Kapitel 5:
Local Search

Inhalt:

- Gradient Descent (Hill Climbing)
- Metropolis Algorithm and Simulated Annealing
- Local Search in Hopfield Neural Networks
- Local Search for Max-Cut
  - Single-flip neighborhood
  - K-flip neighborhood
  - KL-neighborhood
- Nash Equilibria

Finding a Nash Equilibrium

Theorem. The following algorithm terminates with a Nash equilibrium.

```
Best-Response-Dynamics(G, c) {
   Pick a path for each agent

   while (not a Nash equilibrium) {
      Pick an agent i who can improve by switching paths
      Switch path of agent i
   }
}
```

Pf. Consider a set of paths $P_1, \ldots, P_k$.

- Let x_e denote the number of paths that use edge e.
- Let $\Phi(P_1, \ldots, P_k) = \sum_{e \in E} c_e \cdot H(x_e)$ be a potential function.
- Since there are only finitely many sets of paths, it suffices to show that Φ strictly decreases in each step.

$H(0) = 0, \quad H(k) = \sum_{i=1}^{k} \frac{1}{i}$

Finding a Nash Equilibrium

Pf. (continued)

- Consider agent j switching from path P_j to path P_j'.
- Agent j switches because

$$\underbrace{\sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1}}_{\text{newly incurred cost}} < \underbrace{\sum_{e \in P_j - P_j'} \frac{c_e}{x_e}}_{\text{cost saved}}$$

- Φ increases by $\displaystyle \sum_{f \in P_j' - P_j} c_f \left[H(x_f + 1) - H(x_f) \right] = \sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1}$
- Φ decreases by $\displaystyle \sum_{e \in P_j - P_j'} c_e \left[H(x_e) - H(x_e - 1) \right] = \sum_{e \in P_j - P_j'} \frac{c_e}{x_e}$
- Thus, net change in Φ is negative. ▪

Bounding the Price of Stability

Claim. Let $C(P_1, \ldots, P_k)$ denote the total cost of selecting paths $P_1, \ldots, P_k$. For any set of paths $P_1, \ldots, P_k$, we have

$$C(P_1, \ldots, P_k) \;\le\; \Phi(P_1, \ldots, P_k) \;\le\; H(k) \cdot C(P_1, \ldots, P_k)$$

Pf. Let x_e denote the number of paths containing edge e.

- Let E^+ denote set of edges that belong to at least one of the paths $P_1, \ldots, P_k$.

$$C(P_1, \ldots, P_k) \;=\; \sum_{e \in E^+} c_e \;\le\; \underbrace{\sum_{e \in E^+} c_e\, H(x_e)}_{\Phi(P_1, \ldots, P_k)} \;\le\; \sum_{e \in E^+} c_e\, H(k) \;=\; H(k)\; C(P_1, \ldots, P_k)$$

Bounding the Price of Stability

Theorem. There is a Nash equilibrium for which the total cost to all agents exceeds that of the social optimum by at most a factor of $H(k)$.

Pf.

- Let $(P_1^*, \ldots, P_k^*)$ denote set of socially optimal paths.
- Run best-response dynamics algorithm starting from P^*.
- Since Φ is monotone decreasing $\Phi(P_1, \ldots, P_k) \leq \Phi(P_1^*, \ldots, P_k^*)$.

$$C(P_1, \ldots, P_k) \underset{\substack{\uparrow \\ \text{previous claim} \\ \text{applied to } P}}{\leq} \Phi(P_1, \ldots, P_k) \leq \Phi(P_1^*, \ldots, P_k^*) \underset{\substack{\uparrow \\ \text{previous claim} \\ \text{applied to } P^*}}{\leq} H(k) \cdot C(P_1^*, \ldots, P_k^*)$$

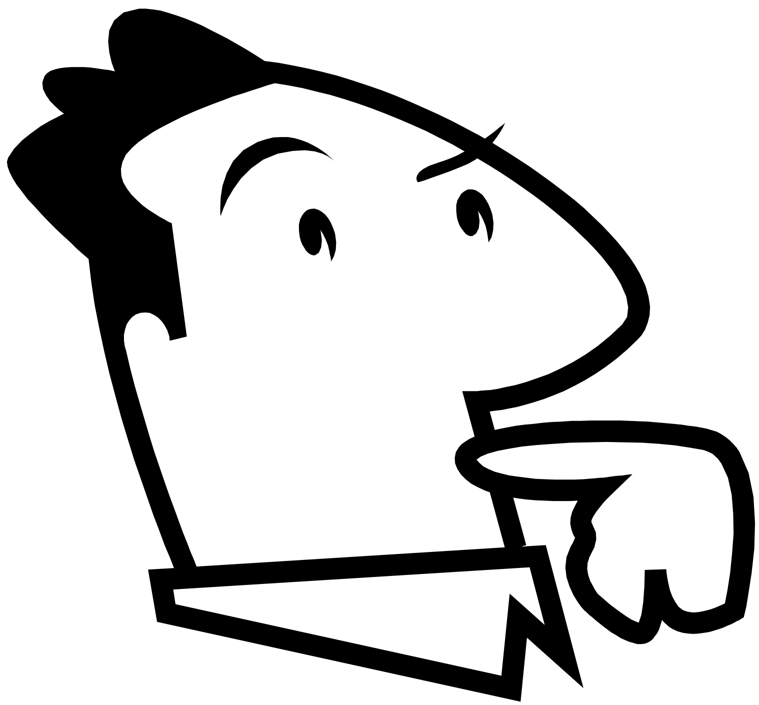
Summary

Existence. Nash equilibria always exist for k-agent multicast routing with fair sharing.

Price of stability. Best Nash equilibrium is never more than a factor of H(k) worse than the social optimum.

Fundamental open problem.
(1) Find any Nash equilibria in poly-time.
(2) Find efficiently the Nash equilibria that achieve the bound H(k).

Fragen?



Kapitel 6:
Randomized Algorithms

Randomization

Algorithmic design patterns.

- Greedy.
- Divide-and-conquer.
- Dynamic programming.
- Approximation.
- Local Search.
- Randomization.

in practice, access to a pseudo-random number generator

Randomization. Allow fair coin flip in unit time.

Why randomize? Can lead to simplest, fastest, or only known algorithm for a particular problem.

Ex. Symmetry breaking protocols, graph algorithms, quicksort, hashing, load balancing, Monte Carlo integration, cryptography.

Kapitel 6:
Randomized Algorithms

Inhalt:

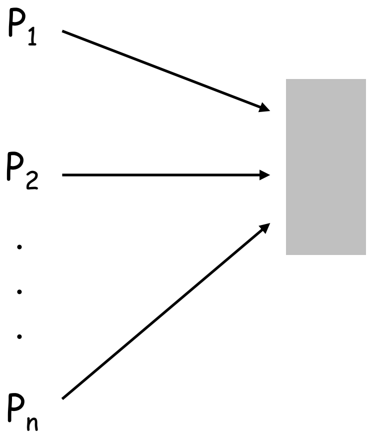
- Contention Resolution (symmetry-breaking)
- Global Minimum Cut (contraction algorithm)
- Random Variables and their Expectations
  - Guessing Cards
  - Coupon Collector
- Max 3-SAT

Contention Resolution in a Distributed System

Contention resolution. Given n processes $P_1, \ldots, P_n$, each competing for access to a shared database. If two or more processes access the database simultaneously, all processes are locked out. Devise protocol to ensure all processes get through on a regular basis.

Restriction. Processes can't communicate.

Challenge. Need symmetry-breaking paradigm.



Contention Resolution: Randomized Protocol

Protocol. Each process requests access to the database at time t with probability $p = 1/n$ independently of the other processes.

Claim. Let $S[i, t]$ = event that process i succeeds in accessing the database at time t. Then $1/(e \cdot n) \le \Pr[S(i, t)] \le 1/(2n)$.

Pf. By independence, $\Pr[S(i, t)] = p\,(1-p)^{n-1}$.

process i requests access

none of remaining n-1 processes request access

- Setting $p = 1/n$, we have $\Pr[S(i, t)] = 1/n\,\underbrace{(1 - 1/n)^{n-1}}_{\text{between } 1/e \text{ and } 1/2}$. ▪

value that maximizes $\Pr[S(i, t)]$

Useful facts from calculus. As n increases from 2, the function:

- $(1 - 1/n)^n$ converges monotonically from 1/4 up to 1/e
- $(1 - 1/n)^{n-1}$ converges monotonically from 1/2 down to 1/e.

Contention Resolution: Randomized Protocol

Claim. The probability that process i fails to access the database in en rounds is at most 1/e. After e·n(c ln n) rounds, the probability is at most n^{-c}.

Pf. Let F[i, t] = event that process i fails to access database in rounds 1 through t. By independence and previous claim, we have
Pr[F(i, t)] $\leq$ (1 - 1/(en)) ^{t}.

- Choose t = $\lceil e \cdot n \rceil$: $$\Pr[F(i,t)] \leq \left(1 - \tfrac{1}{en}\right)^{\lceil en \rceil} \leq \left(1 - \tfrac{1}{en}\right)^{en} \leq \tfrac{1}{e}$$
- Choose t = $\lceil e \cdot n \rceil \lceil c \ln n \rceil$: $$\Pr[F(i,t)] \leq \left(\tfrac{1}{e}\right)^{c \ln n} = n^{-c}$$

Contention Resolution: Randomized Protocol

Claim. The probability that all processes succeed within $2e \cdot n \ln n$ rounds is at least 1 - 1/n.

Pf. Let F[t] = event that at least one of the n processes fails to access database in any of the rounds 1 through t.

$$\Pr\left[F[t]\right] = \Pr\left[\bigcup_{i=1}^{n} F[i,t]\right] \le \sum_{i=1}^{n} \Pr[F[i,t]] \le n\left(1 - \tfrac{1}{en}\right)^{t}$$

union bound

previous slide

- Choosing $t = \lceil en \rceil \lceil 2 \ln n \rceil$ yields $\Pr[F[t]] \le n \cdot n^{-2} = 1/n$. ▪

Union bound. Given events $E_1, \ldots, E_n$,

$$\Pr\left[\bigcup_{i=1}^{n} E_i\right] \le \sum_{i=1}^{n} \Pr[E_i]$$