Theoretical Computer Science 645 (2016) 112–127

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Search-space size in contraction hierarchies ☆

Reinhard Bauer, Tobias Columbus, Ignaz Rutter *, Dorothea Wagner

Karlsruhe Institute of Technology, Germany

ARTICLE INFO

Article history:
Received 24 November 2015
Received in revised form 12 May 2016
Accepted 4 July 2016
Available online 15 July 2016
Communicated by F.V. Fomin

Keywords:
Shortest paths
Search space size
Contraction hierarchies
Speed-up techniques
Theoretical analysis

ABSTRACT

Contraction hierarchies are a speed-up technique to improve the performance of shortest-path computations, which works very well in practice. Despite convincing practical results, there is still a lack of theoretical explanation for this behavior.

In this paper, we develop a theoretical framework for studying search space sizes in contraction hierarchies. We prove the first bounds on the size of search spaces that depend solely on structural parameters of the input graph, that is, they are independent of the edge lengths. To achieve this, we establish a connection with the well-studied elimination game. Our bounds apply to graphs with treewidth k, and to any minor-closed class of graphs that admits small separators. For trees, we show that the maximum search space size can be minimized efficiently, and the average size can be approximated efficiently within a factor of 2.

We show that, under a worst-case assumption on the edge lengths, our bounds are comparable to those in the recent paper “VC-Dimension and Shortest Path Algorithms” of Abraham et al. [1], whose analysis depends also on the edge lengths. As a side result, we link their notion of highway dimension (a parameter that is conjectured to be small, but is unknown for all practical instances) with the notion of pathwidth. This is the first relation of highway dimension with a well-known graph parameter.

1. Introduction

Computing shortest paths in graphs is a fundamental problem in computer science with many applications, the most popular being the computation of routes in transportation networks. Though Dijkstra’s algorithm can be used to efficiently compute shortest paths in $O(n \log n + m)$ time, this is too slow for many applications such as the computation of shortest paths in continental-size road networks. In the last decades substantial progress on improving this query time has been made. So-called speedup-techniques have led to speedups of a factor of one million or more; see [3] for an extensive survey. One particularly successful technique are contraction hierarchies (CH), which are widely used, especially since they have proven to be easily adaptable to various settings, such as time-dependent routes [4], multi-modal transportation networks [11], routing for electrical vehicles [7], shortest paths with multi-criteria objectives [14], and even fast all-pairs-shortest path computations [10].

☆ Partially supported by the DFG under grant WA 654/16-2. A preliminary version of this paper has appeared as R. Bauer, T. Columbus, I. Rutter, D. Wagner, Search Space Size in Contraction Hierarchies, in *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13), pages 93–104, volume 7965 of LNCS, 2013*.

* Corresponding author.
E-mail addresses: reinhard.bauer@kit.edu (R. Bauer), tobias.columbus@kit.edu (T. Columbus), ignaz.rutter@kit.edu (I. Rutter), dorothea.wagner@kit.edu (D. Wagner).

http://dx.doi.org/10.1016/j.tcs.2016.07.003

Contraction hierarchies were introduced by Geisberger et al. [16], who evaluated their performance experimentally. The main idea of contraction hierarchies is the iterative *contraction* of nodes, where the contraction of a node v removes v from the graph and possibly inserts some shortcut edges between neighbors of v in order to preserve shortest-path distances (we present a brief description of contraction hierarchies in Section 2.2).

A common way to theoretically assess the performance of speed-up techniques is the maximum search space size, i.e., the maximum number of vertices a query has to process [6,5]. It turns out that the contraction hierarchy and also its performance in answering shortest-path queries depend strongly on the contraction order of the nodes. Finding a good node ordering that allows for fast shortest-path computations thus is an important problem. Practical implementations, such as the one by Geisberger et al. [16] employ heuristics for which no provable guarantees are known. Previous theoretical expositions rather focus on minimizing the size of the contraction hierarchy [6,21], i.e., the number of edges that are inserted during the contractions. In particular, it is known that minimizing the size of a contraction hierarchy is NP-complete. The only work providing provable performance guarantees for shortest-path computations in contraction hierarches, we are aware of, is the work of Abraham et al. [1,2]. They introduce the notion of highway dimension, a parameter that is conjectured to be small in real-world road networks, and prove sublinear query times under this assumption. However, the highway dimension of real-world instances is unknown, and may change as the length function changes. By contrast, we focus on providing bounds that rely on purely structural parameters of the graph, such as bounded treewidth or excluding a fixed minor. Our algorithms thus apply to classes of graphs that are defined purely by structural criteria, and our upper bounds are agnostic to the length function.

We note that theoretical results with better query times [28,13] exist, some of them even using similar techniques. They are, however, far from being practical. By contrast, our theoretical bounds apply to a widely used speed-up technique. It is also worth noting that recursive graph separation has been used as a heuristic in practical approaches [27], although, without providing theoretical guarantees. There have also been approaches for exploiting small treewidth for computing shortest paths. Chaudhuri and Zaroliagis [9] describe a data structure for answering shortest-path queries on graphs of small treewidth, which directly works on a tree decomposition. Planken et al. [23] apply a separator-based technique to speed up all-pairs-shortest-path computation in graphs of small treewidth. Recently, Dibbelt et al. [12] have followed our approach and demonstrated experimentally that it can be used for efficient customizable route planning. Very recently Funke and Storandt [15] have claimed provable guarantees on the search space for a randomized preprocessing of contraction hierarchies. However, they impose restrictions on the graph structure and the metric, and their query algorithm resembles a multi-level Dijsktra search rather than the query of a contraction hierarchy.

Contribution and outline We develop a theoretical framework for studying search-space sizes in contraction hierarchies. Due to the iterative definition of an algorithmic contraction hierarchy, it seems quite difficult to prove theoretical bounds, as this appears to inherently require arguments that are based on a local construction only. We overcome this drawback by giving a global description of the contraction hierarchy associated with a node ordering in Section 3.

Afterwards, in Section 4, we establish a connection between contraction hierarchies and two classical problems that have been widely studied. Namely, so-called filled graphs, which were introduced by Parter [22] in his analysis of Gaussian elimination, and elimination trees, which were introduced by Schreiber [26] for Gaussian elimination on sparse matrices. For trees, these connections in particular imply efficient algorithms that minimize the maximum search space size and approximate the average search space size within a factor of 2. This contrasts corresponding hardness results for other speed-up techniques, such as arcflags, where even processing trees optimally is NP-complete [5].

In Section 5, we show that nested dissection, a technique for finding elimination trees of small height, can be applied to construct orders α with provable bounds on the maximum search space size. For graphs of treewidth k and for graphs that admit small separators and exclude a fixed minor, we obtain maximum search space size $O(k \log n)$ and $O(\sqrt{n})$, respectively.

Finally, we compare our results with the results of Abraham et al. [1,2] in Section 7. If the length function is such that the highway dimension is maximal, then our results are comparable to theirs. However, our approach neither requires small maximum degree, nor does it depend on the diameter of the graph, and thus applies to a larger class of graphs. As a side result, we find an unanticipated and novel connection between highway dimension and pathwidth. This is, to our knowledge, the first relation between highway dimension and a more widely known graph parameter.

2. Preliminaries

In this section we collect preliminary notation and terminology that is used throughout the paper.

2.1. Graphs, paths, and lengths

A graph $G = (V, E)$ consists of a set V of vertices and an edge set E, where each edge $e \in E$ connects two vertices in V. In a directed graph, we use uv to denote the edge with source u and target v. In an undirected graph, we do not distinguish the source and target of an edge, and uv and vu denote the same edge. All graphs we consider are simple, i.e., they contain neither parallel edges nor loops. A weighted graph $G = (V, E, \text{len})$ is a graph with a weight function $\text{len}\colon E \to \mathbb{R}_{\geq 0}$. In case of ambiguity, we use a subscript to indicate to which graph a weight function refers, e.g., len_G for the weight function of graph G.