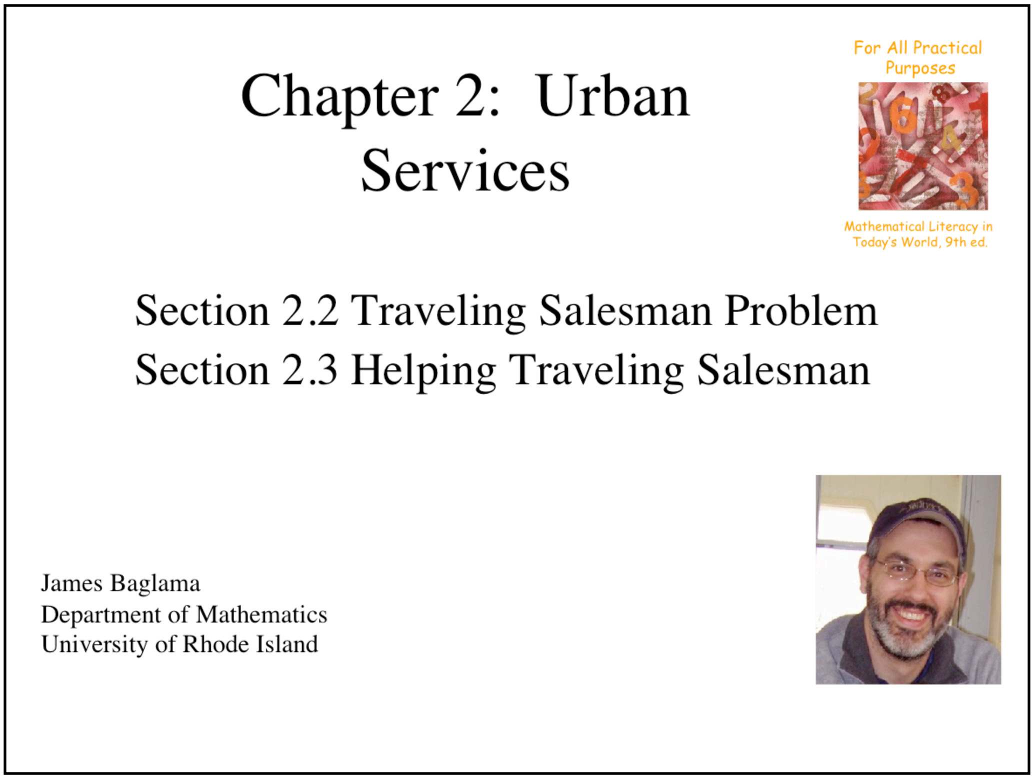


Chapter 2: Urban Services

For All Practical Purposes

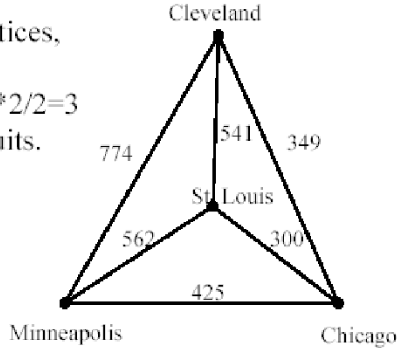
Mathematical Literacy in Today's World, 9th ed.

Section 2.2 Traveling Salesman Problem
Section 2.3 Helping Traveling Salesman

James Baglama
Department of Mathematics
University of Rhode Island

Review our previous example

We have four vertices, thus we have (4-1)!/2=3!/2=3*2/2=3 Hamiltonian circuits.

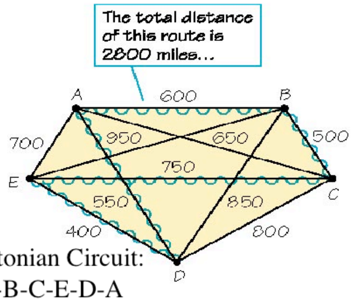


Suppose you have 25 cities then there would be 24!/2 which is approximately 3 x 10^{23} Hamiltonian circuits. If you had a super computer generating 1 million circuits per second this would take 10 billion years to generate them all. (**Brute Force**)

- <u>Traveling Salesman Problem</u> (TSP)
  - Difficult to solve Hamiltonian circuits when the number of vertices in a <u>complete</u> graph increases (*n* becomes very large).
  - This problem originated from a salesman determining his trip that minimizes costs (less mileage) as he visits the cities in a sales territory, starting and ending the trip in the same city.
  - Many applications today: bus schedules, mail drop-offs, telephone booth coin pick-up routes, Lobster fisherman picking up traps, etc.
  - Can you think of an application?

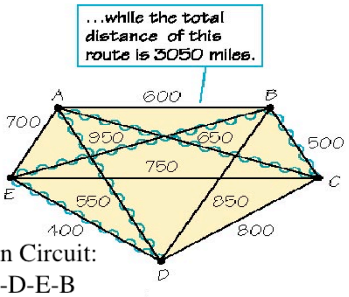
- <u>How can the TSP be solved</u>?
  - Computer program can find optimal route (not always practical).
  - Heuristic methods can be used to find a “fast” answer, but does not guarantee that it is always the optimal answer.
    - Nearest neighbor algorithm
    - Sorted edges algorithm

- <u>Nearest Neighbor Algorithm</u> *(to solve TSP)*
  - Starting from the “home” city (or vertex), first visit the nearest city (one with the least mileage from “home”).
  - As you travel from city to city, always choose the next city (vertex) that can be reached quickest (i.e., nearest with the least miles), that has not already been visited.
  - When all other vertices have been visited, the tour returns home.



Hamiltonian Circuit:
A-B-C-E-D-A

Nearest neighbor starting at vertex A



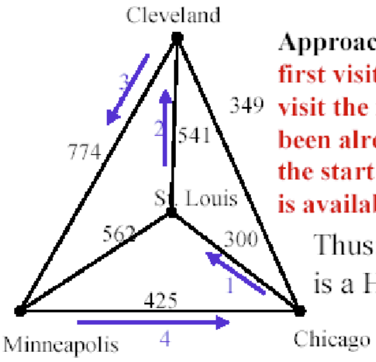
Hamiltonian Circuit:
B-C-A-D-E-B

Nearest neighbor starting at vertex B

Strategies for solving the Traveling Salesman Problem (TSP)

Nearest-Neighbor Algorithm

Goal: to find a Hamilton circuit



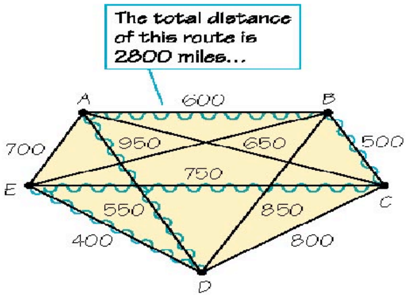
Approach: **starting from Chicago first visit the nearest city, then visit the nearest city that has not been already visited. We return to the start city when no other choice is available.**

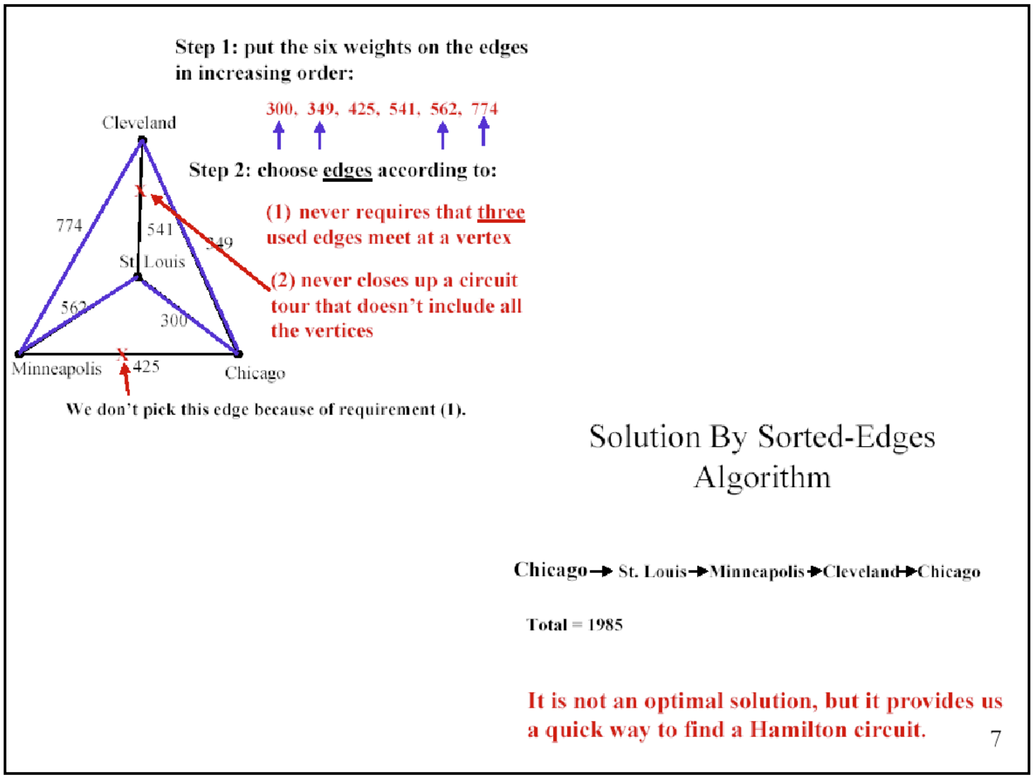
Thus: C→S→CL→M→C is a Hamilton circuit.

Note: This is not the optimal solution we found earlier (Section 2.1 – Brute Force - Methods of Trees). Making the best choice at each stage may not be the best global solution. However, it works quickly for large problems.

- Sorted Edges Algorithm *(to solve TSP)*
  - Start by sorting, or arranging, the edges in order of increasing cost (sort smallest to largest mileage between cities).
  - At each stage, select that edge of least cost until all the edges are connected at the end while following these rules:
    - If an edge is added that results in three edges meeting at a vertex, eliminate the longest edge.
    - Always include all vertices in the finished circuit.

Example using sorted edges Edges selected are DE at 400, BC at 500, AD at 550, and AB at 600 (AC and AE are not chosen because they result in three edges meeting at A). Lastly, CE at 750 is chosen to complete the circuit of 2800 miles.


The total distance of this route is 2800 miles…



Solution By Sorted-Edges Algorithm

Step 1: put the six weights on the edges in increasing order:

300, 349, 425, 541, 562, 774

Step 2: choose edges according to:

(1) never requires that three used edges meet at a vertex

(2) never closes up a circuit tour that doesn't include all the vertices

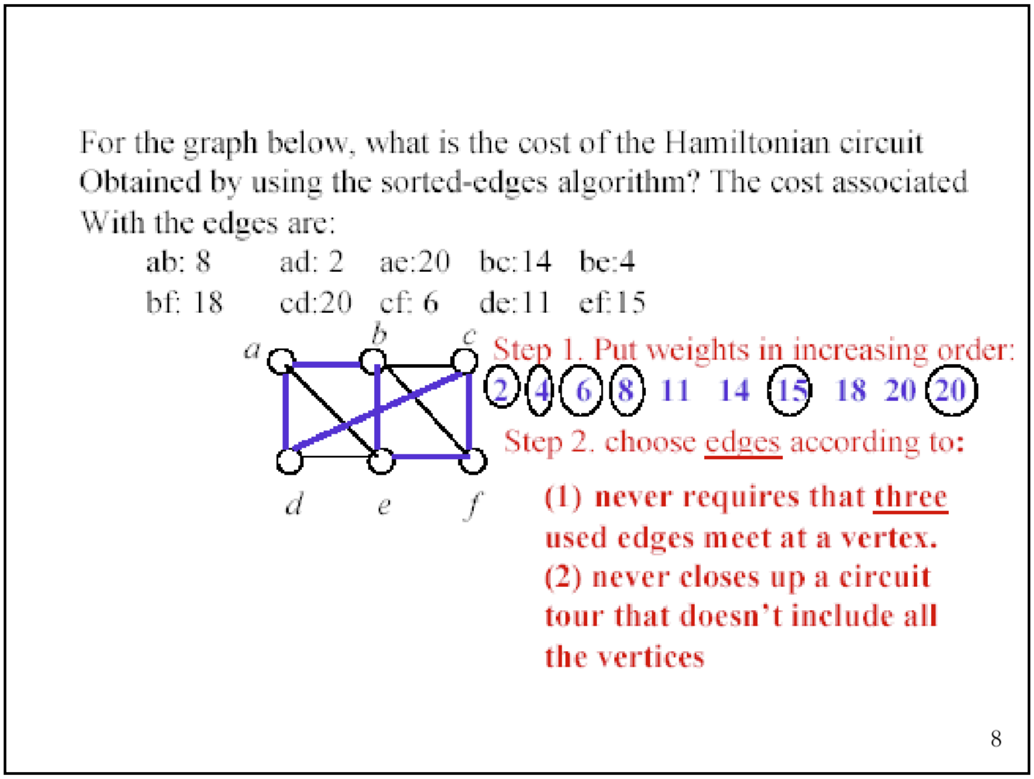
We don't pick this edge because of requirement (1).

Chicago→ St. Louis→Minneapolis→Cleveland→Chicago

Total = 1985

It is not an optimal solution, but it provides us a quick way to find a Hamilton circuit.

7



For the graph below, what is the cost of the Hamiltonian circuit Obtained by using the sorted-edges algorithm? The cost associated With the edges are:

ab: 8 ad: 2 ae:20 bc:14 be:4

bf: 18 cd:20 cf: 6 de:11 ef:15

Step 1. Put weights in increasing order:

2 4 6 8 11 14 15 18 20 20

Step 2. choose edges according to:

(1) never requires that three used edges meet at a vertex.

(2) never closes up a circuit tour that doesn't include all the vertices