Route choice sets for very high-resolution data

Working Paper

Author(s):
Rieser-Schüssler, Nadine; Balmer, Michael; Axhausen, Kay W.

Publication date:
2012

Permanent link:
https://doi.org/10.3929/ethz-a-007136382

Originally published in:
Arbeitsberichte Verkehrs- und Raumplanung 757

Route choice sets for very high-resolution data

Nadine Rieser-Schüssler
Michael Balmer
Kay W. Axhausen

Working paper
Transport and Spatial Planning **February 2012**

IVT Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Working paper

Route choice sets for very high-resolution data

Nadine Rieser-Schüssler
IVT
ETH Zurich
CH-8093 Zurich
phone: +41-44-633 30 85
fax: +41-44-633 10 57
schuessler@ivt.baug.ethz.ch

Michael Balmer
IVT
ETH Zurich
CH-8093 Zurich
phone: +41-44-633 27 80
fax: +41-44-633 10 57
balmer@ivt.baug.ethz.ch

Kay W. Axhausen
IVT
ETH Zurich
CH-8093 Zurich
phone: +41-44-633 39 43
fax: +41-44-633 10 57
axhausen@ivt.baug.ethz.ch

February 2012

Abstract

With the increasing use of GPS in transport surveys, analysts can choose from numerous new ways to model travel behaviour – but also face several new challenges. For instance, information about chosen routes is now available with a high level of spatial and temporal accuracy. However, advanced postprocessing is necessary to make this information usable for route choice modelling. Out of many related issues, this paper focusses on generation of choice sets for car trips extracted from GPS data. The aim is to generate choice sets for about 36,000 car trips made by 2,434 persons living in and around Zurich, Switzerland, on the Swiss Navteq network, a very high-resolution network. This network resolution is essential for an accurate identification of chosen routes. However, it substantially increases the requirements for the choice set generation algorithm in regard to performance as well as choice set composition.

This paper presents a new route set generation based on shortest path search with link elimination. The proposed procedure combines a *Breadth First Search* with a *topologically equivalent network reduction* and ensures a high diversity between the routes, as well as computational feasibility for large-scale problems like the one described above. To demonstrate the usability of the algorithm, its performance and the resulting route sets are compared to those of a stochastic choice set generation algorithm.

Keywords

GPS data, choice set generation, high-resolution transport networks, performance, route choice

Preferred citation style

Rieser-Schüssler, N., M. Balmer and K.W. Axhausen (2012) Route choice sets for very high-resolution data, *Working paper*, , Institute for Transport Planning and Systems (IVT), ETH Zurich, Zurich.

1 Introduction and Related Work

With the increasing use of GPS in transport surveys, analysts can benefit from a more detailed observation of peoples' actual travel behaviour and numerous new ways to model this behaviour. At the same, time micro-simulations of travel behaviour gain importance and research and practice and necessitate exactly this more detailed representation of people's behaviour. For instance, information about chosen routes is now available with a level of spatial and temporal accuracy never seen before. However, several issues have to be resolved before new route choice models based on GPS observations can help to improve micro-simulations. Out of many related issues, this paper focusses on generation of choice sets for car trips extracted from GPS data.

The aim is to generate choice sets for about 36,000 car trips made by 2,434 persons living in and around Zurich, Switzerland, on the Swiss Navteq network, a high-resolution network for navigation systems covering all regions of Switzerland. The trips originate from a study conducted by a private sector company trying to determine whether or not participants noticed certain billboards (Pasquier *et al.*, 2008). We obtained the data, but without the socio-economic details of the respondents, from one of the sponsors of the original data collection effort as part of a joint project. The participants were asked to carry an on-person GPS logger for 6.65 days on average. No additional information, such as modes, trip purposes or the use of navigation devices, was collected. The car trips where extracted using the GPS processing routines described in Schüssler (2010).

The network comprises 408,636 nodes and 882,120 unidirectional links representing the entire Swiss street network, including minor and access roads. Thus, the Navteq network contains 44-times more links than the planning network for the same area (Vrtic *et al.*, 2005), that has so far been used for traditional aggregate transport models and therefore been the benchmark for choice set generation algorithms. However, with the advent of new survey technologies and increasingly detailed transport models, e.g. agent-based micro-simulations, the low network network resolution is not sufficient anymore. Micro-simulation models often operate on a high-resolution representation of the infrastructure to be able to model also local effects of transport policies. To increase the realism of the agents' behaviour route choice models estimated based on high resolution route observations and networks are necessary. High resolution observations allow to determine the exact route a traveller has actually taken and regardless wether this route is determined by a map-matching or modelled using for example with the network-free approach by Bierlaire and Frejinger (2008) it can only be exploited if it is represented in the network and this can only be ensured by using a high-resolution network.

In general, two different approaches can be employed for choice set generation. The analyst can either model the membership of an alternative to the choice set (e.g. Swait, 2001; Morikawa, 1996). Then, following Manski (1977), the probability of observing that decision maker n

chooses alternative i from the universal choice set U depends on the probability $P(i|C_n)$ that he chooses i from choice set C_n and the probability $P(C_n|U)$ that $C_n \subset U$ is his actual choice set. Or the analyst can generate the individual choice set C_n in a step prior to the modelling. In route choice situations, the universal choice set, i.e. all possible routes between an origin and destination pair, is not known. Moreover, the true choice sets of travellers are not known because this is an information that is not easily obtained from survey participants making it difficult to establish the probability that route belongs to a choice set. Thus, the only suitable approach in this context is to use a choice set generation procedures that extracts routes from the network and try to find all routes with a non-zero probability of being chosen.

The most common route set generation approaches can be categorised in two ways: first, by focussing on the path-establishing procedure, into approaches using repeated least cost path search and approaches employing successive path development or, second, by focussing on the output, into stochastic and deterministic procedures. Representative of the stochastic successive path development is the random walk algorithm developed by Frejinger *et al.* (2009). Starting from the origin node, the next link is chosen (based on a Kumaraswamy distribution) depending on length of the link and the shortest path distance between its end node and the destination. This is repeated until the destination node is reached. Examples for deterministic successive path development were presented by Hoogendoorn-Lanser *et al.* (2006) for multi-modal connections and by Prato and Bekhor (2006) for car trips. Both apply branch & bound technique by creating a branch at every node in their respective networks and bounding these branches using several constraints.

A prevalent version of the route set generation with repeated least cost path search is the *Stochastic Choice Set Generation*. Before each least cost path search, the link costs in the network are drawn from a probability distribution, e.g. a normal distribution (Ramming, 2002; Dugge, 2006; Bliemer *et al.*, 2007) or a truncated normal distribution (Nielsen, 2000; Prato and Bekhor, 2007). In addition to the link cost, Nielsen (2000) and Bovy and Fiorenzo-Catalano (2007) also randomised the preference parameters for the generalised cost function. The procedure ends when a predefined number of draws or when the route set reaches a predefined size.

Deterministic route set generation approaches using repeated least cost path search are link elimination, link penalty and path labelling. Path labelling was introduced by Ben-Akiva *et al.* (1984). The least cost path is determined according to different cost functions, called labels. Possible labels include minimum travel time, distance, number of left turns or congestion but also maximum scenery. The maximum number of routes in the set equals the number of labels. In the link penalty approach, presented by de la Barra *et al.* (1993), the cost function remains the same. Instead, link cost on all links of the current least cost path is increased by a certain factor. Then, the new least cost path is searched. This is repeated until a predefined number

of routes are found. For link elimination, one or more links of the current least cost path are eliminated before the next least cost path is searched. The elimination follows a certain order. This order can be random, duplicating the order of appearance in the route (Azevedo *et al.*, 1993) or controlled by criteria (Prato and Bekhor, 2007). The number of links eliminated each time increases until the required number of routes is found. Some link elimination approaches ensure that the k-least cost paths are found (Lawler, 1976; van der Zijpp and Fiorenzo-Catalano, 2005), while others only accept paths within constraints such as maximum amount of overlap with other paths or a maximum detour time (van der Zijpp and Fiorenzo-Catalano, 2005).

In comparing these route set generation approaches, recent research has primarily focussed on the composition of the choice sets. Several authors (e.g. Prato and Bekhor, 2007; Bekhor *et al.*, 2006; Bliemer and Bovy, 2008) showed that the size and composition of the route set strongly influence the outcome of model estimation. Misspecifications lead to biased parameter estimates and choice probabilities. As Bliemer and Bovy (2008) showed, this is especially true when there are correlation between alternatives. Ideally, the choice set contains all relevant and no irrelevant routes.

An issue these recent studies do not address are the challenges imposed by high-resolution behavioural observations and high-resolution networks. The high level of spatial detail considerably amplifies the difficulties in finding all relevant routes and sorting out the irrelevant ones. However, omitting relevant routes in the choice sets leads to biased parameter estimates, as we could show in Schüssler and Axhausen (2009). This is caused by the much higher number of objectively available routes in a high-resolution network that deviate only slightly from each other. Yet, routes are only perceived and considered by the traveller as individual alternatives if they differ enough from other alternatives. This dilemma can either be solved by applying behaviourally advanced choice set generation procedures or by exploring a large number of routes to find all route alternatives relevant for the choice observed and, afterwards, reducing the resulting route set to the individual choice set considering attractiveness, plausibility and similarity between the routes (Bovy, 2009).

The major problem with the first option is that recently developed behaviourally advanced choice set generation procedures are basically not computable in reasonable time for the high level of spatial detail. As it is shown in the next section of this paper, they would run for weeks to generate choice sets for as few as 500 OD pairs. Thus, only the second option of generating large route sets with a simpler algorithm based on repeated least cost path search and subsequently reducing the choice set size is applicable to the problem at hand. But even for these simpler algorithms computational efficiency is a predominant issue. In addition to the large number of routes that have to be generated, the identification of each route becomes substantially more time consuming in a high-resolution network because the least cost path algorithm has to evaluate substantially more nodes and links to find the least cost path. Therefore, more research effort is

needed on the computational efficiency of such algorithms.

This paper contributes to this line of research by presenting an algorithm that is specifically designed to meet the requirements for route set generation in high-resolution networks discussed above: an acceptable computation time as well as resulting route sets in which the routes are reasonable in terms of generalised travel costs and heterogeneous enough to be perceived by the traveller as individual alternatives. Moreover, the algorithm produces a route set as exhaustive as possible which increases the chances that all relevant alternatives, i.e. all alternatives with a choice probability substantially higher than zero, are detected. The subsequent reduction to individual choice sets is beyond the scope of this paper and discussed in Schüssler and Axhausen (2009) and Schüssler (2010).

The proposed link elimination algorithm combines a Breadth First Search with topologically equivalent network reduction. It ensures a significant level of diversity between the routes as well as high computational speed while enabling the use of any given link cost function. Breadth First search trees are chosen over alternatives like Depth-First, Best-First or Multiway tree search Nievergelt and Hinrichs (1993) since they best meet the general goal of producing a route choice set of n diverse, feasible least cost routes, because (i) the tree data structure can be built on demand and does not have to be balanced out as a pre-process, (ii) in highly unbalanced trees, Breadth-First-Search treats tree node expansion order equally for tree nodes at same depth, (iii) does not need any assumptions about relevant, resp. irrelevant sub-trees for tree node expansion order, and (iv) processes tree nodes for short routes earlier than long ones.

The algorithm – as well as the specifications of the state-of-the-art algorithms it is compared to – are described in the next section. Subsequently, the usability of the algorithm is demonstrated by first comparing its computational performance and resulting route sets to those of the Stochastic Choice Set Generation. The paper closes with some conclusions and an outlook on further work.

2 Generating Route Sets

As established in the previous section, the procedures discussed in the remainder of this paper focus on generation of a route set that can afterwards be reduced to the individual choice set. The reduction approaches themselves are beyond the scope of this paper. Thus, the goal is to find the maximum number of feasible and low cost routes in a the shortest amount of computation time possible. Thereby, a *feasible* route is continuous, contains no loops and has low travel costs. The requirement of low costs stems from the assumption that travellers overall to prefer low cost routes, though there perception of what low costs are might differ. *Travel cost*, in this application, is defined as the free-flow travel time since no other (generalised) cost information is

available in the network data. The algorithms themselves, however, operate with any generalised travel cost function, e.g. travel time estimates from loaded networks, when and where available.

In order to reach this goal, four procedures were tested and are presented and discussed in different levels of detail in this section. First, our own procedure, a Breadth First Search on Link Elimination, including two performance optimisation features is presented. Second, three other state-of-the-art choice set generation procedures, a Stochastic Choice Set Generation, a Branch & Bound approach and a random walk are briefly discussed and it is argued why - apart from the approach presented in this paper - only the Stochastic Choice Set Generation could be used for the study at hand.

2.1 Route set generation with Breadth First Search on Link Elimination (basic BFS-LE algorithm)

The new Breadth First Search on Link Elimination (BFS-LE) calculates repeated least cost paths of a given origin-destination (OD) pair for a given network, represented as a strongly connected, weighted, directed graph $G(V, E)$. The vertices of the graph $G(V, E)$ are geo-coded in an Euclidean space with coordinates (x_v, y_v), while an edge $e \in E$ defines its cost as its weight (negative utility). The cost function itself can take any form and depends solely on the available network information. It does neither impair the functionality nor the computation time of the algorithm.

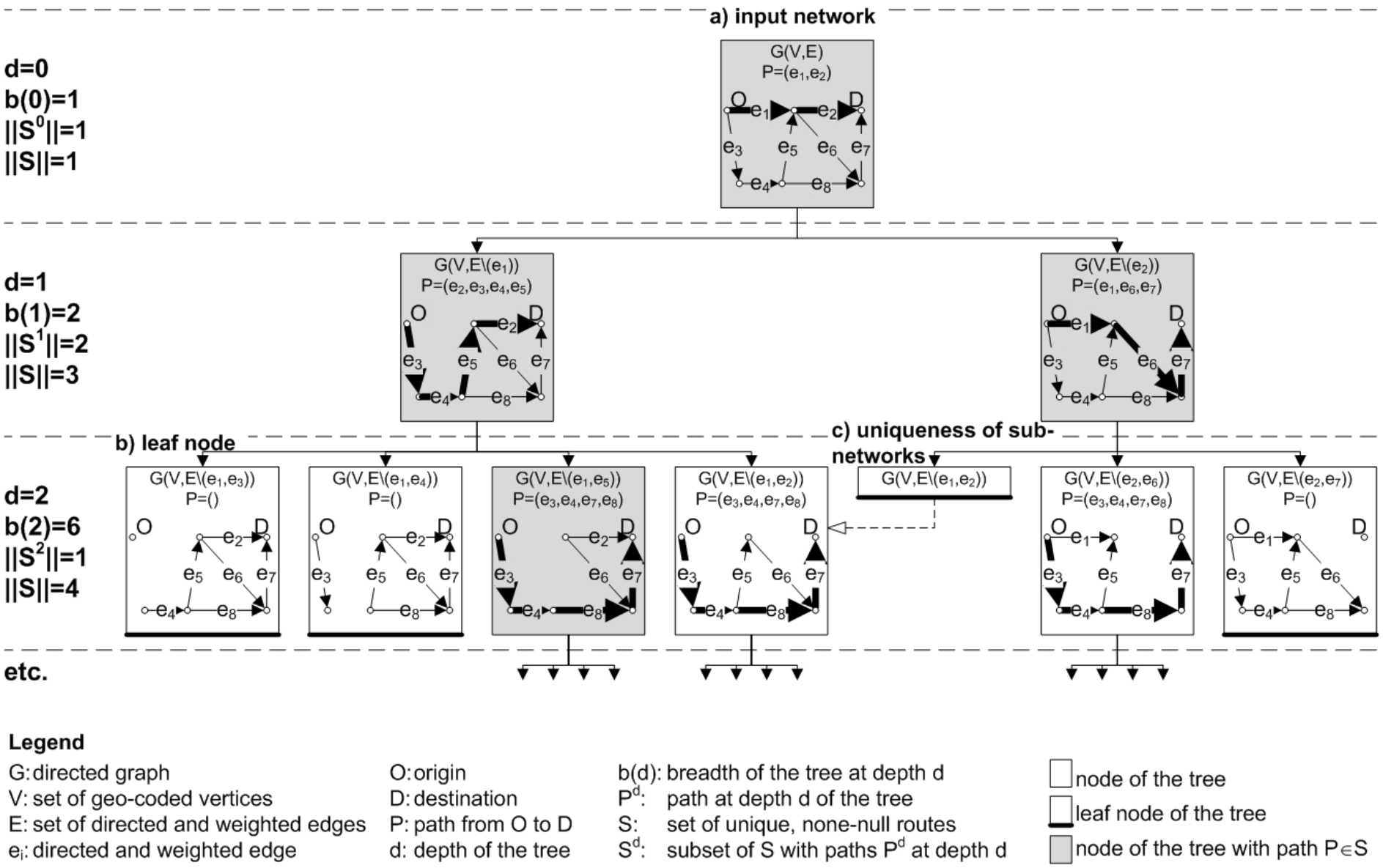
The least cost paths are calculated with the so-called *A-Star Landmarks* routing algorithm presented in Lefebvre and Balmer (2007). Its computational performance is at least one order of magnitude better than the simple Dijkstra. The SCSG method cannot take advantage of the A-Star Landmarks router's performance because it requires a preprocessing to estimate remaining travel costs. Since the SCSG changes link costs, this preprocessing would have to be re-performed after each link cost variation step.

For constructing the BFS-LE tree, some definitions have to be given first: The input network $G(V, E)$ defines the *root* of the tree and is denoted as $G^0 = G(V, E^0)$ (see Figure 1a). $G^d = G(V, E^d)$ is a sub-network of G^0 at depth d of the tree, while d also indicates the number of edges removed from root network G^0. The least cost path from v_O to v_D ($v_O, v_D \in V; v_O \neq v_D$) in a network G^d is called $P^d(v_O, v_D)$ and is defined by the set of p edges $e_i^d \in P^d; i = [1..p]$. If no path exists, P^d is empty. Therefore, each *tree node* of the BFS-LE tree is defined by one sub-network G^d and its least cost path $P^d(v_O, v_D)$.

The construction of the BFS-LE tree is based on the following four rules:

- **Tree node expansion:** the creation of the *child* nodes of a BFS-LE tree node is done using

Figure 1: Basic BFS-LE tree: (a) is the root of the tree with the given network, (b) shows a leaf node (no path from v_O to v_D) and (c) is an example of a sub-network that is already present at depth d.



the following rule: for each edge e_i^d of path P^d a sub-network $G^{d+1} = G(V, E^{d+1}) = G(V, E^d \backslash (e_i^d))$ is constructed and the least cost path $P^{d+1}(v_O, v_D)$ based on network G^{d+1} is calculated (see Figure 1 for an example).

- **Uniqueness of sub-networks:** a child node of a node at depth d will be created only if the sub-network G^{d+1} is not already created at another node of the tree at depth $d+1$, since that node and its sub-tree do not produce new sub-networks, resp. new paths (Figure 1c).
- **Leaf definition:** a tree node is a *leaf* of the BFS-LE tree if the path P^d of graph G^d is empty (Figure 1b).
- **Breadth first:** child nodes (nodes at depth $d+1$) will only be constructed if all parent nodes (nodes at depth d) are already created.

Last, but not least, the BFS-LE tree creates one least cost path per tree node, and at each depth d at most $b(d)$ routes are calculated. Since the routes S have to be *unique, none-null routes*, not all calculated routes in the BFS-LE tree can be part of the set. Therefore, a route P^d of a sub-network G^d of the BFS-LE tree is added to the route set S only if the route is not empty (see Figure 1b) and it does not already exist in S (see Figure 1 where the routes assigned to S are marked with a grey background). Furthermore, since the addition of a path P^d to the route set S is dependent on the paths already assigned before, and these paths are dependent on the order of parsing through the edges e_i^d of a path P^d to create the child tree nodes, it is necessary to complete the whole tree at depth d before assigning the new paths to the set S. If the set S^d keeping the disjoint paths P^d at depth d with $P^d \cap S = \emptyset$ contains more paths than necessary for a route set with size n then a subset of S^d is assigned to S such that $\|S\| = n$. Since the outcome of the route set generation should not depend on the processing order of the links at depth d this subset is created by randomly drawing routes from S^d. If S^d contains less routes than necessary for a route set with size n the whole set S^d is added to S.

The complexity of the BFS-LE tree can be estimated via the breadth of the tree b at depth d, called $b(d)$. Since the number of child nodes of a node containing graph G^d is equal to the number of edges $e_i^d \in P^d$ while respecting *Uniqueness of sub-networks*,

$$b(d+1) \leq \sum_{j=1..b(d)} \|P_j^d\| \;\; ; \;\; \texttt{with} \;\; b(0) = 1 \,. \tag{1}$$

Usually, in real, high-resolution networks a least cost path of an OD pair can easily contain many dozens of edges, which let $b(d)$ grow very fast for increasing depth d. Uniqueness of sub-networks helps to decrease that complexity, but does not typically prevent the production of a wide BFS-LE tree.

The BFS-LE route set generation method presented here produces n different paths from v_O to v_D if n paths exist. Otherwise, it will return the set of all possible paths from v_O to v_D. The algorithm guarantees the shortest paths P^0 to be part of S. Even more, each set S^d at depth d

contains the $(d+1)$*th shortest path* of the input network G. Therefore, assuming $S^0, S^1, .., S^d$ are completely added to the resulting set S, then it contains at least the first $(d+1)$ shortest paths of network G.

While the algorithm performs well for a Manhattan-network it is also necessary to estimate *pathological cases for BFS-LE method.* For BFS-LE, the worst case happens if at each depth d only the $(d+1)$th-shortest path is found. To create a route set S of size n for such a situation the BFS-LE algorithm needs to expand the tree until depth $n-1$. This pathological case happens only if all of the n-shortest paths of an OD pair are disjoint.

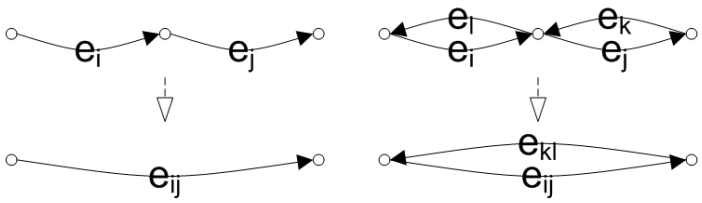
2.1.1 Performance optimisation 1: BFS-LE(PO1)

As mentioned above, it is necessary to create the whole set S^d before adding it, or a randomly drawn subset, to S. But this produces a fair amount of computational overhead for the final depth d. For example, the BFS-LE tree at depth d can already be very wide ($b(d) \gg 1$), but only one additional path of S^d needs to be added to S to reach route set size n. Furthermore, while the generation of G^d via link elimination is not costly at all, the calculation of the shortest path – even with the *Landmarks A-Star* routing algorithm – is the most time-consuming part of the BFS-LE method. A simple, but efficient, way to reduce that computational overhead is to randomly *shuffle* the order for which tree node at depth d the shortest path is calculated. With that, it is not necessary to produce the whole breadth of the tree at depth d when the route set size is already reached and the resulting choice sets do not systematically differ from those generated with the non-optimised algorithm.

Performance gain is located only at depth d where route set size n is reached. For all previous depths, performance stays the same because all routes of the previous depths have to remain in the choice set. This is due to the above stated goal of the route set generation to produce feasible least cost paths and the design of the algorithm. The lower the tree-level the costlier are, in general, the generated routes. Thus, omitting routes in higher levels to include routes in lower levels would lead to higher cost routes, contradicting the objectives of the procedure.

Regarding the relationship between the possible performance gain and the choice set size the following holds. If n is large, then the depth of the tree is large too, and the performance gain of that optimisation is negligible. But the typical route set to generate on a high-resolution transport network lies between 20 and 100 routes per OD pair, which typically ends up with a depth d of the BFS-LE tree between two to four. In such cases, the BFS-LE(PO1) method can efficiently reduce computational time.

Figure 2: Examples of merging edges for producing a topologically equivalent network



2.1.2 Performance optimisation 2: BFS-LE(P02)

The second optimisation of the BFS-LE stems from the problem that even a cleaned network contains many nodes that do not model junctions, intersections or dead-ends. These nodes usually model changes in network characteristics such as free flow speed or number of lanes. They are crucial for a correct routing but can be neglected in the link elimination stages leading to an increase in the computational performance. Thus, before the link elimination stage, the input network $G(V, E)$ is reduced to a *topologically equivalent network* $G' = G(V', E')$.

To create G', so-called "pass" vertices that do not model junctions, intersections or dead-ends, are removed from G and their incident edges are combined per direction. Figure 2 illustrates the procedure to generate G'. Therefore, $V' \subseteq V$ and $E' = E^r \cup E^m$. V' keeps the remaining vertices, called "non-pass" vertices or "non-pass" nodes in the subsequent analysis. E^r contains the set of untouched edges of E. E^m defines the set of merged edges $e_{ij..k} = merge(e_i, e_j, .., e_k); e_i, e_j, .., e_k \in E$. By assigning G' as the root of the BFS tree, the complexity of the tree is markedly reduced compared to the basic BFS-LE method, since some nodes of the tree at depth d are treated as one tree node and therefore $b(d)$ is reduced (i.e. edges e_3 and e_4 of G in Figure 1 will be combined to edge e_{34} and the first two tree nodes of depth $d = 2$ are treated only once which reduces $b(2)$ to 5). In order to ensured that the performance optimisation P02 does not change resulting route set, the least cost path calculation still needs to be executed on the networks G^d due to three reasons:

1. v_O and/or v_D can be part of the removed "pass" vertices and therefore are not part of V'.
2. The costs of merged edges can differ.
3. The edges of the generated routes must be part of G^d.

In contrast to the BFS-LE(PO1), BFS-LE(PO2) can decreases computing time at each depth of the BFS tree. The performance gain strongly depends on (i) the number of edges that can be merged in G, (ii) the OD pair itself and (iii) the number of calculated paths of the BFS tree that contain vertices $v \notin V'$.

2.2 Alternative choice set generation procedures

In addition to the proposed BFS-LE algorithm three other state-of-the-art algorithms have been implemented and tested in the course of this study:

- a Stochastic Choice Set Generation (SCSG)
- a Branch & Bound algorithm proposed by Prato and Bekhor (2006), and
- random walk introduced by Frejinger *et al.* (2009).

In this subsection, these algorithms are introduced and briefly discussed before a short computation time comparison reveals why only the SCSG algorithm was used for further comparison with the BFS-LE in the remainder of this paper. Other algorithms, such as the labelling approach, were found inappropriate for the problem at hand right from the beginning due to the very limited number of network attributes available.

2.2.1 Stochastic Choice Set Generation (SCSG)

The Stochastic Choice Set Generation (SCSG) is a stochastic repeated shortest path search based algorithms. Since only few link attributes were available a simple implementation was chosen in which the network is changed by randomly drawing the cost, i.e. free-flow travel times, of each network link from a probability distribution. The shortest path search itself is carried out using the Dijkstra's algorithm (Dijkstra, 1959). Normal distributions – truncated at zero cost with the mean at the initial link cost and employing different multiples of the initial link cost as standard deviations – were tested. However, insufficient variation in the resulting route costs was created. Thus, the same routes were found over and over again, making it extremely time-consuming to generate route sets of sufficient size. Dugge (2006) reported similar problems for routes consisting of many small links. She reasoned that with an increasing number of links and stable travel cost, the standard deviation of a route decreases. Therefore, and because the aim was to generate as many heterogeneous routes as possible, a uniform distribution was used, ranging from zero to twice the initial link costs. Unfortunately, a further increase in route heterogeneity by introducing randomised the preference parameters for the generalised cost function as done for example by Bovy and Fiorenzo-Catalano (2007), was not possible because there was only one cost attribute, i.e. free-flow travel time, available.

2.2.2 Branch & Bound(B&B)

The Branch & Bound algorithm was implemented because its authors could show that the resulting route sets have attractive properties. In Prato and Bekhor (2007), they compare the algorithm to other choice set generation procedures and conclude that it produces realistic and

heterogeneous routes that allow estimation of models with a higher prediction accuracy than models derived from other choice sets.

The realism of the resulting choice sets stems from the constraints imposed on the branches of the route tree. These constraints impose directional, temporal, loop, similarity and left turn restrictions to limit the number of branches. However, since the algorithm processes each tree level completely before moving on to the next next level, its computation time increases exponentially with the number of links in the paths (Prato, 2009). Thus, in our experiments, the Branch & Bound algorithm terminated in reasonable computation time only for OD pairs connected by very short paths, i.e. paths with less than 30 links. This is not sufficient considering that our chosen routes contain an average number of 65.69 links.

2.2.3 Constrained Random Walk(CRW)

Frejinger *et al.* (2009) hypothesise that the true choice set is the universal choice set. Thus, they propose to use a path set generation that corresponds to an importance sampling approach, because it allows to obtain unbiased parameters in the model estimation by correcting each alternative with regard to its sampling probability. An instance for such an approach is their constrained random walk. The constraint biases the random walk towards the shortest path. This is done by using a double bounded Kumaraswamy distribution whose cumulative distribution function depends on the costs of the already established path and an cost estimate for the remaining path. The extent to which the random walk is biased towards the shortest path depends on the parameters chosen for the Kumaraswamy distribution.

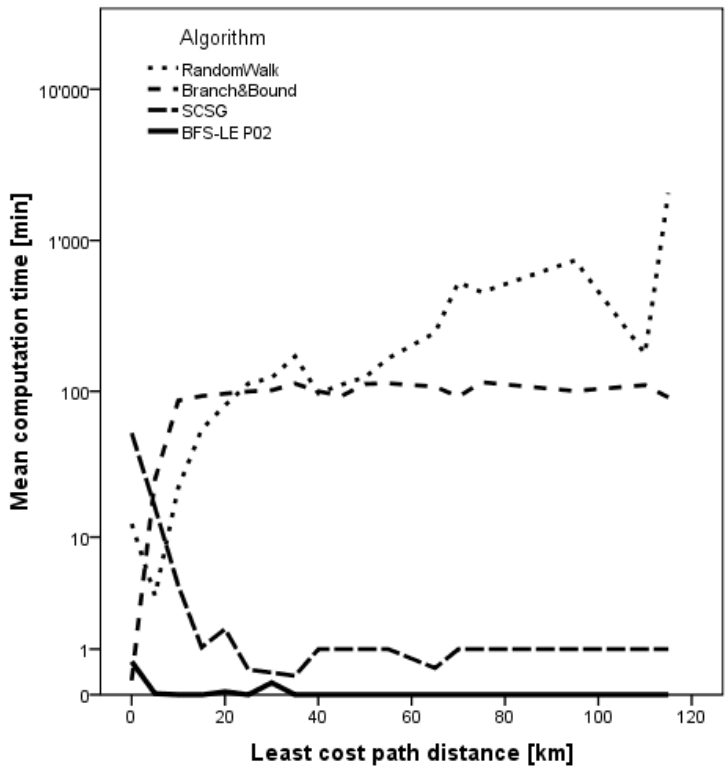
However, finding the appropriate parameter setting for the problem at hand proved to be difficult. Several parameter settings were tested but none were suitable for the entire data set with its high variation in route lengths. If the parameters were too strict, only the shortest path was found. If they were too weak, the algorithm wandered around and needed a long time to terminate i.e. find the end node of the route. The resulting route is unreasonably long, i.e. more then ten times longer than the shortest path.

2.2.4 Computational feasibility

Unfortunately, the issues discussed impact the computation time of the B&B and the CRW so strongly that they become inapplicable for the problem at hand. As can be seen in Figure 3, both approaches led to impractically long computation times, even for a small number of alternatives and short routes.

In Figure 3, the run times of all four algorithms are presented on a logarithmic scale for a target

Figure 3: Average computation time of the algorithms for 20 alternatives and 90 minutes time abort threshold



choice set size of 20 and a sample of 500 OD pairs that are representative of the 36,000 main study OD pairs in terms of distance, main road type of the shortest path and network density at origin and destination. A *time abort threshold* was introduced to capture OD pairs for which the choice set generation could not be completed within a time interval predefined by the analyst. If the computation time for any OD pair exceeds this time abort threshold, the routes generated until then are stored as the choice set for the OD pair and the choice set generation moves on to the next OD pair. In the runs shown in Figure 3, this time abort threshold is set to 90 minutes per OD pair. Since the time criterion is only checked after a route has been completed, an additional abort criterion was imposed on the CRW to prevent it from exploring the network for hours trying to finish just one route and to avoid completely unrealistic routes. If the number of links in the random walk path exceeds ten times the number of links of the shortest path for the given OD pair, the random walk was stopped and restarted. If by then the time abort threshold was violated, the choice set generation for this OD pair was stopped.

Considering the computation time of the B&B depicted in Figure 3, it becomes clear that applying this algorithm is not so much a question of computational performance but feasibility. In our experiments the B&B failed to determine any route at all in 229 out of 500 OD pairs within 90 minutes per OD pair and in 161 of the remaining 271 OD pairs it determined just one

route. Allowing for longer search times did not appear reasonable since the computation of this experiment alone took 17 days. The CRW was at least able to determine more than 5 routes for 466 OD pairs. However, doing this took - even with the additional constraint - 12 days in this experiment while determining the choice sets with the SCSG and the BFS-LE only took 7 days and little more than 2 hours, respectively.

Thus, of the four algorithms tested in this study, only the SCSG and the BFS-LE were appropriate to choice set generation in the very high-resolution network at hand. Consequently, only these two algorithms are more closely evaluated in the remainder of this paper.

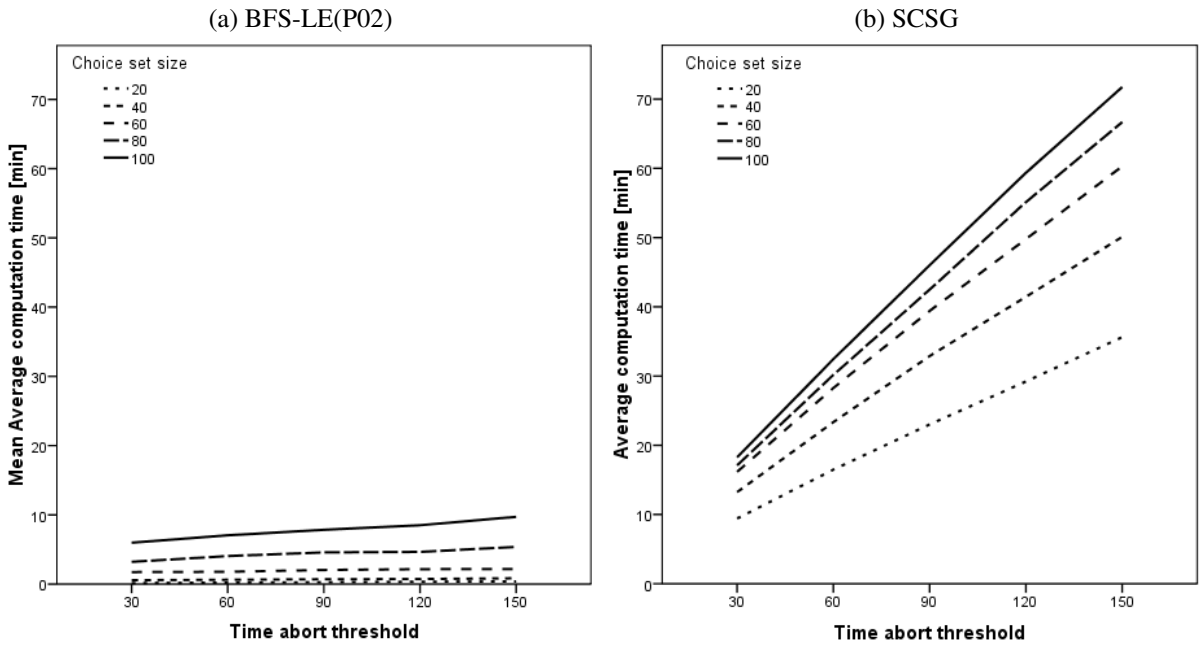
3 Computational Performance

All tests described in this section were run on systems having two Dual-Core AMD Opteron Processors 2222 running at 3 GHz. The 4 GB allocated memory was connected through a front side bus clocked at 1,000 MHz. As the code was not multi-threaded, only one of the CPU cores was actually used by the tests. The performance analysis was conducted on a sub-sample of 500 OD pairs. The OD pairs are representative of the 36,000 main study OD pairs in terms of distance, main road type of the shortest path and network density at origin and destination. This was achieved by using a stratified random sampling approach to draw the OD pairs with the strata defined by the criteria mentioned above. The main road type used by the shortest path was motorway for 9.1%, extra-urban road for 11.6%, urban main road for 62.2% and local road for 7.3% of the OD pairs. The main road type for the remaining 9.8% could not be decided because they used several road types in similar proportions. The shortest path distance distributions for each main road type showed reasonable patterns. The median of the shortest path distance was 18.93 km for motorway, 8.71 km for extra-urban road, 3.75 km for urban main road and 1.42 km for local road as the main road type. Median distance of trips that could not be categorised was 8.18 km. Concerning network density, a threshold of 15 nodes within a radius of 200 metres was chosen to distinguish between high and low density areas at the start and end points. The share for each of the resulting four density classes is about 25%. This share, however, varies with the main road types. Extra-urban trips for instance tend to start and/or end in lower density areas whereas a higher share of trips using main urban main roads start and/or end in dense areas.

For performance evaluation of the BFS-LE algorithm, the BFS-LE(P02) version with both performance optimisation options (topologically equivalent network reduction and shuffling of the sub-network list at depth d) was employed in the performance analysis; whereas in the SCSG, a uniform distribution ranging from zero to twice the initial link costs was used. The computational performance was evaluated according to the following criteria:

- Route set sizes

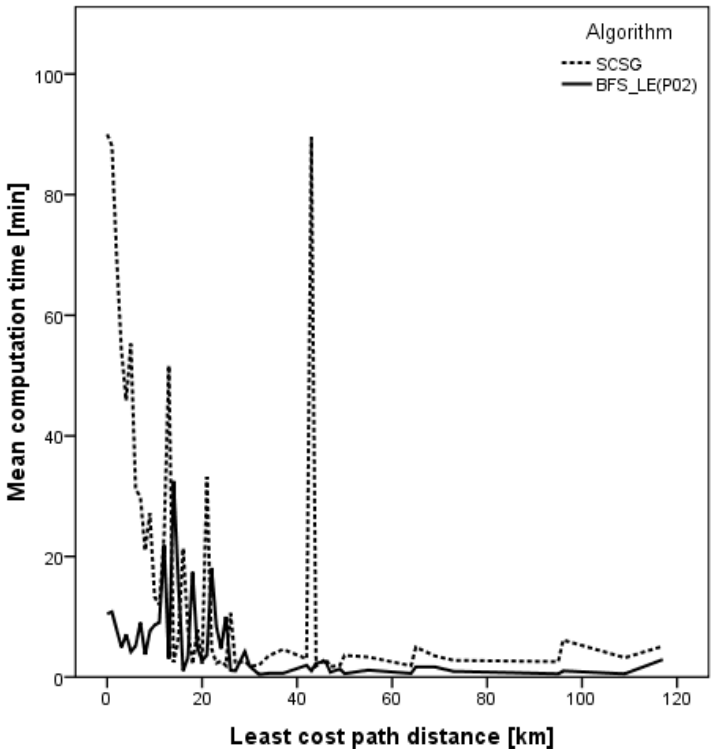
Figure 4: Average computation time depending on route set size and time abort threshold



- Time abort thresholds
- Least cost path distance
- Number of non-pass nodes on the least cost path

Figure 4 shows the computation time depending on the route set size and the time abort threshold. As expected, the computation time rises with increasing route set size and time abort threshold. The impact of the route set size is stronger than that of the time abort threshold. Most prominent, however, is the impact of the choice set algorithm itself on the computation time. The average computation time per OD pair with the BFS-LE algorithm does not exceed 10 minutes even for 100 alternatives and a time abort threshold of 150 minutes, whereas the average computation time with the SCSG algorithm is already 9.5 minutes for 20 alternatives and a time abort threshold of 30 minutes. Thus, the computation time of the SCSG is on average 32 times higher than those of the BFS-LE(P02) for several reasons. First, the SCSG reaches the time abort threshold much more often than the BFS-LE(P02). Second, the average time to adapt the network and determine the new shortest path is 0.45 seconds for the BFS-LE(P02) and 0.79 seconds for the SCSG. Third, the number of routes calculated to derive a route set size of 100 with a time abort threshold of 90 minutes adds up to 1039 on average for the BFS-LE(P02) and 3525 for the SCSG. However, it can also be seen in Figure 4 that the increase in computation time over route set size grows for the BFS-LE(P02) while it lessens for the SCSG. This originates again from the frequency the time abort threshold was reached. For the BFS-LE(P02), this number is very low in the beginning and rises strongly with increasing route set size. For the SCSG, this rise

Figure 5: Computation time depending on the least cost path distance



slows down for route set sizes of 80 and 100. Only a few additional OD pairs, that have not reached the abort threshold for smaller route set sizes, reach the time abort threshold for larger route set sizes.

As discussed in the previous section, the performance of both algorithms depends to a large degree on the network structure. Thus, the subsequent figures evaluate the computational performance with regard to characteristics of the OD pairs for the runs with a route set size of 100 and a time abort threshold of 90 minutes.

Figure 5 presents the computation time relative to the distance of the least cost path between each OD pair is shown. The BFS-LE(P02) clearly outperforms the SCSG for short trips, especially in the band under 10 km, covering 75 % of the car trips in this sample as can be seen in Figure 6. The BFS-LE(P02) reveals unsteady behaviour only for OD pairs that are 10-30 km apart. In Switzerland, this distance band typically contains trips between neighbouring agglomerations and has the highest probability of containing pathological cases as described in the previous section. For trips longer than 40 km the BFS-LE(P02) outperforms the SCSG again. However, these values have to be treated with care because they represent only few observations. This applies especially to the peak at 43 km which represents a single OD pair for which the SCSG algorithm could not find 100 routes within 90 minutes.

Figure 6: Distribution of the least cost path distances

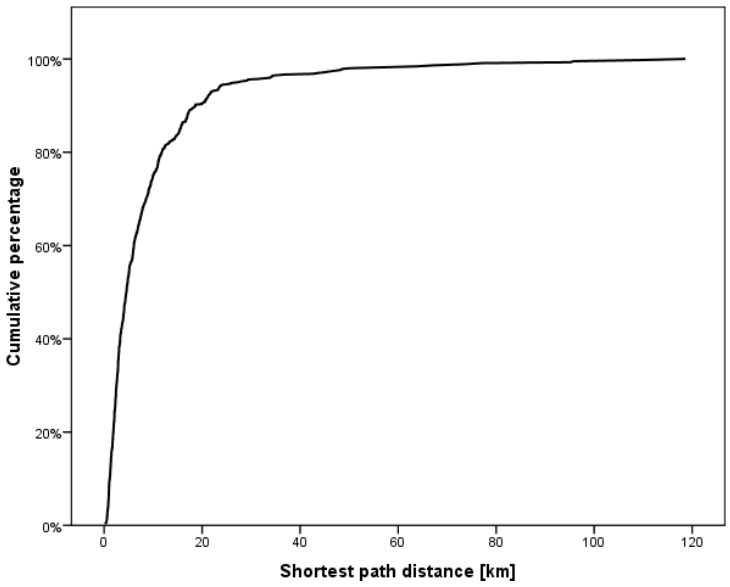
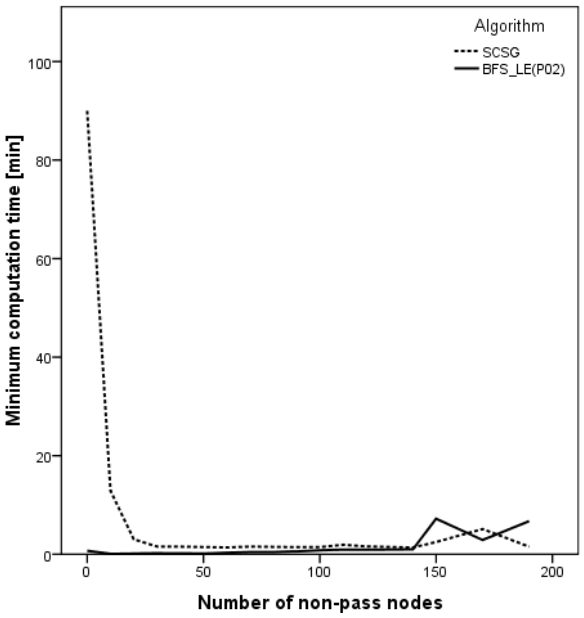
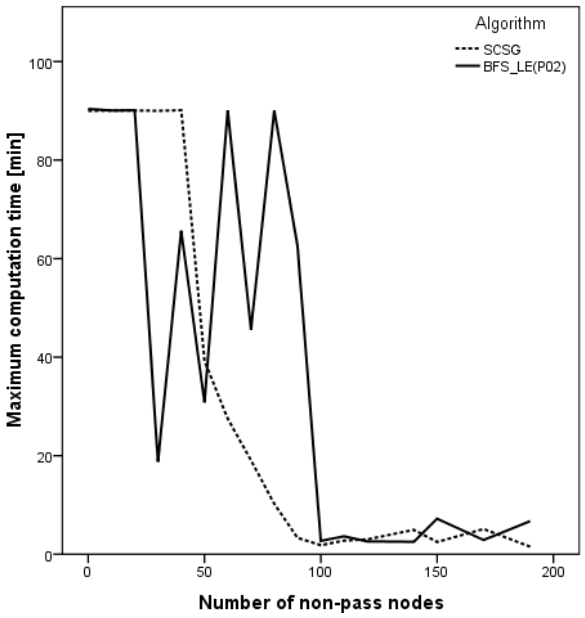


Figure 7: Computation time depending on the number of non-pass nodes on the least cost path

(a) Minimum



(b) Maximum



A closer look at the underlying networks structure was needed to explain the unsteady behaviour of the BFS-LE(P02) for medium-distance trips. This is done in Figure 7, depicting minimum and maximum computation time per OD pair depending on the number of non-pass nodes of the least cost path. The graphs reveal that, in principal, the computation time rises with an increasing number of non-pass nodes. This explains the good performance for short-distance trips. These

trips take place in an urban or suburban environment where not much can and is gained by the topologically equivalent network reduction. The medium and long trips were more affected by the reduction, but each to a different extent, so that the number of non-pass nodes is only weakly correlated with distance. For those trips, the minimum computation time rises, but at the same time the probability of a pathological case, or at least reaching the time abort threshold, decreases. This leads to the unsteady performance behaviour for medium distance trips. Since this depends entirely on the network structure, there is, unfortunately, no algorithmic way to improve the performance other than the time abort threshold used in this study.

4 Structure of the Derived Route Sets

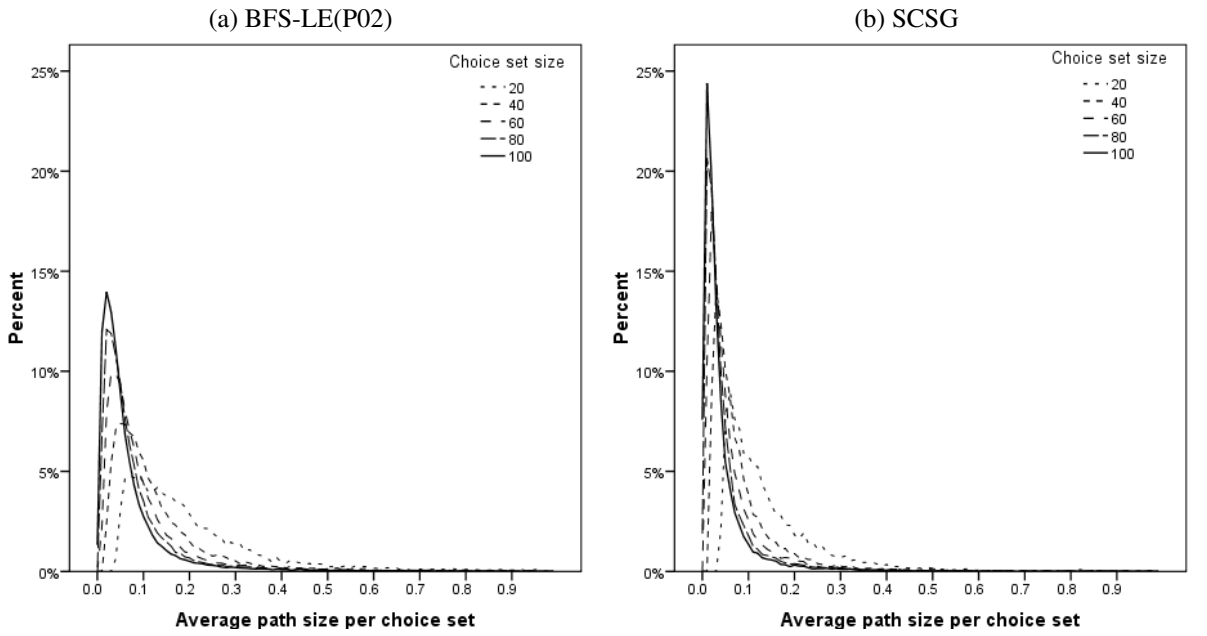
A thorough evaluation of route set generation approaches does not consider only their computational performance but also the structure and quality of resulting route sets. The guideline for this analysis were the following questions suggested by Bovy (2009):

1. Is the size of the route set sufficient?
2. How often/well is the chosen route reproduced?
3. How diverse are the routes?
4. How plausible is the hierarchical sequence?

In principal, the analyst defines for both algorithms what route set size is sufficient, and the algorithms search new routes until this route set size is reached or no further routes exist. Exceptions are OD pairs reaching the time abortion threshold. For the current settings, this occurs only for 6 % (at most) of the OD pairs with the BFS-LE(P02), but for up to 50 % of the OD pairs with the SCSG. The analyst has to decide how to treat the affected OD pairs in subsequent applications.

The reproduction of the chosen route was measured in two ways. First, the number of times the complete chosen route had been reproduced was counted. The BFS-LE(P02) achieved this for 63 % of the OD pairs with a choice set size of 20 and for 73 % of the OD pairs with a choice set size of 100. The respective figures of the SCSG were 64 % and 75 %. These figures are better than those reported by Ramming (2002) and Prato and Bekhor (2007). With their link elimination algorithms, the chosen route was reproduced for 60 % of the OD pairs by Ramming (2002) and for 58 % of the OD pairs by Prato and Bekhor (2007). For their stochastic choice set generation algorithms the respective figures were 38-50 % and 49-61 %. It has to be noted, though, that their choice set generation set-ups were different. Ramming (2002) performed only 48 shortest path searches for each OD pair and choice set algorithm and Prato and Bekhor (2007) even less, namely 10.

Figure 8: Path size of routes resulting choice sets



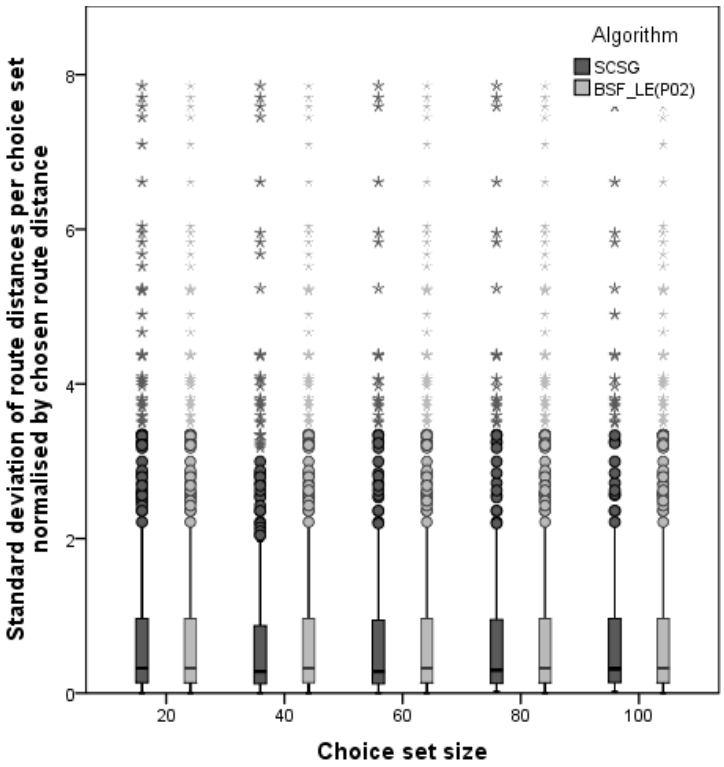
In the second step, the choice sets not containing the chosen route were examined more closely. For them, the overlap between the route that best reproduces the chosen route and the chosen route itself was calculated as percentage of the chosen route length. The resulting mean coverage of the chosen route varies between 76 % and 82 % for the BFS-LE(P02) and between 76 % and 78 % for the SCSG. Considering network resolution and number of possible paths between an OD pair, both algorithms are adequately able to reproduce the chosen route.

To investigate the diversity of the routes set, the path size PS_{in} for each route was determined using the well-known formulation of Ben-Akiva and Bierlaire (1999):

$$PS_{in} = \sum_{a \in \Gamma_i} \left(\frac{l_a}{L_i} \right) \frac{1}{\sum_{j \in C_n} \delta_{aj} \frac{L^*_{C_n}}{L_j}} \tag{2}$$

where Γ_i is the set of all links of path i, l_a the length of link a, L_i the length of path i and $L^*_{C_n}$ the length of the shortest path in C_n using link a. δ_{aj} equals one if link a is on path i and zero otherwise. The path size ranges between 0 and 1, with 0 indicating complete overlap and 1 no overlap at all. Figure 8 depicts the distribution of average path size of all routes in each choice set. As expected, the overlap between the routes increases for both algorithms with choice set size. For all choice set sizes, however, the path size distribution indicates considerably more diversity between the routes generated with the BFS-LE(P02) than those generated with the SCSG. This finding holds even after filtering routes that are unreasonable long (in terms of distance or travel time) or overlap too much with each other as discussed in Schüssler and Axhausen (2009).

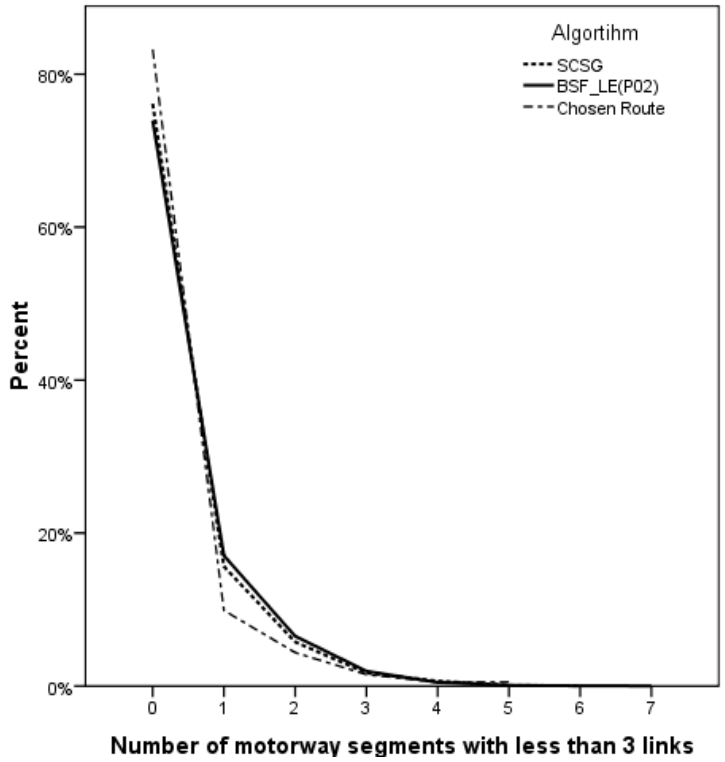
Figure 9: Standard deviation of route distances per choice set normalised by chosen route distance



Another indicator for the diversity of the produced route sets is the distribution of route distances. For each route set the standard deviation of route distances has been calculated and normalised by the distance of the chosen route. Figure 9 shows the distribution of these normalised standard deviations by choice set size and generation algorithm. It can be seen that the distributions are very similar across choice set sizes and generation algorithms. This means that both algorithms produce similar distance distributions and that the distance distribution of routes found later in the choice set generation is similar to that of the routes found earlier on.

The most ambiguous criterion for evaluation of the choice set structure is the plausibility of the hierarchical sequence. Since this study is concerned with the generation of choice sets for car route choice, this analysis focussed on the shares and sequences of road types. Four road-type categories were defined: motorways, extra-urban main roads, urban main roads and local roads. The average share for each road type was calculated as a percentage of the total route length. The road type shares in the chosen routes were then compared to the average road type shares in the BFS-LE(P02) and the SCSG choice sets. The analysis revealed, that road type shares were similar for the choice sets of both algorithms and fairly stable over the different choice set sizes. The chosen routes were travelled on average 12 % on motorways, 16 % on extra-urban roads, 36 % on urban-main roads and 36 % on local roads. The respective figures for the BFS-LE(P02) choice sets are 16 %, 18 %, 35 % and 32 % and for the SCSG choice sets 17 %, 18 %, 34 % and

Figure 10: Number of motorway segments with less than 3 links



31 %. This indicates that the routes generated by both algorithms reflect actual choice behaviour.

Bovy (2009)'s main concern about the plausibility of hierarchical sequences was the unrealistic shifting between different hierarchical levels. Given the structure of the Swiss road networks, repeated switching between extra-urban main roads and urban main roads is inevitable for off-motorway trips between cities. Thus, unrealistic shifting between different road types is most likely to occur with motorways. Figure 10 shows the number of times a routes contains a motorway segment consisting of 3 or less links as percentage of the total number of routes containing at least one motorway segment. Both algorithms perform almost equally well with nearly 80 % of the trips using the motorway containing no segment consisting of 3 links or less, and over 90 % containing at most one such motorway segment. The chosen routes, however, contain even fewer cases with repeated switching between the motorway and other road types. This implies that, in reality, repeated switching between motorways and other road types is rare but it does sometimes occur. Thus, this type of behaviour cannot be excluded a-priori as unrealistic.

5 Conclusion and Outlook

The generation of choice sets in very high-resolution networks is a new challenge for transportation modellers. The number of routes in the universal choice set increases substantially while the size of the individual choice set, i.e. the number of relevant routes, probably remains the same. Thus, a large route set has to be extracted from the network, and afterwards reduced, to increase the chances that all relevant routes are included in the individual choice set. However, extracting routes from a high-resolution network is cumbersome and extremely time-consuming. Therefore, computational performance becomes a very important criterion in the evaluation of choice set generation algorithms again. Many of the advanced choice set generation algorithms presented recently are simply not applicable because they would run for weeks in order to generate choice sets for a few hundred OD pairs.

This paper compares different choice set algorithms and demonstrates that only approaches based on repeated shortest path search are applicable to high-resolution networks. A new algorithm is presented employing Breadth First Search on Link Elimination and includes two performance optimisation features: a randomisation of the processing order within each tree depth and a topologically equivalent network reduction. The computational performance of this BFS-LE(P02) algorithm and the quality of the resulting choice sets is compared to the performance and results of a Stochastic Choice Set Generation (SCSG) algorithm.

In terms of computational efficiency, the BFS-LE(P02) clearly outperforms the SCSG. Particularly for typical urban trips under 10 km, the SCSG struggled to find enough routes to meet the required choice set size, while the BFS-LE(P02) works most efficiently in this setting. Considering reproduction of the chosen route and road type composition, both algorithms perform almost equally well. The routes of the BFS-LE(P02) choice sets are, however, more diverse than those of the SCSG choice sets. Overall, the BFS-LE(P02) is clearly advantageous and can be recommended for generating choice sets in high-resolution networks.

Another advantage of the BFS-LE(P02) is that it can, like the SCSG, use any cost function specified by the analyst without changing the algorithm structure or computational performance. The only requirement is appropriate network information. One way to derive a more reasonable cost function would be to employ loaded dynamic travel time networks instead of free-flow times. This can be implemented straightforwardly because the whole algorithm, including the A-Star Landmarks router, is designed accordingly. Demonstrating this is a future research topic requiring additional data about network loads: for example, from a micro-simulation. Other cost functions could, for instance, be derived from the comparison between the attributes of the minimum time path and the chosen path or the preferences for different road types. Choice sets resulting from different cost functions could then be compared to the results of the SCSG using the same cost functions and in case of multiple-component cost functions also randomised

preference parameters for the different cost components.

The increasing use of GPS in transport surveys makes the issue of choice set generation in high-resolution networks more pressing. But it also opens up new ways to derive choice sets, namely the generation of individual choice sets from repeated GPS observations. This would allow the analyst to approximate the actual individual choice sets and allow valuable insights into the actual decision-process. Though a first attempt in this direction has been undertaken by Rich *et al.* (2007), more research is necessary before this approach can become state-of-the-art in route choice modelling.

6 Acknowledgements

We would like to thank Michel Bierlaire and Stephane Hess for providing helpful comments on earlier versions of this paper, the Swiss National Science Foundation for the funding of this work and Stefan Muff (at that time Endoxon AG) for providing the data. The source code is available under GPL at `http://matsim.org`.

References

Azevedo, J. A., M. E. O. Santos Costa, J. J. E. R. Silvestre Madeira and E. Q. Vieira Martins (1993) An algorithm for the ranking of shortest paths, *European Journal of Operational Research*, **69** (1) 97–106.

Bekhor, S., M. E. Ben-Akiva and M. S. Ramming (2006) Evaluation of choice set generation algorithms for route choice models, *Annals of Operations Research*, **144** (1) 235–247.

Ben-Akiva, M. E., M. J. Bergman, A. J. Daly and R. Ramaswamy (1984) Modelling inter-urban route choice behaviour, in J. Volmuller and R. Hamerslag (eds.) *Proceedings of the Ninth International Symposium on Transportation and Traffic Theory*, chap. 15, 299–330, VNU Science Press, Utrecht.

Ben-Akiva, M. E. and M. Bierlaire (1999) Discrete choice methods and their applications to short-term travel decisions, in R. Hall (ed.) *Handbook of Transportation Science*, chap. 2, 5–34, Kluwer, Dordrecht.

Bierlaire, M. and E. Frejinger (2008) Route choice modeling with network-free data, *Transportation Research Part C: Emerging Technologies*, **16** (2) 187–198.

Bliemer, M. C. J. and P. H. L. Bovy (2008) Impact of route choice set on route choice probabilities, *Transportation Research Record*, **2076**, 10–19.

Bliemer, M. C. J., P. H. L. Bovy and H. Li (2007) Some properties and implications of stochastically generated route choice sets, paper presented at the *6th Triennial Symposium on Transportation Analysis (TRISTAN)*, Phuket Island, June 2007.

Bovy, P. H. L. (2009) On modelling route choice sets in transportation networks: A synthesis, *Transport Reviews*, **29** (1) 43–68.

Bovy, P. H. L. and S. Fiorenzo-Catalano (2007) Stochastic route choice set generation: Behavioral and probabilistic foundations, *Transportmetrica*, **3** (3) 173–189.

de la Barra, T., B. Peréz and J. Añez (1993) Multidimensional path search and assignment, paper presented at the *21st Planning and Transport, Research and Computation (PTRC) Summer Meeting*, Manchester, September 1993.

Dijkstra, E. W. (1959) A note on two problems in connexion with graphs, *Numerische Mathematik*, **1**, 269–271.

Dugge, B. (2006) Ein simultanes Erzeugungs-, Verteilungs-, Aufteilungs- und Routenwahlmodell, Ph.D. Thesis, Technical University Dresden, Dresden.

Frejinger, E., M. Bierlaire and M. E. Ben-Akiva (2009) Sampling of alternatives for route choice modeling, *Transportation Research Part B: Methodological*, **43** (10) 984–994.

Hoogendoorn-Lanser, S., R. van Nes and P. H. L. Bovy (2006) A rule-based approach to multimodal choice set generation, paper presented at the *11th International Conference on Travel Behaviour Research (IATBR)*, Kyoto, August 2006.

Lawler, E. L. (1976) *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart & Winston, New York.

Lefebvre, N. and M. Balmer (2007) Fast shortest path computation in time-dependent traffic networks, paper presented at the *7th Swiss Transport Research Conference*, Ascona, September 2007.

Manski, C. F. (1977) The structure of random utility models, *Theory and Decision*, **8** (3) 229–254.

Morikawa, T. (1996) A hybrid probabilistic choice set model with compensatory and noncompensatory choice rules, in D. A. Hensher, J. King and T. Oum (eds.) *World Transport Research: Proceedings of the 7th World Conference on Transport Research*, vol. 1, 317–325, Pergamon, Oxford.

Nielsen, O. A. (2000) A stochastic transit assignment model considering differences in passengers utility functions, *Transportation Research Part B: Methodological*, **34** (5) 377–402.

Nievergelt, J. and K. H. Hinrichs (1993) *Algorithms and Data Structures: With Applications to Graphics and Geometry*, Prentice Hall, Upper Saddle River.

Pasquier, M., U. Hofman, F. H. Mende, M. May, D. Hecker and C. Körner (2008) Modelling and prospects of the audience measurement for outdoor advertising based on data collection using GPS devices (electronic passive measurement system), paper presented at the *8th International Conference on Survey Methods in Transport*, Annecy, May 2008.

Prato, C. G. (2009) Route choice modeling: Past, present and future research directions, *Journal of Choice Modelling*, **2** (1) 65–100.

Prato, C. G. and S. Bekhor (2006) Applying branch-and-bound technique to route choice set generation, *Transportation Research Record*, **1985**, 19–28.

Prato, C. G. and S. Bekhor (2007) Modeling route choice behavior: How relevant is the composition of choice set?, *Transportation Research Record*, **2003**, 64–73.

Ramming, M. S. (2002) Network knowledge and route choice, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge.

Rich, J., S. L. Mabit and O. A. Nielsen (2007) Route choice model for Copenhagen: A data-driven choice set generation approach based on GPS data, paper presented at the *6th Triennial Symposium on Transportation Analysis (TRISTAN)*, Phuket Island, June 2007.

Schüssler, N. (2010) Accounting for similarities between alternatives in discrete choice models based on high-resolution observations of transport behaviour, Ph.D. Thesis, ETH Zurich, Zurich.

Schüssler, N. and K. W. Axhausen (2009) Accounting for route overlap in urban and sub-urban route choice decisions derived from GPS observations, paper presented at the *12th International Conference on Travel Behaviour Research (IATBR)*, Jaipur, December 2009.

Swait, J. (2001) Choice set generation within the generalized extreme value family of discrete choice models, *Transportation Research Part B: Methodological*, **35** (7) 643–666.

van der Zijpp, N. J. and S. Fiorenzo-Catalano (2005) Path enumeration by finding the constrained k-shortest paths, *Transportation Research Part B: Methodological*, **39** (6) 545–563.

Vrtic, M., P. Fröhlich, N. Schüssler, S. Dasen, S. Erne, B. Singer, K. W. Axhausen and D. Lohse (2005) Erzeugung neuer Quell-/Zielmatrizen im Personenverkehr, *Research Report*, Swiss Federal Department for Environment, Transport, Energy and Communication, Swiss Federal Office for Spatial Development, Swiss Federal Roads Authority and Swiss Federal Office of Transport, IVT, ETH Zurich, Emch und Berger, Institute for Transportation Planning and Traffic, Technical University Dresden, Zurich.