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

Maximum Cut

Maximum cut. Given an undirected graph G = (V, E) with positive integer edge weights w_e, find a node partition (A, B) such that the total weight of edges crossing the cut is maximized.

$$w(A,B) := \sum_{u \in A,\, v \in B,\, e=(u,v)} w_{uv}$$

Toy application.

- n activities, m people.
- Each person wants to participate in two of the activities.
- Schedule each activity in the morning or afternoon to maximize number of people that can enjoy both activities.

Real applications. Circuit layout, statistical physics.

Maximum Cut

Single-flip neighborhood. Given a partition (A, B), move one node from A to B, or one from B to A if it improves the solution.

Greedy algorithm.

```
Max-Cut-Local (G, w) {
   Pick a random node partition (A, B)

   while (∃ improving node v) {
      if (v is in A) move v to B
      else           move v to A
   }
   return (A, B)
}
```

Maximum Cut: Local Search Analysis

Theorem. Let (A, B) be a locally optimal partition and let (A*, B*) be a global optimal partition. Then $w(A, B) \geq \frac{1}{2} \sum_e w_e \geq \frac{1}{2} w(A^*, B^*)$.

weights are nonegative

Pf.

- Local optimality implies that for all $u \in A$: $\sum_{v \in A} w_{uv} \leq \sum_{v \in B} w_{uv}$
  Adding up all these inequalities yields:

$$2 \sum_{\{u,v\} \subseteq A} w_{uv} \leq \sum_{u \in A,\, v \in B} w_{uv} = w(A,B)$$

- Similarly

$$2 \sum_{\{u,v\} \subseteq B} w_{uv} \leq \sum_{u \in B,\, v \in A} w_{uv} = w(A,B)$$

- Now,

each edge counted once

$$\sum_{e \in E} w_e = \underbrace{\sum_{\{u,v\} \subseteq A} w_{uv}}_{\leq \frac{1}{2} w(A,B)} + \underbrace{\sum_{u \in A,\, v \in B} w_{uv}}_{w(A,B)} + \underbrace{\sum_{\{u,v\} \subseteq B} w_{uv}}_{\leq \frac{1}{2} w(A,B)} \leq 2 w(A,B)$$

▪

Maximum Cut: Big Improvement Flips

Local search. Within a factor of 2 for MAX-CUT, but not poly-time!

Big-improvement-flip algorithm. Only choose a node which, when flipped, increases the cut value by at least $\frac{2\varepsilon}{n} w(A, B)$

Claim. Upon termination, big-improvement-flip algorithm returns a cut (A, B) with $(2 + \varepsilon)\, w(A, B) \geq w(A^*, B^*)$.

Pf idea. Add $\frac{2\varepsilon}{n} w(A, B)$ to each inequality in original proof.

Claim. Big-improvement-flip algorithm terminates after $O(\varepsilon^{-1} n \log W)$ flips, where $W = \Sigma_e w_e$.

- Each flip improves cut value by at least a factor of $(1 + \varepsilon/n)$.
- After n/ε iterations the cut value improves by a factor of 2.
- Cut value can be doubled at most log W times.

↑ if $x \geq 1$, $(1 + 1/x)^x \geq 2$

Maximum Cut: Context

Theorem. [Sahni-Gonzales 1976] There exists a ½-approximation algorithm for MAX-CUT.

Theorem. [Goemans-Williamson 1995] There exists an 0.878567-approximation algorithm for MAX-CUT.

↑

$$\min_{0 \le \theta \le \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta}$$

Theorem. [Håstad 1997] Unless P = NP, no 16/17 approximation algorithm for MAX-CUT.

↑

0.941176

Neighbor Relations for Max Cut

1-flip neighborhood. (A, B) and (A', B') differ in exactly one node.

k-flip neighborhood. (A, B) and (A', B') differ in at most k nodes.

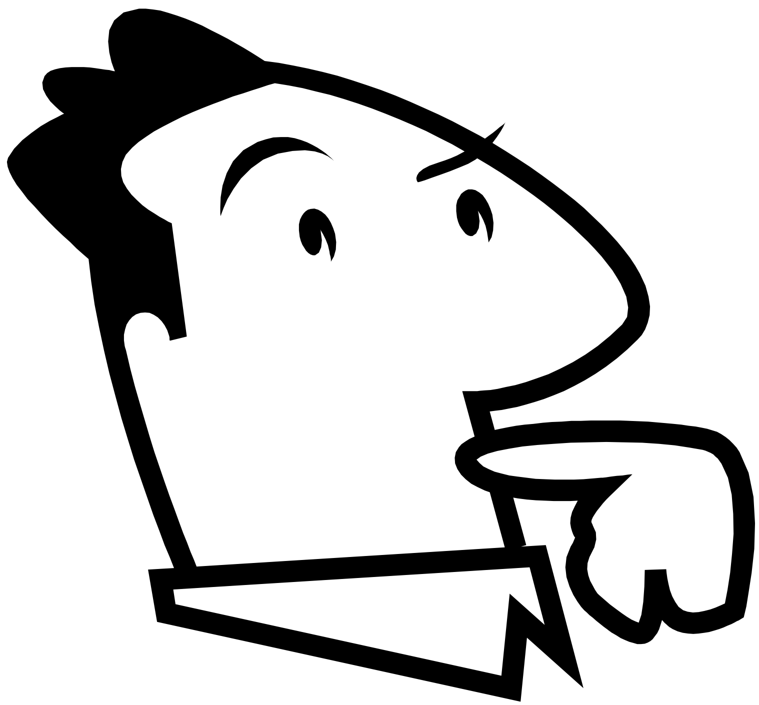
- $\Theta(n^k)$ neighbors.

KL-neighborhood. [Kernighan-Lin 1970]

cut value of (A_1, B_1) may be worse than (A, B)

- To form neighborhood of (A, B):
  - Iteration 1: flip node from (A, B) that results in best cut value (A_1, B_1), and mark that node.
  - Iteration i: flip node from (A_{i-1}, B_{i-1}) that results in best cut value (A_i, B_i) among all nodes not yet marked.
- Neighborhood of (A, B) = $\{(A_1, B_1), \ldots, (A_{n-1}, B_{n-1})\}$.
- Neighborhood includes some very long sequences of flips, but without the computational overhead of a k-flip neighborhood.
- Practice: powerful and useful framework.
- Theory: explain and understand its success in practice.

Fragen?



Kapitel 5:
Local Search

Inhalt:

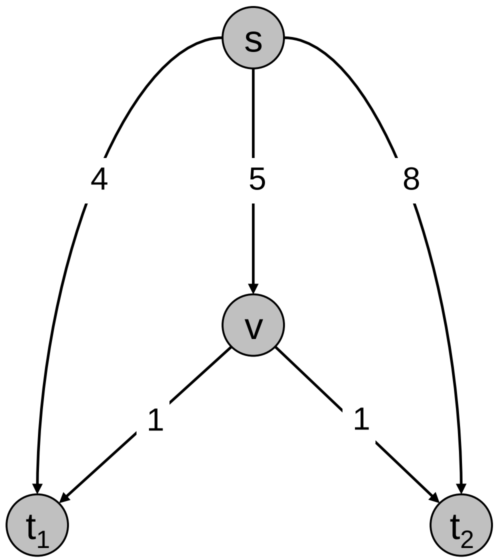
- Gradient Descent (Hill Climbing)
- Metropolis Algorithm and Simulated Annealing
- Local Search in Hopfield Neural Networks
- Local Search for Max-Cut
  - Single-flip neighborhood
  - K-flip neighborhood
  - KL-neighborhood
- Nash Equilibria

Multicast Routing

Multicast routing. Given a directed graph $G = (V, E)$ with edge costs $c_e \geq 0$, a source node s, and k agents located at terminal nodes $t_1, \ldots, t_k$. Agent j must construct a path P_j from node s to its terminal t_j.

Fair share. If x agents use edge e, they each pay c_e / x.

1	2	1 pays	2 pays
outer	outer	4	8
outer	middle	4	5 + 1
middle	outer	5 + 1	8
middle	middle	5/2 + 1	5/2 + 1



Nash Equilibrium

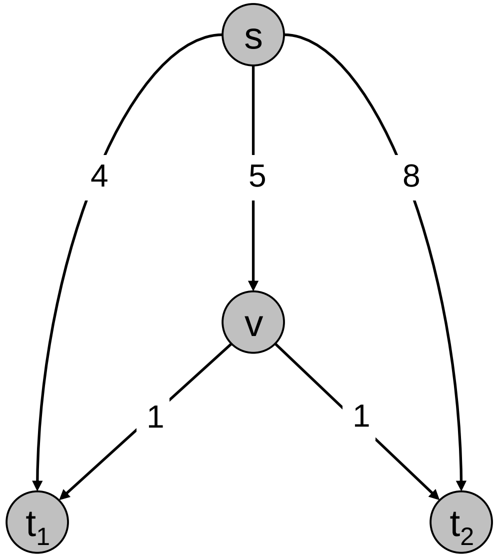
Best response dynamics. Each agent is continually prepared to improve its solution in response to changes made by other agents.

Nash equilibrium. Solution where no agent has an incentive to switch.

Fundamental question. When do Nash equilibria exist?

Ex:

- Two agents start with outer paths.
- Agent 1 has no incentive to switch paths (since $4 < 5 + 1$), but agent 2 does (since $8 > 5 + 1$).
- Once this happens, agent 1 prefers middle path (since $4 > 5/2 + 1$).
- Both agents using middle path is a Nash equilibrium.



Nash Equilibrium and Local Search

Relationship to Local search algorithm. Each agent is continually prepared to improve its solution in response to changes made by other agents.

Analogies.

- Nash equilibrium : local minima/maxima.
- Best Nash equilibrium : global minima/maxima.
- Best response dynamics : local search algorithm.
- Unilateral move by single agent : local neighborhood.

Contrast. Best-response dynamics need not terminate since no single objective function is being optimized.

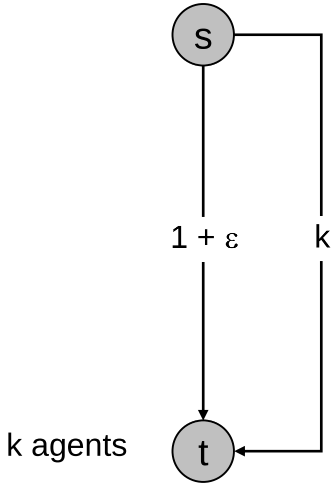
Socially Optimum

Social optimum. Minimizes total cost to all agent.

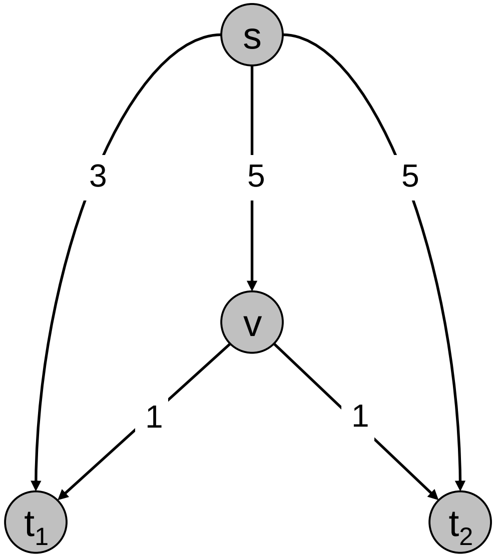
Observation. In general, there can be many Nash equilibria. Even when its unique, it does not necessarily equal the social optimum.

Examples:

"Braess-Paradoxon"



Social optimum = 1 + ε
Nash equilibrium A = 1 + ε
Nash equilibrium B = k



Social optimum = 7
Unique Nash equilibrium = 8

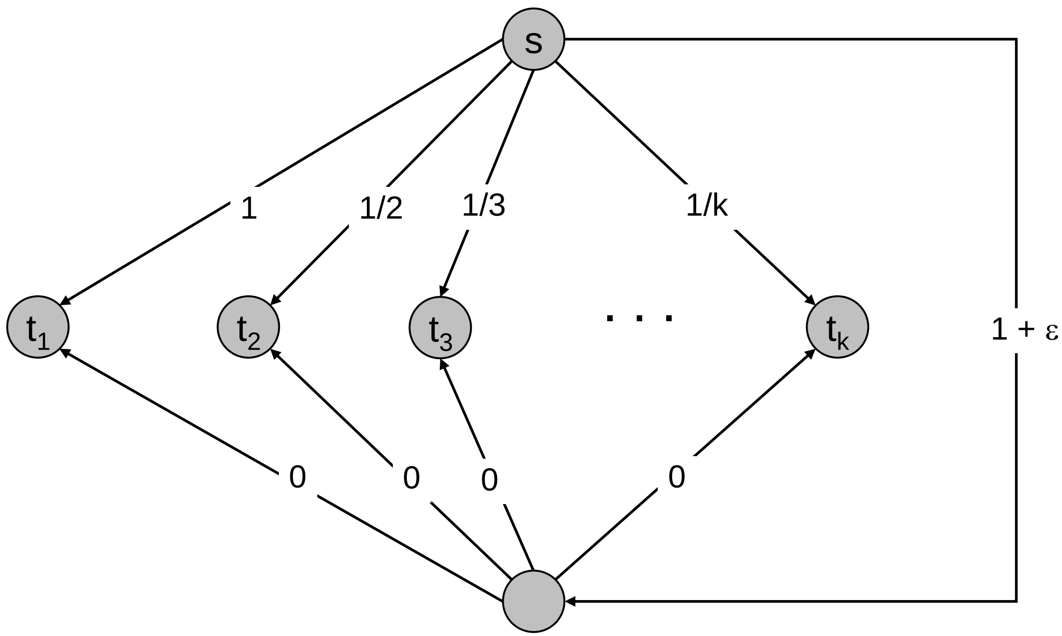
Price of Stability

Price of stability. Ratio of best Nash equilibrium to social optimum.

Fundamental question. What is the price of stability?

Ex: Price of stability = $\Theta(\log k)$.
Social optimum. Everyone takes bottom paths.
Unique Nash equilibrium. Everyone takes top paths.
Price of stability. $H(k) / (1 + \varepsilon)$.

↑ $1 + 1/2 + \ldots + 1/k$



Fragen?