DESIGN AND ANALYSIS OF ALGORITHMS
UE19CS251

Shylaja S S
Department of Computer Science
& Engineering

DESIGN AND ANALYSIS OF ALGORITHMS

Travelling Salesman Problem

Major Slides Content: Anany Levitin

Shylaja S S

Department of Computer Science & Engineering

Exhaustive Search

- Exhaustive Search is a brute – force problem solving technique
- It suggests generating each and every element of the problem domain, selecting those of them that satisfy all the constraints and then finding a desired element
- The desired element might be one which minimizes or maximizes a certain characteristic
- Typically the problem domain involves combinatorial objects such as permutations, combinations and subsets of a given set

Exhaustive Search - Method

- Generate a list of all potential solutions to the problem in a systematic manner
- Evaluate potential solutions one by one, disqualifying infeasible ones and, for an optimization problem, keeping track of the best one found so far
- When search ends, announce the solution(s) found

Travelling Salesman Problem

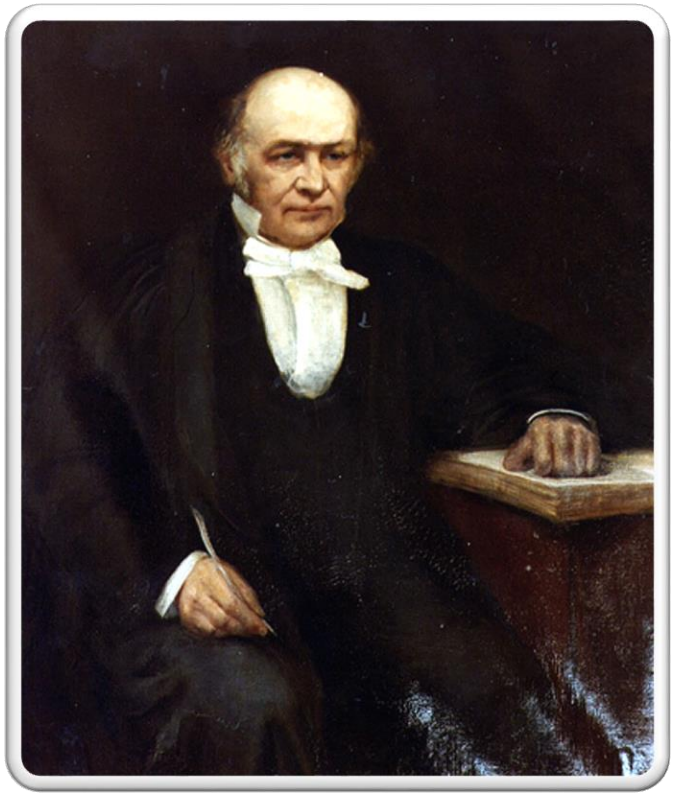
- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?

Alternative way to state the problem:

- Find the shortest Hamiltonian Circuit in a weighted connected graph

TSP: History and Relevance

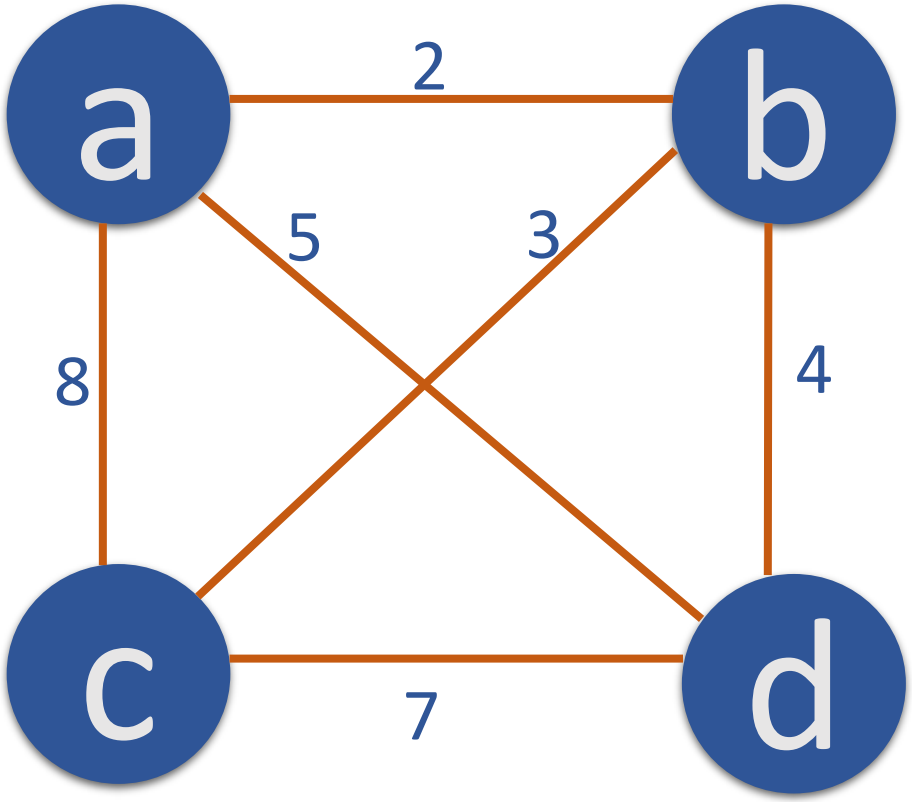
- The Travelling Salesman Problem was mathematically formulated by Irish Mathematician Sir William Rowan Hamilton
- It is one of the most intensively studied problems in optimization
- It has applications in logistics and planning



DESIGN AND ANALYSIS OF ALGORITHMS

Travelling Salesman Problem

Example



Tour	Length
a→b→c→d→a	**2+3+7+5 = 17**
a→b→d→c→a	2+4+7+8 = 21
a→c→b→d→a	8+3+4+5 = 20
a→c→d→b→a	8+7+4+2 = 21
a→d→b→c→a	5+4+3+8 = 20
a→d→c→b→a	**5+7+3+2 = 17**

Travelling Salesman Problem

Efficiency

- The Exhaustive Search solution to the Travelling Salesman problem can be obtained by keeping the origin city constant and generating permutations of all the other n – 1 cities
- Thus, the total number of permutations needed will be (n – 1)!

THANK YOU

Shylaja S S

Department of Computer Science & Engineering

shylaja.sharath@pes.edu