightarrow v$).*

Proof: Assume the contrary. So, there is an example as in figure 1. Assume both e and e' are in the multicut. So:

1. e' is along step 3.
2. e wasn't removed when we checked it. Therefore, there exists a couple (s_j, t_j) s.t. e is the only edge in the path $s_j \hookrightarrow t_j$ that is in D. Define $u = lca(s_j, t_j)$.
3. $depth(u) > depth(e')$ because e' isn't in the path $s_j \hookrightarrow t_j$. Therefore the situation is as in figure 2.

Figure 1:

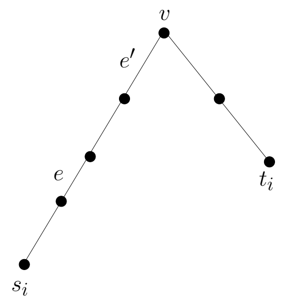


Figure 2:

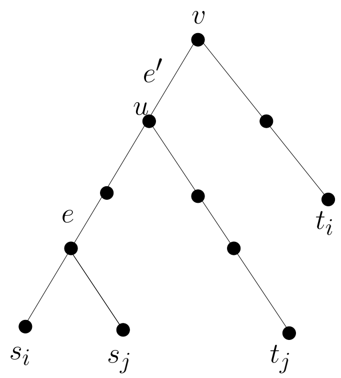
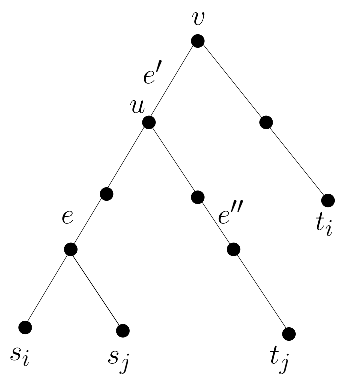


Figure 3:



4. We flowed as much as was possible between $s_i \hookrightarrow t_i \Longrightarrow$ e was added in this itereation or later.

5. after we checked u there has to be an edge e'' between s_j to t_j in D. It can't be e because of 4, therefore the situation is as figure 3.

6. v is deeper than $u \Longrightarrow e$ was added after e''. so e'' is in D when e was checked (in step 3).
   Contadiction to 2.

■

2 Economics and Computation

We'll move to a different subject. We'll focus on Truthful Mechanisms. We'll introduce it by an example:

2.1 The Auction Problem

Suppose we have one item for sale, and 2 players, (A,B). Every player has a value for this product, $v_A, v_B \geq 0$. The goal is to give the product to the person who needs it most (the goal of the "seller" *isn't* to earn as much money as possible). In that way this is a truthful mechanism - you want the players to reveal their real value for the product.

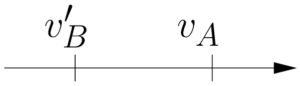
Our assumption is that every buyer wants to maximize his profit function, but there's no interest regarding the other players. Another assumption that if there's no interest of not telling the truth, a player tells the truth.

Proposition 1 (The First Price Auction) *Every person is asked to bid a price. The winner is the highest bidder, and he pays the offer that he offered. The losers pay nothing.*

The problem with this proposition: there's an interest for lying. If A can estimate v_B, his best interest is to bid $v_B + \epsilon$. The problem is that we'll give the product to the person that can estimate the other people pricing, and not to the person that actually has the maximum value for that product.

Proposition 2 (The Second Price Auction, or Vickrey Auction) *Both players are asked to bid. The winner is the person that gave the highest bid, and he pays the price that the other offered.*

Figure 4:



Lemma 2 *The player with the highest value, w.l.o.g., A (i.e.* $v_A > v_B$*), doesn't have interest of lying, and therefore A will say the truth -* v_A*.*

Proof: Let v'_A, v'_B be the bids of A and B. The situation is as in figure 4. A can bid a price $v'_A \leq v_A$. It should be more than v'_B, otherwise, B will win. But A will not gain anything from changing in the region $[v_A, v'_B]$ because anyway he'll pay v'_B. ■

Lemma 3 *The player with the lowest value, B, doesn't have interest of lying.*

Proof: We already know that A will offer v_A. $v'_B \leq v_B < v'_A = v_A$, so B will lose anyway. ■ Notice that when the player thinks about what he'll do he don't know if he has the highest

value or not, but, he knows that either way, he don't have an interest for not revealing his true value. So, he reveals his value.

2.2 The Clacrke Tax Problem

The government wants to build a bridge that would connect the island to the land. Every player in the island has some value for that bridge, for example: $v_A = 100, v_B = 200, v_C = 300$. The bridge cost's to the goverment $v_G = 450$. The goverment has the money, and it's target is only to know if it's the public interest building it in that price or not i.e. whether $v_A + v_B + v_C \geq v_G$ (the government will build it if it's the public interest).
The solution: Every player is asked for what is the value of the bridge for him, and eventually every player pays his "society damage" - the cost of the bridge minus the sum of the prices of all the other players (but if this number is negative, that player will pay 0).
In our example, $P(A) = 0$ because $450 - (300 + 200) < 0$, $P(B) = 50$, and $P(C) = 150$.

Lemma 4 *Player A doesn't have an interest for offering more than* v_A*.*

Proof: Let v'_A be what A offered ($v'_A > v_A$). If A assumes that $v_A + v'_B + v'_C > 450$ then he has no interest of offering more then v_A - it will not affect how much A pays, nor the decision whether the bridge will be built or not. Otherwise, he has an interest, because this is the only way which could make the bridge to be built. But, A will have to pay $450 - v'_B - v'_C > v_A$ which is more then it's worth him. So this manipulation doesn't help A. ■

Lemma 5 *Player A doesn't have an interest for offering less then v_A.*

Proof: Again, let v'_A be what A offered (this time $v'_A < v_A$). if A assumes that $v_A + v'_B + v'_C > 450$ then A doesn't has an interest for giving an offer that is less then v_A - it might make the bridge not to be built, and it will not affect on how much he'll pay. If it's built, he'll pay either way $450 - v'_B - v'_C$ (it's not depended on what he offered). ■

2.3 Combinatorial Auction

In the next recitation we'll go deeply into this problem.
The input of the problem: a set M of different items. $|M| = m$.
For each player i there is a function:
$V_i : 2^m \rightarrow \mathbf{R}^+$ where $V_i(emptyset) = 0$, and
V_i is monotonic: $T \subset S \Rightarrow V_i(S) \geq V_i(T)$
The goal: find the partition $S_1, S_2, ..., S_n$ s.t. $\sum_{i=1}^{n} V_i(S_i)$ is the maximum.