Australasian Transport Research Forum 2024 Proceedings
27-29 November, Melbourne, Australia
Publication website: https://australasiantransportresearchforum.org.au/

High-performance route choice set generation on large networks

Jake Moss [1,2], Pedro Camargo [1] , Christina de Freitas [1], Renata Imai [1]

[1]Outer Loop Consulting

[2] University of Queensland

Email for correspondence (presenting author): jake@outerloop.io

Abstract

The generation of choice sets for route choice algorithms is the most time-consuming step of most well-established route choice algorithms. Still, computationally efficient implementations are not readily available for researchers and practitioners interested in using route choice algorithms in research and practice. To plug that gap, we present an open-source and computationally efficient implementation of two choice set algorithms and analyse their performance against a large vehicle GPS dataset in Australia.

1 Introduction

Despite being the subject of extensive research over the past three decades, route choice has yet to gain significant traction in the transportation modelling practice. This slow adoption can be attributed, in part, to the lack of software availability for model application and estimation, particularly in traditional commercial modelling software. Additionally, the theoretical complexity of the topic, coupled with the lack of consensus on the ideal algorithms for specific use cases, pose substantial barriers to its use. Our research aims to overcome these hurdles and inspire a shift towards route-choice adoption with tangible contributions to the availability of computationally efficient open-source software.

Pushing against these barriers, the importance of several travel types other than peak traffic and fixed-route transit has increased substantially in the last few years. Of specific interest to this research are long-distance travel, particularly freight, and active mode travel. In both cases, route choice does not depend on (or respond to) congestion, making existing equilibrium or dynamic algorithms somewhat useless without heavy modifications not currently supported by commercial software.

Thus, the first step to increasing the adoption of route choice in forecasting models is to make high-performance software for choice set generation and route choice available to researchers and practitioners alike. To that extent, an open-source implementation is ideal to allow users to extend the software with novel algorithms or incremental algorithmic changes without the penalty of having to implement all algorithms from scratch.

2 Literature review and context

Although the treatment of path overlaps has received the most attention in route choice research since the publication of the C-Logit algorithm (Cascetta et al., 1996), one should note that choice set formation is far and above the most time-consuming step in any of the mainstream route choice algorithms available and thus critical to its performance and adoption in the transport modelling practice.

As recently pointed out in the literature (Camargo and Jayakrishnan, 2023), very few new algorithms for choice set formation have been proposed in the last 10 to 15 years, of these, the Directed Acyclic bush set approach proposed for route choice algorithm ReMULAA

(Camargo, 2014) and the Breadth-First Search Link Elimination (BFS-LE) (Rieser-Schüssler et al., 2013), with variations of the traditional link elimination and link penalisation still being the most prolific in the literature, including for research done in Australia (Malhotra et al., 2021). Notwithstanding the relatively small number of innovations in this field, substantial literature has been published on the use of vehicle GPS data in the validation of choice sets (Camargo and Tok, 2014; Rieser-Schüssler et al., 2013; Tahlyan and Pinjari, 2020).

As already mentioned, the computational performance of choice set generation algorithms is critical to their adoption and, therefore, substantial portions of the literature (Malhotra et al., 2021; Tahlyan and Pinjari, 2020; Wang et al., 2023) become somewhat irrelevant as references to our research, as they have either not discussed their underlying implementation or chosen to use packages that are known for having mediocre performance (e.g. Python's networkx), while some of the papers have conflated performance enhancements such as topological simplification with improvements to the choice set generation implementation (Rieser-Schüssler et al., 2013; Tahlyan and Pinjari, 2020). Such optimisations are beneficial, we believe a distinction should be made between general optimisations and those pertinent to specific algorithms and their implementations.

3 Use case

As part of a large statewide freight model development, the state's cadastral network was complemented with a skeleton network covering the remainder of the country as the main component of this model's supply side. The network includes 245,000km of centreline roads, over 338,000 links and 273,000 nodes, translating into over 620,000 directed links, substantially larger than most planning models in Australia and in line with the largest statewide models currently used abroad.

Compass IoT was chosen as the source for the dataset used to underpin the development of the choice sets. The dataset used for the model estimation included ~260,000 individual GPS traces, of which ~183,000 traces meet the data quality parameters for map-matching (valid route set).

The map-matched GPS traces were initially filtered down by selecting only those routes that were map-matched to over 99% accuracy[1] and were not particularly circuitous, indicating the trace is part of a route and not a simple trip.

The map-matched routes that met the above criteria were further processed, and the '*link id*' and order of the links as they were traversed were identified for each route. Routes covering less than 3km in distance, with less than 42 links, or with links travelled in both directions more than once were also removed. We also removed all routes where a single link was responsible for more than 1/8 of the entire route's distance.

Applying the above selection criteria to the valid routes output by the map-matching procedure led to a choice set that included 3,960 trips, or just over 1.5% of the individual trips present in the raw dataset.

4 Implementation

The choice set generation was implemented as a Cython module integrating into the existing AequilibraE infrastructure. This integration allows it to benefit from established optimisations such as graph compression via the construction of a topologically equivalent graph and the removal of provably unimportant dead-ends. While helpful, these optimisations are not specialised to the route choice set generation and should be applied before all graph algorithms. The flexibility of the existing infrastructure enables reconstruction of the network when desired to perform point-to-point route choice set generation instead of traditional

[1] Imprecisions in map-matching were mostly due to the inclusion of vehicles that do not use general network links, such as bus corridors, waterways and private roads.

zone-to-zone, allowing for more granular and accurate modelling of route choices, especially in the context of micro-mobility and active modes while still leveraging all the optimisations. Previous implementations are limited to interzonal results.

Two choice set generation algorithms were implemented for comparison, Link Penalisation (LP), and Breadth-First Search with Link Elimination (BFS-LE). Recognising the importance of performance for adoption and the inadequacies of previous implementations, key decisions were made regarding data structures and implementation medium to minimise the overhead of choice set generation and storage. Both methods were implemented in Cython facilitating integration with AequilibraE and access to standard C++ data structures. As BFS-LE operates on a graph of graphs, exploring unique graphs linked by the removal of a single edge from the shortest path. Each graph can be uniquely categorised by a set of removed links from a common base graph, eliminating the need to explicitly maintain the graph of graphs. Instead, the set of removed links is generated and stored in Breadth-First Search (BFS) order.

To efficiently store and determine the uniqueness of new routes or removed link sets, modified hash functions with specific properties were utilised, enabling storage and nesting within standard C++ data structures. A commutative hash function was employed for the removed link sets to allowing for amortised O(1) order-independent uniqueness testing. Although removed link sets are constructed incrementally, an incremental hash function was not used as it was not deemed a worthwhile optimisation. The removed link sets rarely exceed double digits, even on large networks. This may be an area worth exploring for networks with significantly larger numbers of desired routes than links between origin destination pairs.

For route uniqueness testing of discovered routes, a similar approach to the removed link sets was used. However, due to the order-dependent nature of routes, a traditional, non-commutative hash function was selected. As cryptographic security was not a requirement for our purposes, a general-purpose integer hash function was chosen. Further research could explore the use of specialised integer vector hash functions; as we did not find the hashing had a non-negligible influence on the runtime performance, this optimisation was not tested.

The process is designed to run multiple calculations simultaneously across the origin-destination pairs, utilising multi-core processors and improving computational performance. As N. Rieser-Schüssler et al. noted, pathfinding is the most time-consuming stage in generating a set of route choices. Despite the optimisations implemented to reduce the computational load of maintaining the route set generation overhead, computational time is still not trivial, as pathfinding remains the dominant factor in determining runtime.

5 Results

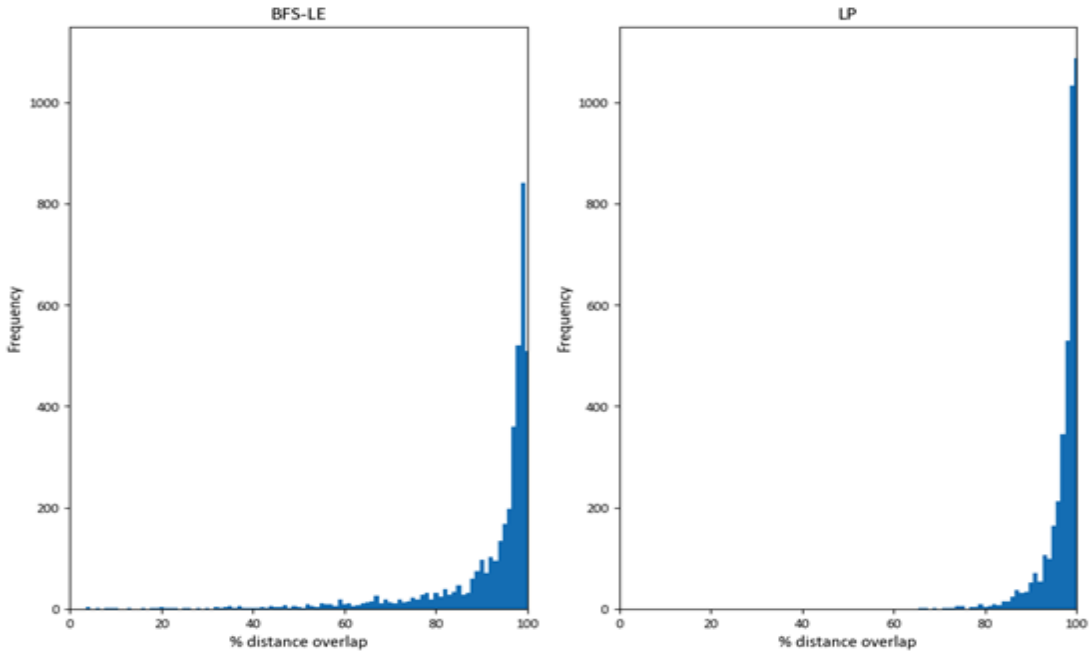
A choice set comprising 20 routes was generated for each trip using the BFS-LE and LP methods. For each choice set generation method, the necessary time to create the choice sets was approximately 6 CPU minutes for BFS-LE and 3.5 CPU minutes for LP, utilising six cores.

Subsequently, a succinct analysis of all routes within each choice set was conducted. This analysis included calculating the percentage of overlapping distance between the unique nodes of all routes and the route derived from GPS traces. The term 'percentage of overlapping distance' denotes the shared distance across all unique links in the choice set and the GPS route.

Figure 1 illustrates the distribution of distance overlap percentages for the BFS-LE and LP-generated choice sets compared to the GPS routes. Notably, the LP method exhibits a

higher frequency of routes with a distance overlap exceeding 99% compared to BFS-LE. Furthermore, the minimum distance overlap percentage for LP is greater than that for BFS-LE, suggesting that the routes in the choice sets obtained via LP potentially share more links with the original GPS route than those derived from BFS-LE.

Figure 1: – Percentage of distance overlap for BFS-LE and LP



5.1 Computational performance

All timings were produced on a Ryzen 5600x, 6c12t, run with four iterations for 3174 unique OD pairs. Only the minimum timings are reported.

Profiling using the tool perf indicates that LP has little overhead, and 99% of assignment runtime is spent on path-finding as expected. In contrast, BFS-LE exhibits more significant overhead, though path-finding remains the predominant component of runtime, accounting for 95%. The processes of link loading, results formation, and general setup and tear down methods did not consistently register in the profiling results, likely obscured by inlining and other optimisations, nor did they impact the benchmark results.

To indicate the impact of the overhead, we instrumented both methods to show the estimated path-finding computations per second. Link penalisation performs 68104 path-finding computations for the 3174 OD pair across six cores at a wall clock runtime of 35s or roughly 324 path-finding computations per second. BFLSE performs 132012 path-finding computations in 57s across six cores, or ~386 path computations per second. This is approximately 1.9x that of link penalisation at 1.6x the runtime.

Table 3: – Summary stats for BFS-LE and LP

	LP	BFS-LE
Runtime (6c, min)	34.74s	56.79s
Path computations	68104 (~324/s)	132012 (~386/s)
Miss count	4624 (6.8%)	68758 (52%)

Mean route length	128	183

The high miss rate of the BFS-LE method, which denotes the ratio of duplicate routes to total routes found, can be attributed to the extended lengths of routes and the relatively limited size of the route sets generated. Specifically, as BFS-LE systematically removes each link to create a new graph, it may generate excessive graphs, surpassing what is necessary to populate the route set. When applied to a high-resolution network, the average number of links contained within a single path may exceed the route set size demanding that the sub-graph exploration order is randomised. This excess can prematurely fill the route set, inhibiting the algorithm from conducting deeper iterative explorations. Moreover, the algorithm's removal of links could compromise the integrity of critical paths. For instance, consider a route that utilises a continuous stretch of highway devoid of exits; removing any segment within this stretch could render the entire path suboptimal. Such disruptions may lead to increased route collisions in subsequent iterations. Link penalisation does not exhibit this behaviour; by only penalising previously found routes, critical paths may still be utilised. Consequently, BFS-LE may not be ideally suited for generating choice sets in long-distance travel scenarios.

The higher routes per second rate of BFS-LE may be attributed to the additional route traversal performed by LP. After the destination is deemed reachable and the route has been formed, it must be traversed again to penalise the links utilised before it is checked for uniqueness. While the link penalisation can be performed during the traversal, the penalisation is comparatively a very small portion of the runtime that adds up over thousands of routes.

6 Future improvements

As previously mentioned, while investigating incremental and vector-based hash functions could yield improved performance, we believe these optimisations are not worthwhile as path-finding dominates the runtime. (Rieser-Schüssler et al., 2013) proposes utilising A-Star Landmarks to remedy this; however, at each step of BFS-LE, a new shortest path must be found on the modified graph. This is always a modification of a single link, building on the previous graph. Currently, the path-finding algorithm, A*, recomputes the entire path, ignoring all previous work. While this is simple to implement, consideration should be given to incremental heuristic searches. This family of heuristics can update the previous shortest path to account for graph modifications without recomputing the whole path. Lifelong Planning A* (Koenig et al., 2004) aims to be an efficient incremental search heuristic for fully dynamic (arbitrary edge insertion/deletion) shortest path problems. It propagates the edge weight updates to the affected nodes' estimated cost. This could provide more efficient path-finding after an edge removal than alternatives focusing on faster single-shot path-finding.

7 Conclusions

This paper presents a high-performance choice set generation module implemented as a Cython module integrated into AequilibraE and leverages existing topological simplification and other optimisation techniques. The flexibility of the existing software allows for point-to-point choice set generation, enabling more granular and accurate modelling of route choices, particularly during the estimation of models utilising real-world GPS data and in the context of applying route choice models for micro-mobility and active modes.
Two choice set generation algorithms, Link Penalisation and Breadth-First Search with Link Elimination, were implemented, focusing on performance. Preliminary results on a large-

scale statewide network demonstrate the module's effectiveness, generating ~63,000 choice sets in approximately 3.5 CPU minutes for LP and 6 CPU minutes for BFS-LE.
Analysis of the generated choice sets reveals that the LP produces routes with higher distance overlap with the map-matched GPS routes than BFS-LE. Profiling results indicate that path-finding dominates the runtime for both algorithms. The performance of BFS-LE may be affected by long route lengths and small route set sizes, leading to more route collisions than LP.
Future work may explore incremental search heuristics, such as Lifelong Planning A* (LPA*), to improve the efficiency of path finding after edge removals in BFS-LE. This could reduce the cost of recomputing entire paths at each step, albeit at the expense of additional overhead.

In conclusion, the proposed high-performance choice set generation module demonstrates the potential for utilising different route choice algorithms in efficient and scalable software. The integration with AequilibraE and the optimisation techniques employed make it a valuable tool for transportation modelling practitioners and researchers alike.

References

Camargo, P.V. de, Tok, A., 2014. Validating truck route enumeration algorithms using GPS data, in: 93rd TRB Conference Proceedings. p. 14.

Camargo, P., Jayakrishnan, R., 2023. Route choice, in: The Future of Modelling Symposium. Austin, Texas, USA.

Camargo, P.V., 2014. ReMulAA - a New Algorithm for the Route Choice Problem.

Koenig, S., Likhachev, M., Furcy, D., 2004. Lifelong Planning A∗. Artificial Intelligence 155, 93–146. https://doi.org/10.1016/j.artint.2003.12.001

Malhotra, R., Advani, C., Bhaskar, A., 2021. A review on discrete route choice set generation algorithms. Presented at the Australasian Transport Research Forum (ATRF), 42nd, 2021, Brisbane, Queensland, Australia.

Rieser-Schüssler, N., Balmer, M., Axhausen, K.W., 2013. Route choice sets for very high-resolution data. Transportmetrica A: Transport Science 9, 825–845. https://doi.org/10.1080/18128602.2012.671383

Tahlyan, D., Pinjari, A.R., 2020. Performance evaluation of choice set generation algorithms for analyzing truck route choice: insights from spatial aggregation for the breadth first search link elimination (BFS-LE) algorithm. Transportmetrica A: Transport Science 16, 1030–1061. https://doi.org/10.1080/23249935.2020.1725790

Wang, H., Moylan, E., Levinson, D., 2023. Route Choice Set Generation on High-Resolution Networks. Transportation Research Record 03611981231188775. https://doi.org/10.1177/03611981231188775