Traveling Salesperson Problem (TSP)

Input: Complete graph $G=(V,E)$ (so $E=\binom{V}{2}$) with distance $d: E \to \mathbb{R}^+$.

Output: Tour $T=(v_1,\dots,v_n,v_1)$ s.t. ① every vertex (except v_1) is visited exactly once. ($n=|V|$)

② $d(T)=\sum_{i=1}^{n} d(v_i,v_{i+1})$ is minimized (define $v_{n+1}=v_1$)

Tour: sequence of vtcs $(u_1 \dots u_k)$ s.t. $(u_i,u_{i+1})\in E$ $\forall i \in [k-1]$, and $u_k=u_1$

Many variants

- complete vs general graph
- undirected vs directed graph
- visit exactly once vs visit at least once.
- restriction on distances (e.g., triangle inequality.
  can $d(i,k) > d(i,j)+d(j,k)$?)

If you care about poly-time approximation, then these differences matter. (roughly 3 really different versions)

For today, any version is fine.

Thm (any version) TSP is NP-hard □

Naive algo: Fix v_1. Try every permutation $(v_1,v_2,\dots,v_n)$.

Running time: $(n-1)!\,\mathrm{poly}(n) = (n/e)^n \cdot \mathrm{poly}(n)$

Will see: $2^n \cdot \mathrm{poly}(n)$-time algo.

Fix arbitrary $v_1 \in V$.

Defining Table. $\forall \{v_1\} \subseteq S \subseteq V$ and $v \in S$, we want to define T s.t.

$$T[S,v] := \min_{\substack{\text{walk } p=(v_1,\dots,v) \\ \text{that visits each vtx} \\ \text{in } S \text{ exactly once.}}} d(p) \quad (*)$$

Sequence $(u_1 \dots u_k)$ of vtcs s.t. $(u_i, u_{i+1}) \in E$ $\forall i \in [k-1]$

Recurrence Relation. Base case: $\forall S = \{u, v_1\}$ with $u \neq v_1$: $T[S,u] = d(v_1,u)$

$\forall \{v_1\} \subseteq S \subseteq V$ with $|S|>2$ and $v \in S \setminus \{v_1\}$

$$T[S,v] = \min_{v' \in S \setminus \{v,v_1\}} \left(T[S \setminus \{v\}, v'] + d(v',v) \right). \quad (**)$$

Final answer: $\min_{v \in V \setminus \{v_1\}} \left(T[V,v] + d(v,v_1) \right)$

Correctness.

Lemma. $T[S,v]$ defined by $(**)$ satisfies $(*)$.

Pf. Induction on $|S|$. If $|S| = \{u, v_1\}$ then $T[S,u] = d(v_1,u)$.

Fix S and $v \in S$ ($|S|>2$ and $v \neq v_1$). If the statement is true for every T with $|T|<|S|$,

$$\min_{\substack{\text{walk } p=(v_1,\dots,v) \\ \text{that visits each vtx} \\ \text{in } S \text{ exactly once.}}} d(p) = \min_{\substack{\text{walk } p=(v_1,\dots,v',v) \\ \text{that visits each vtx} \\ \text{in } S \text{ exactly once.}}} d(p)$$

just specifying second-last vertex.

$$= \min_{\substack{\text{walk } p=(v_1,\dots,v',v) \\ \text{that visits each vtx} \\ \text{in } S \text{ exactly once.}}} \left(d(v_1,\dots,v') + d(v',v)\right) = \min_{\substack{v'\in S \\ v'\neq v_1,v}} \min_{\substack{\text{walk } p'=(v_1,\dots,v') \\ \text{that visits each vtx} \\ \text{in } S\setminus\{v\} \text{ exactly once.}}} \left(d(v_1,\dots,v') + d(v',v)\right)$$

induction hypothesis

$$= \min_{\substack{v'\in S \\ v'\neq v_1,v}} \left(T[S\setminus\{v\}, v'] + d(v',v)\right) = \min_{v'\in S\setminus\{v_1,v\}} T[S\setminus\{v\}, v'] + d(v',v)$$

$\square$

So $\min_{v\in V\setminus\{v_1\}} \left(T[V,v] + d(v,v_1)\right)$ is the length of the optimal tour.

Running time: (# of entries we need to fill) $\leq 2^n \cdot n$.

Each entry can be computed in poly(n)-time.

So total running time $= 2^n \cdot \text{poly}(n)$

Space: $O(2^n \cdot n)$:(if you restrict to poly(n)-space,

$4^n \cdot n^{O(\log n)} = 2^{2n+O(\log^2 n)}$ is the best runtime

Set Cover

set (collection) of subsets of U.
(i.e., each $S \in \mathcal{S}$ satisfies $S \subseteq U$.

Input: "Set system" $(U, \mathcal{S})$.

Output: Subcollection $\mathcal{S}' \subseteq \mathcal{S}$ s.t. $\bigcup_{S \in \mathcal{S}'} S = U$.

Goal: minimize $|\mathcal{S}'|$.

(Seen as "contractor problem" in 376.
proved NP-hardness there.)

Let $n = |U|$, $m = |\mathcal{S}|$.

$2^m \cdot \mathrm{poly}(n,m)$-time is easy.

We can also get $2^n \cdot \mathrm{poly}(n,m)$-time algo. using DP.

$\forall S \subseteq U$ and $k \in [m]$,

$$T[S,k] = \begin{cases} 1 & \text{if } \exists S_1, \dots, S_k \in \mathcal{S} \text{ s.t. } S \subseteq \bigcup_{i=1}^{k} S_i. \\ 0 & \text{o.w.} \end{cases}$$

Then, $T[\emptyset, 0] = 1$ and

$$T[S,k] = 1 \iff \exists T \in \mathcal{S} \text{ s.t. } T[S \setminus T, k-1] = 1.$$

Answer: smallest k s.t. $T[U,k] = 1$.

Running time: $2^n \cdot \mathrm{poly}(n,m)$.