Quantifying Trade-Offs in Networks and Auctions

Tim Roughgarden
Stanford CS
Trade-Offs and Pareto Curves

Well known: trade-offs between competing objectives inevitable in algorithm/system design.

Pareto curve: depicts what's feasible and what's not.

Vilfredo Federico Damaso Pareto (1848-1923)
Quantifying Trade-Offs

Goal: quantitative analysis of Pareto frontier.

Motivation: qualitative insights.
- infer design principles
- identify fundamental trade-offs between competing constraints/objectives
- "litmus test" for algorithm/system design
  - identify solutions spanning the Pareto frontier
Game-Theoretic Applications

New Challenge: algorithm/system design under game-theoretic constraints.

Difficulty #1: system (partially) controlled by independent, self-interested entities.
- network examples: routing paths, rate of traffic injection, topology formation

Difficulty #2: data (partially) controlled by independent, self-interested entities.
- auctions: participants' values for goods
Application #1: Routing

The problem: given a network and a traffic matrix (pairwise traffic rates), select "optimal" routing paths for traffic.
Application #1: Routing

The problem: given a network and a traffic matrix (pairwise traffic rates), select "optimal" routing paths for traffic.

Assumptions: fixed traffic matrix;
• traffic = many small flows
• goal: minimize average delay
• link delay = fixed propagation delay + congestion-dependent queuing delay
Measuring Routing Quality

Holy grail: centralized routing = globally compute optimal routes for all traffic.

Distributed routing: scalable, but larger delay.

Assume: at steady-state, only routes of minimum-delay are in use.
  - delay-minimizing end users ("selfish routing")
  - delay-minimizing routing policies

Next: distributed routing vs. holy grail.
A Bad Example

**Example:** large prop delay + large capacity vs. small prop delay + small capacity
- one unit (comprising many flows) of traffic

\[ d(x) \approx x^{100} \]
\[ d(x) \approx 1 \]

Diagram:
- **s** to **t**: Delay [secs]
- **Rate x**
A Bad Example

Example: large prop delay + large capacity vs. small prop delay + small capacity
- one unit (comprising many flows) of traffic

\[ d(x) \approx x^{100} \]

\[ d(x) \approx 1 \]

"selfish" routing

![Diagram showing delay and rate]
A Bad Example

Example: large prop delay + large capacity vs. small prop delay + small capacity
- one unit (comprising many flows) of traffic

\[ d(x) \approx x^{100} \]

"selfish" routing
0
"optimal" routing
\[ d(x) \approx 1 \]

Delay [secs]

Rate x
A Bad Example

Example: large prop delay + large capacity vs. small prop delay + small capacity
- one unit (comprising many flows) of traffic

Hope: distributed routing improves in overprovisioned network.
Routing: The Pareto Frontier

Trade-off: performance of distributed routing vs. degree of overprovisioning.

- "performance" = ratio between average delay of distributed, optimal routing
  - "Price of Anarchy"
- "overprovisioning" = fraction of capacity unused at steady-state

Goal: quantify Pareto curve to extract general design principles.
Benefit of Overprovisioning

Suppose: network is overprovisioned by $\beta > 0$ ($\beta$ fraction of each edge unused).

Then: Delay of selfish routing at most $\frac{1}{2}(1+1/\sqrt{\beta})$ times that of optimal.

- arbitrary network size/topology, traffic matrix

Moral: Even modest (10%) over-provisioning sufficient for near-optimal routing.
Main Theorem (Informal)

Thm: [Roughgarden 02] 2-node, 2-link networks always exhibit worst-possible performance.

• quantitatively: can compute worst-case performance for given amount of overprovisioning
• qualitatively: performance of distributed routing doesn’t degrade with complexity of network, traffic

---

• complex network + traffic matrix

---

worst case
Application #2: Search Auctions
The Model

• n advertisers, k slots

• advertisers have willingness to pay per click

• slot j has "click-through rate" $\Theta_j$
  - top slots are better: $\Theta_j \geq \Theta_{j+1}$ for all j

• advertisers are ranked by bid
  - possibly with a "reserve price"
  - for simplicity, ignore issue of "ad relevance"

• payments inspired by "second-price auction"
What Is the Objective?

Objective #1: revenue
- [Myerson 81]: Suppose willingness to pay ("valuations") drawn i.i.d. from some distribution.
- Choosing suitable reserve price (+ second-price-esque payments) maximizes expected revenue.

Objective #2: "economic efficiency"
- [Vickrey 61/Clarke 71/Groves 73]: suitable second-price-esque payments (no reserve price) works.
- no assumptions on valuations
Search Auctions: The Pareto Frontier

**Trade-off:** economic efficiency vs. revenue.

**Goal:** quantify revenue loss in efficient auction.

**Motivation:**
- efficient auction closer to those used in practice
- does not require assumption and knowledge of distribution
- good in non-monopoly settings
Competition Obviates Cleverness

Theorem #1 [Roughgarden/Sundararajan 07]:
Expected revenue of the economically efficient auction with \( n+k \) bidders exceeds optimal expected revenue with \( n \) bidders.
- for any fixed distribution (some technical conditions)

Point: a little extra competition outweighs benefit of the optimal reserve price.
Competition Obviates Cleverness

Theorem #2 [Roughgarden/Sundararajan 07]:
Expected revenue of economically efficient auction is at least \((1-k/n)\) times optimal expected revenue.

- \(n\) advertisers in both auctions
- for any fixed distribution (some technical conditions)

Point: under modest competition, efficient auction has near-optimal revenue.
What's Next?

**Networks:** use Pareto frontier to inform algorithm/protocol design
- different curves for different protocols
- want "best" Pareto curve
- see [Chen/Roughgarden/Valiant 07]

**Sponsored Search:** need dynamic models
- competition between search engines, etc.
- short-term vs. long-term revenue trade-offs
- role of budget constraints
- vindictive bidding