The Traveling Salesman Problem
Nearest-Neighbor Algorithm
Lecture 30
Sections 6.4

Robb T. Koether
Hampden-Sydney College
Mon, Nov 12, 2018
1. Greedy and Approximate Algorithms

2. The Nearest-Neighbor Algorithm

3. The Repetitive Nearest-Neighbor Algorithm

4. Assignment
Outline

1. Greedy and Approximate Algorithms
2. The Nearest-Neighbor Algorithm
3. The Repetitive Nearest-Neighbor Algorithm
4. Assignment
Greedy Algorithms

Definition (Greedy Algorithms)

A greedy algorithm is an algorithm that, like greedy people, grabs what looks best in the short run, whether or not it is best in the long run.

- Greedy algorithms optimize locally, but not necessarily globally.
- The benefit of greedy algorithms is that they are simple and fast.
- They may or may not produce the optimal solution.
Definition (Approximate Algorithm)

An approximate algorithm is an algorithm that gives a good solution, but not necessarily the best solution.

- The benefit of approximate algorithms is that they can produce a good solution very quickly.
Approximate Algorithms

**Definition (Approximate Algorithm)**

An **approximate algorithm** is an algorithm that gives a good solution, but not necessarily the best solution.

- The benefit of approximate algorithms is that they can produce a good solution very quickly.
- They operate under the principle “Good is good enough.”
Definition (Approximate Algorithm)

An approximate algorithm is an algorithm that gives a good solution, but not necessarily the best solution.

- The benefit of approximate algorithms is that they can produce a good solution very quickly.
- They operate under the principle “Good is good enough.”
- Also known as “The perfect is the enemy of the good.”
Approximate Algorithms

Definition (Approximate Algorithm)

An approximate algorithm is an algorithm that gives a good solution, but not necessarily the best solution.

- The benefit of approximate algorithms is that they can produce a good solution very quickly.
- They operate under the principle “Good is good enough.”
- Also known as “The perfect is the enemy of the good.”
- “Striving to be better, oft we mar what’s well.” (Shakespeare)
We will look at three greedy, approximate algorithms to handle the Traveling Salesman Problem.

- The Nearest-Neighbor Algorithm
- The Repetitive Nearest-Neighbor Algorithm
- The Cheapest-Link Algorithm
Outline

1. Greedy and Approximate Algorithms
2. The Nearest-Neighbor Algorithm
3. The Repetitive Nearest-Neighbor Algorithm
4. Assignment
Definition (Nearest-Neighbor Algorithm)

The Nearest-Neighbor Algorithm begins at any vertex and follows the edge of least weight from that vertex. At every subsequent vertex, it follows the edge of least weight that leads to a city not yet visited, until it returns to the starting point.
The Nearest-Neighbor Algorithm

Definition (Nearest-Neighbor Algorithm)

The Nearest-Neighbor Algorithm begins at any vertex and follows the edge of least weight from that vertex. At every subsequent vertex, it follows the edge of least weight that leads to a city not yet visited, until it returns to the starting point.

- The result typically depends on the chosen starting point.
Example

Start at A
Example

Distance = 10
The Traveling Salesman Problem

Nearest-Neighbor Algorithm

Distance = 27
Distance = 31
Example

Distance = 36
Distance = 46

The Traveling Salesman Problem
Nearest-Neighbor Algorithm

Mon, Nov 12, 2018 9 / 15
Example

Distance = 54
Example

Distance = 65
Example

Distance = 77
Example

Distance = 111
Distance = 116
Example

Distance = 130

Robb T. Koether (Hampden-Sydney College) The Traveling Salesman Problem Nearest-Neighbor Algorithm Mon, Nov 12, 2018 9 / 15
Example (Nearest-Neighbor Algorithm)

- We ended up with the circuit \( ABIGHFCDEJKA \).
- The length is 130 miles.
- Is it possible to do better?
Example (Nearest-Neighbor Algorithm)

- We ended up with the circuit \( ABIGHFCDJJEKA. \)
- The length is 130 miles.
- Is it possible to do better?
- Yes.
Re-do the previous example, starting at city B.
Re-do the previous example, starting at city C.
Outline

1. Greedy and Approximate Algorithms
2. The Nearest-Neighbor Algorithm
3. The Repetitive Nearest-Neighbor Algorithm
4. Assignment
Definition (Repetitive Nearest-Neighbor Algorithm)

The Repetitive Nearest-Neighbor Algorithm applies the nearest-neighbor algorithm repeatedly, using each of the vertices as a starting point. It selects the starting point that produced the shortest circuit.
Assignment

- Chapter 6: Exercises 33, 34, 35, 36, 37, 40, 41, 44