Australasian Transport Research Forum 2021 Proceedings
8-10 December, Brisbane, Australia
Publication website: http://www.atrf.info

A review on discrete route choice set generation algorithms

Raghav Malhotra[1], Chintan Advani[2], Ashish Bhaskar[3]

[1,2,3]School of Civil and Environmental Engineering, Queensland University of Technology, Brisbane, Australia

Email for correspondence: ashish.bhaskar@qut.edu.au

1. Introduction

Route choice set is an essential input in the traffic assignment process. It represents the subset of the universal set of the paths that exist between an Origin-Destination (OD) pair. Several path generation methods exist that attempt to reproduce the actual routes chosen by the individuals. The quality of these methods relies on the principle governing the algorithm and its input parameters. It is therefore essential to evaluate these algorithms based on certain criteria for practical applications. This paper aims to benchmark these algorithms by providing a qualitative and quantitative comparison among the prominent path generation algorithms in the literature. This study evaluates the algorithms based on several criteria such as: a) runtime complexity; b) heterogeneity of path set; c) flexibility of the algorithm; d) inter-algorithm comparison of the path set, and lastly; e) reproduction rate of the observed routes.

It is essential to have fundamental understanding of the choice set generation algorithms prior to actual testing. Accordingly, we provide a literature review summary in Table 1 amongst the prominent algorithms in the literature, stating their governing principle, input parameters, runtime complexities and the number of paths generated. Interested readers are referred to Prato (2009) for a detailed overview of these algorithms. The comparative evaluation provides an insight for the analyst, to filter the algorithms based on the available inputs and desired outputs. For this study, link labelling approach is avoided in the evaluation process due to unavailability of sufficient labels for the selected OD pair.

Table 1: Literature summary of the algorithms

Algorithm	Principle	Parameters	Runtime	Paths generated
Link Labelling	Define weights for the directed graph (Length, travel time, free flow time, road type etc.) and extract the shortest path.	Link weights	Dependent on shortest path algorithm used.	Single path for every unique label
Link Elimination	Iteratively removes links from the network to find new paths	Number of paths, K	O(K) where K is the number of paths, directly dependent on the shortest path algorithm used.	Pre-determined K number of paths
Link Penalty	Iteratively increases link impedance on the travelled paths to generate new paths	Penalty threshold, Number of Paths, K	Sensitive to the definition of penalty factor and number of paths, K and directly dependent on the	Pre-determined K number of paths

Algorithm	Principle	Parameters	Runtime	Paths generated
			shortest path algorithm used	
Simulation	Extracts link impedances from their individual distributions. Random draws may or may not generate different paths	Choice of Distribution, variance, mean	Sensitive to the definition of the probability distribution and variance. Directly dependent on the shortest path algorithm used.	Dependent on number of draws and distribution, may vary in different iterations using same parameters.
Branch and Bound Algorithm (B&B)	Iteratively searches the network tree by constraining the child nodes based on pre-determined thresholds	Linear distance threshold, temporal threshold, loop constraint threshold and similarity threshold	Depends linearly to the width of the connection tree but exponentially to the depth	All possible paths which are allowed by constraints.

It should be noted that the input parameters affect different aspects of the output such as the runtime and the generated choice set. Accordingly, the algorithms should ideally be evaluated with different input parameters to test its performance. However, the scope of this study is restricted to the most optimal parameters observed in the literature.

2. Evaluation of the runtime and heterogeneity of the choice sets

Heterogeneity in the path choice set is essential as travellers do not perceive highly similar paths as distinct alternatives. The heterogeneity can be defined as the uniqueness in the generated paths and can be measured using Path Size factor (PS) (Ben-Akiva & Bierlaire, 1999). In this study, we evaluate the algorithms for 20 OD pairs between a south-east (Mansfield) and the central suburb (Woolloongabba) of Brisbane city. Trajectories have been observed for a period of 11 months (January to November 2019). The shortest path observed between the two suburbs is 7.2 Km and a total of 8300 trips have been observed. These trips provide a collection of 517 individual paths which have been clustered into 25 unique representative paths and outliers have been removed. These OD pairs possess several path options, passing through the arterials and the motorway network with suitable heterogeneity in it.

The Path Size factor (PS) is given as:

$$\mathrm{PS} = \sum_{\mathrm{a}\in\Gamma_{\mathrm{i}}} \left(\frac{\mathrm{l_a}}{\mathrm{L_i}}\right) \frac{1}{\sum_{\mathrm{j}\in \mathrm{c_n}} \delta_{\mathrm{aj}}} \tag{1}$$

Where PS is the path size, Γ_i is the set of links in route i, l_a is the length of link a, L_i is the length of route i and δ_{aj} is the link-route incidence variable which equals one if link a is on route j and zero otherwise. The Path Size factor is a measure of spatial similarity of the paths. The range of path size factor for a route choice set with n paths is given by $[\frac{1}{n}, 1]$. As PS is dependent on n, we propose normalising it using min-max normalisation technique and taking an average of normalised PS values. This changes the range of PS to [0,1]. A low value of path size indicates high overlap amongst paths (homogeneity), whereas a high value indicates heterogeneity within the choice set. Table 2 provides the runtime and the PS statistics for various algorithms based on their optimal parameters.

Table 2 Route Choice set generation algorithms and the effect of parameters on the outputs

Algorithm	Parameters	Runtime per OD pair (seconds)	Path Size Factor		
			Mean	Median	Standard Dev.
Link Elimination (K = 15)	K = 15 paths	0.637s	0.335	0.3	0.137
Link Penalty (K = 15)	Ɛ = 3%	0.234s	0.374	0.385	0.113
	Ɛ = 5%	0.167s	0.40	0.393	0.098
	Ɛ = 7%	0.149s	0.387	0.37	0.07
	Ɛ = 10%	0.094s	0.42	0.452	0.123
Simulation Approach	48 draws	3.79s	0.35	0.32	0.127
Branch and bound	Time constraint: 1.2 Distance constraint: 1.1 Loop constraint: 1.2 Similarity constraint: 0.8	89s	0.293	0.268	0.092

It can be evident from Table 2 that link penalty computes paths in least time whereas branch and bound technique is the most computationally expensive for the given OD pair. Further, the mean path size is observed the highest for the link penalty and the least for the branch and bound. This indicates that the link penalty approach can generate paths with higher heterogeneity compared to the branch and bound algorithm in much lower runtime. On comparing the link penalty with link elimination approach, it can be observed that the link elimination method can produce paths with almost similar heterogeneity with a marginal increase in the runtime.

It is observed that an increase in the penalty factor for link penalty reduces the runtime, following the principle which governs the algorithm and generates higher heterogeneity. However, a penalty factor value too high could also generate some unrealistic routes as the better routes get penalised by a much higher rate. Simulation approach, on other hand, uses truncated normal distribution of travel time with lower truncation limit of 0.8*(Free Flow time) and no upper truncation limit. The paths are generated through 48 draws on this distribution and all unique paths are considered in the final choice set. The results show a mediocre set of heterogeneous paths with a relatively higher computational cost. All algorithms have been implemented on an intel i7-4790 processor with a base frequency of 3.6 GHz and 16GB RAM clocked at 2600 MHz

3. Inter-algorithm choice set evaluation

The path size factor is a representation of the overall heterogeneity of the choice set. Accordingly, two algorithms with similar path size indicates that the choice set possess similar variability among their respective paths. However, this does not necessarily certain that the algorithms possess similar choice set. Accordingly, it is necessary to evaluate the similarity among the choice sets generated by different the algorithms. This study adopts commonality factor (Cascetta et al., 1996) to identify the proportion of similar paths between algorithms. The Commonality factor (CF) is given by:

$$\mathrm{CF} = \frac{\mathrm{L_{ij}}}{\mathrm{L_i L_j}} \tag{2}$$

where L_{ij} is the length of common links between paths i and j. L_i is the length of path i and L_j is the length of path j. The value of CF lies between [0,1], where 0 indicates that the paths are completely dis-similar and 1 indicates the path being exactly the same. For a given pair of algorithms to be compared, a commonality factor matrix is generated with rows and columns corresponding to paths among a pair of algorithms. Accordingly, the commonality is evaluated for each path in a particular algorithm with the corresponding path in the other algorithm. For each path in a choice set, the maximum value across the rows and columns corresponds to the maximum capture rate of that path by the other algorithm. These maximum values for all the paths in a choice set can be used to get an average capture rate (C_r). A lower value of C_r for algorithm 1 to algorithm 2 indicates that the algorithm 1 is unable to capture the paths given by algorithm 2, while a higher value of C_r would mean that algorithm 1 is able to capture majority of the paths covered by algorithm 2. Table 3 shows the average capture rate for all pairs of algorithms within this review, where the algorithms in the columns explain the average commonality with paths for the algorithm in the row.

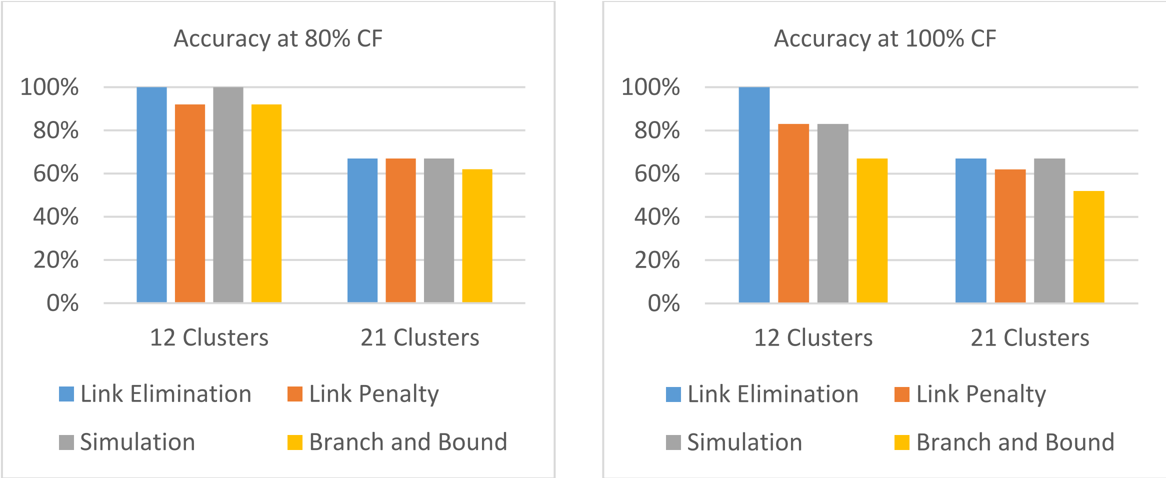
Table 3 Capture rates for all combinations of algorithms

Algorithm	Link Elimination	Link Penalty	Simulation	Branch and bound
Link Elimination	1	0.87	0.85	0.89
Link Penalty	0.79	1	0.71	0.91
Simulation	0.94	0.95	1	0.96
B&B	0.94	0.95	0.89	1

It can be observed that the link elimination can moderately capture the paths generated from another algorithms. Link Penalty, on other hand is able to capture majority of the paths given by other algorithms, while also maintaining the lowest runtime. Simulation approach is able to efficiently capture most of the paths given by other algorithms, with a higher runtime. The B&B algorithm is also able to capture the outputs of other algorithms efficiently. All algorithms have a high C_r against B&B which means the paths generated by B&B method can be generated using other methods while the runtime of branch and bound is exponentially higher, making it less desirable to generate a route choice set.

4. Reproduction rate of observed routes

One of the essential aspects to measure the quality of the choice set is by quantifying how well it is can reproduce the observed trajectories on the network. For the selected OD pairs, 25 representative paths were observed from the trajectories extracted using STATER algorithm (Advani et al., 2021) and clustered using MLTRACER (Advani, 2021). The commonality factor is used to quantify the similarity among the observed path and the paths from the algorithms. The clusters are ranked based on the total flow observed over the period of 11 months and results are presented on two levels, top 12 clusters and top 21 clusters. These both levels represent 91% and 98% of the total flow observed, respectively. The results are explained in terms of the paths observed for the threshold levels of 80% and 100% overlap. For a given pair of paths i and j, belonging to representative paths and generated paths respectively, CF_{ij} is calculated. If $CF_{ij} > \delta$, the cluster is marked as captured. Results are represented as fraction of clusters captured by the respective algorithms. Figure 1 shows the accuracy of various algorithms at the stated levels of commonality factor threshold.



(a) (b)

Figure 1: Accuracy of RCS generation algorithms at (a) 80% and (b) 100% CF

It can be stated that the all the algorithms perform quite well considering only the paths with high flow. Further, the performance of algorithms drops when less travelled paths are also considered in the evaluation. Link elimination is able to capture all the paths at both levels of the threshold while the Branch and bound algorithm shows worse results on a 100% CF threshold.

5. Conclusion

The choice sets generated by link penalty, simulation and link elimination show similar heterogeneity in the generated path set, whereas B&B generates homogeneous paths compared to these methods. On computational based comparison, link penalty and link elimination have favourable runtime followed by simulation approach. Accordingly, B&B algorithm is undesirable in terms of both heterogeneity and runtime complexity.

In terms of the flexibility of the algorithm, link elimination and link penalty provide an additional advantage of controlling the desired paths, whereas the generated paths cannot be controlled for simulation and B&B. However, this flexibility can be a dis-advantage as for the controlled algorithms, it will be essential to generate several paths to obtain a realistic choice set resulting in higher false positive (unobserved paths) than simulation and B&B algorithm. Lastly, in terms of real path reproduction rate, link elimination approach outperforms other algorithms, while simulation and link penalty moderately capture the actual paths, while B&B performed the worst to fully reproduce observed paths.

In conclusion, it was observed that the link elimination approach is the most desirable method for generating desirable paths as it possesses suitable heterogeneity with lower runtime and the ability to capture realistic path choices. This paper is ongoing research, and the extended study will analyse the effect of parameters for each algorithm on the outputs. Secondly, a valuable analysis of Branch and bound method will be another contribution in the future research as it doesn't exist in literature because of its high computational time. The later will provide a better understanding of this approach and help in avoiding conclusions based on insufficient calibration.

6. References

Advani, C., Bhaskar, A., Haque, M. M., & Cholette, M. E. (2021). STATER: Slit-Based Trajectory Reconstruction for Dense Urban Network With Overlapping Bluetooth Scanning Zones. *IEEE Transactions on Intelligent Transportation Systems*, 1-11.

Advani, C. B., A.; Md. Mazharul Haque. (2021). ML TRACER: Multi-Level Trajectory Clustering for identification of path choice set. *IEEE Transactions on Intelligent Transportation Systems*.

Ben-Akiva, M., & Bierlaire, M. (1999). Discrete choice methods and their applications to short term travel decisions. In *Handbook of transportation science* (pp. 5-33). Springer.

Cascetta, E., Nuzzolo, A., Russo, F., & Vitetta, A. (1996). A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. Transportation and Traffic Theory. Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France, 24-26 July 1996,

Prato, C. G. (2009). Route choice modeling: past, present and future research directions. *Journal of Choice Modelling*, *2*(1), 65-100. https://doi.org/https://doi.org/10.1016/S1755-5345(13)70005-8