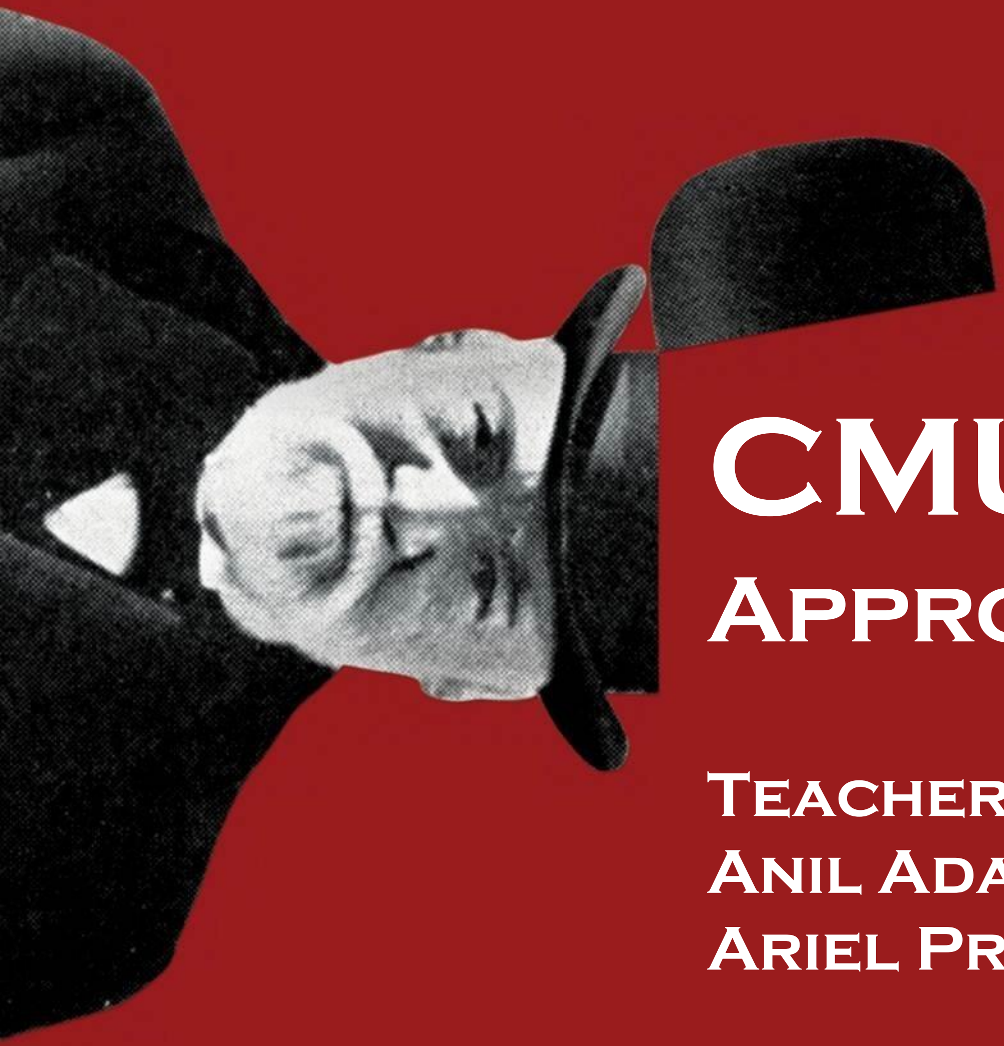


CMU 15-251

Approximation Algs

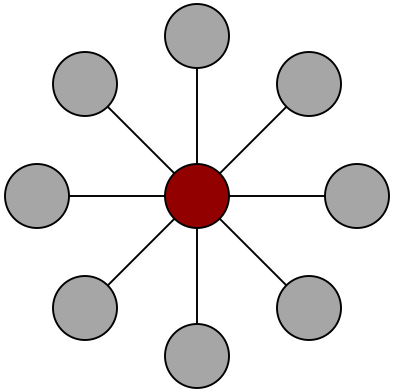
Teachers:
Anil Ada
Ariel Procaccia (this time)

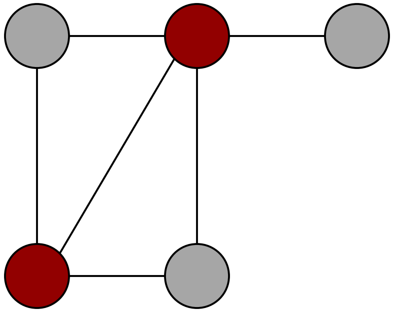
Computational hardness

- We saw that NP-hardness can be a force for good (preventing manipulation)
- But typically it just gets in the way of solving problems we want to solve!
- What can we do?
  - In practice: Heuristics often work well
  - In theory: Run in polynomial time and provide formal guarantees wrt the quality of the solution

Vertex cover

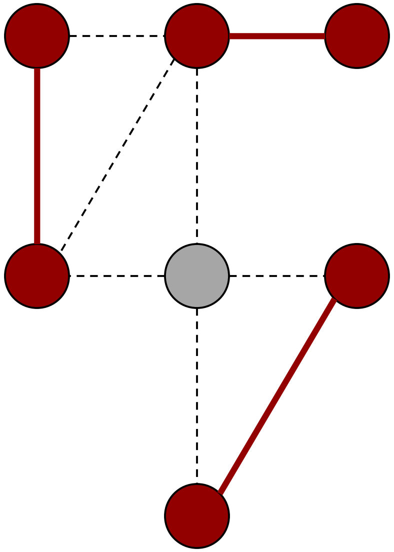
- VERTEX-COVER: Given a graph $G = (V, E)$ find the smallest $S \subseteq V$ such that every edge in E is incident on a vertex in S
- Decision version of the problem is NP-complete





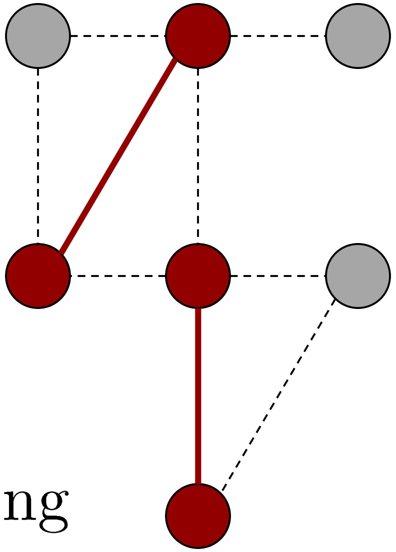
Vertex cover

- We don't know the size of the optimal vertex cover, but…
- Lemma: Let M be a matching in G, and S be a vertex cover. Then $|S| \geq |M|$
- Proof: S must cover at least one vertex for each edge in M; this covers no other edges in M ∎



Vertex cover

- Reminder: A matching M is maximal if $\nexists$ matching $M' \neq M$ such that $M \subseteq M'$
- Poll 1: Which of the following algs would find a maximal matching:
  1. Greedily add edges that are disjoint from the edges added so far, while such edges exist
  2. Compute a maximum cardinality matching
  3. Both
  4. Neither



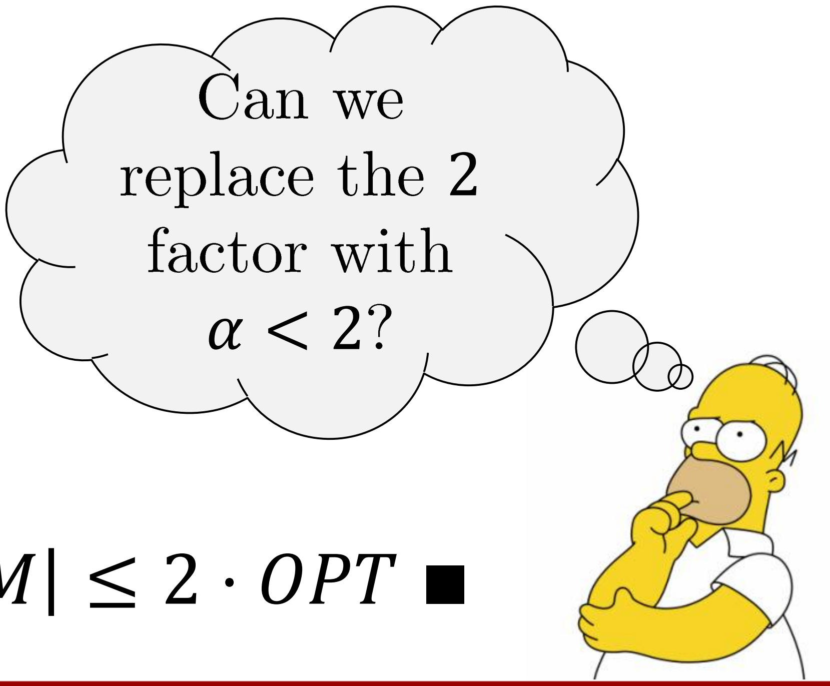
Vertex cover

<u>Approx-VC</u>(G)
$M \leftarrow$ maximal matching on G
$S \leftarrow$ all vertices incident on M
Return S

- Theorem: Given a graph G, let $OPT(G)$ be the size of the optimal vertex cover and $S =$ Approx-VC(G); S is a valid cover with $|S| \leq 2 \cdot OPT(G)$

Vertex cover

- Theorem: Given a graph G, let $OPT(G)$ be the size of the optimal vertex cover and $S = \textsc{Approx-VC}(G)$; S is a valid cover with $|S| \leq 2 \cdot OPT(G)$
- Proof:
    - For each $e \in E$, at least one vertex is in M, so S is a valid vertex cover
    - By the lemma, $|S| = 2|M| \leq 2 \cdot OPT$ ■



Approximation

- For a minimization problem instance I and algorithm ALG, let $ALG(\mathrm{I})$ be the quality of the algorithm's output and $OPT(I)$ be the quality of the optimal solution
- For c > 1, ALG is a c-approximation alg if for every I, $ALG(I) \leq c \cdot OPT(I)$
- Approx-VC is a polytime 2-approximation algorithm for Vertex-Cover

Approximation

- For a maximization problem and $c < 1$, ALG is a c-approximation algorithm if for every I, $ALG(I) \geq c \cdot OPT(I)$

These notions allow us to circumvent NP-hardness by designing polynomial-time algs with formal worst-case guarantees!

Approximation

- Algorithm Stupid-Approx(G): Return all vertices of G (assume G is not empty)
- Poll 2: What is the smallest value of α for which Stupid-Approx is an α-approx algorithm for Vertex-Cover?

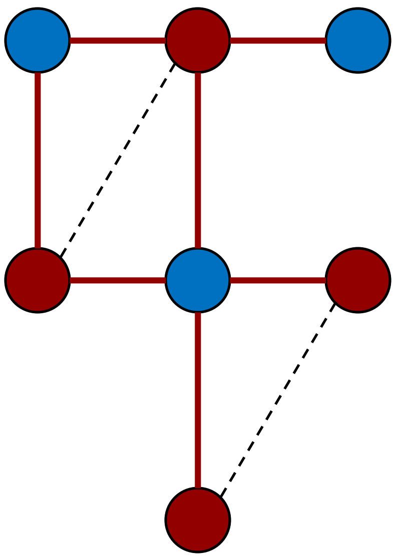
  1. $\alpha = 3$
  2. $\alpha = \log n$
  3. $\alpha = \lceil n/2 \rceil$
  4. $\alpha = n$

Max Cut

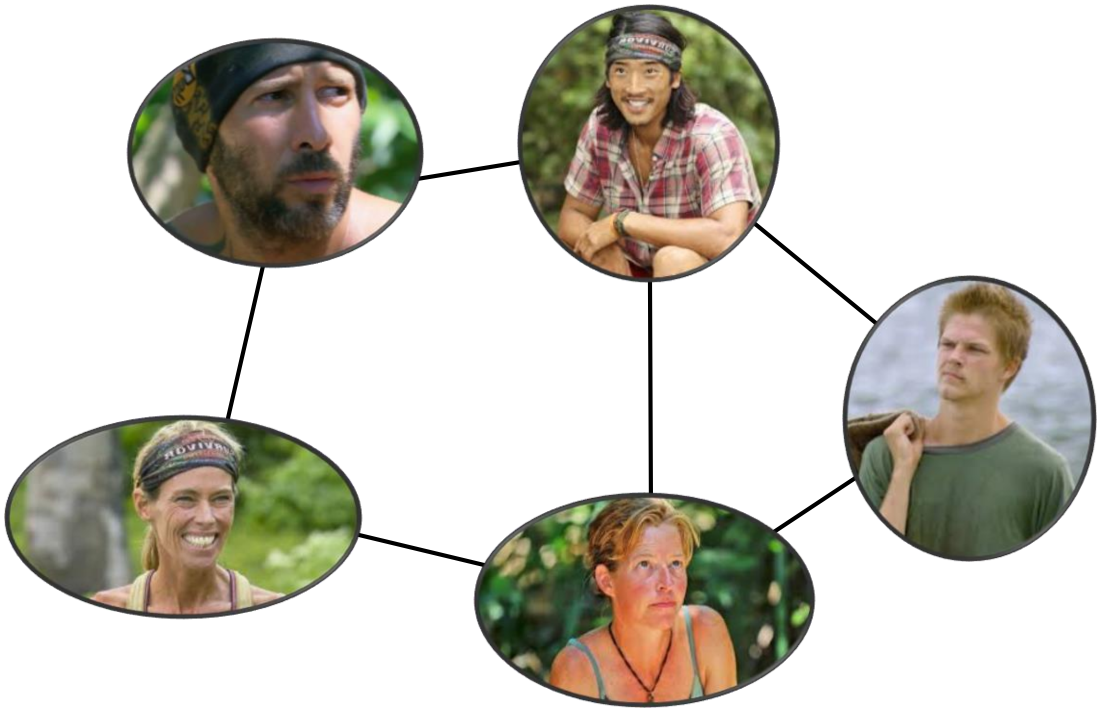
Ryan's favorite problem!

Max Cut

- Given a coloring of vertices in red and blue, an edge is a cut edge if and only if its endpoints have different colors
- Max Cut: Given a graph $G = (V, E)$, find a coloring of V in red and blue that maximizes the number of cut edges



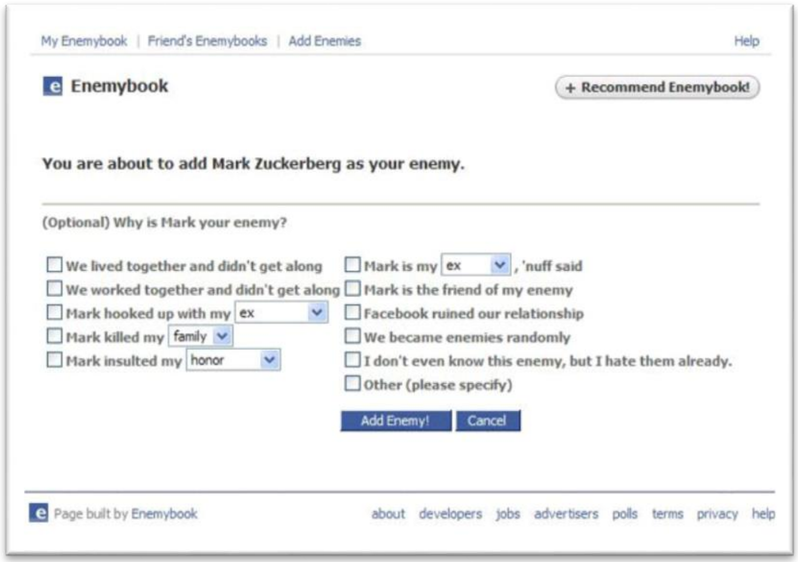
Max Cut



Partition into two tribes to break as many friendships as possible (to maximize drama)

Max Cut

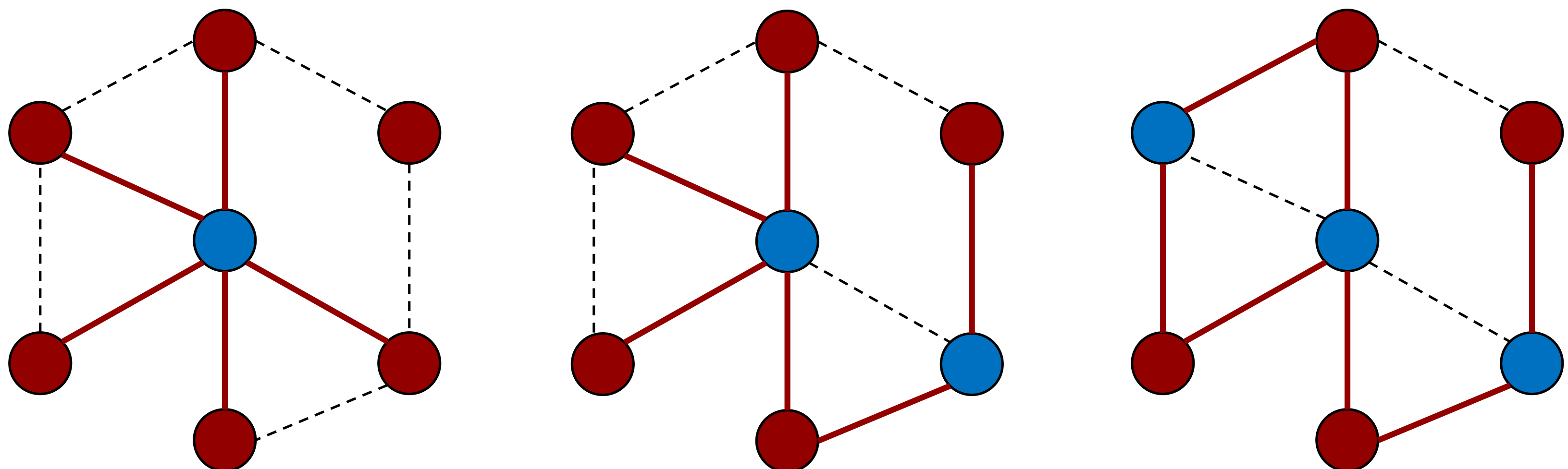
More natural if the social network recorded "enemyships" instead of friendships



Enemybook
Keep your friends close and your enemies closer...

Max Cut

<u>Approx-MC</u>(G)
Start from arbitrary coloring
While $\exists$vertex v such that changing its color increases the number of cut edges
    Change the color of v



Max Cut

Approx-MC(G)
Start from arbitrary coloring
Loop
If $\exists$vertex such that changing its color increases the number of cut edges, change its color

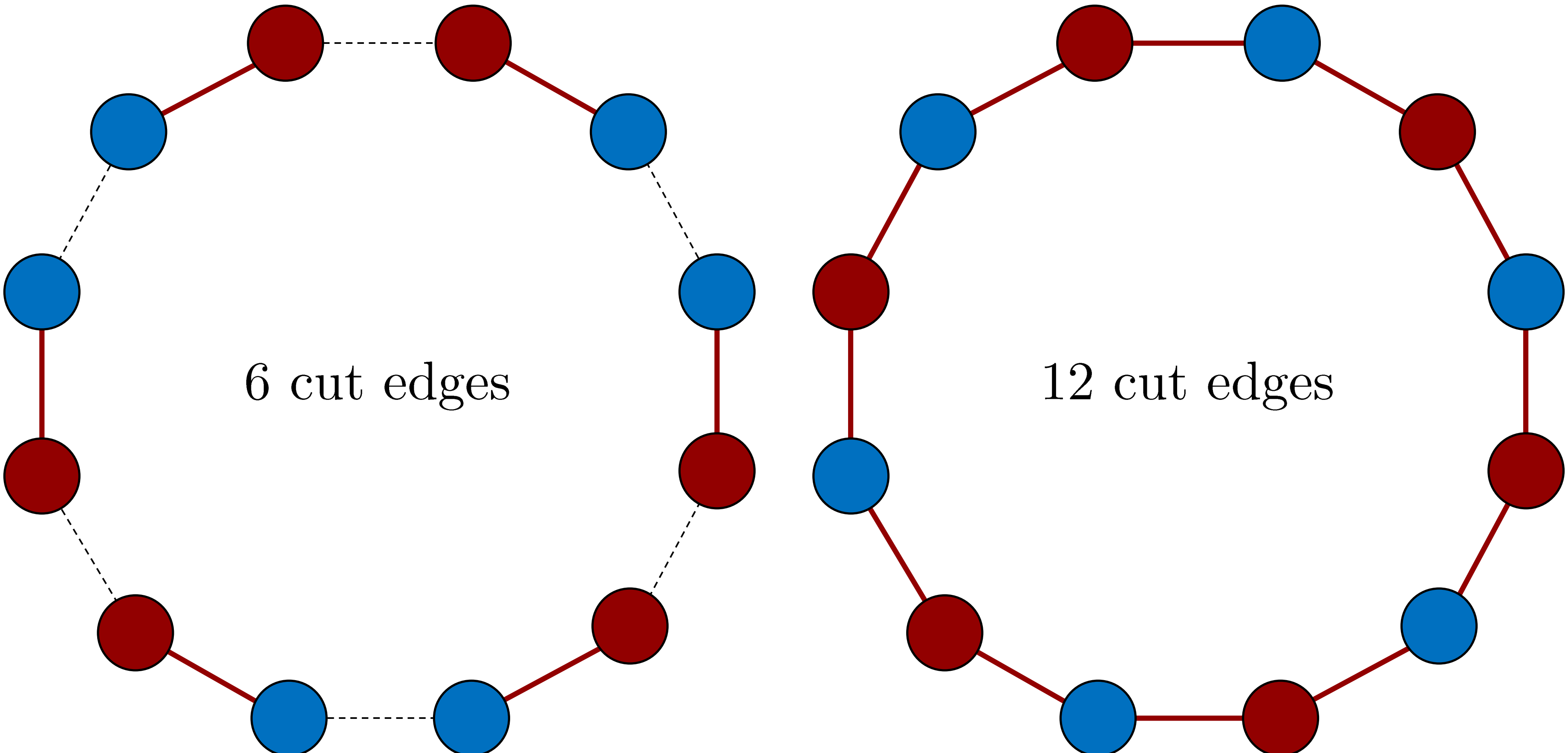
- Poll 3: What is the maximum number of iterations in the worst case?
  1. $\Theta(m)$
  2. $\Theta(mn)$
  3. $\Theta(m^2)$
  4. $\Theta(m^2n)$

Polynomial time!

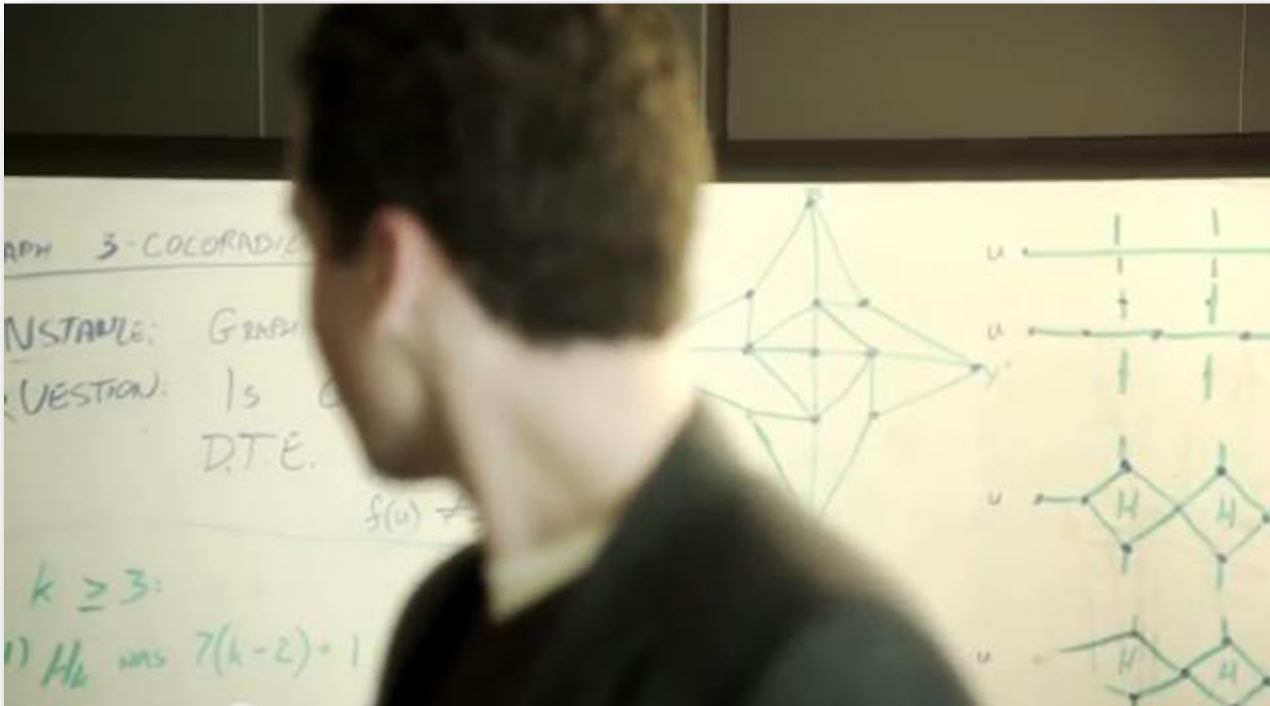
Max Cut

- Theorem: Approx-MC is a $\frac{1}{2}$-approximation algorithm for Max Cut
- Proof:
  - When the algorithm returns, each $v \in V$ has at least $\deg(v)/2$ of its edges cut (why?)
  - Therefore, the solution is guaranteed to have at least $m/2$ cut edges (exercise)
  - $OPT \leq m$ ∎

Max Cut



Interlude

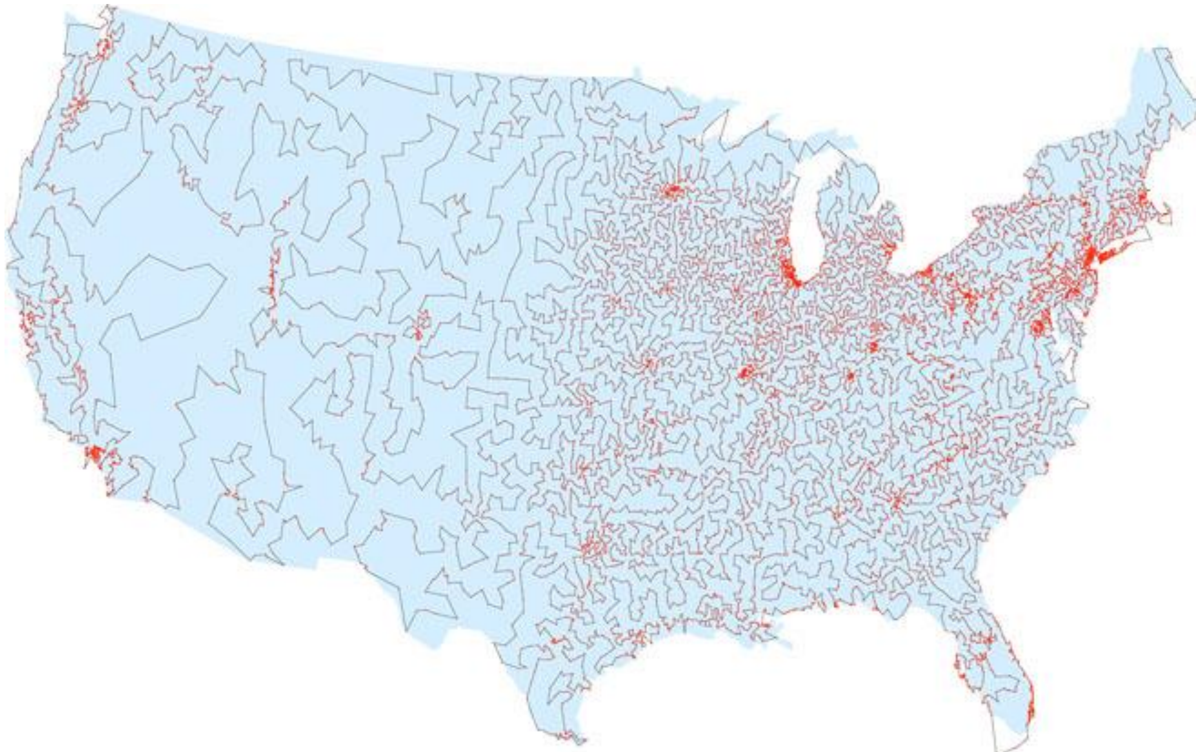


https://youtu.be/6ybd5rbQ5rU

Traveling Salesman

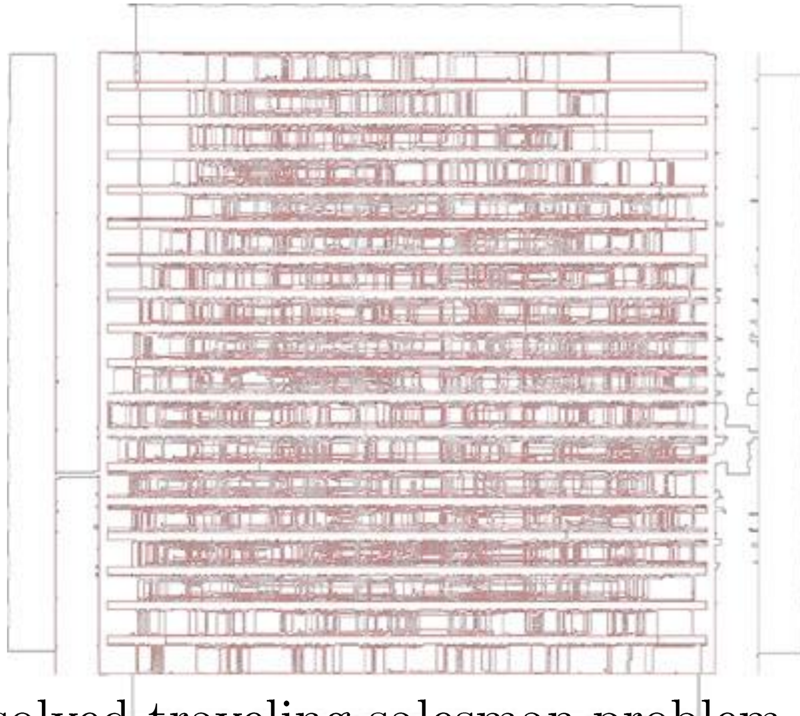
- Traveling-Salesman (TSP): Given a graph $G = (V, E)$ with edge costs $c: E \to \mathbb{N}$, find a minimum cost tour that visits each vertex exactly once
- NP-complete by reduction from Hamiltonian-cycle: Given an instance, assign $c(e) = 1$ for each $e \in E$ and ask whether there is a tour of cost n
- Metric TSP: can visit vertices multiple times (also NP-complete)

Traveling Salesman



Shortest traveling salesman route going through all 13,509 cities in the United States with a population of at least 500 (as of 1998)

Traveling Salesman



The largest solved traveling salesman problem (as of 2013), an 85,900-vertex route calculated in 2006. The graph corresponds to the design of a customized computer chip created at Bell Laboratories, and the solution exhibits the shortest path for a laser to follow as it sculpts the chip.

Traveling Salesman

Finding long chains in kidney exchange using the traveling salesman problem

Ross Anderson[a], Itai Ashlagi[b], David Gamarnik[b], and Alvin E. Roth[c,1]

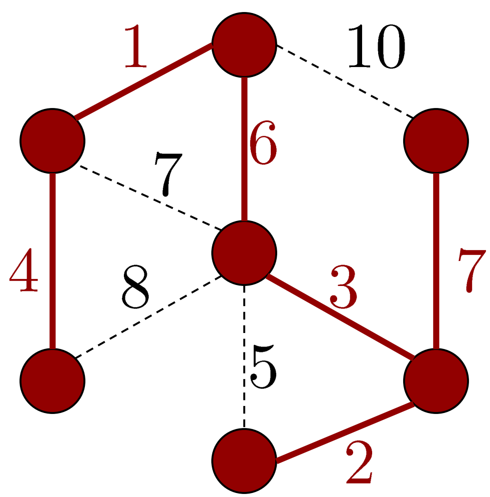
[a]Operation Research Center and [b]Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02142; and [c]Department of Economics, Stanford University, Stanford, CA 94305

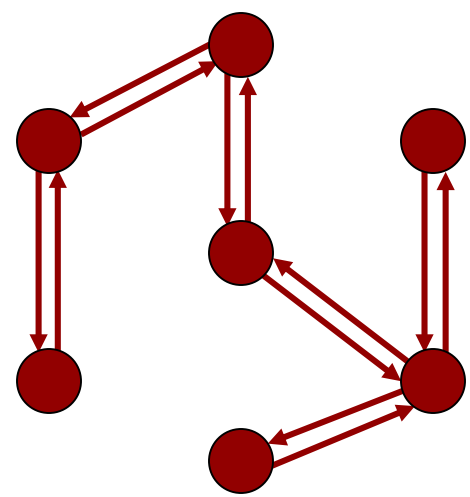
Contributed by Alvin E. Roth, November 18, 2014 (sent for review July 15, 2014; reviewed by Peter Biro)

Anderson et al., PNAS 2015

Traveling Salesman

<u>APPROX-TSP</u>(G)
$T \leftarrow$ Minimum spanning tree of G
$2T \leftarrow$ double edges of T
Return Eulerian tour of $2T$



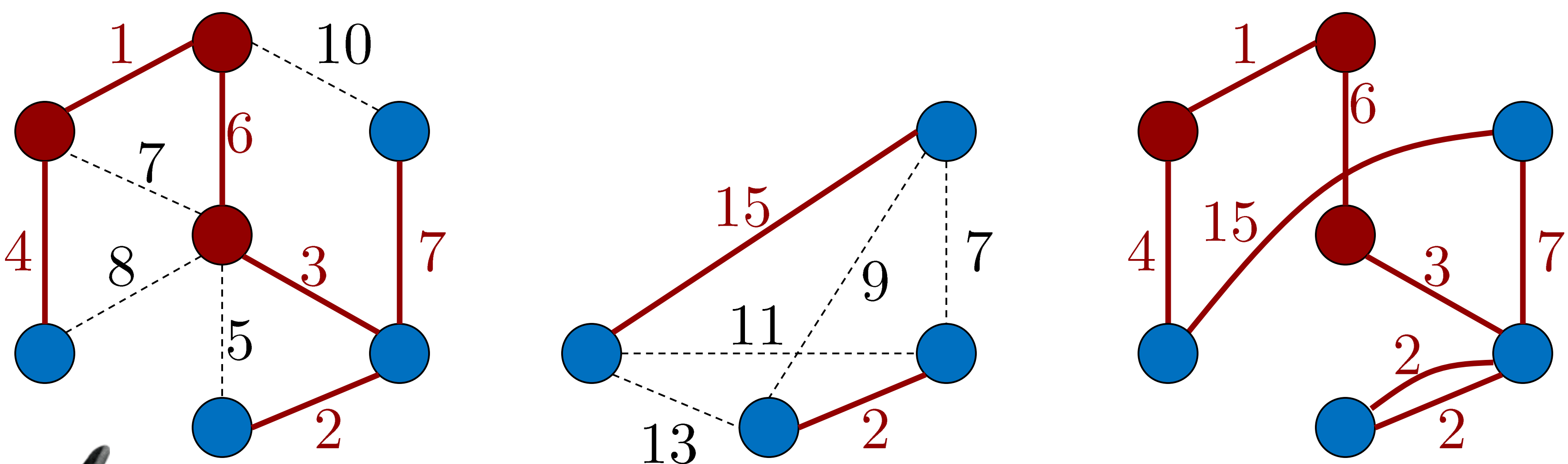


Traveling Salesman

- Theorem: Approx-TSP is a 2-approximation algorithm for Metric TSP
- Proof:
  - A TSP tour can be converted into a lower cost spanning tree (how?), therefore
$$c(T) = \sum_{e \in E(T)} c(e) \leq OPT$$
  - Clearly $c(2T) = 2c(T)$
  - It follows that $c(2T) \leq 2OPT$ ■

Traveling Salesman*

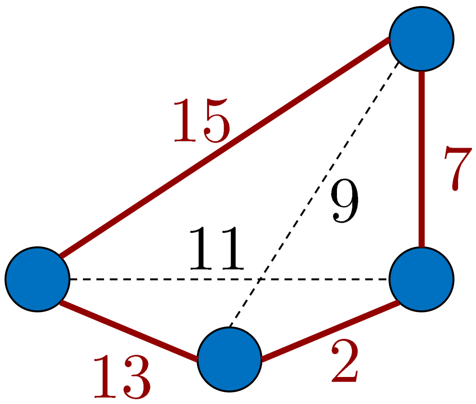
<u>Christofides</u>(G)
$T \leftarrow$ Minimum spanning tree of G
$S \leftarrow$ Vertices of odd degree in T ($|S|$ is even, why?)
$M \leftarrow$ Min cost perfect matching on S in G
Return Eulerian tour of $T \cup M$ (it exists, why?)

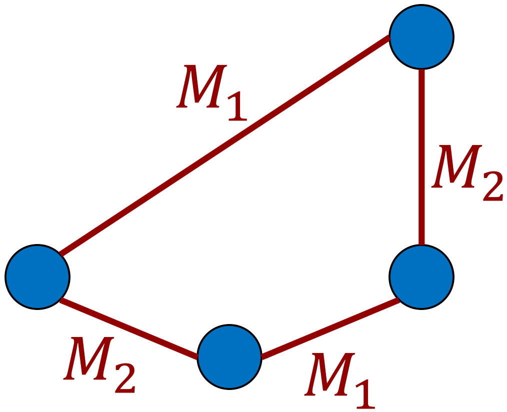


* Just for fun

Traveling Salesman*

- Lemma: $C(M) \leq \frac{1}{2} OPT$
- Proof:
  - $\exists$ tour of S of cost at most OPT (because $S \subseteq V$)
  - Decompose into two matchings M_1 and M_2
  - $c(M_1) + c(M_2) \leq OPT$, but $c(M) \leq c(M_1)$ and $c(M) \leq c(M_2) \Rightarrow c(M) \leq \frac{1}{2} OPT$ ■





Traveling Salesman*

- Theorem: Christofides is a $\frac{3}{2}$-approximation algorithm for Metric TSP
- Proof: Using the lemma,

$$
\begin{aligned}
ALG &= c(M) + c(T) \\
&\leq \frac{1}{2} OPT + \mathrm{OPT} \\
&= \frac{3}{2} \mathrm{OPT} \;\blacksquare
\end{aligned}
$$

* Just for fun

Summary

- Definitions
  - Approximation algorithm
  - Vertex-Cover, Max Cut, traveling-Salesman
- Algorithms
  - 2-approximation for Vertex-Cover
  - 2-approximation for Max Cut
  - 2-approximation for Metric TSP