Heuristic Algorithms for Bike Route Generation

Aidan Pieper
Matthew Anderson (Advisor)

Computer Science Department, Union College

March 3, 2018

- 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.

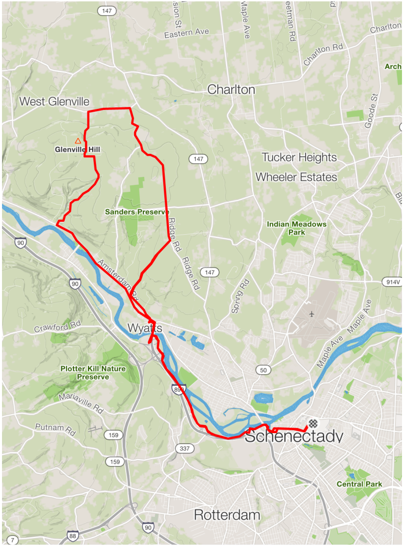


Figure 1: Circular bike route

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.

Related Work

Previous literature models this problem as an instance of the **Arc Orienteering Problem (AOP)**.

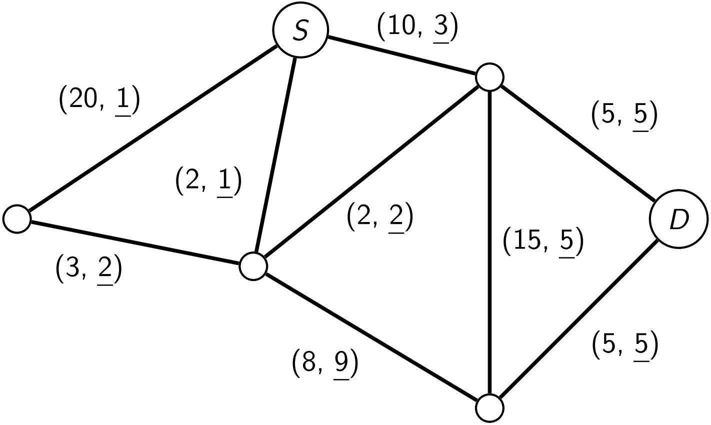


Figure 2: AOP Instance - Edge label: (*score*, cost) Budget: 10

Arc Orienteering Example

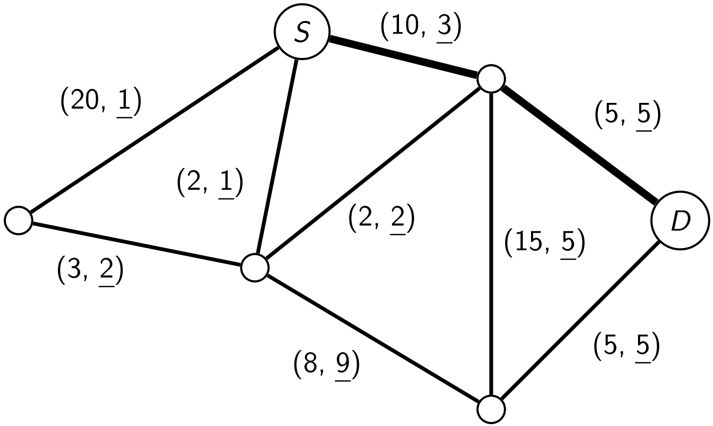


Figure 3: Shortest Path: ($score = 15$, $\underline{cost = 8}$) Budget: 10

Arc Orienteering Example

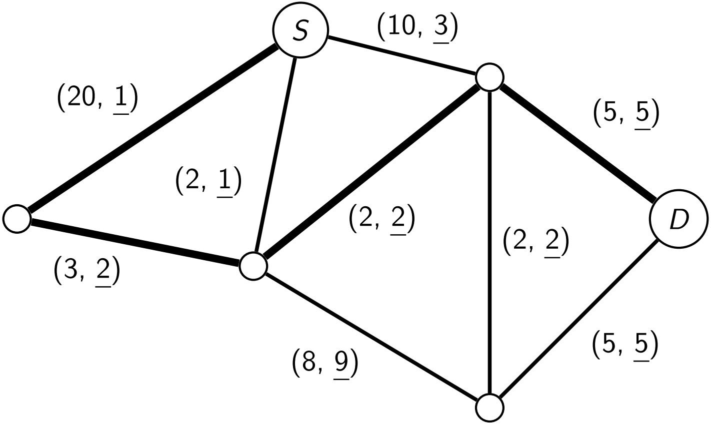


Figure 4: Optimal Path: ($score = 30$, $\underline{cost = 10}$) Budget: 10

Methods

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.

Methods: GraphHopper Routing Engine

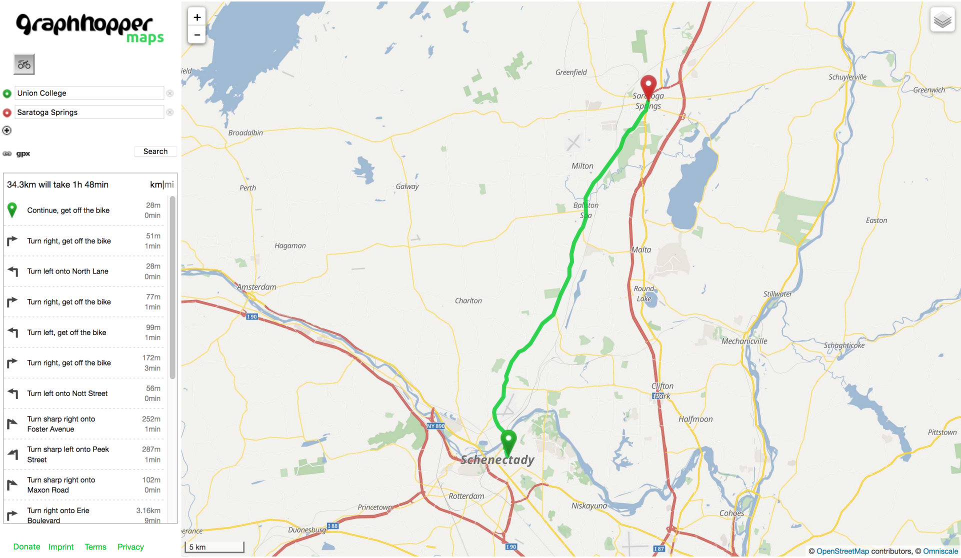
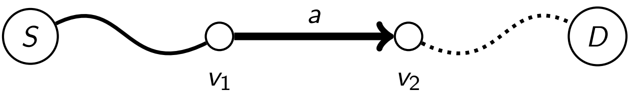


Figure 5: Shortest path Union → Saratoga Springs

DFS Algorithm [VVA14]

- Uses modified **Depth First Search** with max depth.
- Precomputes all-pairs shortest path for feasibility checking.
- Returns first path found fitting criteria.



$$(S \rightarrow v_1).cost + a.cost + ShortestPath(v_2, D) \leq Budget$$

Figure 6: Arc feasibility checking

DFS Algorithm [VVA14]

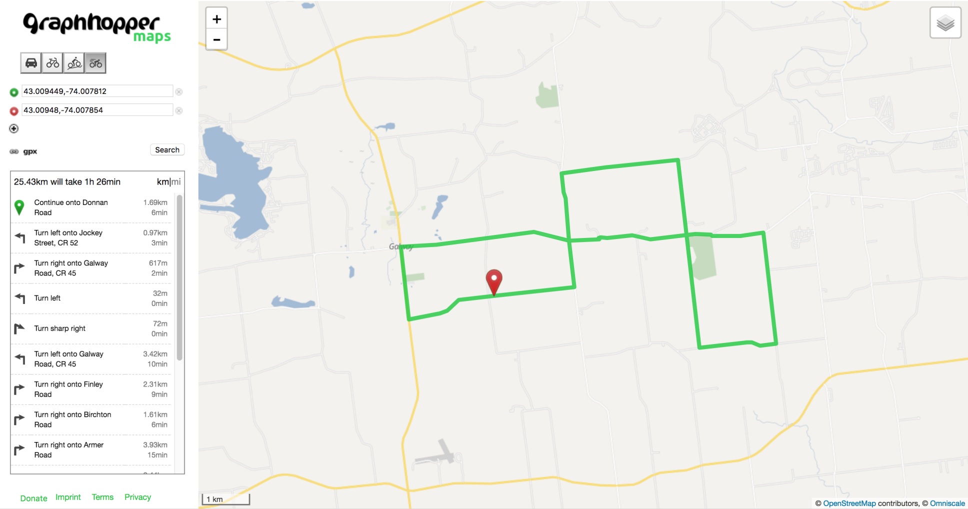


Figure 7: DFS Algorithm Example Route

DFS Algorithm [VVA14]

Limitations:

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

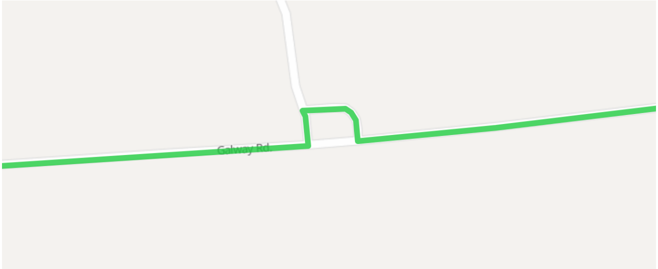


Figure 8: Dangerous route turn

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].

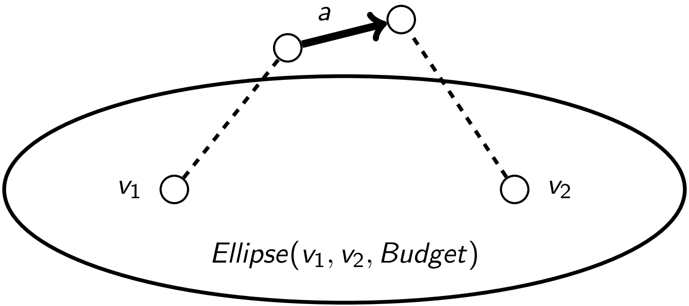


Figure 9: Ellipse pruning technique

Geometric Algorithm [LS15]

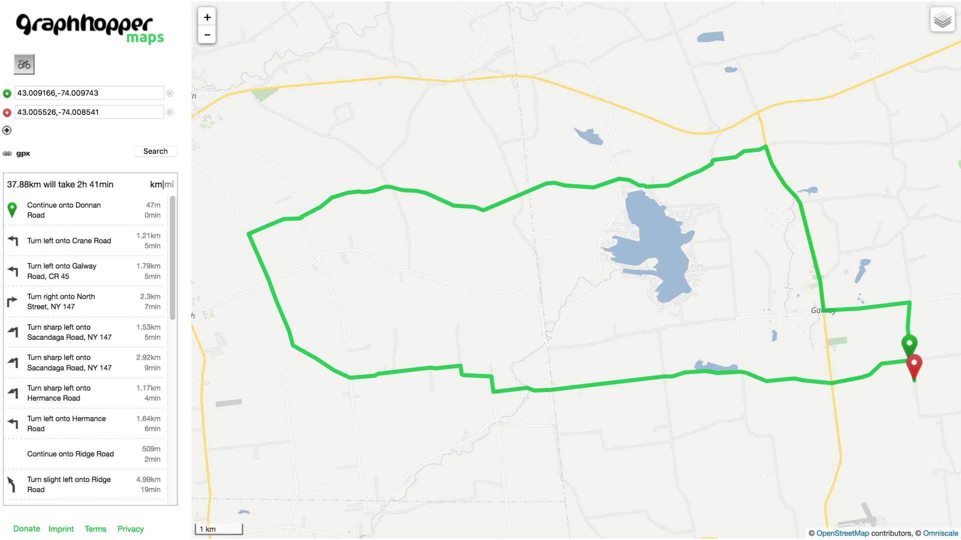


Figure 10: Perfectly circular route generated by Geometric Algorithm.

Geometric Algorithm [LS15]

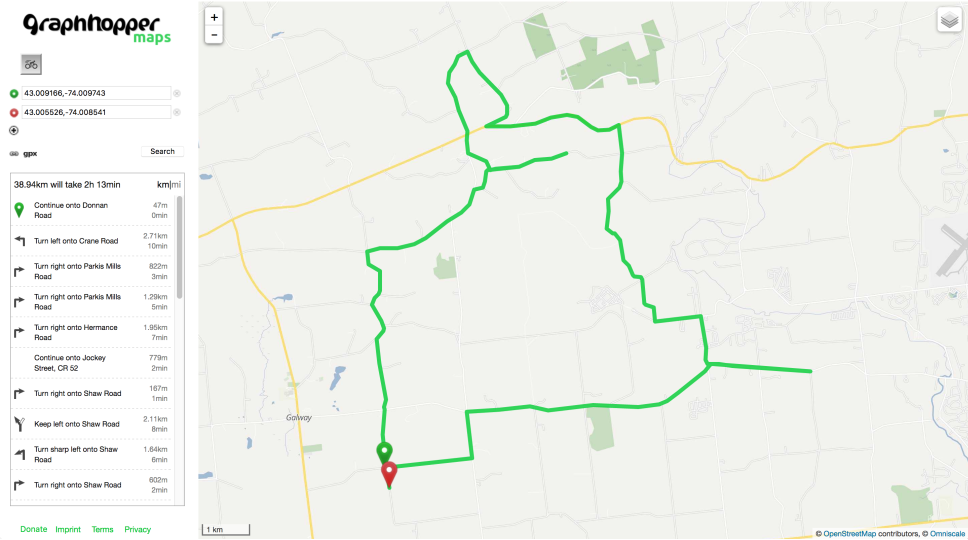


Figure 11: Route with backtracking generated by Geometric Algorithm.

Geometric Algorithm [LS15]

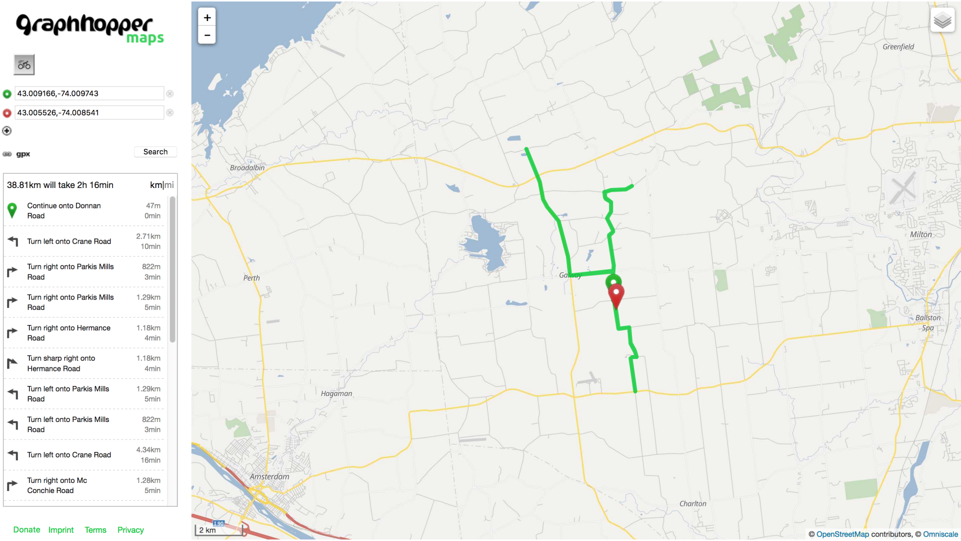


Figure 12: Route with excess backtracking by Geometric Algorithm.

Geometric Algorithm [LS15]

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 gluing together attractive arcs.
- Don't use full budget when generating paths.
- Change which attractive arcs are considered.

Results: DFS [VVA14]

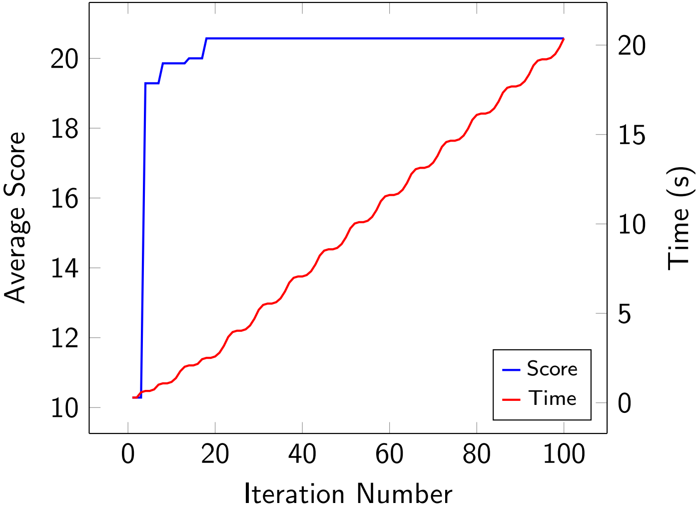


Figure 13: Route generation with DFS Algorithm.

Results: Geometric [LS15]

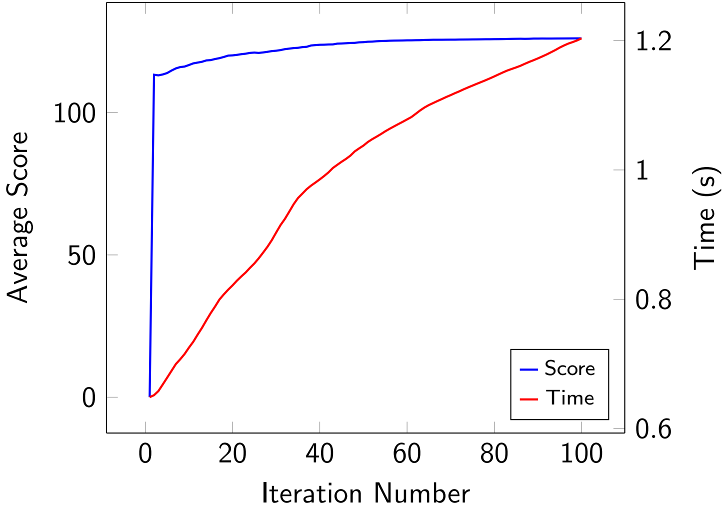


Figure 14: Route generation with Geometric Algorithm.

Results: Geometric + (Budget allowance)

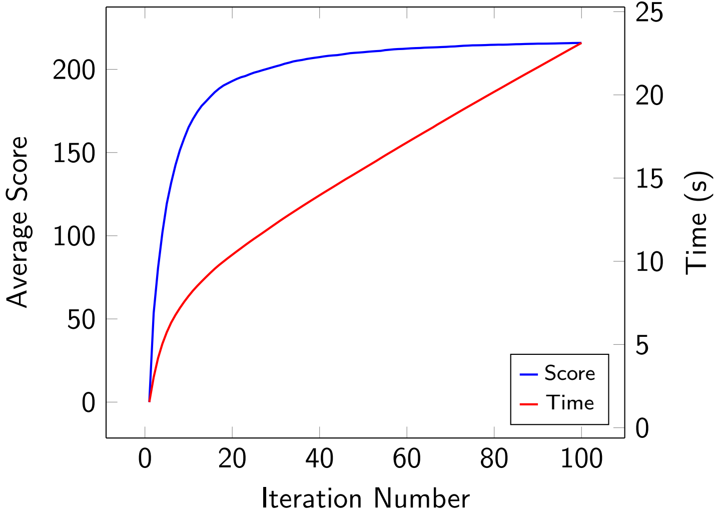


Figure 15: Geometric Algorithm with 50% budget allowance.

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.

Algorithm	**Score**	**Time (s)**
DFS	20.57	20.37
Geometric	126.13	1.20
Geometric + (Budget allowance)	215.87	23.12
Geometric + (Incremental budget)	282.66	119.52
Geometric + (Arc restrictions)	49.85	0.09
Geometric + (No backtracking)	33.36	0.60

Figure 16: Algorithm performance of variants.

Acknowledgements & Comments

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

[GLV16] Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

[GP10] Michel Gendreau and Jean-Yves Potvin. *Handbook of Metaheuristics*, volume 2. Springer, 2010.

[GRA] GraphHopper Routing Engine. `https://github.com/graphhopper/graphhopper`. Visited Nov 2, 2017.

[GSSD08] Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. *Experimental Algorithms*, pages 319–333, 2008.

[LS15] Ying Lu and Cyrus Shahabi. An arc orienteering algorithm to find the most scenic path on a large-scale road network. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 46. ACM, 2015.

[OSM] OpenStreetMap Wiki. `http://wiki.openstreetmap.org/wiki/Develop`. Visited Nov 13, 2017.

[SVBVO11] Wouter Souffriau, Pieter Vansteenwegen, Greet Vanden Berghe, and Dirk Van Oudheusden. The planning of cycle trips in the province of East Flanders. *Omega*, 39(2):209–213, 2011.

[VVA14] Cédric Verbeeck, Pieter Vansteenwegen, and E-H Aghezzaf. An extension of the arc orienteering problem and its application to cycle trip planning. *Transportation Research Part E: Logistics and Transportation Review*, 68:64–78, 2014.

Integer Program Formulation [VVA14]

Given:

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

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

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

Decision variables:

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

$$\textit{Objective:} \text{ Maximize} \sum_{a \in A} p_a * x_a \tag{1}$$

Integer Program Constraints

Given: $\delta(S)$ = set of outgoing arcs, $\lambda(S)$ = set of incoming arcs.

$$\sum_{a \in A} c_a * x_a \leq \mathrm{B} \tag{2}$$

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

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

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

$$z_d = 1 \tag{6}$$

$$x_a + x_{\bar{a}} \leq 1 \quad \forall a \in A : \exists \bar{a} \in A \tag{7}$$