JOURNAL OF SPATIAL INFORMATION SCIENCE
Number 26 (2023), pp. 99–124

doi:10.5311/JOSIS.2023.26.256

RESEARCH ARTICLE

Inferring routing preferences from user-generated trajectories using a compression criterion

Axel Forsch, Johannes Oehrlein, Benjamin Niedermann, and Jan-Henrik Haunert

Working Group Geoinformation, University of Bonn, Germany

Received: September 22, 2022; returned: December 18, 2022; revised: February 5, 2023; accepted: March 15, 2023.

Abstract: The optimal path between two vertices in a graph depends on the optimization objective, which is often defined as a weighted sum of multiple criteria. When integrating two criteria, their relative importance is expressed with a balance factor α. We present a new approach for inferring α from trajectories. The core of our approach is a compression algorithm that requires a graph G representing a transportation network, two edge costs modeling routing criteria, and a path P in G representing the trajectory. It yields a minimum subsequence S of the sequence of vertices of P and a balance factor α, such that the path P can be fully reconstructed from S, G, its edge costs, and α. By minimizing the size of S over α, we learn the balance factor that corresponds best to the user's routing preferences. In an evaluation with crowd-sourced cycling trajectories, we weigh the usage of official signposted cycle routes against other routes. More than 50% of the trajectories can be segmented into five optimal sub-paths or less. Almost 40% of the trajectories indicate that the cyclist is willing to take a detour of 50% over the geodesic shortest path to use an official cycle path.

Keywords: trajectory mining, bicriteria optimization, compression, routing preferences, volunteered geographic information

1 Introduction

Route planning tools play an important role in modern mobility. They are not only used for commuting, i.e., to find the best path from one location to another, but also for leisure

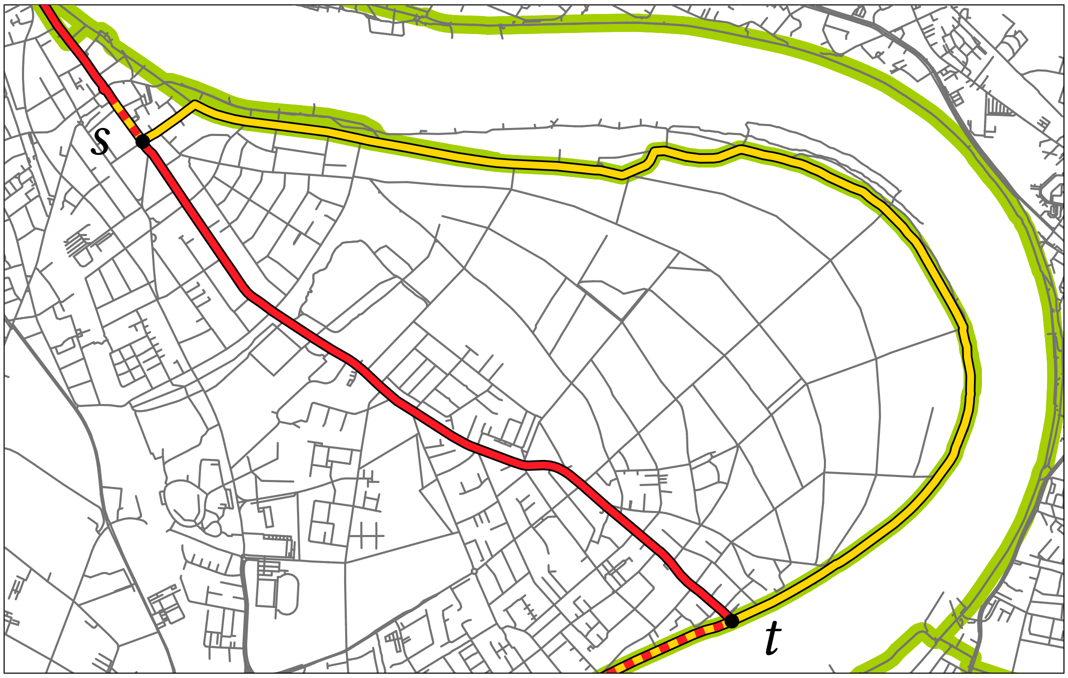


Figure 1: Extract of the road map of the German city of Cologne. Segments of the road network (gray) which are part of official cycle routes are presented with a green background. Furthermore, there are two bicycle trajectories with different s-t paths (red and yellow). The red trajectory is optimal with respect to distance and avoids official cycle routes completely. It has a length of $3.4\,\mathrm{km}$. The cyclist of the yellow trajectory used a path with a length of $5.9\,\mathrm{km}$. It consists mainly of official cycle routes, namely 95% of its length.

activities, such as scenic cycling round trips that start and end at the same location. For both applications, choosing the optimization objective, i.e., the definition of "best" such that it reflects a user's preferences, is crucial. Classically, an optimal path from s to t is the geodesic shortest path in a road network, i.e., it is optimal with respect to the Euclidean length of the path's road links or the path with the smallest travel time. However, many more criteria (e.g., number of traffic lights, total ascent) can be applied. Hence, the definition of an optimal path varies widely. To provide users with good route planning, we must work with their personal preferences. Take the bicycle trajectories depicted in Figure 1 as an example. Between s and t, the red trajectory corresponds to the length-minimal path in the road network. The question arises why the cyclist who recorded the yellow trajectory prefers a slightly longer trip over the length-minimal path. In the concrete example, the cyclist used a path that, in contrast to the length-minimal path, consists of paths that are part of signposted cycle routes (green). Thus, a likely explanation is that the cyclist tried to use recommended paths, still trying to keep the length of the trip short.

While this observation applies to that particular trajectory, a systematic analysis of hundreds of trajectories promises statistical evidence on the chosen optimization criteria and the extent to which the criteria have been taken into account. These insights into the users' preferences and priorities concerning the optimization criteria may help both: individual route planning and infrastructure planning by appropriately balancing the applied optimization criteria. In the following, we distinguish between implicit and explicit routing preferences. In the example given in Figure 1, it may be valid to conclude that the cyclist who generated the yellow trajectory had an *implicit* preference for signposted cycle paths since the trajectory data suggests a strong bias towards such paths. However, the *explicit* preferences of the cyclist might have been to choose scenic routes or routes with low traffic volumes—conditions that are often found at signposted paths. In our work, we focus on inferring the user's implicit preferences based on their recorded trajectories.

In the context of Volunteered Geographic Information (VGI [21]), large sets of crowdsourced trajectories are collected. A key characteristic of these datasets is that they contain many trajectories of recreational activities. The aim of these recreational activities often is not to minimize a certain optimization criterion but to have a nice route of a specific length. As such, they contain large detours that often cannot be explained by a single (combined) optimization criterion. Thus, many established approaches for preference inference fail to analyze these trajectories. Take the example of a round trip: the corresponding trajectories start and end at the same location. Therefore, they are not optimal for any optimization criterion. Nonetheless, it seems natural that the user stuck to favorable roads as much as possible, resulting in a trajectory that consists of multiple optimal subtrajectories. To analyze the rich source of information VGI sources provide concerning the underlying implicit optimization criteria, appropriate tools are needed. In this article, we suggest such a tool. Our approach is to determine the parameter of a routing model such that a given trajectory (or set of trajectories) can be partitioned into a minimum number of paths that are optimal with respect to that model. We base our approach on the findings of Gotsman and Kanza [22] and Lerin et al. [31], who have independently of each other shown how to compute a compact representation of a user's path by partitioning it into a minimum number of optimal paths. The authors have noted that the better the optimization criterion used by the optimal-path algorithm matches the user's routing preferences, the more compact representations are obtained. In this article, we reverse this argument by arguing that if a routing model induces a partition of a trajectory into a small number of optimal paths, then it reflects the user's preferences well. This adheres to the Minimum Description Length Principle, introduced by Grünwald et al. [24], which states that the best explanation for a given data set is the one with the highest compression. With our approach, we aim to infer routing preferences from a wide range of trajectories, e.g., trajectories of commuters as well as trajectories of recreational cyclists. In particular, we neither require that the trajectories have been chosen to minimize a specific optimization criterion nor do we exclude such trajectories.

A preliminary version of the algorithm presented in this publication has already been used by Oehrlein et al. [36, 37] to analyze the influence of the slope on bicycle routing and to approve a classification of road types into favorable and unfavorable types. Both of these publications focus on the application of the algorithm without explaining its functionality in detail. In this article, we give for the first time a detailed presentation of the algorithm. Additionally, we improved the computational complexity of the algorithm. We present the application of the algorithm to the problem of analyzing whether and how cyclists prefer signposted cycling routes over other routes. In our experiments, our algorithm classifies the trajectories in the data set into three classes, with the class representing trajectories willing to take detours to stick to signposted cycling routes accounting for 37% of all trajectories.

The article is structured as follows. In Section 2, we give an overview of related work. In Section 3, we introduce the terminology and basic concepts used in this article. In Section 4, we outline our approach for finding the optimization criterion corresponding to the minimal segmentation of the input path. The results of our experimental evaluation of the approach are presented in Section 5. We conclude the article in Section 6. The implementation of our approach is available as open source at https://gitlab.vgiscience.de/forsch/routing-preferences.

2 Related Work

Analyzing trajectories or movement data is a major research area in spatio-temporal data mining [30, 45]. Problems of interest include the construction of road network maps based on trajectories [1] and the augmentation of road network maps with additional attributes. For example, Yuan et al. [42] have enriched an existing road network representation of Beijing with information about travel times inferred from taxi trajectories. Although these approaches allow users to find the fastest routes in a road network, it does not reveal how users trade off multiple criteria when choosing a route.

Explicit vs implicit preferences In psychology, it is differentiated between two different kinds of preferences: explicit and implicit preferences [35,41]. Explicit preferences describe which preferences the user will state by themself when specifically asked for them. While the answer to this question gives valuable insight into the incentives and deterrents of the user, it is not clear whether the user knows their preference or whether the answer might be biased due to social pressure [23]. In contrast to that, implicit preferences correspond to subconscious behavior. As such, learning these preferences from questionnaires is not possible, instead, they can only be derived from user actions [17].

Trajectory segmentation In trajectory segmentation, a trajectory is split into sub-trajectories by time interval, shape, or semantic meaning [44]. Existing approaches work point-based, e.g. by identifying stay points [32] or by extracting interesting places [16], or segment-based by requiring certain properties to be similar on a segment. We will focus on the latter, which is closely related to our article. Buchin et al. [9] present a framework to segment a trajectory into a minimum number of segments such that each segment fulfills a spatio-temporal criterion. For *monotone* criteria (i.e. criteria that, if they hold for a certain segment, they also hold for every sub-segment), the authors present an efficient algorithm to compute the segmentation. Arnov et al. [4] show that this problem is, in general, NP-hard when considering non-monotone criteria. Nonetheless, they show that for some non-monotone criteria, polynomial-time algorithms exist. Alewijnse et al. [3] introduce a broader class of criteria, namely *stable* criteria, whose validity does not change often along the trajectory. For such criteria, they present an efficient segmentation algorithm. Trajectory segmentation is often combined with a classification of the segments. In a first step, Zhang et al. [43] segment trajectories based on changes in travel mode, and in a second step, they classify the segments according to the travel mode. Similarly, Alewijnse et al. [2] partition trajectories into classifiable sub-trajectories. For the classification, they restrict themselves to spatio-temporal information extracted from the trajectories, whereas we want to derive semantic information from an underlying graph. Depending on the application, this graph yields information about the road network or its surroundings.

Preference mining Much research has been done to predict user behavior by analyzing the historic trajectories of the users, e.g. [11]. These methods, however, strongly rely on frequently repeated routes, such as a user's daily trip to work. Hence, models that have been learned that way usually cannot be applied in areas where no trajectories have been recorded. Probably the first method that infers routing preferences from a sparse set of trajectories has been presented by Balteanu et al. [5]. Their algorithm can even be applied to single trajectories. However, their method has two drawbacks with respect to the practical relevance: Firstly, the algorithm yields reasonable results only if the trajectory is close to the *skyline*, i.e. the set of all Pareto-optimal paths, and thus cannot be applied to round trips. Secondly, the learned preference cannot be expressed as a single edge cost and, thus, it cannot be used to compute new routes with standard routing algorithms (e.g., Dijkstra's algorithm [13]). Since then, further methods that apply to single trajectories have been developed. Funke et al. [18] presented a method for inferring the routing preferences based on linear programming. Their algorithm requires, however, that a user's route is optimal for at least one linear combination of the edge costs. This is often not the case for longer routes. Therefore, several publications focus on combining trajectory segmentation and preference mining. Knapen et al. [29] present a method to segment a path into a minimal number of concatenated least-cost paths. While they do not learn routing preferences directly, they highlight how to use this information in route choice problems. More recently, Knapen et al. [28] extended their approach by identifying network vertices that are preferred as intermediate destinations. Our work is closely related to these authors' work, but instead of using geodesic shortest paths, we consider preference-weighted costs that are a linear combination of different basic costs. Similarly, Barth et al. [6] extended the approach of Funke et al. [18] by segmenting a path into as few as possible subpaths, such that for each subpath, a linear combination of the edge costs can be found, for which it is optimal. Though, as each subpath is explained with a different linear combination, it is not obvious how to calculate new routes from the results. In a later publication, Barth et al. [7] cluster trajectories by searching a minimum set of routing preferences such that each of the trajectories is optimal for at least one of them. To guarantee that a trajectory is optimal for any routing preference, the authors refer to trajectory segmentation approaches. We present such an approach in this article.

Bicycle routing In the field of routing preferences, previous studies examined the explicit preferences of cyclists. Sener et al. [39] performed a survey on preferred route attributes and found that cyclists prefer routes with no on-road parking, low traffic volume, continuous bicycle facilities, and lower roadway speed limits. In a further analysis, they evaluated additional detours a cyclist is willing to take to stick to these preferences. Emond et al. [15] and Majumdar et al. [33] identified perceived safety as a major incentive for route choice. Based on this research, Rosetti et al. [38] propose a choice model to aid cycling infrastructure planning. Similarly, Ehrgott et al. [14] present a route choice model based on a bicriterial routing model. They analyze the trade-off between travel time and suitability for cycling that cyclists are willing to make. Hardinghaus and Nieland [25] use a different approach for learning routing preferences. In their work, they analyzed the routing settings of users of a public bike routing engine. They conclude that the aspects of minimizing the trip's length, using minor roads, and having a high-quality surface heavily out-weight the importance of dedicated cycling infrastructure. The diversity of the route preferences among cyclists is highlighted by Hardinghaus and Weschke [26], who show that route preferences vary significantly between different demographic groups. According to the authors, especially for vulnerable road users, such as older people, dedicated bicycle infrastructure is important. The results of these studies allow for a pre-selection of routing criteria to be analyzed in the context of the implicit routing preferences of cyclists.

3 Preliminaries

In this section, we explain the graph-theoretical background of our approach and define the terminology used throughout the article. Let $G = (V, E)$ be a directed weighted graph with the vertex set V and the edge set E. G represents the road network, with edges being road links and vertices being road intersections or dead ends. Two non-negative edge costs $c_0(e)$ and $c_1(e)$ are assigned to each edge $e \in E$, representing the effort needed to traverse the edge. These costs model the routing criteria. For example, they quantify the distance along an edge, the time needed to traverse the edge or its total ascent. Our algorithms require $c_0(e)$ and $c_1(e)$ to be integer numbers. This is not a severe limitation, however, as we can choose a very fine discretization of continuous edge attributes.

We consider a bicriteria routing model that describes the trade-off between the two criteria c_0 and c_1. To that end, we combine these costs linearly using a *balance factor* $\alpha \in [0, 1]$. We obtain a *personalized* edge cost

$$c_\alpha(e) = (1 - \alpha) \cdot c_0(e) + \alpha \cdot c_1(e). \tag{1}$$

The factor α expresses the preference towards the two initial edge costs: a value close to zero implies a preference towards c_1 while a value close to one implies a preference towards c_0. Given the balance factor of a user, an optimal route can be found by applying a shortest-path algorithm on G using their personalized edge cost.

A *trajectory* is a sequence of location measurements reflecting the movement of a person or an object. To combine the information of the road network and a trajectory, the trajectory is matched to the road network graph G (for reference, see [10,27]). The matched trajectory is a sequence $P = (e_1, \dots, e_n)$ of n not necessarily distinct edges in G, which we refer to as *path*. Equation 1 generalizes to the personalized cost of a path P by

$$\begin{aligned} c_\alpha(P) &= \sum_{e \in P} c_\alpha(e) \\ &= (1 - \alpha) \cdot \sum_{e \in P} c_0(e) + \alpha \cdot \sum_{e \in P} c_1(e) \\ &= (1 - \alpha) \cdot c_0(P) + \alpha \cdot c_1(P) \\ &= c_0(P) + \alpha \cdot \big(c_1(P) - c_0(P)\big). \end{aligned} \tag{2}$$

We say that P is *α-optimal* for a given value α if there is no other path Q that starts and ends at the same vertices as P and whose personalized cost $c_\alpha(Q)$ is less than $c_\alpha(P)$. Following from Equation 2, for a path P, the personalized cost c_α can be treated as a function over the balance factor α, resulting in a line with vertical intercept $c_0(P)$ and slope $c_1(P) - c_0(P)$. Consequently, we can consider the set of all s-t paths, that is, all paths sharing the same terminal vertices s and t, as a family of lines, see Figure 2. In the following, we refer to a family of s-t-lines as *line arrangement*. The paths that are α-optimal for some $\alpha \in [0, 1]$ form the *lower envelope* $\mathcal{E}\colon [0, 1] \to \mathbb{R}_{\geq 0}$ of the corresponding line arrangement, see the orange highlighting in Figure 2b. A path (and thus a line) on the lower envelope for a given α can be found by performing a query for a path from s to t in G that is optimal with respect to c_α.

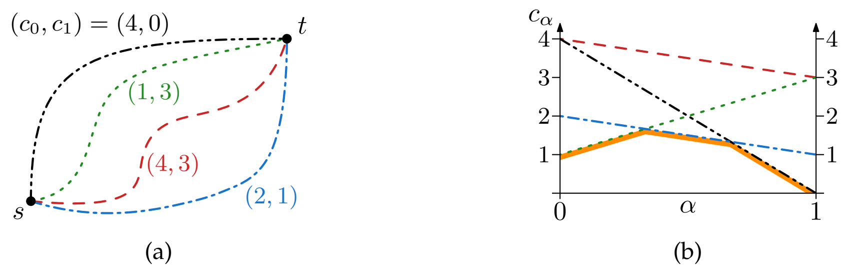


Figure 2: (a) A set of s-t paths with costs (c_0, c_1) and (b) the corresponding line arrangement; colors are chosen accordingly. In (b), the lower envelope $\mathcal{E}$ is highlighted in orange.

The section of a line on the lower envelope forms the *optimality range* of the corresponding path. For this α-interval the path is α-optimal. As a line is linear and defined over the whole interval $[0, 1]$, every line has at most one section on the lower envelope. Thus, the optimality range is either empty, a single value, or an interval. In Figure 2, the optimality range of the red path is empty, and the optimality ranges of the green, blue, and black paths are $[0, 1/3]$, $[1/3, 2/3]$, and $[2/3, 1]$, respectively.

4 Methodology

In this section, we present our approach for inferring routing preferences from sparse sets of trajectories. The approach consists of two major components. At first, we present an algorithm that, given a path, returns the set of balance factors for which this path is optimal (Section 4.1). We then discuss how this algorithm can be used to calculate the balance factor for which the path segmentation into optimal subpaths is minimal (Section 4.2). Unlike in the work of Barth et al. [6], we require this balance factor for each subpath to be the same. This improves the approach's utility by providing an explicit balance factor for future routing.

4.1 Calculating Optimality Ranges for Paths

Given an s-t path P, we want to identify the α-interval for which P is optimal regarding the routing model introduced in Section 3. We call this problem OPTIMALITYRANGE. More formally, we solve the following problem:

Problem 1 (OPTIMALITYRANGE)**.** Given a graph G with edge cost functions c_0 and c_1, and a path P, find $\mathcal{I}_{\text{opt}} = \{\alpha \in [0, 1] \mid P \text{ is } \alpha\text{-optimal}\}$.

A known algorithm for finding elements on the lower envelope Funke and Storandt [20] have solved the problem of deciding whether a path P is part of the lower envelope. Their algorithm, called WITNESSSEARCH, searches for a single α value for which P is optimal. If such a value can be found, P is part of the lower envelope; if no value can be found, P is not part of the lower envelope. We use WITNESSSEARCH as the starting point for our algorithm. Due to the algorithm's importance for this chapter, we elaborate on their findings in more detail in the following. The algorithm of Funke and Storandt works iteratively. In every step, an interval $\mathcal{I}^*$ is considered that is guaranteed to contain the optimality range $\mathcal{I}_{\text{opt}}$ completely. From step to step, this interval $\mathcal{I}^*$ is reduced until a point α^* within the optimality range is found, or its size falls below the minimum size of an optimality range, which implies that the optimality range is empty. If the optimality range is not empty, WITNESSSEARCH returns a point α^* within the optimality range and the interval $\mathcal{I}^*$ as considered in the final iteration of the algorithm. In the other case, the path is not optimal for any α. Our algorithm for finding the full optimality range thus can be omitted in this case. The running time of WITNESSSEARCH is in $\mathcal{O}(\text{SPQ} \cdot \log(Mn))$, where SPQ is the running time of a shortest-path query in G, n is the number of vertices in G, and M is the maximum edge cost among all the edge costs c_0 and c_1.

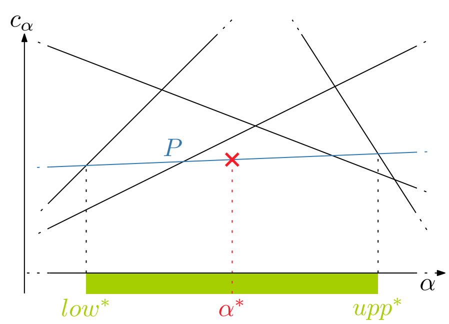


Figure 3: The output of WITNESSSEARCH that we use as input for our algorithm: a balance factor α^* for which path P (blue) is α-optimal and the interval $\mathcal{I} = [low^*, upp^*]$ (green) that is guaranteed to completely contain the optimality range $\mathcal{I}_{\text{opt}}$ of P. In specific, low^* and upp^* are values where the line for P crosses another line of the line arrangement and $\alpha^* = \frac{low^*+upp^*}{2}$.

An extension yielding the optimality range So far, only one element α^* within the optimality range $\mathcal{I}_{\text{opt}} = [\alpha', \alpha'']$ of P is known. We continue by searching for the boundaries of $\mathcal{I}_{\text{opt}}$. To that end, we use the results of WITNESSSEARCH that are displayed in Figure 3. In particular, these are a balance factor α^* for which P is optimal with respect to c_{α^*} and the interval $\mathcal{I}^* = [low^*, upp^*]$ that is guaranteed to contain the optimality range $\mathcal{I}_{\text{opt}}$ of P completely. More specific, low^* and upp^* are values where the line for P crosses another line of the line arrangement and $\alpha^* = \frac{low^*+upp^*}{2}$.

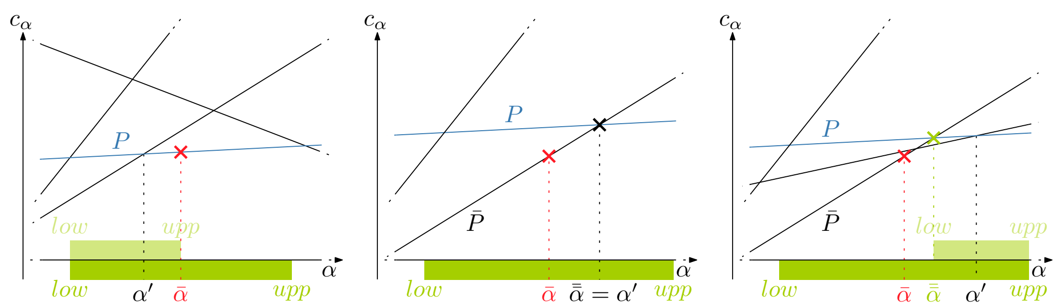
We know that $\alpha', \alpha'' \in \mathcal{I}^* = [low^*, upp^*]$. In particular, $\alpha' \in [low^*, \alpha^*]$ and $\alpha'' \in [\alpha^*, upp^*]$ hold. In the following, we describe how to determine α' with a binary search that uses the structure of the line arrangement. The search for α'' is organized symmetrically. This procedure is summarized in Algorithm 1 and visualized in Figure 4. In every iteration, a search interval $\mathcal{I} = [low, upp]$ is given and the following invariants hold:

(1) the lower bound α' is contained in $\mathcal{I}$,

(2) at value low, the line for P crosses another line of the line arrangement,

(3) P is optimal with respect to c_{upp}.

The interval $\mathcal{I}_1 = [low^*, \alpha^*]$ as given by WITNESSSEARCH fulfills these properties and is used as input for the first iteration. We consider the central value $\bar{\alpha} = \frac{low+upp}{2}$ of $\mathcal{I}$. In each iteration, one of the following cases can occur; see Figure 4. Figure 4a displays Case 1. In this case, P is optimal with respect to $c_{\bar{\alpha}}$, and the next iteration continues with a search in $[low, \bar{\alpha}]$. Otherwise, in Case 2, there exists another path $\bar{P}$ that is optimal with respect to $c_{\bar{\alpha}}$. This path $\bar{P}$ has the same cost as P for some value $\bar{\bar{\alpha}}$ and dominates P in $[0, \bar{\bar{\alpha}}]$. Case 2 has two sub-cases depending on the α-optimality of P with respect to the crossing point $\bar{\bar{\alpha}}$. Figure 4b shows Case 2a, where P is an optimal path with respect to $c_{\bar{\bar{\alpha}}}$ and thus the lower bound of the optimality range is found, and the algorithm terminates. Otherwise, in Case 2b highlighted in Figure 4c, the search continues in $[\bar{\bar{\alpha}}, upp]$. In all cases, the invariants hold, and either the solution is found or the search interval $\mathcal{I}$ is reduced by at least half.



(a) Case 1 (b) Case 2a (c) Case 2b

Figure 4: The search for the lower bound of the optimality range. The search interval is marked green (its current state is darker than the next). (a) If P is optimal with respect to $c_{\bar{\alpha}}$, the search for the lower bound continues in $[low, \bar{\alpha}]$. (b) If P is optimal with respect to $c_{\bar{\bar{\alpha}}}$, the lower bound of the optimality range is found, and the algorithm terminates. (c) At $\bar{\alpha}$, the path P is dominated by $\bar{P}$. The search continues in $[\bar{\bar{\alpha}}, upp]$ where $\bar{\bar{\alpha}}$ marks the intersection of P and $\bar{P}$.

The lower bound, α', is found at the latest when the size of the search interval $\mathcal{I}$ falls below the minimum size of the optimality range. The search for α'' is done symmetrically. Hence, in the worst case, the search interval needs to be reduced from $[0, 1]$ to below the minimal size twice; once for the lower bound α', once for the upper bound α''. Consequently, the optimality range is found in $\mathcal{O}(\mathrm{SPQ} \cdot \log(Mn))$ time. Thus, the computational complexity of the presented algorithm to solve OPTIMALITYRANGE is lower by a factor of $\mathcal{O}(log(Mn))$ compared to the version presented in [37].

Algorithm 1: FindOptimalRangeLowerBound

Input: path P, α^*, low^* as given by WITNESSSEARCH [20]
Output: lower bound α'

$upp \leftarrow \alpha^*$;
while `true` **do**
  $\bar{\alpha} \leftarrow (low + upp)/2$;
  $\bar{P} \leftarrow \mathrm{argmin}_{P'}\, c_{\bar{\alpha}}(P')$; `// optimal path wrt.` $c_{\bar{\alpha}}$
  if $c_{\bar{\alpha}}(P) = c_{\bar{\alpha}}(\bar{P})$ **then** `// Case 1:` P `is optimal wrt.` $c_{\bar{\alpha}}$ `(Figure 4a)`
    $upp \leftarrow \bar{\alpha}$;
  else `// Case 2:` P `is not optimal wrt.` $c_{\bar{\alpha}}$
    $\bar{\bar{\alpha}} \leftarrow \dfrac{c_0(\bar{P}) - c_0(P)}{c_1(P) - c_0(P) - c_1(\bar{P}) + c_0(\bar{P})}$; `// crossing of` $\bar{P}$ `and` P
    $\bar{\bar{P}} \leftarrow \mathrm{argmin}_{P'}\, c_{\bar{\bar{\alpha}}}(P')$; `// optimal path wrt.` $c_{\bar{\bar{\alpha}}}$
    if $c_{\bar{\bar{\alpha}}}(P) = c_{\bar{\bar{\alpha}}}(\bar{\bar{P}})$ **then** `// Case 2a:` P `is optimal wrt.` $c_{\bar{\bar{\alpha}}}$ `(Figure 4b)`
      return $\bar{\bar{\alpha}}$;
    else `// Case 2b (Figure 4c)`
      $low \leftarrow \bar{\bar{\alpha}}$;

4.2 Calculating minimal milestone segmentations

Given a path P and a balance factor α a *milestone segmentation* is a decomposition of P into the minimum number of subpaths $\{P_1, \ldots, P_h\}$, such that every subpath P_i with $i \in \{1, \ldots, h\}$ is α-optimal. We call the split vertices between the subpaths *milestones*. To infer the routing behavior of the user, we search for the balance factor $\alpha \in [0, 1]$ whose milestone segmentation has the minimum number of subpaths over all milestone segmentations of P. The problem we solve is the following:

Problem 2 (MILESTONESEGMENTATION)**.** Given a graph G with edge cost functions c_0 and c_1, and a path P, find $\alpha \in [0, 1]$ and a milestone segmentation of P with respect to α that is as small as possible. That is, there is no $\tilde{\alpha}$ that yields a milestone segmentation of P with a smaller number of subpaths.

To solve MILESTONESEGMENTATION, we compute a set that contains a milestone segmentation of P for every α, then we select the optimal milestone segmentation. Enumerating the set of milestone segmentations is done by adopting a concept known as *start-stop-matrix* [3,4]. A start-stop matrix is a matrix $\mathcal{M}$, where entry $\mathcal{M}[i, j]$ is a Boolean value indicating whether the subpath starting at the i-th vertex of the path and ending at the j-th vertex of the path fulfills the segmentation criterion. In our case, α-optimality is the segmentation criterion. In MILESTONESEGMENTATION, we do not solve the segmentation for a single α, but we search for the α value minimizing the number of subpaths over all $\alpha \in [0, 1]$. Therefore, instead of storing Boolean values in the start-stop-matrix, we store the optimality interval for the corresponding subpath. This allows us to use the same matrix for all α values. Figure 5 shows the (Boolean) start-stop matrix for a single α, retrieved by applying the expression $\alpha \in \mathcal{M}[i, j]$ to all elements of the matrix. For a path P consisting of k vertices, we consider a $(k \times k)$-matrix $\mathcal{M}$ of sub-intervals of $[0, 1]$. The entry $\mathcal{M}[i, j]$

in row i and column j corresponds to the optimality range of the subpath of P starting at its i-th vertex and ending at its j-th vertex. Since we are interested only in subpaths with the same direction as P, we focus on the upper triangle matrix with $i \leq j$. Solving MILESTONESEGMENTATION with a start-stop-matrix splits the problem into two sub-problems: filling the matrix and solving for the optimal segmentation.

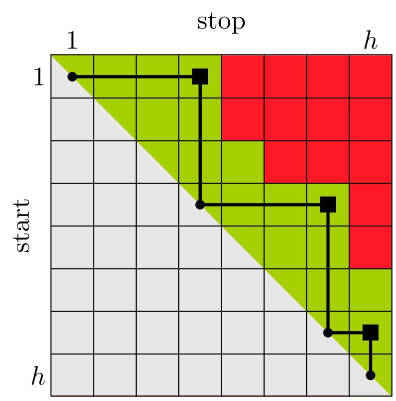


Figure 5: Depiction of a start-stop matrix of a path consisting of $h = 8$ vertices for a fixed $\alpha \in [0, 1]$. Green elements indicate that $\alpha \notin \mathcal{M}[i, j]$ whereas red elements indicate the opposite. The black line represents a minimum path segmentation into three subpaths (v_1, v_4, v_7, v_8) resulting from a forward search.

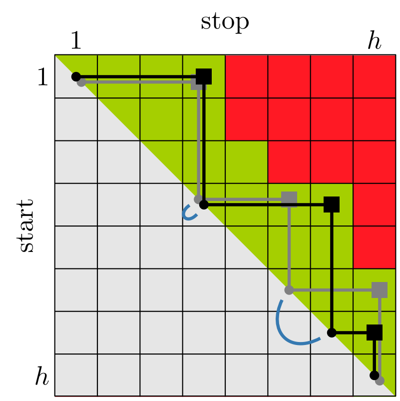


Figure 6: Extension to Figure 5. In addition to the segmentation of the forward search (black), the segmentation when searching backward (gray) is displayed. The blue lines indicate the milestone suites. Note, that the first milestone suite degenerated into a single vertex.

Computing the start-stop-matrix Due to the optimal substructure of shortest paths [12], for $i < k < j$, both $\mathcal{M}[i, j] \subseteq \mathcal{M}[i, k]$ and $\mathcal{M}[i, j] \subseteq \mathcal{M}[k, j]$ hold. This results in the structure visible in Figure 5, where no red cell is below or left of a green cell. Therefore, when computing the start-stop matrix, we start at the lower left corner and fill each row starting from its main diagonal. That way, we can limit the search space for $\mathcal{M}[i, j]$ to the intersection of the already computed values of $\mathcal{M}[i+1, j]$ and $\mathcal{M}[i, j-1]$. This is especially useful if one of these intervals is empty, meaning the current interval is also empty, and computation in this row can be stopped.

Solving for the optimal segmentation As a consequence of the substructure optimality explained above, it is easy to find a solution to the segmentation problem for a fixed α, once the start-stop matrix is set up. According to Buchin et al. [9], for example, an exact solution to this problem can be found with a greedy approach in $\mathcal{O}(h)$ time; see Figure 5. Knapen et al. [29] have shown that applying the greedy approach twice, once starting from the beginning vertex and once from the end vertex of P, results in pairs of milestones that form the terminal vertices of a *milestone suite*. Inside a milestone suite, the milestone can be chosen arbitrarily without violating the segmentation into optimal subpaths. Note, however, that choosing a milestone in a specific milestone suite might fix the milestone location in other milestone suites. The size of a milestone suite is the distance along the path between its terminal vertices. In Figure 6, the forward and backward searches are indicated by a black and gray line, respectively, resulting in one milestone suite of size zero and one milestone suite of size one (blue). In case multiple segmentations have the same number of subpaths, we use their summed milestone suite sizes as a tiebreaker. To have the highest adaptability of the learned balance factor, we consider larger milestone suite sizes are better than smaller ones.

Since we consider a finite set of intervals, we know that if a minimal solution exists for an $\alpha \in [0,1]$, it also exists for one of the values bounding the intervals in $\mathcal{M}$. Consequently, we can discretize our search space without loss of exactness, and each of the $\mathcal{O}(h^2)$ optimality ranges yields at most two candidates for the solution. For each candidate, a minimum segmentation is computed in $\mathcal{O}(h)$ time. Thus, we end up with a total running time of $\mathcal{O}(h^2 \cdot (h + \mathrm{SPQ}\log(Mn))$ where n denotes the number of vertices in the graph and h denotes the number of vertices in the considered path. Thus, the algorithm is efficient and yields an exact solution to MILESTONESEGMENTATION. The solution consists of the intervals of the balance factor producing the best-personalized costs with respect to the input criteria.

Guaranteeing the existence of a milestone segmentation Finally, we note that for a general graph $G = (V, E)$ and a path P in G, there may be no milestone segmentation. Consider, for example, the case in Figure 7a, in which P consists of a single edge $e = \{s, t\}$ whose costs $c_0(e)$ and $c_1(e)$ are larger than the costs $c_0(Q)$ and $c_1(Q)$, respectively. Since P is dominated by Q for any $\alpha \in [0,1]$ and since we cannot subdivide P into shorter paths in G, the path P does not admit a partition into α-optimal subpaths.

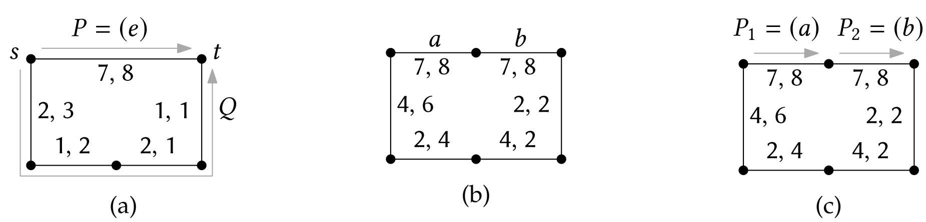


Figure 7: (a) A graph with a path $P = (e)$ that does not admit a milestone segmentation and (b) the result of our method that ensures the feasibility of the problem. Comma-separated numbers represent c_0 and c_1. (c) Milestone decomposition after applying our method. Path P is split into two subpaths, with the newly introduced vertex being the milestone.

With a simple pre-processing step, however, we can ensure that a solution exists. For this, we introduce a new vertex w for each edge $\{u, v\} \in P$ and replace that edge with two edges $a = \{u, w\}$ and $b = \{w, v\}$, both in G and P. We define $c_0(a) = c_0(b) = c_0(e)$ as well as $c_1(a) = c_1(b) = c_1(e)$ and double the costs of each unchanged edge in G; see Figure 7b. Thereby, we ensure that the costs c_0 and c_1 are still integer. The costs of paths in the original graph are exactly doubled. Thus, if they were α-optimal in the original graph, they remain α-optimal after the modification. After the pre-processing step, however, every edge $\{u, v\}$ of P is α-optimal for any $\alpha \in [0,1]$. Therefore, a milestone segmentation exists, and an optimal milestone segmentation of P contains at most $|P|$ paths—or $2|P|$ paths if we consider the size $|P|$ of P in the original graph. Note that P in its entirety is still not α-optimal for any $\alpha \in [0,1]$. We deliberately do not want to change the optimality of paths in the original graph. Instead, P is now guaranteed to have a milestone segmentation using the newly introduced vertices as milestones; see Figure 7c. The asymptotic running time of our algorithms is not affected by our method to guarantee the existence of a milestone segmentation.

5 Experimental Study

In this section, we present the experimental evaluation of our algorithm. Our experiments are set up for an examination of the extent to which cyclists stick to official bicycle routes. For this, we define the following edge costs:

$$\begin{aligned} c_0(e) &= \begin{cases} 0, & \text{if } e \text{ is part of an official bicycle route} \\ d(e), & \text{else.} \end{cases} \\ c_1(e) &= \begin{cases} d(e), & \text{if } e \text{ is part of an official bicycle route} \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{3}$$

Hence, in our cost function, we use edge costs c_0 and c_1 that, apart from an edge's geodesic length d, depend only on whether the edge in question is part of an official bicycle route. Since c_0 and c_1 are discrete cost functions, the costs of each edge are rounded to the accuracy of one decimeter. With respect to the α-cost $c_\alpha = (1-\alpha)\cdot c_0 + \alpha \cdot c_1$, the possible outcome can be interpreted as follows. Suppose there is a path P_1 entirely using official bicycle routes. Its cost is: $c_\alpha(P_1) = \alpha \cdot c_1(P_1) = \alpha \cdot d(P_1)$. Further, let P_2 be a path entirely using routes *not* part of official bicycle routes. Its cost is $c_\alpha(P_2) = (1-\alpha)\cdot c_0(P_2) = (1-\alpha)\cdot d(P_2)$. If both paths have the same cost given a certain value for α, the following equation holds:

$$\alpha \cdot d(P_1) = (1-\alpha)\cdot d(P_2) \tag{4}$$

A minimal milestone segmentation for $\alpha = 0.5$ suggests that minimizing geodesic length outweighs the interest in official cycle paths. A value of $\alpha < 0.5$ indicates avoidance of official cycle paths whereas $\alpha > 0.5$ indicates the opposite, i.e. preference. In specific, if we are interested in the relative detour δ_{PRO} a cyclist is willing to take in order to stick to official bicycle routes, we can transform Equation 4:

$$\delta_{\text{PRO}} = \frac{d(P_2)}{d(P_1)} = \frac{\alpha}{(1-\alpha)} \tag{5}$$

In contrast, the detour δ_{CON} a cyclist is willing to take in order to avoid official bicycle routes can be computed using the inverse of Equation 5. For our analysis we will use the following combined detour function:

$$\delta(\alpha) = \begin{cases} \delta_{\text{CON}} - 1, & \text{if } \alpha < 0.5 \\ 0, & \text{if } \alpha = 0.5 \\ \delta_{\text{PRO}} - 1, & \text{if } \alpha > 0.5 \end{cases} \tag{6}$$

Table 1 provides the values of this function for some exemplary inputs.

In our experiments, we limit ourselves to $\alpha \in [0.05, 0.95]$ in order to avoid $\alpha \in \{0, 1\}$. In these two cases, for every road segment e either $c_0(e) = 0$ or $c_1(e) = 0$ holds. In such a graph, for any s-t path, there are numerous alternatives of the same cost which only differ in additional segments of cost zero. Computing shortest paths is time-consuming in this graph since lots of vertices need to be explored. Besides, in our experiments, considering a detour of up to 1800% seems sufficient.

Table 1: A selection of α values and their meaning. The column $\alpha = 0.5 \pm \mathbf{0.1}$ indicates that with $\alpha = 0.4$ (or 0.6, respectively) the route planners are willing to cover 50% distance extra to use preferred road segments.

$\alpha = 0.5 \pm x$	0	0.0025	0.005	0.01	0.1	$1/6$	0.25	$1/3$	0.4	0.45
detour δ	0	1%	2%	4%	50%	100%	200%	400%	800%	1800%

5.1 Data and Experimental Setup

We now describe the setup in more detail. The road network including the information about bicycle routes is obtained from penStreetMap[1] and comprises the area $5.99° - 7.67°\ E$ and $50.47° - 51.47°\ N$. It contains only roads that are, according to OpenStreetMap tags, open to cyclists.

Figure 8a shows an extract of this road network data. We transformed the road network into a directed graph $G = (V, E)$, such that for each edge $(u, v) \in E$ also the reverse edge (v, u) is contained in E. This means that we do not consider one-way restrictions in our experiments, though generally, our algorithm can handle them. The two cost functions $c_0, c_1 : E \to \mathbb{N}$ are defined as in Equation 3.

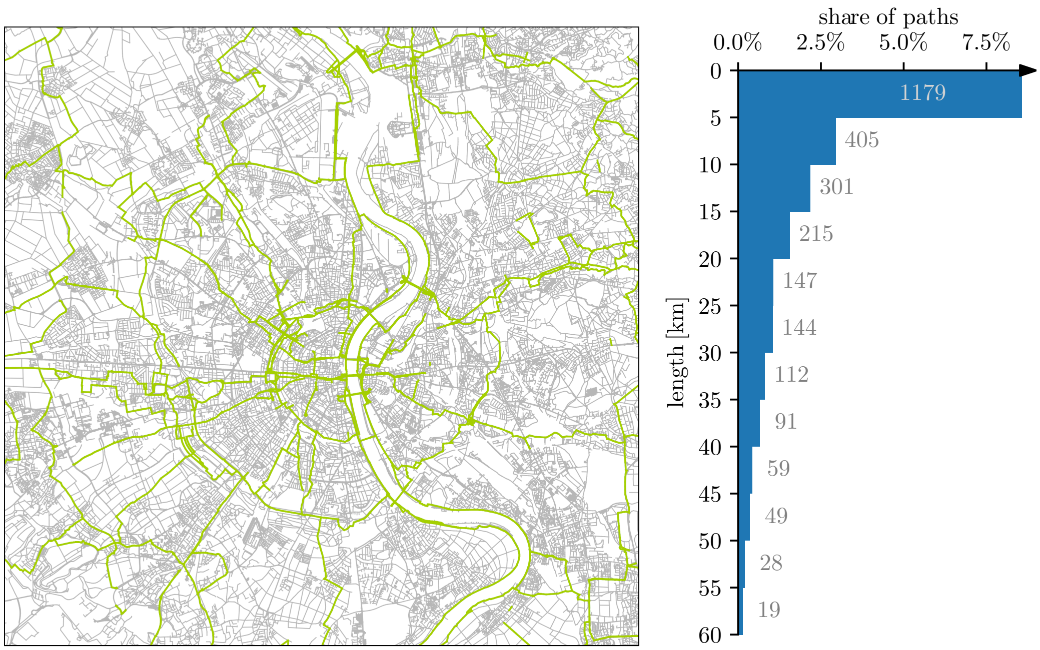
In order to reduce the complexity of the graph, we removed vertices with degree two from G. More precisely, when removing a vertex u with two outgoing edges (u, v) and (u, w) as well as the corresponding incoming edges (v, u) and (w, u), we replace the four edges with (v, w) and (w, v). The costs of (v, w) are the summed costs of (v, u) and (u, w); the costs of (w, v) are the summed costs of (w, u) and (u, v). We denote the resulting graph by H. That way, the graph could be reduced from containing 3.5 million vertices and 7.8 million edges to containing 0.9 million vertices and 2.5 million edges only.

The trajectory data used in our experiments stems from the user-driven platform GPSies[2]. This platform is used by different groups of cyclists and features a wide range of cycling activities, starting from scenic bike touring to mountain biking to road biking. Thus, we expect to find different routing preferences in our analysis. We obtained 1016 trajectories by querying trajectories recorded by cyclists around the German city of Cologne in a search radius of $25\,\mathrm{km}$. The trajectories have a length up to $60\,\mathrm{km}$.

To analyze the trajectories with our algorithm, they need to be converted to paths in G. We are working with data from VGI sources; therefore, we need a map matching that is robust to possible data inconsistencies. To that end, we use the map-matching algorithm Baseline+ [8]. This algorithm is specifically designed for incomplete road networks or trajectories leaving the road network. If such an inconsistency is detected, the corresponding

[1]www.openstreetmap.org

[2]www.gpsies.com, [no longer available], downloaded: May 2018



(a) road network (b) path lengths

Figure 8: (a) The road network of the city of Cologne, the center of the area containing the examined trajectories. Only roads and paths that are accessible for cyclists are on display. Roads and paths that are part of an official signposted cycle route are drawn as green lines. (b) Geometric length of the matched paths.

trajectory is split into multiple parts, and each part is matched to an individual path. The map-matching algorithm has two parameters: the candidate cost and the unmatched cost. We set the parameter candidate cost to 0.02 and the parameter unmatched cost to 3.0. This parameter choice is recommended by the authors and performed well on our input data, as validated by a visual examination of the results. We filtered the resulting paths to have a minimum length of 100 meters. We end up with a total number of 2750 paths with a mean distance of 12.3 kilometers (see Figure 8b).

In case a start or end point of a path P was mapped to a point p in the interior of an edge e of G, we temporarily injected p as a new vertex of G by splitting e into two edges. Thereby, we obtain P as a sequence of edges of G. We linearly distributed the costs of the edge e over the two new edges. In the reduced graph H, we temporarily insert the sequence of degree-two vertices of P containing additional vertices (like p) as well as the connecting edges, which allows us to consider P as a sequence of edges in H. We further apply the pre-processing step described in Section 4.2 to ensure that the path has a milestone segmentation. All changes on H are undone before the next path is considered.

The experiments were performed on an AMD Ryzen Threadripper 3970X 32-Core processor. The machine is clocked at 3.7 GHz and provided our program with 13 GB RAM. Our implementation[3] is written as a single-threaded Java application, and we executed the

[3]https://gitlab.vgiscience.de/forsch/routing-preferences

experiments in a Java 11 Runtime Environment. We use two open-source libraries, namely GeoTools [40] for handling geographic data and JGraphT [34] for the graph structures and algorithms.

5.2 Case study on a single trajectory

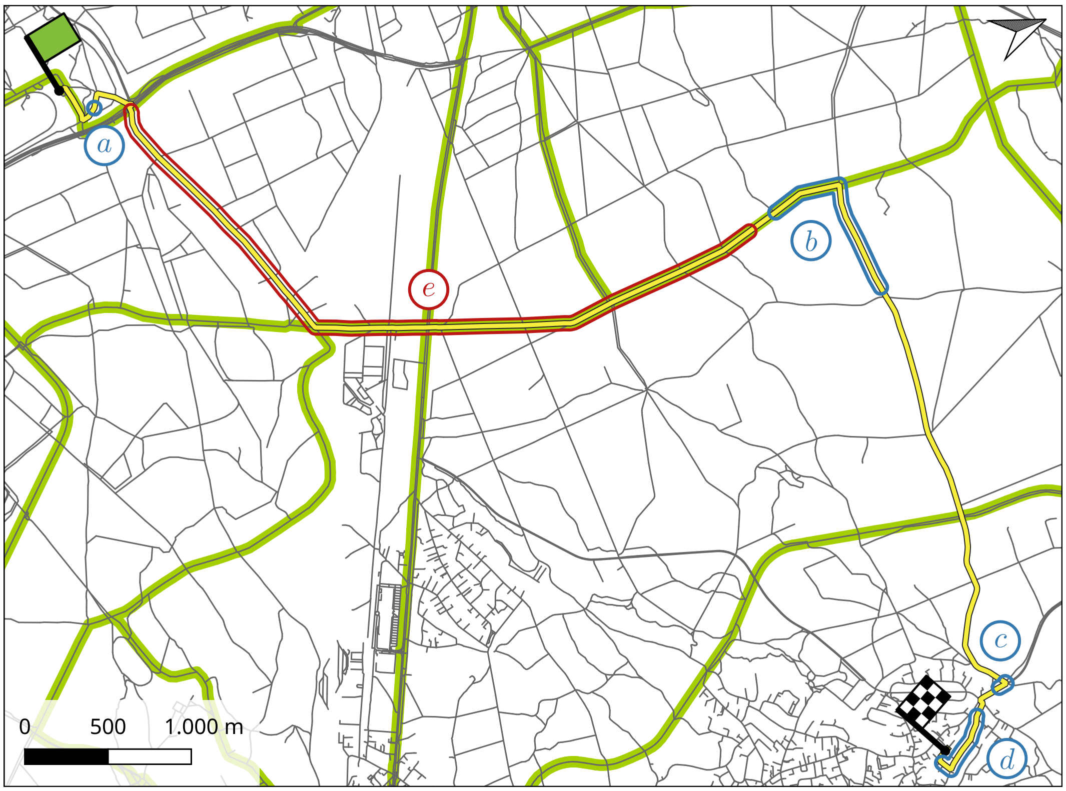


Figure 9: The user's path (yellow) has an optimal milestone segmentation with four milestones a, b, c, and d (blue). Disregarding minor shiftings of a and/or b, this is a valid optimal segmentation for each $\alpha \in [0.511, 0.607]$ (compare with Figure 10). The subpath (a, b) is α-optimal for $0.511 \leq \alpha \leq 0.607$ only. Therefore, for $\alpha = 0.5$, i.e. the geodesic shortest path, an additional milestone e (red) is needed.

Before we present an analysis of the examined 2750 paths, we scrutinize the results for a single path of $10.0\,\mathrm{km}$ length. Figure 9 shows the trajectory as well as an optimal solution of MILESTONESEGMENTATION. In this example, we found an optimal segmentation by assuming that cyclists are willing to take a detour between $\delta = 4\%$ and $\delta = 54\%$ over the length of the shortest path, corresponding to $\alpha \in [0.511, 0.607]$. The milestone suites for this interval are shown in blue. When assuming that cyclists are sticking to the geodesic shortest path, an additional milestone must be placed inside milestone suite e depicted in red. Figure 10 (right) depicts every segmentation our algorithm finds for this trajectory.

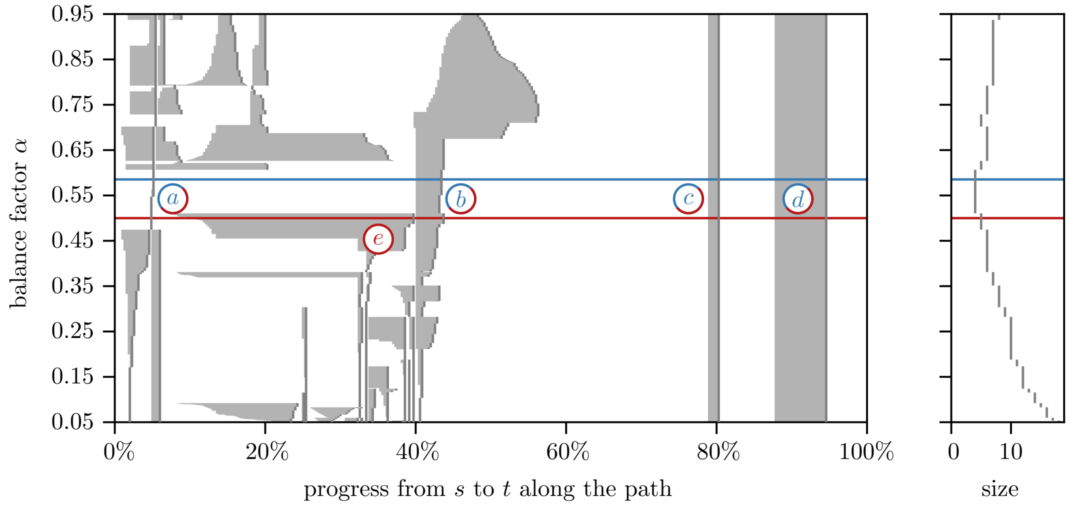


Figure 10: (left) Milestone plot for the path presented in Figure 9. The milestone segmentations over all α values are shown. The gray marks indicate the positions of the milestone as found by a forward search. The light gray areas denote the milestone suites for the corresponding milestones. The blue line represents an optimal milestone segmentation at $\alpha = 0.58$ with milestones in a, b, c, and d. For $\alpha = 0.5$ (red), an additional milestone in e is needed. (right) The number of milestones needed per α.

The two segmentations shown in Figure 9 are highlighted by a blue and red line, respectively. In dark gray, the milestones as found by the forward search are shown with the corresponding milestone suite depicted in light gray. We observe that the sizes of the individual milestone suites grow towards the optimal interval $\alpha \in [0.511, 0.607]$. In specific, take a look at milestone suite e: where it first appears at $\alpha = 0.511$ it spans a large portion of the path, while it quickly shrinks until it is replaced by two separate milestones at $\alpha \approx 0.36$. This observation supports our initial assumption that larger milestone suites are more adaptive and, thus, less prone to overfitting. The milestone plot shown in Figure 10 is suitable for further analysis of the trajectory and its surrounding road network as it shows for every α value where milestones are necessary. Since our focus is set on the analysis of user preferences, we postpone research on this matter. Figure 10 (left) summarizes the milestone plot by showing how the size of the shortest milestone segmentation depends on α.

5.3 Analyzing all trajectories

First, we use our algorithm to divide the set of trajectories into three groups PRO, CON, and INDIFF. The group PRO comprises trajectories for which a milestone segmentation of minimal size can only be found for $\alpha > \frac{1}{2}$. We interpret such a result that way that the trajectory was planned in favor of official cycle routes. Conversely, we consider it as avoidance of official cycle routes if milestone segmentations of minimal size exist only for $\alpha < \frac{1}{2}$. The group CON represents these trajectories. Considering trajectories lacking this clarity, we refrain from a strict categorization into one of the two groups above. These trajectories form a third group, INDIFF. The results of this classification are displayed in Figure 11. The group PRO being over four times larger than the group CON is a first indicator that cyclists prefer official cycle routes over other roads and paths. The group CON is the smallest group with a share of 8% of all paths. We assume this group mainly consists of road cyclists who prefer using roads over cycleways. In future research, this could be verified by analyzing a data set of annotated trajectories.

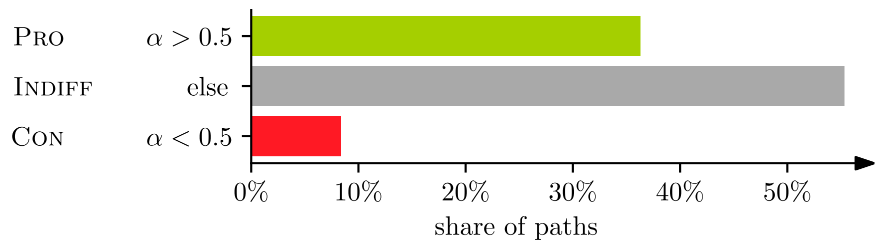


Figure 11: Size of the categories PRO (green, 998 paths), CON (red, 230 paths), and INDIFF (gray, 1522 paths)

For a more detailed evaluation of the results, we introduce the *deficiency* Δ of a segmentation $\mathcal{S}$ to be able to compare segmentations of different paths to each other. For a path P, the deficiency $\Delta(\alpha)$ that compares the number of subpaths in $\mathcal{S}(\alpha)$ to the number of subpaths in the optimal segmentation of P, $|\mathcal{S}_{\text{opt}}|$:

$$\Delta(\alpha) = \frac{|\mathcal{S}(\alpha)|}{|\mathcal{S}_{\text{opt}}|} - 1. \tag{7}$$

Put in other words, the deficiency describes the share of additional subpaths needed for a given α value. Thus, it can be considered a percentage. Following this definition, the deficiency of the optimal segmentation of P is 0% and the worse an α value fits the user's preference, the more the deficiency increases.

Figure 12 displays for every α the share of the evaluated paths that are optimal with respect to the corresponding combined cost function c_α. Phrased differently, the figure shows the share of paths that have a deficiency of zero for the given α. Considering the group PRO (green), the maximum share is reached with $\alpha \approx 0.5721$ as balance factor. This corresponds to detours of more than 30% and emphasizes the preference for official cycle routes. The second group, INDIFF (gray), reaches its optimum for $\alpha = 0.5$, i.e. for minimizing route length. Indeed, more than 94% of this group's paths have an optimum milestone segmentation for this α value. This causes a significant shift in the overall results (black). The balance factor that has, over all groups, the highest share of optimal paths is $\alpha \approx 0.5004$, corresponding to detours of about 0.2%. Although this value is relatively small, the relevance of official cycle paths is undeniable. Figure 13 supports this claim. It displays for every α the average deficiency of the evaluated paths. A segmentation for $\alpha \approx 0.5004$ takes, on average, only about 14% more milestones than the respective minimum segmentation. Also, on average less than 50% milestones extra compared to the minimum segmentation is achieved for $\alpha \in [0.4824, 0.5980]$ corresponding to detours of less than 8% (avoiding cycle routes) and 50% (favoring cycle routes), respectively. This shows how much better the trajectories can be described under the assumption that official cycle routes are preferred. Considering an accepted detour of 50% likewise to avoid official cycle routes (i.e., $\alpha = 0.4$),

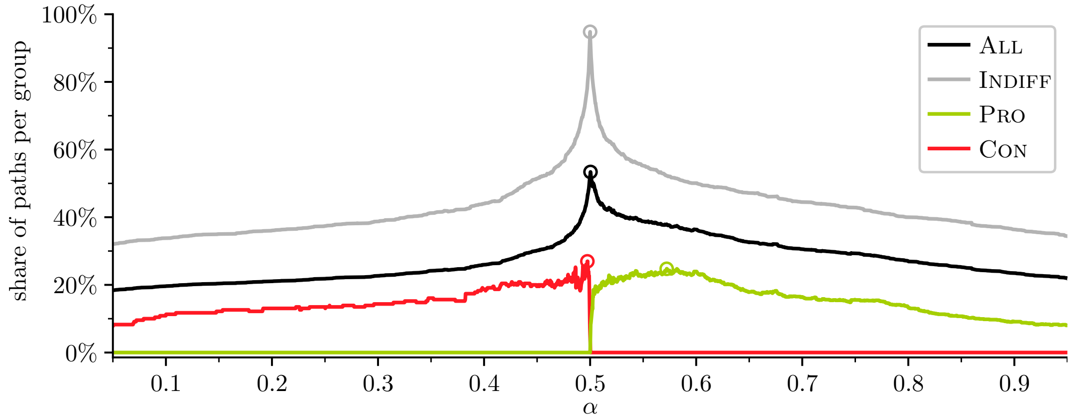


Figure 12: Share of paths per group that are optimal for the given α. The maxima are marked with a circle and are achieved for $\alpha \approx 0.4974$ (CON, red), $\alpha \approx 0.5721$ (PRO, green), $\alpha = \frac{1}{2}$ (INDIFF, gray), and $\alpha = 0.5004$ (all trajectories, black).

one needs, on average, more than 2.5 times the number of milestones compared to the minimum segmentation, see the red line in Figure 13.

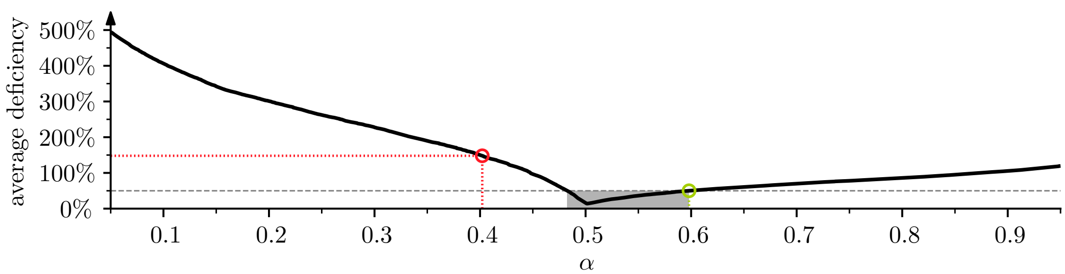
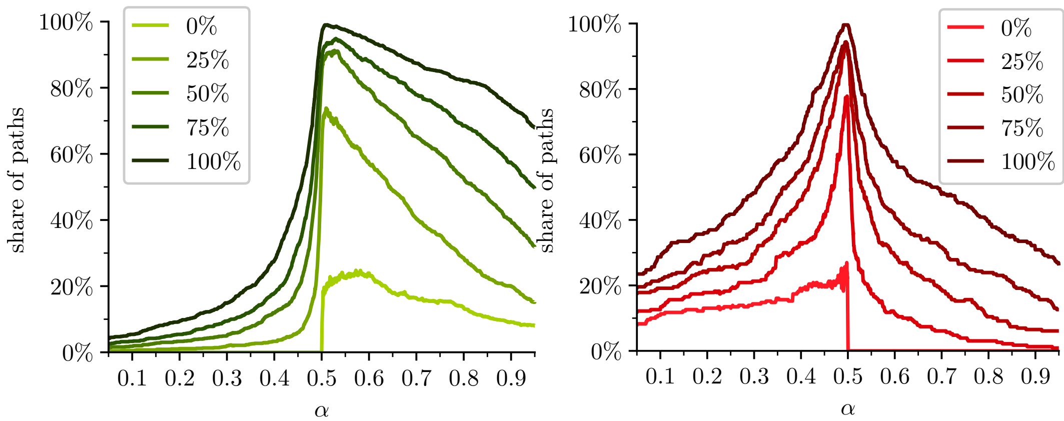


Figure 13: For every α value, the average deficiency over all evaluated paths is displayed. For the interval $\alpha \in [0.4824, 0.5980]$ (gray), the average deficiency is lower than 50% (indicated by the dashed line). For $\alpha = 0.4120$ (red), the symmetric value to $\alpha = 0.5980$ (green), assuming an avoidance of official cycle routes, the deficiency is three times as high.

Even if a cyclist wants to stick to official cycle routes, they might not take the optimal path because the difference between the optimal path and the path taken is too small to be noticeable. Still, in our mathematical model, a milestone needs to be inserted. Therefore, instead of only looking at the optimal segmentations, we can evaluate for which α values a path can be segmented with a given maximum deficiency. Not requiring a perfect deficiency of zero allows for introducing some inaccuracies into the model. Specifically, for a path P, we call a balance factor α *acceptable* if $\Delta(\alpha) \leq \Delta^*$ for a given maximum deficiency Δ^*. Figure 14 shows the share of trajectories in each of the groups that are acceptable for maximum deficiencies of $\{0\%, 25\%, 50\%, 75\%, 100\%\}$. Following the definition of the groups, for $\Delta^* = 0\%$, group PRO contains no acceptable paths for $\alpha < 0.5$, and group CON has no acceptable paths for $\alpha > 0.5$. When the maximum allowed deficiency is increased, the share of acceptable paths naturally increases for a given α. Note, however, how the



(a) group PRO (b) group CON

Figure 14: Share of trajectories per group acceptable for a given maximum deficiency Δ^*. Each line per graph shows one maximum deficiency value between 0% and 100%. Note that the lines for 0% are already displayed in Figure 12.

shapes of the plotted lines change: while the lines for group PRO are very asymmetric with a steep increase at around $\alpha = 0.5$, the lines for group CON get more or less symmetric around $\alpha = 0.5$. In fact, for $\Delta^* = 100\%$, the share of acceptable paths for $\alpha > 0.5$ is larger than the share of acceptable paths for $\alpha < 0.5$. While cyclists of this group seem to avoid official cycling routes, the earlier-mentioned observations reveal that this is only a minor repulsion.

Next to the inference of routing preferences, our approach can be used to compress trajectory data. Assuming that there are no equal-cost s-t-paths in the graph G, a path P in G can be fully reconstructed given s, t, α, and the corresponding milestones. The paths in our data set consist of 172614 vertices in G. Using our approach, this data can be expressed by 2357 vertices only, resulting in a compression factor of 98.63%.

Finally, we take a look at the running times. For each of the 2750 paths, we measured the running time of our algorithm. The median running time of our algorithm per path was 1.40 minutes, and the maximum running time was 4165 minutes. Of the 2750 runs, 984 runs needed more than five minutes, while 1307 required less than one minute. We further observe that the shortest path queries dominate the running time. More precisely, the shortest path queries consume 60% of the running time on average. Hence, the shortest path queries offer considerable potential for further improvement in the running times. Until now, we are using a bidirectional variant of Dijkstra's algorithm [13] as provided by JGraphT. In previous years, speed-up techniques such as contraction hierarchies have been optimized for multi-dimensional routing criteria. Using the algorithm presented by Funke et al. [19] would result in considerable speed-ups for our approach.

6 Conclusion and Future Work

In this article, we have presented and elaborated an idea for analyzing user preferences in bicriteria routing models. To this end, we have formalized the problems OPTIMALITYRANGE and MILESTONESEGMENTATION. Assuming integral edge costs in the road graph, we developed and presented algorithms that yield exact solutions for both problems. Both algorithms are exact and efficient with an overall running time in $\mathcal{O}(h^2 \cdot (h + \mathrm{SPQ} \cdot \log(Mn))$. We have demonstrated the benefits of our approach with experimental results obtained with a prototypical implementation. In contrast to previous work, our approach segments a user's path into optimal sub-paths of the same underlying cost function. On the one hand, this improves the approach's utility by providing an explicit balance factor for future routing. On the other hand, this also reduces the risk of overfitting the model.

In our experiments, we analyzed the extent to which cyclists prefer official cycle routes over the shortest paths. We did this by determining balance factors that yield minimal milestone segmentations. Out of the 2750 trajectories we processed, only 483 were α-optimal paths in the road network. This allows us to conclude that the considered criteria—geodesic length and usage of official cycle routes—do not suffice to fully explain how cyclists choose their routes. On the other hand, we could segment each of the given trajectories into a small number of α-optimal paths. Thus, we can conclude that the bicriterial model yields reasonable results, especially if we take its simplicity into account. The high number of trajectories having an optimal segmentation for α values close to 0.5 affirms the importance of minimizing the geodesic distance when planning a route. However, it is only a little more than 35% of all trajectories with a minimal segmentation for exactly 0.5. In particular, considering all trajectories, the least number of necessary milestones is achieved for $\alpha \approx 0.5013$ (see Figure 13), which indicates that the average cyclist in our experiment prefers official cycle paths. Furthermore, more than 36% of all trajectories have minimal milestone segmentations only if we assume that official cycle routes are preferred, i.e., if $\alpha > 0.5$. That means a large group of cyclists accept actual detours to use cycle routes. Here, the highest number of minimum milestone segmentations can be achieved assuming that cyclists are willing to cover 50% distance extra, i.e., $\alpha \approx 0.5721$; see Fig. 14.

Although we developed an efficient algorithm, the observed running times are too long to be suitable for large-scale applications. On the one hand, running times can probably be substantially reduced by using specialized shortest-path algorithms, such as those presented by Funke and Storandt [20] and Funke et al. [19]. On the other hand, future work could relax the exactness of our algorithm to speed up the computation. We currently think of three ways to achieve this: firstly, a deficiency could already be introduced in the computation of the optimality ranges. Instead of strictly requiring a subpath to be optimal, small losses in the cost function could be accepted. Secondly, instead of computing the milestone segmentation continuously for $\alpha \in [0, 1]$, the interval could be sampled. Last but not least, the method for guaranteeing the existence of a milestone segmentation increases the number of vertices per path by a factor of two and, thus, the size of the interval matrix by a factor of four. The definition of a milestone segmentation could be relaxed to a segmentation of basic path components (i.e., optimal subpaths or single, non-optimal edges) [29]. While this might lead to additional milestones compared to our current approach, the introduction of the additional vertices is not needed. This has the added benefit that the graph remains unchanged for different paths, benefiting the compression of the data.

The results of our algorithm yield much more information than we have used and analyzed in this article. Besides testing sets of trajectories on the relevance of specific criteria, we think of a different application with a more critical role for individual trajectories. Figure 10 shows that the milestones are not at all distributed equally along the path. In particular, there are accumulations of milestones for various α values. This data promises information about the surrounding road network, which may be interesting for traffic planners as it indicates where improvements of the considered network are reasonable or necessary.

An interesting open question is whether the discussed problems provide a basis for efficient algorithms if more than two criteria are considered. Using our algorithm, one approach for this multi-criteria problem is first to merge two criteria into a combined criterion. Then, this combined criterion can be merged with an additional criterion. This process can be iterated for an arbitrary amount of criteria. Initial tests of this method using six different criteria show promising results. Nonetheless, there are open questions regarding this approach. For example, the influence of the merging order on the final result needs to be analyzed. This analysis will be part of future research.

Finally, we note that users of online route planners such as Google Maps[4] frequently plan their routes by (first) computing a route between a start and an end vertex and (second) adding intermediate vertices to tailor the route according to their preferences. We think our algorithm is beneficial in this route-planning scenario since by using the α-value that yielded the best milestone segmentation for existing trajectories; the user will probably be able to plan new routes with few intermediate vertices. To verify this assumption, however, a further evaluation of our method is necessary, particularly in interactive route planning.

Acknowledgments

This research was supported by the German Research Foundation (DFG) [grant number 5451/7-1], within the DFG priority program *VGIscience* [grant number 1894].

References

[1] AHMED, M., KARAGIORGOU, S., PFOSER, D., AND WENK, C. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica 19*, 3 (2015), 601–632. doi:10.1007/s10707-014-0222-6.

[2] ALEWIJNSE, S., BUCHIN, K., BUCHIN, M., SIJBEN, S., AND WESTENBERG, M. A. Model-based segmentation and classification of trajectories. *Algorithmica 80*, 8 (2018), 2422–2452. doi:10.1007/s00453-017-0329-x.

[3] ALEWIJNSE, S. P. A., BUCHIN, K., BUCHIN, M., KÖLZSCH, A., KRUCKENBERG, H., AND WESTENBERG, M. A. A framework for trajectory segmentation by stable criteria. In *Proc. 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '14)* (Dallas Texas, 2014), Association for Computing Machinery, pp. 351–360. doi:10.1145/2666310.2666415.

[4] www.google.com/maps

[4] ARONOV, B., DRIEMEL, A., KREVELD, M. V., LÖFFLER, M., AND STAALS, F. Segmentation of trajectories on nonmonotone criteria. *ACM Transactions on Algorithms 12*, 2 (2016), 1–28. doi:10.1145/2660772.

[5] BALTEANU, A., JOSSÉ, G., AND SCHUBERT, M. Mining driving preferences in multi-cost networks. In *Proc. 13th International Symposium on Advances in Spatial and Temporal Databases (SSTD '13)* (Berlin, Heidelberg, 2013), Lecture Notes in Computer Science (LNISA), Springer, pp. 74–91. doi:10.1007/978-3-642-40235-7_5.

[6] BARTH, F., FUNKE, S., JEPSEN, T. S., AND PROISSL, C. Scalable unsupervised multi-criteria trajectory segmentation and driving preference mining. In *Proc. 9th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (BIGSPATIAL '20)* (New York, NY, USA, 2020), Association for Computing Machinery, pp. 1–10. doi:10.1145/3423336.3429348.

[7] BARTH, F., FUNKE, S., AND PROISSL, C. Preference-based trajectory clustering - an application of geometric hitting sets. In *32nd International Symposium on Algorithms and Computation (ISAAC 2021)* (Dagstuhl, Germany, 2021), H.-K. Ahn and K. Sadakane, Eds., vol. 212 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 15:1–15:14. doi:10.4230/LIPIcs.ISAAC.2021.15.

[8] BEHR, T., VAN DIJK, T. C., FORSCH, A., HAUNERT, J.-H., AND STORANDT, S. Map matching for semi-restricted trajectories. In *Proc. 11th International Conference on Geographic Information Science (GIScience '21) - Part II* (Dagstuhl, Germany, 2021), K. Janowicz and J. A. Verstegen, Eds., vol. 208 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 12:1–12:16. doi:10.4230/LIPIcs.GIScience.2021.II.12.

[9] BUCHIN, M., DRIEMEL, A., VAN KREVELD, M. J., AND SACRISTAN, V. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Information Science 3*, 3 (2011), 33–63. doi:10.5311/JOSIS.2011.3.66.

[10] CHAO, P., XU, Y., HUA, W., AND ZHOU, X. A survey on map-matching algorithms. In *Databases Theory and Applications* (Cham, 2020), Lecture Notes in Computer Science (LNISA), Springer, pp. 121–133. doi:10.1007/978-3-030-39469-1_10.

[11] CHEN, L., LV, M., YE, Q., CHEN, G., AND WOODWARD, J. A personal route prediction system based on trajectory data mining. *Information Sciences 181*, 7 (2011), 1264–1284. doi:10.1016/j.ins.2010.11.035.

[12] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, 2009.

[13] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1 (1959), 269–271. doi:10.1007/BF01386390.

[14] EHRGOTT, M., WANG, J. Y., RAITH, A., AND VAN HOUTTE, C. A bi-objective cyclist route choice model. *Transportation Research Part A: Policy and Practice 46*, 4 (2012), 652–663. doi:10.1016/j.tra.2011.11.015.

[15] EMOND, C. R., TANG, W., AND HANDY, S. L. Explaining gender difference in bicycling behavior. *Transportation Research Record 2125*, 1 (2009), 16–25. doi:10.3141/2125-03.

[16] FEUERHAKE, U., KUNTZSCH, C., AND SESTER, M. Finding interesting places and characteristic patterns in spatio-temporal trajectories. In *Proc. 8th International Symposium on Location-Based Services* (2011), p. 18.

[17] FRIESE, M., WÄNKE, M., AND PLESSNER, H. Implicit consumer preferences and their influence on product choice. *Psychology & Marketing 23*, 9 (2006), 727–740. doi:10.1002/mar.20126.

[18] FUNKE, S., LAUE, S., AND STORANDT, S. Deducing individual driving preferences for user-aware navigation. In *Proc. 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '16)* (New York, NY, USA, 2016), Association for Computing Machinery, pp. 1–9. doi:10.1145/2996913.2997004.

[19] FUNKE, S., LAUE, S., AND STORANDT, S. Personal routes with high-dimensional costs and dynamic approximation guarantees. In *16th International Symposium on Experimental Algorithms (SEA 2017)* (Dagstuhl, Germany, 2017), C. S. Iliopoulos, S. P. Pissis, S. J. Puglisi, and R. Raman, Eds., vol. 75 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 18:1–18:13. doi:10.4230/LIPIcs.SEA.2017.18.

[20] FUNKE, S., AND STORANDT, S. Polynomial-time construction of contraction hierarchies for multi-criteria objectives. In *2013 Proc. Meeting on Algorithm Engineering and Experiments (ALENEX '13)* (2013), pp. 41–54. doi:10.1137/1.9781611972931.4.

[21] GOODCHILD, M. F. Citizens as sensors: the world of volunteered geography. *GeoJournal 69*, 4 (2007), 211–221. doi:10.1007/s10708-007-9111-y.

[22] GOTSMAN, R., AND KANZA, Y. Compact representation of GPS trajectories over vectorial road networks. In *Advances in Spatial and Temporal Databases* (Berlin, Heidelberg, 2013), Lecture Notes in Computer Science (LNISA), Springer, pp. 241–258. doi:10.1007/978-3-642-40235-7_14.

[23] GREENWALD, A. G., AND BANAJI, M. R. Implicit social cognition: attitudes, self-esteem, and stereotypes. *Psychological review 102*, 1 (1995), 4–27. doi:10.1037/0033-295x.102.1.4.

[24] GRÜNWALD, P. D., MYUNG, I. J., AND PITT, M. A. *Advances in minimum description length: Theory and applications*. Neural Information Processing. MIT press, 2005. doi:10.7551/mitpress/1114.001.0001.

[25] HARDINGHAUS, M., AND NIELAND, S. Assessing cyclists' routing preferences by analyzing extensive user setting data from a bike-routing engine. *European Transport Research Review 13*, 1 (2021), 1–19. doi:10.1186/s12544-021-00499-x.

[26] HARDINGHAUS, M., AND WESCHKE, J. Attractive infrastructure for everyone? different preferences for route characteristics among cyclists. *Transportation Research Part D: Transport and Environment 111* (2022), 103465. doi:10.1016/j.trd.2022.103465.

[27] JIANG, L., CHEN, C., CHEN, C., HUANG, H., AND GUO, B. From driving trajectories to driving paths: a survey on map-matching algorithms. *CCF Transactions on Pervasive Computing and Interaction* (2022), 1–16. doi:10.1007/s42486-022-00101-w.

[28] KNAPEN, L., HARTMAN, I. B.-A., AND BELLEMANS, T. Using path decomposition enumeration to enhance route choice models. *Future Generation Computer Systems 107* (2020), 1077–1088. doi:10.1016/j.future.2017.12.053.

[29] KNAPEN, L., HARTMAN, I. B.-A., SCHULZ, D., BELLEMANS, T., JANSSENS, D., AND WETS, G. Determining structural route components from GPS traces. *Transportation Research Part B: Methodological 90* (2016), 156–171. doi:10.1016/j.trb.2016.04.019.

[30] LAUBE, P. *Computational Movement Analysis*. Springer Briefs in Computer Science. Springer, Cham, 2014. doi:10.1007/978-3-319-10268-9.

[31] LERIN, P. M., YAMAMOTO, D., AND TAKAHASHI, N. Encoding network-constrained travel trajectories using routing algorithms. *International Journal of Knowledge and Web Intelligence 4*, 1 (2013), 34–49. doi:10.1504/IJKWI.2013.052724.

[32] LI, Q., ZHENG, Y., XIE, X., CHEN, Y., LIU, W., AND MA, W.-Y. Mining user similarity based on location history. In *Proc. 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '08)* (New York, NY, USA, 2008), Association for Computing Machinery, pp. 1–10. doi:10.1145/1463434.1463477.

[33] MAJUMDAR, B. B., MITRA, S., AND PAREEKH, P. Methodological framework to obtain key factors influencing choice of bicycle as a mode. *Transportation research record 2512*, 1 (2015), 110–121. doi:10.3141/2512-13.

[34] MICHAIL, D., KINABLE, J., NAVEH, B., AND SICHI, J. V. JGraphT - a Java library for graph data structures and algorithms. *ACM Transactions on Mathematical Software 46*, 2 (2020), 1–19. doi:10.1145/3381449.

[35] NOSEK, B. A. Implicit–explicit relations. *Current Directions in Psychological Science 16*, 2 (2007), 65–69. doi:10.1111/j.1467-8721.2007.00477.x.

[36] OEHRLEIN, J., FÖRSTER, A., SCHUNCK, D., DEHBI, Y., ROSCHER, R., AND HAUNERT, J.-H. Inferring routing preferences of bicyclists from sparse sets of trajectories. In *Proc. 3rd International Conference on Smart Data and Smart Cities* (2018), vol. IV-4/W7 of *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 107–114. doi:10.5194/isprs-annals-IV-4-W7-107-2018.

[37] OEHRLEIN, J., NIEDERMANN, B., AND HAUNERT, J.-H. Inferring the parametric weight of a bicriteria routing model from trajectories. In *Proc. 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '17)* (New York, NY, USA, 2017), Association for Computing Machinery, pp. 59:1–59:4. doi:10.1145/3139958.3140033.

[38] ROSSETTI, T., GUEVARA, C. A., GALILEA, P., AND HURTUBIA, R. Modeling safety as a perceptual latent variable to assess cycling infrastructure. *Transportation Research Part A: Policy and Practice 111* (2018), 252–265. doi:10.1016/j.tra.2018.03.019.

[39] SENER, I. N., ELURU, N., AND BHAT, C. R. An analysis of bicycle route choice preferences in Texas, US. *Transportation 36*, 5 (2009), 511–539. doi:10.1007/s11116-009-9201-4.

[40] TURTON, I. Geo tools. In *Open Source Approaches in Spatial Data Handling*, G. B. Hall and M. G. Leahy, Eds., Advances in Geographic Information Science, vol 2 (AGIS). Springer, Berlin, Heidelberg, 2008, ch. 8, pp. 153–169. doi:10.1007/978-3-540-74831-1_8.

[41] WILSON, T. D., LINDSEY, S., AND SCHOOLER, T. Y. A model of dual attitudes. *Psychological review 107*, 1 (2000), 101–126. doi:10.1037/0033-295x.107.1.101.

[42] YUAN, J., ZHENG, Y., ZHANG, C., XIE, W., XIE, X., SUN, G., AND HUANG, Y. T-drive: driving directions based on taxi trajectories. In *Proc. 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)* (2010), Association for Computing Machinery, pp. 99–108. doi:10.1145/1869790.1869807.

[43] ZHANG, L., DALYOT, S., EGGERT, D., AND SESTER, M. Multi-stage approach to travel-mode segmentation and classification of GPS traces. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-4/W25* (2011), 87–93. doi:10.5194/isprsarchives-XXXVIII-4-W25-87-2011.

[44] ZHENG, Y. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology 6*, 3 (2015), 1–41. doi:10.1145/2743025.

[45] ZHENG, Y., AND ZHOU, X. *Computing with Spatial Trajectories*. Springer, New York, NY, USA, 2011. doi:10.1007/978-1-4614-1629-6.