Autonomous Mobile Robot Design
Topic: Traveling Salesman Problem
Dr. Kostas Alexis (CSE)
The Traveling Salesman Problem

- **Definition**: A complete graph $K_N$ is a graph with $N$ vertices and an edge between every two vertices.
The Traveling Salesman Problem

- **Definition:** A complete graph $K_N$ is a graph with $N$ vertices and an edge between every two vertices.

- **Definition:** A Hamiltonian circuit is a circuit that uses every vertex of a graph once.
The Traveling Salesman Problem

- **Definition:** A complete graph $K_N$ is a graph with $N$ vertices and an edge between every two vertices.
- **Definition:** A Hamiltonian circuit is a circuit that uses every vertex of a graph once.
- **Definition:** A weighted graph is a graph in which each edge is assigned a weight (representing the time, distance, or cost of traversing that edge).
The Traveling Salesman Problem

- **Definition:** A **complete graph** $K_N$ is a graph with $N$ vertices and an edge between every two vertices.
- **Definition:** A **Hamiltonian circuit** is a circuit that uses every vertex of a graph once.
- **Definition:** A **weighted graph** is a graph in which each edge is assigned a weight (representing the time, distance, or cost of traversing that edge).
- **Definition:** The **Traveling Salesman Problem (TSP)** is the problem of finding a minimum-weight Hamiltonian circuit in $K_N$. 
The Traveling Salesman Problem

Example: Simone has the following list of errands:
- Pet store (P)
- Greenhouse (G)
- Cleaners (C)
- Drugstore (D)
- Target (T)

In which order should she do these errands in order to minimize the time spent on her broom?
The Traveling Salesman Problem
The Traveling Salesman Problem
The Traveling Salesman Problem

Hamilton circuit: HDTGPCH
The Traveling Salesman Problem

Hamilton circuit: HDTPCGH
The Traveling Salesman Problem

Hamilton circuit:
HCDTPGH
The Traveling Salesman Problem

- The number of vertices is $N=6$, so...
The Traveling Salesman Problem

- The number of vertices is $N=6$, so...
  - the number of Hamilton circuits is $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$

The general rule is $(N - 1)!$
The Traveling Salesman Problem

- What we have just done is the **Brute-Force Algorithm**:
  - Make a list of all possible Hamiltonian circuits
  - Calculate the weight of each Hamiltonian circuit by adding up the weights of its edges
  - Choose the Hamiltonian circuit with the smallest total weight

- The Brute-Force Algorithm is optimal: it is guaranteed to find the best solution possible.
  - But it is extremely inefficient: It has to calculate \((N - 1)!\) Hamiltonian circuits and this can take a long time.
The Traveling Salesman Problem

- A rough, suboptimal heuristic: **Nearest-Neighbor Algorithm**
  - At each stage of the tour, choose the closest vertex that you have not visited yet.
The Traveling Salesman Problem

Consider the following:

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>P</th>
<th>G</th>
<th>C</th>
<th>D</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home (H)</td>
<td>0</td>
<td>36</td>
<td>32</td>
<td>54</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Pet store (P)</td>
<td>36</td>
<td>0</td>
<td>22</td>
<td>58</td>
<td>54</td>
<td>67</td>
</tr>
<tr>
<td>Greenhouse (G)</td>
<td>32</td>
<td>22</td>
<td>0</td>
<td>36</td>
<td>42</td>
<td>71</td>
</tr>
<tr>
<td>Cleaners (C)</td>
<td>54</td>
<td>58</td>
<td>36</td>
<td>0</td>
<td>50</td>
<td>92</td>
</tr>
<tr>
<td>Drugstore (D)</td>
<td>20</td>
<td>54</td>
<td>42</td>
<td>50</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>Target (T)</td>
<td>40</td>
<td>67</td>
<td>71</td>
<td>92</td>
<td>45</td>
<td>0</td>
</tr>
</tbody>
</table>
The Traveling Salesman Problem

Consider the following:

If Simone starts at home, the closest destination is the drugstore.
So, maybe the Hamiltonian circuit should start from the edge H-D

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>P</th>
<th>G</th>
<th>C</th>
<th>D</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home (H)</td>
<td>0</td>
<td>36</td>
<td>32</td>
<td>54</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Pet store (P)</td>
<td>36</td>
<td>0</td>
<td>22</td>
<td>58</td>
<td>54</td>
<td>67</td>
</tr>
<tr>
<td>Greenhouse (G)</td>
<td>32</td>
<td>22</td>
<td>0</td>
<td>36</td>
<td>42</td>
<td>71</td>
</tr>
<tr>
<td>Cleaners (C)</td>
<td>54</td>
<td>58</td>
<td>36</td>
<td>0</td>
<td>50</td>
<td>92</td>
</tr>
<tr>
<td>Drugstore (D)</td>
<td>20</td>
<td>54</td>
<td>42</td>
<td>50</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>Target (T)</td>
<td>40</td>
<td>67</td>
<td>71</td>
<td>92</td>
<td>45</td>
<td>0</td>
</tr>
</tbody>
</table>
The Traveling Salesman Problem
The Traveling Salesman Problem

- Eventually, we end up with the Hamiltonian circuit:
  \[ H \rightarrow T \rightarrow C \rightarrow P \rightarrow G \rightarrow D \rightarrow H \]

- Weight of this circuit: 274
- Weight of an optimal circuit: 229
- Average weight of a circuit: 287.6
The Traveling Salesman Problem

- The Nearest-Neighbor Algorithm is efficient but **suboptimal**
  - It is quick and easy, but does not typically find the lowest-weight Hamiltonian circuit.
Code Examples and Tasks

- [https://github.com/unr-arl/LKH_TSP](https://github.com/unr-arl/LKH_TSP)
Thank you!

Please ask your question!