TSP

Tours: \(1 \rightarrow \pi(1) \rightarrow \pi(0) \rightarrow \ldots \rightarrow \pi(n) \rightarrow 1\)

\(\pi\) is a permutation.

# Solutions = \((n-1)!\)

D. P. solves problem in \(O(n^2 2^n)\)

\[f(x, S) = \min_{\pi \in \mathcal{P}[n]} \text{length path} \]

\(4 \in S \subseteq [n] \quad \forall x \in S\)

\[= \min_{\pi \in \mathcal{P}[n]} c(2, x) + f(2, S \setminus \{x\})\]

Min length tour =

\[
\min_{x} f(x, [n]) + c(x, x_1)
\]

\[\sum_{k=3}^{n-2} (k-1)(k-2)(n-3) = \sum_{k=3}^{n-2} (n-3)(n-2)(n-1) \]

\[= (n-1)(n-2) \sum_{k=3}^{n-2} (k-3) \]

\[2^{n-3}\]
Possible Cost Sequence

4 8 4 8 4 8
3 3 3 3 3 3
2 4 2 4 2 4

---

\[ \text{Net Present Value} \], \quad F_0 < a < 1, \quad \text{cost of } c_1, c_2, c_3, \ldots \]

Discounted cash flow is

\[ c_1 + \alpha c_2 + \alpha^2 c_3 + \ldots + \alpha^{n-1} c_n \]
Dynamic Programming with Infinite horizon

Production problem

$N$ states

Start somewhere. Jump around.

You have to choose an infinite sequence to follow.

An optimal strategy says when I am at $i$ go to $T(i)$ for some $T: [N] \to [N]$.

Strategies is $N^N$. This is not per a permutation.
Evaluate Strategy Ti.

\[ y_n(i) = \text{discounted cost, starting from state } i \text{ at time } 0. \]

\[ y_n(i) = C(i, \pi(i)) + \alpha y_{n-1}(\pi(i)) \]

**Example**

\[ y_1 = 4 + \frac{1}{2} y_2 = 10 \]
\[ y_2 = 6 + \frac{1}{2} y_2 = 12 \]
\[ y_3 = 5 + \frac{1}{2} y_2 = 11 \]
\[ y_4 = 7 + \frac{1}{2} y_4 = 14 \]
Costs $2 + \frac{1}{2} y_+ = 9 < 10$

Test for optimality

$y_+ = \min_j \{ C(i,j) + \alpha y_j \}$

Claim: $\Pi$ is optimal iff \( y_+ \) holds.
Optimality Condition

\[ Y_0 = \min_j \ c(i, j) + \alpha y_j \]

Claim

(1) If \( \oplus \) holds and \( \hat{\pi} \) is any other policy with value \( \hat{y}_0 \), then \( y_i \leq \hat{y}_i \) for all \( i \).

\[ \hat{y}_i = \alpha(i, \hat{\pi}(i)) + \alpha y_{\hat{\pi}(i)} \]

\[ \varepsilon_i = y_i - \hat{y}_i \geq \varepsilon_i \leq \varepsilon_i \]

\[ y_i \leq c(i, \hat{\pi}(i)) + \alpha y_{\hat{\pi}(i)} \]

\[ \Rightarrow \]

\[ \varepsilon_i \geq 0 \]
(11) If $\not\exists \delta$ does not hold then we can improve all $\pi$:

Let $I = \{i : y_i > C(\hat{\pi}(i), y) = \min_i \frac{1}{2} [C(i, a) + a y_i] \}

\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\text{C}(\hat{\pi}(1), y) & 8 & 10 & 12 & 14 & 16 & 18 & 20 & 22 & 24 \\
\text{C}(\hat{\pi}(2), y) & 10 & 12 & 14 & 16 & 18 & 20 & 22 & 24 & 26 \\
\text{C}(\hat{\pi}(3), y) & 12 & 14 & 16 & 18 & 20 & 22 & 24 & 26 & 28 \\
\text{C}(\hat{\pi}(4), y) & 14 & 16 & 18 & 20 & 22 & 24 & 26 & 28 & 30 \\
\text{C}(\hat{\pi}(5), y) & 16 & 18 & 20 & 22 & 24 & 26 & 28 & 30 & 32 \\
\text{C}(\hat{\pi}(6), y) & 18 & 20 & 22 & 24 & 26 & 28 & 30 & 32 & 34 \\
\text{C}(\hat{\pi}(7), y) & 20 & 22 & 24 & 26 & 28 & 30 & 32 & 34 & 36 \\
\text{C}(\hat{\pi}(8), y) & 22 & 24 & 26 & 28 & 30 & 32 & 34 & 36 & 38 \\
\text{C}(\hat{\pi}(9), y) & 24 & 26 & 28 & 30 & 32 & 34 & 36 & 38 & 40 \\
\text{C}(\hat{\pi}(10), y) & 26 & 28 & 30 & 32 & 34 & 36 & 38 & 40 & 42 \\
\end{array}

New policy $l \rightarrow \hat{\pi}(l)$, $l \in I$
Combinatorial Optimization

Shortest Path
- Assignment Problem
- Matroids

Shortest Path

\[ D = (V, E) \text{ is a digraph.} \]
\[ V = [n], \quad P = \{ \text{paths in } D \} \]
\[ l : P \to \mathbb{R}, \quad l(P) = \text{"length" of } P \]

Initially assume \( l((p)) = l_{\text{rg}}(P) = \sum_{e \in P} l(e) \)
Assume \( l(e) \geq 0 \) for all \( e \).
Problem: Find a path of minimum length from $d$ to every other vertex.

Dijkstra's Algorithm

For $i = 1, 2, ..., n$ do
\[ d(0) = 0, \quad d(i) = \infty, \quad s_i = \emptyset \]
for $i = 2, ..., n$ do
\[ d(i) = \min(d(i), d(j) + w(j, i)) \]
\[ s_i = s_{j(i)} \]
\[ d(0) = \text{min}(d(0), d(i)) \]
\[ V \subseteq S \]
\[ d(0) = \text{min} \]
For a path of form $d \rightarrow i \rightarrow \cdots \rightarrow n$
Claim: On termination, \( d(v) = \) length of shortest path.

Suppose \( P \) is any other path from \( s \) to \( v \).

\[ P = s, x_1, x_2, \ldots, x_r = v \]

\[ (P) \geq (P(\omega)) = d(\omega) = d(v) \]

Suppose \( \omega \) is added at step \( j \).