On Computation and Application of k Most Locally-Optimal Paths in Road Networks

Holger Döbler and Björn Scheuermann
Humboldt-Universität zu Berlin, Berlin, Germany
Email: {holger.doebler, scheuermann}@informatik.hu-berlin.de

***Abstract*—For some applications, e.g. route planning services, it is desirable to answer a point-to-point shortest path query on a road network with a set of alternative paths. We discuss the general requirements for such sets of paths such as shortness, diversity, etcetera. As a measure to rank reasonable alternatives we propose the local optimality ratio, because it implicitly covers all of these requirements. We present an algorithm that computes the k best alternatives in terms of this measure.**

I. Introduction

In every-day life, car drivers use route planning software, either web-based or as part of a dedicated navigation device, to find the optimal route for a pair of origin and destination locations. The objective function that a user seeks to obtain an optimal path for, may depend on the user as well as the situation. Therefore the creators of route planning software provide the user with a set of objective functions to select from. Typical objective functions are path lengths with respect to an arc weight function such as "travel distance", "travel time", or "travel time with the side condition that certain road classes are avoided". But users are often not satisfied with blindly trusting the automatically computed result and like to be presented with some of the suboptimal alternative paths as well in order to have the freedom to choose from a set of paths.

The problem, which paths should reasonably be presented to a user, is not mathematically well-defined, as opposed to the problem of existence and finding of the optimal path with respect to some given arc weights. First we discuss (non-exhaustively) different answers to this vague question found in literature.

Assuming that the user's desired arc weight function is given and is non-negative, we address the question of reasonable alternatives to the minimal cost paths without considering different arc weight functions. One well-known selection criterion for alternative paths is "local optimality," but existing algorithms [1] perform a Boolean test for local optimality on candidate paths, not considering whether one accepted candidate has a higher degree of local optimality than other, including discarded, candidate paths.

Our contribution is to introduce a quantitative measure for paths, the local optimality ratio (LOR), and to present an algorithmic solution to finding the set of k paths with highest LOR for directed graphs with static non-negative arc weights.

The rest of this paper is organized as follows. In Section I-A we give an overview of related work. We discuss use cases of alternative route sets and the associated objectives to search for in Section II. In Section III we explain the notion of local optimality and introduce related quantities. We present our proposed algorithm in Section IV along with an evaluation that indicates that the algorithm is feasible for online queries on urban area sized networks in practice. In Section V we discuss remaining deficiencies of the algorithm and present approaches to alleviate them as well as to extend the algorithm for use in other classes of road networks.

A. Related work

A set of alternative paths can be obtained by optimizing paths with respect to different arc weights, or combinations thereof, or by computing paths that are Pareto-optimal with respect to different arc weights [2], [3]. Instead we focus on alternative paths with respect to a single arc weight function.

The iterative penalty method [4]–[6] as well as the gateway path method [7] produce sets of paths that are short[1] as well as notably different.

The method of "alternative graphs" [8] can be seen as an extension of the iterative penalty method that abolishes some of its deficiencies and efficiently encodes the resultant set of alternative paths in a reduced sub-graph. In [9] is shown how these ideas can be turned into algorithm suitable for interactive use.

The authors of [10] introduce the concept of plateaus as a quantitative measure to rank single via paths, i.e. paths that can be represented as concatenation of two shortest paths. A plateau is maximal path sub-graph, whose nodes as via-nodes induce the same resultant single via path. It is argued, that paths corresponding to long plateaus are preferable, but the explicit selection criterion applied is not specified.

Abraham et al. [1] also focus on single via paths and propose that a sub-optimal path can be considered "admissible" if every subpath up to a certain length is optimal, if the relative stretch of every subpath is bounded by a given constant, and if the sharing with the shortest path is bounded. They propose an algorithm that approximatively tests single via paths for passing these criteria for threshold values that are parameters to the algorithm.

[1]Hereafter, we shall always understand "short" as "optimal with respect to the selected arc weight function."

II. Applications of alternative paths

We will discuss two use cases of alternative paths: giving freedom of choice to users of route planning software and providing a reduced search space for more complex optimization problems. In the bigger part of this paper we will focus on the former application and discuss the latter application only for the sake of additional motivation.

As explained in the introduction, the need for alternative routes arises when presenting point-to-point query results to a user of route planning software. In particular we consider the following requirements to the set of paths being proposed:

All best paths shall be included.
: A user is likely to be interested in each path that is best with respect to the chosen arc weights.

The paths shall be pairwise notably different.
: A user would anticipate that a tiny variation of a "good" path has similar properties in most respects. Therefore such tiny variations need not to be explicitly presented.

All paths shall be "somewhat short."
: In the end the user wants a path that is short with respect to the given arc weight function. Note that this is contrary to the request for notably different paths, since in most cases there are several tiny variations of the shortest path that are much shorter than any notably different alternative.

These properties are intentionally formulated loosely and without an exact definition here. Several concrete methods that generate sets of alternative routes found in literature [1], [4]–[10] do, explicitly or implicitly, respect all of these requirements.

Besides enabling a user to make her own choice, alternative paths can be used as reduced search space in a more complex optimization problem. For example, consider the problem of system optimal route assignment. A stochastic optimization scheme like simulated annealing or a genetic algorithm would subsequently evaluate the objective function for different elements in the search space, giving favor to better route assignments. The search space of all origin-destination paths is usually huge and contains many paths that are either likely to be dismissed by the stochastic optimizer anyway, or undesirable by the user, and therefore do not qualify as parts of practically good solutions.[2] Assuming that the objective function's evaluation is expensive, it would be economic to reduce the search space a priori to a set of reasonable routes. But does a good reduced search space have the same requirements on paths as a good selection to be presented directly to the user? Obviously the requirement of containing all of the shortest paths holds for search spaces, as well as the rather vague requirement that paths should be short. Whether two nearly identical paths should be part of a reduced search space is not that clear, but as we assumed that the objective function is expensive to evaluate, we can argue that if the search space to be explored has to be as small as feasible, more diverse paths would bear a greater potential for optimization.

For now the application as reduced search space is only intended as motivating example and will not be discussed further within this publication.

III. Local optimality

Now we introduce a quantitative measure for paths, with the property that sets of paths with highest values of this measure tend to fulfill the requirements stated above. From Abraham et al. [1] we adopt the notion of local optimality: for a given path, let the "interior" of a path be the sub-path that is gained by removing the first and last edge.[3] A path P is T-locally optimal, if every sub-path whose interior is shorter than T is a shortest path. We define $\ell_{\mathrm{LO}}(P)$ of a path P from s to t (origin and destination node) as the largest number x, such that P is x-locally optimal. Then we define the local optimality ratio (LOR) $q_{\mathrm{LO}}(P) := \ell_{\mathrm{LO}}(P)/\ell(P)$ where $\ell(P)$ denotes the length of P. A shortest path has $q_{\mathrm{LO}}(P) = \infty$ and all other paths have a local optimality ratio in $[0, 1)$, where higher values correspond to more locally-optimal paths. A path with high LOR consists of long shortest sub-paths that are telescoped with long consecutive overlaps.

We claim that paths with high LOR are likely to be relevant to a user, because one can show that

- Shortest paths always have the highest LOR.
- A significant overlap with a high-LOR path leads to a small LOR.

Whether all paths with high LOR are decently short, depends on the prioritization of "short" and "divers". In our experiments we never found stretch factors beyond 2 for paths that were computed, but it is easy to construct malicious examples where an arbitrarily long path is the second-best in terms of LOR. To eliminate this problem one could either introduce a tight bound the total stretch of paths or change the normalization to $q_{\mathrm{LO}}(P) = \ell_{\mathrm{LO}}(P)/\ell(P)^s$ with some $s > 1$ giving favor to shorter paths. Though not evaluated, the latter approach could turn out to be a good tool to balance path length versus path diversity.

IV. Proposed algorithm

We will now roughly sketch the developed algorithm. Given a weighted digraph G, a source s and a destination t, we grow one shortest path tree (SPT) from s and a reversed one from t. The connected components of the intersection of both trees are the plateaus. To each plateau corresponds one single via path. The plateau length yields a lower bound on the LOR [1]: $q_{\mathrm{LO}} \geq d(u, v)/\ell(P)$. Upper bounds for all plateaus can be computed in linear time by traversing each shortest path tree once.

When searching for k-highest LOR paths, we must employ additional shortest path searches to refine the bounds of all candidate paths. We do not need to compute exact values of

[2] We assume that a solution to the system optimal route assignment problem is practically good, if the routes are fair enough, that human car drivers are likely to comply.

[3] The interior of a path consisting of less than tree arcs has length zero.

ℓ_{LO}; we grow just enough SPTs to separate the k best paths from the rest, i.e., until the $(k-1)^{\mathrm{th}}$ highest lower bound is higher than the k^{th} highest upper bound. Deducing lower and upper bounds on $\ell_{\mathrm{LO}}(P)$ from a list of forward SPTs rooted at nodes in P is simple. Let u (v) be the first (last) node of the plateau. Let R be the list of nodes between s and u that are roots of known SPTs, including u as last element, sorted by their occurrence in P, and let's assume that P is not a shortest path, because otherwise $\ell_{\mathrm{LO}}(P) = \infty$ anyway. Let $\hat{P}_x$ be the longest sub-path of P starting at x that is a shortest path. Then $\ell_{\mathrm{LO}}(P) \leq \min_{x \in R} \ell(\hat{P}_x)$.

$$\text{Let } g_P(x,y) := \begin{cases} \min(\ell(\hat{P}_x), \ell(\hat{P}_y)) & \text{if } (x,y) \in P \\ \ell(\hat{P}_x \cap \hat{P}_y) & \text{else} \end{cases}$$

where $(x,y) \in P$ means that (x,y) is one arc of P. Then $\ell_{\mathrm{LO}}(P) \geq \min_{(x,y) \in R} g_P(x,y)$, where $(x,y) \in R$ means the set of pairs (x,y) that lie consecutively in R. Therefore, each additional SPT has the potential to refine the bounds on $\ell_{\mathrm{LO}}(P)$ for one or more candidate paths. One can use properties of the SPTs known at a time to predict which nodes are irrelevant as SPT roots to further refine bounds on LOR, and we use heuristics to predict which nodes are most promising to grow a SPT from, in order to separate the k best paths from the rest. To restrict grown SPTs in size, we use a tight bound of 1.5 on the path's stretch factor during evaluation. Details about heuristics will be given in a follow-up publication. Though our algorithm still requires $O(N)$ SPTs in the worst case, our evaluation on OpenStreetMap data of the metropolitan area of Berlin shows, that we typically require less than $20k$ partial SPTs, scanning approximately $k \cdot \left|\{v \mid d_{sv} + d_{vt} < 1.5 d_{st}\}\right|$ nodes, where d denotes the shortest path distance.

Abraham et al. [1] argued that the computation of local optimality length requires quadratically many shortest path queries. Their objective is to test only whether $\ell_{\mathrm{LO}}(P)$ is above some threshold value and they favored a cheap 2-approximation that may dismiss paths with $T \leq \ell_{\mathrm{LO}}(P) < 2T$ false negatively.

A. *Evaluation*

For evaluation we used OpenStreetMap data of the area of Berlin. On the largest strongly connected component of the graph we computed arc weights as the geometrical length of an arc divided by the speed limit. That is, we computed an approximation of the free flow travel time but without taking into account delays caused by traffic lights. For $k \in \{2,3,4,5,6,8,10,12,16,20,24\}$ we chose pairs (s,t) uniform at random, neglecting pairs of nodes whose distance is less than 5 minutes, because intuitively if origin and destination are too close together the number of reasonable alternatives becomes small. For 100 (s,t) pairs, and for each value of k we computed the k-highest LOR single via paths and recorded the number of SPTs needed and the number of nodes scanned during all SPT growths. Afterwards, we computed exact values of LOR for each path that was returned.

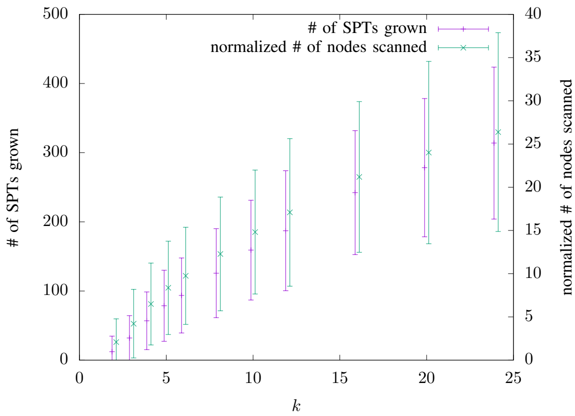

Fig. 1. Average number of SPTs grown and average normalized number of nodes scanned.

In our implementation (C++11, GCC 5.3.0), each query for k alternative paths took less than 50ms on a single core of a Intel(R) Core(TM) i5-4210U CPU, which can be considered fast enough for online queries in route planning software. We varied the hard limit on paths' stretch and found that even for values greater than 1.5, paths with stretch > 1.5 where rarely returned. Even without any stretch bound, paths with stretch > 2 where never returned.

In Fig. 1 we show that the average number of SPTs that where grown to compute the k-highest LOR paths. The figure also shows the average ratio of number of nodes scanned in all consecutive SPT searches and the number of nodes whose via-path-length is less than 1.5 times the source-destination-distance. The latter shows that indeed the average number of nodes needed to scan to find k-highest LOR paths is only about a factor k as large as the number of nodes scanned to find the shortest path. Together with the large number of SPTs grown, typically more than $10k$, this means that the SPTs grown are on average much smaller than the initial two SPTs. In Fig. 2 we show the average stretch of alternative paths depending on their LOR rank. The stretch saturates quickly for ranks greater than 2 at a value of about 1.2 but with a huge standard deviation. In Fig. 3 we show the paths' average LOR depending on the rank.

V. Conclusion and Outlook

We have extended the concept of locally-optimal paths by introducing the quantitative measure of LOR and argued, why paths with a high LOR are good candidates for alternative routes in road networks. We proposed an algorithm that computes k best single via paths in terms of LOR in graphs with static non-negative arc weights. Despite its super-quadratic worst case run-time we have demonstrated in our evaluation that it is still feasible for alternative paths online queries in urban area sized road networks.

The proposed algorithm relies on single via paths that are constructed using bidirectional SPTs. Therefore non-single-

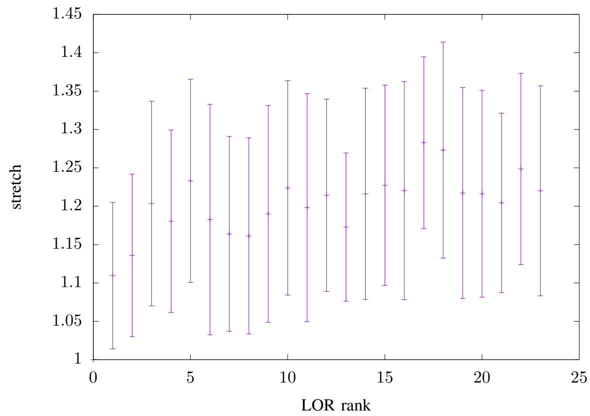

Fig. 2. Average stretch of paths found, depending on LOR rank.

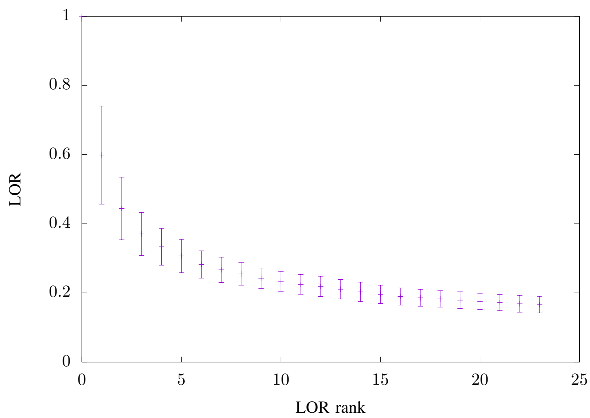

Fig. 3. Average LOR of paths found, depending on LOR rank.

via[4] paths will never be found by the algorithm. One can show that a non-single-via path's LOR cannot exceed a certain constant value, but the number of single via paths is finite, since it is bounded by the number of nodes. Since the LOR of single via paths can be arbitrarily small (typically many single via paths have $q_{\mathrm{LO}}(P) = 0$), it follows that there are non-single-via paths with a higher LOR than the n^{th} best single via path, for some finite n. One has grow additional SPTs in order to construct 2-via paths at all, and even more trees in order to classify them in terms of LOR.

But generalizing the above algorithm to an unconstrained search space is not the only motivation for stacking forward shortest path trees. In a graph where both the arc weights and the arc traversal times are time-dependent, single source shortest path search is still possible, if waiting at nodes is allowed. But since the arrival time at the destination node differs across paths, the bidirectional Dijkstra approach is not feasible. Obviously one possible approach is to avoid the need for a single destination backward SPT and use only telescoped forward shortest path searches just as in the static case of non-single-via paths.

Another topic that we are very interested in is the empirical evaluation of users' preferences for alternative paths, because as far as we know, the question of which properties qualify a path as a relevant alternative for a typical driver is only answered in terms of reasoning and speculation, just as we did it here. Obviously this is far out of scope for computer scientists, but project partners from the Department of Psychology are willing to design and perform experiments to answer these questions.

[4]A non-single-via path shall be a path that cannot be represented as concatenation of two shortest paths.

References

[1] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "Alternative Routes in Road Networks," *J. Exp. Algorithmics*, vol. 18, pp. 1.3:1.1–1.3:1.17, Apr. 2013.

[2] E. Q. V. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, May 1984.

[3] M. Müller-Hannemann and K. Weihe, "Pareto Shortest Paths is Often Feasible in Practice," in *Algorithm Engineering*, ser. Lecture Notes in Computer Science, G. S. Brodal, D. Frigioni, and A. Marchetti-Spaccamela, Eds. Springer Berlin Heidelberg, Aug. 2001, no. 2141, pp. 185–197.

[4] P. E. Johnson, D. S. Joy, D. B. Clarke, and J. M. Jacobi, "Highway 3. 1: An Enhanced Highway Routing Model: Program Description, Methodology, and Revised User's Manual," Oak Ridge National Lab., TN (United States), Tech. Rep. ORNL/TM-12124, Mar. 1993. [Online]. Available: http://www.osti.gov/scitech/biblio/6549364

[5] V. Akgün, E. Erkut, and R. Batta, "On finding dissimilar paths," *European Journal of Operational Research*, vol. 121, no. 2, pp. 232–246, Mar. 2000.

[6] Y. Chen, M. G. H. Bell, and K. Bogenberger, "Reliable Pretrip Multipath Planning and Dynamic Adaptation for a Centralized Road Navigation System," *Trans. Intell. Transport. Sys.*, vol. 8, no. 1, pp. 14–20, Mar. 2007.

[7] K. Lombard and R. L. Church, "The gateway shortest path problem: generating alternative routes for a corridor location problem," *Geographical Systems*, vol. 1, pp. 25–45, 1993.

[8] R. Bader, J. Dees, R. Geisberger, and P. Sanders, "Alternative Route Graphs in Road Networks," Rome, Apr. 2011.

[9] M. Kobitzsch, M. Radermacher, and D. Schieferdecker, "Evolution and Evaluation of the Penalty Method for Alternative Graphs." in *ATMOS*, ser. OASICS, D. Frigioni and S. Stiller, Eds., vol. 33. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, pp. 94–107.

[10] Cambridge VehicleInformation Tech. Ltd., "Choice Routing," 2005. [Online]. Available: http://www.camvit.com/camvit-technical-english/Camvit-Choice-Routing-Explanation-english.pdf