Problems at the Interface of Algorithms and Economics

Subhash Suri

Computer Science Department
UC Santa Barbara

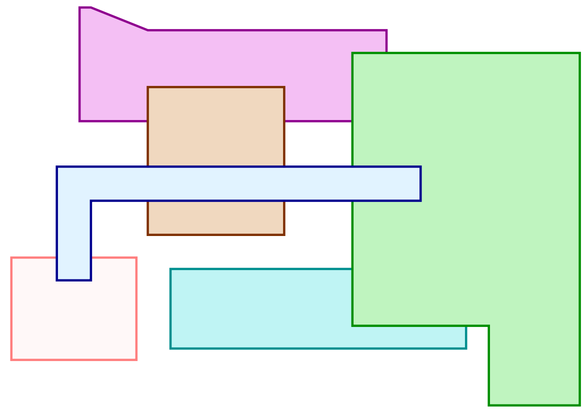
Collaborators: Hershberger, Kothari, Sandholm, Toth, Zhou

Themes

- **Auctions**.
- **VCG Payments**.
- **Nash Equilibria**.

Combinatorial Auctions

- **A set of distinct items:** $S = \{1, 2, \ldots, m\}$**.**
- **A set of bundle bids:** $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$**, where** $B_i \subset S$**.**
- **Each bid has an associated positive price.**
- **Winner determination problem is to choose a collection of item-disjoint bids with maximum total value.**



- **Originally proposed for airport landing slot auctions in seventies; repopularized in '90s by FCC and e-commerce.**

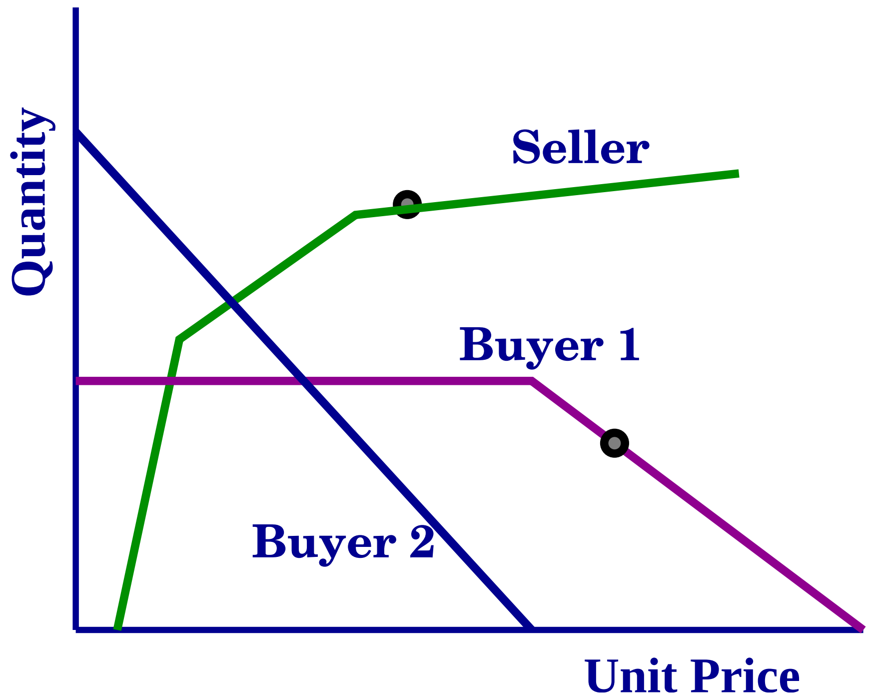
Complexity

- **Complementarity and substitutability among items.**
- **Airport takeoff slot valuable only if one can get a matching landing slot at destination.**
- **A hotel booking in Maui is valuable only if you also get a matching airline booking.**
- **FCC's wireless spectrum license auctions. Bidders have regional complementarities.**
- **However, winner determination is NPC, and inapproximable to $\Omega(n^{1-\varepsilon})$.**
- **Special cases, branch-and-bound, commercial MIP solvers etc.**

Single Good Auctions

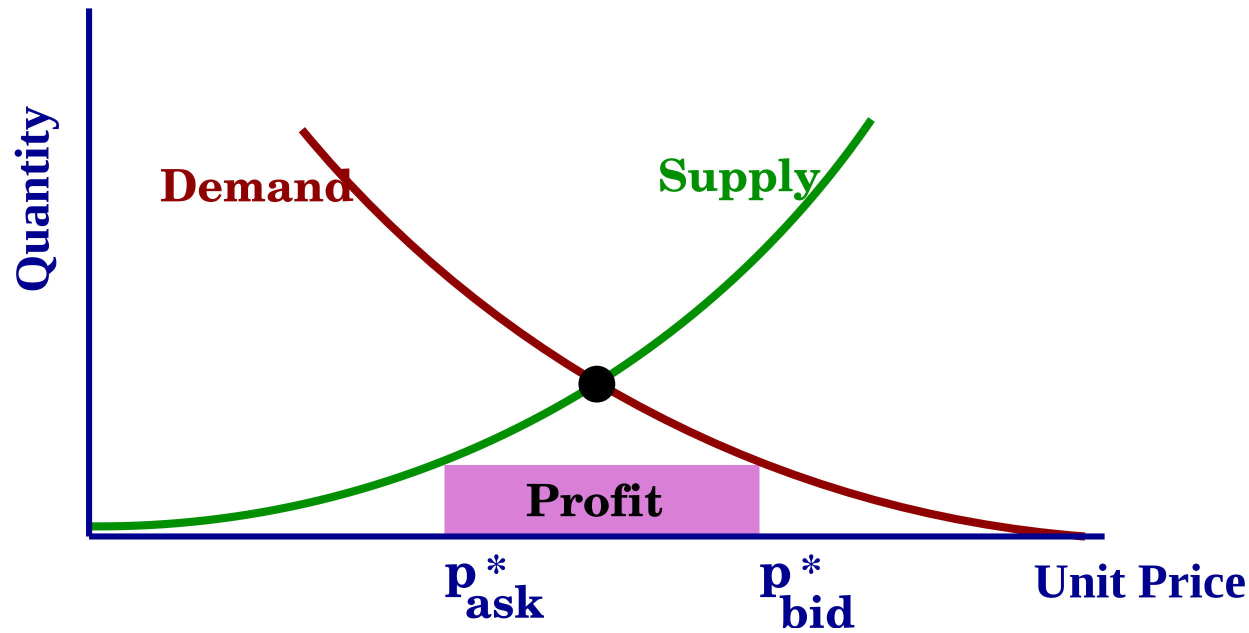
- Multiple indistinguishable units of a good.
- Examples: bandwidth, oil, raw materials, electricity, equities etc.
- Goldberg, Hartline, Karlin consider digital goods auctions: music, video etc. GHK focus on incentives and demand discovery.
- We focus on computational complexity for clearing the auction.
- If only bids on single units allowed, not very interesting. But expressive bidding (demand curves) make the problem more interesting.

Demand Curves



- **Buyers and sellers must be matched on their curves.**
- **Piecewise linear curves can express bidders' marginal decreasing values, and sellers' volume discounts.**

Pricing Policies



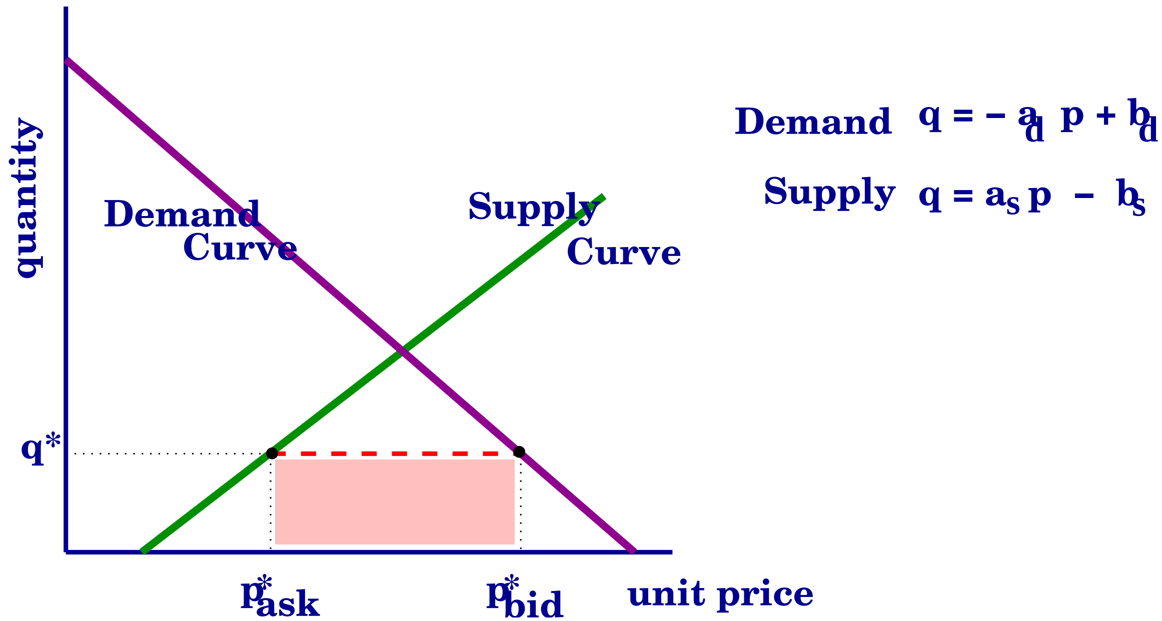
- **One price for everyone. "Equilibrium". No profit.**
- **One price for seller, one for buyers. Non-discriminatory.**
- **One price for each agent. Discriminatory.**

Various Auctions

- **Forward Auction: Single seller, multiple buyers.**
- **Reverse Auction: Single buyer, multiple sellers.**
- **Exchanges: Multiple buyers, multiple sellers.**
- **Results hold for all cases. I will say "auction" generically.**

Non-discriminatory Price Auction

- **Consider 1 buyer and 1 seller, with linear curves.**

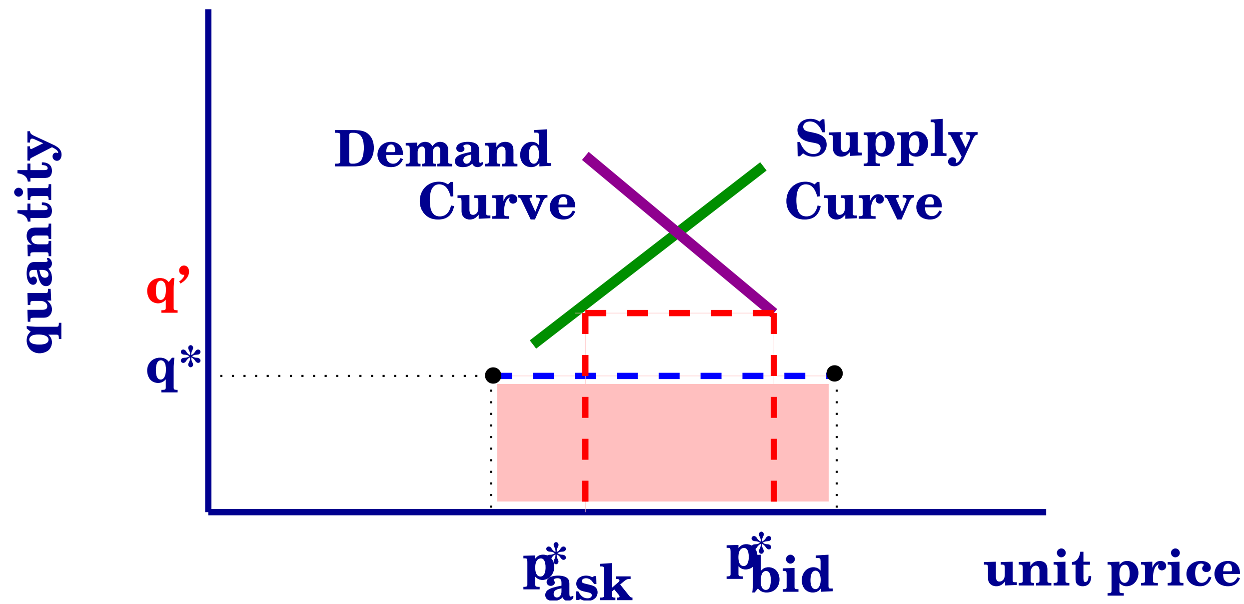


- **Optimal trade occurs at quantity** $q^* = \frac{1}{2}\left(\frac{a_s b_d - a_d b_s}{a_s + a_d}\right)$.
- **Optimal clearing prices are**

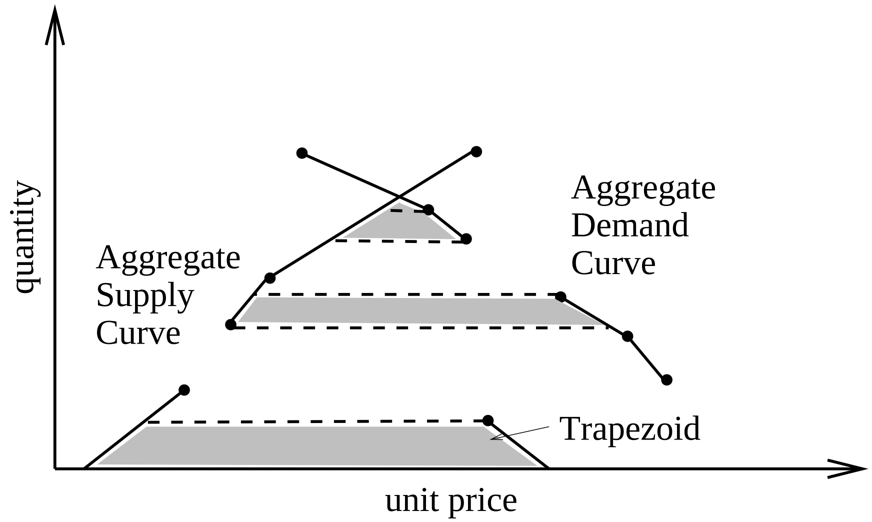
$$p^*_{ask} = \frac{1}{2}\left(\frac{b_s}{a_s} + \frac{b_s + b_d}{a_s + a_d}\right), \qquad p^*_{bid} = \frac{1}{2}\left(\frac{b_d}{a_d} + \frac{b_s + b_d}{a_s + a_d}\right)$$

General Piecewise Linear Curves

- **If buyer and seller can trade only in the quantity range $[q', q'']$, then optimum occurs either at q^* (if $q^* \in [q', q'']$), or at that endpoint of the range $[q', q'']$ which is *closer* to q^*.**



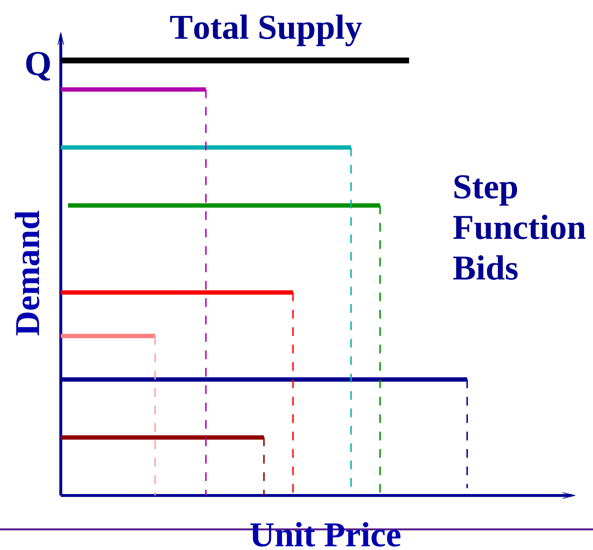
Algorithm



- **Build aggregate demand D and aggregate supply S curves.**
- **Decompose the feasible region into trapezoids.**
- **Each trapezoid is 1-seller, 1-buyer trade problem.**
- **From prices, p^*_{bid} and p^*_{ask}, determine each agent's quantity.**

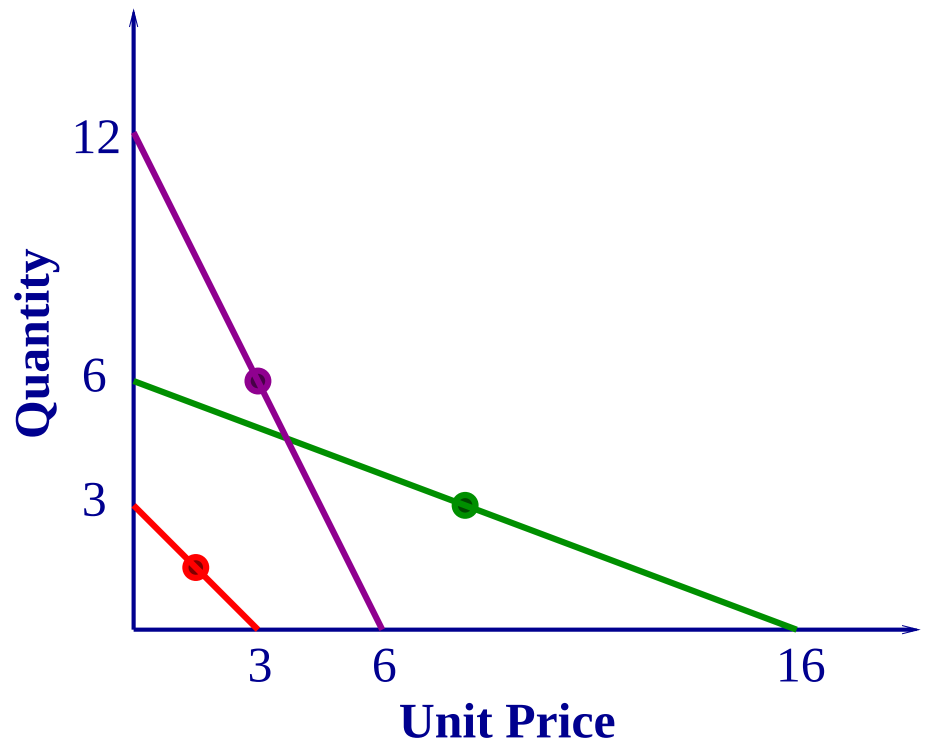
Discriminatory Price Auctions

- Agents pay/receive different prices. Potentially higher revenue.
- But the problem is NP-Complete—reduction from integer Knapsack.
- Item (size s, value v) maps to "s units at total price $\leq v$." Knapsack capacity maps to Q units for auction.



The Linear Case

- **If all bids are linear functions, then an $O(n \log n)$ algorithm.**



- **A cute combinatorial algorithm. (Greedy doesn't work.)**

An Example

- **Maximize** $\sum_{j=1}^{n} p_j q_j$ **s.t.** $q_j = -a_j p_j + b_j$ **and** $\sum_j q_j \leq Q$.
- **Eliminate p_j's from obj. and add supply constraint using Lagrangian multiplier**

$$\max \quad \left(\frac{b_j q_j}{a_j} - \frac{q_j^2}{a_j} \right) + \lambda(Q - \sum_{j=1}^{n} q_j).$$

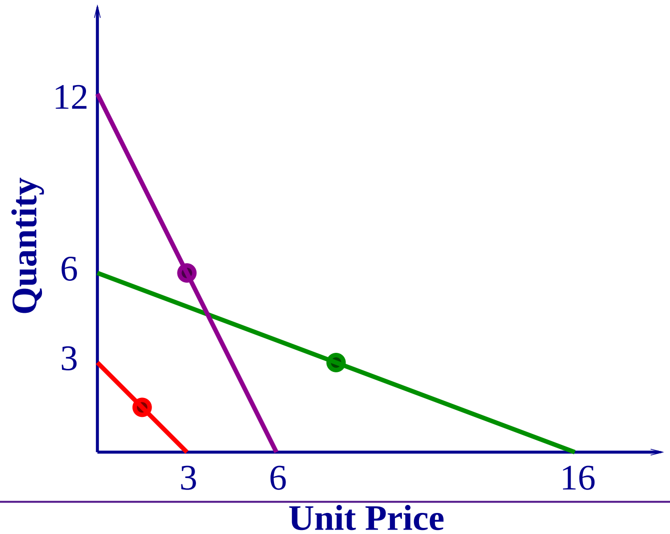
- **We get**

$$q_j = \frac{b_j}{2} - \frac{a_j}{2} \left(\frac{\sum b_j - 2Q}{\sum a_j} \right) \qquad p_j = \frac{b_j}{2a_j} + \frac{1}{2} \left(\frac{\sum b_j - 2Q}{\sum a_j} \right)$$

- **Raise clearing prices for all buyers uniformly.**
- **But invalid because q_j can be negative!**

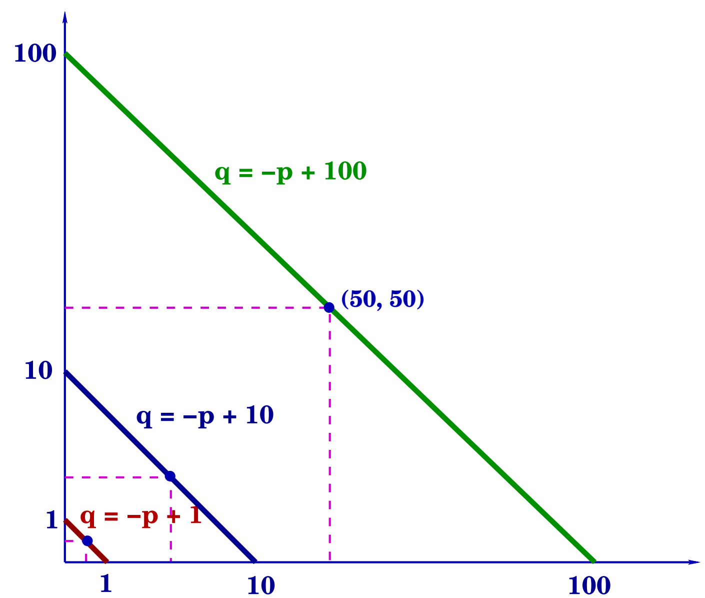
Raise-and-Drop Algorithm

1. **Bids $S = \{1, 2, \ldots, n\}$.**
2. **Initialize $(p_j, q_j) = \left(\frac{b_j}{2a_j}, \frac{b_j}{2}\right)$. If $\sum_{j \in S} q_j \leq Q$, done.**
3. **Let ℓ be bid with min p_j.**
4. **Set $p'_j = p_j + p_\ell$, $q'_j = -a_j p'_j + b_j$, for $j \in S$.**
5. **If $\sum_{j \in S} q'_j \leq Q$, output the Lagrangian solution using bids of S. Otherwise, set $S = S - \{\ell\}$, and go to step 3.**



An Example

- **Total supply** $Q = 50$.
- **Unconstrained:** $q_1 = 0.5$, $q_2 = 5$, $q_3 = 50$.
- **Revenue with buyer 3 alone:** $\$2500$.
- **Optimal revenue** $\$2512.5$**:** $q_1 = 0$, $q_2 = 2.5$, $q_3 = 47.5$,



Some Problems and Directions

- **Tractable Cases: Shoreline properties, tree structured etc. [RPH '98], [SS 01, 02].**
- **Even with rectangle shaped bids, the problem is NP-Complete.**
- **What if items were points, and all bids were on Delaunay triples? Is this tractable? Approximable?**
- **Approximation bounds for discriminatory auctions with piecewise linear curves.**
- **Extend Demand Curve auction to multi-dimensions.**
- **Independent demand curves, but tied together via buyer's budget constraints.**

Bibliography

- **M. Tennenholtz. Some tractable combinatorial auctions. AAAI '00.**
- **Rothkopf, Pekec, Harstad. Computationally manageable combinatorial auctions. Management Science, 1998.**
- **T. Sandholm and S. Suri. Market Clearability. IJCAI 2001.**
- **T. Sandholm and S. Suri. Optimal Clearing of Supply/Demand Curves. ISAAC 2002.**

Vickrey-Clark-Groves

Vickrey Payments

- **What if geometric objects had minds of their own?**
- **What if they were selfish and strategic?**
- **How would you build a DT or MST if "points" won't tell you their position?**
- **Think of *ad hoc* network nodes. Battery life is precious, yet nodes are expected to route other's packets.**
- **How would you solicit true "position, cost or type" of a point?**
- **Economists' answer: make it worth their while!**

Rationality Based Computing

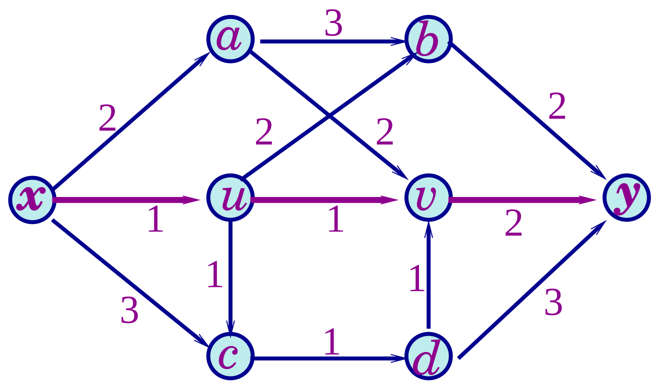
- Algorithmic Mechanism Design, algorithmic game theory. Nisan–Ronen, Feigenbaum-Papadimitriou-Shenker, Roughgarden-Tardos etc.
- Internet is a huge, dynamic, heterogeneous system without any centralized control.
- Different participants (users and network service providers) have different/conflicting goals.
- A user cares about his individual download, search, or computational job.
- Network provider cares about congestion, 3rd party traffic, revenue, network capacity etc.
- Cooperation not likely, not easy to attempt, not at all enforcible.

TCP: An example

- TCP widely used for Internet communication.
- Breaks messages into packets and reassembles them back at destination. Handles packet delay, out of sync, or losses.
- Congestion control in TCP implements exponential decrease in transmission rate when packet loss is detected.
- But any user can modify the TCP code on his desktop, and replace it with something else!
- For example: double the transmission rate at packet drop!
- So, TCP relies on good citizenship but has no way to enforce it.
- How to design rules of the game so that desired outcome is reached?

Shortest Path Routing

- **A user wants to send data from node x to node y.**
- **Assume links in the network are owned by different agents (network providers).**



- **Links make bids to route the data.**
- **User chooses the x–y shortest path, using these bids as weights. Pays the winning edges.**
- **What can go wrong?**

Selfishness and Speculation

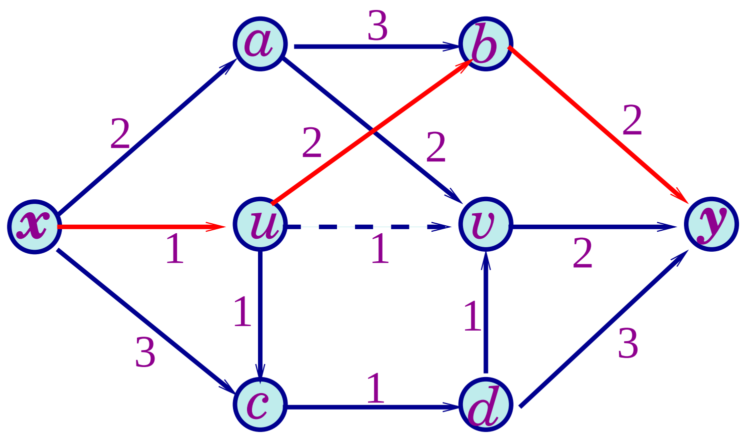
- **Each edge wants to maximize its utility: (payment − cost).**
- **If edges are paid what they bid, they have incentive to lie.**
- **Creates an endless cycles of speculation and counter-speculation for strategic bidders.**
- **Vickrey mechanism is strategy-proof—an agent maximizes his utility by declaring his true cost.**
- **In Vickrey, each winning edge e is given a bonus:**

$$d(x, y; G \setminus e) \; - \; d(x, y).$$

Vickrey Scheme

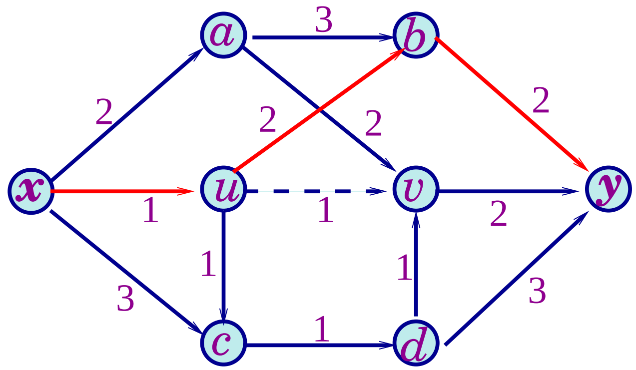
- **Let $c(e)$ be the bid of edge e.**
- **Using $c()$'s as weights, compute shortest path from x to y.**
- **Edges not on the winning path receive no payment.**
- **An edge e on the winning path receives**

$$c(e) + (d(x, y; G \setminus e) - d(x, y))$$



An Algorithmic Problem

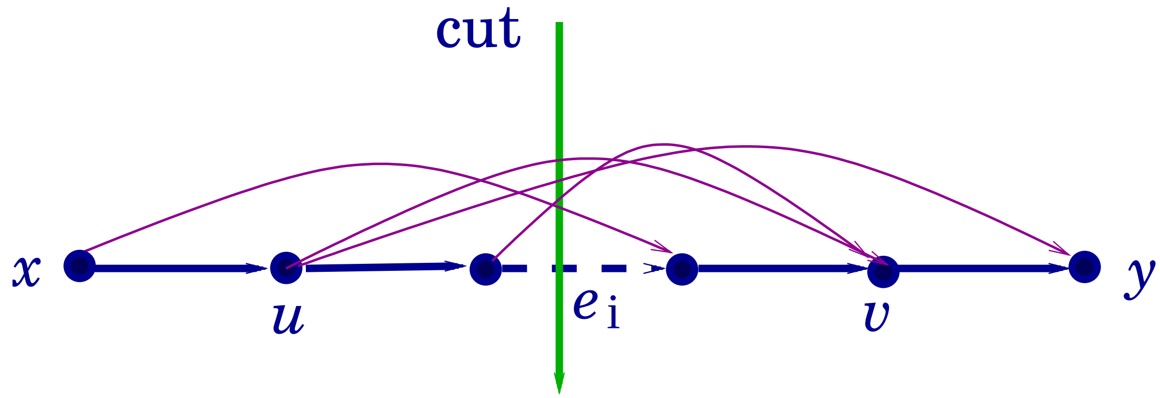
- **How quickly can one compute Vickrey payments for all the edges?**
- **The naive method requires $\Theta(n)$ single source shortest path computations.**



- $O(m + n \log n)$ **total time, for undirected graphs. [HS '01]**
- $\Omega(m\sqrt{n})$ **"lower bound," for directed graphs. [HSB '03]**

A Path Graph

- **Shortest path path**$(x, y) = (e_1, e_2, \ldots, e_k)$ **includes all vertices of** V**.**

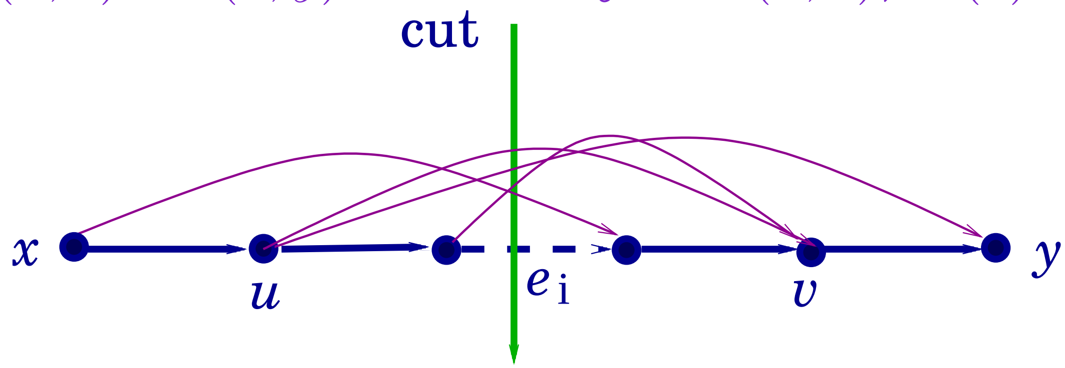


- **Let** $E_i \subset E$ **be edges crossing the cut for** e_i**.**
- **Let** $d_{-i}(x, y) = d(x, y; G \setminus e_i)$**. Then,**

$$d_{-i}(x, y) = \min_{\substack{(u,v) \in E_i \\ (u,v) \neq e_i}} d(x, u) + c(u, v) + d(v, y).$$

Algorithm

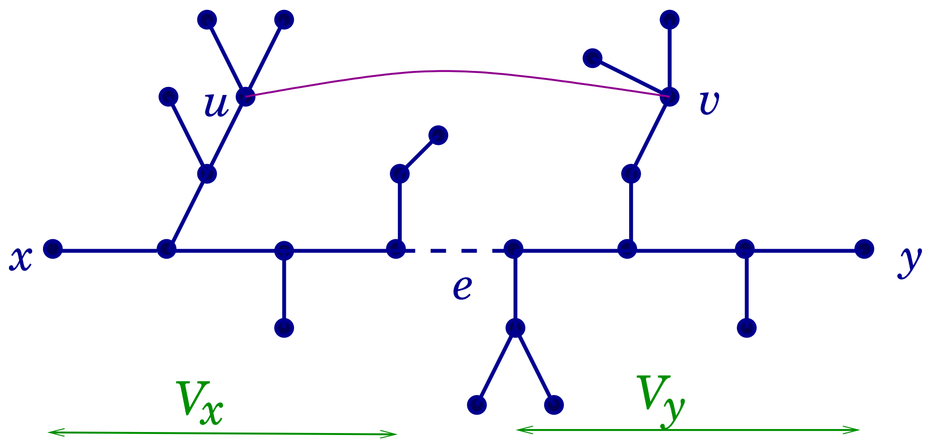
- **Process** $e_1, \ldots, e_k$ **left to right. Heap stores edges of cut** E_i**.**
- $d(x,u) + c(u,v) + d(v,y)$ **is the key for** (u,v)**;** $O(1)$ **time.**



- **Min key in the heap determines** $d_{-i}(x,y)$**.**
- **Moving from** (v_{i-1}, v_i) **to** (v_i, v_{i+1})**, delete from** H **edges terminating at** v_i**, and insert edges starting at** v_i**.**
- **Ordinary heap implementation computes** d_{-i} **for all** i **in total** $O((n+m)\log n)$ **time. Fibonacci Heap for improved bound.**

General Undirected Graphs

- **Focus on SP Tree T_x rooted at x; cuts (V_x, V_y) defined by its partition.**
- **Replacement path still involves a (u, v) jump across the cut.**

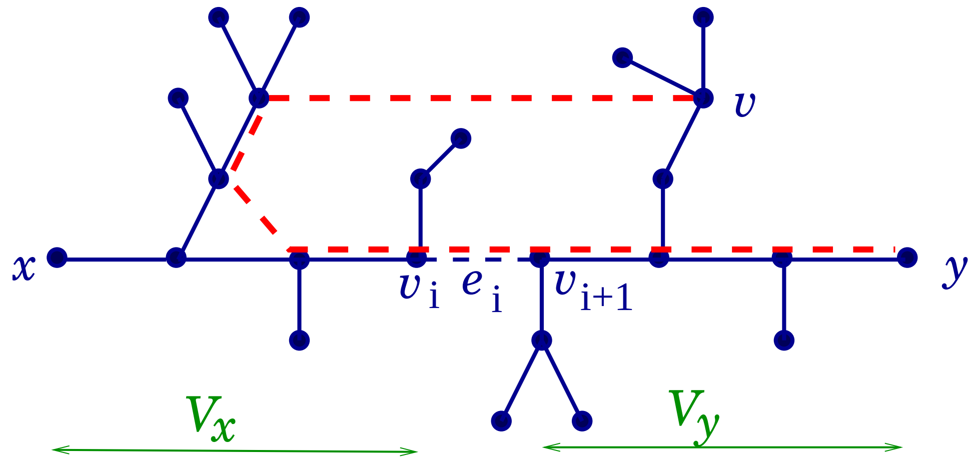


- **Distance formula remains:**

$$d_{-i}(x, y) = \min_{\substack{(u,v) \in E_i \\ (u,v) \neq e_i}} d(x, u) + c(u, v) + d(v, y).$$

Maintaining Tails

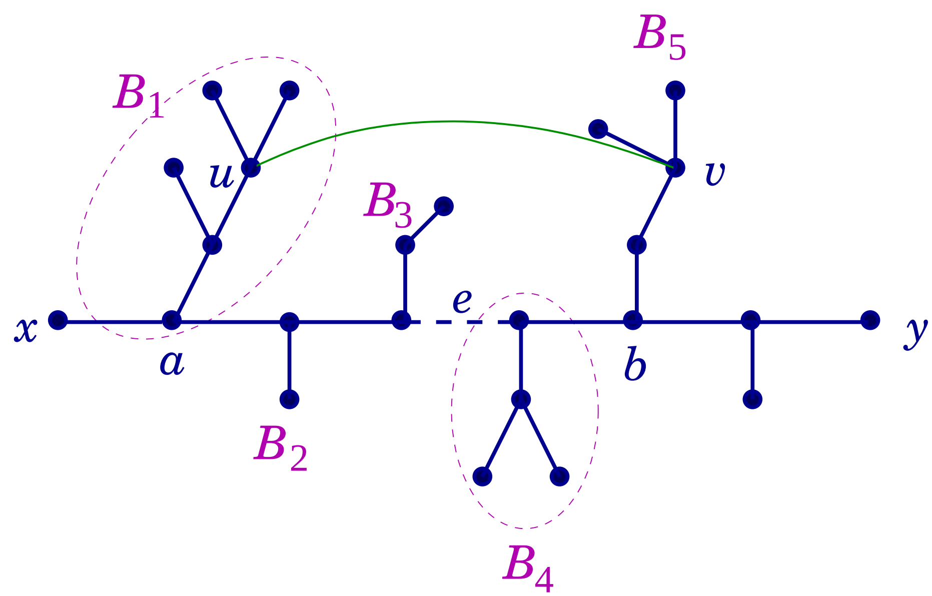
- **Use SP tree into y for distances $d(v, y)$.**
- **Claim: If $v \in V_y$, then $d_G(v, y) = d_{-i}(v, y)$.**



- **If $path(v, y; G \setminus e_i)$ includes e_i, then the red path from v to v_{i+1} is shorter than the blue path from v_{i+1} to v.**
- **But that is impossible. So, $d_G(v, y) = d_{-i}(v, y)$.**

Maintaining Cuts

- **How to identify and maintain cut edges.**
- **Group off-spine nodes in blocks.**



- (u, v) **belongs to all cuts between** $a = block(u)$ **and** $b = block(v)$**.**
- **Cut for** (v_i, v_{i+1}) **has** $V_x = \cup_{j=0}^{i} B_j$**, and** $V_y = \cup_{j=i+1}^{k} B_j$**.**

Problems and Directions

- Similar work on MST, matching, scheduling etc.
- Imagine having to incentivize geometric objects. Which problems fit this bill?
- Some examples: Obstacles paid to move out of the way. Marginal value of a facility.
- Obvious question ones: efficient Vickrey payment. Less obvious ones: What is possible?

Problems and Directions

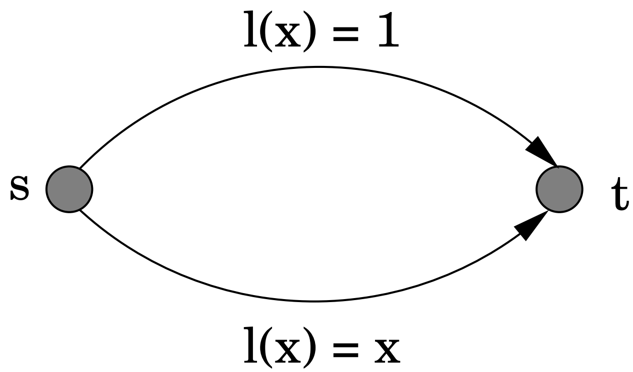
- **Some ad hoc networking applications—node have incentive to lie about their position, their range etc.**
- **Adhoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. Eidenbenz and Anderegg, Mobicom 2003.**
- **On the approximability of range assignment on radio networks in presence of selfish agents. Ambuhl, Clementi, Penna, Ross, Silvestri.**

Nash Equilibria—Price of Anarchy

Optimization without Coordination

- **Users of the Information Superhighway are selfish.**
- **Each wants to minimize his own latency (cost).**
- **Each also causes congestion for other users.**
- **A centralized routing won't work—users on slow links will want to change their routes.**
- **A Nash equilibrium routing is the best one can hope for: no single user is motivated to deviate.**
- **Price of anarchy is the ratio between NE solution and centralized optimum.**

Selfish Routing



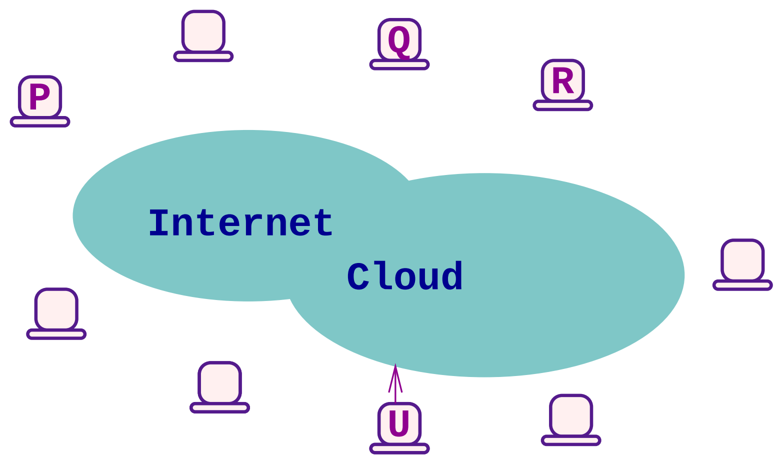
- **Social optimum: $\frac{1}{2}$ traffic on each link. Total latency $\frac{3}{4}$.**
- **Nash has entire traffic on lower link. Total latency 1.**
- **[RT, KP, CV] analyze how bad selfish routing is.**
- **Examples. If the latency function is arbitrary, then price of anarchy is unbounded. With linear latency functions, the worst case ratio is $4/3$.**

Load Balancing

- m **servers or machines.** n **clients or jobs.**
- **Each job can be run only on a subset of servers.**
- **Load balancing problem: find an optimal assignment of jobs to servers.**
- **Clients are selfish and strategic; however, each client's latency depends on other clients' actions.**
- **Anarchic load balancing—no coordinator.**
- **A non-cooperative game among clients—strategies are server choices, payoff is latency.**
- **Nash assignment: no client motivated to switch unilaterally.**
- **Forthcoming by [Kothari, Suri, Toth, Zhou].**

P2P Application

- Decentralized, distributed data-sharing networks.
- Napster, Gnutella, Freenet, CAN, Chord, Pastry, Tapestry, Morpheus, KaZaa, Farsite, Jxta, OceanStore...

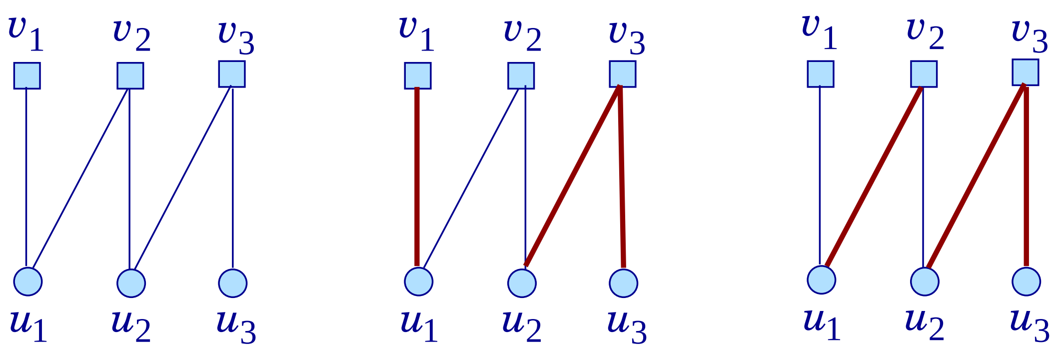


"A Beautiful Mind"

- No central authority: all nodes autonomous and same functionality (democracy of peers).
- Resource sharing by direct exchange between peers.

The Model

- **A bipartite graph between n clients and m servers.**
- **A matching assigns each client to an adjacent server.**



An assignment A Nash assignment

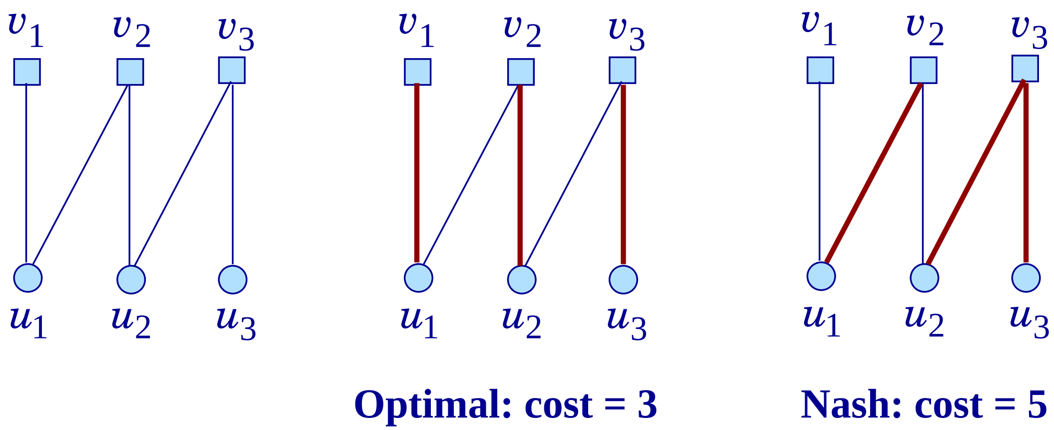
- **Let v_j be the speed of server j. If j is assigned ℓ_j clients, then the latency $\lambda()$ to each client is v_j/ℓ_j.**
- **Cost of a matching** $\text{cost}(M) = \sum_{i=1}^{n} \lambda(u_i)$.

Various Models

1. Atomic Assignment
   - Each client matched to at most 1 server.
   - Server j has speed v_j.
   - If server j has $\deg(j)$ clients, then its load is $\ell_j = \deg(j)$.
   - Each client of server j experiences latency ℓ_j / v_j.
2. Fractional Assignment
   - A client can split its job among multiple servers: x_{ij}.
   - Server j's load is $\ell_j = \sum_i x_{ij}$.
   - Server j completes all its assigned jobs at time ℓ_j.
   - Client i's latency is $\lambda_i = \max_j \{\ell_j \mid x_{ij} > 0\}$.
   - Fractional model similar to one used by KaZaa.
3. L_p norm latency functions.

Optimal vs. Nash

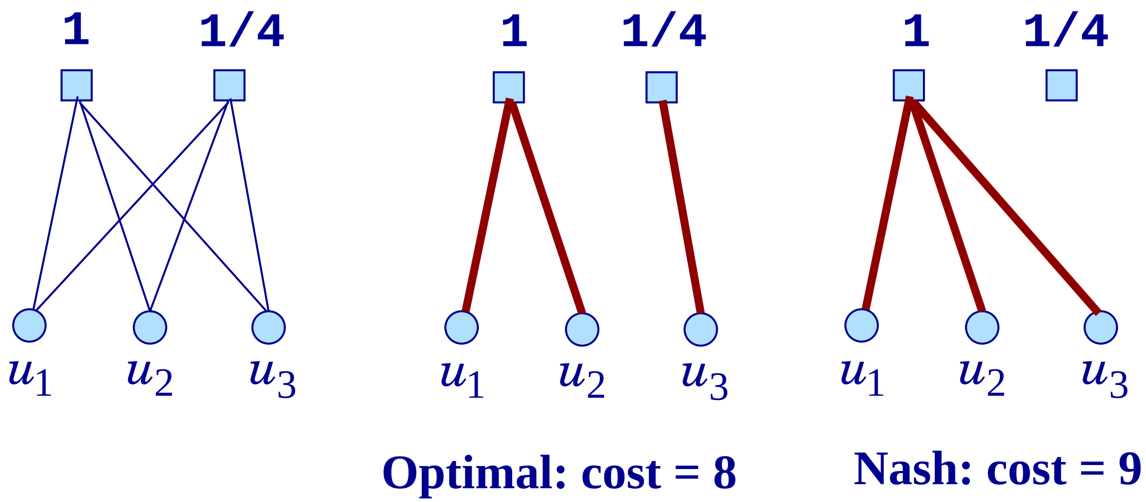
- **Assuming $v_j = 1$, optimal always Nash, but not vice versa.**



- **Proof: Suppose a client can switch from server i to j.**
- $$\begin{aligned} \text{cost}(M') - \text{cost}(M_{\text{opt}}) &= \left((d_j+1)^2 + (d_i-1)^2\right) - (d_i^2 + d_j^2) \\ &= 2(d_j - d_i + 1) < 0. \end{aligned}$$

Optimal vs. Nash

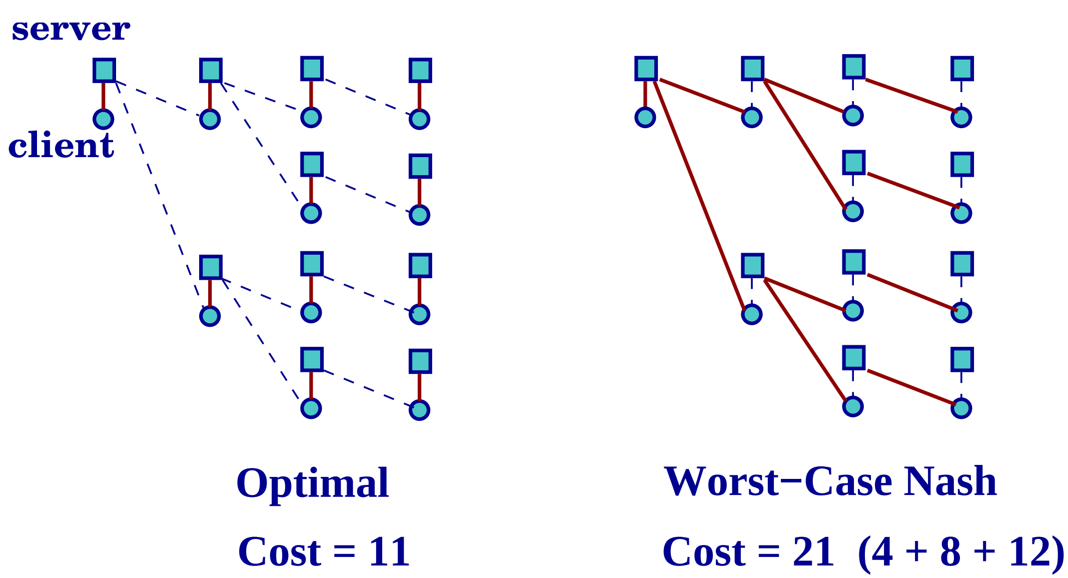
- **But with** **different server speeds**, **optimal is not always Nash.**



- $\mathrm{cost(Opt)} = 2 + 2 + 4; \quad \mathrm{cost(Nash)} = 3 + 3 + 3.$

Lower Bound on Worst-case Nash

- A tree structure: each node has a client-server pair. Edges between neighboring levels only.
- In optimal, each client assigned to its server in the pair, for a total cost of n.
- In Nash, each client matched to a server at higher level. The total cost becomes $2n - 1$.



Upper Bounds: Unit Speeds

- **Theorem:**
  The price of anarchy is at most $1 + 2/\sqrt{3} \approx 2.15$.

- **The price of anarchy has form $1 + 2m/n$, which tends to 1 as the ratio between jobs and servers tends to ∞.**

- **Awerbuch et al. proved that Greedy has competitive ratio** $(1 + \sqrt{2})^2 \approx 5.82$.

- **So, rationality helps!**

- **Same techniques improve greedy's ratio to $2 + \sqrt{5} = 4.236$.**

General Upper Bounds

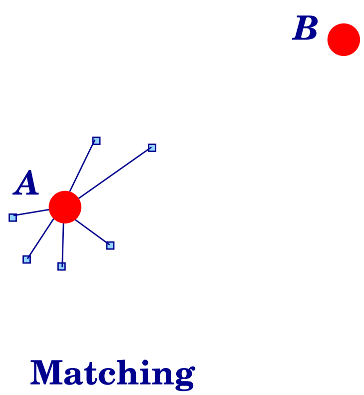
- **Theorem: The price of anarchy is at most $5/2$.**
- **Theorem: With latency measured by L_p norm, the price of anarchy is $\frac{p}{\log p}(1 + o(1))$.**
- **Theorem: With fractional model, Nash is always optimal.**

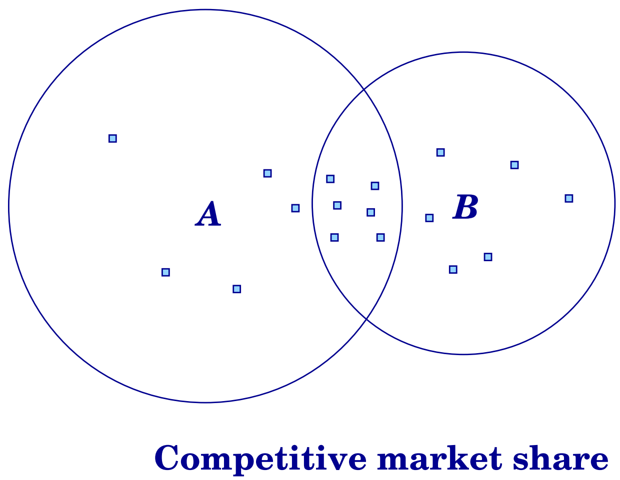
Related work

- **Voronoi Games. Played on continuous plane. Bound the second mover's advantage. [Cheong, Har-Peled, Linial, Matousek], [Fekete-Meijer].**
- **Location Games. Discrete location choices, non-uniform customer distribution. [Chawla, Rajan, Ravi, Sinha].**

Problems and Directions

- **Price of anarchy in geometric matching (client-server assignment)? Cost $\propto$ to distance, load. Prefer close servers, but switch if load too high.**





- **Competitive market share. Cost and benefit $\propto$ radius (advertisement budget, customers). How bad is a NE solution compared to centralized optimum?**