Effective User Interface Design in Route Planners for Cyclists and Public Transportation Users: An Empirical Analysis of Route Selection Criteria

Submitted: July 31, 2007
Word Count: 5,484 words plus 8 figures/tables at 250 words each = 7,484 words

By
Hartwig H. Hochmair
University of Florida, Geomatics Program
3205 College Avenue
Ft. Lauderdale, FL-33314
Phone: (954) 577-6316
Fax: (954) 475-4125
E-mail: hhhochmair@ufl.edu
ABSTRACT
The purpose of electronic route planners is to provide the optimal route to the user. The optimal route is defined over a set of evaluation criteria considered by the user. Requesting user input on all possible route selection criteria within the route finding process would overcrowd the user interface of the route planner, especially when several transportation modes are combined. Based on a set of bicycle trips that were optimized for various criteria in a multi-modal, inner-urban street network, an exploratory study based on Principal Components Analysis (PCA) identifies correlations between the performances of route selection criteria. The results suggest that the variability of routes can be more parsimoniously described with a smaller set of factors. The implications of these findings to simplify the user interface design of electronic multi-modal route planners for cyclists are discussed.
INTRODUCTION
Various existing route planners provide access to public transportation systems, which encourage their use and help reduce individual car traffic. Current initiatives that promote bicycle transportation in combination with public transportation means [1] show the increasing popularity of this transportation type.

Especially when it comes to multi-modal route planning, the number of optimization criteria involved in this type of multi-criteria decision problem is extensive. As humans are capable of dealing only with a limited amount of information [2] a simple user interface is crucial for the success of a route planning system. This research aims at finding an inherent number of underlying factors that explain the variability of the route selection criteria under consideration, and could thus be used to simplify the user interface. Previous research found that route selection criteria used for bicycle trip planning in urban street environments can be grouped into four general criteria, namely fast, safe, simple, attractive [3; 4]. The exploratory study of this paper extends these findings to bicycle route planning in combination with transit use.

The next section reviews relevant concepts of route selection and the functionality of current route planner applications. This is followed by an introduction to the empirical simulation study. The subsequent section describes the implemented modeling approach used to run the multi-modal route searches. The set of routes found is then analyzed with regard to grouping of route selection criteria, and the implications for the user interface design of route planners are discussed.

ROUTE SELECTION CRITERIA FOR CYCLISTS AND PUBLIC TRANSPORTATION USERS
Route selection problems involve a set of route alternatives from which a choice must be made under consideration of several evaluation criteria. A decision rule is a procedure that integrates information on alternatives and the decision maker’s preferences to produce an evaluation of the set of alternatives. With a compensatory decision rule, the high performance of an alternative achieved in one or more criteria can compensate for the weak performance of the same alternative in other criteria. Contrarily, under the non-compensatory approach a poor performance by an alternative in a criterion cannot be offset by another criterion’s good outcome. Evaluation criteria (also called attributes or decision variables) considered in route choice include benefit criteria and cost criteria. With a benefit criterion, a higher attribute value, also called attribute score or raw data, of an alternative means a higher attractiveness of that attribute score (e.g., number of cultural landmarks along a route), whereas for a cost criterion, a lower attribute score is more attractive (e.g., number of transfers). Eliminatory constraints impose limitations on the set of decision alternatives. An alternative is feasible if it satisfies all eliminatory constraints.

Eliciting route preferences for cyclists has become a prominent research topic over the last two decades [5-8] using various survey techniques, such as self-explicated models, stated preference surveys, and revealed preference surveys. A detailed overview of factors that impact bicycle use and route choice can be found in [8], including characteristics of facilities, individuals, and environments. Preferential behavior has also been elicited for public transportation use [9] and combined bicycle-public transit transportation [10]. Prominent criteria identified in these studies were the high importance of short travel time, the willingness to incur additional travel time to use bicycle facilities, or preference for trips with few transfers between
public transportation routes. Route search often involves multiple conflicting criteria to be met or optimized and thus include a compensatory decision component [11; 12]. This means that the optimal route can often not be characterized by a single optimization criterion.

**Route Planner Designs**
Existing route planners vary in the number of route selection criteria they support. Some tourist route planners for cyclists allow for manual selection from a set of pre-defined routes [13], not eliciting an explicit optimization criterion from the user. Most route planners either apply a fixed optimization function, such as shortest path, or allow the user to select a single optimization criterion, such as safe route [14]. Some designs provide an additional function for importance weighting, such as weighting between fast, scenic, and short route [15], and setting eliminatory constraints on the route, such as avoiding street surfaces with cobble stones [16]. A recent empirical study showed that slider bars, which provide importance weighting functionality, support the user better in defining the optimal route than radio buttons or hyperlinks [17]. Advanced multi-modal door-to-door route planners allow the user to select a single optimization function, such as fewest transfers, and to set constraints on the route, such as the request to leave the bicycle at the transit station [18]. In these multi-modal route planners it is, however, difficult, if not impossible, to specify compromise routes that optimize several criteria at a time.

**STUDY SETUP**

**Test Network**
The dataset used is part of the street and public transportation network of Vienna, Austria (FIGURE 1). The test network has 404 nodes (i.e., intersections or ends of cul-de-sacs), and 631 links connecting the nodes. Some of the links have one-way restrictions for car or bicycle transportation, and turn restrictions are considered in the route search as well. Public transportation is provided through three metro lines (U2, U3, U6) (dashed lines), 15 tram lines (5, 6, 9, 18, 33, 37, 38, 40, 41, 42, 43, 44, 46, 49, J) (dark-red, continuous lines), and two bus routes (13A, 48A) (light-blue, continuous lines). Metro stations are marked as a “U” symbol, and tram and bus stops as color-coded circles.

For the study, a set of 50 start-destination pairs was selected, and for each pair, a set of Pareto optimal paths was computed. The search algorithm was programmed in Delphi and run on a desktop PC.
FIGURE 1 Street network and public transportation routes for the Vienna test area.

Included Route Selection Criteria
21 criteria were selected and implemented in the explorative study for finding the set of Pareto optimal routes. The distinction into link-related and node-related criteria below refers to whether the attribute values for that criterion are stored with links or nodes in the network graph. (c) stands for cost, and (b) for benefit criteria.

- link related (at the street level): separate bicycle facility (b), parks (b), cultural landmarks (b), shopping streets (b), walking or biking distance (c), heavy car traffic without bicycle facilities (c), slope (c), unpaved (c), tramway tracks on street (c), driving against oncoming car traffic in one-way streets without separate bicycle facilities (c)
- node related (at the street level): traffic lights (c), intersections (c), turns (c), street crossings during public transportation transfer (c)
- link related (at the public transportation level): public transportation portion [%] (b), number of transfers (c), fare (c), transfer waiting time (c)
- node related (at the public transportation level): number of choice options at transfer (c)
- link related (at the street and public transportation level): travel time (c)
- link and node related (at the street and public transportation level): turns and transfers combined (c)
ROUTE COMPUTATION AND MODELING APPROACH

Problem Formulation and Challenges
We analyze underlying factors in route characteristics from a set of Pareto optimal routes. Such a route set would be used as the pool of alternatives from which the best route would be returned.

Single-criterion shortest path problems (SSP) find the shortest path using a single optimization criterion, e.g., travel time, whereas multi-criteria shortest path problems (MSPP) consider two or more independent criteria in evaluating the solution. Solving a SSP or MSPP can help to build the Pareto optimal route set. Historically many MSPP are reduced to a SPP by using a weighted linear combination of all criteria as the cost function. However, it may be difficult to compute an appropriate set of weightings for the criteria involved, and optimal solutions may be overlooked [19]. SSP and MSPP approaches cannot be used to solve problems that involve benefit criteria (negative cost). There exists no polynomial-time algorithm for the longest path problem if the network contains negative cycles, which is generally true for street networks. This paper uses a genetic algorithm to handle this problem.

Multi-modal route planning systems need to account for the transfer between different transportation modes, which involves modeling the physical complexity of the transfer [9]. Further, a dynamic waiting time needs to be modeled that depends on the time the commuter arrives at the station and the departure time of the vehicle. For the Vienna public network a fixed amount of 2 Euro is charged per trip. The fare is independent of the number of transfers made or stops traveled. Taking a bicycle onto the vehicle costs an additional Euro.

Network Modeling and Graphs
The common model of a road network is a weighted, directed graph model G=(V,E). A node graph G=(V, E) comprises a set of vertices V and edges E connecting these vertices. Each edge is described by the pair of vertices (called endnodes) that it links. A loop is an edge where the endnodes of that edge are identical. An undirected graph is a graph whose edges are unordered pairs of vertices. A directed graph has directions assigned to its edges, and edges are represented as arrows.

Edges have some cost attached for cost criteria, w: E → R+. If the costs c for traversing two consecutive edges e_i, e_j are greater than w(e_i)+w(e_j), then traversing the shared vertex contributes to c. Thus another cost function w: E → E → R+ can be introduced for the vertex costs. A node graph G (the term primal graph is used for better distinction) can be converted to a line graph D(N_D, E_D) if edges of G are mapped to different vertices in D, pairs of consecutive edges in G are mapped to different edges in D, and if there exists a cost function for vertices w^V: N_D → R+ and edges w^E: E_D → R+ in D [20]. Thus cost functions for traversing a segment and a vertex in the primal graph can be attached as attributes to graph elements in a line graph.

FIGURE 2a shows a primal graph consisting of three undirected edges (gray). Each undirected edge is first replaced with two directed edges (black). From this, the line graph D with six edges (black) can be created (FIGURE 2b). The weighting used in the figure demonstrates the modeling of turn cost, i.e., w=1 for turn edges, and w=0 for straight edges.
FIGURE 2 Primal graph shown with undirected (gray) and directed edges (black) (a), and the corresponding line graph (b, black) with turn costs on edges (after [20]).

The line graph is particularly useful for finding routes with the fewest traffic lights, fewest intersections, or fewest turns, where the SP algorithm is executed on the line graph of the primal street network graph. In search for routes that minimize one of the street bound cost criteria (e.g., turns), public transportation segments were excluded from the SP route search.

Modeling Travel on Public Transportation Routes
To model the transfer between routes, a technique that is based on node explosion [21] is used. This approach takes the primal street graph and transforms it into the expanded graph $G'=(V', E')$ through the following steps (slightly modified from the original reference):

For each directed public transportation route $K$, do the following (FIGURE 3):

- For each stop $v_i$ along the route add a new vertex $v_{i,K}$ to the expanded graph.
- Replace each directed edge $e_{i,j}$ along its route by three new directed edges, namely (a) an access edge $e_{i,i,K}$ that connects the access point $v_i$ to the transit route, (b) a traveling edge $e_{i,K-i,j,K}$ that represents the commuter travel on the transit route from stop $v_i$ to $v_j$, and (c) an alighting edge $e_{j,K,j}$ for exiting the transit route $K$ at stop $v_j$.
- Add transfer edges $e_{i,K-i,L}$ to all other traveling edges.
- Add a parking edge $e_{i-i}$ (a loop), which represents the traveler parking the bicycle before entering the transit route.

After the node explosion, the expanded graph is mapped to a line graph to facilitate all necessary shortest path computations.

FIGURE 3 Node explosion for one public transportation route (a) and two public transportation routes (b). The loop is omitted in b) for readability reasons.
Travel times for edges of the primal graph were computed as segment length divided by travel speed. An average speed of 12 km/h was assumed for traveling by bike, and 4 km/h for walking. At the beginning of each trip, the traveler was “equipped” with a bicycle, and thus took the role of a cyclist. If the bicycle was left at a station, the role would change to pedestrian. Travel times between stop on public transportation routes were read from time tables. A time stamp was stored with the fictive traveler to dynamically calculate the waiting times on transfer edges when entering a transit route. Transfer complexity was modeled as the total number of directions the traveler can choose from at a transit station. Each transfer was assigned a Boolean variable that describes whether a street needs to be crossed when transferring between two transit routes.

**Pareto Front**

The simulation is used to find a Pareto optimal set of routes which will be analyzed for potential underlying factors explaining the variability of route characteristics. Let $a$ designate a feasible route within the set of all alternatives $A$. Each route is evaluated with regard to $n$ attributes or criteria, such as route length. Thus, each route can be considered as being mapped into a point in an $n$-dimensional consequence space (which is also known as criteria space or attribute space). The subspace that satisfies potential eliminatory constraints is often referred to as solution space. In multi-attribute decision theory, the solution space is already partially ordered even without making any multicriteria decisions. The distinction is made between dominated and non-dominated solutions. Solution $a$ is said to dominate $b$ if $a$ performs at least as well on all attributes as $b$, and is superior on at least one attribute to $b$. A solution is non-dominated if there is no solution that dominates it. All dominated solutions can be eliminated from consideration before the multicriteria decision is made. Non-dominated solutions, i.e., the Pareto optimal set, are on the boundary of the solution space, which is also called Pareto frontier, Pareto front, or Pareto optimal tradeoff-surface.

FIGURE 4 depicts an example of a two-criteria space with the cost criterion “Travel Time” (to be minimized) and the benefit criterion “Separate Bicycle Lane” (to be maximized). A population of paths is plotted using each individual’s attribute values as coordinates. Value pairs are shown for six data points. Individuals A, B, C, D, and E are non-dominated. For these individuals, there exists no other individual that performs at least equally well on both criteria and better on at least one criterion. As opposed to this, for example, individual F performs worse in both criteria than E, and is therefore dominated and excluded from the Pareto front.

FIGURE 4 Criteria space and Pareto front in a two-criteria problem.
Genetic Algorithm Framework

Over the last three decades genetic algorithms (GA) [22], which are more recently also referred to as evolutionary algorithms (EAs), have gained high importance for exploring the Pareto optimal front in multi-objective problems that are too complex to be solved by exact methods [23]. [24] provides an overview of common GA methods used for finding the Pareto Optimal set. Independent sampling performs multiple single-criterion searches to optimize one criterion or a linear combination of criteria at a time where weights are varied from search to search. Simultaneous parallel search for multiple members of the Pareto optimal front includes among others criteria selection, aggregation selection, and Pareto selection. The latter method favors Pareto optimal solutions above others, and no preferences are given within the Pareto optimal equivalence class. Many of these efforts have incorporated some form of active diversity promotion, such as GA niching, to find and maintain an even distribution sampling of points along the Pareto front.

FIGURE 5a depicts the structure of the genetic algorithm used in the exploratory network application. An initial population $X_0$ is created through random walking [19] on the line graph of the expanded network graph (FIGURE 3). In addition to this, initial routes are found through separate single-criterion shortest path computations on the line graph that minimize selected cost criteria. This gives the “corners” of the Pareto optimal surface for cost criteria (see also point A in FIGURE 4). Next a repeated standard one-point crossover and mutation are executed to find additional solutions on the Pareto front. A copy of $X_i$ ensures that good solutions are not destroyed during crossover and mutation. This is followed by the elimination of duplicate solutions from the intersection of $X_i$ and the modified set $X_i^{gen}$, and Pareto elitist selection, which gives the Pareto optimal set of candidates $X^{PO(1)}$. This set provides the population for the next iteration and grows with each iteration. As no decision making and multicriterion objective is required in finding the Pareto optimal front at this point, a fitness function and quality metric are not included.
Crossover describes the process where two chromosomes (the parents) line up and then swap the portions of their genetic code beyond the crossover point, which creates two offspring. In the framework of this paper, candidate routes can be viewed as chromosomes, with the sequence of route segments being their genes. A one-point crossover is used in the model (FIGURE 5b).

Mutations make a random modification of the chromosomes. Whereas mutation is traditionally applied on one string (chromosome), the approach in this paper uses two parents to create a mutated offspring that replaces one parent. A random path is computed to connect the two randomly chosen points on both parent routes and to mutate the first parent (FIGURE 5b).

After the genetic operations, the offspring need to be checked for possible failures, such as violation of turn restrictions or self-intersection. Offspring with a biking/walking distance more than twice the shortest path biking/walking distance, or offspring with more than twice the fastest travel time were removed from the route pool, as these routes were presumed unacceptable by a user.
RESULTS OF THE EXPLORATIVE STUDY

Pareto Optimal Paths
A total of 1278 Pareto optimal paths were found in the search process for the 50 start-destination pairs. On the average, Pareto optimal paths had a total distance of 3222m (SD=±933), a walking respectively cycling portion of 2358m (SD=±1151), an average trip time of 17.1 minutes (SD=±5.3), and utilized 1.84 different public transit routes per trip (SD=±0.97). Routes that used public transportation showed an average waiting time of 4.0 minutes (SD=±2.8). 5.8% of the overall trip distance was made on metro, 12.9% on tram, and 6.1% on bus, which amounts to 24.9% for all public transportation means. In 121 trips the bike was parked at an access point before using the public route, whereas in 535 trips the bicycle was taken onto the public transportation vehicle. In 622 trips no public transit was used.

Principal Components Analysis
This section explores whether the variability of the Pareto optimal routes can be more parsimoniously described by a smaller number of components using Principal Components Analysis (PCA).

Except for the public transportation portion of a route (given in %) and trip fare, route values, such as the portion of shopping streets of a route in meters, were divided by the shortest path distance of its start-destination pair. This scaling makes route characteristics for different start-destination pairs comparable, because attribute values, such as sights passed by, will generally increase with a longer trip distance. We model the complexity of a multi-modal route as a linear combination of turns and route transfers. Based on the Pareto optimal set of routes retrieved, a complexity measure of \(1*\text{turns} + 3*\text{transfers}\) was found to yield positive correlations between these two attributes and the combined measure (\(r_{\text{turn,c}}=0.690; r_{\text{transfer,c}}=0.453; p=0.000\)), so that this linear combination was used in PCA. After the scaling, all attributes were normalized using z-scores.

TABLE 1 lists the variance accounted for by successive components. The eigenvalues in the “Total” column describe the observed variable variance explained by each component. For example, the first component with an eigenvalue of 8.139 accounts for about 39% of the variability of the 21 variables. By applying the Kaiser criterion, which suggests retaining all components with an eigenvalue of 1 or higher, PCA yields five components. This is one component more than suggested for bicycle route planners without public transit [3; 4]. The first five components accounts for about 76% of the variability, as shown under “Cumulative %”.

TABLE 1

<table>
<thead>
<tr>
<th>Component</th>
<th>Variance Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39%</td>
</tr>
<tr>
<td>2</td>
<td>16%</td>
</tr>
<tr>
<td>3</td>
<td>10%</td>
</tr>
<tr>
<td>4</td>
<td>7%</td>
</tr>
<tr>
<td>5</td>
<td>4%</td>
</tr>
</tbody>
</table>
### TABLE 1 Results of the PCA: Extracted components with initial eigenvalues and explained variance.

<table>
<thead>
<tr>
<th>Component</th>
<th>Initial Eigenvalues</th>
<th>% of Variance</th>
<th>Cumulative %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.193</td>
<td>39.015</td>
<td><strong>39.015</strong></td>
</tr>
<tr>
<td>2</td>
<td>3.141</td>
<td>14.955</td>
<td><strong>53.970</strong></td>
</tr>
<tr>
<td>3</td>
<td>1.984</td>
<td>9.449</td>
<td><strong>63.419</strong></td>
</tr>
<tr>
<td>4</td>
<td>1.455</td>
<td>6.929</td>
<td><strong>70.347</strong></td>
</tr>
<tr>
<td>5</td>
<td>1.131</td>
<td>5.385</td>
<td><strong>75.732</strong></td>
</tr>
<tr>
<td>6</td>
<td>0.897</td>
<td>4.272</td>
<td>80.004</td>
</tr>
<tr>
<td>7</td>
<td>0.771</td>
<td>3.669</td>
<td>83.673</td>
</tr>
<tr>
<td>8</td>
<td>0.736</td>
<td>3.504</td>
<td>87.177</td>
</tr>
<tr>
<td>9</td>
<td>0.601</td>
<td>2.863</td>
<td>90.041</td>
</tr>
<tr>
<td>10</td>
<td>0.567</td>
<td>2.699</td>
<td>92.740</td>
</tr>
<tr>
<td>11</td>
<td>0.344</td>
<td>1.638</td>
<td>94.378</td>
</tr>
<tr>
<td>12</td>
<td>0.333</td>
<td>1.584</td>
<td>95.962</td>
</tr>
<tr>
<td>13</td>
<td>0.248</td>
<td>1.180</td>
<td>97.142</td>
</tr>
<tr>
<td>14</td>
<td>0.167</td>
<td>0.793</td>
<td>97.935</td>
</tr>
<tr>
<td>15</td>
<td>0.146</td>
<td>0.697</td>
<td>98.632</td>
</tr>
<tr>
<td>16</td>
<td>0.097</td>
<td>0.462</td>
<td>99.093</td>
</tr>
<tr>
<td>17</td>
<td>0.085</td>
<td>0.406</td>
<td>99.499</td>
</tr>
<tr>
<td>18</td>
<td>0.056</td>
<td>0.266</td>
<td>99.765</td>
</tr>
<tr>
<td>19</td>
<td>0.038</td>
<td>0.181</td>
<td>99.946</td>
</tr>
<tr>
<td>20</td>
<td>0.011</td>
<td>0.054</td>
<td>100.000</td>
</tr>
<tr>
<td>21</td>
<td>0.000</td>
<td>0.000</td>
<td>100.000</td>
</tr>
</tbody>
</table>

Component loadings describe the correlation between components and variables. To obtain a clear pattern of loadings an orthogonal rotation that maximizes the variance on the new axes is obtained. The rotated component matrix (TABLE 2) reveals what the different components represent.
TABLE 2 Rotated component matrix. Factor loadings > 0.5 or < -0.5 are printed in boldface.

<table>
<thead>
<tr>
<th>Component</th>
<th>Component 1</th>
<th>Component 2</th>
<th>Component 3</th>
<th>Component 4</th>
<th>Component 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking or biking distance</td>
<td>-0.399</td>
<td><strong>0.721</strong></td>
<td>0.436</td>
<td>0.226</td>
<td>0.198</td>
</tr>
<tr>
<td>Heavy traffic</td>
<td>-0.157</td>
<td>0.046</td>
<td><strong>0.791</strong></td>
<td>0.028</td>
<td>0.025</td>
</tr>
<tr>
<td>Street with tram tracks</td>
<td>-0.308</td>
<td>0.241</td>
<td><strong>0.605</strong></td>
<td>0.216</td>
<td>-0.375</td>
</tr>
<tr>
<td>Traffic lights</td>
<td>-0.390</td>
<td>0.237</td>
<td><strong>0.544</strong></td>
<td>0.249</td>
<td><strong>0.545</strong></td>
</tr>
<tr>
<td>Turns</td>
<td>-0.116</td>
<td><strong>0.889</strong></td>
<td>0.123</td>
<td>0.071</td>
<td>0.003</td>
</tr>
<tr>
<td>Against one-way</td>
<td>-0.076</td>
<td>0.396</td>
<td>-0.360</td>
<td>0.299</td>
<td>-0.101</td>
</tr>
<tr>
<td>Parks</td>
<td>-0.064</td>
<td>0.100</td>
<td>0.001</td>
<td><strong>0.837</strong></td>
<td>0.056</td>
</tr>
<tr>
<td>Shopping street</td>
<td>-0.016</td>
<td>-0.050</td>
<td><strong>0.644</strong></td>
<td>0.111</td>
<td>0.142</td>
</tr>
<tr>
<td>Intersections</td>
<td>-0.400</td>
<td><strong>0.718</strong></td>
<td>0.429</td>
<td>0.198</td>
<td>0.175</td>
</tr>
<tr>
<td>Unpaved surface</td>
<td>-0.126</td>
<td><strong>0.832</strong></td>
<td>-0.182</td>
<td>0.013</td>
<td>-0.186</td>
</tr>
<tr>
<td>Steep slope</td>
<td>-0.276</td>
<td><strong>0.841</strong></td>
<td>-0.007</td>
<td>-0.100</td>
<td>0.267</td>
</tr>
<tr>
<td>Travel time</td>
<td><strong>0.771</strong></td>
<td>0.370</td>
<td>0.204</td>
<td>0.109</td>
<td>0.042</td>
</tr>
<tr>
<td>Transfers</td>
<td><strong>0.885</strong></td>
<td>-0.265</td>
<td>-0.202</td>
<td>-0.114</td>
<td>-0.129</td>
</tr>
<tr>
<td>Trip fare</td>
<td><strong>0.710</strong></td>
<td>-0.316</td>
<td>-0.267</td>
<td>-0.148</td>
<td>-0.171</td>
</tr>
<tr>
<td>Waiting time</td>
<td><strong>0.795</strong></td>
<td>-0.238</td>
<td>-0.161</td>
<td>-0.075</td>
<td>-0.096</td>
</tr>
<tr>
<td>Public Transit portion</td>
<td><strong>0.633</strong></td>
<td><strong>-0.560</strong></td>
<td>-0.350</td>
<td>-0.177</td>
<td>-0.168</td>
</tr>
<tr>
<td>Separate bicycle lane</td>
<td>-0.177</td>
<td>0.017</td>
<td>0.086</td>
<td>-0.012</td>
<td><strong>0.943</strong></td>
</tr>
<tr>
<td>Sights</td>
<td>-0.119</td>
<td>-0.030</td>
<td>0.299</td>
<td><strong>0.808</strong></td>
<td>-0.032</td>
</tr>
<tr>
<td>Crossing street at transfer</td>
<td><strong>0.652</strong></td>
<td>-0.051</td>
<td>-0.013</td>
<td>-0.047</td>
<td>-0.002</td>
</tr>
<tr>
<td>PT choices at transfer</td>
<td><strong>0.738</strong></td>
<td>-0.358</td>
<td>-0.236</td>
<td>-0.121</td>
<td>-0.113</td>
</tr>
<tr>
<td>Combined turns+transfers</td>
<td><strong>0.583</strong></td>
<td><strong>0.692</strong></td>
<td>-0.035</td>
<td>-0.019</td>
<td>-0.099</td>
</tr>
<tr>
<td>The meaning attached to rotated components</td>
<td>fast  simple quiet scenic safe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The meaning attached to rotated components is subjective, however, some tendencies can be identified.

The first component is marked by high loadings on attributes related to public transit use, such as transfers, trip fare or waiting time. As this component is the only one receiving high loadings on travel time, and as travel time is one of the most prominent route selection criteria, this factor should be preferably ascribed to fastness. Reducing the weight for this factor yields faster and simpler routes, reduces the public transit portion and the number of transfers. The complexity related measure (bottom most attribute) receives, however, lower loadings than on the second component.

The second component receives high loadings on walking respectively cycling distance, surface quality, slope, and complexity related criteria. As opposed to the first component, reducing the importance weight of this component yields a simpler route with an increased public transit portion. If considering physical effort (which is afforded by slope segments and unpaved surfaces) to be negatively related to a simple route, this component can be mainly ascribed to route simplicity.

The third component is mostly correlated with quiet and uninterrupted travel which allows the user to reduce streets with heavy traffic or tram tracks, traffic lights, or shopping streets.

The meaning of the fourth component can be ascribed to route scenery, whereas the fifth component has high loadings on safety related attributes.
User Interface Design and Implementation

Using the 21 factor loadings in each component as coefficients in a weighted linear combination with the z-scored criterion values will allow the user to access a maximum number of Pareto optimal routes, which is about 76% of the complete route set in this case, with only five weight settings on the components (see mockup interfaces in FIGURE 6b and c), as opposed to the 21 original criteria (FIGURE 6a). Selecting a route that is optimized for exactly one of the five components can be implemented in an even simpler user interface using radio buttons (FIGURE 6d). With a default setting for weightings or a pre-selected radio button (FIGURE 6d), the multi-criteria decision is taken from the user, and a default optimal route, such as the fastest route, would be returned. It is recommended to provide also a second user interface for detail-oriented users who want to choose from the original 21 criteria.

Design c) provides the user with sensitivity information through tic marks (the short lines above the slider bars). They indicate how far to the left or right a slider needs to be dragged to retrieve the neighboring optimal route. Dynamic exploration, which means that the search result is immediately updated upon a changed user input (e.g., a moved slider), has been identified as a very effective design feature in geographic information and exploration systems [25]. A recent study shows that tic marks in combination with dynamic exploration are effective in helping the user to identify the optimal route from a set of pre-computed routes [17]. Check boxes to specify eliminatory constraints (e.g., exclusion of unpaved streets) are helpful in defining the optimal route as well [26].

![Diagram of user interface designs: (a) slider bars for all 21 criteria, (b) slider bars for five components without sensitivity tic marks, (c) slider bars with sensitivity tic marks, (d) radio buttons for five components.](image)

FIGURE 6 User interface designs: Slider bars for all 21 criteria (a), slider bars for five components without (b) and with sensitivity tics marks as used in dynamic exploration (c), and radio buttons for five components (d).
Most route planners consider only one cost criterion at a time, which allows utilization of an SP algorithm based on a criterion selected from radio buttons or a pull-down menu. For the multi-criteria shortest path problem, a rate of substitution needs to be estimated a priori between attributes, for example through direct questioning [27], or through stated preference surveys [8]. The latter study found, for example, that the provision of secure parking at the destination is equivalent to a reduction of 26.5-minute cycling in mixed traffic. With default trade-off rates and criterion weights set by a user, a single best route can be found as a weighted linear combination (WLC). It is, however, hard for the user to understand what a weighting value means, as the importance of a criterion will depend on the range of criterion values, e.g., the difference between minimum and maximum number of turns. Results of PCA cannot be utilized with SP algorithms because of negative component loadings, and negative component weights are not applicable as well.

A possible way to resolve these problems is to use a genetic algorithm that, based on a fitness function for individual routes, adjusts the genetic search toward the Pareto frontier [28]. Presuming that the overall utility for an alternative is the sum of utilities for each attribute and that there are no interaction effects between attributes, the fitness function \( f \) for a route \( i \) can be modeled as

\[
f_i = \sum_{p=1}^{t} w_p y_{ip} \tag{1}\end{equation}

where: 
- \( w_p \) is the importance of attribute \( p \)
- \( y_{ip} \) is the desirability rating of attribute \( p \) in route \( i \)
- \( t \) is the number of attributes taken into consideration

The desirability rating \( y \) for a route attribute is computed through a single-attribute value function, such as the mid-point method [27], which maps raw data of the different attribute levels to value scores between 0 and 1. Importance weights \( w_p \) are computed as a linear combination of the \( p \)-th factor loading \( l \) in each component with component weights \( W \) as coefficients.

\[
w_p = \sum_{j=1}^{C} W_j l_{j,p} \tag{2}\end{equation}

where: 
- \( C \) is the number of components identified in PCA
- \( W_j \) is the weight assigned to a component by the user
- \( l_{j,p} \) is the \( p \)-th component loading on component \( j \)

Finally, the route with the highest \( f \) value can be retrieved from the Pareto optimal route set. Pre-computing a route set, and using range lines or tic marks on slider bars, provides the user with a better understanding of the consequences of changed criterion weights on the search result [17].

**CONCLUSIONS**

The results of this explorative study suggest that a large number of route selection criteria in combined bicycle-transit trips can be more parsimoniously described by fewer, more general criteria (components). This means that a user interface for route choice can be simplified by reducing the number of interactive weighting or selection tools while preserving a large portion of the route variability. For future work an expansion of the test network to a larger area is
planned in order to grasp a potential impact of scale on the PCA results. The inclusion of benefit criteria requires genetic search algorithms which are more costly in terms of computation time than a single SP search. An expansion of the test network would therefore also allow an assessment of the performance and usability of genetic algorithms in building a Pareto front using a fitness function on networks of more realistic sizes.

References


