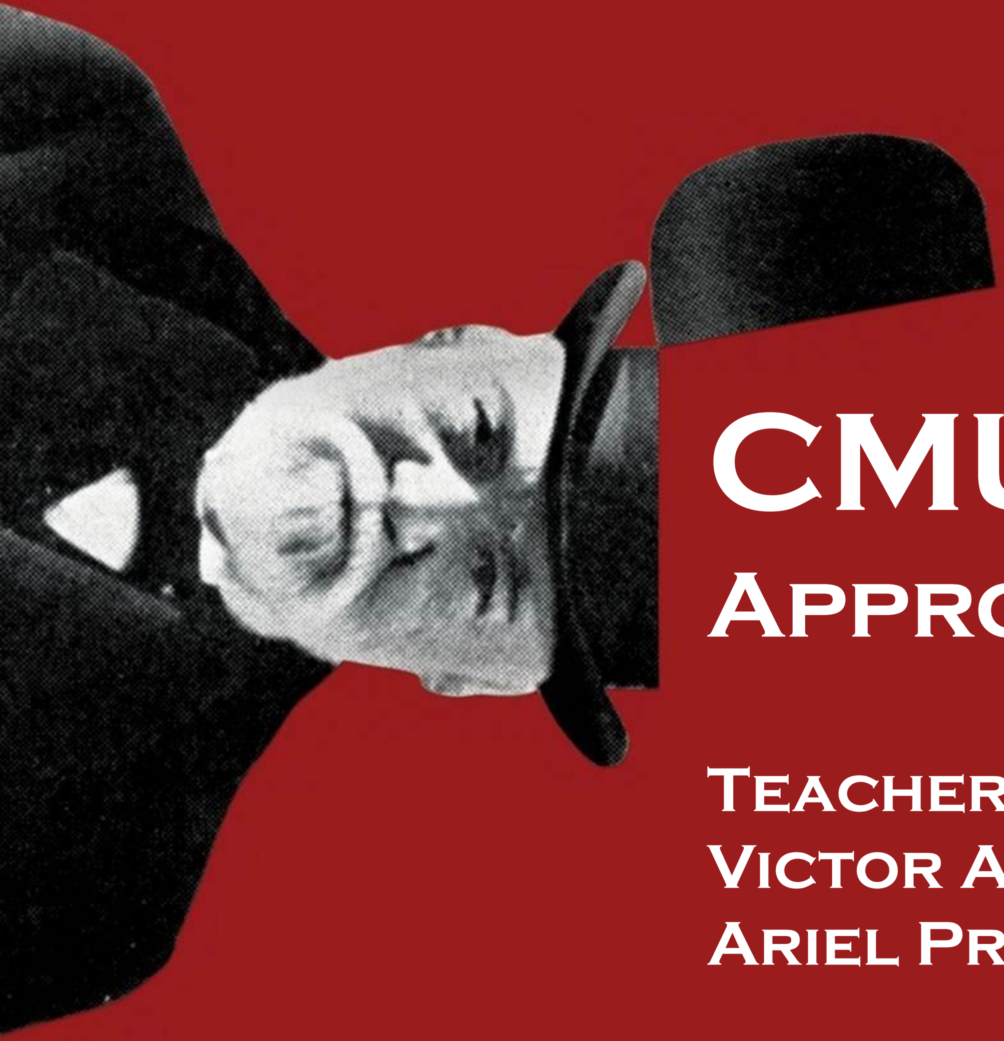


CMU 15-251
Approximation Algs

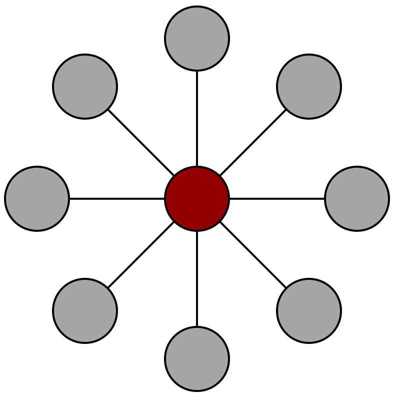
Teachers:
Victor Adamchik
Ariel Procaccia (this time)

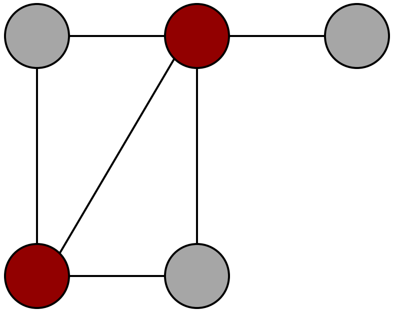
Computational hardness

- 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

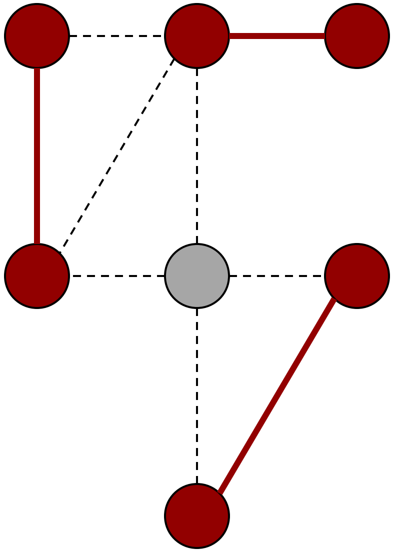
- Vertex-Cover: Given a graph $G = (V, E)$ find the smallest $S \subseteq V$ such that every edge in E is incident on a vertex in S
- Decision version of the problem is NP-complete (HW7 Q4)





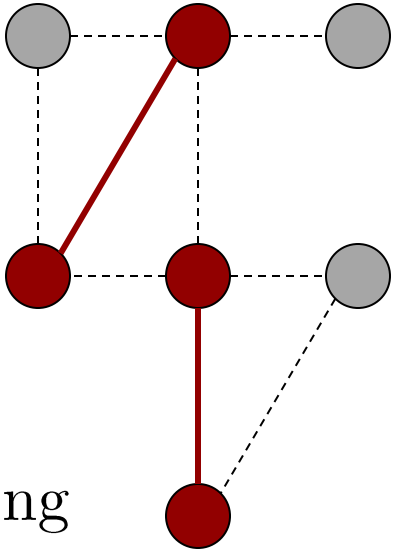
Vertex cover

- 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

- A matching M is maximal if $\nexists$ matching $M' \neq M$ such that $M \subseteq M'$
- Vote: 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

Approx-VC(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 $S = \text{Approx-VC}(G)$; S is a valid cover with $|S| \le 2 \cdot OPT(G)$
- **Proof:**
  - For each edge, at least one vertex is in M, so S is a valid vertex cover
  - By the lemma, $|S| = 2|M| \le 2 \cdot OPT$ ■

Can we replace the 2 factor with $\alpha < 2$?

Approximation

- For a minimization problem instance I and algorithm ALG, let ALG(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, $ALG(I) \geq c \cdot OPT(I)$

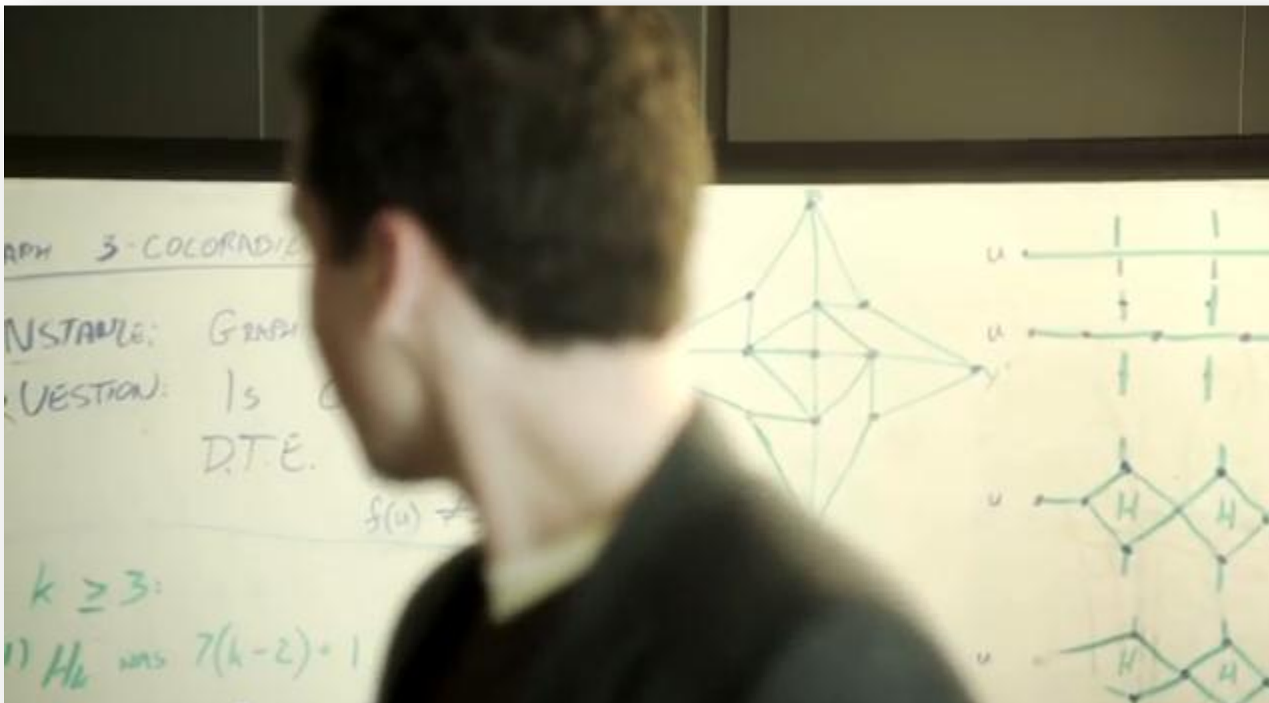
These notions allow us to circumvent NP-hardness by designing polynomial-time algs with formal worst-case guarantees!

Approximation

- Algorithm Stupid-Approx(G): Return all vertices of G
- Vote: 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. $\alpha = n$

Interlude

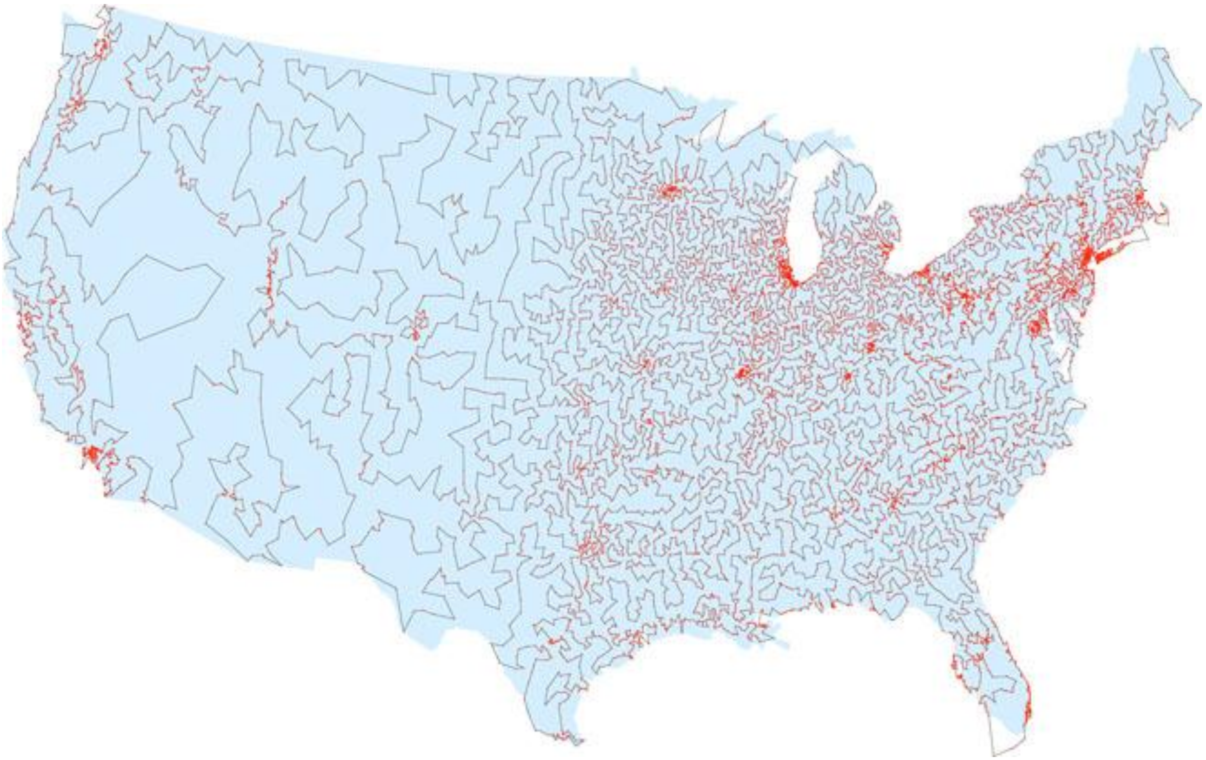
- http://www.youtube.com/watch?v=6ybd5rbQ5rU



Traveling Salesman

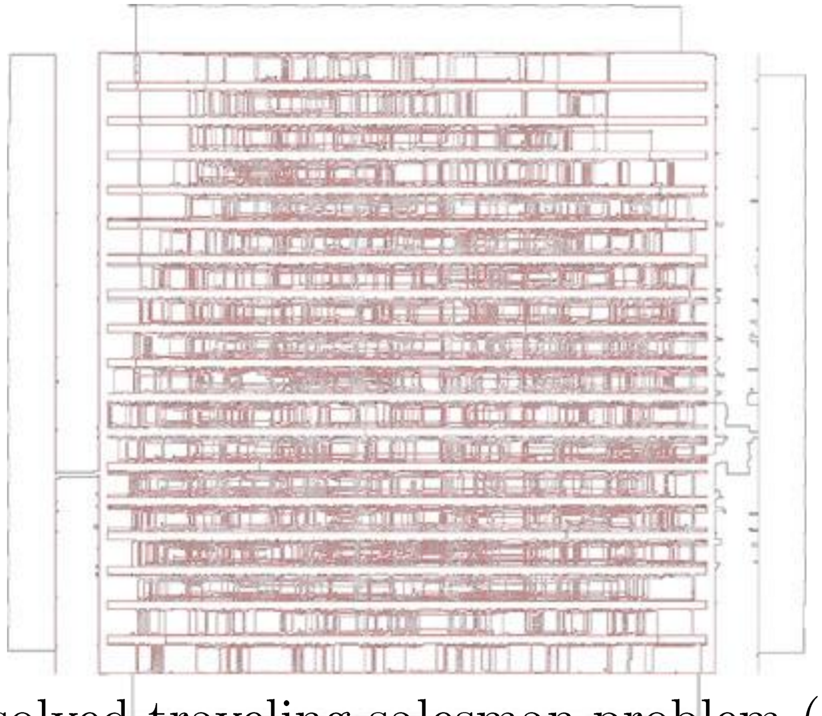
- Traveling-Salesman (TSP): Given a graph $G = (V, E)$ with edge costs $c: E \to \mathbb{N}$, find a minimum cost tour that visits each vertex exactly once
- NP-complete by reduction from Hamiltonian-cycle: Given an instance, assign $c(e) = 1$ for each $e \in E$ and ask whether there is a tour of cost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

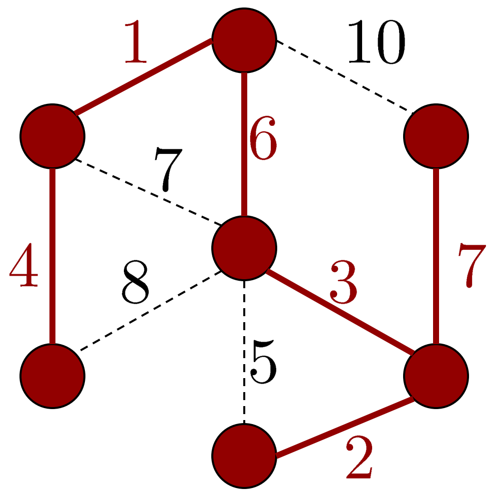
Traveling salesman

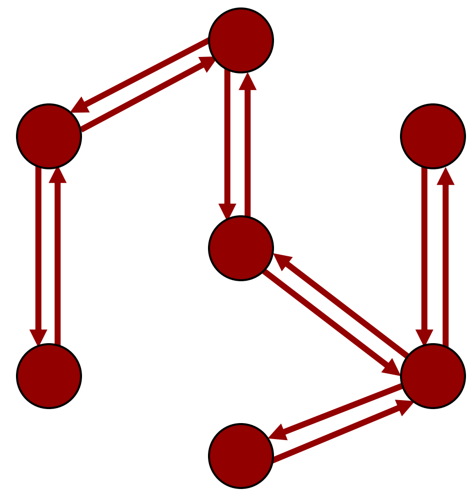


The largest solved traveling salesman problem (as of 2013), an 85,900-vertex route calculated in 2006. The graph corresponds to the design of a customized computer chip created at Bell Laboratories, and the solution exhibits the shortest path for a laser to follow as it sculpts the chip.

Traveling salesman

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





Traveling Salesman

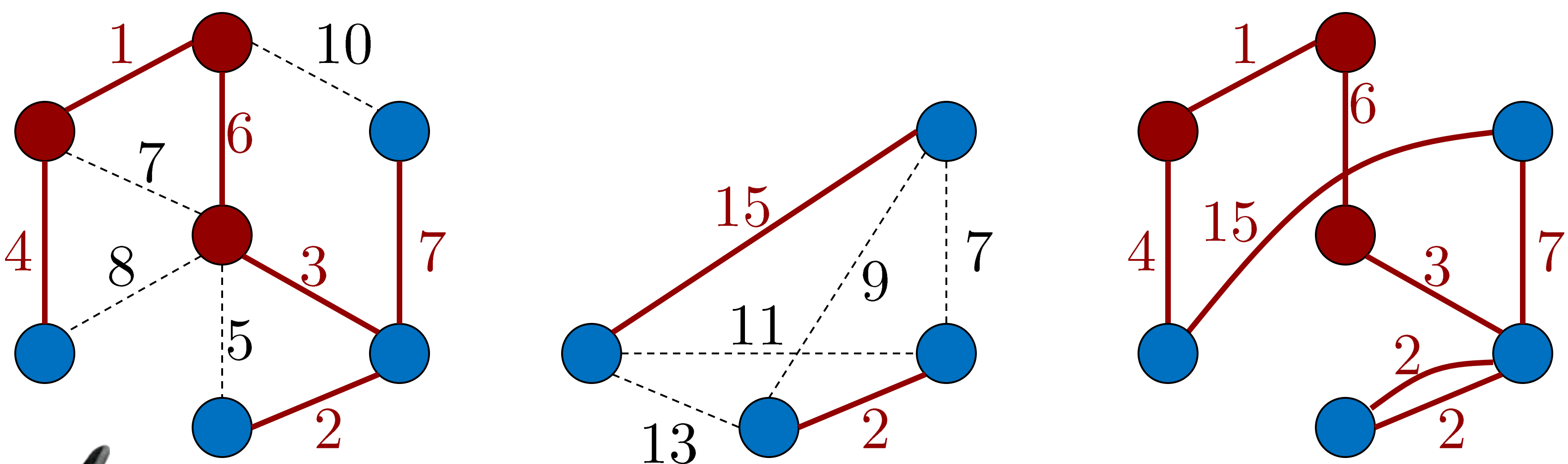
- **Theorem**: APPROX-TSP is a 2-approximation algorithm for Metric TSP
- **Proof:**
  - A TSP tour can be converted into a lower cost spanning tree (how?), therefore

$$c(T) = \sum_{e \in E(T)} c(e) \leq OPT$$

  - Clearly $c(2T) = 2c(T)$
  - It follows that $c(2T) \leq 2OPT$ ■

Traveling salesman*

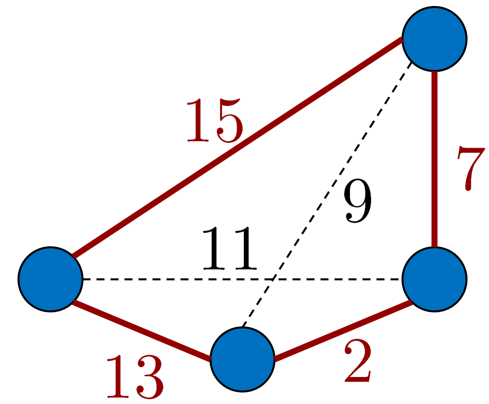
CHRISTOFIDES(G)
$T \leftarrow$ Minimum spanning tree of G
$S \leftarrow$ Vertices of odd degree in T ($|S|$ is even, why?)
$M \leftarrow$ Min cost matching on S in G
Return Eulerian tour of $T \cup M$

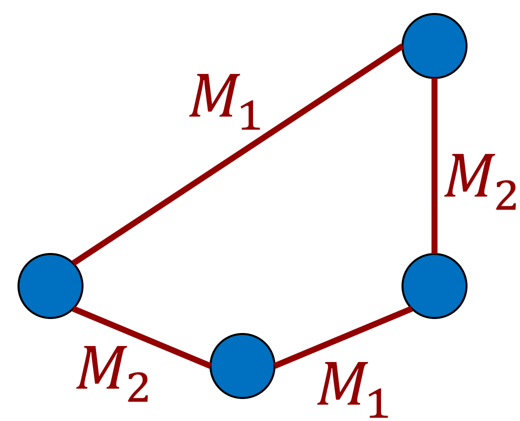


* Not for the exam

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 $c(M) \leq c(M_2) \Rightarrow c(M) \leq \frac{1}{2} OPT$ ■





* Not for the exam

TRAVELING SALESMAN*

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

$$ALG = c(M) + c(T)$$
$$\leq \frac{1}{2}OPT + \text{OPT}$$
$$= \frac{3}{2}\text{OPT} \blacksquare$$

* Not for the exam

What we have learned

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