Graph Theory: Hamilton Circuits, TSP

Contemporary Math

Josh Engwer

TTU

24 July 2015

Hamilton Paths & Hamilton Circuits (Definition)

Recall from the previous section the notions of Euler paths & Euler circuits.
Euler paths/circuits traverse all edges of a graph exactly once.

What if it's desired to pass through all vertices, instead??

Definition

(Hamilton Path, Hamilton Circuit)

A **Hamilton path** is a path that passes through all vertices exactly once.
A **Hamilton circuit** is a Hamilton path that begins & ends at the same vertex.

Complete Graphs (Definition)

With respect to finding Hamilton paths and Hamilton circuits,
graphs with relatively few edges compared to vertices are not interesting.

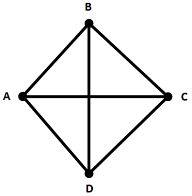
So, let's consider graphs which have many edges joined to each vertex:

Definition

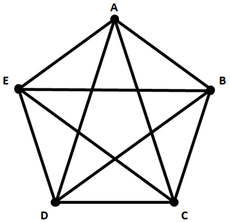
(Complete Graph)

A **complete graph** has every pair of vertices is joined by exactly one edge.

NOTATION: $K_n \equiv$ the complete graph with n vertices.



K_4



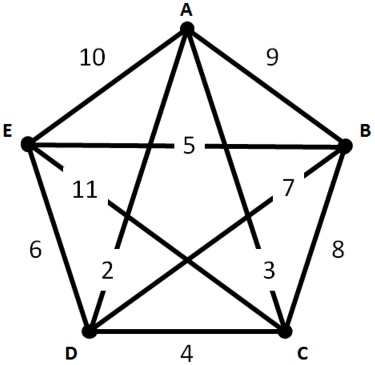
K_5

Weighted Graphs (Definition)

Definition

(Weighted Graph)

A **weighted graph** is a graph with numbers assigned to the edges.
The assigned numbers are called **weights**.
The **weight of a path** is the sum of the weights of the edges in the path.

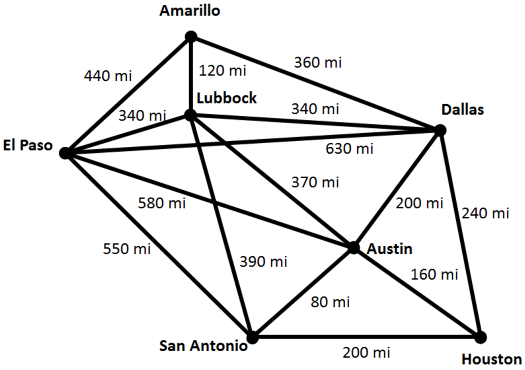


The Traveling Salesperson Problem (TSP)

Definition

(Traveling Salesperson Problem)

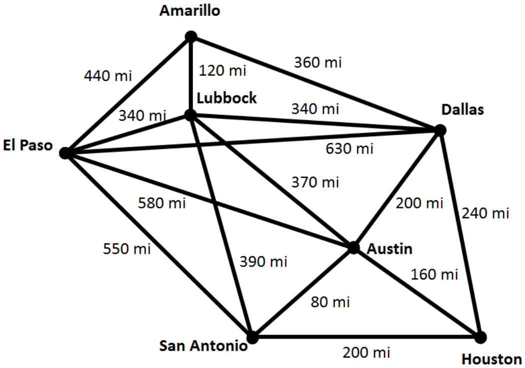
The **Traveling Salesperson Problem (TSP)** is the problem of determing the most efficient way for a salesperson to schedule a trip to a series of cities and then return home.



Definition

(Traveling Salesperson Problem)

Given a connected weighted graph,
find the Hamilton circuit with the smallest weight.



Solving TSP (Brute Force Algorithm)

Proposition

(Brute Force Algorithm)

INPUT: A connected weighted graph.

(1) List all Hamilton circuits in the graph.
(2) Find the weight of each circuit found in (1).

OUTPUT: Optimal Hamilton circuit(s) with the smallest weights

Brute Force Algorithm is Impractical!

Definition

(Factorial) $n! = n(n-1)(n-2)\cdots(4)(3)(2)(1)$ (n is a positive integer)

n	# Edges in K_n	# of Hamilton Circuits in K_n
5	10	$4! = 24$
6	15	$5! = 120$
7	21	$6! = 720$
8	28	$7! = 5040$
9	36	$8! = 40,320$
10	45	$9! = 362,880$
11	55	$10! \approx 3.6$ million
12	66	$11! \approx 40$ million
13	78	$12! \approx 479$ million
14	91	$13! \approx 6.2$ billion
15	105	$14! \approx 87$ billion
16	120	$15! \approx 1.3$ trillion
20	190	$19! \approx 121$ quadrillion

Suppose a computer could find 1000 Hamilton circuits per second.

Brute Force Algorithm is Impractical!

Definition

(Factorial) $n! = n(n-1)(n-2)\cdots(4)(3)(2)(1)$ (n is a positive integer)

n	# Edges	# of Hamilton Circuits in K_n	Time to find all Hamilton circuits
5	10	$4! = 24$	<1 second
6	15	$5! = 120$	<1 second
7	21	$6! = 720$	<1 second
8	28	$7! = 5040$	5 seconds
9	36	$8! = 40,320$	40 seconds
10	45	$9! = 362,880$	6 minutes
11	55	$10! \approx 3.6$ million	1 hour
12	66	$11! \approx 40$ million	11 hours
13	78	$12! \approx 479$ million	5.5 days!
14	91	$13! \approx 6.2$ billion	10 weeks!!
15	105	$14! \approx 87$ billion	2.76 **years!!!**
16	120	$15! \approx 1.3$ trillion	41 **years!!!!**
20	190	$19! \approx 121$ quadrillion	$38,000$ **centuries!!!!!**

Suppose the computer could find 1000 Hamilton circuits per second.

Solving TSP (Nearest Neighbor Algorithm)

Unfortunately, the Brute Force Algorithm is impractical for complete graphs with more than 5 vertices!
There's an algorithm that's much faster, but the solution is only "near optimal":

Proposition

(Nearest Neighbor Algorithm)

<u>INPUT:</u> A connected weighted graph.

(1) Start at any vertex.
(2) Choose the edge joined to the vertex with the smallest weight. Traverse the chosen edge.
(3) Repeat (2) until all vertices have been touched.
(4) Close the circuit by returning to the starting vertex.

<u>OUTPUT:</u> Semi-optimal Hamilton circuit with "nearly" smallest weight.

<u>REMARK:</u> There are other similar fast algorithms, but none of them promise the optimal solution.

Fin.