# TSP

No restrictions on costs implies NP-hard to get an M-approximation for any M.

Reduction from Hamilton cycle.

Given graph G: give cost one to edge in G
give cost Mn+1 to edge not in G.

TSP has tour of length n iff G is Hamiltonian, otherwise minimum tour length is at least (M+1) n.

An M-approximation algorithm would find an H-cycle iff G is Hamiltonian.

We assume then that the triangle inequality holds: $C(i,j) \le C(i,k) + C(k,j)$.

Assume also that $C(i,j) = C(j,i)$.

Fact: a connected graph with all degrees even contains an Euler tour – a walk that crosses each edge exactly once.

Proof

G must contain a cycle C.

G – C has components $C_1, C_2, \cdots, C_r$.

By induction each $C_i$ contains an Euler tour.

We get an Euler tour of G by traversing C and visiting each $C_i$ as we go.

# Tree heuristic

If $T$ is a minimum spanning tree then

$\ell(T) \le \ell(H^*)$. $H^*$ = minimum cost H-cycle.

Double each edge of $T$ to get graph $T^*$.

$T^*$ has an Euler tour $C$.

$$\ell(T^*) = 2\ell(T) \le 2\ell(H^*).$$

Now traverse $C$, short cutting as we go to avoid visiting same vertex twice.

Shortcutting does not increase length and so we obtain a tour $H$ such that

$$\ell(H) \le \ell(T^*) \le 2\ell(H^*)$$

# Christofides heuristic

Let $T^*, H^*$ be as before.

Let $S$ be the set of odd degree vertices in $T^*$ and let $M^*$ be a minimum cost matching of $S$.

Add $M^*$ to $T^*$ and find an Euler tour and short cut. This gives a tour $H$ where

$$\ell(H) \le \ell(T^*) + \ell(M^*)$$
$$\le \ell(H^*) + \frac{1}{2}\ell(H)^* = \frac{3}{2}\ell(H^*).$$

Given $H^*$ we can short cut it to a tour of $S$ which contains two matchings of $S$.

So $\ell(H^*) \ge 2\ell(M^*)$.

# Knapsack problem

Maximize $\sum_{i=1}^{n} p_i x_i$ s.t. $\sum_{i=1}^{n} a_i x_i \le B$

$x_i = 0$ or $1$.

Assume that $\frac{p_1}{a_1} \ge \frac{p_2}{a_2} \ge \cdots \ge \frac{p_n}{a_n}$

## Greedy Algorithm

Find $i$ such that $a_1 + a_2 + \cdots + a_{i-1} \le B < a_1 + a_2 + \cdots + a_i$

Then choose better of $\{a_1, a_2, \ldots, a_{i-1}\}$ & $\{a_i\}$ as the solution. Let $S = a_1 + a_2 + \cdots + a_{i-1}$

We have $\quad OPT \le S + \frac{B-S}{a_i} p_i$ - fractional optimum

So $\quad S \ge \frac{1}{2} OPT$

or $\quad p_i \ge \frac{B-S}{a_i} p_i \ge \frac{1}{2} OPT$

$<1$

Let $\hat{p}_i = \left\lfloor \frac{p_i}{p_{max}} N \right\rfloor$ where $N = \left\lfloor \frac{n}{\epsilon} \right\rfloor$

Solve Knapsack problem with profits $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ via dynamic programming.

Compute $g_r(p) = \min \; a_1 x_1 + \cdots + a_r x_r$

$$\text{s.t. } \hat{p}_1 x_1 + \cdots + \hat{p}_r x_r \geq p$$

$$x_i = 0/1$$

Note that

$$g_r(p) = \min \begin{cases} g_{r-1}(p) \\ a_r + g_{r-1}(p - \hat{p}_r) \end{cases}.$$

$1 \leq r \leq n, \; 0 \leq p \leq nN$

Let $\hat{S}$ solve $(\hat{p})$ and $S^*$ solve $(p)$

$$\sum_{i \in \hat{S}} \hat{p}_i \geq \sum_{i \in S^*} \hat{p}_i \quad \text{and so}$$

$$\frac{p_{max}}{N} \sum_{i \in \hat{S}} \hat{p}_i \geq \frac{p_{max}}{N} \sum_{i \in S^*} \hat{p}_i$$

But

$$\sum_{i \in \hat{S}} p_i \geq \sum_{i \in \hat{S}} \left\lfloor \frac{p_i}{p_{max}} N \right\rfloor \frac{p_{max}}{N} = \frac{p_{max}}{N} \sum_{i \in \hat{S}} \hat{p}_i$$

$$\frac{P_{max}}{N} \sum_{i \in S^*} \left\lfloor \frac{p_i}{P_{max}} N \right\rfloor$$

$$\geq \frac{P_{max}}{N} \sum_{i \in S^*} \left( \frac{p_i}{P_{max}} N - 1 \right)$$

$$\geq \sum_{i \in S^*} p_i - \frac{n P_{max}}{N}$$

$$\geq \sum_{i \in S^*} p_i - \epsilon P_{max}$$

$$\geq \sum_{i \in S^*} p_i - \epsilon \, OPT$$

A function $f : 2^X \longrightarrow \mathbb{R}$ is submodular if for every $A, B \subseteq X$

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B).$$

Example

Simple plant location problem.

$$f(S) = \sum_{i \in S} f_i + \sum_{j} \max\{a_{ij} : i \in S\}$$

$\underbrace{a_{ij}}$ = Value obtained

---

$f$ is monotone increasing if

$$f(B) \geq f(A) \text{ whenever } B \supseteq A.$$

---

Greedy Algorithm

0: $S = \emptyset$

i: $S_i = S_{i-1} \cup \{x\}$ where $x$ maximises over $y$ $f(S_{i-1} \cup \{y\})$

## Theorem

After $k$ steps of Greedy

$$f(S_k) \geq (1 - e^{-1})\, f(S_k^*)$$

where $S_k^*$ maximises $f$ over $k$-sets.

## Proof

Let

$$\Delta(v|T) = f(T \cup \{v\}) - f(T).$$

Note that if $S \supseteq T$ then

$$\Delta(v|S) \leq \Delta(v|T)$$

or

$$\underset{A \cup B}{f(S \cup \{v\})} + \underset{A \cap B}{f(T)} \leq \underset{A}{f(T \cup \{v\})} + \underset{B}{f(S)}$$

Assumed $v \notin S$

If $v \in S$, monotonicity

$$f(S_k^*) \leq f(S_k^* \cup S_i)$$

$$= f(S_i) + \sum_{j=1}^{k} \Delta(v_j^* \mid S_i \cup \{v_1^*, \ldots, v_{j-1}^*\})$$

$$\leq f(S_i) + \sum_{j=1}^{k} \Delta(v_j^* \mid S_i)$$

$$\leq f(S_i) + \sum_{j=1}^{k} (f(S_{i+1}) - f(S_i))$$

$$= f(S_i) + k(f(S_{i+1}) - f(S_i))$$

This implies that

$$f(S_k^*) - f(S_{i+1}) \leq \left(1 - \frac{1}{k}\right)(f(S^*) - f(S_i))$$

and so

$$f(S_k^*) - f(S_k) \leq \left(1 - \frac{1}{k}\right)^k f(S^*) - \underbrace{f(\emptyset)}_{0}$$

$\square$