Tour4Me: A Framework for Customized Tour Planning Algorithms

Kevin Buchin
TU Dortmund
Dortmund, Germany
kevin.buchin@tu-dortmund.de

Mart Hagedoorn
TU Dortmund
Dortmund, Germany
mart.hagedoorn@tu-dortmund.de

Guangping Li
TU Dortmund
Dortmund, Germany
guangping.li@tu-dortmund.de

ABSTRACT

The touring problem aims to find an 'interesting' (round) trip of a given length. Here, what is considered interesting depends on the type of the desired route, e.g., a user may be looking for a off-road cycling trip or fast running route. There are two main perspectives on the touring problem, maximizing profit or minimizing cost, which result in very different algorithmic solutions. We provide a framework that allows for straightforward integration of new algorithms for both perspectives on the touring problem. In this demonstration we have included a new exact solver, a heuristic, and two greedy methods. The user can experiment with the algorithms and different profits/costs. The generated tours can be explored in an easy-to-use web interface.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Kevin Buchin, Mart Hagedoorn, and Guangping Li. 2018. Tour4Me: A Framework for Customized Tour Planning Algorithms. *J. ACM* 37, 4, Article 111 (August 2018), 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Most people who do outdoor activities run into the problem of finding an appropiate route. Depending on the activity from hiking and jogging to gravel and road cycling, requirements from users can greatly vary. To this end we have developed Tour4Me. The tool Tour4Me constists out of an intuitive UI that allows users to create tours customed to their specific demands in their own webbrowser. Furthermore, Tour4Me contains a few algorithms for computing solutions for the arc orienteering problem (AOP) and the more general touring problem.

Authors' addresses: Kevin Buchin, TU Dortmund, Dortmund, Germany, kevin.buchin@tu-dortmund.de; Mart Hagedoorn, TU Dortmund, Dortmund, Germany, mart.hagedoorn@tu-dortmund.de; Guangping Li, TU Dortmund, Dortmund, Germany, guangping.li@tu-dortmund.de.

0004-5411/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

For the touring problem there exist two distinct perspectives. The first is the perspective of maximizing profit as with the arc orienteering problem (AOP). Given a length budget and an profit function over every edge, find a route that does not exceed the budget and maximizes the total profit. An important detail to this approach is that the profit of an edge is only counted once, whilst the cost of an edge is added every time. In an effort of finding better and nicer tours more attributes could be maximized. For example, we can maximize the total area covered by the tour, to get a tour that is closed to a circle as possible, or minimize turn cost as this requires slowing down.

The second perspective is perhaps more intuitive and involves minimizing cost. Here edges that are desirable are reduced in cost resulting in a new graph representing the underlying streets. A shortest path algorithm can be used to find a route between two points which will result in a route where desired edges are prioirtized by the algorithm as they are less costly to traverse. In order to create a cyclic tour with the minimizing cost approach, intermediate waypoints need to be choosen. Between these waypoints a shortest path calculation is performed resulting in the

1.1 Related Work

1.2 Contribution

2 SYSTEM

The main contribution of this work is the framework that allows for easy

2.1 Architecture

2.2 Data

To represent streets and roads as a graph we use the data from OpenStreetMap[1]. The street data is thereafter processed by OSMnx []. The

2.2.1 Backbone. In order to get exact solutions for routes of length larger than 2 kilometers we can choose to run these exact algorithms on a simplified graph. We include roads in our backbone if they are part of a registered bike route in OSM. Registered bike routes are usually routes maintained by local and regional governments. Therefore, for cycling, these usually give a reasonable network of roads and cycle ways that are fit to cycle on.

[1] https://www.openstreetmap.org

2.3 Interface

3 ALGORITHM

For this iteration we have implemented four dirrent algorithms for calulation and improvement of routes.

3.1 Greedy Selection

The greedy selection algorithm uses a straightforward greedy choice for selecting

3.2 Jogging Tour

3.3 Iterative Local Search

We have implemented the iterative local search from Verbeeck et al. [?]. We however do not use the initialization phase of their algoirthm as this does not produce suffiently good results for us. Therefore, we run the Jogging tour alogirthm from Section 3.2 to obtain an intial solution. This solution is thereafter improved by the iterative local search using the time budget given by the user.

3.4 Integer Linear Programming

The integer linear program (ILP) gives the optimal solution for an instance of the AOP. The ILP used in Tour4Me is a modified version from Verbeeck et al. The ILP from [] introduces a constraint for every subset of the vertices in order to avoind disconnected components, resulting in $O(2^n)$ constrains.

$$somethingBad \tag{1}$$

The ILP from [] uses Equation 1 to avoid subcycles. Instead we introduce a variable ρ_{kij}, for $1 \leq k \leq L$ and $1 \leq i, j \leq m$. Variable ρ_{kij} denotes whether edge e_{ij} is included in the path at location k.

$$\sum_{i=1}^{m} \sum_{j=1}^{m} \rho_{kij} = 1 \qquad \forall 1 \leq k \leq L \tag{2}$$

$$\sum_{k=1}^{L} \rho_{kij} = \begin{cases} h_{ij} & \text{if } e_{ij} \text{ is an edge} \\ 0 & \text{otherwise} \end{cases} \qquad \forall 1 \leq i, j \leq m \tag{3}$$

$$2 \cdot \rho_{kij} \leq p[k][i] + p[k+1][j] \tag{4}$$

We include Constraint 2 for every $1 \leq k \leq L$ so that the path only has one edge at every position.

Constraint 3.

4 CONCLUSION