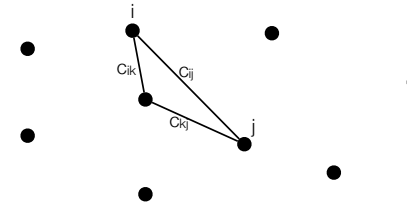


# Traveling salesman problem

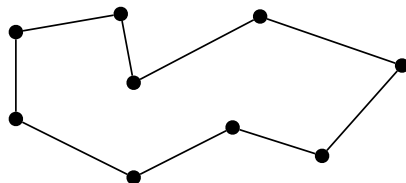
Inge Li Gørtz

## Traveling Salesman Problem (TSP)



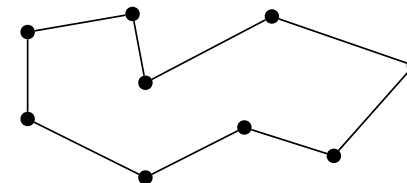
- Set of cities  $\{1, \dots, n\}$
- $c_{ij} \geq 0$ : cost of traveling from  $i$  to  $j$ .
- $c_{ij}$  a metric:
  - $c_{ii} = 0$
  - $c_{ij} = c_{ji}$
  - $c_{ij} \leq c_{ik} + c_{kj}$  (triangle inequality)
- Goal: Find a *tour of minimum cost visiting every city exactly once*.

## Traveling Salesman Problem (TSP)



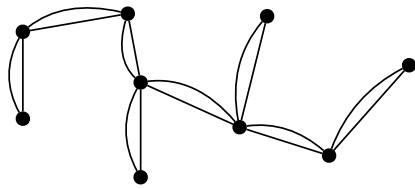
- Set of cities  $\{1, \dots, n\}$
- $c_{ij} \geq 0$ : cost of traveling from  $i$  to  $j$ .
- $c_{ij}$  a metric:
  - $c_{ii} = 0$
  - $c_{ij} = c_{ji}$
  - $c_{ij} \leq c_{ik} + c_{kj}$
- Goal: Find a *tour of minimum cost visiting every city exactly once*.

## Double tree algorithm



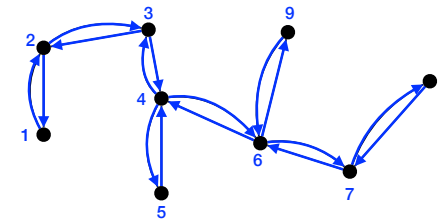
- MST is a lower bound on TSP.
  - Deleting an edge  $e$  from OPT gives a spanning tree.
  - $\text{OPT} \geq \text{OPT} - c_e \geq \text{MST}$ .

## Double tree algorithm



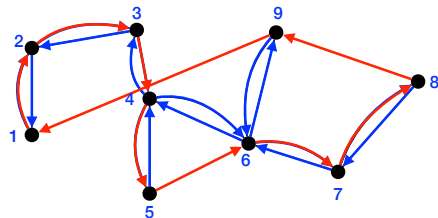
- Double tree algorithm
  - Compute MST  $T$ .
  - Double edges of  $T$
  - Construct Euler tour  $\tau$  (a tour visiting every edge exactly once).

## Double tree algorithm



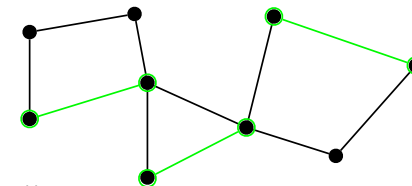
- Double tree algorithm
  - Compute MST  $T$ .
  - Double edges of  $T$
  - Construct Euler tour  $\tau$  (a tour visiting every edge exactly once).

## Double tree algorithm



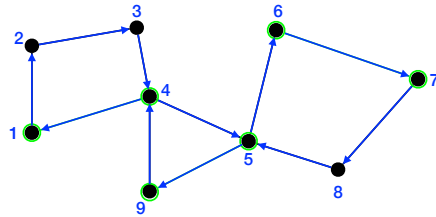
- Double tree algorithm
  - Compute MST  $T$ .
  - Double edges of  $T$
  - Construct Euler tour  $\tau$  (a tour visiting every edge exactly once).
  - Shortcut  $\tau$  such that each vertex only visited once ( $\tau'$ )
- $\text{length}(\tau') \leq \text{length}(\tau) = 2 \text{ cost}(T) \leq 2 \text{ OPT}$ .
- The double tree algorithm is a 2-approximation algorithm for TSP.

## Christofides' algorithm



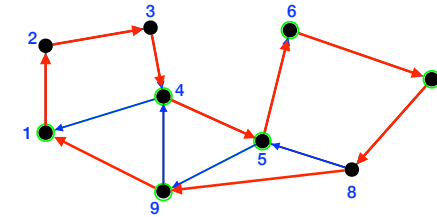
- Christofides' algorithm
  - Compute MST  $T$ .
  - No need to double all edges:
    - Enough to turn it into an Eulerian graph: A graph Eulerian if there is a traversal of all edges visiting every edge exactly once.
      - G Eulerian iff G connected and all nodes have even degree.
    - Consider set  $O$  of all odd degree vertices in  $T$ .
    - Find minimum cost perfect matching  $M$  on  $O$ .
      - Matching: no edges share an endpoint.
      - Perfect: all vertices matched.
      - Perfect matching on  $O$  exists: Number of odd vertices in a graph is even.
    - $T + M$  is Eulerian (all vertices have even degree).

## Christofides' algorithm



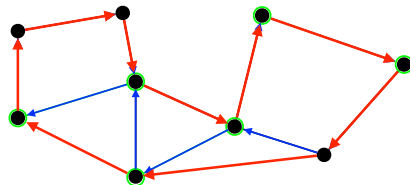
- Christofides' algorithm
  - Compute MST  $T$ .
  - $O = \{\text{odd degree vertices in } T\}$ .
  - Compute minimum cost perfect matching  $M$  on  $O$ .
  - Construct Euler tour  $\tau$
  - Shortcut such that each vertex only visited once ( $\tau'$ )

## Christofides' algorithm



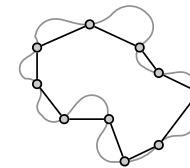
- Christofides' algorithm
  - Compute MST  $T$ .
  - $O = \{\text{odd degree vertices in } T\}$ .
  - Compute minimum cost perfect matching  $M$  on  $O$ .
  - Construct Euler tour  $\tau$
  - Shortcut such that each vertex only visited once ( $\tau'$ )

## Christofides' algorithm



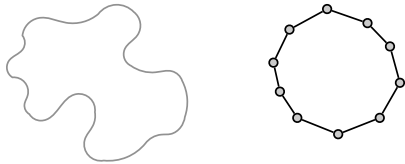
- Christofides' algorithm
  - Compute MST  $T$ .
  - $O = \{\text{odd degree vertices in } T\}$ .
  - Compute minimum cost perfect matching  $M$  on  $O$ .
  - Construct Euler tour  $\tau$
  - Shortcut such that each vertex only visited once ( $\tau'$ )
- $\text{length}(\tau') \leq \text{length}(\tau) = \text{cost}(T) + \text{cost}(M) \leq \text{OPT} + \text{cost}(M)$ .

## Analysis of Christofides' algorithm



- $\text{cost}(M) \leq \text{OPT}/2$ .
  - $\text{OPT}_o = \text{OPT}$  restricted to  $O$ .
  - $\text{OPT}_o \leq \text{OPT}$ .

## Analysis of Christofides' algorithm



- $\text{cost}(M) \leq \text{OPT}/2$ .
  - $\text{OPT}_O = \text{OPT}$  restricted to  $O$ .
  - $\text{OPT}_O \leq \text{OPT}$ .

## Analysis of Christofides' algorithm



- $\text{cost}(M) \leq \text{OPT}/2$ :
  - $\text{OPT}_O = \text{OPT}$  restricted to  $O$ .
  - $\text{OPT}_O \leq \text{OPT}$ .
  - can partition  $\text{OPT}_O$  into two perfect matchings  $O_1$  and  $O_2$ .
  - $\text{cost}(M) \leq \min(\text{cost}(O_1), \text{cost}(O_2)) \leq \text{OPT}/2$ .
- $\text{length}(\tau') \leq \text{length}(\tau) = \text{cost}(T) + \text{cost}(M) \leq \text{OPT} + \text{OPT}/2 = 3/2 \text{ OPT}$ .
- Christofides' algorithm is a  $3/2$ -approximation algorithm for TSP.

## Hardness of Approximation

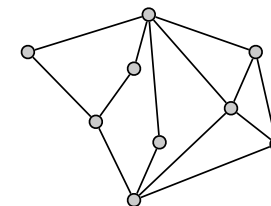
Inge Li Gørtz

## TSP: Inapproximability

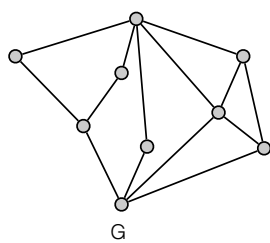
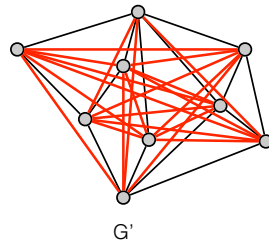
- There is no  $\alpha$ -approximation algorithm for the non-metric TSP for unless  $P=NP$ .



- *Hamiltonian cycle*. Given  $G=(V,E)$ . Is there a cycle visiting every vertex exactly once?

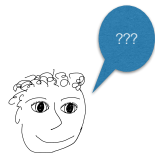


## TSP: Inapproximability



— cost 1  
— cost  $5n+1$   
= 46

- $G$  has a Hamiltonian cycle  $\Leftrightarrow$  optimal cost of TSP in  $G'$  is  $n = 9$ .
- $G$  has no Hamiltonian cycle  $\Leftrightarrow$  optimal cost of TSP in  $G'$  is at least  $n - 1 + 5n + 1$   
=  $6n = 8 + 46 = 54$

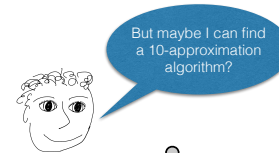
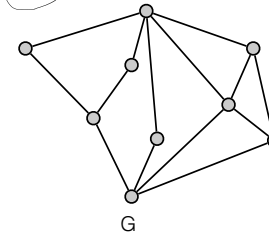
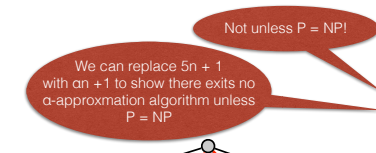


If there is a HC in  $G$  then the 5-approximation algorithm returns a tour of cost  $\leq 45$ .

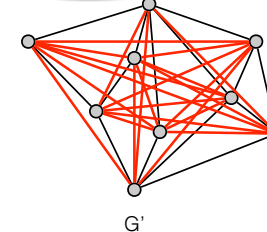


If there is **no** HC in  $G$  then the 5-approximation algorithm returns a tour of cost  $\geq 54$ .

## TSP: Inapproximability



We can replace  $5n + 1$  with  $\alpha n + 1$  to show there exists no  $\alpha$ -approximation algorithm unless  $P = NP$ .



— cost 1  
— cost  $\alpha n + 1$

- $G$  has a Hamiltonian cycle  $\Leftrightarrow$  optimal cost of TSP in  $G'$  is  $n$ .
- $G$  has no Hamiltonian cycle  $\Leftrightarrow$  optimal cost of TSP in  $G' \geq n - 1 + (\alpha n + 1)$   
=  $(\alpha + 1)n$

## k-center: Inapproximability

- There is no  $\alpha$ -approximation algorithm for the k-center problem for  $\alpha < 2$  unless  $P=NP$ .
- **Proof.** Reduction from dominating set.
- **Dominating set.** Given  $G=(V,E)$  and  $k$ . Is there a (dominating) set  $S \subseteq V$  of size  $k$ , such that each vertex is either in  $S$  or adjacent to a vertex in  $S$ ?
- Given instance of the dominating set problem construct instance of k-center problem:
  - Complete graph  $G'$  on  $V$ .
  - All edges from  $E$  has weight 1, all new edges have weight 2.
  - Radius in k-center instance 1 or 2.
  - $G$  has an dominating set of size  $k \Leftrightarrow$  opt solution to the k-center problem has radius 1.
  - Use  $\alpha$ -approximation algorithm  $A$ :
    - $\text{opt} = 1 \Rightarrow A$  returns solution with radius at most  $\alpha < 2$ .
    - $\text{opt} = 2 \Rightarrow A$  returns solution with radius at least 2.
    - Can use  $A$  to distinguish between the 2 cases.

