

DM865 - Heuristics & Approximation Algorithms

(Marco) (Lore)

Combinatorial problems:

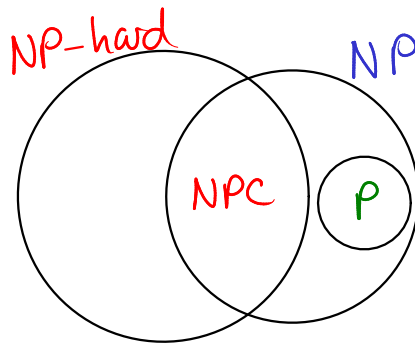
First 2 weeks →

- Traveling Salesman (TSP)
- MAX SAT
- Set Cover
- Knapsack
- Bin packing
- Scheduling

} decision version
∈ **NPC**

Polynomial algorithm:

algo. with running time $O(n^c)$,
for some constant c .



P: The set of decision problems that allow for a poly. algo.

NP: A problem belongs to NP, if solutions can be verified in poly. time.

If any NP-hard problem has a poly. algo.,
then all problems in NPC have poly. algo.s.

Optimal solutions in poly. time for all instances

(1) (2) (3)

- Choose two! $\binom{(2) \text{ \& } (3)}{2}$

Section 1.1

An approximation algorithm comes with a performance guarantee:

Def 1.1: α -approximation algorithm

An α -approximation algorithm for an optimization problem P is a poly. time algo. ALG s.t. for any instance I of P ,

- $\frac{ALG(I)}{OPT(I)} \leq \alpha$, if P is a minimization problem
- $\frac{ALG(I)}{OPT(I)} \geq \alpha$, if P is a maximization problem

Thus, for max. problems, $0 \leq \alpha \leq 1$,
and, for min. problems, $\alpha \geq 1$.

The approximation factor / approximation ratio is

- the smallest possible α (for min. problems)
- the largest possible α (for max. problems)

More precisely, the approx. factor R is

$$R = \inf \{ \alpha \mid \forall I : \frac{ALG(I)}{OPT(I)} \leq \alpha \} \text{ for min. problems}$$

$$R = \sup \{ \alpha \mid \forall I : \frac{ALG(I)}{OPT(I)} \geq \alpha \} \text{ for max. problems}$$

We will cover the rest of Section 1.1 later.

Techniques: (with Set Cover as an example)

- Solve LP and round solution (Sec. 1.3 + 1.7)
- Primal-dual alg.: combinatorial alg.
based on LP formulation (Sec. 1.4 + 1.5)
- Greedy alg. (Sec. 1.6)

Section 2.4: TSP

The Traveling Salesman Problem (TSP)

Input: Weighted complete graph G

$$C_{ij} = C_{ji}, \quad i, j \in V$$

$$C_{ii} = 0, \quad i \in V$$

$$C_{ij} \geq 0, \quad i, j \in V$$

Output: Hamiltonian cycle of min. total weight
Cycle visiting each vertex exactly once.

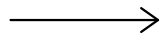
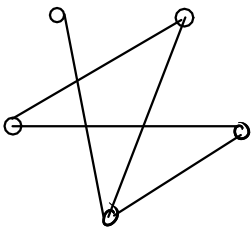
Decision version of TSP:

Does there exist a tour of cost $\leq x$?

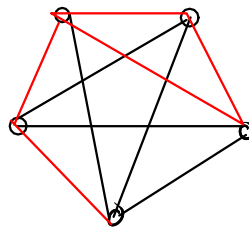
Decision version of TSP is NP-hard

Reduction from Hamilton Cycle:

Ham. cycle



TSP



$$C_{ij} = 1$$

$$C_{ij} = 2$$

\exists ham. cycle

\Leftrightarrow

\exists tour of cost n

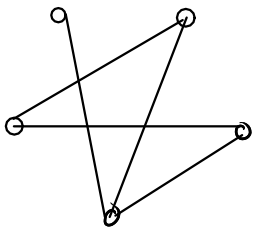
Even worse, no approximation guarantee possible:

Theorem 2.9

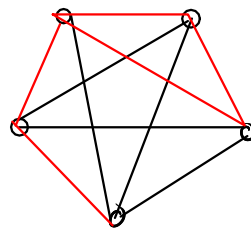
$\forall \alpha > 1$, \nexists α -approx alg. for TSP (unless $P = NP$)

Proof: Reduction from Hamilton Cycle:

Ham. cycle



TSP



$$c_{ij} = 1$$
$$c_{ij} = \alpha n + 1$$

\exists ham. cycle

$\Leftrightarrow \exists$ tour of cost n

$\Leftrightarrow \alpha$ -approx. alg. gives tour of cost $\leq \alpha n$

$\Leftrightarrow \alpha$ -approx. alg. returns a tour with no red edges, i.e., a tour of cost n .

□

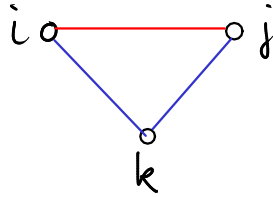
Note: The proof does not require α to be a constant. In fact, it could be 2^n , or any function computable in poly. time.

Thus, we will only consider a special case of TSP:

Metric TSP :

The edge weights satisfy the triangle inequality:

$$c_{ij} \leq c_{ik} + c_{kj}, \text{ for all } i, j, k \in V$$



For metric TSP, the proof of Thm 2.9 does not work (the max. possible cost of the red edges would be 2).

For Metric TSP, we will consider three algorithms:

The Nearest Addition algorithm

2-approx.

The Double Tree algorithm

2-approx.

Christofide's Algorithm

$\frac{3}{2}$ -approx

Nearest Addition (NA)

$u, v \leftarrow$ two nearest neighbors in V

$Tour \leftarrow \langle u, v, u \rangle$

For $i \leftarrow 1$ to $n-2$

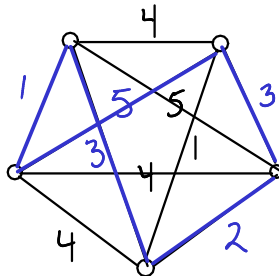
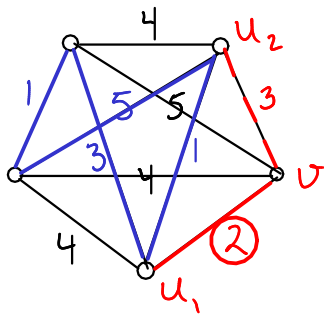
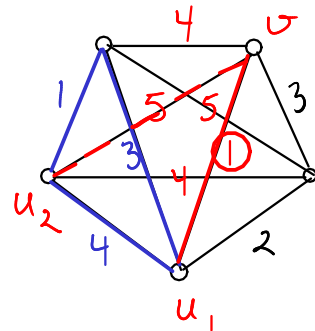
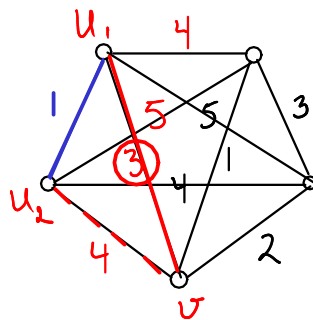
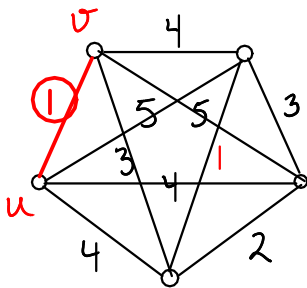
$v \leftarrow$ nearest neighbour of $Tour$

$u_1 \leftarrow$ nearest neighbor of v in $Tour$

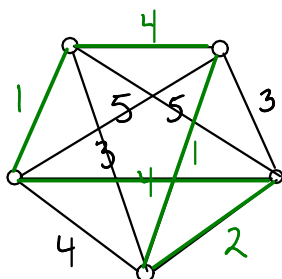
$u_2 \leftarrow u_1$'s successor in $Tour$

Add v to $Tour$ between u_1 and u_2

Ex:



$$C_{NA} = 1 + 3 + 2 + 3 + 5 = 14$$

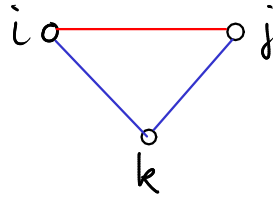


$$C_{OPT} \leq 1 + 4 + 1 + 2 + 4 = 12$$

Metric TSP :

The edge weights satisfy the triangle inequality:

$$c_{ij} \leq c_{ik} + c_{kj}, \text{ for all } i, j, k \in V$$



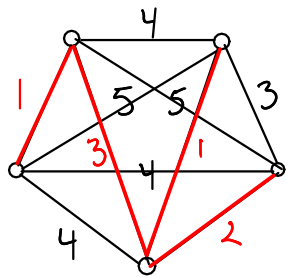
For Metric TSP, Nearest Neighbor is a 2-approx. alg.:

We will prove that

$$(1) \quad c_{NA} \leq 2 \cdot c(MST)$$

$$(2) \quad c(MST) \leq c_{opt} \quad (\text{Lemma 2.10})$$

(1): The solid red edges are exactly those chosen by Prim's Algorithm:

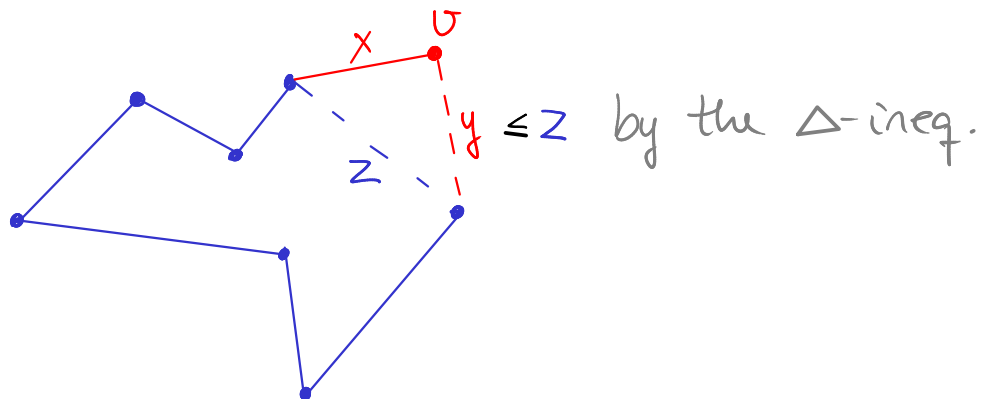


$$C = 1 + 3 + 1 + 2 = 7$$

Thus, the total cost C of these edges is that of a minimum spanning tree:

$$C = c(\text{MST})$$

Adding a new vertex v to the tour, we add two edges and delete one:



Adding v costs

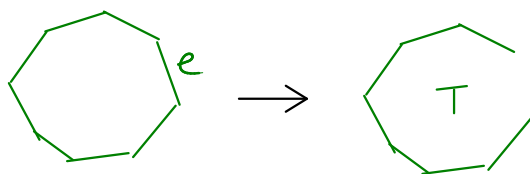
$$x + y - z \leq x + (x + z) - z = 2x$$

where x is the cost of Prim's Alg. in this step

Thus,

$$C_{NA} \leq 2C = 2c(\text{MST})$$

(2): Deleting any edge from a tour, we get a spanning tree:



For any spanning tree T obtained by deleting an edge e from an optimal tour,

$$\begin{aligned} c_{\text{OPT}} &\geq c(T), \text{ since } w(e) \geq 0 \\ &\geq c(\text{MST}) \end{aligned}$$

Now,

$$(1) \ \& \ (2) \Rightarrow c_{\text{NA}} \leq 2 c(\text{MST}) \leq 2 c_{\text{OPT}}$$

This proves:

Theorem 2.11

For Metric TSP, Nearest Addition is a 2-approx. alg.