Scheduling and Optimization Course (MPRI)

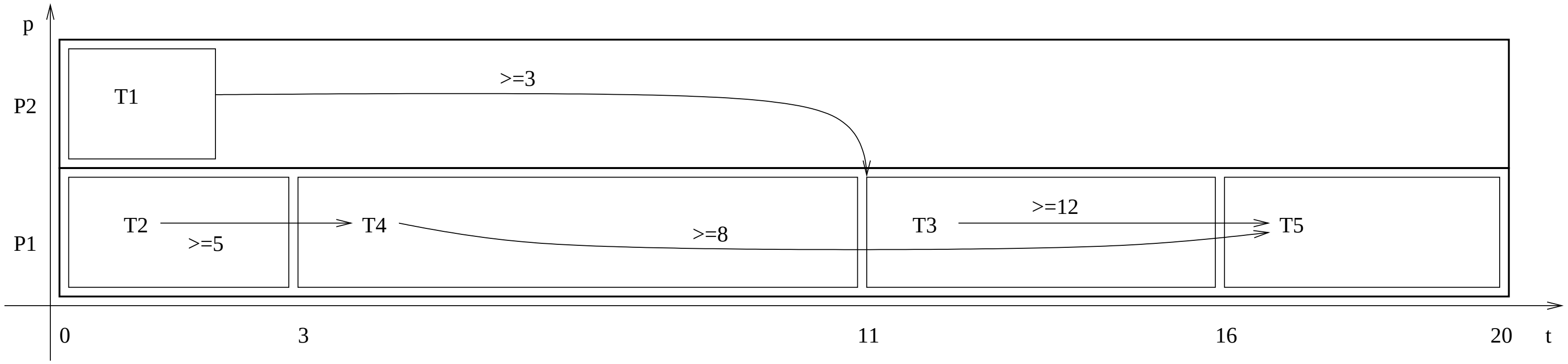
Leo Liberti

LIX, École Polytechnique, France

Teachers

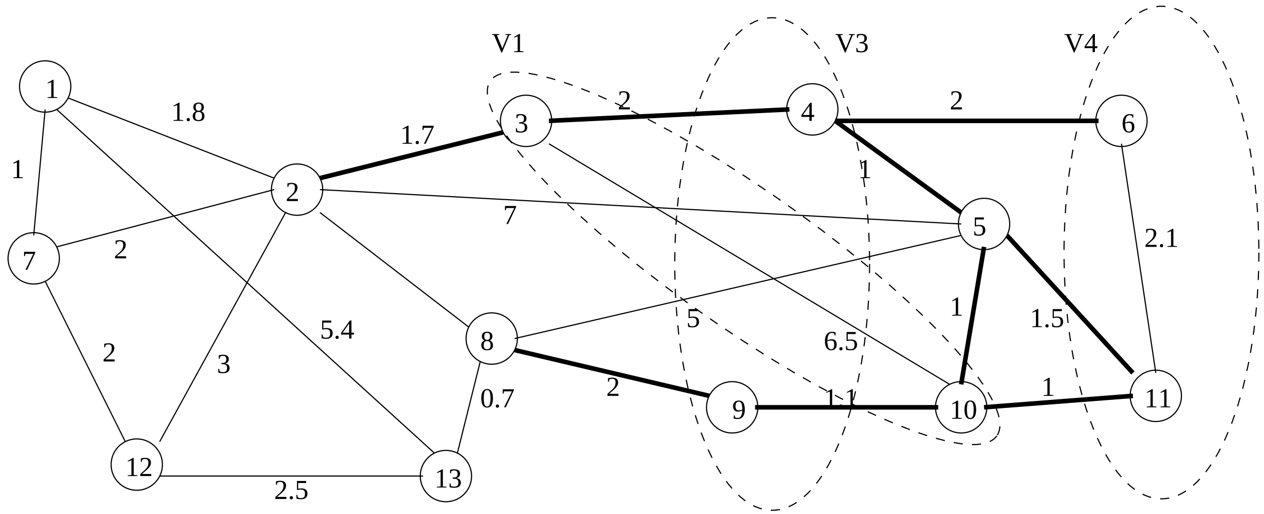
- Christoph Dürr
  http://www.lix.polytechnique.fr/~durr
  Refer to Christoph for any information
- Philippe Baptiste
  http://www.lix.polytechnique.fr/~baptiste
- Leo Liberti
  http://www.lix.polytechnique.fr/~liberti

Scheduling



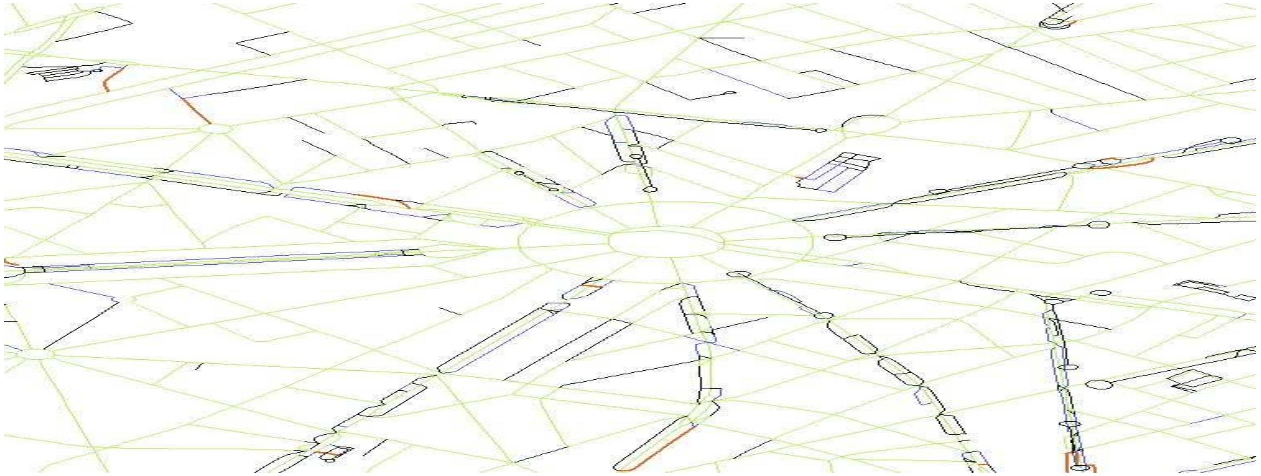
- **Schedule n tasks on m machines such that the sum of completion times is minimum**
- Scheduling = assignment + ordering
- Additional constraints: given precedence on tasks, delays under certain circumstances, time windows. . .
- Many industrial applications
- Similar problems arise in project management

Network design



- **Break an existing telecom network such that the subnetworks have as few interconnections as possible**
- Happens when a huge telecom giant wants to sell off or sublet some subnetworks
- Associate a variable to each vertex i and partition h, arc presence can be modelled by quadratic term $x_{ih}x_{jk}$

Shortest paths



- **Find a shortest path between two geographical points**
- Variants: find shortest paths from one point to all others, find shortest paths among all pairs, find a set of k paths such that total length is shortest, . . .
- Additional constraints: arc weights as travelling times, real time computation, dynamic arc weights evolve with traffic

Important concepts

- **Optimization**: given a point set X and an *objective function* $f : X \to \mathbb{R}$, find the *optimal solution* x^* attaining the minimum (or maximum) value f^* on X
- X is called the *feasible region*
- Any point $x \in X$ is a *feasible point*
- Supposing $X \subseteq \mathbb{R}^n$, $x = (x_1, \ldots, x_n)$
- For $i \leq n$, x_i is a *problem variable*
- Any numerical constant on which f, X depend is a *problem parameter*

Main optimization problem classes

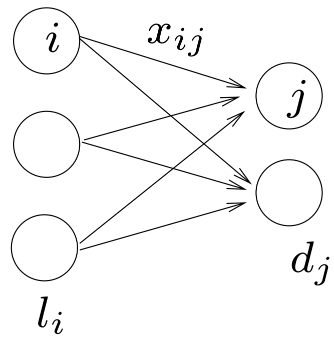
- X is usually of the form $\mathbb{R}^{n-k} \times \mathbb{Z}^k$
- $k = 0$: continuous problem, $k = n$: integer program; otherwise, mixed-integer problem
- If $X = \{x \in Y \mid \forall\, i \leq m\ (g_i(x) \leq 0)\}$, $g_i : Y \to \mathbb{R}$ are the *constraints*
  - f, g_i linear & $k = 0$: *Linear Programming* (LP)
  - f, g_i linear & $k > 0$: *Mixed-Integer Linear Programming* (MILP)
  - f, g_i nonlinear & $k = 0$: *NonLinear Programming* (NLP)
  - f, g_i nonlinear & $k > 0$: *Mixed-Integer NonLinear Programming* (MINLP)

Transportation problem

Let x_{ij} be the (discrete) number of product units transported from plant $i \leq m$ to customer $j \leq n$ with respective unit transportation cost c_{ij} from plant i to customer j.

Problem: find x minimizing the total cost, subject to production limits l_i at plant i and demand d_j at customer j.

$$\left.\begin{array}{rrcl}
\min_x & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & & \\
\forall\, i \leq m & \displaystyle\sum_{j=1}^{n} x_{ij} & \leq & l_i \\
\forall\, j \leq n & \displaystyle\sum_{i=1}^{m} x_{ij} & \geq & d_j \\
\forall\, i,j & x_{ij} \in \mathbb{Z}_+ & &
\end{array}\right\}$$

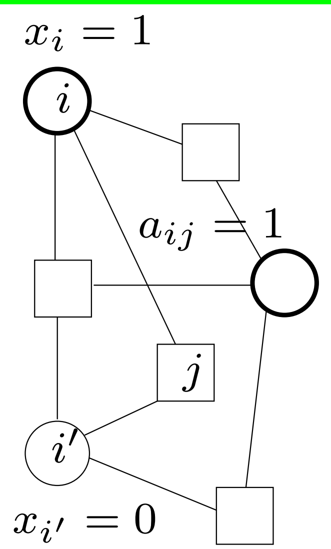


Facility Location problem

Let $x_i = 1$ if a servicing facility will be built on geographical region $i \leq m$ and 0 otherwise. The cost of building a facility on region i is c_i, and $a_{ij} = 1$ if a facility on region i can serve town $j \leq n$, and 0 otherwise.

Problem: find $x \in \{0,1\}^m$ so that each town is serviced by at least one facility and the total cost is minimum.

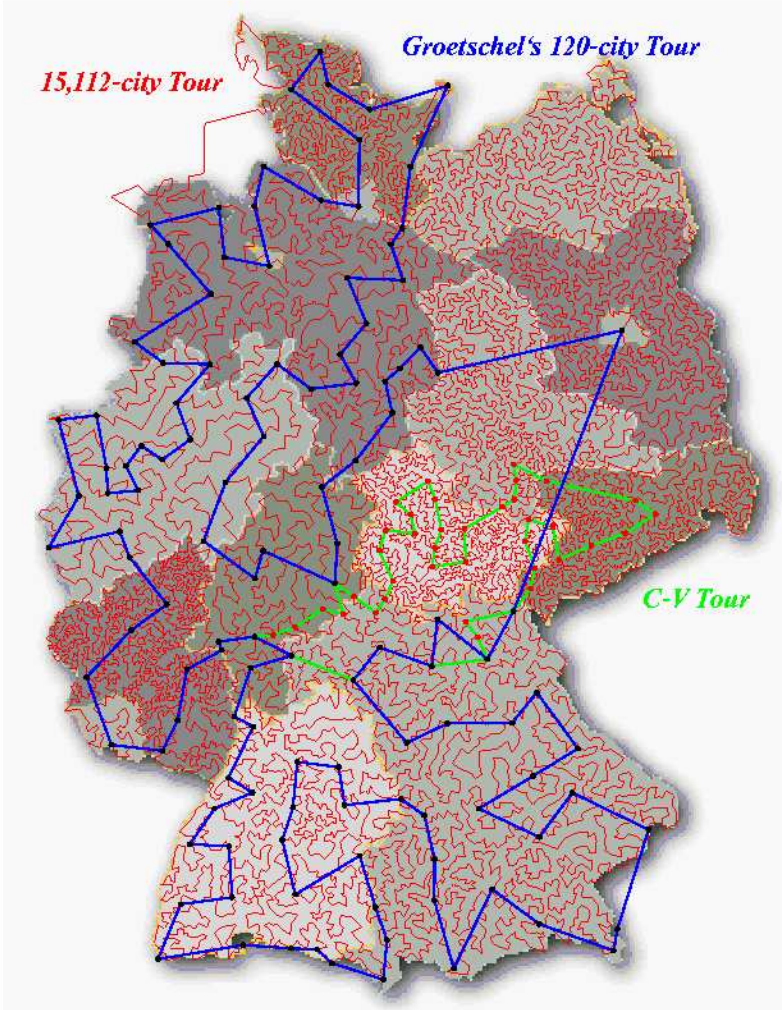
$$\left.\begin{array}{rrcl} \min_x & \sum_{i=1}^{m} c_i x_i & & \\ \forall\, j \leq n & \sum_{i=1}^{m} a_{ij} x_i & \geq & 1 \\ \forall\, i \leq m & x_i & \in & \mathbb{Z}_+ \end{array}\right\}$$



Travelling Salesman problem

A travelling salesman must visit n cities; each city must be visited exactly once.

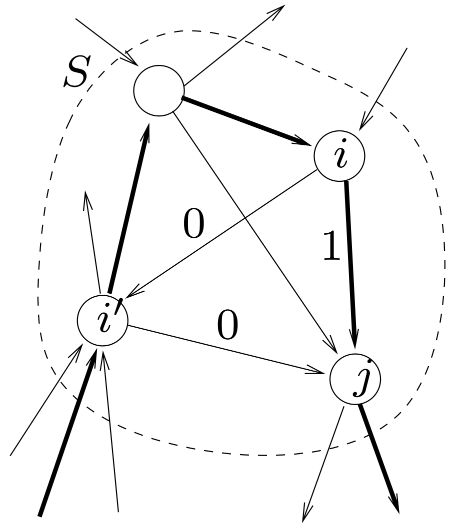
Problem: find the visit order so that the total distance is minimized.



TSP Formulation I

Let c_{ij} be the distance from city i to city j, and $x_{ij} = 1$ if the travelling salesman goes from city i to city j and 0 otherwise.

$$\left.\begin{array}{rl} \min_x & \sum_{i \neq j \leq n} c_{ij} x_{ij} \\ \forall\, i \leq n & \sum_{j \leq n} x_{ij} = 1 \\ \forall\, j \leq n & \sum_{i \leq n} x_{ij} = 1 \\ \forall\, S \subsetneq \{1, \ldots, n\} & \sum_{i \neq j \in S} x_{ij} \leq |S| - 1 \\ \forall i \neq j \leq n & x_{ij} \in \{0, 1\} \end{array}\right\}$$



Exponentially many constraints!

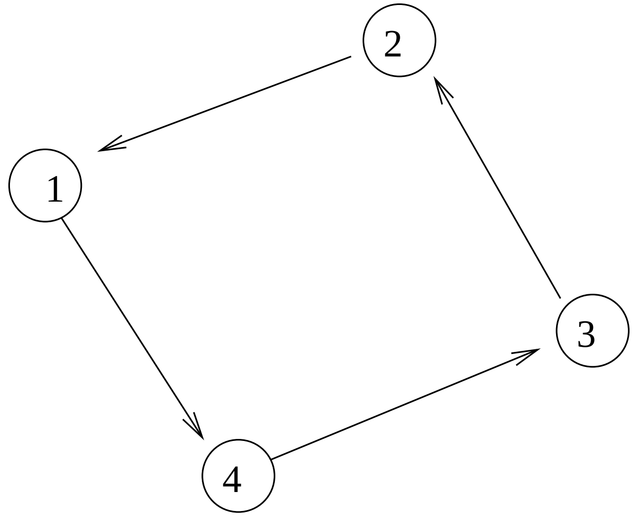
TSP Formulation II

$$\left.\begin{array}{rrcl}
\min & \sum\limits_{i \neq j \leq n} c_{ij} x_{ij} & & \\
\forall\, i \leq n & \sum\limits_{j \leq n} x_{ij} & = & 1 \\
\forall\, j \leq n & \sum\limits_{i \leq n} x_{ij} & = & 1 \\
\forall\, i \neq j \leq n,\ i, j \neq 1 & u_i - u_j + 1 & \leq & (n-1)(1 - x_{ij}) \\
\forall\, i \neq j \leq n & x_{ij} & \in & \{0, 1\} \\
\forall\, i > 2 & u_i & \in & \{2, \ldots, n\} \\
 & u_1 & = & 1.
\end{array}\right\}$$

Only polynomially many constraints

Is this a valid formulation? Does it describe Hamiltonian cycles?

Testing TSP2



- $x_{14} = x_{43} = x_{32} = x_{21} = 1$, all other $x_{ij} = 0$
- set, for example: $u_1 = 1$, $u_2 = 4$, $u_3 = 3$, $u_4 = 2$
- for $(i, j) \in \{(4, 3), (3, 2)\}$, constraints reduce to $u_i - u_j \leq -1$:

  $u_4 - u_3 = 2 - 3 = -1, \qquad u_3 - u_2 = 3 - 4 = -1$ OK

- for all other i, j constraints also valid

Formulations and reformulations

Defn. A *formulation* is a pair (f, X)

Defn. A formulation (h, Y) is a *reformulation* of (f, X) if there is a function $\phi : Y \to X$ such that for each optimum y^* of (h, Y) there is a corresponding optimum $x^* = \phi(y^*)$ of (f, X) and $h^* = f^*$.

Thm. TSP2 reformulates TSP1.

Reformulation proof

Proof. By contradiction, suppose $\exists$ a point (x, u) feasible in TSP2 s.t. x represents two disjoint cycles. Let $C = (V, A)$ be the cycle not containing vertex 1, and let $q = |A| > 0$. If all constraints are satisfied, then arbitrary sums of constraints must also be satisfied. Summing constraints

$$u_j \geq u_i + 1 - (n-1)(1 - x_{ij})$$

over A, since $x_{ij} = 1$ for all $(i, j) \in A$, we obtain

$$\sum_{j \in V} u_j \geq \sum_{i \in V} u_i + q,$$

whence $q \leq 0$, contradicting $q > 0$. Therefore every feasible point in TSP2 represents a cycle of length n in the graph. Since $f \equiv h$, the function ϕ sending each point (x, u) in TSP2 to the corresponding point x in TSP1 is a reformulation.

Exercise 1

Prove that TSP1 reformulates TSP2

(**Hint**: show that given an optimum x^* for TSP1, there exists u^* such that (x^*, u^*) is feasible in TSP2. Why is this sufficient to show that TSP1 reformulates TSP2?)

Solution algorithms

- Exact (provide a guarantee of optimality or ε-optimality for given $\varepsilon > 0$ (in nonlinear continuous problems)

  Simplex Algorithm, Branch and Bound

- Approximation algorithms (provide a guarantee on the solution quality)

  Christofides' TSP Approximation Algorithm

- Heuristic algorithms (do not provide any guarantee, but common sense suggests solution would be good)

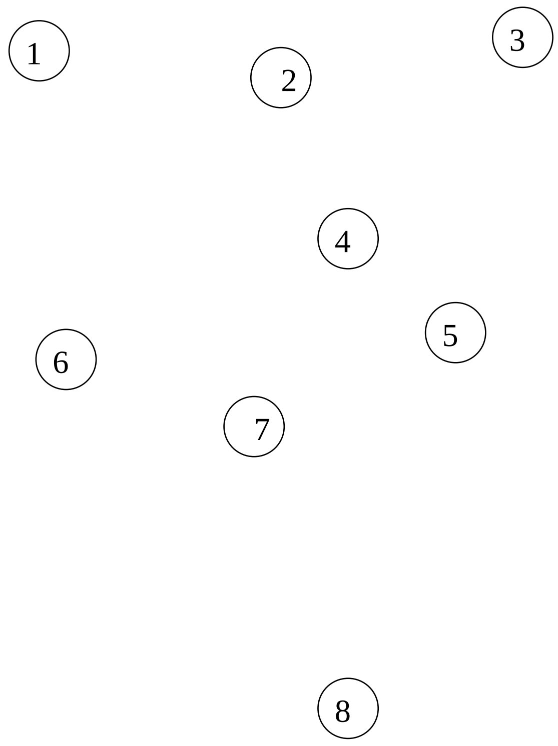
  Variable Neighbourhood Search

Approximation algorithms

- Let $\bar{f}$ be the objective function value at the solution $\bar{x}$ provided by the appr. alg.
- Alg. is a k-*approximation algorithm* for a minimization problem if $\bar{f} \leq kf^*$
- How could we ever prove this without knowing f^*???
- Notation: given an undirected graph $G = (V, E)$ let $\bar{\delta}(v)$ be the set of edges in E adjacent to $v \in V$

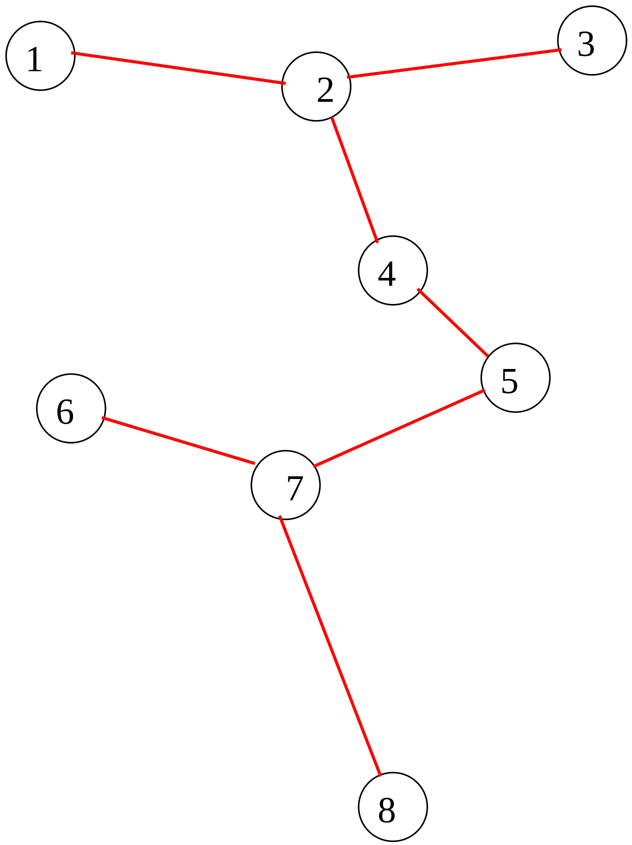
Christofides' TSP Alg. I

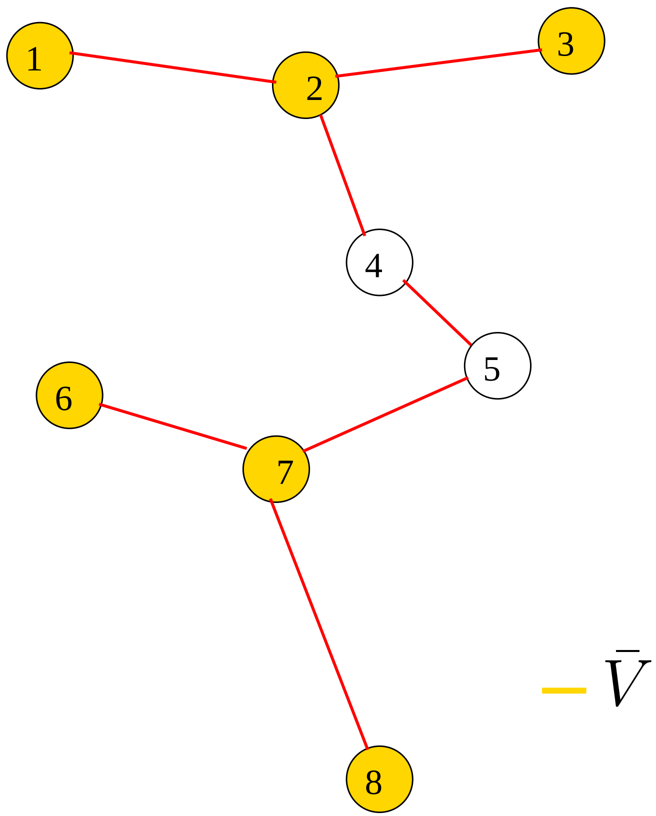
- $\frac{3}{2}$-approximation algorithm for the metric TSP (i.e. distances obey a triangular inequality)
- Consider a complete graph $G = (V, E)$ weighted by $c : E \to \mathbb{R}$, aim to find a "reasonably short" Hamiltonian cycle in G



Christofides' TSP Alg. II

(1) Let $T = (V, F)$ be a spanning tree of G (connected subgraph covering V) of minimum cost

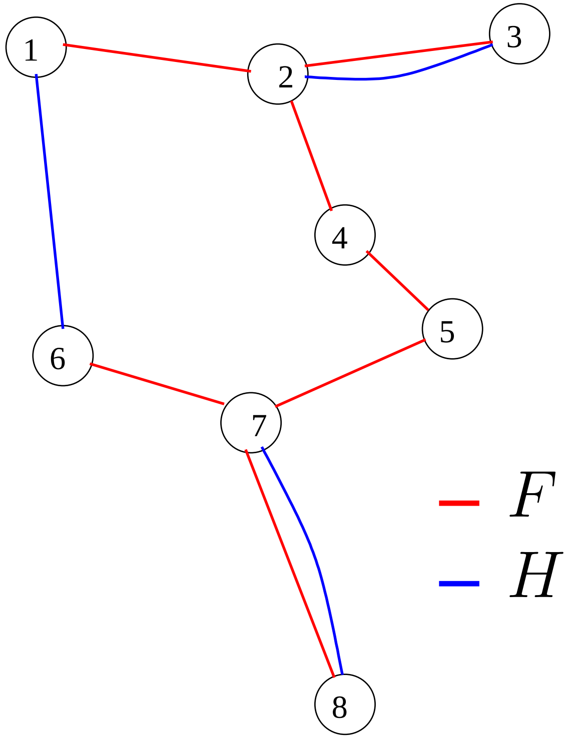




(2) Let $\bar{V} = \{v \in V \mid |\bar{\delta}(v) \cap F| \mod 2 = 1\}$

Christofides' TSP Alg. III

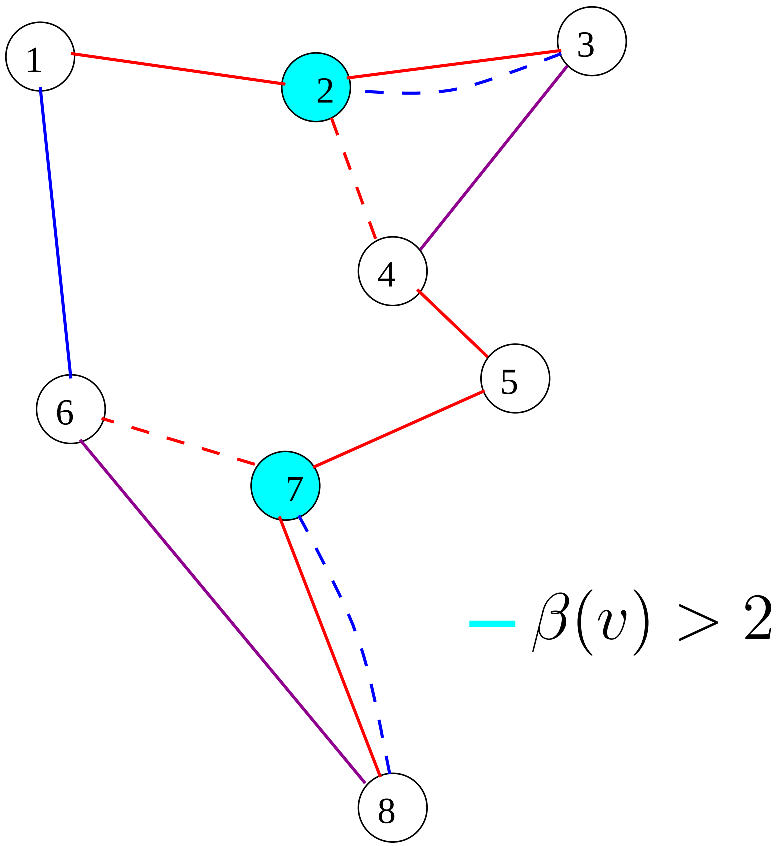
(3) Let $M = (\bar{V}, H)$ be a matching of $(\bar{V}, E(\bar{V}))$ of minimum cost

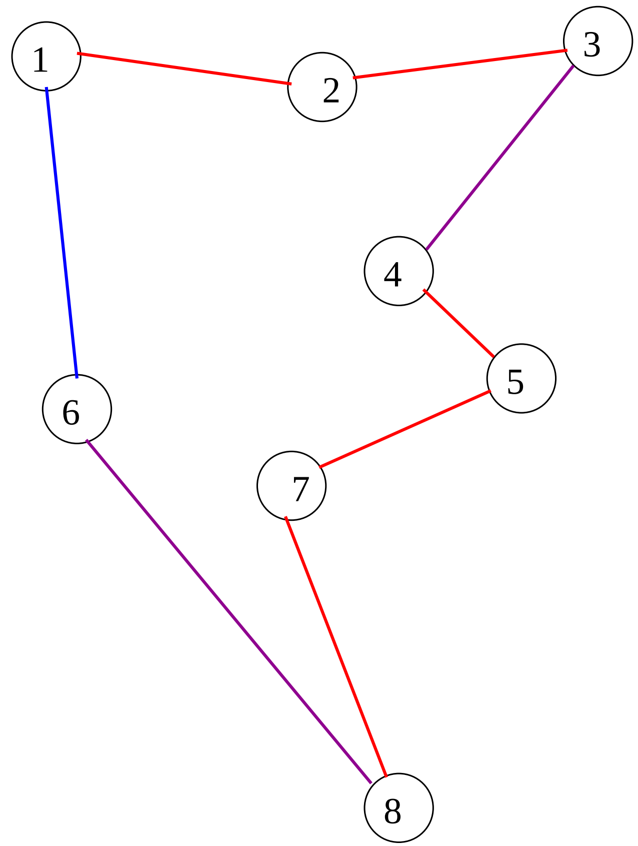


(4) Let $L = F \cup H$, and $K = T \cup M = (V, L)$. This is a Eulerian cycle (i.e. passing through each edge exactly once) because by definition $|\bar{\delta}(v) \cap L| \mod 2 = 0$

Christofides' TSP Alg. IV

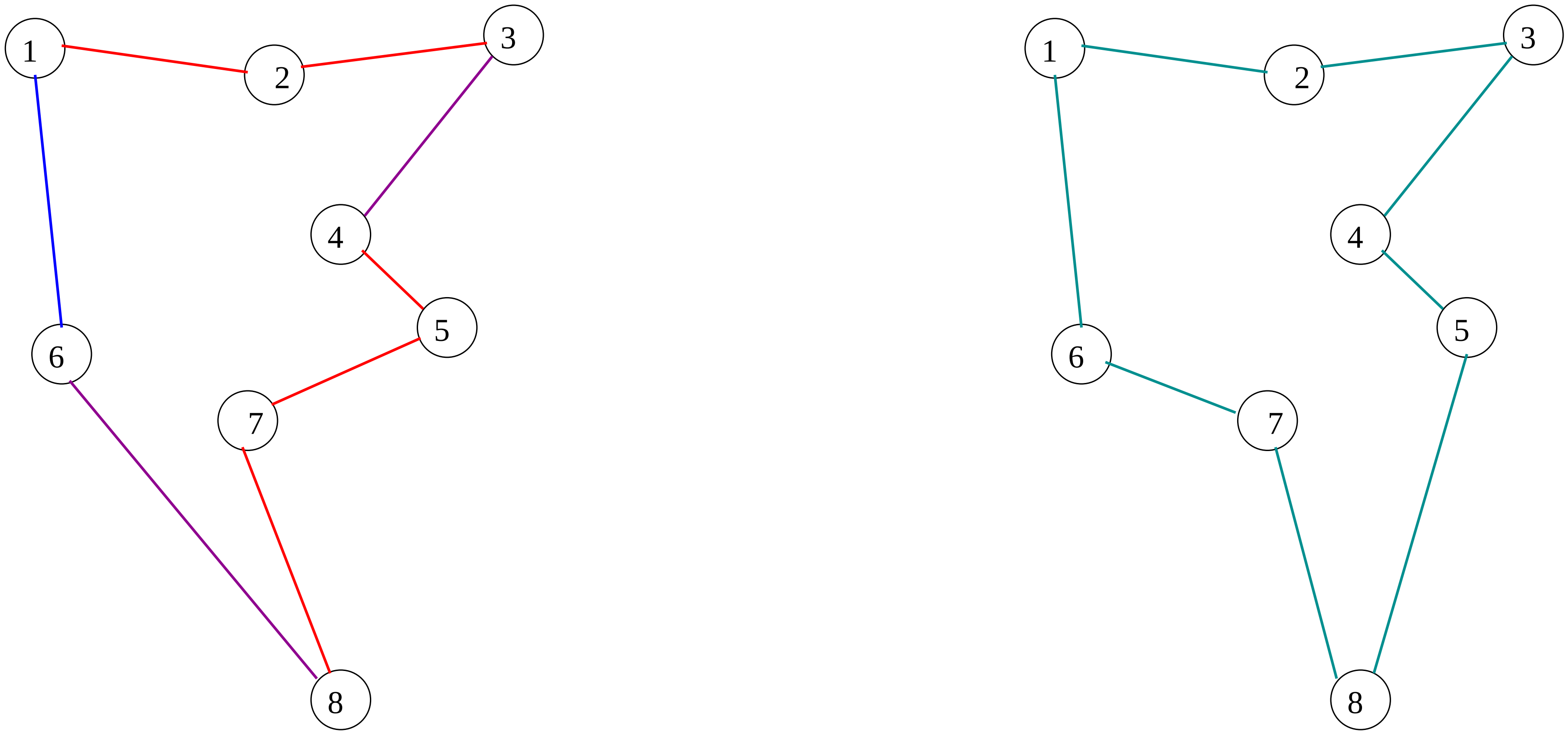
(5) For each v s.t. $\beta(v) = |\bar{\delta}(v) \cap L| > 2$, pick $\frac{\beta(v)}{2} - 1$ distinct pairs of distinct vertices u, w adjacent to v and set $L \leftarrow L \setminus \{\{u,v\},\{v,w\}\} \cup \{u,w\}$





Christofides' TSP Alg. V

The Hamiltonian cycle found with Christofides' approximation algorithm (left) and the optimal one (right)



Christofides' TSP Alg.VI

Lemma. L is a Hamiltonian cycle in G (Exercise 2)

Thm. Let $\bar{f}$ be the cost of L and f^* be the cost of an optimal Hamiltonian cycle. Then $\bar{f} \leq \frac{3}{2} f^*$

Proof. For a set of edges $S \subseteq E$, let $f(S) = \sum_{\{i,j\} \in S} c_{ij}$. Every Hamiltonian cycle (including the optimal one) can be seen as a spanning tree union an edge. Since T is of minimum cost, $f(F) \leq f^*$. On the other hand, each Hamiltonian cycle is also a 2-matching (each vertex is adjacent to precisely two other vertices), and M is of minimum cost, $2f(H) \leq f^*$. Therefore $f(F \cup H) = f(F) + f(H) \leq f^* + \frac{1}{2} f^*$. By the triangular inequality, $f(L) \leq f(F \cup H)$ (why? — exercise 3). □

Christofides' TSP Alg.

- Minimum cost spanning tree: polynomial algorithm
- Minimum cost matching: polynomial algorithm
- Rest of algorithm: polynomial number of steps
- $\Rightarrow$ Polynomial approximation algorithm

Exercise 4

Find a 2-approximation algorithm for the TSP

(**Hint.** Consider the algorithm: (i) let T be a min spanning tree of G (ii) duplicate each edge of T to obtain T' (iii) perform step (5) of Christofides' algorithm on T' to obtain L. Show that L is a Hamiltonian cycle in G of cost $\leq 2f^*$)