Heuristic Algorithms for Bike Route Generation

Aidan Pieper
Matthew Anderson (Advisor)

Computer Science Department, Union College

March 3, 2018
Introduction

- Routing for recreational cyclists is different than traditional routing problems.
- Cyclists prefer longer more scenic routes, not the shortest one.
- Our focus is on circular routes.

Figure 1: Circular bike route
Informal Problem Statement

Given:
- A road network
- A starting location
- A distance budget

Goal: Find the “best” bike route which starts and ends at the specified location and is no longer than the budget.
Previous literature models this problem as an instance of the **Arc Orienteering Problem (AOP).**

**Figure 2**: AOP Instance - Edge label: \((score, \ cost)\) Budget: 10
Figure 3: Shortest Path: \((\text{score} = 15, \text{cost} = 8)\) Budget: 10
Figure 4: Optimal Path: \( (\text{score} = 30, \ \text{cost} = 10) \) Budget: 10
The AOP is NP-Hard:
- Our focus is on heuristic algorithms for the AOP.
- **Iterated Local Search (ILS)** is the algorithm of interest.

**Research Question:**
To what extent can ILS algorithms be improved to generate better bike routes?

We implemented two ILS algorithms using:
- **GraphHopper**: An open source routing library.
- **OpenStreetMaps**: An open mapping dataset.
Figure 5: Shortest path Union → Saratoga Springs
Uses modified **Depth First Search** with max depth.

- Precomputes all-pairs shortest path for feasibility checking.
- Returns first path found fitting criteria.

\[(S \rightarrow v_1).cost + a.cost + \text{ShortestPath}(v_2, D) \leq \text{Budget}\]

**Figure 6**: Arc feasibility checking
Figure 7: DFS Algorithm Example Route
DFS Algorithm [VVA14]

Limitations:

- Search space large in road dense areas.
- Requires pre-computed all-pairs shortest path.
- Does not penalize turns.

Figure 8: Dangerous route turn
Geometric Algorithm [LS15]

- Generates paths by “gluing together” Attractive Arcs from a Candidate Arc Set.
- Uses spatial techniques to reduce search space.
- Uses online shortest path computations [GSSD08].

$\text{Figure 9: Ellipse pruning technique}$
Figure 10: Perfectly circular route generated by Geometric Algorithm.
Figure 11: Route with backtracking generated by Geometric Algorithm.
Figure 12: Route with excess backtracking by Geometric Algorithm.
Limitations:

- Does not avoid backtracking.
- Tries to hit budget exactly.
- Shortest path not necessarily preferable.
- Does not penalize turns.

We designed and implemented variants:

- Avoid backtracking when glueing together attractive arcs.
- Don’t use full budget when generating paths.
- Change which attractive arcs are considered.
Figure 13: Route generation with DFS Algorithm.
Results: Geometric [LS15]

![Graph showing Iteration Number vs. Average Score and Time (s) for Route generation with Geometric Algorithm.](image)

**Figure 14:** Route generation with Geometric Algorithm.
Results: Geometric + (Budget allowance)

Figure 15: Geometric Algorithm with 50% budget allowance.
Conclusions

- Spatial techniques definitely speed up ILS.
- Modifying budget over time greatly increases average score at a hefty time penalty.
- Attractive arc definition and data set matter a lot in algorithm performance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Score</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFS</td>
<td>20.57</td>
<td>20.37</td>
</tr>
<tr>
<td>Geometric</td>
<td>126.13</td>
<td>1.20</td>
</tr>
<tr>
<td>Geometric + (Budget allowance)</td>
<td>215.87</td>
<td>23.12</td>
</tr>
<tr>
<td>Geometric + (Incremental budget)</td>
<td>282.66</td>
<td>119.52</td>
</tr>
<tr>
<td>Geometric + (Arc restrictions)</td>
<td>49.85</td>
<td>0.09</td>
</tr>
<tr>
<td>Geometric + (No backtracking)</td>
<td>33.36</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Figure 16: Algorithm performance of variants.
Major kudos to **David Frey** for helping me set up computing resources to run my experiments!

I glossed over a lot of technical details! Ask me about the following:

- Road scoring
- OpenStreetMap dataset
- Online shortest path computation (Contraction Hierarchies)
- Iterated Local Search
- Details of Algorithm 1 & 2
- Integer Programming solutions to the AOP
References


Given:

- An incomplete directed graph $G = (V, A)$
- A start vertex $d \in V$
- A distance budget $B \in \mathbb{R}$.

Each arc, $a \in A$ has the following:

- A cost $c_a \in \mathbb{R}$
- A profit $p_a \in \mathbb{R}$
- A complementary arc $\bar{a} \in A \cup \{\emptyset\}$

Decision variables:

- $x_a \in \{0, 1\}, \forall a \in A$
- $z_v \in \mathbb{Z}^+, \forall v \in V$

**Objective:** Maximize $\sum_{a \in A} p_a * x_a$ (1)
Given: $\delta(S) =$ set of outgoing arcs, $\lambda(S) =$ set of incoming arcs.

\[
\sum_{a \in A} c_a \cdot x_a \leq B \quad (2)
\]

\[
\sum_{a \in \lambda(v)} x_a - \sum_{a \in \delta(v)} x_a = 0 \quad \forall v \in V \quad (3)
\]

\[
\sum_{a \in \delta(v)} x_a = z_v \quad \forall v \in V \quad (4)
\]

\[
\sum_{a \in \delta(S)} x_a \geq \frac{\sum_{v \in S} z_v}{\sum_{v \in S} |\delta(v)|} \quad \forall S \subseteq V \setminus \{d\} \quad (5)
\]

\[
z_d = 1 \quad (6)
\]

\[
x_a + x_{\bar{a}} \leq 1 \quad \forall a \in A : \exists \bar{a} \in A \quad (7)
\]