Networks: Lecture 2

Amedeo R. Odoni

November 20, 2006

Outline

- **Generic heuristics for the TSP**
- **Euclidean TSP: tour construction, tour improvement, hybrids**
- **Worst-case performance**
- **Probabilistic analysis and asymptotic result for Euclidean TSP** [*Separate handout*]
- **Extensions**

- ***Reference: Sections 6.4.5-6.4.13 + Handouts***

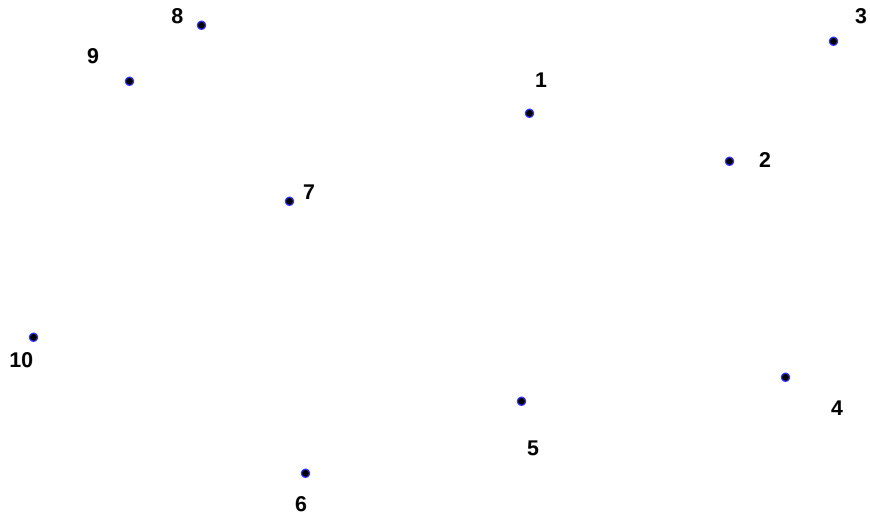
Node Covering (TSP, VRP, et al)

- **Huge literature, endless applications**
- **Traveling Salesman Problem (TSP) is the prototypical "hard" problem**
- **Some applications:**
  - _ Routing of all kinds
  - _ Job shop scheduling
  - _ Vehicle routing problem (VRP)
  - _ Dial-a-ride problem (DARP)
  - _ Electronics industry
  - _ Biotechnology
  - _ Air traffic control
  - _ Genomics

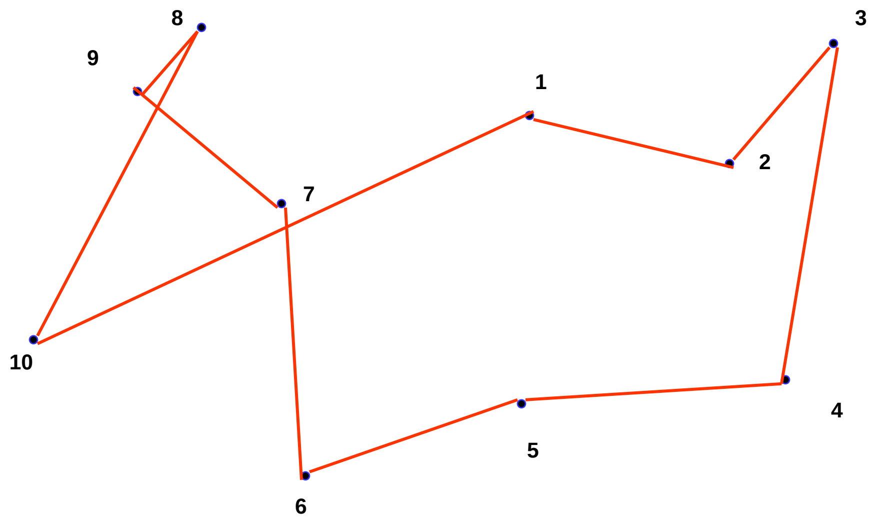
Solving the TSP

- **Best existing exact algorithms can solve optimally problems with up to 15,000 points (as of 2001)**
- **Numerous heuristic approaches for good solutions to MUCH larger problems**
- **For practical purposes, heuristics are very powerful. A classification:**
  - _ Tour construction
  - _ Tour improvement
  - _ Hybrid
- **Analysis of heuristics:**
  - _ Worst case
  - _ Asymptotic
  - _ Empirical
  - _ Probabilistic

Heuristics: Euclidean TSP



The Nearest Neighbor Heuristic

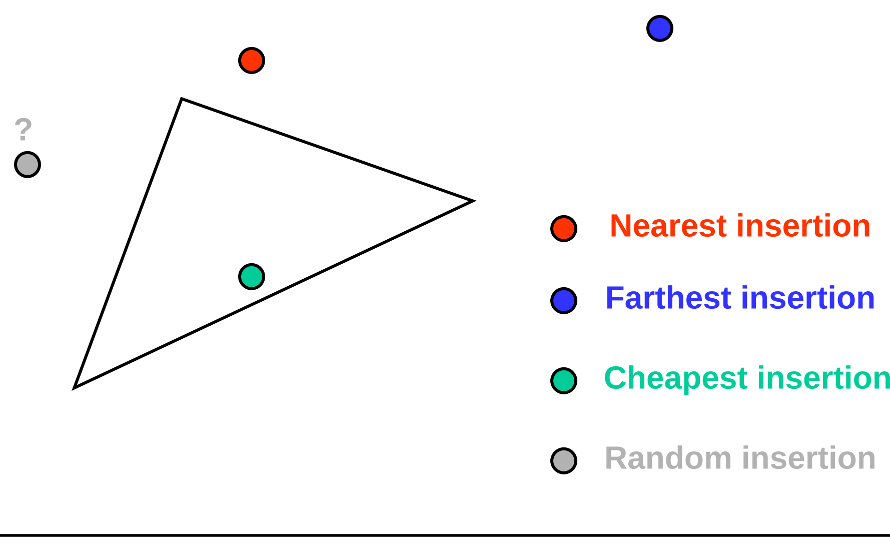


Performance: Nearest Neighbor

$$\frac{L(NEARNEIGHBOR)}{L(TSP)} \leq \frac{1}{2}\lceil \log_2 n \rceil + \frac{1}{2}$$

- Poor performance in practice (+20%)
- Can be improved through refinements (e.g., “likely subgraph”)

Insertion Heuristics



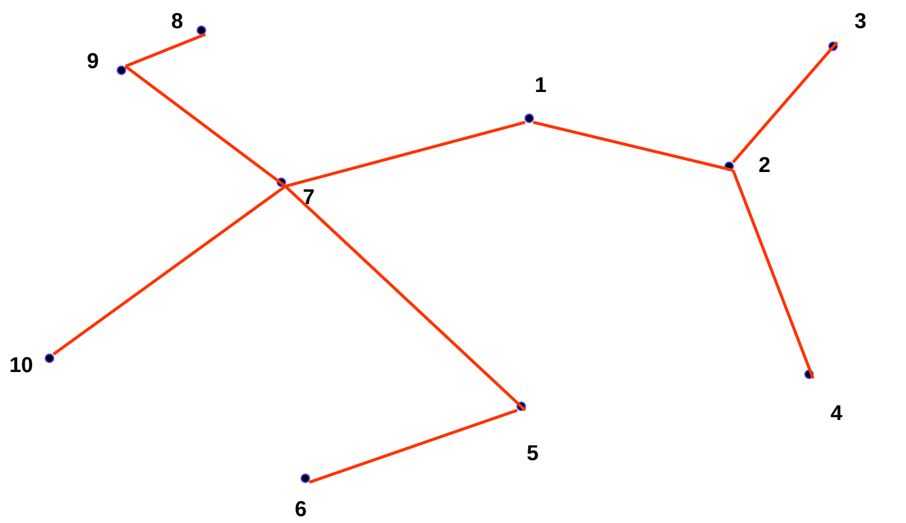
Worst-case Performance: Insertion Heuristics

- $$\frac{L(RANDOM\ INSERT)}{L(TSP)} \le \lceil \log_2 n \rceil + 1$$
- $$\frac{L(NEAR\ INSERT)}{L(TSP)} < 2$$
- $$\frac{L(FAR\ INSERT)}{L(TSP)}$$ => Unknown
- $$\frac{L(CHEAP\ INSERT)}{L(TSP)} < 2$$

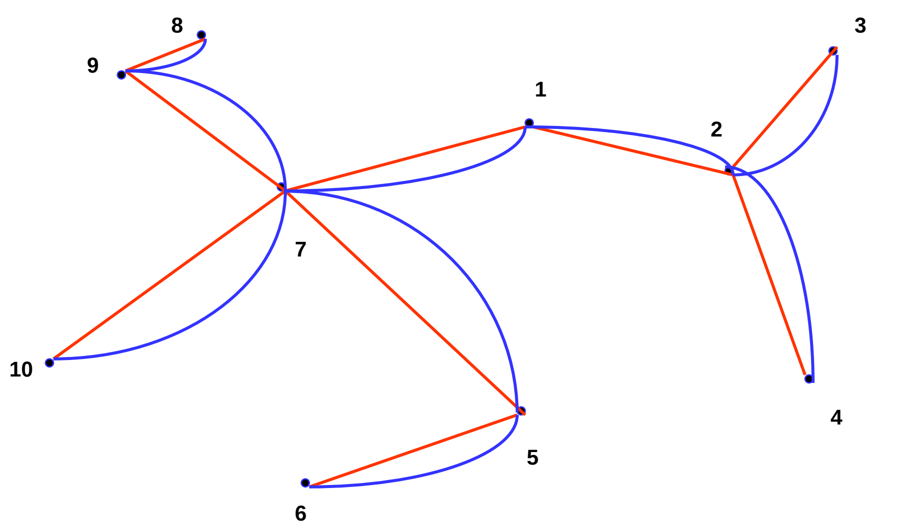
Empirical Performance: Insertion Heuristics

- In practice "Farthest Insertion" and "Random Insertion" (+9%, +11%) seem to perform better than "Cheapest" and "Nearest" (+16%, +19%)
- Can be further refined (e.g., though the Convex Hull heuristic)

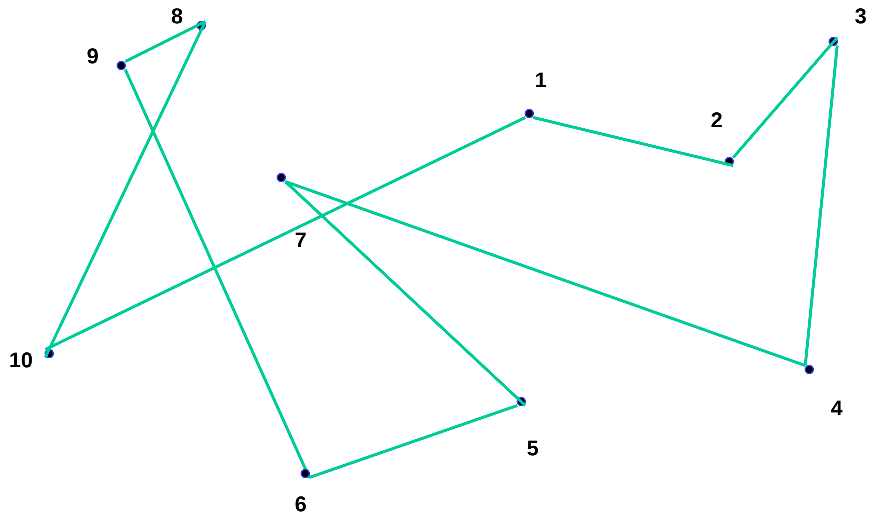
The MST Heuristic for the TSP



Merging with a second copy of the MST



Improve Solution by Skipping Points Already Visited



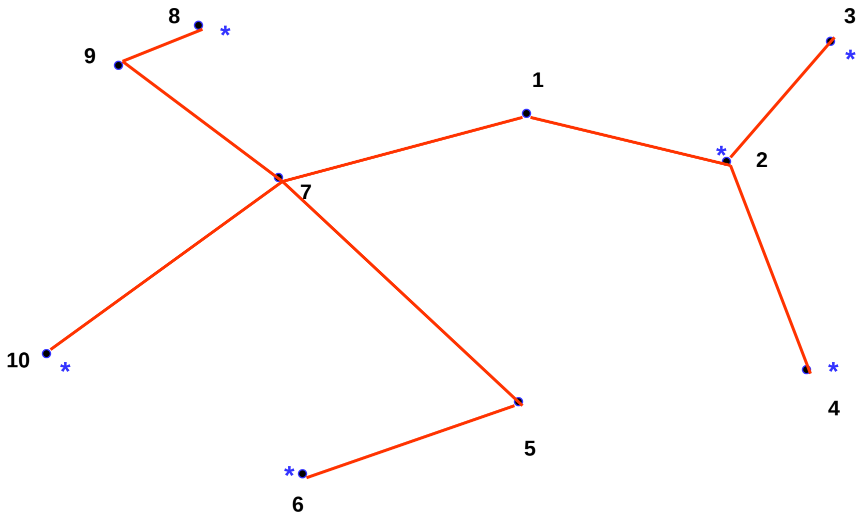
Worst-case Performance: MST Heuristic for TSP

L(MST) ≤ L(TSP-(longest edge of TSP)) < L(TSP)

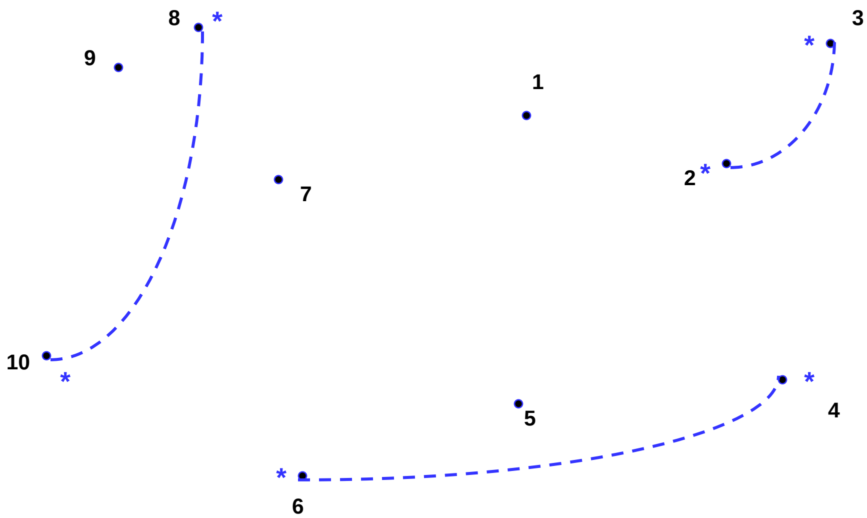
=> *L(MST-TOUR) = 2*L(MST) < 2*L(TSP)*

=> $$\frac{L(MST-TOUR)}{L(TSP)} < 2$$

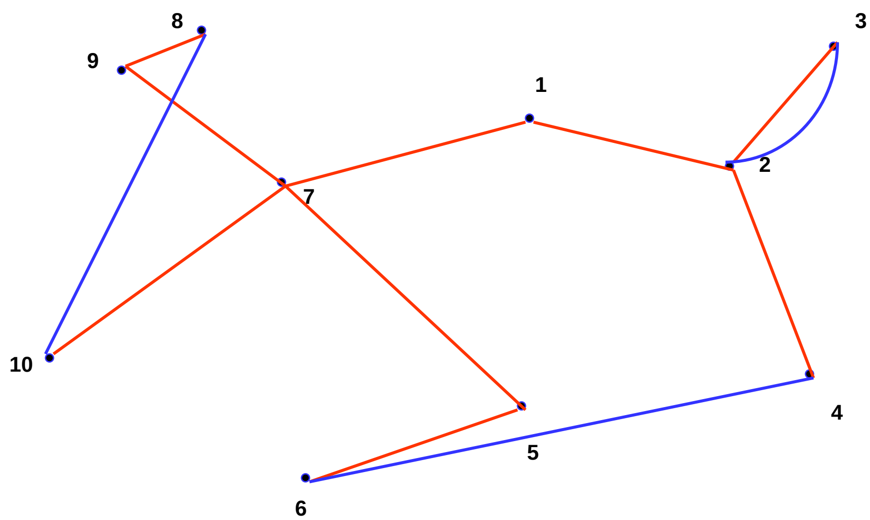
The Christofides Heuristic: Step 1



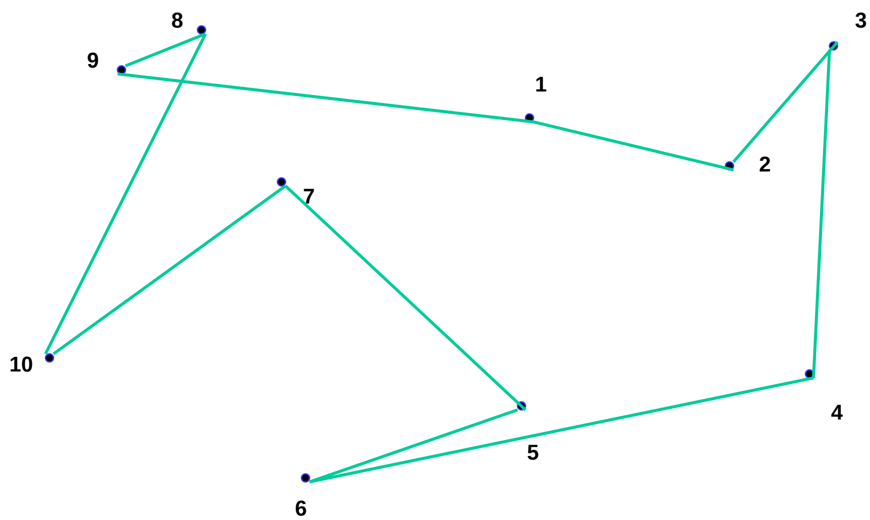
The Christofides Heuristic: Step 2



The Christofides Heuristic: Step 3



Improve Solution by Skipping Points Already Visited



Worst-case Performance: The Christofides Heuristic

- *L(CHRISTOFIDES) = L(MST) + L(M)*

- *But,* *L(MST) < L(TSP)*

and *L(M) ≤ L(M') ≤ L(TSP) / 2*

(*M'* = minimum length pairwise matching of odd- degree nodes of MST using only links that are part of TSP)

Worst-case Performance: The Christofides Heuristic

L(CHRISTOFIDES) = L(MST) + L(M)

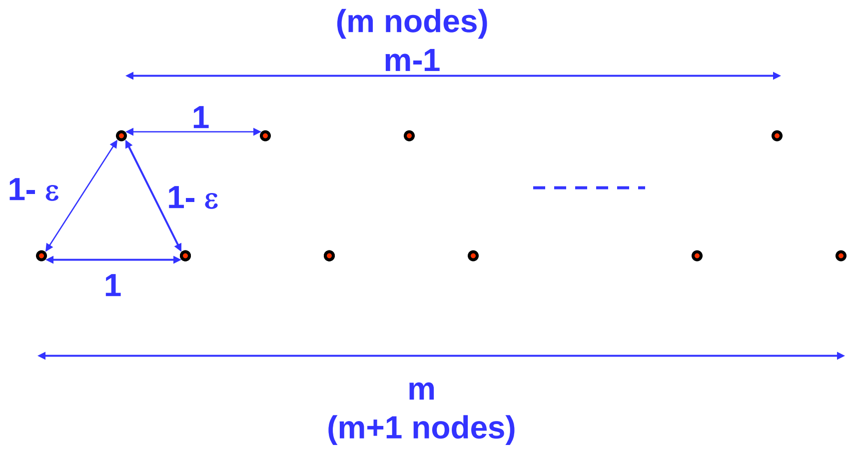
But, *L(MST) < L(TSP)*

and *L(M) ≤ L(M') ≤ L(TSP) / 2*

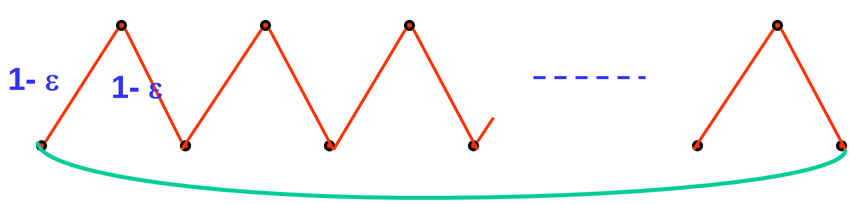
(*M'* = minimum length pairwise matching of odd-degree nodes of MST using only links that are part of TSP)

=> $$\frac{L(CHRISTOFIDES)}{L(TSP)} < \frac{3}{2}$$

A Worst-Case Example for the Christofides Heuristic

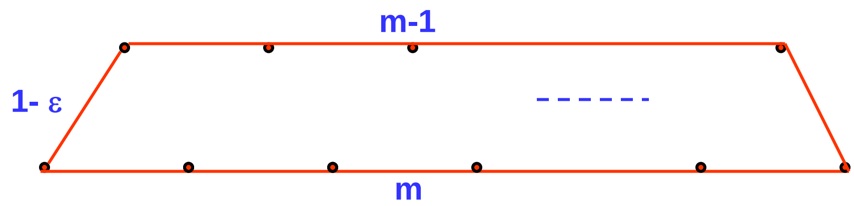


A Worst-Case Example for the Christofides Heuristic (2)



L(Christofides) = 2*m*(1-ε) + m ≈ 3*m

A Worst-Case Example for the Christofides Heuristic (3)

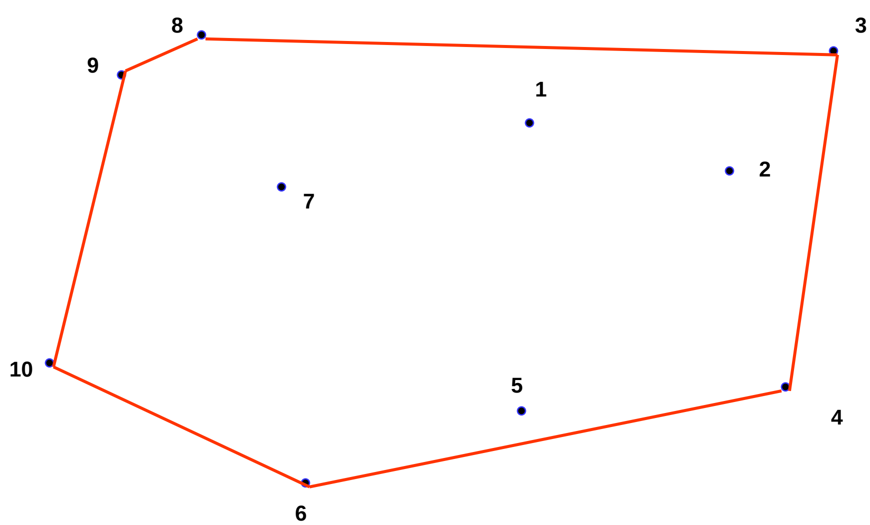


L(TSP) = m + m – 1 + 2*(1 – ε) ≈ 2*m + 1

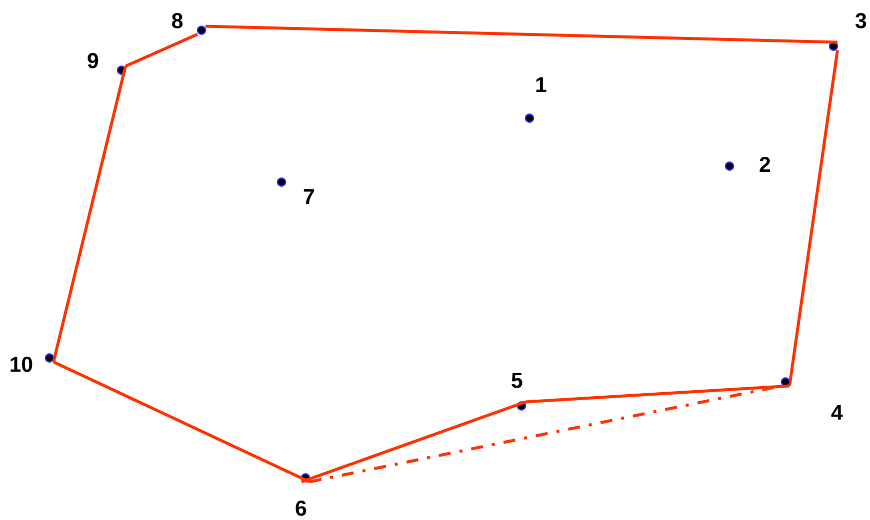
Therefore:

$$\frac{L(CHRISTOFIDES)}{L(TSP)} \approx \frac{3m}{2m+1} \rightarrow \frac{3}{2} \quad as \quad m \rightarrow \infty$$

The Convex Hull Heuristic: Euclidean Plane



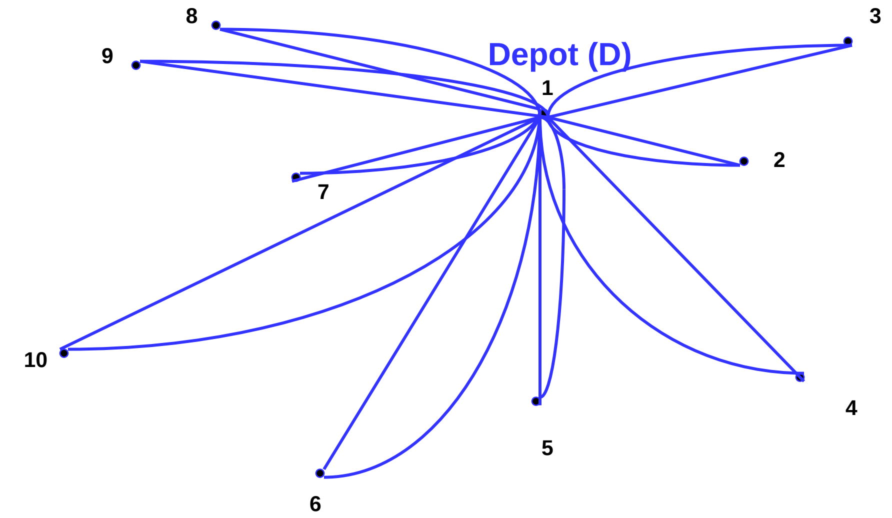
Adding New Points



Convex Hull Heuristic (Euclidean TSP)

- Optimal TSP tour cannot intersect itself
- Therefore, points on the convex hull must appear in same order on optimal TSP tour
- Provides good starting point; for instance, improves insertion heuristics by 2-3%, on average

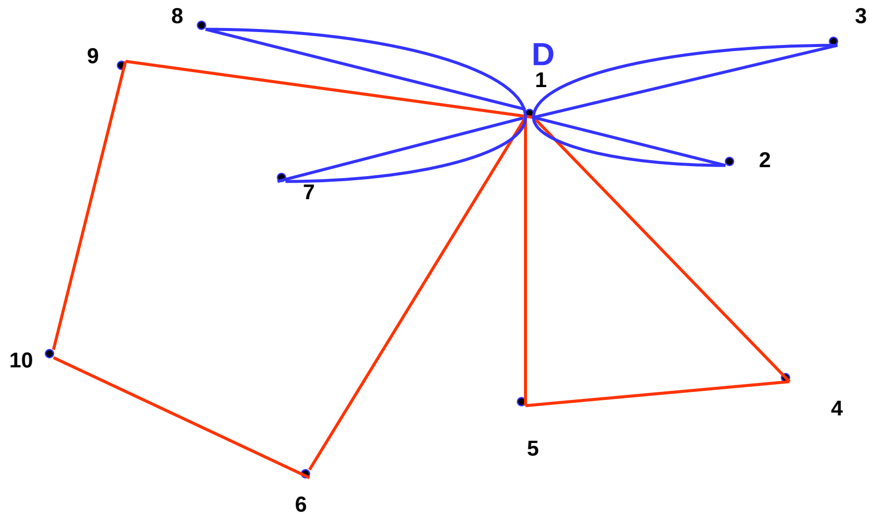
The Savings Algorithm



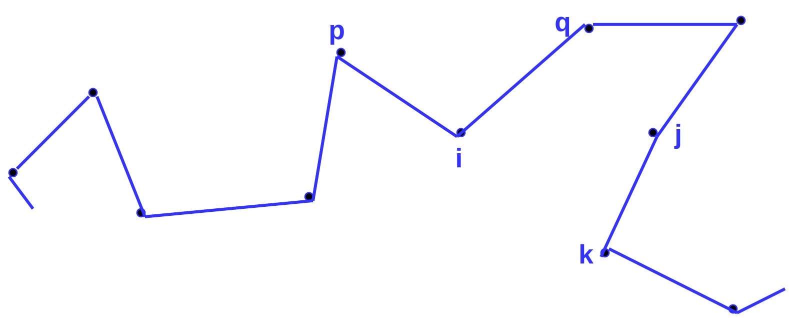
The Savings Algorithm (2)

- **Invented for vehicle routing; works well for TSP**
- **Connect every node to the origin ("depot") through a "round trip" (n-1 tours)**
- **Merge tours, one node at a time, by maximizing "savings" s(i,j) = d(D,i) + d(D,j) – d(i,j)**
- **Tours should not violate such constraints as:**
  - _ Vehicle capacity
  - _ Maximum length of a tour
  - _ Maximum number of stops per tour
- **$O(n^3)$**
- **Performs very well in practice; very flexible**
- **Li, F., B. Golden and E. Wasil (2005), "Very large-scale vehicle routing", *Computers and Opers. Research***

The Savings Algorithm (3)

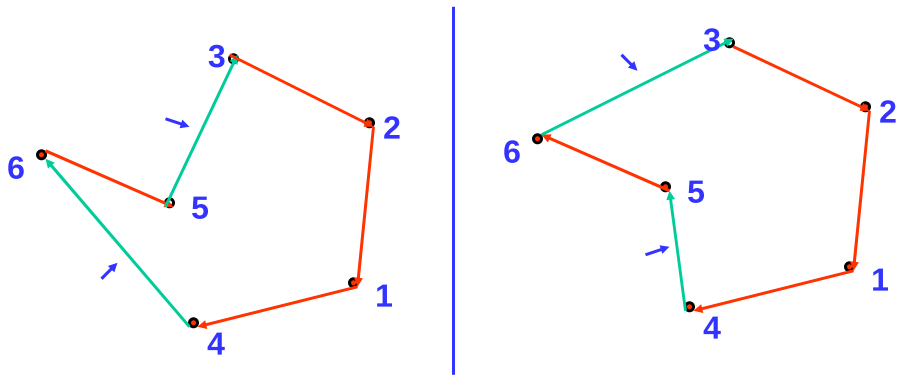


Tour Improvement Heuristics: Node Insertion



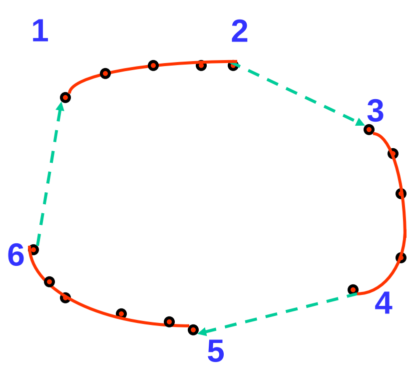
- d(p,q) + d(j,i) + d(i,k) vs. d(p,i) + d(i,q) + d(j,k)
- $O(n^2)$ computational effort on each iteration

Tour Improvement Heuristics: 2-exchange (or "2-opt")



$$\binom{n}{2} = \frac{n(n-1)}{2} \rightarrow O(n^2)$$

Tour Improvement Heuristics: 3-exchange (or "3-opt")



$$\begin{Bmatrix} 6-1 \\ 2-3 \\ 4-5 \end{Bmatrix} \quad \begin{Bmatrix} 6-2 \\ 1-3 \\ 4-5 \end{Bmatrix} \quad \begin{Bmatrix} 6-4 \\ 3-2 \\ 1-5 \end{Bmatrix} \quad \begin{Bmatrix} 6-1 \\ 2-4 \\ 3-5 \end{Bmatrix}$$

↔ 2-opt really

$$\begin{Bmatrix} 6-3 \\ 4-1 \\ 2-5 \end{Bmatrix} \quad \begin{Bmatrix} 6-2 \\ 1-4 \\ 3-5 \end{Bmatrix} \quad \begin{Bmatrix} 6-3 \\ 4-2 \\ 1-5 \end{Bmatrix} \quad \begin{Bmatrix} 6-4 \\ 3-1 \\ 2-5 \end{Bmatrix}$$

$$\binom{n}{3} \rightarrow O(n^3)$$

Tour Improvement Heuristics: Variable Depth Search

- Lin and Kernighan (1973)
- Use combinations of 2-opt and 3-opt searches
- Initially many “short-depth”, later fewer
- Has been extended to “deeper” searches than 3-opt
- Numerous variations