Lower Bounds and Approximation Algorithms for Search Space Sizes in Contraction Hierarchies

Johannes Blum 🆔
University of Konstanz, Germany
blum@inf.uni-konstanz.de

Sabine Storandt
University of Konstanz, Germany
storandt@inf.uni-konstanz.de

---- **Abstract** ----

Contraction hierarchies (CH) is a prominent preprocessing-based technique that accelerates the computation of shortest paths in road networks by reducing the search space size of a bidirectional Dijkstra run. To explain the practical success of CH, several theoretical upper bounds for the maximum search space size were derived in previous work. For example, it was shown that in minor-closed graph families search space sizes in $\mathcal{O}(\sqrt{n})$ can be achieved (with n denoting the number of nodes in the graph), and search space sizes in $\mathcal{O}(h \log D)$ in graphs of highway dimension h and diameter D. In this paper, we primarily focus on lower bounds. We prove that the average search space size in a so called weak CH is in $\Omega(b_\alpha)$ for $\alpha \geq 2/3$ where b_α is the size of a smallest α-balanced node separator. This discovery allows us to describe the first approximation algorithm for the average search space size. Our new lower bound also shows that the $\mathcal{O}(\sqrt{n})$ bound for minor-closed graph families is tight. Furthermore, we deeper investigate the relationship of CH and the highway dimension and skeleton dimension of the graph, and prove new lower bound and incomparability results. Finally, we discuss how lower bounds for strong CH can be obtained from solving a HittingSet problem defined on a set of carefully chosen subgraphs of the input network.

1 Introduction

The concept of contraction hierarchies (CH) was introduced by Geisberger et al. [15] to accelerate shortest path planning in road networks. The basic idea is to precompute an overlay graph in which the search space size of a bidirectional Dijkstra run can be drastically reduced. For example, on the road network of Western Europe with 18 million nodes, a bidirectional Dijkstra run in the original graph scans almost 5 million nodes on average, while in the CH overlay graph only 280 nodes are scanned. This decreases the query time from over two seconds to about one millisecond [5].

   There exist two CH variants, referred to as strong and weak contraction hierarchies. In a strong CH, the overlay graph construction takes the edge weights directly into account. In a weak CH, the preprocessing phase is split into a metric-independent overlay graph construction phase, and a subsequent customization phase in which the edges are augmented with weights. Weak CH is used in practice to deal with dynamically changing edge weights, as changes in the metric only require to repeat the customization phase but not the overlay graph construction [12]. In [7], the first theoretical upper bounds for search space sizes in weak CH were described, which are valid for strong CH as well. Other lines of research focus on strong CH directly. Graph parameters such as the highway dimension [1] or the skeleton dimension [17] were explictly introduced with the purpose of analyzing search space sizes

of preprocessing-based route planning techniques. However, most existing work focuses on upper bounding the maximum search space. In this paper, we are particularly interested in constructing tight bounds for the average search space size, as those are more expressive for judging whether the CH technique is useful for a given graph (family). We provide a multitude of novel results, including several lower bounding techniques as well as the first approximation algorithm for the average search space size in a weak CH.

1.1 Related Work

In [2, 1], the *highway dimension h* was introduced as a novel graph parameter to capture the shortest path structure of road networks. It was proven that in a strong CH maximum search space sizes in $\mathcal{O}(h \log D)$ can be achieved where D denotes the diameter of the graph. At the same time, the size of the overlay graph is bounded by $\mathcal{O}(n \cdot h \log D)$ where n is the number of nodes in the graph. In [7], the first upper bounds for weak CH were proven. There, *nested dissection* was used to guide the overlay graph construction; a technique that relies on recursive decomposition of the graph into smaller subgraphs by removing balanced node separators. It was shown that in graphs of *treewidth tw* an upper bound of $\mathcal{O}(tw \log n)$ for the maximum search space size S_{max} can be achieved with an overlay graph size of $\mathcal{O}(n \cdot tw \log n)$. For minor-closed graph families that exhibit $\mathcal{O}(\sqrt{n})$ balanced separators, $S_{max} \in \mathcal{O}(\sqrt{n})$ and an overlay graph size in $\mathcal{O}(n \log n)$ was proven. In [14, 8], the *bounded growth model* was proposed as a theoretical framework for studying road networks. There, CH search space sizes in $\mathcal{O}(\sqrt{n} \log n)$ and overlay graph sizes in $\mathcal{O}(n \log D)$ were shown for graphs with unit edge costs that adhere to the model.

Concerning lower bounds, White [22] proved that for any h, D and n, there exists a graph with at least n nodes that has a highway dimension h, a diameter of $\Theta(D)$, and in which the average search space size in a strong CH is in $\Omega(h \log D)$. Therefore, the upper bound by Abraham et al. [1] for strong CH search space sizes parametrized by h is tight. An analogue result was proven for the related route planning technique *hub labeling*. In [20], it was shown on the example of a carefully weighted grid graph that there exists graphs for which the average search space is in $\Omega(\sqrt{n})$ for any strong CH (and for hub labeling as well). In addition, a greedy algorithm was proposed to compute lower bounds for the average search space size of any strong CH in a given weighted graph. In [7], it was observed that the maximum search space size in a weak CH coincides with the so called elimination tree height. As proven by Bodlaender et al. [9], the minimum elimination tree height of a graph equals the *treedepth td*, and the following relations to other graph parameters hold

$$b - 1 \leq tw \leq pw \leq td$$

where b is the *balanced separator number* of the graph, tw is the *treewidth*, and pw the *pathwidth*. Accordingly, the balanced separator number, the treewidth, the pathwidth, and the treedepth are all valid lower bounds for S_{max}. In [6], it was proven by reduction from VertexCover that it is NP-hard to find a contraction order for a strong CH that minimizes the average search space size while adding at most K shortcuts to the graph. Furthermore it was observed that in a complete graph $S_{avg} \in \Omega(n)$ holds, and in path graphs $S_{avg} \in \Omega(\log n)$. For the special case of trees, a linear-time optimal preprocessing algorithm is known that minimizes S_{max} [21]. Complementarily, it was shown in [7] that a CH-graph constructed with a nested dissection contraction order provides a 2-approximation for S_{avg} in trees.

1.2 Contribution

We significantly extend the set of known results on (average) search space sizes in weak and strong contraction hierarchies. Our main findings are listed in the following.

- Although the balanced separator number, the treewidth, the pathwidth and the treedepth are all lower bounds for the maximum search space size S_{max} in a weak CH, we prove that in general none of them is a valid lower bound for S_{avg}. However, we prove that the average search space size S_{avg} in a weak CH is in $\Omega(b_\alpha)$ for $\alpha \geq 2/3$ where b_α is the size of a smallest α-balanced node separator in G.

- We establish the first approximation result for S_{avg} in weak CHs which applies to general graphs. In particular, we prove that a nested dissection CH construction scheme leads to an average search space size within a factor of $1 + 1.5 \log_{1.5} n$ of the optimum. This answers an open question from [7]. We also discuss how to turn the nested dissection approach into an efficient CH construction algorithm on general graphs.

- We furthermore show that for every unweighted graph, one can choose metric edge weights such that the highway dimension h of the graph equals 1, and hence constitutes a trivial lower bound for the average search space size in a strong CH. On such graphs, the algorithms described in [2, 1] with an upper bound on the search space size of $\mathcal{O}(h \log D)$ are close-to-optimal. However, we also show that in case the metric is given, there can be a gap of size $\Omega(n)$ between the highway dimension and the maximum search space size.

- For a given weighted graph, we prove that lower bounds on the average search space size in the respective strong CH can be obtained by solving a (hierarchical) HittingSet problem defined on a set of specific subgraphs of G.

2 Preliminaries

In this section, we describe the concepts of strong and weak CH more formally and provide the definitions of some separator-related graph parameters that will be relevant in our analysis.

2.1 Strong and Weak Contraction Hierarchies

Given an undirected[1], connected graph $G(V, E)$ with edge costs $c : E \to \mathbb{R}^+$, the preprocessing phase for *strong CH* works as follows: First a node permutation is fixed which constitutes the so called contraction order of the nodes. Note that any contraction order is feasible but that the overlay graph size as well as the search space sizes crucially depend on that order. Let $r : V \to \{1, \ldots, n\}$ with $n = |V|$ be the respective ranks of the nodes in the contraction order. Then an overlay graph is constructed with the following property: A shortcut edge $\{v, w\}$ is introduced between v and w if and only if there exists a shortest path π between them that except for v and w only contains nodes u with $r(u) < \min\{r(v), r(w)\}$. The cost of that shortcut edge is set to $c(\pi)$. When constructing a *weak CH* instead, shortcuts are introduced between nodes v and w iff there exists any path between them that except for v and w only contains nodes u with $r(u) < \min\{r(v), r(w)\}$. In either case, the set of shortcuts E^+ is then added to the original graph G to obtain the CH-graph $G^+(V, E \cup E^+)$. For strong CH, this completes the preprocessing phase. For weak CH, a so called customization phase follows in which edge weights are first assigned to the original edges, and are then propagated to the shortcuts. For further details, we refer to [12].

---

[1] With slight modifications, CH also works on directed graphs. To simplify the presentation, we only consider undirected graphs throughout the paper.

In practical implementations, the node contraction order is not always fixed a priori but might be determined in a process which interleaves the overlay graph construction and the node rank assignments. The insertion of shortcut edges is then based on the so called node contraction operation: Here, a node v and its incident edges are removed from the graph and new edges are inserted between certain pairs of its former neighbors. In a strong CH, a shortcut between the neighbors u, w of v is only inserted if the path u, v, w is a shortest path. In a weak CH, shortcuts are inserted between all pairs of neighbors (as without knowledge about the metric, any of those paths could become a shortest path later). The preprocessing then consists of contracting all nodes in the graph one after the other, using e.g. the current degree of the nodes as guidance which of them to contract next (or more complicated criteria, see [15]). But independently of the chosen implementation, the formal characterization of the resulting overlay CH-graph given above always applies.

To compute a shortest path between nodes $s, t \in V$ in the CH-graph G^+, a bidirectional Dijkstra run is started from s and t in G^+ with the restriction that from any node v incident edges $\{v, w\}$ are only relaxed if $r(w) > r(v)$. The set of nodes that can be reached from a node $v \in V$ via paths which go strictly upwards with respect to the contraction order is called the search space $SS(v)$. By construction of G^+ one can show that there always exists a node $p \in SS(s) \cap SS(t)$ which lies on a shortest path between s and t in G, and the shortest path distance from s to p as well as the shortest path distance from p to t are correctly computed in G^+. Accordingly, the computed shortest path distance in G^+ equals the shortest path distance in G.

Note that in general there is a difference between the search space size and the query time. The search space size $|SS(v)|$ of a node v is the number of nodes settled in the Dijkstra run from v in the CH-graph while the query time also accounts for edge relaxation and priority queue operations. However, an upper bound U on the search space size also yields an upper bound of U^2 on the query time as there can be at most a quadratic number of edges between U nodes. The other way around, a lower bound on the search space size is automatically also a lower bound for the query time. Therefore, we stick to the notion of search space sizes throughout the paper. We distinguish between the maximum CH search space size $S_{max} = \max_{v \in V} |SS(v)|$ and the average CH search space size $S_{avg} = \frac{1}{n} \sum_{v \in V} |SS(v)| \leq S_{max}$.

2.2 Balanced Separators

Existing upper bounds for search space sizes in weak CH were obtained by using nested dissection based overlay graph construction, which involves the recursive computation of balanced node separators. We will investigate which separator-related graph parameters can be used to lower bound the average search space size. The *balanced separator number* is defined as the smallest integer b such that for every $V' \subseteq V$, the induced subgraph $G[V']$ admits a balanced separator of size at most b, i.e. after the removal of at most b nodes every remaining connected component of $G[V']$ contains at most $\lceil (|V'| - b)/2 \rceil$ nodes. While $b - 1$ is known to lower bound the maximum CH search space size in any given graph, we will show that the same is not true for S_{avg}. However, we will prove a lower bound of $\alpha \cdot b_\alpha$ for $\alpha \geq 2/3$ which later will be used to get an approximation guarantee for S_{avg} when constructing the CH with a variant of nested dissection. The parameter b_α, for an $\alpha \in [0, 1]$, is the size of a minimum α-balanced node separator of $G(V, E)$, i.e., the smallest number of nodes that have to be removed from the input graph $G(V, E)$ such that all remaining connected components have size at most $\alpha \cdot |V|$. Note that the value of b_α is solely determined by considering the input graph as a whole, while for b all induced subgraphs are relevant as well. By definition,

the following hierarchy holds: $b_{2/3} \leq b_{1/2} \leq b$.

3 Bounds and Algorithms for Search Space Sizes in Weak CH

In this section, we discuss at the beginning how S_{avg}, S_{max}, and the balanced separator number relate to each other in a weak CH. Subsequently, we describe new lower bounding techniques and present the first approximation algorithm for S_{avg}.

3.1 Maximum versus Average Search Space Size

As discussed above, the smallest possible maximum search space size of a weak CH is equal to the treedepth of the input graph. However, there can be a large gap between treedepth and average search space size. In fact, not even the balanced separator number b yields a lower bound for S_{avg} as the following lemma shows.

▶ **Lemma 1.** *There exist graphs* $G(V, E)$ *with* $S_{avg} \in o(b)$.

Proof. Take a star graph with k leaves and replace one leaf with a clique C of size $\sqrt{k}$. Every balanced separator of the subgraph induced by C contains $\Omega(\sqrt{k})$ vertices, which implies that $b \in \Omega(\sqrt{k})$. Consider a contraction order where the central node obtains rank $n = |V|$. Then the search space size of the central node is 1, the search space size of every leaf is 2 and the search space size of a node in the clique is at most $\sqrt{k} + 1$. It follows that the average search space size is

$$S_{avg} \leq \frac{1 + (k-1) \cdot 2 + \sqrt{k} \cdot (\sqrt{k} + 1)}{n} = \frac{3k + \sqrt{k} - 1}{n} = \frac{3k + \sqrt{k} - 1}{k + \sqrt{k}} < 3 \in o(\sqrt{k}). \quad \blacktriangleleft$$

With $b - 1 \leq tw \leq pw \leq td$, the lemma implies that none of those parameters is a valid lower bound for S_{avg}; and that there can be an exponential gap between the average and the maximum search space size.

3.2 Balanced Separators and Average Search Space Size

In a weak CH, a node $w \in V$ of rank R is in the search space $SS(v)$ of another node $v \in V$ with rank $r < R$ if there exists a path from v to w which – except for w – exclusively contains nodes of rank $< R$. This fact leads to the following central observation.

▶ **Observation 2.** *Let* $G(V, E)$ *be a connected graph. In a weak CH on* G, *for a node* $w \in V$ *of rank* R *to* not *be in the search space* $SS(v)$ *of a node* $v \in V$ *of rank* $r < R$, *there needs to be a set of nodes of rank higher than* R *that separates* w *from* v.

We will now present our first main theorem, which exploits this observation to show a general relationship between the average serach space size S_{avg} in a weak CH and balanced node separators in G.

▶ **Theorem 3.** *For any weak CH and any* $\alpha \geq \frac{2}{3}$, *at least* αn *nodes in* G *have a search space of size at least* b_α.

Proof. Consider some contraction order of the nodes. We identify the smallest integer k such that the k nodes with highest rank in the given order form an α-balanced node separator S in G. Hence, if we remove the $k - 1$ nodes of highest rank, there is still a connected component C of size $> \alpha n$. If we now remove the node v_k with kth highest rank, we can

split C into two subgraphs of size at most αn each, which are not connected to each other. We observe that the larger of the two subgraphs C^* (which is not necessarily connected) has to contain at least $\frac{\alpha}{2}n$ nodes. Therefore, there are at most $n - \frac{\alpha}{2}n$ nodes that are not contained in C^*, which for any $\alpha \geq \frac{2}{3}$ is at most αn. It follows that the separator nodes $S^* \subseteq S$ that are adjacent to C^* form an α-balanced separator already. By definition of b_α, we have $|S^*| \geq b_\alpha$. It remains to be shown that between any node v in $C \setminus v_k$ (which has rank $r(v) < n - k$) and any separator node $s \in S^*$ (which has rank $r(s) \geq n - k$) there exists a path in G on which all nodes have rank at most $r(s)$. For $s = v_k$ this is true, because C is connected and v_k has maximum rank among all nodes of C. If we have $s \neq v_k$, the node s has a neighbor w contained in C. As C is connected, there is a path from v to w in C and hence, the maximum rank on this path is at most $n - k < r(s)$. It follows that G contains a path from v to s of maximum rank $r(s)$. This means that there are αn nodes (recall that C has size $> \alpha n$), which have a search space size of at least $|S^*| \geq b_\alpha$. ◀

It follows that for any $\alpha \geq \frac{2}{3}$ the sum of the search space sizes of all nodes in G is at least $\alpha n \cdot b_\alpha$ and hence the average search space size is lower bounded by $\alpha \cdot b_\alpha$.

▶ **Corollary 4.** *In any weak CH graph, $S_{avg} \geq \alpha \cdot b_\alpha$ for $\alpha \in \left[\frac{2}{3}, 1\right)$.*

The theorem implies that weak CH performs poorly on graph families with no small balanced separators. For the class of planar graphs, and $\alpha = \frac{2}{3}$, there exist graphs with a smallest α-balanced separator of size at least $1.55\sqrt{n}$ [13]. Hence, according to Corollary 4, their induced average CH search spaces are in $\frac{2}{3}1.55\sqrt{n} = 1.0\overline{3}\sqrt{n} \in \Omega(\sqrt{n})$. This matches the known upper bounds for S_{max} in minor-closed graph families with $\mathcal{O}(\sqrt{n})$ sized balanced separators, proving them to be tight.

We prove next that $S_{avg} \in \Omega(b_{1/2})$ holds as well which will later be important for establishing our approximation guarantee. Note that if we just consider balanced separators in G directly, there could be an arbitrary large gap between $b_{1/2}$ and $b_{2/3}$. Hence, we will consider separators in G and in a selected subgraph of G simultaneously to get a meaningful bound. We first show a helping lemma which might also be of independent interest.

▶ **Lemma 5.** *Let G' be a connected subgraph of G with n' nodes and let b'_α be the size of a smallest α-balanced separator in G' for an $\alpha \geq \frac{2}{3}$, then it follows that $S_{avg} \geq \frac{n'}{n}\alpha b'_\alpha$.*

Proof. The proof is the same as the proof of Theorem 3 with the only modification that we consider the k nodes with highest rank in G' that form an α-balanced separator in G' instead of considering G as a whole. ◀

The lemma improves the lower bound on S_{avg} shown in Theorem 6 in case a subgraph of G has a larger balanced separator than G as a whole and this subgraph G' is not too small compared to the total size of G. And it also allows us to prove a general relationship between S_{avg} and $b_{1/2}$ as manifested in the following theorem.

▶ **Theorem 6.** *The average search space size in a weak CH is lower bounded by $\frac{2}{9}b_{1/2}$.*

Proof. Let $b_{1/2}$ and $b_{2/3}$ be minimum balanced separator sizes in G for the respective α-values. Obviously, $b_{2/3} \leq b_{1/2}$ holds. Let G^* be the larger of the two parts that results from removing the $b_{2/3}$ nodes in the respective separator from G. The smaller part has to contain less then $\frac{1}{2}n$ nodes; but G^* may contain between $\frac{1}{3}n$ and $\frac{2}{3}n$ nodes. Now let $b'_{2/3}$ be the minimum size of a 2/3-balanced separator in G^*. The largest part of G^* after the removal of such a separator has size at most $\frac{2}{3} \cdot \frac{2}{3}n = \frac{4}{9}n < \frac{1}{2}n$. Accordingly, the union of the nodes in the 2/3-balanced separator in G and the 2/3-balanced separator in G^* form a 1/2-balanced separator in G. It

follows that $b_{1/2} \le b_{2/3} + b'_{2/3} \le 2\max(b_{2/3}, b'_{2/3})$. Using Lemma 5 together with the relation between the separators in G and G^* leads to the following set of inequalities that lower bound the average search space size: $S_{avg} \ge \max(\frac{2}{3}b_{2/3}, \frac{2}{3} \cdot \frac{2}{3}b'_{2/3}) \ge \frac{4}{9}\max(b_{2/3}, b'_{2/3}) \ge \frac{2}{9}b_{1/2}$ ◄

3.3 An Approximation Algorithm for the Average Search Space Size

The crucial part of CH construction is fixing the contraction order of the nodes. When using nested dissection, as proposed in [7], the input graph G is partitioned recursively into smaller subgraphs via a balanced separator decomposition, and for each obtained subgraph the nodes in the separator are contracted before the other nodes. It was proven that this contraction order allows to upper bound the resulting maximum search space size S_{max}^{ND}. More precisely, an (α, f_α)-balanced separator decomposition of $G(V, E)$ for $\alpha \in (0, 1)$ is a rooted tree T whose nodes are disjoint subsets of V and that is recursively defined as follows. If $n = 1$, then T consists of a single node $X = V$. If $n > 1$, then the root of T is a set $X \subseteq V$ of size at most $f_\alpha(n)$ whose removal separates G into $d \ge 2$ subgraphs $G_1, \ldots, G_d$ with at most αn vertices, each. Moreover, the children of X are the roots of (α, f_α)-balanced separator decompositions of $G_1, \ldots, G_d$. A nested dissection order can then be obtained via a post-order traversal of such a (α, f_α)-balanced decomposition (nodes within one separator X can be contracted in an arbitrary order). It follows that $S_{max}^{ND} \le (1 + \log_{1/\alpha} n) \cdot B$ where B is an upper bound on the separator sizes f_α in all subgraphs. As every graph has a $(1/2, b)$-balanced separator decomposition where b is the balanced separator number of G, it follows that $S_{max}^{ND} \in \mathcal{O}(b \log n)$. Combined with $S_{max}^{ND} \ge S_{max}^* = td > b$, where S_{max}^* is the smallest possible value of S_{max}, this proves an $\mathcal{O}(\log n)$ approximation guarantee for the maximum search space size[2].

For trees, a $(1/2, 1)$-balanced separator decomposition exists. Accordingly, $S_{max}^{ND} \in \mathcal{O}(\log n)$ holds for trees. Furthermore, it was proven in [7] that the nested dissection contraction order for trees also is an approximation algorithm for S_{avg}. In particular, $S_{avg}^{ND} \le 2 \cdot S_{avg}^*$ was shown where S_{avg}^* is the optimal average search space size. It was posed as one of the main open questions whether there are ways to approximate the average search space size in general graphs. Note that the approximation guarantee proof given for trees explicitly leverages the fact that trees are cycle free, and that balanced separators in trees always consist of a single node. Hence new proof concepts are required for generalizing the result.

Exploiting our novel lower bound shown in Corollary 4, we will now prove that for general graphs, the contraction order induced by a (α, b_α)-balanced decomposition indeed comes with an approximation guarantee for S_{avg}. Our proof consists of the following three steps:

1. *Upper bounding $S_{avg}^{ND}(G)$.* We show that $S_{avg}^{ND}(G) \le \frac{1}{n}\sum_{i=1}^{d} n_i \cdot S_{avg}^{ND}(G_i) + b_\alpha$ where b_α is the size of an α-balanced separator B in G, whose removal splits G into the connected subgraphs $G_1, \ldots, G_d$ of sizes $n_1, \ldots, n_d$ (cf. Lemma 7).

2. *Lower bounding $S_{avg}^*(G)$.* We prove that for the optimal average search space size S_{avg}^* it holds that $S_{avg}^*(G) \ge \frac{1}{n}\sum_{i=1}^{d} n_i \cdot S_{avg}^*(G_i)$ (cf. Lemma 8).

3. *Combining upper and lower bounds.* Combining the outcomes of steps 1 and 2 as well as Corollary 4, we show that with every level of the separator decomposition, the ratio between S_{avg}^{ND} and S_{avg}^* increases at most by $\frac{1}{\alpha}$. This results in an overall approximation factor of $1 + \frac{1}{\alpha}\log_{1/\alpha}$ for $\alpha \in [\frac{2}{3}, 1)$ (cf. Theorem 9).

---

[2] This result does not hold in directed graphs as shown in [11].

▶ **Lemma 7.** *Given a graph G, it yields $S_{avg}^{ND}(G) \leq \frac{1}{n} \sum_{i=1}^{d} n_i \cdot S_{avg}^{ND}(G_i) + b_\alpha$ where b_α is the size of a given α-balanced separator B in G, and $G_1, \ldots, G_d$ are the connected subgraphs of G that remain after removing B, with $n_1, \ldots, n_d$ being their respective node set sizes.*

Proof. The average search space size S_{avg} is defined as the sum of the individual search space sizes divided by n. For each node in the separator B, the search space size can be at most $|B| = b_\alpha$ as the nodes in B all have higher rank in the contraction order than the nodes in $V \setminus B$. For $i = 1, \ldots, d$, the n_i nodes of G_i have a total search space sizes of $n_i \cdot S_{avg}^{ND}(G_i)$ if we do not count the nodes from B. The search space size of each individual node increases by at most b_α when adding B to the graph. Hence in total we get $n \cdot S_{avg}^{ND} \leq \sum_{i=1}^{d} n_i \cdot S_{avg}^{ND}(G_i) + n \cdot b_\alpha$. Dividing both sides by n concludes the proof. ◄

▶ **Lemma 8.** *Given a graph G, the optimal average search space size in a weak CH is lower bounded by $S_{avg}^*(G) \geq \frac{1}{n} \sum_{i=1}^{d} n_i \cdot S_{avg}^*(G_i)$ where G_i are disjoint subgraphs of G.*

Proof. Let r^* be the optimal contraction order for G, leading to an average search space size of $S_{avg}^*(G) = \frac{1}{n} \sum_{v \in V} |SS^*(v)|$. By r_i^* we denote the restriction of r^* to subgraph G_i, that is, the nodes in G_i are sorted by their r^* values and $r_i^* : V(G_i) \to \{1, \ldots, n_i\}$ then assigns each node $v \in V(G_i)$ the rank of v in the obtained order. Let $v \in V(G_i)$ and denote by $SS'(v)$ the search space of v in G_i when using contraction order r_i^*. We show that $|SS^*(v) \cap V(G_i)| \geq |SS'(v)|$, implying that the part of a search space of a node v that intersects a certain subgraph is as least as large as the search space of v in that subgraph when using the globally optimal contraction order restricted to that subgraph. For proving this property, consider a node $w \in SS'(v)$. As w is contained in this search space, there exists a path from v to w in G_i with all nodes on the path having a rank of at most $r_i^*(w)$. This path then also exists in G and as for nodes x, y with $r^*(x) > r^*(y)$, we have $r_i^*(x) > r_i^*(y)$ by construction, it follows that $w \in SS^*(v) \cap V(G_i)$ holds as well. Therefore, $SS^*(v) \cap V(G_i) \supseteq SS'(v)$ applies. Plugging the resulting size inequality into the definition of the average search space results in the following lower bound:

$$n \cdot S_{avg}^*(G) = \sum_{v \in V} |SS^*(v)| \geq \sum_{i=1}^{d} \sum_{v \in V(G_i)} |SS^*(v)| \geq \sum_{i=1}^{d} \sum_{v \in V(G_i)} |SS^*(v) \cap V(G_i)|$$
$$\geq \sum_{i=1}^{d} \sum_{v \in V(G_i)} |SS'(v)|$$

As the application of any contraction order to G_i results in summed search space sizes that are at least as large as $n_i \cdot S_{avg}^*(G_i)$ where $S_{avg}^*(G_i)$ is the optimal average search space in G_i, we get $S_{avg}^*(G) \geq \frac{1}{n} \sum_{i=1}^{d} \sum_{v \in V(G_i)} |SS'(v)| \geq \frac{1}{n} \sum_{i=1}^{d} n_i \cdot S_{avg}^*(G_i)$. ◄

▶ **Theorem 9.** *Given a graph G, nested dissection contraction results in search space sizes $S_{avg}^{ND} \leq (1 + \frac{1}{\alpha} \log_{1/\alpha} n) \cdot S_{avg}^*$ for any $\alpha \geq \frac{2}{3}$, if optimal separators are used.*

Proof. Consider the (α, b_α)-balanced separator decomposition which induces the nested dissection order. For every leaf X in the decomposition we have $S_{avg}^{ND}(X) = S_{avg}^*(X)$ as X contains only one vertex. Consider now some non-leaf node X from the decomposition. Let H be the subgraph of G induced by X and its descendants in the separator decomposition and denote the connected components of $H \setminus X$ by $H_1, \ldots, H_d$. Denote the size of H_i by n_i and assume that for the average search spaces of $H_1, \ldots, H_d$ we have an approximation

factor of γ, i.e. $S_{avg}^{ND}(H_i) \leq \gamma \cdot S_{avg}^*(H_i)$. Lemma 7 implies

$$S_{avg}^{ND}(H) \leq \frac{1}{n} \sum_{i=1}^{d} n_i \cdot S_{avg}^{ND}(H_i) + |X| \leq \gamma \cdot \frac{1}{n} \sum_{i=1}^{d} n_i \cdot S_{avg}^*(H_i) + |X|$$

Moreover, X is an optimal α-balanced separator, so Corollary 4 implies that $S_{avg}^*(H) \geq \alpha \cdot |X|$, which can be rearranged to $|X| \leq \frac{1}{\alpha} S_{avg}^*(H)$. In combination with Lemma 8 we obtain

$$\gamma \cdot \frac{1}{n} \sum_{i=1}^{d} n_i \cdot S_{avg}^*(H_i) + |X| \leq \gamma \cdot S_{avg}^*(H) + \frac{1}{\alpha} \cdot S_{avg}^*(H) \leq (\gamma + \frac{1}{\alpha}) \cdot S_{avg}^*(H)$$

As for every leaf X we have $S_{avg}^{ND}(X) = S_{avg}^*(X)$ and the height of the separator decomposition is $\log_{1/\alpha} n$, it follows by induction that $S_{avg}^{ND}(G) \leq (1 + \frac{1}{\alpha} \log_{1/\alpha} n) \cdot S_{avg}^*(G)$. ◄

▶ **Corollary 10.** *The average search space size in a weak CH using nested dissection based on recursive decomposition with $b_{2/3}$ is at most $1 + 1.5 \log_{1.5} n$ times the optimal size.*

However, as computing optimal balanced separators is NP-hard in general, nested dissection does not directly yield a polynomial time approximation algorithm. But we can exploit the existence of a pseudo-approximation for balanced separators. In [18] it was proven that one can find a 3/4-balanced separator that has size at most $\mathcal{O}(\log n) \cdot b_{2/3}$ in polynomial time[3]. In general, if we have a γ-approximation for $b_{2/3}$, Lemma 7 can be modified to show that $S_{avg}^{ND} \leq \sum_{i=1}^{d} n_i \cdot S_{avg}^{ND}(G_i) + \gamma b_{2/3}$ which plugged into Theorem 9 results in an average search space size of $S_{avg}^{ND} \leq (1 + 1.5\gamma \log_{1.5} n) S_{avg}^*$. Using the pseudo-approximation result, we have $\gamma = \mathcal{O}(\log n)$, and additionally the depth of the recursion increases from $\log_{3/2} n$ to $\log_{4/3} n$. We call the nested dissection based contraction algorithm which leverages the pseudo-approximation algorithm to compute the node separators the pND-algorithm. Combining the aforementioned observations, we get the following theorem.

▶ **Theorem 11.** *Given a graph $G(V,E)$, then a weak CH obtained from the pND-algorithm in polynomial time leads to an average search space size of $S_{avg}^{pND}(G)$ with $S_{avg}^{pND}(G) \leq \mathcal{O}(\log^2 n) S_{avg}^*(G)$, where $S_{avg}^*(G)$ denotes the minimum average search space size.*

Note that for some graph classes better approximation algorithms for balanced node separators exist, which then can be plugged into our analysis to achieve tighter overall results. For example, for any graph class for which optimal balanced separators can be approximated by some constant factor $c > 0$ in polynomial time (as it is the case e.g., for planar graphs [4]), we achieve an overall approximation ratio of $\mathcal{O}(\log n)$ for S_{avg}. For the special case of $\sqrt{n} \times \sqrt{n}$ rectangular grids, the results from [7] imply that nested dissection leads to search space sizes of at most $3\sqrt{n}$. This matches our lower bound of $\alpha \cdot b_\alpha$ (translating for grids to $\frac{2}{3}\sqrt{n}$) up to a factor of 4.5. As the respective separators can be found efficiently in grids, this implies that nested dissection yields a constant factor approximation algorithm for this graph class.

Regarding space consumption, it was argued in [7] that a weak CH-graph constructed by nested dissection contains at most $n \cdot S_{avg}$ shortcut edges. However, this bound is rather loose on some graphs. For example, it is known for planar graphs that contraction orders exist such that the CH-graph size is in $\mathcal{O}(n \log n)$ while e.g., for grids we know by

---

[3] This is called a pseudo-approximation or bicriteria approximation as the approximation factor compares the output for a relaxed problem version to the optimum of the unrelaxed version.

Corollary 4 that $n \cdot S_{avg}$ amounts to $\Omega(n\sqrt{n})$. In [3], it was proven that nested dissection indeed approximates the number of shortcut edges within a factor of $\mathcal{O}(\sqrt{d}\log^4 n)$ where d is the maximum degree of the input graph. To achieve polynomial running time, the nested dissection variant described there relies on a pseudo-approximation for $1/2$-balanced separators. As proven in Theorem 6, the average search space size is also lower bounded by $2/9 \cdot b_{1/2}$. This can be plugged in our approximation algorithm analysis to also get an $\mathcal{O}(\log^2 n)$ overall approximation factor for S_{avg}. For road networks, where the maximum node degree is a small constant, it therefore yields that a polynomial time variant of nested dissection approximates the three important aspects of a CH-graph – space consumption (number of inserted shortcuts), maximum search space size, and average search space size – all by polylogarithmic factors.

4   Relation to Road Network Dimensions

So far, we considered search space sizes in weak CHs. Now we shift our focus to strong CH. In particular, we now study the relationship between the average search space size in a strong CH and the highway dimension h as well as the skeleton dimension k of the graph. Those two parameters were both previously used to show upper bounds on the search space size of preprocessing-based route planning techniques.

4.1   Highway Dimension Lower Bound

The highway dimension of a weighted graph $G(V, E)$ is defined as follows [1]: For $u \in V$ and $r > 0$, the ball $B_r(u)$ consists of all nodes at distance at most r from u. Consider now the set $\mathcal{P}$ of all shortest paths longer than r that are contained within the ball $B_{4r}(u)$ and let $H_r^u \subseteq V$ be a minimum hitting set for $\mathcal{P}$. The highway dimension of G is defined as $h = \max_{u,r} |H_r^u|$. For a graph with highway dimension h, $S_{max} \in \mathcal{O}(h \log D)$ was proven for strong CH [1]. Furthermore, it was shown in [7] that there exist graphs and edge weights such that $h \in \Omega(pw/\log n)$ holds. We now establish a complementary result where the aim is to find edge weights such that the value of h is as small as possible.

▶ **Lemma 12.** *For any unweighted graph G there are metric edge weights such that the highway dimension of G is 1.*

Proof. Let $V = \{1, \ldots, n\}$ and choose the weight of an edge $\{u, v\}$ as $9^{\max\{u,v\}}$. To bound the highway dimension consider some ball $B_{4r}(v)$ and choose the largest j such that $9^j \le 4r$. If $v > j$, any edge incident to v has length at least $9^v \ge 9^{j+1} > 4r$. This means that $B_{4r}(v) = \{v\}$ and the ball contains no shortest path that needs to be hit. Let now $v \le j$. Consider some shortest path π that passes only through vertices $i < j$. The path π is simple and hence by the choice of the edge weights it contains no three edges of same length. This means that we can bound the length of π by $\sum_{i=j/2}^{j-1} 2 \cdot 9^i < 2 \cdot \sum_{i=0}^{j-1} 9^i = (9^j - 1)/4 \le r$. Moreover, any $u > j$ has distance $9^u \ge 9^{j+1} > 4r$ from v and hence no shortest path passing through u is contained in $B_{4r}(v)$. This means that every relevant shortest path contains j and can be hit by the set $\{j\}$.   ◀

The lemma shows that the known upper bound on the maximum search space size of $\mathcal{O}(h \log D)$ is almost tight when the metric is chosen appropriately. White [22] showed for a special family of graphs called $G_{t,k,q}$ graphs (introduced in [19]) that the average search space size is in $\Omega(h \log D)$, which matches the upper bound asymptotically. His proof strongly relies on the characteristics of that graph family, though, and especially on the graph topology.

Our lower bound, however, is independent of the graph structure but just considers the metric. Our result also generalizes to the hub labeling technique where a matching upper bound on the maximum number of hubs per node was proven in [1].

4.2 General Incomparability

While we showed above that we can always choose edge weights such that the highway dimension lower bounds the average search space size, this is not necessarily true if the edge weights are given. In fact, we will prove that in general, it is not possible to lower bound the maximum (or average) CH search space size in terms of the highway dimension or the skeleton dimension. For the skeleton dimension k, no upper bounds for CH are known so far, but for the related technique of hub labeling [17], upper bounds of $\mathcal{O}(k \log n)$ were shown.

Intuitively, the skeleton dimension measures how many "important" branches every shortest path tree contains. For a concise definition we refer to [17]. Our incomparability result just exploits the fact that the maximum degree is a lower bound for the skeleton dimension.

▶ **Lemma 13.** *There are graphs with skeleton dimension k and highway dimension h such that $S_{\max} \in o(k)$ and $S_{\max} \in o(h)$.*

Proof. Take a star graph, subdivide every edge by inserting one vertex and assume unit edge weights. The maximum degree Δ of the graph is linear in the number of nodes n and as Δ is a lower bound for the skeleton dimension k, it follows that $k \in \Omega(n)$. Moreover half of the edges are incident to a leaf node. All these edges are pairwise disjoint and every such edge forms a shortest path of length $1 =: 2r > r$, which intersects the ball of radius $4r = 2$ around the central vertex of the graph. This means that the highway dimension is in $\Omega(n)$.

Consider a contraction order where the highest rank is assigned to the central vertex of the graph and the lowest ranks are assigned to the leaves. Then the maximum search space size is $3 \in o(n)$, which is assumed in the leaves. ◀

Accordingly, search space sizes might be significantly smaller than the discussed road network dimensions. Therefore, the upper bounds derived in dependency of those parameters might be very loose on some graphs. This motivates further research into finding other graph parameters where the possible gap between the average/maximum search space size and the parameter value is in $o(n)$; and to investigate these gaps on real-world networks.

5 Lower Bounds for Strong CH

In this section, we present algorithms to obtain lower bounds for the average search space size of a strong CH in a given weighted graph, e.g., to judge how large the gaps to the road network dimensions are, or to investigate whether a contraction order used in practice produces search space sizes close to the optimum.

5.1 HittingSet Lower Bound

In a strong CH, a node w of rank R is in the search space of a node v with rank $r < R$, if on the shortest path between v and w there is no node with a rank higher than R. (Note that this is a sufficient but not necessary condition.) Accordingly, if we consider the node with maximum rank in the contraction order, we know that it has to be contained in the search space of each node in G. This leads us to the definition of the inverse search space of a node v as $ISS(v) := \{w \in V | v \in SS(w)\}$ which can be used as an alternative mean to determine the average search space $S_{avg} = \frac{1}{n} \sum_{v \in V} |SS(v)| = \frac{1}{n} \sum_{v \in V} |ISS(v)|$. For the node v_{max}

with the highest rank, we know that $|ISS(v_{max})| = n$. Now the goal is to show large inverse search spaces also for other nodes of sufficiently high rank. Note however, that the inverse search space sizes are not necessarily proportional to the rank, as nodes of even higher rank might block many shortest paths. Therefore, we have to take the topology of the graph as well as the shortest path structure into account.

▶ **Observation 14.** *Let G' be the subgraph of G induced by a node set $V' \subseteq V$ and let $v'_{max} \in V'$ be the node with the highest rank in the contraction order among all nodes in V'. Then the number of nodes in V' with their shortest path towards v'_{max} being completely contained in G' is a lower bound for $|ISS(v'_{max})|$.*

To get a lower bound which adheres to all possible contraction orders we need to deal with the fact that we do not know which node in V' will become v'_{max}. To make the bound more general, we can iterate through all nodes $v \in V'$ and compute the number of nodes in V' with the shortest path towards v being completely contained in G', keeping track of the minimum. We call this value the minimum shortest path tree size in G' or $misp(G')$.

▶ **Theorem 15.** *For any $\beta \in (0, 1]$, let $\mathcal{G}$ be a collection of subgraphs of G with $misp(G') \geq \beta n$ for all $G' \in \mathcal{G}$. Furthermore, let H be a minimum HittingSet for the set system $(V, \mathcal{G})$. Then it yields $S_{avg} \geq \beta |H|$.*

Proof. We want to count the inverse search space sizes induced by the nodes of highest rank within each subgraph. To that end, we interpret H as the set of nodes contracted last. We know by definition of $\mathcal{G}$ and minimality of H that for every node $h \in H$, we have $|ISS(h)| \geq \beta n$. Furthermore, by H being a minimum HittingSet, we conclude that there can't be fewer than $|H|$ nodes with that property. Accordingly, we get $S_{avg} = \frac{1}{n} \sum_{v \in V} |ISS(v)| \geq \frac{1}{n} \sum_{h \in H} |ISS(h)| \geq \frac{1}{n} \beta n |H| = \beta |H|$ which completes the proof.   ◀

▶ **Example 16.** To illustrate the usefulness of the HittingSet lower bound, we consider a complete graph G with n nodes and metric edge weights, and we choose $\beta = \frac{1}{2}$. We observe that any induced subgraph G' of G of size $\geq \frac{n}{2}$ has $misp(G') \geq \frac{n}{2}$ as well. It follows that we need a HittingSet H of size $\frac{n}{2} + 1$ to hit all these subgraphs. According to Theorem 15, we hence get $S_{avg} \geq \frac{n}{4}$ which matches the upper bound on S_{max} of $n - 1$ asymptotically.

For practical exploitation in real networks, we remark that for every shortest path π of length βn in G, we automatically have $misp(\pi) = \beta n$ and that for any selected subgraph the misp-value can be computed in polytime.

5.2   Hierarchical Lower Bound

In general graphs, the question arises how to choose β in practice to get the best bound. For large β the size of H is expected to be small, while gains in the HittingSet size for small values β might be diminished by the multiplication with β itself. To get a useful lower bound nevertheless, we propose a hierarchical scheme based on the following observation.

▶ **Observation 17.** *Let $\beta_1, \beta_2 \in (0, 1]$ be two parameters with $\beta_1 > \beta_2$ and $\mathcal{G}_1$, H_1 as well as $\mathcal{G}_2$, H_2 the respective subgraph collections and minimum HittingSets. Then we have $S_{avg} \geq \beta_1 |H_1| + \beta_2 (|H_2| - |H_1|)$.*

This observation can be generalized to an arbitrarily fine-grained succession of β-values.

▶ **Corollary 18.** *Given a weighted graph $G(V, E)$, as well as $\beta_1 > \beta_2 > \cdots > \beta_k$ with $\beta_i \in (0, 1]$, let H_i be the respective HittingSet sizes for all subgraphs G' of G with $misp(G') \geq \beta_i n$. Then the average search space size S_{avg} is lower bounded by $\beta_1 |H_1| + \sum_{i=2}^{k} \beta_i (|H_i| - |H_{i-1}|)$.*

▶ **Example 19.** To illustrate that the hierarchical scheme can be beneficial, we consider a path graph with $n = 2^q$ nodes. For any choice of β, we know that a trivial HittingSet of size $\frac{1}{\beta}$ exists. Therefore the bound we get from Theorem 15 only amounts to $S_{avg} = \beta \cdot \frac{1}{\beta} = 1$. If we now use Corollary 18 with $\beta_i = 2^{-i}$ for $i = 0, \ldots, \log_2 n$, we get

$$S_{avg} \geq 1 + \sum_{i=1}^{q} 2^{-i}(2^i - 2^{i-1}) = 1 + \sum_{i=1}^{q} \tfrac{1}{2} = \tfrac{1}{2} \log_2 n$$

which matches the known upper bound up to constant factors.

6   Conclusions and Future Work

We described several novel lower bounding techniques for average and maximum search space sizes in contraction hierarchies. Lower bounds are an important mean to judge the quality of existing construction schemes and theoretical upper bounds. While for hub labeling (a preprocessing-based route planning technique closely related to contraction hierarchies), approximation algorithms for the average number of nodes that have to be scanned in a query were known for some time [10], we proved the first general approximation result for the average search space size in weak contraction hierarchies. As the construction of weak contraction hierarchies is closely related to graph triangulation and to solving systems of linear equations [3], there might be cross-implications to explore.

Future work could also consider weak contraction hierarchies on directed graphs. As the graph parameters used in our analysis are classically defined on undirected graphs only, the established relationships to CH search space sizes do not automatically transfer to directed graphs. One possibility would be to consider parameter variants, as the directed treewidth [16], which are explicitly defined on directed graphs and to check whether approximation results can be obtained in dependency of those parameters.

Another interesting open question is whether there are efficient approximation algorithms for the average or maximum search space size in strong CHs. While upper bounds for search space sizes in weak CHs are valid for strong CHs as well, it is the other way around for lower bounds. Therefore, strong CHs are not covered by our approximation results for weak CHs. Our lower bounds for average search space sizes in strong CHs rely on HittingSet computation, though, as do some strong CH construction schemes which come with provable upper bounds [1]. It hence might be possible to link those results.

Finally, it might be interesting to experimentally investigate the strength of our lower bounds and the efficiency of the proposed algorithms on real-world (road) networks.

──── **References** ────

1   Ittai Abraham, Daniel Delling, Amos Fiat, Andrew V. Goldberg, and Renato F. Werneck. Highway dimension and provably efficient shortest path algorithms. Technical Report MSR-TR-2013-91, Microsoft Research, September 2013.

2   Ittai Abraham, Amos Fiat, Andrew V. Goldberg, and Renato Fonseca F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 782–793, 2010.

3   Ajit Agrawal, Philip Klein, and R Ravi. Cutting down on fill using nested dissection: provably good elimination orderings. In *Graph Theory and Sparse Matrix Computation*, pages 31–55. Springer, 1993.

4   Eyal Amir, Robert Krauthgamer, and Satish Rao. Constant factor approximation of vertex-cuts in planar graphs. In *ACM Symposium on Theory of Computing (STOC)*, pages 90–99, 2003.

5   Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. Route planning in transportation networks. In *Algorithm engineering*, pages 19–80. Springer, 2016.

6   Reinhard Bauer, Tobias Columbus, Bastian Katz, Marcus Krug, and Dorothea Wagner. Preprocessing speed-up techniques is hard. In *International Conference on Algorithms and Complexity (CIAC)*, pages 359–370. Springer, 2010.

7   Reinhard Bauer, Tobias Columbus, Ignaz Rutter, and Dorothea Wagner. Search-space size in contraction hierarchies. *Theoretical Computer Science*, 645:112–127, 2016.

8   Johannes Blum, Stefan Funke, and Sabine Storandt. Sublinear search spaces for shortest path planning in grid and road networks. In *AAAI Conference on Artificial Intelligence*, 2018.

9   Hans L. Bodlaender, John R. Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995. `doi:10.1006/jagm.1995.1009`.

10   Edith Cohen, Eran Halperin, Haim Kaplan, and Uri Zwick. Reachability and distance queries via 2-hop labels. *SIAM Journal on Computing*, 32(5):1338–1355, 2003.

11   Tobias Columbus. Search space size in contraction hierarchies. Diploma thesis, Karlsruhe Institute of Technology, 2012.

12   Julian Dibbelt, Ben Strasser, and Dorothea Wagner. Customizable contraction hierarchies. In *International Symposium on Experimental Algorithms (SEA)*, pages 271–282, 2014.

13   Hristo Nicolov Djidjev. On the problem of partitioning planar graphs. *SIAM Journal on Algebraic Discrete Methods*, 3(2):229–240, 1982.

14   Stefan Funke and Sabine Storandt. Provable efficiency of contraction hierarchies with randomized preprocessing. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 479–490, 2015.

15   Robert Geisberger, Peter Sanders, Dominik Schultes, and Christian Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46(3):388–404, 2012. `doi:10.1287/trsc.1110.0401`.

16   Thor Johnson, Neil Robertson, Paul D Seymour, and Robin Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

17   Adrian Kosowski and Laurent Viennot. Beyond highway dimension: Small distance labels using tree skeletons. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1462–1478, 2017.

18   Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE MICROSYSTEMS RESEARCH CENTER, 1989.

19   Nikola Milosavljević. On optimal preprocessing for contraction hierarchies. In *ACM SIG-SPATIAL International Workshop on Computational Transportation Science (IWCTS)*, pages 33–38. ACM, 2012.

20   Tobias Rupp and Stefan Funke. A lower bound for the query phase of contraction hierarchies and hub labels. In *Computer Science in Russia (CSR)*, 2020.

21   Alejandro A Schäffer. Optimal node ranking of trees in linear time. *Information Processing Letters*, 33(2):91–96, 1989.

22   Colin White. Lower bounds in the preprocessing and query phases of routing algorithms. In *Annual European Symposium on Algorithms (ESA)*, pages 1013–1024, 2015.