10/10/14

Approximation algorithms:

Knapsack: aim is to find an algorithm that runs in polynomial time and has a good guaranteed performance.

maximise $p_1x_1 + \cdots + p_nx_n$ s.t. $w_1x_1 + \cdots + w_nx_n \leq W$ $x_j = 0 \text{ or } 1 \ \forall i$

(I) Suppose $P = \max\{p_1, p_2, \cdots, p_n\}$. Then 0/1 knapsack can be solved in $O(n^2 P)$ time.

(II) $\epsilon > 0$ is given and $K = \frac{\epsilon P}{n}$. $\hat{p}_j = \left\lfloor \frac{p_j}{K} \right\rfloor \leq \frac{n}{\epsilon}$

(a) Let x^* = optimal using (p_j) & $\hat{x}$ = optimal using $\hat{p}_j$
We can find $\hat{x}$ in $O(n^3/\epsilon)$ time.

(b) $p(\hat{x}) \geq (1-\epsilon)\, p(x^*)$ — $p(x) = p_1 x_1 + \cdots + p_n x_n$.

(1) Dynamic programming: for $p = 0, 1, 2, \cdots, nP$ ≥ max. profit

$$g_r(p) = \min_{w} \left[\begin{array}{l} \max \; p_1 x_1 + \cdots + p_r x_r \\ \text{s.t.} \quad w_1 x_1 + \cdots + w_r x_r \le w \end{array} \right] \ge p$$

$g_r(p)$ is the smallest knapsack size that can achieve p with items $1, 2, \cdots, r$.

Solve knapsack problem by finding largest p s.t. $g_n(p) \le W$.

Recurrence $g_r(p) = \min \left\{ \underset{x_r = 0}{g_{r-1}(p)}, \; \underset{x_r = 1}{w_r + g_{r-1}(p - p_r)} \right\}$

$n^2 P$ choices for r, p.

(i) $p(x^*) \leq K\hat{p}(x^*) + Kn$ $\qquad \hat{p}_j = \left\lfloor \frac{p_j}{K} \right\rfloor \Rightarrow p_j \leq K(\hat{p}_j + 1)$

$\geq \frac{p_j}{K} - 1$

(ii) $p(x^*) \leq K\hat{p}(x^*) + nK \leq K\hat{p}(\hat{x}) + nK$

$\leq p(\hat{x}) + nK = p(\hat{x}) + \epsilon P \leq p(\hat{x}) + \epsilon p(x^*)$

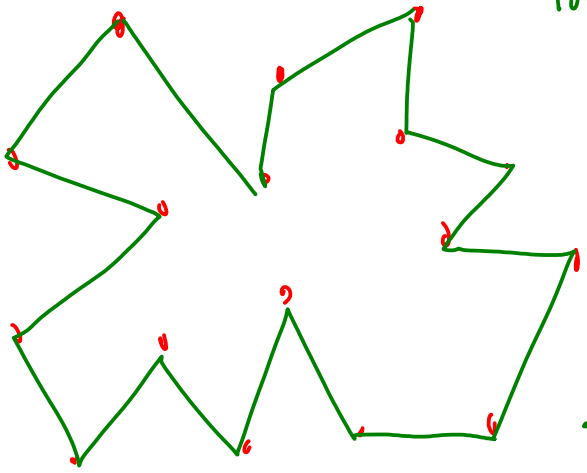
$p(x^*) \geq P$ because we can just put variable $x_j = 1$ where $p_j = P$

Traveling Salesperson Problem: TSP

n cities. d_{ij} = distance from i to j

Problem: find sequence $\pi(1), \pi(2), \ldots, \pi(n)$

that minimizes $\underbrace{d_{\pi(1),\pi(2)} + d_{\pi(2),\pi(3)} + \cdots + d_{\pi(n),\pi(1)}}_{d(T),\ T = \pi(1), \pi(2), \ldots}$



Tour : goes through each city exactly once.

NP-hard

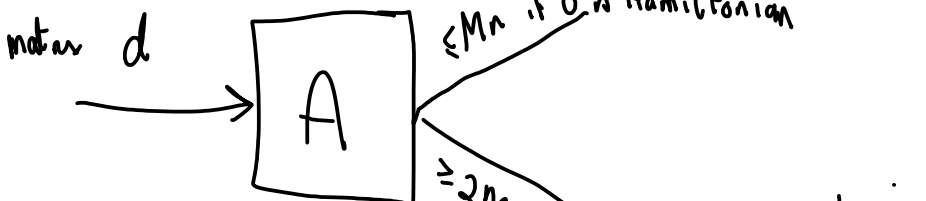
Suppose first that the d_{ij} are arbitrary.

Suppose I have a polynomial time algorithm that finds a tour T such that $d(T) \leq M\, d(T^*)$

Suppose there was such an algorithm A.

Then I can in polynomial time check whether a graph has a Hamilton cycle.

Given graph G let: $d_{ij} = \begin{cases} 1 : (i,j) \text{ is an edge} \\ 2Mn : (i,j) \text{ is not an edge} \end{cases}$



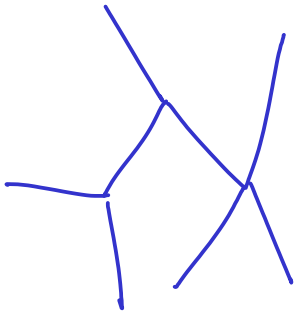
So algorithm would solve
H-cycle problem in poly time.

Suppose now that (d_{ij}) satisfy the triangle inequality.

$$d_{ik} \leq d_{ij} + d_{jk} \qquad \forall\, i, j, k$$

Simple algorithm that produces a tour of at most twice optimum.

① Solve minimum spanning tree problem.



(11) Double all its edges