Polynomial-Time Construction of Contraction Hierarchies for Multi-Criteria Objectives (Extended Abstract)

Stefan Funke
Universität Stuttgart
Institut für Formale Methoden der Informatik
70569 Stuttgart, Germany
funke@fmi.uni-stuttgart.de

Sabine Storandt
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
79110 Freiburg, Germany
storandt@informatik.uni-freiburg.de

Abstract

In this paper we consider a variant of the multi-criteria shortest path problem where the different criteria are combined in an arbitrary conic combination at query time. We show that contraction hierarchies (CH) - a very powerful speed-up technique originally developed for standard shortest path queries (Geisberger et al. 2008) - can be adapted to this scenario and lead - after moderate preprocessing effort - to query times that are orders of magnitudes faster than standard shortest path approaches. On the theory side we prove via some polyhedral considerations that the crucial node contraction operation during the CH construction can be performed in polynomial-time, while on the more practical side we complement our theoretical results with experiments on real-world data. Our approach extends previous results (Geisberger, Kobitzsch, and Sanders 2010) which only considered the bicriteria case.

This is an extended abstract of the full paper published in (Funke and Storandt 2013).

Introduction

In many routing applications the objective cannot be described sufficiently by only a single weight on the edges. For example, a driver is certainly keen on reaching his destination as quickly as possible, but he might be also interested in keeping the fuel costs low (see Figure 1 for an illustration). Similarly, when planning a bicycle trip one is not only interested in the length of the trip but also wants to avoid steep climbs. Most of the time we have conflicting objectives, that is, minimizing both values (e.g. travel time and fuel costs, or distance and positive height difference) at the same time is impossible. Therefore one either aims for a fair trade-off of both values or searches for the path which minimizes one of the values but does not exceed a given bound on the other, see Figure 1 for an example.

Here we focus on the former problem variant and hence ask for a minimum cost path for a *conic combination* of the edge costs, i.e. given a (di)graph $G(V, E)$ and edge weights $c_1, c_2 : E \rightarrow \mathbb{R}^+$ the optimal path $\pi = s, \cdots, t$ for given $\alpha, \beta \in \mathbb{R}^+$ is the one minimizing $\sum_{e \in \pi} (\alpha c_1(e) + \beta c_2(e))$. We call this problem the *conic combination shortest path (CCSP)* problem. If the coefficients α, β are known beforehand, the problem reduces to the single edge weight case and all available algorithms and speed-up techniques for this scenario apply as well. If the conic combination is revealed at
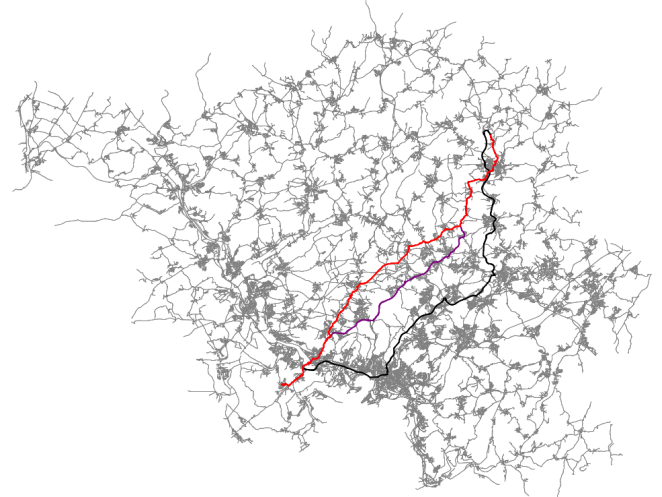

Figure 1: Alternative paths in the Saarland graph: The purple path is optimal in terms of travel time (37min), but has the highest fuel costs (€ 4.61). The black path has minimal fuel costs (€ 3.81), but needs time 44min. The red path is a fair trade-off with costs € 3.87 and a travel time of 40min.

query time only, still Dijkstra's algorithm can be applied in a straightforward manner. In larger road networks plain Dijkstra is too slow for most applications, though. Hence – like for conventional shortest path computations – developing preprocessing techniques to speed up query times is mandatory. For conic combinations of edge costs this was considered first in (Geisberger, Kobitzsch, and Sanders 2010). In this paper we improve upon their approach in several aspects, in particular generalizing the bicriteria setting to the multicritiera setting and proving polynomial time bounds for the crucial node contraction operation.

Outline of our Approach

In the general, multicriteria CCSP problem, each edge e bears d *cost values* $c_1(e), \ldots, c_d(e)$ and for a query specified by source s, target t and weights $\alpha_1, \ldots, \alpha_d \geq 0$ with $\sum \alpha_i = 1$ we are to find a path $\pi = v_0 v_1 \ldots v_k$ with edges $e_i = (v_i, v_{i+1}) \in E$ from $v_0 = s$ to $v_k = t$ minimizing the weighted sum $c(\pi) = \sum_{j=1}^{d} \alpha_j c_j(\pi)$ where

$$c_j(\pi) = \sum_{i=0}^{k-1} c_j(e_i).$$

In a preprocessing step, we construct a contraction hierarchy (CH) (Geisberger et al. 2008) which preserves optimal paths for any choice of the α_i. The basic idea of a CH is the iterative removal of nodes from the graph without affecting shortest path distances between the remaining nodes in the graph. This is achieved by inserting so-called *shortcuts*. Here the crucial operation is the decision whether edges uv and vw have to be replaced by a shortcut uw when a node v is contracted/removed. If not too many shortcuts are added, the original graph augmented with all inserted shortcuts allows for very efficient answering of shortest path queries by a modified bidirectional Dijkstra. Our main contributions of this paper are twofold: 1. a polynomial-time algorithm to decide whether a shortcut should be created or not in the multicriteria case and 2. an experimental evaluation that the number of added shortcuts is small enough that there is a considerable speed-up compared to ordinary Dijkstra.

Decision about Shortcut Creation

How can we characterize the set of s-t paths that are optimal for some choice of the α_i? In a dual view, we can associate with each path π a hyperplane

$$h_\pi : y = \sum_i (\alpha_i c_\pi^i)$$

in $\mathbb{R}^{d+1}$ (the space where d dimensions correspond to $\alpha_1, \ldots \alpha_d$ and one dimension to the objective function value). A path π is optimal for some choice of the α_i, iff its respective plane bounds the *lower envelope* of the hyperplanes corresponding to all s-t paths. The decision whether a shortcut uw has to be created – via the dual view – boils down to the test whether the path uvw appears on the lower envelope of the hyperplanes of all u-w paths.

While this lower envelope might have exponential complexity, we show that membership in the lower envelope for such a hyperplane can be decided in *polynomial time*. The high-level idea of our proof is as follows: we first show that vertices in this arrangement of hyperplanes have a certain minimum pairwise distance. This can be used to derive a lower bound on the hypervolume of facets of the lower envelope. Our algorithm in each round deliberately chooses values for the α_i, computes the optimal path for this choice of α_i's, and thereby decreases the hypervolume of the facet corresponding to the path π on the lower convex hull by a constant factor in each round. Hence polynomially many rounds suffice to decide whether π bounds the lower envelope. For a constant number of weights per edge, any algorithm for half-space intersection (or convex hull) can be used to obtain the desired polynomal running time guarantee.

Experimental Results

We have implemented our CH construction scheme for two and three edge weights and evaluated them on real-world data. The used test graphs (SL - Saarland, HE - Hessen, BW - Baden-Württembeg and CAL - California) are based on OpenStreetMap data. The implementation was written in C++, timings were taken on a single core of an Intel Core i5-3360M CPU with 2.80GHz and 16 GB RAM.

For the two-weight case, some results can be found in Table 1. While there is some effort required for preprocessing (less than 20 minutes for CAL), the resulting query times are orders of magnitudes faster than ordinary Dijkstra. The case of three edge weights exhibits similar speed-ups.

Note that we only contracted about 99.95% of the nodes during the shortcut creation as between the remaining nodes there was a large number of pareto-optimal paths, so a complete contraction would have added a too large number of shortcuts – slowing down both preprocessing and query processing. The query times are slightly worse than CH-query times for single edge weights which seems natural given the more complex problem setting.

The running times – both of preprocessing as well as for the queries – also depend on the metrics involved. Metrics that are somewhat similar – like euclidean distance and travel time – produce less shortcuts and therefore better query times. In contrast to that metrics of opposing nature – e.g. travel time vs. quietness of the route – result in more CH-edges (about 11 % more) and higher query times (by a factor of 6.4). Still the speed-up compared to ordinary Dijkstra is considerable and in the order of several magnitudes.

	nodes	edges	Dijkstra	
			polls	time(ms)
SL	203731	404521	$1.2 \cdot 10^5$	23.72
HE	1121082	2269020	$6.1 \cdot 10^5$	176.67
BW	2459354	4993582	$1.3 \cdot 10^6$	409.77
CAL	11283833	22918849	$7.2 \cdot 10^6$	2097.42

	CH Preprocessing		Dijkstra-CH Query		
	time(s)	edges	polls	time(ms)	speed-up
SL	4	715571	152	0.05	460
HE	65	4089958	506	0.29	611
BW	121	9186484	599	0.42	965
CAL	1078	44806866	2434	1.98	1059

Table 1: Experimental results: graph sizes and timings for ordinary Dijkstra (upper table); preprocessing and query times when employing our multicriteria CH (travel time and fuel costs). Query times and the poll numbers are averaged over 1000 random queries with the weight parameters α_1 being chosen u.a.r in $[0, 1]$ for each query ($\alpha_2 = 1 - \alpha_1$).

References

Funke, S., and Storandt, S. 2013. Polynomial-time construction of contraction hierarchies for multi-criteria objectives. In *In Workshop on Algorithm Engineering and Experiments (ALENEX)*, 41–54.

Geisberger, R.; Sanders, P.; Schultes, D.; and Delling, D. 2008. Contraction Hierarchies: Faster and simpler hierarchical routing in road networks. In *Workshop on Experimental Algorithms*, 319–333.

Geisberger, R.; Kobitzsch, M.; and Sanders, P. 2010. Route planning with flexible objective functions. In *ALENEX'10*, 124–137.