Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2018) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

World Conference on Transport Research - WCTR 2019 Mumbai 26-31 May 2019

Using observed route complexity in GPS traces to improve bicycle route choice set generation

Thomas Koch[1,2], Niels Wardenier, Dr. ir. Luk Knapen[2,3], Dr. Elenna Dugundji[1,2]

[1] Centrum Wiskunde en Informatica, Science Park 123, 1098XG Amsterdam.
[2] Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081HV Amsterdam.
[3] Universiteit Hasselt, Agoralaan building D, 3590 Diepenbeek, Belgium

Abstract

Everyday route choices made by bicyclists are known to be more difficult to explain than vehicle routes, yet prediction of these choices is essential for guiding infrastructural investment in safe cycling. In this paper we study how the concept of *route complexity* can help generate plausible choice sets in the demand modeling process. The complexity of a given path in a graph is the minimum number of shortest paths that is required to specify that path. *Complexity* is a path attribute which is considered to be important for route choice in a similar way as the number of left turns, the number of speed bumps, distance and other. The complexity was determined for a large set of observed routes and for routes in the generated choice sets for the corresponding origin-destination pairs. The respective distributions seem to significantly differ so that the choice sets do not reflect the traveler preferences. This paper proposes a technique to improve the choice set generation.

Keywords: Bicycling; Route complexity; Route choice set generation; GPS traces

1. Introduction

Route choice models play an important role in many transport applications and help to understand why people travel the way they do and to predict what they will do in the future. Route *choice set generation* is an essential part of route choice modeling in order to establish the weight of several route attributes in the decision process and to predict chosen routes in simulators.

Route choice modeling for bicyclists is a topic of increasing interest as more and more people travel by bicycle for their daily commute, leading to problems with congestion in cycling lanes and at traffic lights as well as parking problems with bicycles. This in turn leads to traffic conflicts with both vehicles and pedestrians, creating unsafe situations. Understanding more about how and why cyclists travel and where they deviate from the shortest path, helps us to propose ways to improve safe cycling infrastructure and to subsequently study the effects of the modifications.

Several attributes of a route are significant factors in the choice process: e.g. the number of left turns, the number of speed bumps, distance, slope, scenery etc. This study investigates the use of route complexity as an *additional* attribute.

The *complexity* of *given* (observed) path in a graph is the *minimum* number of shortest paths that is required to specify that path in the network. It can be interpreted as the (minimum) number of intermediate destinations that are connected by shortest sub-paths. Note that *complexity* is a graph theoretical property and is not related to geometric properties of the route. *Complexity* is a path attribute which is considered to be important for route choice.

The complexity was determined (i) for each route in a large set of routes observed by means of GPS traces and (ii) for routes in the choice sets for the origin-destination pairs corresponding to the observed routes generated by the POSDAP tool (ETH-Zurich 2012). The respective distributions seem to significantly differ. The complexity of the routes in the generated choice sets is higher than the observed one so that the choice sets do not reflect the traveler behavior.

This study proposes a technique to improve the choice set generation. The paper is organized as follows: the background briefly reviews the concept of choice set generation and various choice set generators that are described in the literature. Next, we formally define the concept of route complexity and demonstrate an algorithm to compute route complexity for a given route. The case study describes the data set of chosen bicyclist routes, the distribution for the observed complexity and the relations between route properties. Subsequently we show that the distribution for *route complexity* in generated choice sets may significantly differ from the observed routes. Finally, two methods to improve the generated choice sets are proposed.

2. Background

Choice sets play a crucial role in route choice modelling and prediction. In choice set generation, the universal set *U* contains all possible routes from the origin to the destination. Such a universal set can be infinitely large if it is allowed to include cycles (hence not only graph theoretical *paths* but also *walks*).

In *route based* choice models, finite choice sets are established. Each route in the choice set bears a collection of attributes (distance, number of junctions, scenery etc.). A discrete choice model is used to predict the traveller's choice from the attributes. Most models are based on multinomial logistic regression (MNL) and correction factors are introduced to account for correlation between overlapping routes. Model parameters and correction factors are determined using the finite choice set.

Recursive logit (RL) models described by Fosgerau, Frejinger and Karlstrom (2013) and by Mai, Fosgerau en Frejinger (2015) do not require a choice set for model *estimation*. Conceptually, they are equivalent to MNL models for route choice from an infinite number of alternatives. The model allows computation of the ratio of the probabilities of two routes due to the Independence of Irrelevant Alternatives (IIA) property. RL uses link-additive attributes as opposed to route attributes and conceptually applies an MNL at each junction in order to predict the next link.

However, in order to *apply* route choice models in stochastic travel simulators, candidate routes need to be generated and compared also after estimating an RL model. A typical choice set faced by a cyclist can include different paths with detours from the shortest path (i) to avoid dangerous situations such as busy highways, poor pavement conditions, unlighted cycle paths in the dark or unsafe neighbourhoods or (ii) because of personal preference for certain areas like a park, slope, signalized junctions or a familiar path. There are various choice set generators for the construction of a choice set.

Prato and Bekhor (2006) propose a method called Branch and Bound, which looks for paths that satisfy the boundary conditions: directional, temporal, similarity, loop and movement (avoiding left turns). For example, with the temporal constraints, a route with only be included if its travel time is not higher than the shortest time by a certain factor.

Rieser-Schüssler, Balmer en Axhausen (2013) introduced a shortest path method, called Breadth First Search Link Elimination (BFS-LE). The BFS-LE method first computes the least cost path from origin to destination. Then links are eliminated in a particular order and a new shortest path is found. BFS refers to the fact that a tree of networks is considered and in each network a shortest path is determined using the A* algorithm. The tree is constructed by consecutively eliminating each element from the shortest path such that each recursively generated network differs in exactly one edge from the parent network in the recursion.

Kazagli, Bierlaire en Flötteröd (2016) introduced the concept of Mental Representation Items (MRI), to construct a data set they made use of a layer system. The first layer is used to determine a MRI choice set, such as $C_1 = \{avoidCC, aroundCC, throughCC\}$ where CC stands for the city center. A layer on top of that can provide additional details. In order to make the choice set operational, an attribute is assigned to each MRI by calculating the expected maximum utility, by taking the sums of the logarithms of all utilities on the path.

The Double Stochastic Generation Function method (DSGF) described by Nielsen (2000) for public transportation and subsequently considered by Bovy and Fiorenzo-Catalano (2007) produces heterogeneous routes because both the cost and parameters used in the cost function for the links are drawn from a probability function. A possible difficulty of this method is the high computational cost, however Hood, Sall and Charlton (2011) show DSGF to be faster than the BFS-LE. Halldórsdóttir, et al. (2014) show that DSGF has a high coverage level of replicating routes taken by bicyclists and that it performs well up to 10 kilometers. Furthermore, Bovy and Fiorenzo-Catalano (2007) state that the method guarantees, with high probability, that attractive routes are in the choice set, while unattractive routes are not.

Whichever method is chosen, in order to generate realistic predictions, the distribution for each route attribute in the choice set needs to comply with the corresponding distribution found in observed sets. In the next section this requirement is investigated for the route complexity.

3. Route Complexity

The complexity of a given path in a graph is the minimum number of Basic Path Components (BPC) in the decomposition of the path where a basic path component is defined as either a least cost path or a non-least cost edge. A non-least cost edge is an edge e whose edges are connected by a path having a lower cost than the cost to traverse e. Figure 1 shows the minimum decompositions for a sample path p in a graph having complexity $c(p) = 3$. The example shows that multiple decompositions do exist for path p.

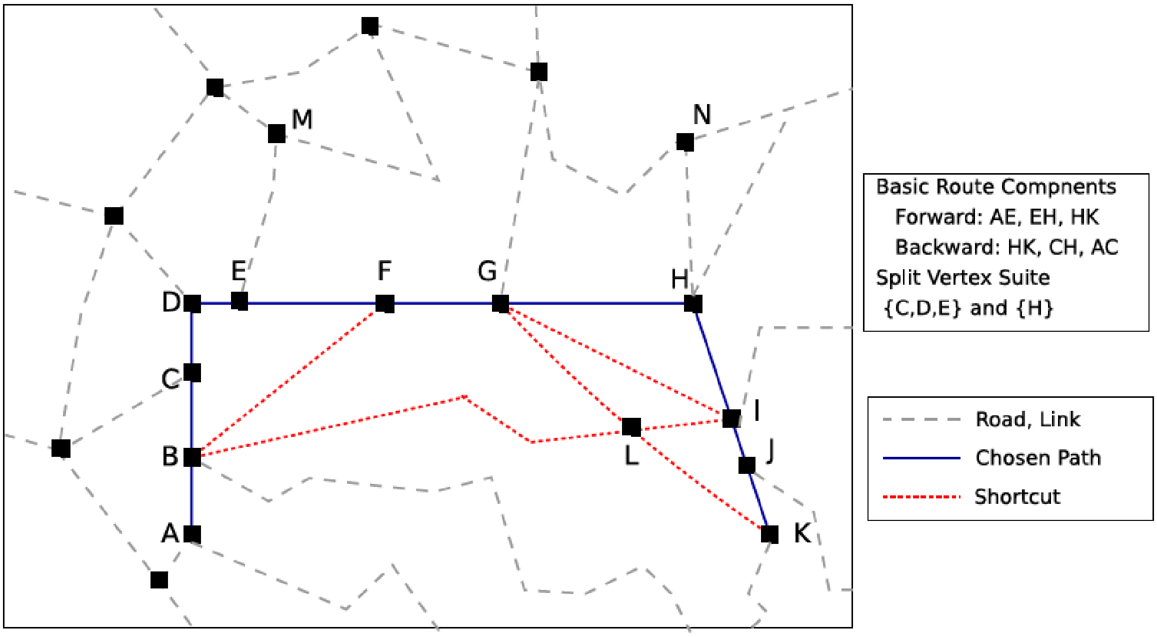


Figure 1. The blue continuous line visiting vertices A, B, C,…, I, J, K is the path followed by the traveller. Paths BF, BLI, GLI, GLK, etc represent shortcuts to the chosen path. There are two sets of split vertices: {C,D,E} and {H}. Hence there are three basic path components (BPC). Sample decompositions are ((A,C),(C,H),(H,K)) and (A,E),(E,H),(H,K)).

Knapen et al. (2016) formulate the hypothesis that in utilitarian trips, individuals tend to construct their routes as a concatenation of a small number of basic path components. Utilitarian trips have a purpose different from the fun of driving. They are driven with the intention to perform an activity at the destination location. They present Algorithm 1 to determine the complexity of a path (i.e. the minimum number of basic path components).

Algorithm 1: To determine the size of the minimum decomposition of a path into basic path components

```
Input Graph $G$, Edge costs $c$, $P = (v_0, v_1, .., v_l)$ containing no non-least-cost edges
start ← 0
k ← 1    (k is the minimum decomposition size)
while $P(v_{start}, v_l)$ is not a least cost path do
                    (find the first vertex in $v_j$ in $P(v_{start}, v_l)$ such that $lc(v_{start}, v_j) < c\left(P(v_{start}, v_j)\right)$)
    $v_j$ ← findFirstJoinVertex(P,$v_{start}$)
    k ← k + 1
    $v_{start}$ ← $v_j - 1$
return k
```

In algorithm 1 we have a graph G with positive edge costs c and a path $P = (v_0, v_1, .., v_l)$ with no non-least-cost edges. Variable start is the index of the first vertex in a basic path component. Variable k is the minimum decomposition size. In the ‘while’ loop we look for the first vertex v_j for which we can find a shorter path from v_{start} to vertex v_j; such vertices are called *join* vertices because in such vertex the given path and a shortcut *join* (see Knapen et al. 2016 for details). In a *join* vertex we increment counter k by one. The predecessor of the join vertex is used to continue. After the loop completes we can split the path at the vertex right before each *join* vertex, the vertex preceding a join vertex is called the split vertex.

Using this algorithm, a splitting is found at *k-1* vertices, splitting our path *P* into *k* basic path components. Knapen et al. (2016) proved that the decomposition is minimal but not necessarily unique. For example, by running the algorithm in reverse direction of the path we may find a different but minimal decomposition by identifying *fork* vertices.

Figure 2 from Knapen et al (2016) shows the distribution for the complexity found in several data sets for which the majority (Belgian case) or all (Italian case) trips are car trips. This supports the hypothesis that utilitarian trips are composed of a small number of basic path components. Note that 95% of all car trips had a complexity lower than 6 basic path components.

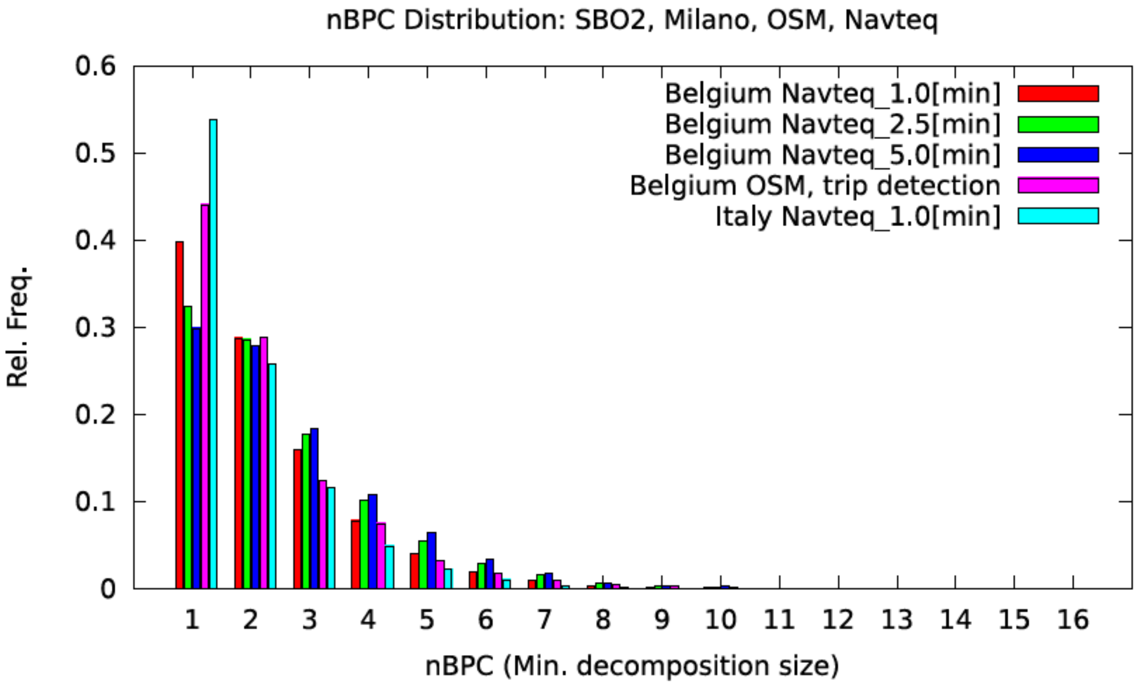


Figure 2 Relative frequency distribution for the size of the minimum decomposition of paths derived from GPS recordings. The Belgian set consists of *person* traces. It was map-matched using different networks and gap-filling thresholds. The Italian set consists of car traces only (recorded by on-board-unit (OBU).

4. Case Study

4.1. Collecting data of bicycle movements

The Dutch 2016 FietsTelWeek ('Bicycle Counting Week') data set (Bikeprint 2017) available at http://www.bikeprint.nl/fietstelweek/ contains 282,796 unique trips (although the corresponding infographic http://fietstelweek.nl/data/resultaten-fiets-telweek-bekend/ mentions 416,376 trips having a total distance of 1,786,147 kilometers).

It was collected by 29,600 cyclists who voluntarily participated in a week-long survey to track their bicycle movements using a smart-phone app in the week of 19th of September 2016. The application ran in the background to collect the bicycle movements of all participants using the phone's GPS and acceleration sensors. The cyclists involved use their bike, in a way as often seen in The Netherlands, using their bike as transportation from and to work, supermarket, school, friends, etc. For privacy reasons the resulting data was anonymized by the data provider before making it publicly available (i) by the removal of user information to make it impossible to trace multiple trips to a single person and (ii) by rounding of the trip departure time into one-hour bins to the nearest hour.

4.2. Route complexity in real-life GPS traces

The route complexity for the 282,796 collected by the Dutch FietsTelWeek2016 routes was computed and the distribution is shown in Figure 3 (blue line). For Flanders (Belgium) no detailed results for the *bike counting week* are made publicly available; hence, direct comparison is impossible. However, the distribution for the complexity of bicycle routes in the Netherlands significantly differs from the distribution for complexity found in *person traces* for Flanders shown in Figure 2.

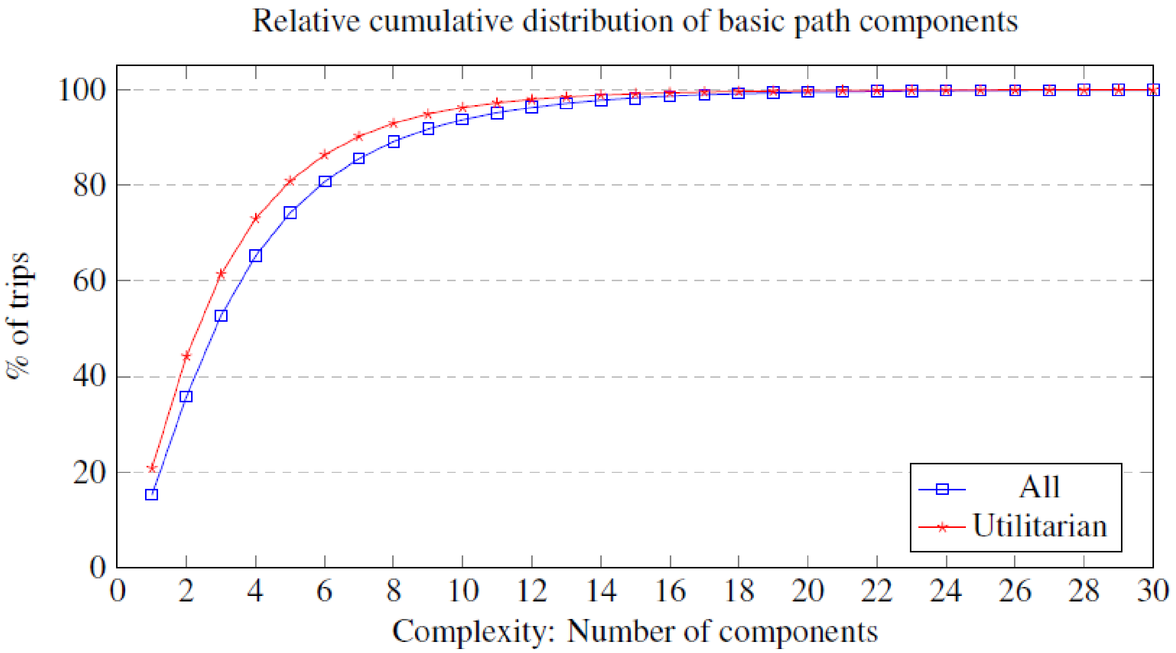


Figure 3 Cumulative distribution of the complexity of paths taken by bicyclists. Blue for unfiltered, red for only utilitarian trips with r_d⩽1.08

Car mode is the prevalent mode in Flanders according to the recurrent travel behavior survey (https://mobielvlaanderen.be/ovg/ovg52-0.php).Hence most *person traces* consist of car trips and, as a consequence, most trips in the sets investigated by Knapen et al (2016) are car trips. The difference may result from:

- Behavioral difference between car drivers and bicyclists.
- Regional behavior differences
- Parameters chosen for the map-matching process because some map-matching algorithms fill gaps by connecting positions by the shortest path.

We had no control over the map-matching process because that was performed by the *FietsTelWeek* organizer. Access to raw GPS traces is required to exclude the latter possibility.

4.3. Utilitarian vs fun trips

The trip purpose determines the route choice. Hence, route prediction for *utilitarian* trips on one hand and for *fun* trips on the other hand may result in different weight coefficients for the predictor variables. In this section we investigate the case for *route complexity* as a predictor.

Since the data set did not include any information about the purpose of each trip, the collection of trips could include non-utilitarian trips (i.e. fun trips) that may influence the distribution of the route complexity. To get more information on the non-utilitarian trips we looked at the ratio $r\_d = d\_c/d\_min$, the ratio between observed distance and shortest distance, to find a threshold $\acute{r}_d$. We assume that a trip is utilitarian if and only if $r_d < \acute{r}_d$, since trips for the purpose of fun are likely to have a longer distance than necessary.

Let $F^{-1}(r^d)$ denote the inverse of the distribution function of the variable r_d; then $\acute{r}_d = F^{-1}(1 - F^{NU})$. Pucher and Buehler (2012) define f_{NU}, the fraction of non-utilitarian trips in the Netherlands as 0.27 and thus $1 - f_{NU}$is 0.73. We assume that the 73% fraction of the trips having the smallest r_d values are utilitarian trips. This leads to $\acute{r}_d = 1.08$ for all trips observed in the FietsTelWeek data set and $\acute{r}_d = 1.10$ for the trips in Amsterdam.

Since the cumulative distributions for the complexity in the sets of all observed trips on one hand and presumed utilitarian trips on the other hand did not show a major difference as shown in Figure 3, we decided not to exclude likely utilitarian trips with a high r_d.

4.4. Correlating route complexity

It is interesting to find out whether complexity is related to network properties, to travel behaviour properties or to both. The following network related properties were investigated: the length of a path p: $len(p)$, the number of links (street segments between intersections): p, the ratio between length of the observed path and the shortest length $len\,(p)/lenShortest\,(p)$, the ratio between length of the observed path and the euclidean (straight line distance) between origin and destination:$len\,(p)/lenEuclidean\,(p)$.

The correlation coefficient (or population Pearson coefficient) for two random variables X, Y is given by:

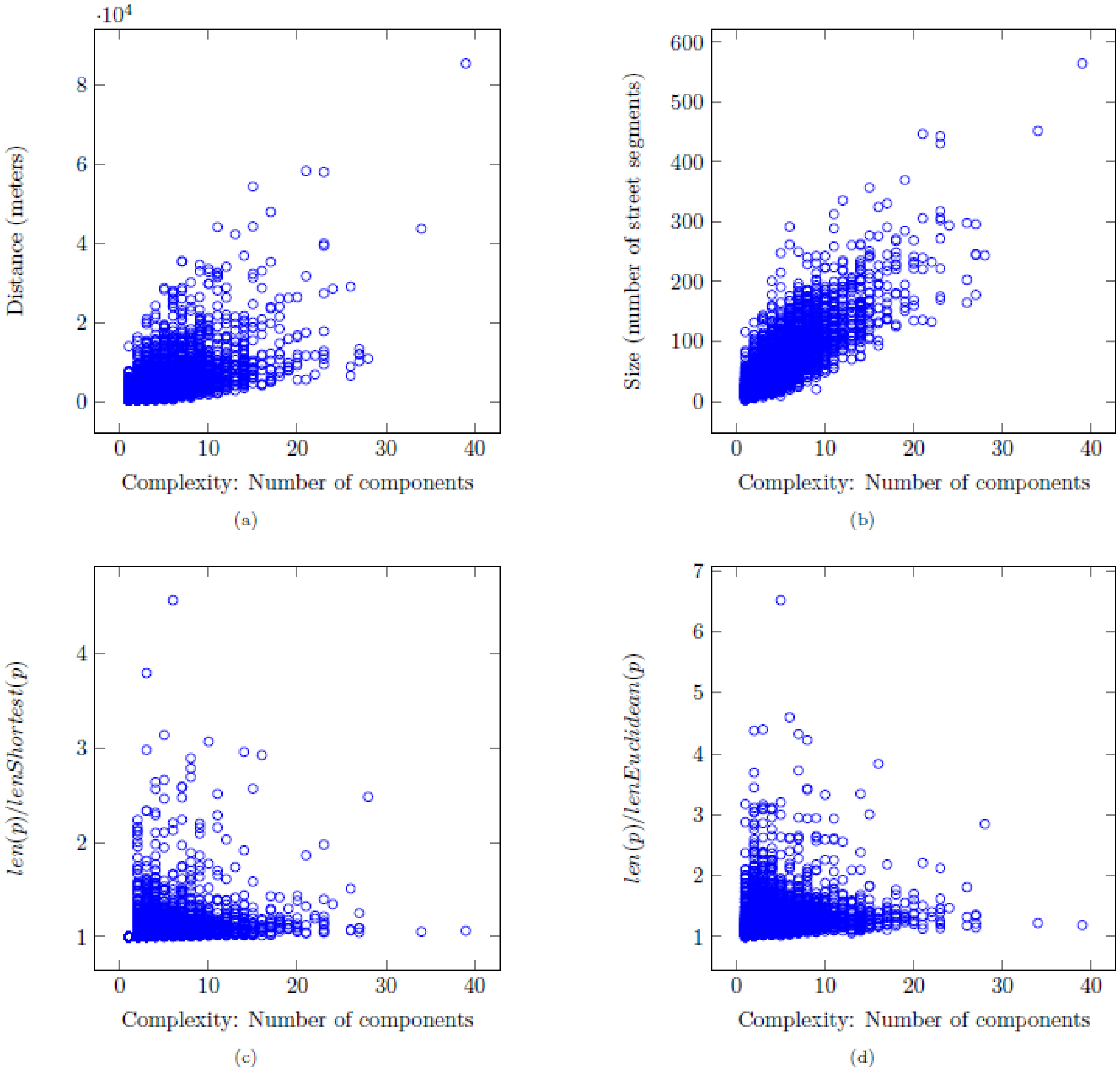
$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - EX)(Y - EY)]}{\sigma_X \sigma_Y}$$

By the Cauchy-Schwarz inequality this coefficient is bounded between -1 and +1, where +1 means two random variables are perfectly positive linearly related and -1 perfectly negative. When the coefficient approaches 0, the variables are more and less uncorrelated. Full sample tests were executed, using all collected observations in the region of Amsterdam and the results are shown in Table 1.

Table 1. Correlation between path complexity and other path properties.

Variable	**Sample correlation coefficient**	**Confidence interval**
$len(p)$	0.646	(0.639,0.651)
p	0.833	(0.830, 0.837)
$len\,(p)/lenShortest\,(p)$	0.239	(0.229, 0.250)
$len\,(p)/lenEuclidean\,(p)$	0.149	(0.138, 0.159)

The correlation with the size of a path seems to be the strongest, which makes sense as with fewer intersections, there are fewer opportunities to deviate from the shortest path. There is less correlation between the length of a path and the number of basic path components. There seems to be no correlation between the number of components and ratio actual length and (Euclidean) distance. The scatter plots in Figure 4 of the same variables against the number of components, support the story told by the correlation coefficient. Note: in this figure we only plotted a random sample of 5000 observations, this did not affect the shape of the correlation and only omitted a few outliers.



(a)

(b)

(c)

(d)

Figure 3 Scatter plot of number of components against length, size and the ratios: $len\,(p)/lenShortest\,(p)$ and$len\,(p)/Euclid\,(p)$

5. Route Complexity in Choice Set Generation

To analyze the conformance to reality of the routes generated by the route choice set generation method *Double Stochastic Generation Function (DSGF)* byHalldórsdóttir, et al. (2014) we proceeded as follows. The distribution for the path complexity was determined for the *observations* recorded in the Netherlands by the FietsTelWeek data-collection. For each observed trip, the origin and destination (OD-pair) were extracted. The DSGF implementation from POSDAP (ETH-Zurich 2012) was used to generate a route choice set for each OD-pair.

Figure 5 applies to a trip in Amsterdam that was randomly chosen from the FietsTelWeek dataset. The black line shows the route chosen by the traveler. Eight of the sixteen routes predicted by the POSDAP software are shown in different colors. Only eight predictions are shown because the overlap among routes in the choice set prevents to show them all at once.

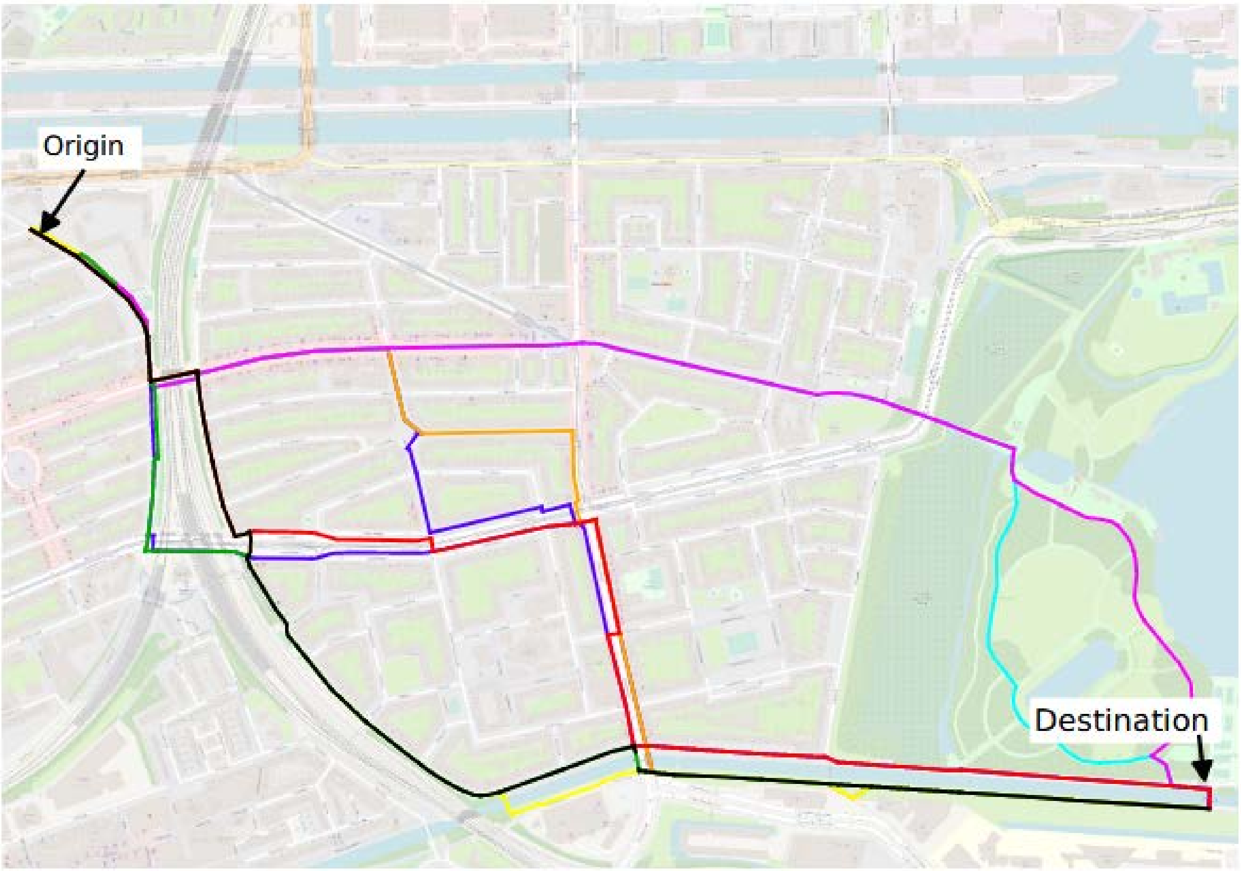


Figure 5 Chosen (black) and predicted (colored) routes for a trip in Amsterdam that was selected at random from the FietsTelWeek dataset.

The distribution of the path complexity was determined for the set of *predicted* paths (i.e. the paths in the generated choice sets). The first option we considered was to include the number of basic path components as extra attribute in the cost function used in the POSDAP DSGF algorithm, to increase the cost of predicted paths having a complexity that is improbable according to the observed distribution. This way more routes with a lower complexity would end up in the choice set. We did not pursue this option for this paper because of the high cost to adapt the POSDAP algorithm, but it still would be an interesting option for future research.

We decided to run DSFG in the same way Halldórsdóttir, et al. (2014) did and to post-process the generated choice sets. Only link length (travel distance) was used in the experiment. POSDAP allows to specify a set of link specific attribute values (like scenery, separate bike lanes etc.): this was not used due to lack of data. Thus we compute the complexity for each route in the choice set generated by POSDAP using the algorithm specified in Knapen et al (2016). After that we adapt the choice set, keeping in mind the idea that routes with a high number of basic path components are highly unlikely as observed in the recorded data.

As there is no agreement on the size N^0 of the route choice sets, we arbitrarily state that the DSGF method should produce $N^0 = 16$ routes for each origin destination pair. The POSDAP software was slightly modified in order to execute at most a given number of $M = 128$ iterations (instead of running for a given duration) so that it behaves identically on different machines. For some origin destination pairs POSDAP is not able to find as many as N^0 routes in M iterations, in which case we will use all found routes. The choice sets are written to CSV files for further processing.

5.1. Choosing attributes

To improve the set of routes generated by POSDAP, we took a look at the attributes taken into account in the generator decision process. For example Prato and Bekhor (2006) take into account the following variables:

- Directional: a link is not taken into account if it brings the bicyclist further away from the destination
- Temporal: a link is not taken into account if it takes significantly more time to travel a link in comparison with other links
- Similarity: a route is not included if it is too similar to a route already included in the choice set
- Loop constraints to avoid segments that causes too large detours
- Left turns: turning left in a route mean more interaction with other traffic and preferably avoided.

For bicycle routes, there are some specific variables that influence a cyclist's route choice, like the number of cars on a link. A cyclist might prefer a route that is longer but involves less conflicts with cars. Furthermore, if a route has a maximum speed for cars that exceeds 50 km/h (31 mph), a cyclist might prefer a route with segregated bicycle lanes, this matches research by (Halldórsdóttir, et al. 2014) who uses the variable `BikeLanes`. An additional variable could be the road surface, a cyclist might prefer smooth asphalt over cobble stones, a variable call `RoadType`. We did not fill the variable `LandUse`, but a cyclist might prefer routes in green surroundings over a concrete jungle.

5.2. Complexity distribution of predicted paths

Before adapting the choice set generated by POSDAP, we compared the complexity in the routes in Amsterdam between those observed and predicted by POSDAP.

Specifically, we compared the cumulative probability distribution functions of predicted routes:$F_{A,P}^{P}(c)$ with the function for observed data: $F^{A,O}(c)$. As plotted in Figure 6 the complexity distributions show a different shape; the predicted paths have a higher complexity overall than what we observed for cyclists in Amsterdam.

Statistically the Kolmogorov-Smirnov and χ^2test reject the possibility at an $\alpha = 0.05$ the null hypothesis that the two data sets describe the same distribution: the Kolmogorov-Smirnov rejects with a p-value of $2.2e^{-16}$ and χ^2 rejects with a$p < 0.00050$.

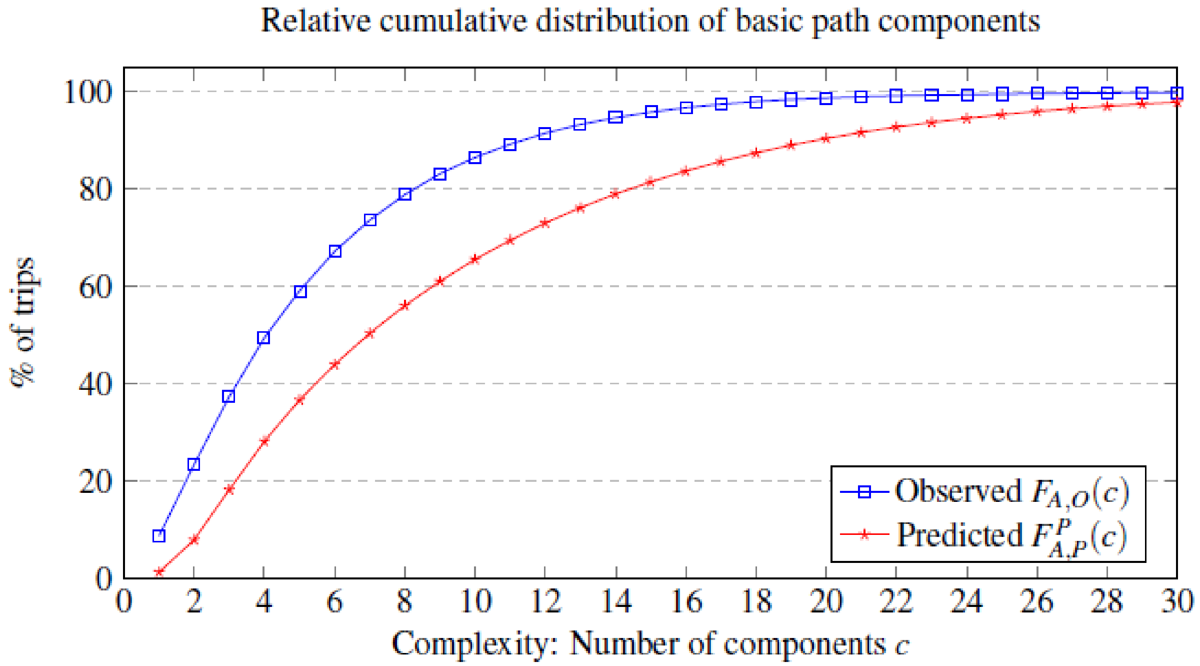


Figure 4 Cumulative distributions of number of basic path components of observed bicycling routes in the Amsterdam (blue) and the number of components in paths predicted by POSDAP's Double Stochastic Generation Function implementation

6. Choice Set Adjustment

6.1. Using a discrete sampler

We propose to sample a subset of each choice set predicted by the POSDAP software so that the probability mass function for the path complexity $f^P_{A,P}(c)$is closer to the observed complexity probability mass function $f_{A,O}(c)$. To do this we define a discrete sampler function $f_s(c)$:

$$\forall c \in N^+ : f_s(c) = \beta \cdot \frac{f_{A,O}(c)}{f^P_{A,P}(c)}$$

$$\sum_{c \in N^+} f_s = \beta \cdot \sum_{c \in N^+} \frac{f_{A,0}(c)}{f^P_{A,P}(c)} = 1$$

$$\beta = \left(\sum_{c \in N^+} \frac{f_{A,O}(c)}{f^P_{A,P}(c)} \right) - 1$$

With the second expression we can compute the constant by normalization.

After POSDAP generates a route r, we determine complexity$c(r)$. A random number $0 < k \leq 1$ is generated and route r is kept if $f_s\big(c(r)\big) \leq r(k)$. A difficulty in this approach is that for large values of$c$, those with many basic path

path components, we have very few observations. So, when we encounter such observation the method we propose assigns a too high amount of mass to it.

We tried two methods to fix this problem. The first method was to group boxes for large values of c so that each box has a mass of at least 1% attached to it. In the second method we just straight out discard all trips with a complexity c of $c \geq 41$. The second method proved to work better since it was easier to apply since we did not have to define the boxes with at least 1% of mass, it kept more routes: 21k vs 17k and finally the maximum error in comparison with $f_{A,O}(c)$was 0.3% in this method versus 0.6% in the box method.

The found sample distribution $f_s(c)$ was applied to the 494546 routes generated by POSDAP and kept 21425 routes, the used constant for $\beta$was 0.043. Initially the POSDAP software tried to fin d 16 alternatives for each origin-destination pair and after applying the sampling function it kept the following number of alternatives (Table 2).

Table 2. Distribution of number of kept alternatives..

Number of alternatives	Frequency
0	16985
1	9831
2	3711
3	1043
4	200
5	40
6	6
7	1

If the sample function is applied the maximal error between $f_{A,O}(c)$and probability mass function after sampling is 0.3%, so in that aspect the sample is working well. However in 84.2% of the cases the sampler only kept at most one route. And the goal was to make a choice set having a size larger than one (leaving some options to choose from).

6.2. Using a maximum likelihood function in the sampler

For each origin-destination pair we want at least N_1 alternatives in our choice set. To do this, define $N_0(i)$ as the initial number of found alternatives for origin-destination pair $<O_i, D_i>$ and $N_1(i)$ as the number of alternatives we want to keep for OD pair $<O_i, D_i>$. Finally $N_1(i)$ is smaller than or equal to both $N_0(i)$ and N_1.

For each $<O_i, D_i>$we collect the routes predicted by POSDAP for $<O_i, D_i>$: the set $P_{A,P}(O_i, D_i)$. We are interested in the set $P_{N_1}(O_i, D_i)$of all the subsets of $P_{A,P}(O_i, D_i)$with cardinality $N_1(i)$. The likelihood for a set $S_i^k \in P_{N_1}(O_i, D_i)$ to have been drawn from a set with the complexity distribution $f_{A,O}(c)$ is given by:

$$L(S_i^k) = \prod_{r \in S_i^k} f_{A,O}\left(c(r)\right)$$

The subset with the maximal value is kept as the choice set and denoted by S_i:

$$S_i = \underset{s \in P_{N_1}(O_i, D_i)}{argmax}\ L(s)$$

The required number of subset evaluations n_E is

$$\binom{N_0(i)}{N_1(i)}.$$

Note that $L(S_i^k) = 0$ as soon as it contains at least one route r having a non-observed complexity i.e. such that $f_{A,O}(c(r)) = 0$. This leads to a problem if $\forall s \in P_{N_1}(O_i, D_i)\colon L(s) = 0$, then a random subset $S_i$has to be selected. To fix this we define $f'_{A,O}(c(r))$ as:

$$f'_{A,O}(c(r)) = \begin{cases} f_{A,O}(c(r)) & \text{if} \quad f_{A,O}(c(r)) > 0 \\ \dfrac{1}{c(r).\,|P_{A,O}|} & otherwise \end{cases}$$

and we replace $L(S_i^k)$ by $\prod_{r \in S_i^k} f'_{A,O}(c(r))$.

We performed the sampling process outlined above on the POSDAP output with predicted paths for $N_1 = 3,4,5,6,7,8,9$ to see what the effects of choosing N_1 is. We encountered a strange effect occurring on routes that only had a single basic path component for all values of N_1. In the data we observed in Amsterdam 8.7% of the routes was the shortest route between origin and destination and thus having a complexity of one basic path component. Using this sampling method, we only found a percentage of 0.4% of routes with only a single component for all possible values of N_1. It is likely that this effect occurs because $f_{A,O}(2) > f_{A,O}(1)$, if there are routes consisting of a few basic path components it is more likely to include only the routes consisting of two components.

In general we observe that the amount of mass given to routes with an higher complexity increases, if the N_1 increases. We determined which number N_1performed best by looking at the maximal error in relative frequency for the routes with a route complexity higher than a single basic path component and exclude those where errors occurred. For $N_1 = \{6,7,8\}$ this resulted in a maximal error around 2%. For all values of N_1 we saw an overestimation for the tails, but for $N_1 = 6$there was an additional overestimation when the complexity was 2.

We concluded that the best results were achieved with a $N_1$of 7 or 8, the results were not perfect but a worthwhile compromise to generate sufficiently large choice-sets.

7. Conclusion

There are various methods to generate route choice sets. In this paper we used the Double Stochastic Generation Function, because it generates heterogeneous routes, performs well for trips up to a length of 10 kilometers and puts the more attractive routes in the choice set. The problem with route choice generation is that, the generated route can be overcomplicated and unrealistic.

This study formally defines the concept of *route complexity* and computes complexity distributions for both a set of observed routes and for routes generated by the POSDAP software. The distributions are shown to significantly differ and a technique is proposed to enhance the generated choice set w.r.t. complexity.

Trips for which the length d is much larger than the shortest distance d_{min} between their origin and destination may be considered as non-utilitarian (fun) trips. Different complexity may be expected for *utilitarian* and *fun* trips. However, because the complexity distributions for the set $S_O$of all observed trips and the set $S_F$of trips that may be classified as fun trips based on the ratio $\frac{d}{d_{min}}$ are similar, it was decided to consider all observed trips in the study.

In order to generate route choice sets we slightly adapted the code of POSDAP and compared the cumulative density function of both observed and the predicted data output from POSDAP. From this comparison we concluded that the DSGF method indeed produces too many complex routes. This is what we expected from earlier results.

In a first attempt to fix this, a sampler was built to filter routes with a high complexity out of the predicted data. By doing this the idea was to get a choice set with a complexity distribution more similar to the complexity distribution found in observed data in Amsterdam. This is something we were able to achieve but in 84.2% of the cases we ended up with choice set left that no longer had a choice between two or more options.

In a second approach, the preferred size of the final choice set was stated in advance. Subsets of the choice set produced by POSDAP and having the preferred size are considered. The one showing the largest likelihood w.r.t. complexity to have been drawn from the observed distribution is retained. This technique causes a bias towards lower complexity when predicted routes having an unobserved complexity are retained (due to lack of better ones); this is caused by the $c(r)$ in the denominator of equation of the approximate frequency. Using this method we observed that keeping 7 or 8 of the 16 predicted routes provided us with a choice set for which the complexity distribution is most similar to the one for the observed data.

From our results it might be useful for further research to filter out unlikely routes when constructing a choice set. This can be done by applying the method of basic path components.

8. Future Research

For further research it might be interesting to look at other areas and cities than just Amsterdam. Distributions for the route complexity have been determined for 16 cities and for 3 large regions (north, south, center). Ongoing research shows that the distributions differ. Figure 7 shows distributions for the bike routes complexity in the regions of Amsterdam and Gouda and Nijmegen. If there are spatial correlations that affect the distributions of the route complexity, then the effects of it may require modifications to the sampler to fit the right distribution for the surrounding region.

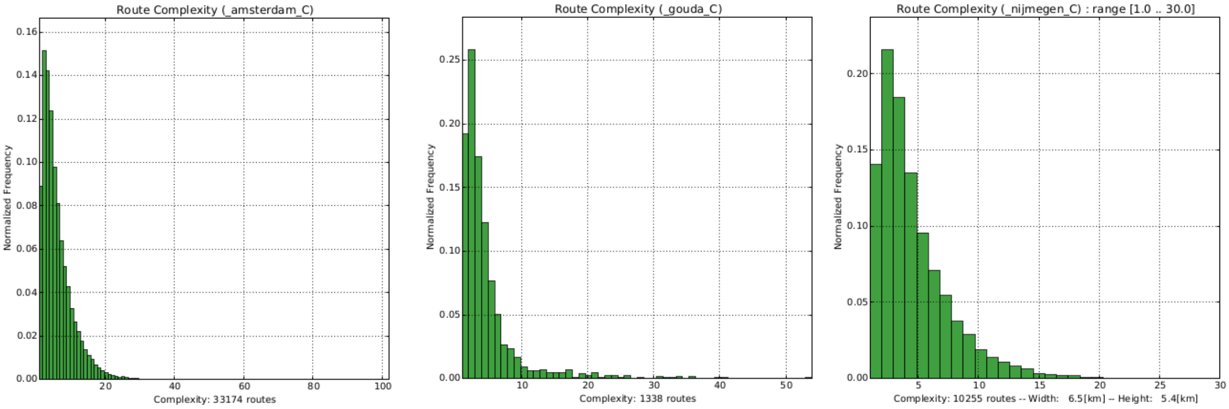


Figure 5 *showing distributions for the bike routes complexity in the regions of Amsterdam and Gouda and Nijmegen*

The POSDAP software also implements other prediction algorithms such as Breadth First Search Link Elimination (BFS-LE). Future research may look at how route complexity distributions on predicted choice sets from other algorithms differ from the routes we generated using the Double Stochastic Generation Function.

A final point of search could be the occurrence frequency of the split vertices in observed routes as determined in Knapen, Hartman and Bellemans (2017). What makes it that some vertices occur more frequently as a split vertex than others? The idea is to focus on spatial patterns in split vertex occurrence frequency by relating it to land-use properties and to road network structure.

References

Bikeprint. 2017. "Download bestanden Nationale Fietstelweek 2015 en 2016." http://www.bikeprint.nl/fietstelweek/.

Bovy, Piet H. L., and Stella Fiorenzo-Catalano. 2007. "Stochastic route choice set generation: behavioral and probabilistic foundations." Transportmetrica (Taylor & Francis) 3: 173-189.

ETH-Zurich. 2012. "Position Data Processing."

Fosgerau, Mogens, Emma Frejinger, and Anders Karlstrom. 2013. "A link based network route choice model with unrestricted choice set." Transportation Research Part B 56: 70-80. doi:10.1016/j.trb.2013.07.012.

Halldórsdóttir, Katrín, Nadine Rieser-Schüssler, Kay W. Axhausen, Otto A. Nielsen, and Carlo G. Prato. 2014. "Efficiency of choice set generation techniques for bicycle routes." European journal of transport and infrastructure research (Delft University of Technology) 14: 332-348.

Hood, Jeffrey, Elizabeth Sall, and Billy Charlton. 2011. "A GPS-based bicycle route choice model for San Francisco, California." Transportation letters (Taylor & Francis) 3: 63-75.

Kazagli, Evanthia, Michel Bierlaire, and Gunnar Flötteröd. 2016. "Revisiting the route choice problem: A modeling framework based on mental representations." Journal of choice modelling (Elsevier) 19: 1-23.

Knapen, Luk, Irith Ben-Arroyo Hartman, and Tom Bellemans. 2017. "Using path decomposition enumeration to enhance route choice models." Future Generation Computer Systems. doi:10.1016/j.future.2017.12.053.

Knapen, Luk, Irith Ben-Arroyo Hartman, Daniel Schulz, Tom Bellemans, Davy Janssens, and Geert Wets. 2016. "Determining structural route components from GPS traces." Transportation Research Part B: Methodological (Elsevier) 90: 156-171.

Mai, Tien, Mogens Fosgerau, and Emma Frejinger. 2015. "A nested recursive logit model for route choice analysis." Transportation Research Part B: Methodological 75: 100-112. doi:10.1016/j.trb.2015.03.015.

Nielsen, Otto Anker. 2000. "A stochastic transit assignment model considering differences in passengers utility functions." Transportation Research Part B: Methodological (Elsevier) 34: 377-402.

Prato, Carlo, and Shlomo Bekhor. 2006. "Applying branch-and-bound technique to route choice set generation." Transportation Research Record: Journal of the Transportation Research Board (Transportation Research Board of the National Academies) 19-28.

Pucher, John R., and Ralph Buehler. 2012. City cycling. Vol. 11. MIT Press Cambridge, MA.

Rieser-Schüssler, Nadine, Michael Balmer, and Kay W. Axhausen. 2013. "Route choice sets for very high-resolution data." Transportmetrica A: Transport Science (Taylor & Francis) 9: 825-845.