HEURISTICS STUDY FOR THE TRAVELING SALESMAN PROBLEM

Joaquim Meireles

pro09011@fe.up.pt
Traveling Salesman Problem (TSP) Overview

Exact Methods

Heuristics

Constructive Heuristic: Nearest Neighbour

Improvement Heuristic: 2-Optimal

Metaheuristic: Tabu-Search

Simulation

Results

Conclusions
Traveling Salesman Problem: Overview

• Goal: find the shortest route in a set of cities
  • Every city must be visited only once
  • Starts and ends in the same city

• Well-known NP-hard combinatorial optimization problem

• Practical Applications:
  • School Schedule
  • Tourist Itineraries...
Solving TSP: Exact Methods

• Tries to get an **optimal** tour for a problem
• Computational time is a problem... **why**?

• Example:

[Diagram showing points A, B, and C]

Possible Combinations:

<table>
<thead>
<tr>
<th>Combination</th>
<th>Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-B-C-A</td>
<td>B-C-A-B</td>
</tr>
<tr>
<td>A-C-B-A</td>
<td>C-A-B-C</td>
</tr>
<tr>
<td>B-A-C-B</td>
<td>C-B-A-C</td>
</tr>
</tbody>
</table>

\[=3!\Rightarrow 6\]

Brute force Method = N!

11! = 39916800 possible combinations
Solving TSP: Heuristics

- Objective is not only the quest for the optimal solution but...
  - Tries to achieve an “approximated optimal value”;
- Useful when optimal values are not essential
- “Short” running times;

- **Heuristics Types**: 
  - Constructive Heuristics
  - Improvement Heuristics
  - Metaheuristics
Constructive Heuristic

- **Nearest Neighbour**
  - Fast execution:
    - Determines a tour according to some construction rules
    - Final tour order depends on the start city
    - Provides a good initial solution for 2-Optimal

- Function mode:
  - Starts at a city, and selected the nearest unvisited city;
  - Repeat until all cities are visited.
Improvement Heuristic

• 2-Optimal
  • K-Opt family
  • Tries to improve solution provided by NN Algorithm
  • Removes two edges AB and CD, and in their place inserts the two edges AC and BD, if \( d(a, c) + d(b, d) < d(a, b) - d(c, d) \);
  • This action reduces the length of the tour while maintaining a connected path through all points.

The initial tour is shown on the left, and the final tour on the right.
Metaheuristic

• Tabu-Search (Over 2-Optimal)
  • Perform a movement even if it does not improve the value of the objective;
  • More precisely, pick the best solution in the neighborhood even if it is not improving;
• Tabu Lists
  • Short-term memory;
  • Forbid tabu movements unless one of them provides a better solution than the currently best one (aspiration criteria);

• Intensification:
• Diversification.
• Stop Criteria
Simulation

• Objectives:
  • Compare obtained values for each heuristic;
  • Compare final solution with optimal value.

• Tests Data obtained from TSPLIB
  • Each library has a known optimal value.

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>Nr Executions</td>
<td>Total number of the test library’s cities</td>
</tr>
<tr>
<td>Tabu-Search</td>
<td>Nr Executions</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>Number Iterations</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Iterations without</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>improvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tabu list size (TL)</td>
<td>5 and 7</td>
</tr>
</tbody>
</table>
Results

• Each heuristic has a limit value (in this study). Why?

• Similar average for Tabu-Search with different TL sizes, but...

• First good approach

Average obtained results compared to known optimal value
Conclusions

• Exact solutions are desired, but sometimes not realistic;

• Heuristics approach provides “good enough“ solutions for practical applications
  • Often the search for an optimal solution is not even worth the effort.

Future Work

• Introduce computational time variable for each algorithm;

• More specific study on Tabu-Search:
  • Tabu list sizes;
  • Intensification;
  • Diversification.

• Study with greater heuristics variety
Questions