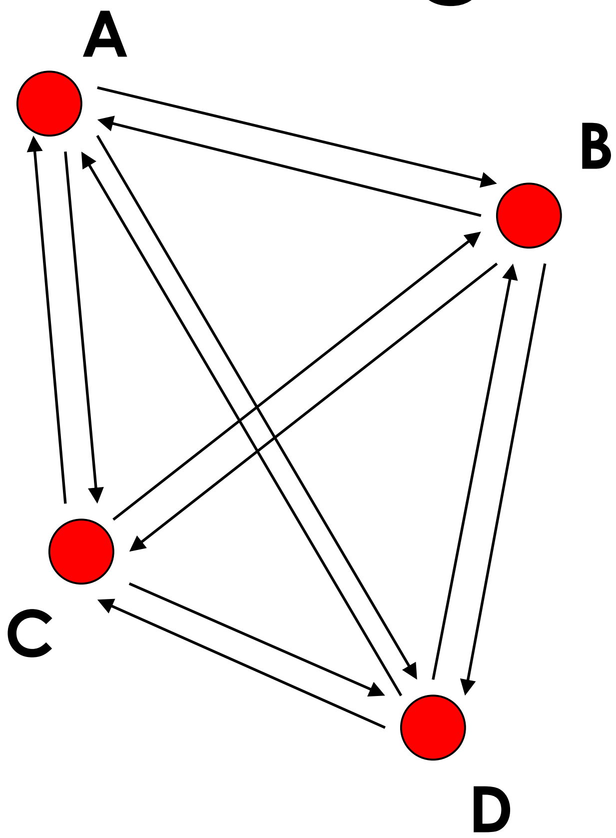
CS302 - Data Structures
using C++

Topic: Traveling Salesman Problem

Kostas Alexis

Traveling Salesman Problem (TSP)

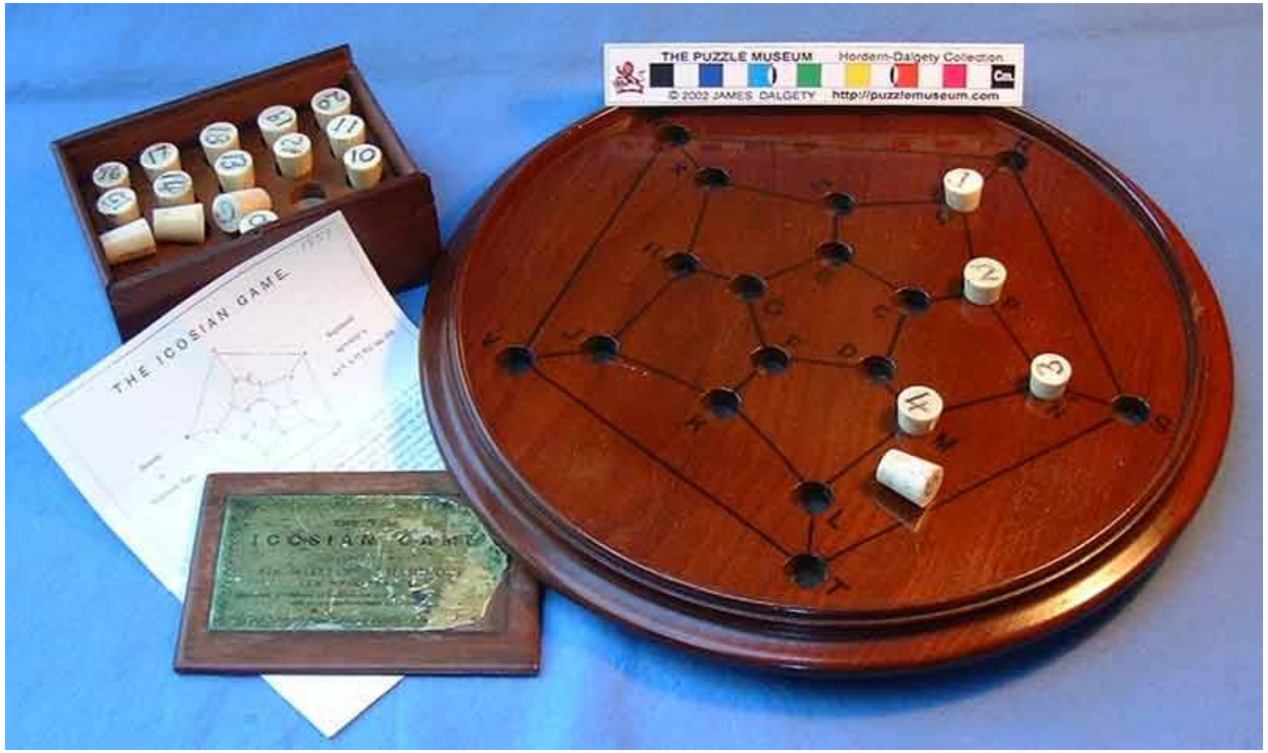


	A	B	C	D
A	-	3	4	6
B	2	-	5	2
C	7	4	-	3
D	5	6	7	-

Optimization Problem: Find the least cost tour starting at A, traveling through the other three cities exactly once and returning to A.

Decision Problem: Is there a TSP tour with cost less than k?

Traveling Salesman Problem (TSP)

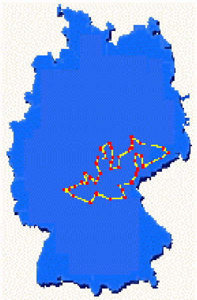


Icosian Game
1857

Traveling Salesman Problem (TSP)

Der Handlungsreisend
1832 German Manual

". . . it may not be possible to arrange the tours through Germany with more economy regarding distances, which is the main consideration of the traveler . . ."

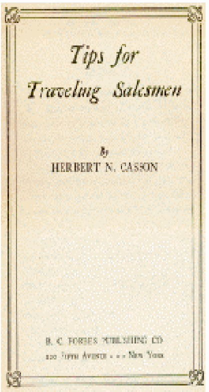


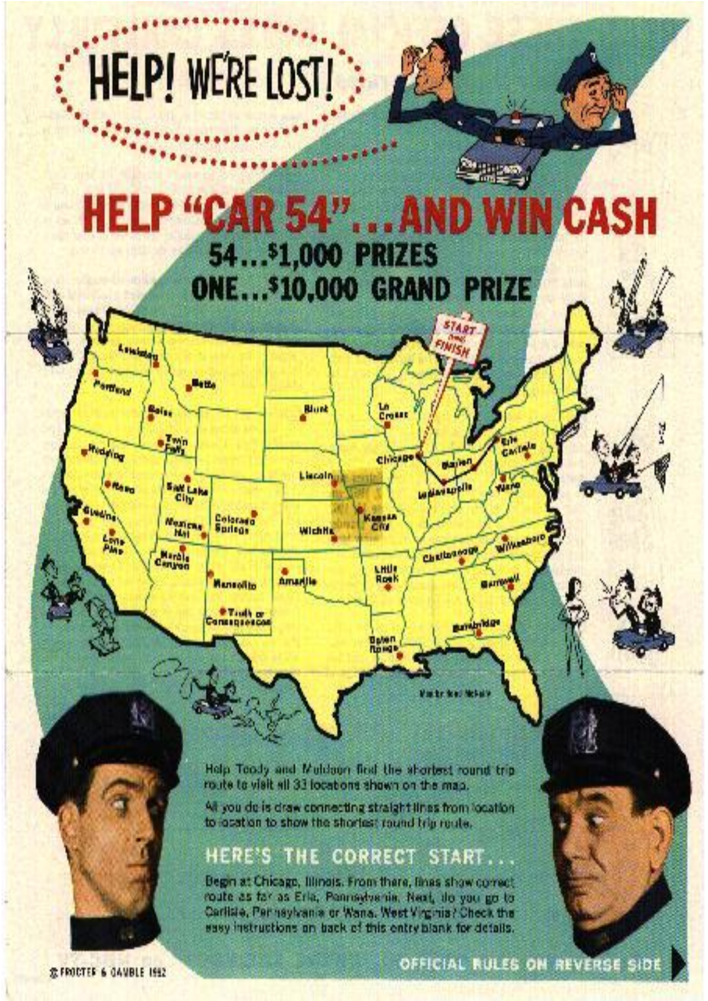
The Commercial Traveller's Guide Book
Linus Pierpont Brockett, 1871

How to Become a Commercial Traveller
E. Cadwell, 1893

The Tales of a Traveller
Simon S. Skidelsky, 1916

Tips for Traveling Salesmen
Herbert N. Casson, 1927





Proctor & Gamble Contest 1962

20^{th} Century History

- 1931-32 Merrill Flood, A.W.Tucker, Hassler Whitney (Princeton) - pose the Traveling Salesperson Problem.
- 1947 George B. Dantzig designs simplex method for linear programming.
- 1954 Connection with the Assignment Problem established.
- 1962 Dynamic programming formalized.
- 1972 Karp proved TSP is NP-complete.
- 1985 – 2008 Techniques leading to solving a 85,000-city problem

Traveling Salesman Problem

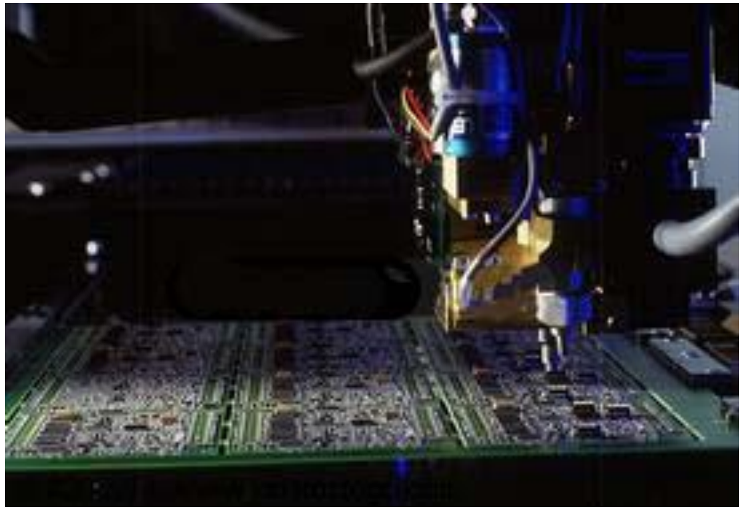
- **The travelling salesman problem (TSP) asks the following question:** "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?" It is an NP-hard problem in combinatorial optimization, important in operations research and theoretical computer science.

Traveling Salesman Problem (TSP)

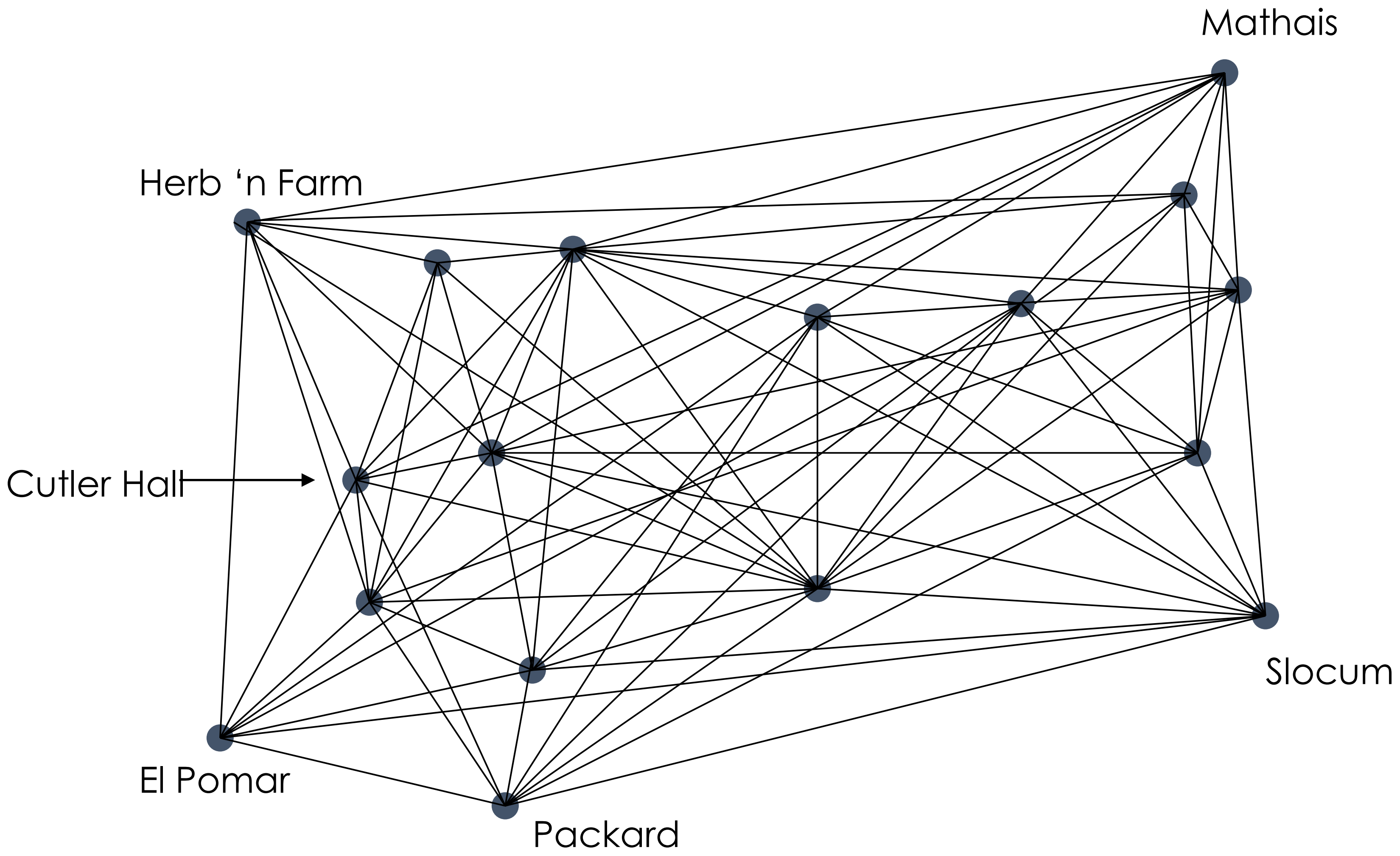
- **The travelling salesman problem (TSP) asks the following question:** "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?" It is an NP-hard problem in combinatorial optimization, important in operations research and theoretical computer science.
- **Modeled as a Graph:** TSP can be modelled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's weight. It is a minimization problem starting and finishing at a specified vertex after having visited each other vertex exactly once. Often, the model is a complete graph (i.e. each pair of vertices is connected by an edge). If no path exists between two cities, adding an arbitrarily long edge will complete the graph without affecting the optimal tour.

TSP Applications

- School Bus routing.
- Robotic welding in the car industry.
- Printed circuit board drilling and laser cutting of integrated circuits.
- Genome sequencing. Finding most likely ordering of markers.
- Job processing: Chemical plants – cost of setup to produce chemicals.



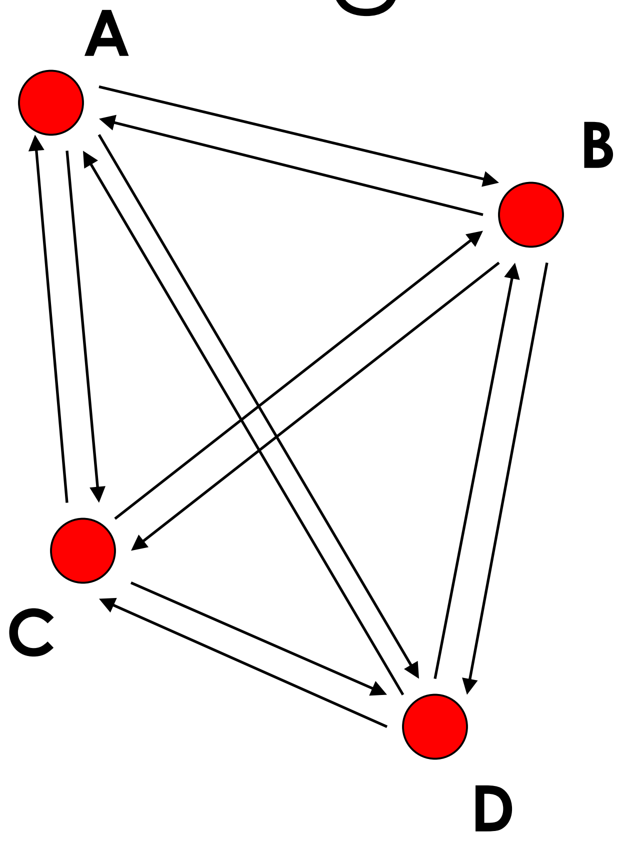
Graph Example



Versions of TSP

- Complete or Incomplete graph.
- Symmetric vs. Asymmetric cost matrix.
- Euclidean (Triangle inequality holds.)
- Upper Triangular cost matrix.
- Circulant cost matrix.
- Bipartite graph.

Traveling Salesman Problem (TSP)



	A	B	C	D
A	-	3	4	6
B	2	-	5	2
C	7	4	-	3
D	5	6	7	-

Traveling Salesperson Problem: Find the least cost tour starting at A, traveling through the other three cities exactly once and returning to A.

Traveling Salesman Problem (TSP)

Generating Permutations:

Recursion:

A < BCD	B < ACD	C < ABD	D < ABC
BDC	ADC	ADB	ACB
CBD	CAD	BAD	BAC
CDB	CDA	BDA	BCA
DBC	DAC	DAB	CAB
DCB	DCA	DBA	CBA

Single exchange:

ABCD	CABD	DCBA	DBAC
BACD	ACBD	DCAB	DBCA
BCAD	ACDB	DACB	BDCA
BCDA	CADB	ADCB	BDAC
CBDA	CDAB	ADBC	BADC
CBAD	CDBA	DABC	ABDC

Naïve Algorithm

1. List all tours and their costs:

   A BCD A = 18
   A BDC A = 19
   A CBD A = 15
   A CDB A = 15
   A DBC A = 24
   A DCB A = 19

2. Find a tour with minimum cost:

   A CBD A = 15 (one optimal tour)

Timing Results for Naïve Algorithm

Cities	Seconds	Tours	Sec/Tour
11	0.50	3,628,800	1.38×10^{-7}
12	5.61	39,916,800	1.41×10^{-7}
13	69.12	479,001,600	1.44×10^{-7}

Estimate for 17 cities: 34.9 days (2.09×10^{13} tours)

Complexity

- If there are n-1 cities (other than the home city A), then there are (n-1)! possible tours.
- The naïve algorithm takes at least (n-1)! steps.
- A random algorithm could possibly find an optimal tour in (n-1) steps.
- Is there a deterministic algorithm that can find an optimal tour in a polynomial number of steps?

Traveling Salesman Problem (TSP)

- **Theorem:** (Karp, 1972) TSP is NP-Complete. (That is, every other hard problem is reducible to TSP.)
- **Open problem:** Is there a polynomial time algorithm for TSP? If not, can you prove it? (In technical terms, does **P** = **NP**?)
- Solving the open problem earns you $1,000,000 from the Clay Mathematics Institute.

Traveling Salesman Problem (TSP)

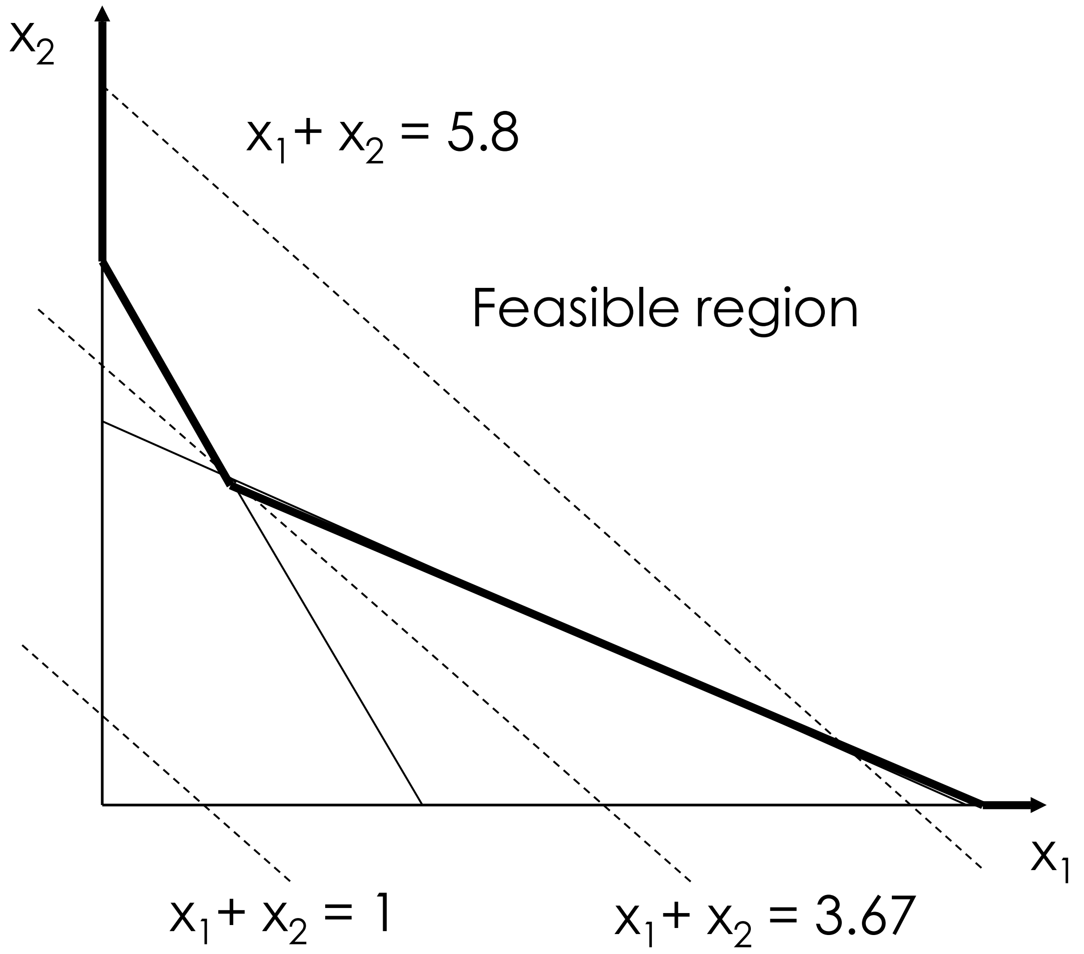
- **Approaches to TSP**
  - Constant TSP: Analyze the cost matrix.
  - Linear Programming
  - Euclidean TSP: Probabilistic analysis.
  - Approximation algorithms
  - Search techniques
    - Branch and Bound
    - Dynamic Programming
    - Genetic Algorithm
    - Ant Colony

Linear Programming

Linear Program

Minimize: $x_1 + x_2$

Subject to: $2x_1 + x_2 >= 5$
$x_1 + 2x_2 >= 6$
$x_1 >= 0$
$x_2 >= 0$



The TSP Polytope

- **The TSP Polytope**
  - Let $x = (x_{ij})$ be a vector with an entry 1 indicating that edge ij is included. Entry 0 indicates the edge is excluded. Hence the vector has length equal to the number of edges.
  - Let $\mathbf{x}^t$ be the vector corresponding to the tour **t**.
- **Polytope P = convex hull of $\{\mathbf{x}^t | \mathbf{t} \text{ is a tour}\}$**
  - Interestingly, dim P = $|E| - |V| = n(n-3)/2$
  - (Where E is the set of edges and V is the set of vertices.)

Linear Programming for TSP

Linear Programming formulation of the TSP:

$$\text{Minimize:} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{Subject to:} \quad \text{x} \in \text{TSP Polytope}$$

Relaxation of the Linear Program:

$$\text{Minimize:} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{Subject to:} \quad \sum_{i} x_{ij} = 1 \text{ and } \sum_{j} x_{ij} = 1$$

Additional constraints:

$$\text{Assignment Problem:} \quad x_{ij} = 0 \text{ or } x_{ij} = 1$$

$$\text{Subtour Elimination:} \quad \sum_{i \in S}\ \sum_{j \in V-S} x_{ij} \geq 1 \text{ for all } S \subset V, S \neq \emptyset$$

Branch and Bound Technique

Add A to the possible list and assign arbitrary cost.

While the possible list is not empty do:

- Select node with smallest estimated cost and remove from list.
- Generate children of selected node and add to list.
- If children complete a tour, update optimal tour, otherwise estimate cost of completed tour using this child.
- If estimated cost is greater than current optimal, remove node from list.

Estimate must be a lower bound on the cost and is called the **bounding function**. Technique depends on how accurate estimate is.

A Possible Bounding Function B(i)

Each node **i** in the search tree represents a partial tour. Those cities not in the partial tour form a set of “remaining” cities.

C_1 = cost of partial tour.

C_2 = sum of the minimum edges into each remaining city.

B(i) = C_1 + C_2

B(i) <= Cost of any tour containing the partial route.

Thank you