CS 559: Algorithmic Aspects of Computer Networks **Spring 2010**

Lecture 18 — March 25,30

Lecturer: John Byers *BOSTON UNIVERSITY* *Scribe: Maryam Ghasemi*

In today's lecture, we learned fundamental basic about *Algorithmic mechanism design* [2]. Most of research was done in routing or load balancing assuming that each participant acts as instructed and information is globally known while they follow their own selish interest rather than algorithm. Mechanism Design as a subfield of game theory deals with such problems. It tries to motivate agents to tell the truth by giving payments to the agents. Thus the mechanism is able to guide the agents towards a social choice.

we define the following properties:

- There are n strategic agents $A^1, ..., A^n$, with private types $t^1, ..., t^n$

  Each agent has associated with cost or some kind of valuation about how unwilling they are to execute tasks which is known as private type. for example routing a package from i to j has some cost for agent A^i that can be shown as constant.

- k tasks

- O (output) which maps a task to agents.

For social optimality, we suppose that there is a planner which takes tasks and agents' private types and assigns a task to each agent such that chosen assignments are cost minimizing. But if the aganets don't reveal their private types, social optimality is not realistic. So mechanism design try to solve this problem.

An agent's valuation function is defined as $v^i(t^i, O) = -\sum_{k \in Z_i} t^i_k$ which Z_i is the set of jobs allocated to agent i (the output space Z is equal to $Z_1 \cup ... \cup Z_n$) and t^k_i is agent i's private type for job k. Indeed valuation function is representative of how unhappy agent i is of allocating jobs Z_i.

Each agent i has some types in his mind and he reveals an strategy a^i. The mechanism pays $p^i = p^i(a^1, ..., a^n)$ to agent i in such a way to get agent i to reveal t^i i.e. $a^i = t^i$.

Agent is utility is $u^i = p^i + v^i(t^i, O)$, and it is this quantity that the agent seeks to maximize.

A mechanism is *strategy proof* if:

$$\forall t^i, a^i, a^{-i}: \quad p^i(a^{-i}, t^i) + v^i(t^i, O(a^{-i}, t^i)) \geq p^i(a^{-i}, a^i) + v^i(t^i, O(a^{-i}, a^i)).$$

(where a^i denotes the vector of strategies of all agents except agent i)
Which means that for each agent i revealing the truth value of t^i is the best strategy.

18.1 VGC (Vickrey-Clarkw-Groves) payments:

Consider player i and $O(t^1, ..., t^n)$

- Consider the case when agent j is not present,
- The disutility is non-decreasing, i.e. i may incur additional cost S_{ij}
- i would be indifferent to j dropping out or making a side payment S_{ij} to incent j to participate
- value that j contributes by playing is $\sum_i S_{ij}$ which is j's VCG payment.

Now consider a point routing problem. Graph G is a biconnected graph with source X and destination Y. Here players are corresponded to edges, t^i = cost for edge i to route a packet and a^i = claimed cost/strategy. The porpuse is to find a path P such that $\sum_{i \in P} t^i$ is minimized. So if edge $i \in P$ does not participate in routing $Cost(P^{alt}) \geq Cost(P)$. VCG says that edge i is owed payment $= Cost(P^{alt}) - Cost(P)$.

Today, ISPs route package to maximize profits, peering agreements orchestrated similarly. So social optimality is the last on their list and definitely, they choose their strategies in a self-interested way. This is not compatible with social optimality. So AMD + VCG-style payments primarily try to re-align incentives to incent ISPs to move toward socialy optimal routing.

18.2 Lowest Cost Path (LCP)

Feigenbaum et al [1] formulate a version of the routing-mechanism design problem to compute lowest cost path (LCP) and incent providers (nodes/ISPs) to publish their true cost.

Consider a biconnected graph G which T_{ij} = traffic intensity from i to j and c_k = cost a node (ISP) incurs in transiting a packet (assume independent of source and destination).

Let $I_k(c; i, j)$ be the indicator function for the LCP from i to j; i.e., $I_k(c; i, j) = 1$, if node k is an intermediate node on the LCP from i to j.

Price of transit a packet from i to j through node k is computed as follow:

$$p_{ij}^k = c_k I_k(c; i, j) + \underbrace{\sum_{r \in N} I_r(c|^k \infty; i, j).c_r}_{best\ path\ without\ node\ k} - \underbrace{\sum_{r \in N} I_r(c; i, j).c_r}_{best\ path\ with\ k}$$

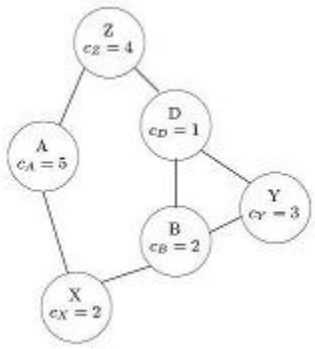

Figure 18.1. Example graph for overcharging

p_{ij}^k is zero if the LCP between i and j does not traverse k.

Feigenbaum et al [1] prove the following theorem:

Theorem 18.1. *When routing picks lowest-cost paths, and the network is biconnected, there is a unique strategyproof pricing mechanism that gives no payment to nodes that carry no transit traffic.*

Claim: Both LCP and p_{ij}^k can be computed with constant-factor overhead in existing protocol(BGP/ISPs).

p_{ij}^k can be written as $p_{ij}^k = c_k + Cost(P^{-k}(c; i, j)) - c(i, j)$ which $c(i, j)$ denotes the cost of LCP from i to j and $P^{-k}(c; i, j)$ the lowestcost k-avoiding path from i to j.

c_k and $Cost(P^{-k}(c; i, j))$ is easy to report. BGP has ability to compute cost of alternative path. Having the alternative path is not enough. Another thing we need to know is the cost of all nodes involve in alternative path.

However, VCG mechanism has two issues:

1. Who does make the payments?
2. Overcharging.

An example of overcharging in figure 18.1 occurs in sending a packet from Y to Z. The LCP is YDZ, which has transit cost 1. However, the next best path is $YBXAZ$ which has cost 9, and hence D's payment for this packet is $1 + [9 - 1] = 9$, even though D's cost is still 1. There is a tendecy here to reform the network such that some other nodes cover payment partially.

Bibliography

[1] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A bgp-based mechanism for lowest-cost routing. *Distrib. Comput.*, 18(1):61–72, 2005.

[2] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140, New York, NY, USA, 1999. ACM.