PAROS: Pareto Optimal Route Selection

Franz Graf, Hans-Peter Kriegel, Matthias Renz, Matthias Schubert Institute for Informatics, Ludwig-Maximilians-Universität München Oettingenstr. 67, D-80538 München, Germany {graf,kriegel,renz,schubert}@dbs.ifi.lmu.de

ABSTRACT

Modern maps provide a variety of information about roads and their surrounding landscape allowing navigation systems to go beyond simple shortest path computation. In this demo, we show how the concept of skyline queries can be successfully adapted to routing problems considering multiple road attributes. In particular, we demonstrate how to compute several pareto-optimal paths which contain optimal results for a variety of user preferences. The PAROS-system has two main purposes. The first is to calculate the route skyline for a starting point and a destination. Our demonstrator visualizes the result set for up to three road attributes. Therefore, we provide a dual view on the computed skyline paths. The first view displays the result paths on the road map itself. The second view describes the result paths in the property space, displaying the trade-off between the underlying criteria. Thus, a user can browse through the results in order to find the path which fits best to his personal preferences. The second component of our system suits analysis issues. In this component, we illustrate the functionality of the underlying route skyline algorithm. Thus, we provide benchmark information about processing time and the search space visited during route skyline computation.

Categories and Subject Descriptors

I.6.5 [Simulation And Modeling]: Model Development—*Modeling methodologies*; E.1 [Data]: Data Structures

General Terms

Algorithms, Measurement, Performance

Keywords

route finding, skyline query, road network, GIS, performance, query processing

1. INTRODUCTION

Modern spatial databases describing traffic networks provide a variety of information about the connections of two locations. For

Copyright 2010 ACM 978-1-4503-0032-2/10/06 ...\$10.00.

example, a database might store the distance, the speed limit, the altitude difference or the number of traffic lights for each road segment. Thus, a driver looking for the route which fits best to his personal preferences might want to consider various criteria at the same time. However, employing ordinary shortest path routing would require to select a single criteria or define a preference function by weighting each criterion. For example, a user might enter that his major preference is driving the fastest path with a weight of 80%, but still wants to consider driving distance with a minor weight of 20%. By still considering travel distance with a minor weight, the selected route might be considerable shorter and only slightly slower than the fastest path. Thus, the gas consumption and the risk of getting into a congestion should be considerably smaller. However, providing an exact weighting of quality criteria requires a level of expertise which cannot be expected from the ordinary user. To circumvent this problem, we adapted the wellknown skyline operator to route computation. The skyline operator computes a result set containing all data objects that are optimal under a linear combination of the underlying quality criteria. Thus, a user only has to determine the underlying quality criteria and the systems provides him with the selection of all pareto optimal paths. For example, the route skyline would contain a route which optimally fits to any possible trade-off between travel distance and travel time. To compute the route skyline, we proposed an efficient algorithm in [2], called ARSC which is capable to calculate all pareto optimal paths in efficient time. Let us note that computing a skyline of routes cannot be accomplished by the same algorithms as in ordinary vector databases. Since the database is represented as a network, the paths being elements of the route skyline must be constructed on-the-fly during query processing. Thus, the ARSC algorithms integrates the skyline operator into the network traversal allowing to prune suboptimal paths at an early state of computation. Though posing a route skyline query does not require a large effort by the user, it might provide a result set of various paths. In order to increase the usability of multi-preference path planning, the user must be enabled to intuitively browse and understand the provided alternative routes.

In this proposal, we introduce PAROS (Pareto-Optimal Route Selection) which offers a user interface to browse and understand the calculated skyline of alternative routes. The PAROS system can be applied to any multi-attribute traffic network and can visualize up to 3 quality criteria. As in an ordinary navigation system, PAROS contains a map viewer allowing to display the calculated routes in a 2D view. However, since route skylines also represent a vector in the feature space of quality criteria, PAROS simultaneously visualizes the resulting routes in the vector space of quality criteria. A second aspect of the PAROS system is the analysis of the underlying route skyline algorithm. Therefore, PAROS addition-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMOD'10, June 6-11, 2010, Indianapolis, Indiana, USA.



Figure 1: Screen shot of the PAROS interface for 2D Skyline Queries.

ally implements performance benchmarks and analysis functionalities. Furthermore, it is possible to compare the computational cost of route skyline computation to ordinary shortest path computation.

The rest of this paper is organized as follows. Section 2 surveys the underlying traffic network database and algorithm for computing the path skyline. In Section 3, we describe the functionality of the PAROS system which is designed around the surveyed query processor. Thus, we will describe the functionalities for posing a query and analyzing the result that will be shown at the demonstration session. Furthermore, we will introduce the analysis tool being part of the PAROS system which allows to investigate the behavior of the routing algorithms. The proposal closes with a short summary of the system features in Section 4.

2. ROUTE SKYLINE COMPUTATION

The database of our system consists of an multi-attribute graph describing an arbitrary traffic network. Each edge in this graph describes a road segment and is labeled with a feature vector describing the cost of traversing the segment w.r.t. all considered quality criteria. In order to have a lower bound approximation for the cost of reaching the destination for each criterion, we employed a network embedding w.r.t. a number of k landmark locations on the network as described in [1]. For each node, we store the cost of the shortest path to each landmark location w.r.t. each criterion. The embedding can be used to calculate a lower bound approximation of the remaining cost to reach the destination w.r.t. an arbitrary linear combination of quality criteria.

To calculate the route skyline for a given set of quality criteria, we employ the ARSC algorithm described in [2]. The basic idea of this algorithm is a best first traversal of the graph beginning with the starting location. During query procession the algorithm maintains two data structures. The first is a priority queue containing all nodes that still must be visited to find all skyline paths. The second structure consists of a table storing the already encountered pareto-optimal sub-routes for each visited node. Due to the monotonicity of local sub-routes, it can be shown that each sub route of a skyline route ending at the destination must be a skyline route between the starting location and its ending location. Thus, extending any path which is not part of the local skyline of its ending location cannot lead to a skyline route to the destination. To further speed up skyline computation, we additionally compare the lower bound approximation for any path to the current skyline of paths of the destination. If the lower bound approximation is already dominated by a member of the current skyline of the destination, the path can be pruned as well. The algorithms terminates when there is no path left that could be extended into a member of the route skyline to the destination node. For a more detailed description of the algorithm please refer to [2].

3. THE PAROS SYSTEM

In this section, we describe the functionalities of our demonstrator being built on top of the skyline query processor surveyed in the previous section.

3.1 Query Processing and Browsing

For our PAROS system, we rely on the publicly available maps from the Open Street Map (OSM) project¹ enhanced by topographic data from the SRTM² program. The available maps contain a multitude of information about streets, crossings, addresses, landmarks and the environment between them. To display the maps available at OSM, there already exist functional Java components which already allow zooming and moving the focus. In PAROS, we use the map view component of SwingX-WS³ which contains versatile viewing controls.

In order to apply route skyline computation to this data repository, we cannot rely on the rich map representation provided by OSM. First of all, the OSM format contains a lot of unnecessary information for route computation. A second more important reason is that several of the employed optimization criteria are not directly maintained in the maps. For example, we have information about traffic lights and altitudes connected to the nodes which have

¹http://www.openstreetmap.org

²http://www2.jpl.nasa.gov/srtm/

³http://swinglabs.org



Figure 2: Simplex Control for displaying the Skyline for three quality criteria.

to be reassigned and post processed into edge attributes of a multiattribute graph allowing skyline queries. Another reason making preprocessing of the map information advisable is that available maps often contain a lot of nodes which are not required for routing purposes, e.g. nodes that are integrated to display turns in an edge. In order to allow efficient path computation, deleting these nodes and combining the neighboring edged can significantly reduce the number of considered paths. Finally, since the algorithms underlying PAROS rely on a reference point model, we have to compute the corresponding embedding and store it within network. To display the computed paths in the map viewer and pose queries based on the addresses stored in OSM, PAROS provides mapping functions that can transfer locations and paths between the original OSM format and the internal PAROS format.

The PAROS frontend provides similar functionalities as an ordinary navigation system. To process a query, it is first of all necessary to input a start and a destination position. PAROS allows the input of these locations in two different ways. The first possibility is clicking directly into the displayed map. The second method is a textual input. After specifying a starting point and a destination, we now have to select the quality criteria for route computation. PAROS is designed for using up to three quality criteria. Though the route skyline query processor of PAROS basically has no limitations in the number of preference criteria, there are two reasons for the restriction to a maximum of three criteria. First of all, it was shown in [2] that the number of skyline elements is exponentially increasing with the number of dimensions. Thus, the magnitude of the result set might be too extensive even when having a good browsing functionalities. A second reason is that it is difficult to design a control which allows intuitive browsing for four or more preference weights. Thus, in the current version of PAROS, we provide the following three use cases for displaying the query results corresponding to the number of selected quality criteria.

USE CASE I:When selecting a single attribute, PAROS calculates the shortest path minimizing the cost w.r.t. the selected quality criterion only. The result is displayed in the map. Additionally, we provide all available information about the calculated paths like the expected travel time, the travel distance, the number of passed crossings, the number of traffic lights and so on.

USE CASE II: The user selects two quality criteria and thus, it is necessary to calculate a route skyline instead of a single path. The result of this type of query might contain a variety of different paths representing pareto-optimal trade-offs of both quality criteria. In other words, there exists an optimal route for all possible weightings between criterion A and criterion B. Simply displaying

all skyline paths in the same map is often not very helpful. Since skyline paths might strongly intersect, it might be hard to distinguish them from each other. Furthermore, the user needs to know the properties of the result paths and not only their way points in the map. To allow a user a better overview over this possibilities, we provide a view on the skyline in the vector space of quality criteria. In this view, the user can select the paths or preferences he is interested in by moving around a slide control describing the tradeoff between both criteria. While selecting one or more paths in the skyline view, PAROS simultaneously displays the selected routes in the map view and display the cost of each path w.r.t all available quality criteria. Figure 1 displays the 2D result browser with the skyline control on the left side and the map view on the right side of the window.

USE CASE III: Selecting three attributes also triggers the computation of a route skyline. However, since we have an additional attribute, we cannot simply draw the skyline in the vector space of quality criteria. Since displaying a three dimensional point set on a two dimensional screens requires some kind of projection, we introduce a new control which allows browsing the three dimensional user preferences on a two dimensional surface. To display the result and select a certain trade-off between the quality criteria, we display the simplex of all possible weightings between three criteria. This simplex can be visualized by a equilateral triangle where each inlying point represents a weighting like 20%-20%-60%. The corners of the triangle represent a weighting only considering one criterion while neglecting both other criteria. Thus, the closer a point is located to one of the corners the more important is the corresponding criterion in the underlying rating. Figure 2 displays the simplex control representing the space of all possible trad-offs. The skyline paths are represented by black dots in the triangle. Each path is located at the position corresponding to the ratio between the costs in each criterion. To select the optimal solution to a given preference the user can click an arbitrary position in the triangle and the system will display the corresponding skyline path in the control and on the map. Let us note that the actual cost of the path cannot be directly derived from the control as in the two dimensional case. However, as in the previous type of query the system reports all known path properties when a path is selected.

3.2 Benchmarking and Analysis Tool

A second purpose of the PAROS system beyond advanced route planing was the benchmarking and the illustration of the underlying skyline algorithms. Thus, the PAROS analysis tool allows us to monitor and report general performance measures for route planning algorithms like query time, result size, the number of accessed network nodes or the number of extended paths. Furthermore, we additionally might be interested in the progress of the algorithms during the graph traversal. Thus, PAROS allows to monitor the data structures of the underlying algorithms and thus, allows to report the progress of the current search. To display this progress, the accessed nodes are displayed in the map view.

4. CONCLUSION

In this proposal, we introduced PAROS a system for route selection based on multiple quality criteria like travelling time, distance, traffic lights, altitude difference etc.. To circumvent the explicit definition of a preference function, our system is based on the skyline operator computing all pareto-optimal routes between a start and a destination location. Since this skyline can contain multiple routes, it is necessary to enable the user to browse all paretooptimal paths in two views: the map and the feature space of quality criteria. In addition to query processing, PAROS yields a second type of functionality allowing to analyze the progress and the performance of the underlying query algorithms.

Acknowledgements

This research has been supported in part by the THESEUS program in the MEDICO and CTC projects. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020. The responsibility for this publication lies with the authors.

5. REFERENCES

- A. V. Goldberg and R. F. Werneck. "Computing Point-to-Point Shortest Paths from External Memory". In *Proc. of the 7th WS* on Algorithm Engineering and Experiments (ALENEX), SIAM, 2005.
- [2] H.-P. Kriegel, M. Renz, and M. Schubert. Route skyline queries: A multi-preference path planning approach. In Proceedings of the 26th International Conference on Data Engineering (ICDE), Long Beach, CA, USA, 2010.