CE 205A
Transportation Logistics

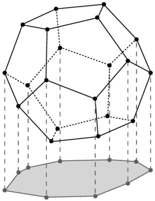
Lecture 5

Traveling Salesman Problem

Previously on Transportation Logistics

Which of these two formulations are better? In such settings we make use of the idea of projections to compare them on even footing.

Suppose relaxation of formulation P_1 has decision variables $x \in P_1 \subseteq \mathbb{R}^n$. Consider a new formulation whose decision variables are of the type $(x, w) \in Q_2 = X \times W \subseteq \mathbb{R}^n \times \mathbb{R}^l$.



We then define a projection of Q_2 into the subspace $\mathbb{R}^n$ as follows

$$P_2 = proj_x(Q_2) = \{x \in \mathbb{R}^n : (x, w) \in Q_2 \text{ for some } w \in W\}$$

The new formulation is better only if $P_2 \subset P_1$. Using the point $x_t = d_t$ and $y_t = d_t/M$ can you comment on the strength of the two lot sizing formulations?

Lecture Outline

Lecture Outline

Traveling Salesman Problem

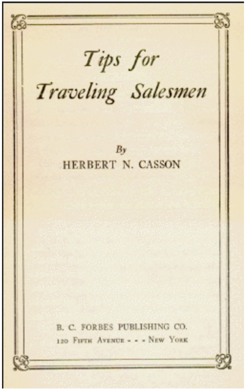
Traveling Salesman Problem

Introduction

The travelling salesman problem is one of the most celebrated examples of integer programming.

You have to visit n cities starting from an origin and return to the origin. Find the cheapest tour.

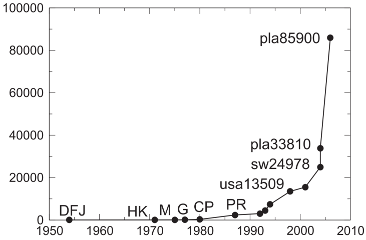
In addition to logistics, TSPs have been widely used in DNA sequencing and PCB manufacturing.

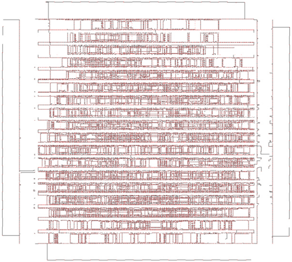


https://www.math.uwaterloo.ca/tsp/concorde.html

One of the earliest exact solutions for the TSP problem was proposed by Dantzig, Fulkerson, and Johnson using 49 cities in the US.

Mahalanobis looked at empirical estimates of the objective as a function of number of vertices in the late 30s in the context of surveying jute crops in Bengal. He hypothesized that the cost of Euclidean instances is $\propto \sqrt{n}$.





Several breakthroughs followed in the next few decades. The pla85900 is a chip design instance that was solved optimally in 2006.

Traveling Salesman Problem

TSP Greatest Hits

Dantzig, Fulkerson, and Johnson (1956)

Gomory (1958)

Miller, Tucker, and Zemlin (1960)

Bellman, Held, Karp (1962)

Little et al. (1963)

Hong (1972)

Lin and Kernighan (1973)

Grotschel and Padberg I & II (1979)

Crowder and Padberg (1980)

Padberg and Hong (1980)

Padberg and Rao (1982)

Padberg and Rinaldi (1987)

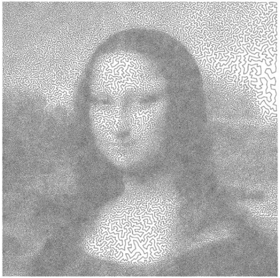
Grotschel and Holland(1991)

Padberg and Rinaldi (1991)

Padberg and Sung (1991)

Helsgaun (2000)

Mona Lisa TSP challenge: 100,000 node instance



July 27, 2012: GAP=107. A truncated branch-and-cut search, using an an artificial upper bound of 5757092 and the LP from April 18, was terminated after 11.5 CPU years and 20,787 search nodes. The run improved the gap by 4 units.

Traveling Salesman Problem

Challenge Accepted?

A world tour dataset with 1,904,711 vertices. An optimal solution is yet to be discovered.

Consider an undirected graph $G = (V, E)$ with vertices V and undirected edges E. For simplicity, suppose that G is a complete graph. Every pair of cities connected by an edge $e = \{u, v\}$ has a distance/cost c_e.

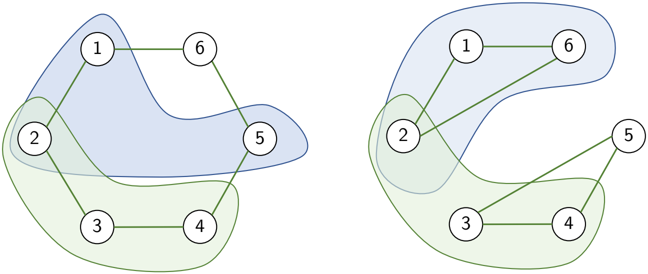
$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e & \\
\text{s.t.} \quad & \sum_{u \in e} x_e = 2 & \forall\, u \in V \\
& \sum_{u \in S} \sum_{v \in S} x_{\{u,v\}} \leq |S| - 1 & \forall\, S \subset V, S \neq \emptyset \\
& x_e \in \{0, 1\} & \forall\, e \in E
\end{aligned}
$$

The first constraint ensures that each city is visited exactly once and the second constraint prevents sub-tours.

Traveling Salesman Problem

DFJ Formulation

What if we did not have sub-tour elimination constraints? When is the constraint strictly $<$ and when is it $=$?



An alternate way of writing the sub-tour elimination constraints (SEC) is

$$\sum_{u \in S} \sum_{v \in S^c} x_{\{u,v\}} \geq 2 \qquad \forall\, S \subset V, S \neq \emptyset$$

Apply this version to the above examples? Show that the two SEC are equivalent.

Traveling Salesman Problem

E-δ Notation

The following short-hand notation is widely used in TSP literature. Let $S \subseteq V$.

- $E(S)$: Edges with both end points in S (also called the *edge set*).
- $\delta(S)$ or $\delta(S, S^c)$: Set of edges with one end in S and another in S^c (also called the *cut set*).
- If $|S| = 1$, we write $\delta(u)$ instead of $\delta(\{u\})$ to indicate the set of edges which have u as one end point.

Additionally, we define $x(E(S))$ and $x(E(S))$ as follows

$$x(E(S)) = \sum_{u \in S} \sum_{v \in S} x_{\{u,v\}}$$

$$x(\delta(S)) = \sum_{u \in S} \sum_{v \in S^c} x_{\{u,v\}}$$

The xs are also called *incidence vectors*. The subgraph with edges for which x values are positive is called the *support graph*.

Traveling Salesman Problem

E-δ Notation

The DFJ formulation using E-δ notation can be written as

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & x(\delta(u)) = 2 && \forall\, u \in V \\
& x(E(S)) \leq |S| - 1 && \forall\, S \subset V, S \neq \emptyset \\
& x_e \in \{0, 1\} && \forall\, e \in E
\end{aligned}$$

The formulation with the alternate SEC constraints take the form

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & x(\delta(u)) = 2 && \forall\, u \in V \\
& x(\delta(S)) \geq 2 && \forall\, S \subset V, S \neq \emptyset \\
& x_e \in \{0, 1\} && \forall\, e \in E
\end{aligned}$$

We can further restrict $3 \leq |S| \leq |V|/2$. (Why?)

Lecture Outline

Variants

Variants

Asymmetric TSP

In general, the cost of traveling from cities i to j maybe different from j to i. Consider a directed graph $G = (N, A)$ and suppose that G is complete. The cost between a pair of cities i and j is c_{ij}.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & \sum_{j\in N} x_{ij} = 1 & \forall\, i \in N \\
& \sum_{i\in N} x_{ij} = 1 & \forall\, j \in N \\
& \sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1 & \forall\, S \subset N, S \neq \emptyset \\
& x_{ij} \in \{0,1\} & \forall\, (i,j) \in A
\end{aligned}
$$

The first two constraints ensure that each vertex is visited exactly once and the third constraint prevents sub-tours.

Variants

Asymmetric TSP

For directed graphs, the shorthand A-δ notation can be extended by additionally defining the following symbols

- $\delta^+(S)$: Set of arcs with tail node in S and head node in S^c.
- $\delta^-(S)$: Set of arcs with tail node in S^c and head node in S.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & x(\delta^+(i)) = 1 & \forall\, i \in N \\
& x(\delta^-(i)) = 1 & \forall\, i \in N \\
& x(A(S)) \leq |S| - 1 & \forall\, S \subset N, S \neq \emptyset, |S| \geq 2 \\
& x_{ij} \in \{0, 1\} & \forall\, (i,j) \in A
\end{aligned}
$$

The SEC constraint can be replaced with $x(\delta^+(S)) + x(\delta^-(S)) \geq 2$. How many variables and constraints are present in this DFW formulation?

Variants

MTZ Formulation

Alternately, Miller, Tucker, and Zemlin's model (MTZ) can be used which keeps track of u_i which is the sequence in which city i is visited. Wlog, let node 1 be the origin of the tour, i.e., $u_1 = 1$.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} & \\
\text{s.t.} \quad & x(\delta^+(i)) = 1 & \forall\, i \in N \\
& x(\delta^-(i)) = 1 & \forall\, i \in N \\
& u_i - u_j + 1 \le (n-1)(1 - x_{ij}) & \forall\, i, j \in \{2, \ldots, n\}, i \ne j \\
& 2 \le u_i \le n & \forall\, i \in N \backslash \{1\} \\
& x_{ij} \in \{0, 1\} & \forall\, (i, j) \in A
\end{aligned}
$$

What happens to the SEC when x_{ij} equals 1 and 0? Note: The SEC constraint appears in slightly different formats in different papers.

Variants

MTZ Formulation

How does the MTZ formulation eliminate sub-tours? Is this a stronger formulation compared to DFJ? Since the decision variables are in a different space, we need to compare the projection.

Padberg and Sung show that the MTZ formulation is weaker compared to the DFJ version. For some intuition, consider a directed cycle C and add the MTZ SEC inequalities to eliminate the u variables.

$$\sum_{(i,j)\in C} x_{ij} \le \left(1 - \frac{1}{n-1}\right)|C|$$

Compare this with $x(A(C)) \le |C| - 1$. Which one is tighter? Descrochers and Laporte proposed a tightened version using the following constraint.

$$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \le n-2 \qquad \forall i,j \in \{2,\dots,n\}$$

Variants

GG Single Commodity Flow Formulation

Gavish and Graves proposed the following single commodity flow formulation for the TSP where the traveler starts from node 1 with $n-1$ units of a commodity and one unit is delivered at each node.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & y(\delta^+(1)) = n-1 & \\
& y(\delta^+(i)) - y(\delta^-(i)) = 1 & \forall\, i \in \{2,\ldots,n\} \\
& y_{ij} \le (n-1)x_{ij} & \forall\, (i,j) \in A \\
& x_{ij} \in \{0,1\} & \forall\, (i,j) \in A
\end{aligned}
$$

This formulation is again weaker than DFJ's version.

Variants

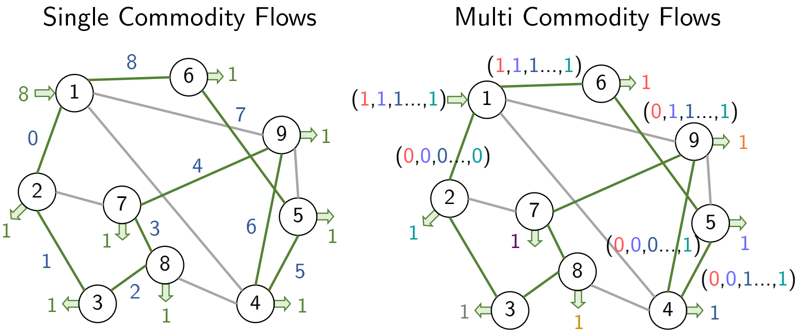
CW Multi-Commodity Flow Formulation

Claus and Wong independently proposed the following multi-commodity version which has the same LP bounds as that of DFJ. Assume that there are n commodities that start at node 1 and are delivered to the other cities.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & x(\delta^+(i)) = x(\delta^-(i)) = 1 & \forall\, i \in N \\
& y^k(\delta^+(1)) = 1 & \forall\, k \in \{2,\dots,n\} \\
& y^k(\delta^-(1)) = 0 & \forall\, k \in \{2,\dots,n\} \\
& y^k(\delta^+(k)) = 0 & \forall\, k \in \{2,\dots,n\} \\
& y^k(\delta^-(k)) = 1 & \forall\, k \in \{2,\dots,n\} \\
& y^k(\delta^+(i)) - y^k(\delta^-(i)) = 0 & \forall\, i,k \in \{2,\dots,n\}, i \neq k \\
& y^k_{ij} \leq x_{ij} & \forall\, (i,j) \in A, k \in \{1,\dots,n\} \\
& x_{ij} \in \{0,1\} & \forall\, (i,j) \in A
\end{aligned}
$$

Variants

Single vs. Multi Commodity Flow Formulation



In the TSPTW, each city i in a directed graph $G = (N, A)$ has an associated time window $[a_i, b_i]$ within which it must be visited. Suppose the travel time to go from i to j is t_{ij}.

The traveler can reach early, in which case they wait till the start of the time window. Suppose 0 is the starting city and y_i is the time at which customer i is visited, then the TSPTW can be formulated as

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & x(\delta^+(i)) = 1 & \forall\, i \in N \\
& x(\delta^-(i)) = 1 & \forall\, i \in N \\
& y_j \geq y_i + t_{ij} - M(1 - x_{ij}) & \forall\, i, j \in N, j \neq 0 \\
& a_i \leq y_i \leq b_i & \forall\, i \in N \\
& x_{ij} \in \{0, 1\} & \forall\, (i,j) \in A
\end{aligned}
$$

Let $G = (N, A)$ be a directed network. In this version, the traveler is not forced to visit all cities. Instead, they get a prize w_i for visiting a city i and pay a penalty p_i for not visiting i.

The goal of the traveler is to find a tour that minimizes the cost and penalty while ensuring that the total prize received is at least U.

As before, traveling between cities i and j has a cost c_{ij}. Let x_{ij} be 1 if arc (i, j) is visited, and let y_i equal 1 if i is part of the tour and is 0 otherwise. Suppose $G(x, y)$ indicates the support graph.

Formulate this problem as a MIP.

Variants

Prize Collecting TSP (PCTSP)

The SEC constraints for this problem imply that $G(x,y)$ has a single cycle.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{i\in N} p_i(1-y_i) & \\
\text{s.t.} \quad & \sum_{j\in N\setminus\{i\}} x_{ij} = y_i & \forall\, i \in N \\
& \sum_{i\in N\setminus\{j\}} x_{ij} = y_j & \forall\, j \in N \\
& \sum_{i\in N} w_i y_i \geq U & \\
& x(A(S)) \leq \sum_{i\in S\setminus\{k\}} y_i + (1-y_l) & \forall k \in S, l \in S^c, S \subset N, |S| \geq 2 \\
& x_{ij} \in \{0,1\} & \forall\, (i,j) \in A \\
& y_i \in \{0,1\} & \forall\, i \in N
\end{aligned}
$$

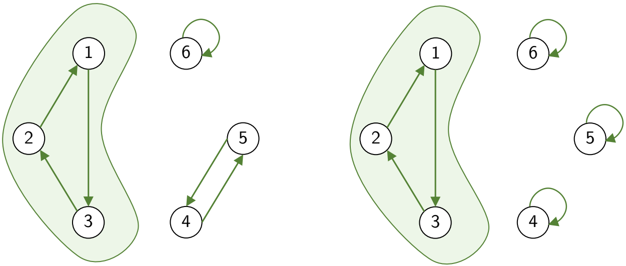
Variants

Prize Collecting TSP (PCTSP)

Alternately, one could allow self-loops and set $A \leftarrow A \cup \{(i,i) : i \in N\}$, $x_{ii} = 1 - y_i$, and $c_{ii} = p_i$.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
\text{s.t.} \quad & \sum_{j\in N} x_{ij} = 1 & \forall\, i \in N \\
& \sum_{i\in N} x_{ij} = 1 & \forall\, j \in N \\
& \sum_{i\in N} w_i x_{ii} \leq \sum_{i\in N} w_i - U & \\
& x(\delta(S)) + x_{kk} + x_{ll} \geq 1 & \forall k \in S, l \in S^c, S \subset N, |S| \geq 2 \\
& x_{ij} \in \{0,1\} & \forall\, (i,j) \in A
\end{aligned}
$$

The support graph may now contain a single cycle with length ≥ 2 and self-loops.



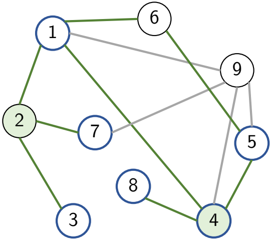
Can you verify if the SEC constraints are satisfied for the above examples?

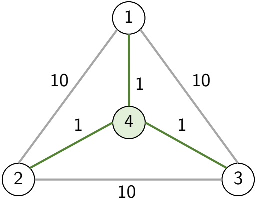
Variants

Steiner TSP

This version is apt for real-world road networks and not complete graphs in which it may be necessary to revisit nodes.

Let's start with an undirected graph $G = (V, E)$, and a set of required nodes V_R, the objective is to find a minimum cost tour that visits each of the required nodes at least once.





Note that revisits can also reduce costs in problems where all cities must be visited as seen in the above example.

The edges in the graph can be visited more than once as well. Let x_e be the number of times an edge is traversed.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & x(\delta(u)) \text{ is even} && \forall\, u \in V \\
& x(\delta(S)) \geq 2 && \forall S \subset V, S \cap V_R \neq \emptyset, V_R \setminus S \neq \emptyset \\
& x_e \in \mathbb{Z}^+ && \forall\, e \in E
\end{aligned}
$$

How do you enforce the first constraint?

Variants

Steiner TSP

Flow-based formulations (single or multi-commodity) are commonly used in practice to avoid dealing with exponential SECs.

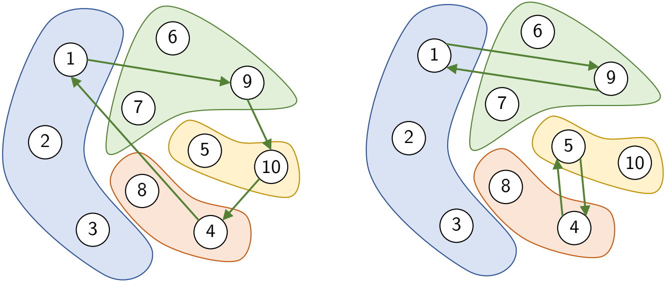
$$
\begin{aligned}
\min \ & \sum_{e \in E} c_{ij} x_{ij} & \\
\text{s.t.} \ & x(\delta^+(i)) \geq 1 & \forall\, i \in V_R \\
& x(\delta^+(i)) = x(\delta^-(i)) & \forall\, i \in N \\
& y(\delta^+(i)) - y(\delta^-(i)) = 1 & \forall\, i \in V_R \setminus \{1\} \\
& y(\delta^+(i)) - y(\delta^-(i)) = 0 & \forall\, i \in V \setminus V_R \\
& 0 \leq y_{ij} \leq (n_R - 1) x_{ij} & \forall (i,j) \in A \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in A
\end{aligned}
$$

Variants

Generalized TSP problem

In the Generalized TSP or Set TSP, we are given a graph $G = (N, A)$ and a partition of the nodes $N = S_1 \cup S_2 \cup \ldots \cup S_m$, $S_k \cap S_l = \emptyset$.

The goal is to find a minimum cost cycle of $m \geq 4$ cities which includes exactly one city from each S_k.



Note that given the city in each S_k to be visited, the problem is a simple TSP. But we must also determine which city in each set to visit.

Formulate this problem as a MIP using just the x_{ij} variables.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{s.t.} \quad & x(\delta^+(S_k)) = 1 && \forall\, k = 1, \ldots, m \\
& x(\delta^-(S_k)) = 1 && \forall\, k = 1, \ldots, m \\
& x(\delta^+(i)) = x(\delta^-(i)) && \forall\, i \in N \\
& \sum_{Q\in S} \sum_{R\in S^c} x(\delta(Q, R)) \geq 1 && \forall\, S \subset \{S_1, \ldots, S_m\}, 2 \leq |S| \leq m-2 \\
& x_{ij} \in \{0, 1\} && \forall\, (i,j) \in A
\end{aligned}
$$

The SEC constraints are similar but are applied to subsets of sets instead of subsets of cities.

Variants

Arc Routing Problem

Arc routing requires visiting every arcs in a graph instead of nodes. This has several applications such as snow plowing, garbage collection, newspaper distribution, road inventory, surveillance, street views, etc.

The problem was originally proposed by a Chinese mathematician Meigu Guan in 1960 and is also known as the Chinese Postman Problem (CPP) or route inspection problem.

The goal in the problem is to find a tour that visits all edges and return back to the starting point in the shortest time/distance. Nodes may be revisited.

The 'Steiner' version of CPP is called the Rural Postman Problem (RPP) in which a subset of arcs must be visited.

The CPP problem is polynomially solvable but many of its variants are not.

Given an undirected graph $G = (V, E)$, we keep track of x_{uv} which is the number of times we move from u to v.

In a way, it suffices to visit at least one of the 'sides' of the street. Multiple visits on an edge may be required depending on the costs.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{\{u,v\}}(x_{uv} + x_{vu}) & \\
\text{s.t.} \quad & x_{uv} + x_{vu} \geq 1 & \forall \{u, v\} \in E \\
& x(\delta^+(u)) = x(\delta^-(u)) & \forall\, u \in V \\
& x_{uv}, x_{vu} \in \mathbb{Z}_+^n & \forall \{u, v\} \in E
\end{aligned}
$$

There are other VRP like extensions called Capacitated Arc Routing Problem (CARP) in which vehicles starting at a depot must visit edges and serve demands.

Variants

Subway/Tube Challenge

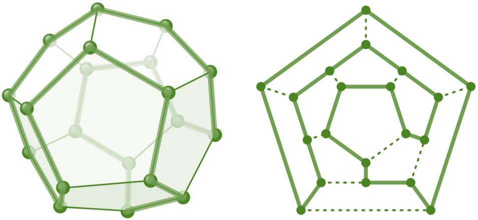
The goal is to find tours in a transit network in the fastest possible time.

Travelers are either required to pass through each station or stop at each station. Fastest record holders have a place in Guinness records! See the rules and records for New York City and London.

Variants

Related Problems

The icosan game, invented by William Hamilton, is puzzle that involves visiting the corners of a dodecahedron – polygon with 12 faces.



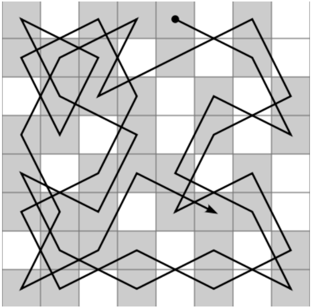
A *Hamiltonian path* is one that visits each vertex of a graph exactly once. If the starting node is included, it forms a *Hamiltonian circuit*.

Complete graphs always have Hamiltonian paths. Determining if such a path/circuit exists is non-trivial for general settings.

Variants

Related Problems

The Knight's tour involves finding a sequence of moves on the chess board that traverses every square.

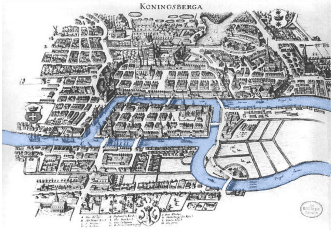


One of the earliest references of this problem can be found in the works of a 9th century Indian poet Rudrata.

Variants

Related Problems

The seven bridges of Königsberg problem is to find a tour around the bridges on river Pregel without passing through a bridge more than once.



Euler in the 1700s showed that it is not possible to find such a path. An *Eulerian path* is one which passes through each edge exactly once with node revisits allowed.

If the path starts and begins at the same node, it is called a *Eulerian circuit*. He showed that a connected undirected graph has an Eulerian circuit iff every vertex is of even degree.

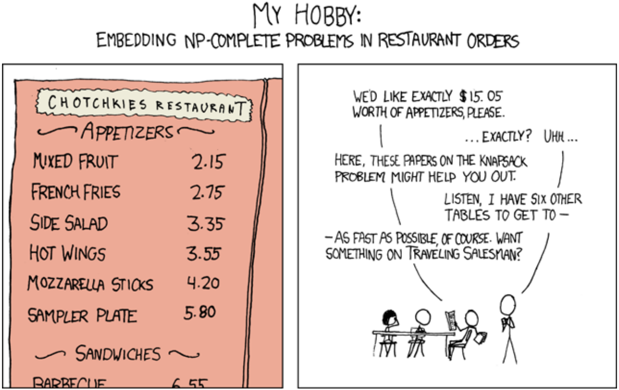
Variants

Related Problems

TSPs in which the edges satisfy triangle inequality are called metric TSPs. This property can be exploited in heuristics for finding near optimal tours.

A special case of metric TSP is the Eucledian TSP. Here, the cities are points in $\mathbb{R}^n$ and the edges connecting them are straight lines joining the end points.

Your Moment of Zen



Source: xkcd