上海科技大学
ShanghaiTech University

CS240 Algorithm Design and Analysis

Lecture 15

Local Search

Quan Li
Fall 2022
2023.11.30

Local Search

Gradient descent

Maximum Cut

Nash Equilibria

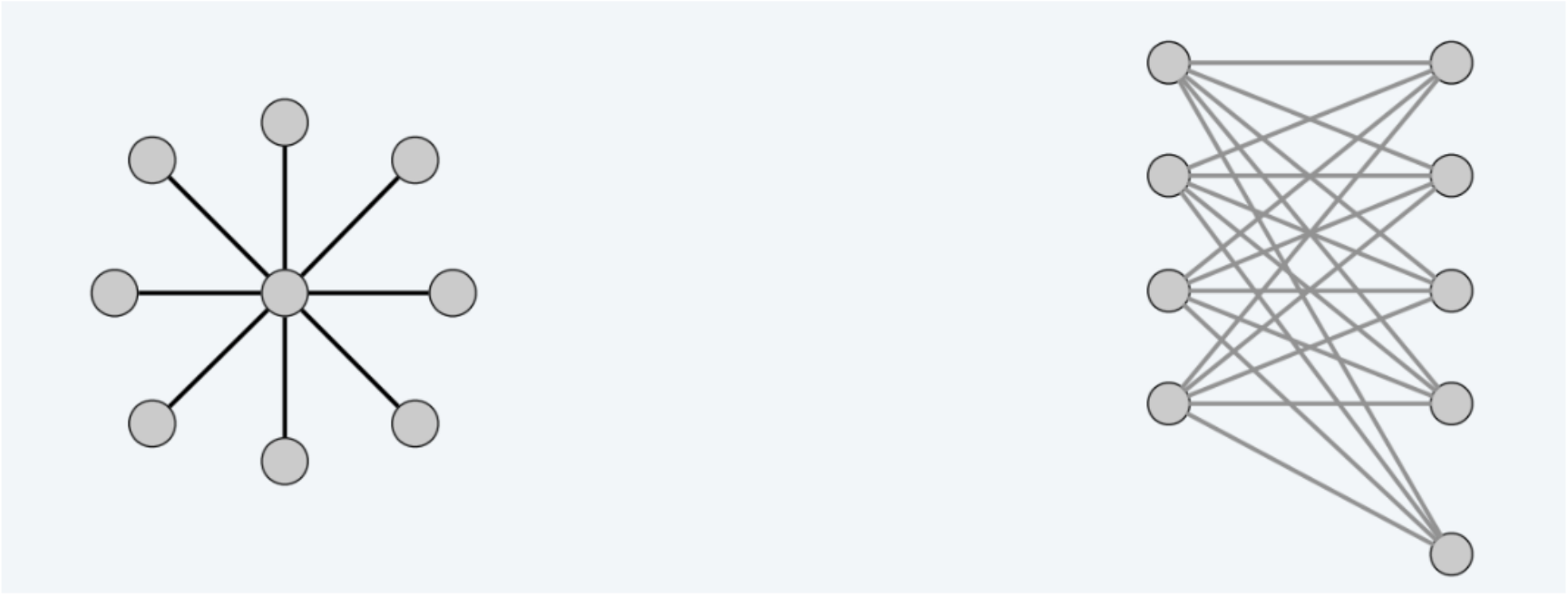
Local Search: Gradient Descent

Gradient Descent: Vertex Cover

- **Vertex cover.** Given a graph $G = (V, E)$, find a subset of nodes S of minimal cardinality such that for each $(u, v) \in E$, either u or v (or both) are in S
- **Neighbor relation.** $S \sim S'$ if S' can be obtained from S by adding or deleting a single node. Each vertex cover S has at most n neighbors
- **Gradient descent.** Start with $S = V$. If there is a neighbor S' that is a vertex cover and has lower cardinality, replace S with S'
- **Remark.** Algorithm terminates after at most n steps since each update decreases the size of the cover by one

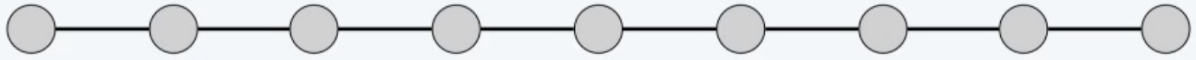
Gradient Descent: Vertex Cover

- **Local optimum.** No neighbor is strictly better



Optimum = center node only
Local optimum = all other nodes

Optimum = all nodes on left side
Local optimum = all nodes on right side



optimum = even nodes

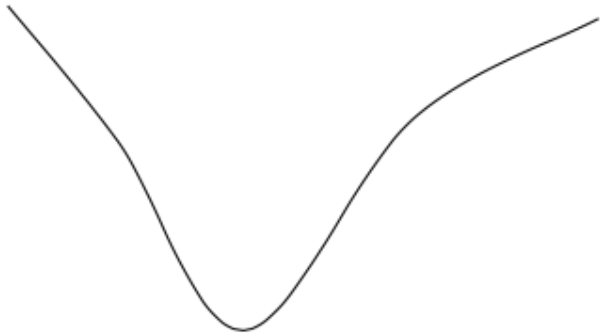
local optimum = omit every third node

Local Search

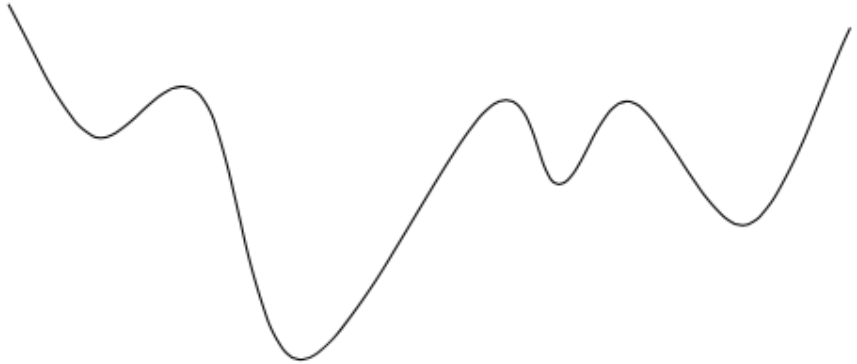
Local search. Algorithm that explores the space of possible solutions in sequential fashion, moving from a current solution to a "nearby" one.

Neighbor relation. Let S ~ S' be a neighbor relation for the problem.

Gradient descent. Let S denote current solution. If there is a neighbor S' of S with strictly lower cost, replace S with the neighbor whose cost is as small as possible. Otherwise, terminate the algorithm.



A funnel



A jagged funnel

Local Search: Maximum Cut

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.

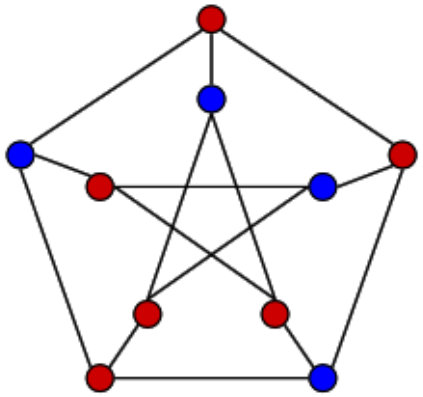
$$w(A,B) := \sum_{u \in A,\, v \in B} w_{uv}$$

Note:

- The Min-Cut problem can be solved in poly time.
- The Max-Cut problem is NP-hard.

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.

Local search 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 the optimal partition. Then $w(A, B) \geq \frac{1}{2} \sum_e w_e \geq \frac{1}{2} w(A^*, B^*)$.

Weights are nonnegative

Pf.

- Local optimality implies that for any $u \in A$: $\sum_{v \in A} w_{uv} \leq \sum_{v \in B} w_{uv}$
  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

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

- Now,

each edge counted once

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

.

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 $\log_2 W$ times.

if $x >= 1$, $(1+1/x)^x >= 2$

Nash Equilibria

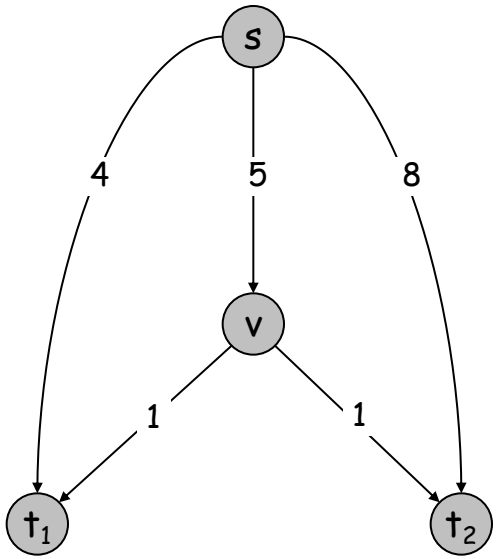
Multicast Routing with Fair Cost Sharing

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



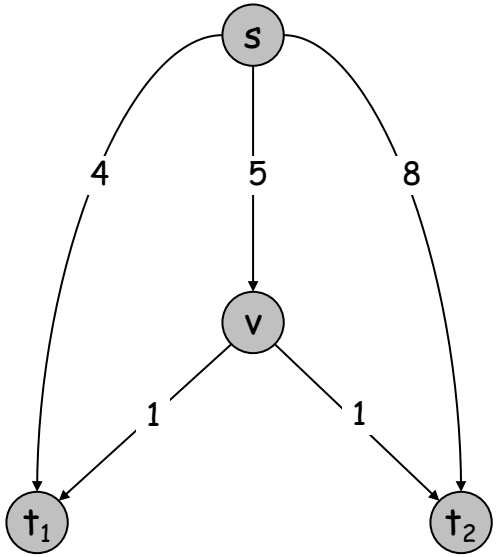
Nash Equilibrium

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.

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.

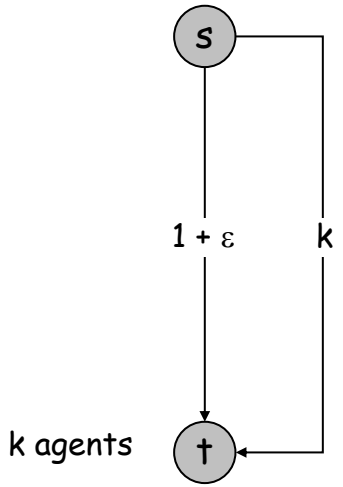


Note. Best response dynamics may not terminate since no single objective function is being optimized.

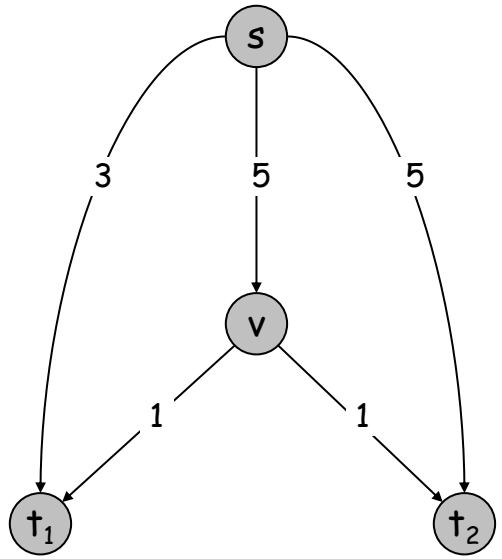
Social Optimum

Social optimum. Minimizes total cost to all agents.

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



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



Social optimum = 7
Unique Nash equilibrium = 8

Price of Stability

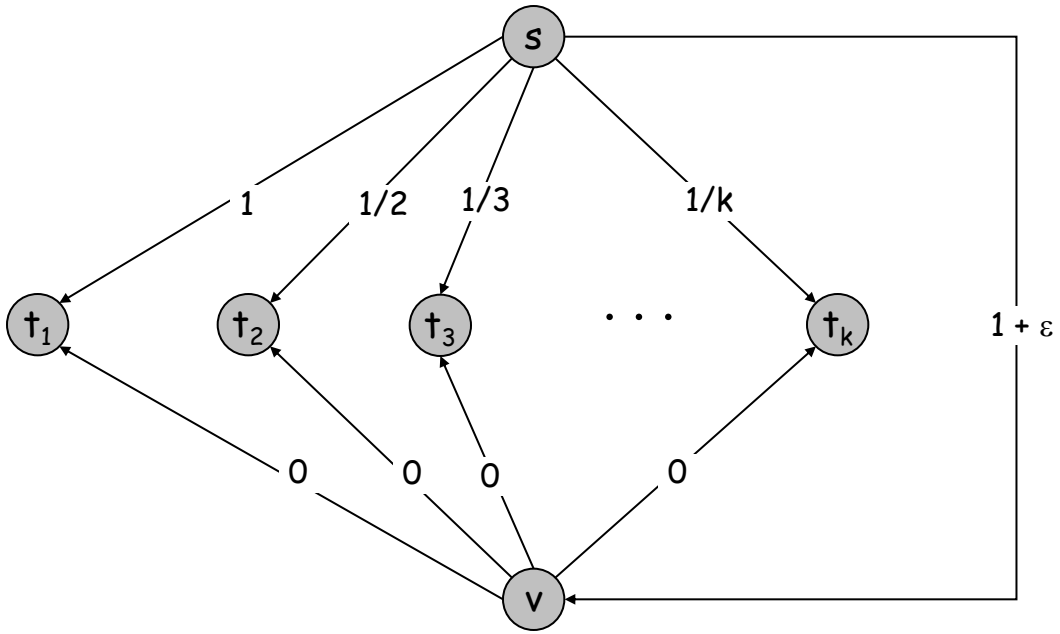
Price of stability. Ratio of best Nash equilibrium to social optimum.

Fundamental question. What is 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:** $H(k) / (1 + \varepsilon)$.

$1 + 1/2 + \ldots + 1/k$



Finding a Nash Equilibrium

Theorem. The following algorithm terminates with a Nash equilibrium (but its running time may be exponential).

```
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.

$H(0) = 0$

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

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

$$\sum_{f \in P_j' - P_j} c_f \left[H(x_f + 1) - H(x_f) \right] = \sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1}$$

- Φ decreases by

$$\sum_{e \in P_j - P_j'} c_e \left[H(x_e) - H(x_e - 1) \right] = \sum_{e \in P_j - P_j'} \frac{c_e}{x_e}$$

- 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.

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

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 $\Phi(P_1, \ldots, P_k) \leq \Phi(P_1^*, \ldots, P_k^*)$.

$$C(P_1, \ldots, P_k) \leq \Phi(P_1, \ldots, P_k) \leq \Phi(P_1^*, \ldots, P_k^*) \leq H(k) \cdot C(P_1^*, \ldots, P_k^*)$$

previous claim applied to P

previous claim applied to P*

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.

Big open problem. Find any Nash equilibrium in poly-time

Next Time: Lower Bounds