Project Number 288094

eCOMPASS

eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

STREP
Funded by EC, INFSO-G4(ICT for Transport) under FP7

eCOMPASS – TR – 029

Search-Space Size in Contraction Hierarchies

Reinhard Bauer and Tobias Columbus and Ignaz Rutter and Dorothea Wagner

July 2013

Search-Space Size in Contraction Hierarchies

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

Karlsruhe Institute of Technology
firstname.lastname@kit.edu

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 the recent results 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

Contraction hierarchies were introduced by Geisberger et al. [7], who evaluated their performance experimentally. Given a directed graph $G = (V, E)$ and a vertex $v \in V$, *contraction of* v means (i) removing v, and (ii) inserting a shortcut uw of length $\mathrm{dist}_G(u, w)$ for each unique shortest path (u, v, w) in G. Given an order $\alpha \colon V \to \{1, \dots, n\}$ of the vertices of $G = (V, E)$, a *contraction hierarchy* $\bar{G}_\alpha = (\bar{G}_\alpha^\wedge, \bar{G}_\alpha^\vee)$ of G is obtained by iteratively contracting the vertices in the order specified by α. Let E' denote the set of shortcuts that is created in this process. Then $\bar{G}_\alpha^\wedge = (V, E_\alpha^\wedge)$ and $\bar{G}_\alpha^\vee = (V, E_\alpha^\vee)$, where $E_\alpha^\wedge = \{uv \in E \cup E' \mid \alpha(u) < \alpha(v)\}$ and $E_\alpha^\vee = \{uv \in E \cup E' \mid \alpha(v) < \alpha(u)\}$.

The correctness of shortest path computation relies on the following proposition, which is due to Geisberger et al. [7]. It immediately implies that distance queries can be performed by a bidirectional query, searching $\bar{G}_\alpha^\wedge$ and $\bar{G}_\alpha^\vee$.

Proposition 1. $\mathrm{dist}_G(s,t) = \min_{v \in V} \mathrm{dist}_{\bar{G}}^\wedge(s, v) + \mathrm{dist}_{\bar{G}}^\vee(v, t)$ *for all* $s, t \in V$.

F.V. Fomin et al. (Eds.): ICALP 2013, Part I, LNCS 7965, pp. 93–104, 2013.

In the following, we refer to such a contraction hierarchy as an *algorithmic contraction hierarchy.* Obviously, the contraction hierarchy depends strongly on the ordering α. Finding a good node ordering that allows fast shortest-path computations thus is an important problem. Practical implementations, such as the one by Geisberger et al. [7] employ heuristics for which no provable guarantees are known. Previous theoretical expositions rather focus on minimizing the size of the contraction hiearchy [4,11]. 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 use separator decompositions and 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 a larger class of graphs, and are not dependent on the length function.

We note that theoretical results with better query times [17,6] 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 [16], although, without providing theoretical guarantees.

Contribution and Outline. We develop a theoretical framework for studying search-space sizes in contraction hierarchies. The iterative definition of an algorithmic contraction hierarchy is difficult to work with. In Section 2 we derive a global description of the contraction hierarchy associated with a node ordering.

Afterwards, in Section 3, we establish a connection between contraction hiearchies and two classical problems that have been widely studied. Namely, so-called filled graphs, which were introduced by Parter [12] in his analysis of Gaussian elimination, and elimination trees, which were introduced by Schreiber [15] for Gaussian elimination on sparse matrices. For trees, this implies an efficient algorithm for minimizing the maximum search space and a 2-approximation for the average search space. This contrasts hardness results for other speed-up techniques, such as arcflags, where optimal preprocessing for trees is NP-complete [3].

In Section 4, 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 5. 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.

2 A Formal Model of Contraction Hierarchies

In this section, we develop a theoretical model of contraction hierarchies that is simpler to work with than algorithmic contraction hierarchies. Let $G = (V, E)$ be a directed graph and let α be an ordering of V. Let $P_\alpha(s,t) = \{v \in V \mid \alpha(v) \geq \min\{\alpha(s), \alpha(t)\}$ and $\text{dist}_G(s,v) + \text{dist}_G(v,t) = \text{dist}_G(s,t) < \infty\}$, i.e., $P_\alpha(s,t)$ contains the vertices that lie on a shortest path from s to t and lie above at least one of s and t. The following theorem provides a global characterization of the algorithmic contraction hierarchy.

Theorem 1. *Let $G = (V, A)$ be a weighted digraph and let α be an order of its vertices. The arcs $A_\alpha^\wedge$ and $A_\alpha^\vee$ of $\bar{G}_\alpha^\wedge$ and $\bar{G}_\alpha^\vee$ are*

$$\begin{aligned} A_\alpha^\wedge &= \{uv \in A \mid \alpha(u) < \alpha(v)\} \cup \{uv \mid \alpha(u) < \alpha(v) \text{ and } P_\alpha(u,v) = \{u,v\}\} \\ A_\alpha^\vee &= \{uv \in A \mid \alpha(u) > \alpha(v)\} \cup \{uv \mid \alpha(u) > \alpha(v) \text{ and } P_\alpha(u,v) = \{u,v\}\}. \end{aligned}$$

The length of a shortcut uv in $\bar{G}_\alpha^\wedge$ or $\bar{G}_\alpha^\vee$ is $\text{dist}_G(u,v)$.

Not only does this theorem shed some light on the structure of algorithmic contraction hierarchies, but also suggests an alternative definition. Consider an arc st of G that is no unique shortest path. Then, removing st from G does not change any distances. If st is a unique shortest path, then $P_\alpha(s,t) = \{s,t\}$ anyway. Thus the following definition works equally well. For a weighted digraph $G = (V, E)$ and an order α of its vertices, we define $G_\alpha = (G_\alpha^\wedge, G_\alpha^\vee)$, where $G_\alpha^\wedge = (V, A_\alpha^\wedge)$ and $G_\alpha^\vee = (V, A_\alpha^\vee)$ by

$$\begin{aligned} A_\alpha^\wedge &= \{uv \mid \alpha(u) < \alpha(v) \text{ and } P_\alpha(u,v) = \{u,v\}\} \\ A_\alpha^\vee &= \{uv \mid \alpha(u) > \alpha(v) \text{ and } P_\alpha(u,v) = \{u,v\}\}\,. \end{aligned}$$

As in Theorem 1, we set $\text{len}_G^\alpha(uv) = \text{dist}_G(u,v)$. We call the pair $G_\alpha = (G_\alpha^\wedge, G_\alpha^\vee)$ a *formal contraction hierarchy*.

We remark that it immediately follows from this definition that if H is the digraph obtained from G by reversing all arcs, $H_\alpha^\wedge = G_\alpha^\vee$ and $H_\alpha^\vee = G_\alpha^\wedge$ hold. This allows us to prove statements about G_α by only considering $G_\alpha^\wedge$ since the analogous statement for $G_\alpha^\vee$ follows by reversing all arcs.

We now carry over several useful concepts from the algorithmic definition. A *shortcut* is an arc uw of G_α that is not contained in G, or for which $\text{len}_G(uw) > \text{dist}_G(u,w)$. Note that the latter type of shortcuts are included only to model the possible overwriting of arclengths in algorithmic contraction hierarchies. Given a shortcut uw in $A^\wedge$ or $A^\vee$, we are able to recover the vertex v that would have caused the insertion of uw in the corresponding algorithmic contraction hierarchy. Namely, let $S = \{v \in V \setminus \{u,w\} \mid \text{dist}_G(u,v) + \text{dist}_G(v,w) = \text{dist}_G(u,w)\}$. Note that $S \neq \emptyset$, for otherwise uw would be no shortcut. Note further that $\alpha(v) < \alpha(u)$ and $\alpha(v) < \alpha(w)$ for all $v \in S$ since otherwise we would have $v \in P_\alpha(u,w) = \{u,w\}$. We call the vertex $v \in S$ with $\alpha(v)$ maximal the *supporting vertex of uw*. It is easy to show that exactly the contraction of v causes the insertion of uw in the algorithmic contraction hierarchy $\bar{G}_\alpha$.

Lemma 1. *Let uw be a shortcut of G_α and let v be its supporting vertex. Then G_α contains $uv \in A_\alpha^\vee$ and $vw \in A_\alpha^\wedge$.*

We call the arcs, whose existence is guaranteed by Lemma 1, the *supporting arcs* of uw, and write $\sup(uw) = (uv, vw)$. Observe that if uv is a supporting arc of uw, then $\alpha(v) < \alpha(w)$, and thus chains of supporting arcs in $G_\alpha^\wedge$ and $G_\alpha^\vee$ are acyclic and do not descend indefinitely. This allows us to perform induction on the depth of the nested shortcuts below a given arc uw. We define the *shortcut depth* $\mathrm{scd}(uw)$ of an arc of G_α by $\mathrm{scd}(uw) = 1$ if uw is no shortcut, and by $\mathrm{scd}(uw) = \mathrm{scd}(uv) + \mathrm{scd}(vw)$ if uw is a shortcut with $\sup(uw) = (uv, vw)$. It is readily seen that, for a shortcut uw with $\sup(uw) = (uv, vw)$, we have $\mathrm{len}_G^\alpha(uw) = \mathrm{len}_G^\alpha(uv) + \mathrm{len}_G^\alpha(vw)$. Hence G_α still possesses the most essential properties of the algorithmic contraction hierarchy $\bar{G}_\alpha$.

A close look at the proof of Proposition 1 in Section 1 reveals that it merely depends on the fact that each arc uw with $P_\alpha(u,w) = \{u,w\}$ is contained in $\bar{G}_\alpha$ and has length $\mathrm{len}_{\bar{G}}^\alpha(uw) = \mathrm{dist}_G(u,w)$. Thus Proposition 1 also holds for formal contraction hierarchies, implying that a bidirectional variant of Dijkstra's algorithm on the contraction hierarchy G_α can be used to compute shortest paths in G.

To measure the performance of such computations, we define the search space of a query as $S(s, G_\alpha^\wedge) = \{u \in V \mid \mathrm{dist}_G^\wedge(s,u) < \infty\}$ and $R(t, G_\alpha^\vee) = \{u \in V \mid \mathrm{dist}_G^\vee(u,t) < \infty\}$. Clearly, the shortest-path query from s to t in G_α settles at most the vertices in $S(s, G_\alpha^\wedge) \cup R(t, G_\alpha^\vee)$. To maximize the performance of query algorithms, one is interested in an ordering α that minimizes $\max_{s,t \in V} |S(s, G_\alpha^\wedge)| + |R(t, G_\alpha^\vee)|$. To simplify the analysis, we rather concentrate on minimizing the *maximum search space size* $S_{\max}(G_\alpha) = \max\{|S(v, G_\alpha^\wedge)|, |R(v, G_\alpha^\vee)|\}$. Note that $2 \cdot S_{\max}(G_\alpha)$ is an upper bound on the number of vertices that is settled in any query, and thus bounding $S_{\max}(G_\alpha)$ gives a guarantee on the query performance in terms of the number of settled nodes. We denote the minimum maximum search space size by $S_{\max}(G) = \min_\alpha S_{\max}(G_\alpha)$. Similarly, we define the *average search space size* $S_{\mathrm{avg}}(G_\alpha) = 1/n^2 \cdot \sum_{s,t \in V} |S(s, G_\alpha^\wedge)| + |R(t, G_\alpha^\vee)|$.

There is still one downside of formal contraction hierarchies: Practical implementations of contraction hierarchies do not compute an actual formal (or even algorithmic) contraction hierarchy. Instead of inserting a shortcut only when it is strictly necessary, fast heuristics are used to quickly exclude the necessity of a shortcut in many cases. In some cases, this results in the addition of shortcuts that are not necessary. Thus bounding $S_{\max}(G_\alpha)$ may not have any practical implications since the additional shortcuts might increase the search space arbitrarily. To overcome this downside, we introduce one final type of contraction hierarchies, which also allows for additional shortcuts, yet preserve the properties that have turned out to be key to contraction hierarchies.

A *weak contraction hierarchy* H_α of a weighted digraph $G = (V, A)$ is a pair $(H_\alpha^\wedge, H_\alpha^\vee)$ of digraphs $H_\alpha^\wedge = (V, B_\alpha^\wedge)$ and $H_\alpha^\vee = (V, B_\alpha^\vee)$, such that the following conditions are satisfied.

(w1) $G_\alpha \subseteq H_\alpha$
(w2) $\alpha(u) < \alpha(v)$ for each $uv \in B_\alpha^\wedge$ and each $vu \in B_\alpha^\vee$

(w3) If uw is an arc of H_α that is not contained in G, then there is at least one pair of arcs $uv \in B_\alpha^\vee$ and $vw \in B_\alpha^\wedge$.

In the remainder of this section, we indicate how to extend our previous findings for contraction hierarchies to weak contraction hierarchies and investigate the relationship between different weak contraction hierarchies for the same ordering α. For this purpose, we fix a weighted digraph G and an ordering α of its vertices. As usual, we denote its formal contraction hierarchy by $G_\alpha = (G_\alpha^\wedge, G_\alpha^\vee)$. Additionally, we fix a weak contraction hierarchy $H_\alpha = (H_\alpha^\wedge, H_\alpha^\vee)$, whose arcs we denote by $B_\alpha^\wedge$ and $B_\alpha^\vee$, as above.

The notions of shortcuts and shortcut depth carry over literally to H_α. It follows immediately from (w1) and (w3) that for each shortcut uw in H_α, there is a pair of supporting arcs $uv \in B_\alpha^\vee$ and $vw \in B_\alpha^\wedge$. Although distances and arc lengths are only of secondary importance in the remaining sections, we still want to point out that it is not hard to give a "correct" definition of arc lengths on H, such that the following lemma holds true.

Lemma 2. *Let H_α be a weak contraction hierarchy.*
(a) $\mathrm{len}_H^\alpha(uw) \geq \mathrm{dist}_G(u, w)$ *for all arcs uw of H_α.*
(b) $\mathrm{len}_H^\alpha(uw) = \mathrm{dist}_G(u, w)$ *for all arcs uw of H_α with $P_\alpha(u, w) = \{u, w\}$.*

As indicated above, the proof of Proposition 1 relies on exactly the containment of G_α and the properties guaranteed by Lemma 2, and it thus holds also for any weak contraction hierarchy. In particular, the same shortest-path algorithm works for weak contraction hierarchies.

In view of property (w1) it is clear that G_α is the smallest weak contraction hierarchy. Moreover, if H_α and K_α are weak contraction hierarchies, then $(H_\alpha^\wedge \cup K_\alpha^\wedge, H_\alpha^\vee \cup K_\alpha^\vee)$ is a weak contraction hierarchy. Thus, there exists a unique maximal weak contraction hierarchy, which we denote by M_α. It is not difficult to see that $S(u, G_\alpha^\wedge) \subseteq S(u, H_\alpha^\wedge) \subseteq S(u, M_\alpha^\wedge)$, and symmetrically $R(u, G_\alpha^\vee) \subseteq R(u, H_\alpha^\vee) \subseteq R(u, M_\alpha^\vee)$ for all weak contraction hierarchies H_α. In particular, this shows that $S_{\max}(G_\alpha) \leq S_{\max}(H_\alpha) \leq S_{\max}(M_\alpha)$ for all weak contraction hierarchies H_α. Thus, we will concentrate on bounding $S_{\max}(M_\alpha)$ in the following sections. Before we do so, we give a more explicit description of M_α.

Lemma 3. *A weak contraction hierarchy H_α is maximal if and only if H_α satisfies the following properties.*
(i) Each arc of G is contained in H_α.
(ii) For any two arcs $uv \in B_\alpha^\vee$ and $vw \in B_\alpha^\wedge$, H_α also contains uw.

Note that this immediately implies an efficient way to construct the arcs of M_α by inserting shortcuts between each pair of neighbors during the contraction. In particular, the structure of M_α is independent of the weights on G.

Finally, we note that the query performance does not solely depend on the number of vertices in the search space, but also on the number of arcs. For a search space $S(u, M_\alpha)$, this number is certainly bounded by $|S(u, M_\alpha)|^2$. Moreover, the size $|M_\alpha|$ has a crude upper bound in terms of $S_{\max}(M_\alpha)$.

Lemma 4. *For any n-vertex directed graph with an ordering α of its vertices, we have $|M_\alpha| \leq 2n \cdot S_{\max}(M_\alpha)$.*

3 Contraction Hierarchies and Filled Graphs

In this section, we establish a link between contraction hierarchies and the more well-studied graph elimination game, which was introduced by Parter [12]. Let $G = (V, E)$ be an undirected graph and let α be an ordering of its vertices. We consider the so-called *elimination game* played on G. Beginning at $G^1 = G$, one removes in each step $i = 1, \dots, n$ the vertex $v_i = \alpha^{-1}(i)$ and its incident edges from G^i. Afterwards, the graph G^{i+1} is obtained from G^i by inserting *fill edges*, such that the neighbors of v_i form a clique. Denote by F^i the set of edges inserted in step i, and let $F = \bigcup_{i=1}^{n} F_i$. The filled graph G^α is now commonly defined to be the undirected graph with edge set $E \cup F$. For our purposes it is more convenient to define the filled graph as the according directed graph with all arcs pointing upwards with respect to α. That is, the *filled graph* $G^\alpha = (V, A^\alpha)$ is defined by $A^\alpha = \{uv \mid \{u, v\} \in E \cup F \text{ and } \alpha(u) < \alpha(v)\}$. Note that the only difference from the construction of M_α is that M_α is constructed from a digraph, whereas the elimination game is played on an undirected graph. In what follows, we denote for a digraph G by ${}^*G$ the underlying undirected graph. The following theorem immediately follows from the construction of M_α and G^α.

Theorem 2. *Let G be a directed graph with an ordering α of its vertices. Let further $\overleftarrow{M}_\alpha^\vee$ denote $M_\alpha^\vee$ with reversed arcs. Then $M_\alpha^\wedge, \overleftarrow{M}_\alpha^\vee \subseteq {}^*G^\alpha$. Moreover, if G contains for each arc uv also the opposite arc vu, then $M_\alpha^\wedge = \overleftarrow{M}_\alpha^\vee = {}^*G^\alpha$.*

Theorem 2 has many far-reaching consequences, and we will only explore a few of them in this paper. It turns out that the definition of M_α is nothing essentially new, and has indeed already been defined and studied by Rose and Tarjan [13]. However, much of the work on filled graphs is primarily concerned with the problem of minimizing the number of arcs in G^α; see the survey by Heggernes [10]. Minimizing the fill-in corresponds to minimizing the number of shortcuts in a contraction hierarchy, and hence its space requirements. We rather focus on the implications of Theorem 2 regarding search spaces and their size.

Corollary 1. *Let $G = (V, A)$ be a weighted digraph with vertex ordering α. Then $S(u, M_\alpha^\wedge), R(u, M_\alpha^\vee) \subseteq S(u, {}^*G^\alpha)$. In particular $S_{\max}(G_\alpha) \leq S_{\max}({}^*G^\alpha)$.*

An analogous statement holds for the average search space size S_{avg}. Our next goal is an alternative description of $S_{\max}({}^*G^\alpha)$ known as the height of the elimination tree of ${}^*G^\alpha$. For this purpose consider again an undirected graph G and a filled graph G^α. Associated with G^α is the so-called *elimination tree* $T(G^\alpha)$ of G. The elimination tree $T(G^\alpha)$ has vertex set V but contains for each $u \in V$ only the arc uv of G^α with minimal $\alpha(v)$. Again the usual definition of $T(G^\alpha)$ is undirected, but it is natural to choose $\alpha^{-1}(n)$ as the root. With this choice the usual definition coincides with our definition. The height of $T(G^\alpha)$ with respect to this root is the *elimination tree height*, denoted by $\text{ht}(G^\alpha)$. The following lemmas relate the search space size in G^α with $\text{ht}(G^\alpha)$.

Lemma 5. *Let G be a connected graph, α an order of V and let $T = T(G^\alpha)$. Then $S_{\max}(T) = 1 + \text{ht}(G^\alpha)$.*

Proof. Let r denote the root of T, and denote by $p(u)$ the vertices lying on the path from u to r. Then $\mathrm{ht}(G^\alpha) = \max_{u \in V} |p(u)| - 1$ and it therefore suffices to show $S(u,T) = p(u)$ for all $u \in V$. This is trivially satisfied, since due to the connectivity of G, each vertex $u \in V$ distinct from r is the source of precisely one arc uv. □

The crucial property of $T(G^\alpha)$ is that if u and v are two vertices connected by a path in p in the filled graph G^α of G, then there is a path p' from u to v in $T(G^\alpha)$. It obviously suffices to prove this statement when p is an arc, as the general case then follows by induction.

Lemma 6. *If uv is an arc of the filled graph G^α, then there exists a path with source u and target v in $T(G^\alpha)$.*

Proof. Denote by $p(u)$ the unique path in $T(G^\alpha)$ from u to the root r. We show by descending induction on $\alpha(u)$ that we have $p(v) \subseteq p(u)$ for all arcs uv of G^α. Note that this implies our claim, for it then follows that $p(u) = q \cdot p(v)$, where q is a path from u to v.

If $\alpha(u) = n - 1$, then $\alpha(v) = n$, and hence $v = r$. The claim holds trivially.

If $\alpha(u) < n - 1$, let uw be the unique arc of $T(G^\alpha)$ with source u. By the definition of $T(G^\alpha)$, we have $\alpha(w) \leq \alpha(v)$, and $p(u) = uw \cdot p(w)$. If $v = w$, we are done. Otherwise, G^α contains the arc wv as both v and w are neighbors of u at the time of its removal during the elimination game. The induction hypothesis therefore implies $p(v) \subseteq p(w)$, and hence $p(v) \subseteq p(u)$. □

This allows us to finally relate search spaces in G^α and $T(G^\alpha)$.

Corollary 2. *Let $G = (V, E)$ be a graph, α an order on V and $T = T(G^\alpha)$ the corresponding elimination tree. Then $S(u,T) = S(u, G^\alpha)$ for all $u \in V$.*

Corollary 1 and 2 immediately imply the following.

Corollary 3. *For any connected weighted digraph G with vertex ordering α,*

$$S_{\max}(M_\alpha) \leq \mathrm{ht}({}^*G^\alpha) + 1 .$$

Despite its innocent appearance, the above corollary is central to our analysis of search spaces in contraction hierarchies, for it enables us to translate upper bounds on $\mathrm{ht}({}^*G^\alpha)$ into upper bounds on $S_{\max}(G)$. Upper bounds from the literature are not seldomly accompanied by algorithms to determine orders α so that $\mathrm{ht}({}^*G^\alpha)$ is at most the upper bound at hand. Without any further modifications these algorithms may be used to compute contraction orders α with good upper bounds on $S_{\max}(G^\alpha)$.

As a first application of the above result let us consider contraction hierarchies of undirected trees T. In this case, shortest paths are unique and each possible shortcut is thus present in the contraction hierarchy T_α. Hence, T_α, the maximal weak contraction hierarchy M_α and the filled graph T^α coincide. Moreover, Schäffer [14] has given a linear-time algorithm to compute optimal elimination orders for trees, and we may thus conclude that the problem of minimizing $S_{\max}(T_\alpha)$ is solvable in linear time. The techniques of the next section can also be used to obtain a 2-approximation for $S_{\mathrm{avg}}(T_\alpha)$; we omit the details.

4 Contraction Orders from Nested Dissection

Next we show how to compute orders that yield small search spaces for several classes of graphs. The main idea is to exploit Corollary 3, which relates search space sizes and elimination tree heights. One way to construct orderings that give guarantees on the elimination tree height is the use of *nested dissection*, which goes back to George [8].

Let $0 < b < 1$ and let $f(n)$ be a monotonically increasing function. A *(b,f)-balanced separator decomposition* of an undirected n-vertex graph $G = (V, E)$ is a rooted tree $\mathcal{T} = (\mathcal{X}, \mathcal{E})$ whose nodes $X \in \mathcal{X}$ are disjoint subsets of V and that is recursively defined as follows. If $n \leq n_0$ for some fixed constant n_0, then $\mathcal{T}$ consists of a single node $X = V$. If $n > n_0$, then a (b, f)-balanced separator decomposition of G consists of a root $X \subseteq V$ of size at most $f(n)$ whose removal separates G into at least two subgraphs $G_1, \ldots, G_d$ with at most bn vertices, each. The children of X in $\mathcal{T}$ are the roots of (b, f)-balanced separator decompositions of $G_1, \ldots, G_d$. For clarity, we will always refer to the vertices of $\mathcal{T}$ as *nodes*. We use $\mathcal{T}_X$ to denote the subtree of $\mathcal{T}$ rooted at a node X, and by G_X the connected subgraph of G induced by the vertices contained in $\mathcal{T}_X$. For a vertex $u \in V$, we denote the unique node X of $\mathcal{T}$ with $u \in X$ by X_u. A node X of $\mathcal{T}$ has *level* $\text{level}(X) = i$ if the unique simple path from X to the root of $\mathcal{T}$ has length i.

Remark 1. Let G be an undirected graph and let $\mathcal{T}$ be a (b, f)-balanced separator decomposition of G. If $\{u, v\}$ is an edge of G with $\text{level}(X_u) \geq \text{level}(X_v)$, then X_v is an ancestor of X_u.

Proof. Consider the lowest common ancestor X of X_u and X_v in $\mathcal{T}$. If $X \neq X_v$, then X_u and X_v lie in distinct subtrees of $\mathcal{T}_X$. However, by construction of $\mathcal{T}$, this means that X separates X_u from X_v, contradicting the existence of the edge $\{u, v\}$. □

Given a (b, f)-balanced separator decomposition of G, we determine an associated (b, f)-*balanced nested dissection order* $\alpha = \alpha(\mathcal{T})$ on the vertices of G by performing a post-order traversal of $\mathcal{T}$, where the vertices of each node are visited in an arbitrary order. It follows immediately from Remark 1 and the construction of α that for any edge $\{u, v\}$ of G with $\alpha(u) < \alpha(v)$ the node X_v is an ancestor of X_u. This property remains valid also for the corresponding filled graph G^α.

Lemma 7. *Let $G = (V, E)$ be an undirected graph and $\alpha = \alpha(\mathcal{T})$ a nested dissection order associated with a given (b, f)-balanced separator decomposition $\mathcal{T} = (\mathcal{X}, \mathcal{E})$ of G. Then X_v is an ancestor of X_u for any arc uv of the filled graph G^α with $\alpha(u) < \alpha(v)$.*

Proof. It suffices to that that X_v is an ancestor of X_u for each edge $\{u, v\}$ with $\alpha(u) < \alpha(v)$ of G^i, where G^i is the graph before the ith step in the elimination game. Observe that the above discussion establishes exactly this property for $G = G^1$. We show that G^{i+1} satisfies the property if G^i does.

Consider the vertex $v_i = \alpha^{-1}(i)$ that is removed in the ith step of the elimination game. Let $\{u, v\}$ be a fill-edge with $\alpha(u) < \alpha(v)$ that is inserted in this step. Then G^i contains edges $\{v_i, u\}$ and $\{v_i, v\}$. By the induction hypothesis, we have that both X_u and X_v are ancestors of X_{v_i} in $\mathcal{T}$ since $\alpha(v_i)$ is minimal among the vertices in G^i. Note that this implies that either X_u is an ancestor of X_v or vice versa. Our assumption $\alpha(u) < \alpha(v)$ and the construction of α imply that X_u is an ancestor of X_v. This finishes the proof. □

To simplify notation, we denote by $\mathcal{T}(u)$ the union of all nodes that lie on the unique simple path from X_u to the root of $\mathcal{T}$. The above lemma has immediate implications in terms of search spaces and elimination tree height. An easy induction over the length of a path in G^α yields the following.

Corollary 4. *Let $G = (V, E)$ be an undirected graph, $\mathcal{T} = (\mathcal{X}, \mathcal{E})$ an (b, f)-balanced separator decomposition of G, and let $\alpha = \alpha(\mathcal{T})$ be an associated nested dissection order. Then (i) $S(u, G^\alpha) \subseteq \mathcal{T}(u)$, and (ii) $\mathrm{ht}(G^\alpha) \leq |\mathcal{T}(u)|$.*

In particular, Corollary 1 and 4 together imply the following theorem.

Theorem 3. *Let $G = (V, E)$ be a weighted digraph and $\alpha = \alpha(\mathcal{T})$ a nested dissection order of a (b, f)-balanced separator decomposition of $^*G$. Then we have $|S(u, M_\alpha^\wedge)|, |R(u, M_\alpha^\vee)| \leq |\mathcal{T}(u)|$.*

In order to find upper bounds on $S_{\max}(M_\alpha)$, it remains to bound $|\mathcal{T}(u)|$. This issue cannot be handled simultaneously for all families of graphs, but needs special treatment depending on the properties of G. We will first study a rather general setting, and afterwards specialize to graphs that exclude a fixed minor.

It is not hard to see that $|\mathcal{T}(u)| \leq n_0 + h \cdot f(n)$, where h is the height of $\mathcal{T}$. Further, it follows from the balance of the decomposition that $h \leq \log_{1/b} n$. In particular, for graphs with treewidth at most k, the $(1/2, 1)$-balanced separator decomposition for trees facilitates a $(1/2, k+1)$-balanced separator decomposition with $n_0 = k + 1$. We have the following theorem.

Theorem 4. *Let G be a weighted digraph of treewidth at most k. There exists an order α, such that $S_{\max}(M_\alpha) \leq (k+1)(1+\log n)$ and $|M_\alpha| \leq 2n(k+1)(1+\log n)$.*

If the separator size is not fixed, but depends on the graph size, better bounds can be achieved. For example for minor-closed graph classes that admit $(b, a\sqrt{n})$-balanced separators, we have $\mathrm{ht}(G^\alpha) \leq n_0 + \sum_{i=0}^{\infty} a\sqrt{b^i n} = n_0 + a/(1-\sqrt{b})\sqrt{n} = O(\sqrt{n})$. According to Lemma 4, this yields a contraction hierarchy of size $O(n^{3/2})$. However, a more sophisticated analysis due to Gilbert and Tarjan [9] proves that the number of fill arcs is $O(n \log n)$. The next theorem summarizes this discussion.

Theorem 5. *Let $\mathcal{C}$ be a minor-closed graph class with balanced $O(\sqrt{n})$-separators. Any $G \in \mathcal{C}$ admits an order α with $S_{\max}(M_\alpha) = O(\sqrt{n})$ and $|M_\alpha| = O(n \log n)$.*

5 Comparison with Highway Dimension

In this section, we compare our bounds with the ones obtained by Abraham et al. [1,2]. Their results employ the highway dimension, a notion that, unlike the graph parameters we use, also depends on the edge lengths. We show that their bounds are comparable to ours if the edge lengths are sufficiently ill-behaved.

First, we briefly recall the definition of highway dimension. Let $G = (V, E)$ be a weighted undirected graph. Given a vertex $u \in V$, the set of vertices v of distance at most ε from u is called the *ball of radius* ε around u, and is denoted by $B_\varepsilon(u)$. We say that a vertex u covers a shortest path p if p contains u. The *highway dimension* $\mathrm{hd}(G)$ of G is the smallest integer d, such that all the shortest paths p in G of length $\varepsilon < \mathrm{len}(p) \leq 2\varepsilon$ that intersect a given ball of radius 2ε can be covered by at most d vertices. Abraham et al. [1] prove that for a n-vertex weighted graph G with highway dimension d and diameter D and maximum degree Δ, there exists an ordering α, such that the size of G^α is $O(nd \log D)$, and such that distance queries can be answered in time $O((\Delta + d \log D) \cdot d \log D)$.

In the remainder of this section, we proceed as follows. We consider the edge lengths of G that maximize $\mathrm{hd}(G)$. For this particular choice of lengths, we construct a $(2/3, \mathrm{hd}(k))$-balanced separator decomposition of G, which then provides the link to nested dissection orders. We note that the same proof can be adapted to the slightly different definition of highway dimension in [2].

Lemma 8. *Let $G = (V, E)$ be a connected graph, let k be the maximum highway dimension over all possible edge lengths on G, and let $H \subseteq G$ be a connected subgraph with $|V(H)| \geq 2k + 2$. Then H can be separated into at least two connected components of size at most $\lceil |V(H)|/2 \rceil$ by removing at most k vertices.*

Proof. Let H be a connected subgraph of G with h vertices and let $H_1 \subseteq H$ be a connected subgraph with $\lfloor h/2 \rfloor$ vertices. Denote the vertex sets of H and H_1 by V_H and V_1, respectively. Define lengths $\mathrm{len}\colon E \to \mathbb{R}^+$ by setting the length of an edge to 1 if it has exactly one endpoint in V_1, and to $\varepsilon = 3/h$, otherwise. Observe that the length of a simple path in H_1 is at most $h\varepsilon/2 \leq 3/2$.

Consider a ball B with radius $3/2$ around any vertex u of H_1. Then $V_1 \subseteq B$. By our choice of k, and the definition of highway dimension, there exists a set S of at most k vertices, such that each shortest path in G that intersects H_1 and has length between $3/4$ and $3/2$ contains at least one element of S. We claim that the removal of $S \cap V_H$ separates H into at least two connected components with at most $\lceil h/2 \rceil$ vertices, each.

Consider any path with source $s \in V_1$ and target $t \in V \setminus V_1$. This path necessarily contains an edge $\{u, v\}$ with $u \in V_1$ and $v \in V \setminus V_1$. By the choice of edge lengths, this edge is a shortest path of length 1, and thus one of its endpoints is in S. Hence S separates H_1 from $H \setminus V_1$. It remains to verify that the connected components of $H \setminus S$ contain at most $\lceil h/2 \rceil$ vertices, each, and that there are at least two such components. The former claim follows immediately from our choice of H_1. For the latter claim, note that $|V_1| \geq k + 1$ and $|V_H \setminus V_1| \geq k + 1$ imply $|V_1 \setminus S| \geq |V_1| - k \geq 1$ and $|V_H \setminus (V_1 \cup S)| \geq |V_H \setminus V_1| - k \geq 1$. □

This lemma allows us to separate each subgraph of $n' \geq 2k+2$ vertices into at least two connected components with at most $\lceil n'/2 \rceil \leq \frac{2}{3}n'$ vertices by removing at most k vertices. We have the following corollary.

Corollary 5. *Let $G = (V, E)$ be a connected undirected graph with maximum highway dimension k. Then G admits a $(2/3, k)$-balanced separator decomposition, whose leaves have size at most $2k+1$.*

A simple calculation shows that $\mathrm{ht}(G) \leq 2k + 1 + k \cdot \log_{3/2} n$. It is known for the pathwidth $\mathrm{pw}(G)$ that $\mathrm{pw}(G) \leq \mathrm{ht}(G)$ [5].

Theorem 6. *Let G be a weighted undirected graph. There exist edge lengths on G, such that* $\mathrm{hd}(G) \geq (\mathrm{pw}(G) - 1)/(\log_{3/2} n + 2)$.

To our knowledge, this is a novel and unanticipated relation between highway dimension and more commonly used graph parameters. Moreover, Corollary 5 allows a comparison of our results with those of Abraham et al. [1].

Theorem 7. *Let G be an undirected graph with diameter D and maximum degree Δ. Let β denote the order constructed by Abraham et al. [1]. There exist edge lengths on G and a nested dissection order α, such that*
(a) $|M_\alpha| \leq O(\log n / \log D)|G_\beta|$, and
(b) the worst-case running time of distance queries in M_α is at most a factor of $O(\log^2(n)/\log^2(D))$ greater than that in G_β.

Proof. Choose the edge lengths such that G attains its maximum highway dimension k. Recall from [1], that their optimal order β results in a contraction hierarchy G_β that has $m_\beta = O(nk \log(D))$ arcs and on which a distance query has worst-case running time $T^{\beta}_{\text{query}} = O((\Delta + k \log D) \cdot k \log D)$.

By virtue of Theorem 4 and Corollary 5, there exists a nested-dissection ordering α, such that $S_{\max}(M_\alpha) = O(2k + 1 + k \log n) = O(k \log n)$. Using Lemma 4, we have $|M_\alpha| = O(nk \log n)$, which immediately implies (a). For (b), we use that Dijkstra's algorithm relaxes at most $S_{\max}(M_\alpha)^2$ edges. □

We note that, for graphs that bear some resemblance to road networks, it seems quite likely that $\Theta(\log n) = \Theta(\log D)$. It is remarkable that the results are so close, given that Abraham et al. bound both the vertices and arcs in the search space, while our crude bound on the number of arcs is simply the square of the number of vertices in the search space. Any improvement on this bound would immediately imply faster query times. It is moreover worth noting that our machinery neither requires small maximum degree nor small diameter.

Conclusion. We have developed a theoretical framework for studying search spaces in contraction hierarchies. Our main contributions are a global description of contraction hierarchies and the connection to elimination games. Using nested dissection, we are able to compute contraction orders with sublinear search spaces for large classes of graphs. Under a worst-case assumption on the highway dimension, our results, even though our constructions ignore edge lengths, are comparable to those of Abraham et al. [1]. Our main open questions are: (i) Are there stronger bounds on the number of arcs in search spaces? (ii) Is there an efficient approximation for the maximum or average search space size?

References

1. Abraham, I., Delling, D., Fiat, A., Goldberg, A.V., Werneck, R.F.: VC-Dimension and Shortest Path Algorithms. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 690–699. Springer, Heidelberg (2011)
2. Abraham, I., Fiat, A., Goldberg, A.V., Werneck, R.F.: Highway Dimension, Shortest Paths, and Provably Efficient Algorithms. In: Proc. 21st Annual ACM–SIAM Symp. Disc. Alg. (SODA 2010), pp. 782–793. SIAM (2010)
3. Bauer, R., Baum, M., Rutter, I., Wagner, D.: On the Complexity of Partitioning Graphs for Arc-Flags. In: Proc. 12th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS 2012). OpenAccess Series in Informatics (OASIcs), pp. 71–82 (2012)
4. Bauer, R., Columbus, T., Katz, B., Krug, M., Wagner, D.: Preprocessing Speed-Up Techniques Is Hard. In: Calamoneri, T., Diaz, J. (eds.) CIAC 2010. LNCS, vol. 6078, pp. 359–370. Springer, Heidelberg (2010)
5. Bodlaender, H.L., Gilbert, J.R., Kloks, T., Hafsteinsson, H.: Approximating treewidth, pathwidth, and minimum elimination tree height. In: Schmidt, G., Berghammer, R. (eds.) WG 1991. LNCS, vol. 570, pp. 1–12. Springer, Heidelberg (1992)
6. Fakcharoenphol, J., Rao, S.: Negative weight edges, shortest paths, near linear time. In: Proceedings of the 42nd Symposium on Foundations of Computer Science (FOCS 2001), pp. 232–241 (2001)
7. Geisberger, R., Sanders, P., Schultes, D., Vetter, C.: Exact Routing in Large Road Networks Using Contraction Hierarchies. Transport. Sci. 46(3), 388–404 (2012)
8. George, A.: Nested dissection of a regular finite element mesh. SIAM Journal on Numerical Analysis 10(2), 345–363 (1973)
9. Gilbert, J.R., Tarjan, R.E.: The analysis of a nested dissection algorithm. Numerische Mathematik 50(4), 377–404 (1986)
10. Heggernes, P.: Minimal triangulations of graphs: A survey. Discrete Mathematics 306(3), 297–317 (2006)
11. Milosavljević, N.: On optimal preprocessing for contraction hierarchies. In: Proc. 5th Internat. Workshop Comput. Transport. Sci., IWCTS 2012 (2012)
12. Parter, S.V.: The use of linear graphs in gaussian elimination. SIAM Review 3(2), 119–130 (1961)
13. Rose, D.J., Tarjan, R.E.: Algorithmic aspects of vertex elimination on directed graphs. SIAM Journal on Applied Mathematics 34(1), 176–197 (1978)
14. Schäffer, A.A.: Optimal node ranking of trees in linear time. Information Processing Letters 33(2), 91–96 (1989)
15. Schreiber, R.: A new implementation of sparse Gaussian elimination. ACM Transactions on Mathematical Software (TOMS) 8(3), 256–276 (1982)
16. Schulz, F., Wagner, D., Zaroliagis, C.D.: Using multi-level graphs for timetable information in railway systems. In: Mount, D.M., Stein, C. (eds.) ALENEX 2002. LNCS, vol. 2409, pp. 43–59. Springer, Heidelberg (2002)
17. Thorup, M.: Compact oracles for reachability and approximate distances in planar digraphs. Journal of the ACM 51(6), 993–1024 (2004)