Distance-Preserving Subgraphs of Interval Graphs

Kshitij Gajjar*[1] and Jaikumar Radhakrishnan[2]

1 **Tata Institute of Fundamental Research, Mumbai, India**
`kshitij.gajjar@tifr.res.in`
2 **Tata Institute of Fundamental Research, Mumbai, India**
`jaikumar@tifr.res.in`

Abstract

We consider the problem of finding small distance-preserving subgraphs of undirected, unweighted interval graphs that have k terminal vertices. We show that every interval graph admits a distance-preserving subgraph with $O(k \log k)$ branching vertices. We also prove a matching $\Omega(k \log k)$ lower bound by exhibiting an interval graph based on bit-reversal permutation matrices. In addition, we show that interval graphs admit subgraphs with $O(k)$ branching vertices that approximate distances up to an additive term of $+1$.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases interval graphs, shortest path, distance-preserving subgraphs, bit-reversal permutation matrix

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.39

1 Introduction

We consider the following problem. Given an undirected graph $G = (V, E)$ with k vertices designated as terminals, our goal is to construct a *small* subgraph H of G. Our notion of smallness is non-standard: we compare solutions based on the number of vertices of degree three or more. We have the following definition.

▶ **Definition 1.** Given an undirected, unweighted graph $G = (V, E)$ and a set $R \subseteq V$ (the terminals), we say that a subgraph $H(V, E')$ of G is distance-preserving for (G, R) if for all terminals $u, v \in R$, $d_G(u, v) = d_H(u, v)$, where d_G and d_H denote the distances in G and H respectively. Let $\deg_{\geq 3}(H)$ denote the number of vertices in H with degree at least three (referred to as *branching vertices*). Let

$$\mathsf{B}(G, R) = \min_H \deg_{\geq 3}(H),$$

where H ranges over all subgraphs that are distance-preserving for (G, R). For a family of graphs $\mathcal{F}$ (such as planar graphs, trees, interval graphs), let

$$\mathsf{B}_{\mathcal{F}}(k) = \max_G \mathsf{B}(G, R),$$

where G ranges over all graphs in $\mathcal{F}$, and R ranges over all subsets of $V(G)$ of size k.

In this work, we obtain essentially tight upper and lower bounds on $\mathsf{B}_{\mathcal{I}}(k)$, where $\mathcal{I}$ is the class of interval graphs.

* Supported by a DAE scholarship.

25th Annual European Symposium on Algorithms (ESA 2017).
Editors: Kirk Pruhs and Christian Sohler; Article No. 39; pp. 39:1–39:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Theorem 2** (Main result). *Let $\mathcal{I}$ denote the class of interval graphs (see Theorem 6).*

(a) *(Upper bound)* $\mathsf{B}_{\mathcal{I}}(k) = O(k \log k)$.

(b) *(Lower bound) There exists a constant c such that for each k, a positive power of two, there exists an interval graph G_{int} with $|R| = k$ terminals such that $\mathsf{B}(G_{\text{int}}, R) \geq c\, k \log k$. This implies that $\mathsf{B}_{\mathcal{I}}(k) = \Omega(k \log k)$.*

Parts (a) and (b) imply that $\mathsf{B}_{\mathcal{I}}(k) = \Theta(k \log k)$.

Remark (i). Part (a) is constructive. Our proof of the upper bound can be turned into an efficient algorithm that, given an interval graph G with n vertices, produces the required distance-preserving subgraph H in running time polynomial in n.
Remark (ii). Our interval graphs are unweighted. If we consider the family of interval graphs with non-negative weights on their edges ($\mathcal{I}_w$), then using [9, Section 5], it is easy to prove that $\mathsf{B}_{\mathcal{I}_w}(k) = \Theta(k^4)$. Details appear in the full version of the paper.

1.1 Motivation and Related Work

The problem of constructing small distance-preserving subgraphs bears close resemblance to several well-studied problems in graph algorithms: graph compression [5], graph spanners [3, 11], Steiner point removal [7, 8], graph contractions [4], etc.

We emphasize two motivations for studying distance-preserving subgraphs, while basing the measure of efficiency on the number of branching vertices. First, this problem is closely related to the notion of distance-preserving minors introduced by Krauthgamer and Zondiner [10]. Second, although the problem restricted to interval graphs is interesting in its own right, it can be seen to arise naturally in contexts where intervals represent time periods for tasks. Let us now elaborate on our first motivation. Later, we elaborate on the second.

▶ **Definition 3.** Let $G(V, E, w)$ be an undirected graph with weight function $w : E \to \mathbb{R}^{\geq 0}$ and a set of terminals $R \subseteq V$. Then, $H(V', E', w')$ with $R \subseteq V' \subseteq V$ and weight function $w' : E' \to \mathbb{R}^{\geq 0}$ is a distance-preserving minor of G if: (i) H is a minor of G, and (ii) $d_H(u, v) = d_G(u, v)\, \forall u, v \in R$.

Subsequent work by Krauthgamer, Nguyên and Zondiner [9, 10] implies that $\mathsf{B}_{\mathcal{G}}(k) = \Theta(k^4)$, where $\mathcal{G}$ is the family of all undirected graphs. Details appear in the full version of the paper.

Using a reduction from the set cover problem, we prove that it is NP-hard to determine if $\mathsf{B}(G, R) \leq m$, when given a general graph $G \in \mathcal{G}$, a set of terminals $R \subseteq V(G)$, and a positive integer m. Details appear in the full version of the paper.

Following the work of Krauthgamer and Zondiner [10], Cheung *et al.* [1] introduced the notion of distance-approximating minors.

▶ **Definition 4.** Let $G(V, E, w)$ be an undirected graph with weight function $w : E \to \mathbb{R}^{\geq 0}$ and a set of terminals $R \subseteq V$. Then, $H(V', E', w')$ with $R \subseteq V' \subseteq V$ and weight function $w' : E' \to \mathbb{R}^{\geq 0}$ is an α-distance-approximating minor (α-DAM) of G if: (i) H is a minor of G, and (ii) $d_G(u, v) \leq d_H(u, v) \leq \alpha \cdot d_G(u, v)\, \forall u, v \in V$.

In analogy with distance-approximating minors one may ask if interval graphs admit distance-approximating subgraphs with a small number of branching vertices.

▶ **Theorem 5.** *Every interval graph G with k terminals admits a subgraph H with $O(k)$ branching vertices such that for all terminals u and v of G*

$$d_G(u, v) \leq d_H(u, v) \leq d_G(u, v) + 1.$$

A proof of Theorem 5 will appear in the full version of the paper.

We now elaborate on our second motivation. The following example[1] illustrates the relevance of distance-preserving (-approximating) subgraphs for interval graphs.

1.2 The Shipping Problem

The port of Bandarport is a busy sea port. Apart from ships with routes originating or terminating at Bandarport, there are many ships that dock at Bandarport en route to their final destination. Thus, Bandarport can be considered a hub for many ships from all over the world.

Consider the following shipping problem. A cargo ship starts from some port X, and has Bandarport somewhere on its route plan. The ship needs to deliver a freight container to another port Y, which is not on its route plan. The container can be dropped off at Bandarport and transferred through a series of ships arriving there until it is finally picked up by a ship that is destined for port Y. Thus, the container is transferred from X to Y via some "intermediate" ships at Bandarport[2].

However, there is a cost associated with transferring a container from one ship to another. This is because each transfer operation requires considerable manpower and resources. Thus, the number of ship-to-ship transfers that a container undergoes should be as small as possible.

Furthermore, there is an added cost if an intermediate ship receives containers from multiple ships, or sends containers to multiple ships. This is mainly because of the bookkeeping overhead involved in maintaining which container goes to which ship. If a ship is receiving all its containers *from* just one ship and sending all those containers *to* just one other ship, then the cost associated with this transfer is zero (since a container cannot be directed to a wrong ship if there is only one option), and this cost increases as the number of *to* and *from* ships increases.

Thus, given the docking times of ships at Bandarport, and a small subset of these ships that require a transfer of containers between each other, our goal is to devise a transfer strategy that meets the following objectives.

- Minimize the number of transfers for each container.
- Minimize the number of ships that have to deal with multiple transfers.

Representing each ship's visit to the port as an interval on the time line, this problem can be modelled using distance-preserving (-approximating) subgraphs of interval graphs. In this setting, a shortest path from an earlier interval to a later interval corresponds to a valid sequence of transfers across ships that moves forward in time. The first objective corresponds to minimizing pairwise distances between terminals; the second objective corresponds to minimizing the number of branching vertices.

[1] This is not a real-life problem, though we learnt that minimizing the number of branching vertices in shipping schedules is logistically desirable.

[2] The container cannot be left at the warehouse/storage unit of Bandarport itself beyond a certain limited period of time.

Let us now quantify this. Suppose that there are a total of n ships that dock at the port of Bandarport. Out of these, there are k ships that require a transfer of containers between each other (typically $k \ll n$). Our results for interval graphs imply the following.

1. If we must make no more than the minimum number of transfers required for each container, then there is a transfer strategy in which the number of ships that have to deal with multiple transfers is $O(k \log k)$.
2. If we are allowed to make *one* more than the minimum number of transfers required for each container, then there is a transfer strategy in which the number of ships that have to deal with multiple transfers is $O(k)$.
3. Neither bound can be improved, i.e. there exist scheduling configurations in which $\Omega(k)$ and $\Omega(k \log k)$ ships, respectively, have to deal with multiple transfers.

1.3 Our Techniques

The linear upper bound for $\mathsf{B}_{\mathcal{I}}(k)$ mentioned in Theorem 5 is easy to prove, and appears in the full version of the paper. However, if we require that distances be preserved exactly, then the problem becomes non-trivial. We now present a broad overview of the techniques involved in proving our main result.

The Upper Bound: We may restrict attention to interval graphs that have interval representations where the terminals are intervals of length 0 (their left and right end points are the same) and the non-terminals are intervals of length 1 (details appear in the full version of the paper). It is well-known that shortest paths in interval graphs can be constructed using a simple greedy algorithm. We build a subgraph consisting of such shortest paths starting at different terminals and add edges to it so that all inter-terminal shortest paths become available in the subgraph. We use a divide-and-conquer strategy, repeatedly “cutting” the graph down the middle into smaller interval graphs. Then we glue the solutions to the two smaller problems together. For this, we need a key observation (which appears to be applicable specifically to interval graphs) that allows one shortest path to “hop” onto another. In this, our upper bound method is significantly different from methods used previously for other families of graphs.

The Lower Bound: We construct an interval graph and arrange its vertices on a two-dimensional grid instead of the more natural one-dimensional number line. We then show that this grid can be thought of as a matrix, in particular, the bit-reversal permutation matrix (where the ones corresponding to terminals and the zeros to non-terminals). The bit-reversal permutation matrix has seen many applications, most notably in the celebrated Cooley-Tukey algorithm for Fast Fourier Transform [2]. Prior to our work too, it has been used to devise lower bounds (e.g. [6, 12]). Examining the routes available for shortest paths in our interval graph constructing using the bit-reversal permutation matrix requires (i) an analysis of common prefixes of binary sequences, and (ii) building a correspondence between branching vertices and the $k \log k/2$ edges of a $(\log k)$-dimensional Boolean hypercube.

In our formulation, we count the number of branching vertices (vertices with degree ≥ 3). It is also reasonable to consider the number of edges incident on non-terminal branching vertices (we refer to such edges as *branching edges*) as the measure of complexity. Our $\Omega(k \log k)$ lower bound is clearly applicable to the number of branching edges as well. In fact, using a more direct argument, one can show that there are interval graphs with k terminals that admit distance-preserving subgraphs with $O(k)$ branching vertices, but need $\Omega(k \log k)$ branching edges. Details appear in the full version of the paper. However, we do not know if all interval graphs admit distance-preserving subgraphs with $O(k \log k)$ branching edges: the best upper bound we know for this variant is $O(k \log^2 k)$.

2 Interval Graphs

We work with the following definition of interval graphs.

▶ **Definition 6.** An interval graph is an undirected graph $G(V, E, \mathsf{left}, \mathsf{right})$ with vertex set V, edge set E, and real-valued functions $\mathsf{left} : V \to \mathbb{R}$ and $\mathsf{right} : V \to \mathbb{R}$ such that:

- $\mathsf{left}(x) \le \mathsf{right}(x) \quad \forall x \in V$;
- $(u, v) \in E \Leftrightarrow [\mathsf{left}(u), \mathsf{right}(u)] \cap [\mathsf{left}(v), \mathsf{right}(v)] \ne \emptyset$.

We order the vertices of the interval graph according to the end points of their corresponding intervals. For simplicity, we assume that all the end points of the intervals have distinct values. Define relations "$\preceq$" and "$\prec$" on the set of vertices V as follows.

$$u \preceq v \Leftrightarrow \mathsf{right}(u) \le \mathsf{right}(v) \qquad \forall u, v \in V.$$
$$u \prec v \Leftrightarrow \mathsf{right}(u) < \mathsf{right}(v) \qquad \forall u, v \in V.$$

Note that if $u \prec v$, then $u \ne v$.

It is well-known that shortest paths in interval graphs can be constructed using a greedy algorithm which proceeds as follows. Suppose we need to construct a shortest path from interval u to interval v (assume $u \prec v$). The greedy algorithm starts at u. In each step it chooses the next interval that intersects the current interval and reaches farthest to the right. It stops as soon as the current interval intersects v. Let $P_G^{\mathrm{gr}}(u, v)$ be the shortest path produced by this greedy algorithm between u and v ($u \prec v$).

Given real numbers $a, b \in \mathbb{R}$ such that $a \le b$, let $G[a, b]$ be the induced subgraph on those vertices v of G such that $[\mathsf{left}(v), \mathsf{right}(v)] \cap [a, b] \ne \emptyset$. Similarly, let $G[a, b)$ be the induced subgraph on those vertices v of G such that $[\mathsf{left}(v), \mathsf{right}(v)] \cap [a, b) \ne \emptyset$.

3 Proof of the Upper Bound

In this section, we show that any interval graph G with k terminals has a distance-preserving subgraph with $O(k \log k)$ branching vertices, which is simply Theorem 2 (a), restated here for completeness.

▶ **Theorem 7.** *If $\mathcal{I}$ is the family of all interval graphs, then* $\mathsf{B}_{\mathcal{I}}(k) = O(k \log k)$.

Fix an interval graph G on k terminals. Our goal to obtain a distance-preserving subgraph H of G with $O(k \log k)$ branching vertices. Note that the H that we obtain is not necessarily an interval graph. This is because H need not be an induced subgraph of G. We may assume (details in the full version of the paper) that all terminals in G are point intervals and all non-terminals are unit intervals.

Consider the greedy path $P_G^{\mathrm{gr}}(t_i, t_k)$ ($i < k$), where t_k is the rightmost terminal. Our distance-preserving subgraph includes greedy paths from t_i to t_k for all $1 \le i < k$. Let

$$H_0 = \bigcup_{1 \le i < k} P_G^{\mathrm{gr}}(i, k). \tag{1}$$

Now, H_0 already provides for shortest paths from each terminal t_i to t_k. In fact, it can be viewed as a shortest path tree with root t_k, but constructed backwards. Thus, the total number of branching vertices in H_0 is $O(k)$. We still need to arrange for shortest paths between other pairs of terminals (t_i, t_j). The path $P_G^{\mathrm{gr}}(t_i, t_j)$ (for $i < j < k$) is either entirely contained in $P_G^{\mathrm{gr}}(t_i, t_k)$, or it follows $P_G^{\mathrm{gr}}(t_i, t_k)$ until it reaches a neighbour of t_j and then *branches off* to connect to t_j. We can consider including all paths of the form $P_G^{\mathrm{gr}}(t_i, t_j)$ in

H_0. That is, we need to link each such t_j to vertices from H_0 so that each path $P_G^{\mathrm{gr}}(t_i, t_j)$ becomes available. If this is done without additional care, we might end up introducing $\Omega(k)$ additional branching vertices per terminal, and $\Omega(k^2)$ branching vertices in all, far more than we claimed.

The crucial idea for overcoming this difficulty is contained in the following lemma.

▶ **Lemma 8.** *Suppose $v \prec w$ and $d(v, w) = 1$. Let $(v, v_1, v_2, \ldots, v_\ell)$ and $(w, w_1, w_2, \ldots, w_{\ell'})$ be greedy shortest paths starting from v and w respectively. Suppose $\mathsf{right}(v_\ell) < \mathsf{right}(w_{\ell'})$. Then, $\ell \leq \ell'$.*

Proof. Since $d(v, w) = 1$, the greedy strategy reaches at least as far in $j + 1$ steps from v as it does in j steps from w. Suppose for contradiction that $\ell > \ell'$ (that is $\ell \geq \ell' + 1$). Then, we have $\mathsf{right}(w_{\ell'}) \leq \mathsf{right}(v_{\ell'+1}) \leq \mathsf{right}(v_\ell)$, contradicting our assumption that $\mathsf{right}(v_\ell) < \mathsf{right}(w_{\ell'})$. ◀

The above lemma is crucial for the construction of our subgraph H. For example, suppose t_i and t_j both need to reach t_r via a shortest path. Suppose (w_i, t_r) is the last edge of $P_G^{\mathrm{gr}}(t_i, t_r)$ and (w_j, t_r) is the last edge of $P_G^{\mathrm{gr}}(t_j, t_r)$. We claim that it is sufficient to include **only** one of these edges in H. If $\mathsf{right}(w_j) < \mathsf{right}(w_i)$, then it is enough to include the edge (w_j, t_r) in H; as long as t_i has a shortest path to w_j, this edge serves for shortest paths *to* t_r *from* both t_i and t_j. In the construction below, we add links to the greedy paths of H_0 so that we need to provide only one such edge per terminal. This idea forms the basis of the divide-and-conquer strategy which we present below.

Suppose G has 2ℓ terminals. We find a point x so that both $G_{\mathrm{left}} = G[-\infty, x]$ and $G_{\mathrm{right}} = G[x, \infty]$ have ℓ terminals. By induction, we find distance-preserving subgraphs H_{left} and H_{right} of G_{left} and G_{right} with at most $f(\ell)$ branching vertices each. The union of H_{left} and H_{right} has just $2f(\ell)$ branching vertices, but it does not yet guarantee shortest paths from terminals in H_{left} to terminals in H_{right}. Using Theorem 8 and the discussion above, we connect each terminal t_j in H_{right} to **only** one of the greedy shortest paths of terminals from H_{left}, and ensure that shortest paths to t_j are preserved from all terminals t_i in H_{left}. This creates $O(\ell)$ additional branching vertices and give us a recurrence of the form

$$f(2\ell) \leq 2f(\ell) + O(\ell),$$

and the desired upper bound of $O(k \log k)$. Unfortunately, there are technical difficulties in implementing the above strategy as stated. It is therefore helpful to augment H_0 by adding all greedy paths $P_G^{\mathrm{gr}}(t_i, t_j)$, where $d(i, j) \leq 4$. As a result, for each terminal t_i, the first three vertices on $P_G^{\mathrm{gr}}(t_i, t_k)$ might become branching vertices. In all, this adds a *one-time cost* of $O(k)$ branching vertices to our subgraph. We now present the argument formally.

For each (a, b), let $f(a, b)$ be the minimum number of non-terminals in a subgraph H of $G[a, b]$ such that $H_0 \cup H$ preserves all inter-terminal distances in $G[a, b]$; let

$$f(\ell) = \max_{(a,b)} f(a, b),$$

where (a, b) ranges over all pairs such that $G[a, b]$ has at most ℓ terminals. The following lemma is the basis of our induction.

▶ **Lemma 9.**
(i) $f(1) = 0$*;*
(ii) $f(2\ell) \leq 2f(\ell) + O(\ell)$.

Proof. Part (i) is trivial. For part (ii), fix a pair (a, b) such that $G[a, b]$ has at most 2ℓ terminals. If $b - a \leq 1$, H_0 already preserves distances between every two terminals in $G[a, b]$. So, we may take H to be empty. Now assume that $b - a > 1$. Pick $x \in [a, b]$ as large as possible such that (i) $b - x \geq 1$, and (ii) $G[x, b]$ has at least ℓ terminals.

Let $G_{\text{left}} = G[a, x)$ and $G_{\text{right}} = G[x, b]$. Since G_{right} has at least ℓ terminals, G_{left} has at most ℓ terminals. So, we obtain (by induction) a subgraph H_{left} of $G[a, b]$ with at most $f(\ell)$ non-terminals, such that $H_0 \cup H_{\text{left}}$ preserves inter-terminal distances in G_{left}. If $b - x > 1$, then G_{right} has exactly ℓ terminals, and we obtain by induction a subgraph H_{right} with at most $f(\ell)$ non-terminals such that $H_0 \cup H_{\text{right}}$ preserves all inter-terminal distances in $G[x, b]$. If $b - x = 1$, then we may take H_{right} to be empty (for H_0 already preserves inter-terminal distances in $G[x, b]$).

Our final subgraph H shall be of the form $H_{\text{left}} \cup H_{\text{right}} \cup H_A \cup H_B$, where H_A and H_B are defined as follows. First, consider H_A. Let P_{left} be the set of greedy paths from the terminals in H_{left} to the terminal t_k. Let V_A be the set of all non-terminal intervals of P_{left} that intersect with the interval $[x, x + 1]$. It is easy to see that any path in P_{left} contributes at most 4 non-terminals to V_A. So, $|V_A| \leq 4\ell$. Let H_A be the subgraph of $G[a, b]$ induced by V_A and the terminals in $G[x, x + 1]$.

Note that $H_0 \cup H_{\text{left}} \cup H_{\text{right}} \cup H_A$ preserves all inter-terminal distances in $G[a, x + 1]$ as well as all inter-terminal distances in $G[x + 1, b]$. It, in fact, does more. For each terminal t_i in $G[a, x)$, let v_i be the last vertex on the greedy path $P^{\text{gr}}_G(t_i, t_k)$ that is in V_A. Then, the above graph contains the greedy shortest path from every terminal t_j in $G[a, x]$ to v_i.

Now, it only remains to ensure that distances between terminals in $G[a, x)$ and terminals in $G[x + 1, b]$ are preserved. Let us now define H_B. For each terminal t_j in $G[x + 1, b]$, let v be the earliest interval (with respect to $\prec$) of P_{left} that contains t_j. Then, we include the edge (v, t_j) in H_B. Thus, H_B contains at most one non-terminal per vertex in $G[x + 1, b]$, that is, $O(\ell)$ non-terminals in all. This completes the description of H_A and H_B. The final subgraph is $H = H_{\text{left}} \cup H_{\text{right}} \cup H_A \cup H_B$.

▶ **Claim 10.** *Let t_i be a terminal in $G[a, x)$ and t_r be a terminal in $G[x, b]$. Then, $H_0 \cup H$ preserves the distance between terminal t_i and t_r.*

Proof of Claim 10. Let v be the vertex that we attached to t_r in H_B. If v is on $P^{\text{gr}}_G(t_i, t_k)$, then it follows that $P^{\text{gr}}_G(t_i, t_r)$ is in H, and we are done. So we assume that v is not on $P^{\text{gr}}_G(t_i, t_k)$. Then, let $j \neq i$ be such that $v \in P^{\text{gr}}_G(t_j, t_k)$. Then, we have paths

$$P_G(t_i, t_r) = (t_i, w_1, w_2, \ldots, w_p, w_{p+1}, \ldots, w_{\ell'}, t_r);$$
$$P_H(t_i, t_r) = (t_i, w_1, w_2, \ldots, w_p, v_{q+1}, \ldots, v_\ell = v, t_r),$$

where v_{q+1} is the last vertex on $P^{\text{gr}}_G(t_j, t_k)$ in $G[x, x + 1]$, and w_p is the first vertex on $P^{\text{gr}}_G(t_i, t_r)$ such that $(w_p, v_{q+1}) \in E(G)$. From the construction of H_A, $(w_p, v_{q+1}) \in E(H)$. Following v_q, $(v_{q+1}, \ldots, v_\ell = v, t_r)$ are the subsequent vertices on $P^{\text{gr}}_G(t_j, t_r)$. Note that: (i) $v_{q+1} \prec w_{p+1}$ (otherwise v is on $P^{\text{gr}}_G(t_i, t_k)$), (ii) $d(v_{q+1}, w_{p+1}) = 1$ (both intervals contain $\mathsf{right}(w_p)$), and (iii) $\mathsf{right}(v_\ell) < \mathsf{right}(w_{\ell'})$ (since v is the earliest interval of P_{left} that contains t_j). By Theorem 8, $\ell - q - 1 \leq \ell' - p - 1$. Thus, $P_H(t_i, t_r)$ is no longer than $P^{\text{gr}}_G(t_i, t_r)$. ◀

◀

We can now complete the proof of Theorem 7. By Theorem 9, there is a subgraph H' of G such that $H = H_0 \cup H'$ preserves all inter-terminal distances in G, H_0 has $O(k)$ branching vertices and H' has $O(k \log k)$ non-terminals. It follows that H has $O(k \log k)$ branching vertices.

4 Proof of the Lower Bound

In this section, we show that there exists an interval graph G_{int} such that any distance-preserving subgraph of G_{int} has $\Omega(k \log k)$ branching vertices, which is simply Theorem 2 (b), restated here for completeness.

▶ **Theorem 11.** *If $\mathcal{I}$ is the family of all interval graphs, then $\mathsf{B}_{\mathcal{I}}(k) = \Omega(k \log k)$.*

4.1 Preliminaries

We first set up some terminology that we use in this section. Let $k = 2^\gamma$, where γ is a positive integer. We identify the numbers in the set $\{0, 1, \ldots, k-1\}$ with elements of $\{0,1\}^\gamma$ using the γ-bit binary representation. We index the bits of the binary strings from left to right using integers $i = 1, 2, \ldots, \gamma$. Thus, $x[i]$ denotes the i-th bit of x (from the left); we use $x[i, j]$ to denote the string $x[i]\, x[i+1] \ldots x[j]$ of length $j - i + 1$ (here i, j satisfy $1 \le i \le j \le \gamma$).

For a string of bits a, we use $\mathbf{rev}_\gamma(a)$ to represent the reverse of a, that is, the binary string obtained by writing the bits of a in the reverse order (e.g., $\mathbf{rev}_\gamma(00010) = 01000$). We may arrange binary strings in a binary tree. Refer to Figure 1 for an example. The root is the empty string; the left child of a vertex x is the vertex $x\,0$, and its right child is the vertex $x\,1$. In particular, the string y is a *descendant* of the string x if y is obtained by concatenating x with some (possibly empty) string z, that is, $y = x\,z$. Consider the binary tree of depth γ, whose leaves correspond to elements of $\{0,1\}^\gamma$. For distinct elements $x, y \in \{0,1\}^\gamma$, let $\mathbf{lca}(x, y)$ be the *lowest common ancestor* of x and y defined as follows:

$$\mathbf{lca}(x, y) = x[1, \ell - 1] = y[1, \ell - 1], \text{ where } \ell = \min\{i \in [\gamma] : x[i] \neq y[i]\}.$$

For example, $\mathbf{lca}(0100111, 0101010) = 010$. Let $\lfloor \mathbf{lca}(x, y) \rfloor$ be the *floor of* $\mathbf{lca}(x, y)$, and $\lceil \mathbf{lca}(x, y) \rceil$ be the *ceiling of* $\mathbf{lca}(x, y)$ defined as follows:

$$\lfloor \mathbf{lca}(x, y) \rfloor = \mathbf{lca}(x, y)\, 0\, 1^{\gamma - \ell}$$
$$\lceil \mathbf{lca}(x, y) \rceil = \mathbf{lca}(x, y)\, 1\, 0^{\gamma - \ell}$$

Since $\lfloor \mathbf{lca}(x, y) \rfloor, \lceil \mathbf{lca}(x, y) \rceil \in \{0,1\}^\gamma$, we may regard $\lfloor \mathbf{lca}(x, y) \rfloor$ and $\lceil \mathbf{lca}(x, y) \rceil$ as numbers in the set $\{0, 1, \ldots, k-1\}$. Note that $\lfloor \mathbf{lca}(x, y) \rfloor = \lceil \mathbf{lca}(x, y) \rceil - 1$, and if $x < y$, then $\lfloor \mathbf{lca}(x, y) \rfloor \in [x, y)$ and $\lceil \mathbf{lca}(x, y) \rceil \in (x, y]$[3].

Strings in $\{0,1\}^\gamma$ can also be viewed as vertices of an γ-dimensional hypercube, with edge set

$$\mathcal{H}_\gamma = \{(x, x') : x, x' \in \{0,1\}^\gamma \text{ and } x < x' \text{ and } \mathtt{Ham}(x, x') = 1\},$$

where $\mathtt{Ham}(x, x')$ is the Hamming distance between x and x'. Thus, if $(x, x') \in \mathcal{H}_\gamma$, then x and x' differ at a unique location where x has a zero and x' a one.

▶ **Claim 12.** *Suppose (x, x') and (y, y') are distinct edges of $\mathcal{H}_\gamma$.*
(a) *If $\mathbf{lca}(x, x') = \mathbf{lca}(y, y')$, then $[\mathbf{rev}_\gamma(x), \mathbf{rev}_\gamma(x')] \cap [\mathbf{rev}_\gamma(y), \mathbf{rev}_\gamma(y')] = \emptyset$.*
(b) *If $\{\lfloor \mathbf{lca}(x, x') \rfloor, \lfloor \mathbf{lca}(y, y') \rfloor\} \subseteq [x, x') \cap [y, y')$, then*

$$[\mathbf{rev}_\gamma(x), \mathbf{rev}_\gamma(x')] \cap [\mathbf{rev}_\gamma(y), \mathbf{rev}_\gamma(y')] = \emptyset.$$

[3] $[x, y] \triangleq \{x, x+1, x+2, \ldots, y\}$ and $[x, y) \triangleq \{x, x+1, x+2, \ldots, y-1\}$.

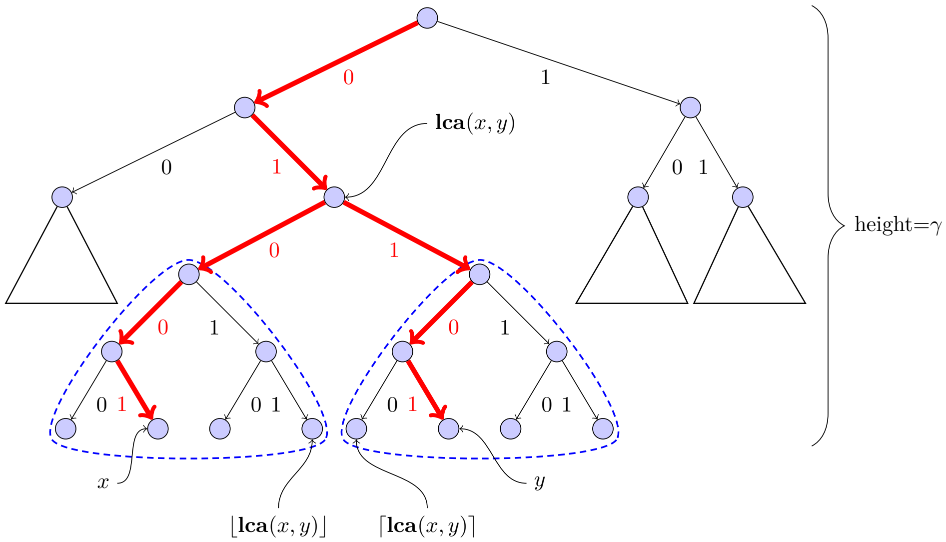

Figure 1 A complete binary tree of height γ having $k = 2^\gamma$ leaves. In this example, $\gamma = 5$, $x = 01001$ and $y = 01101$. Thus, $\texttt{Ham}(x, y) = 1$ and $|\mathbf{lca}(x, y)| = 2$.

Proof. Although part (b) implies part (a), it is easier to show part (a) first, and then derive part (b) from it. For part (a), let $|\mathbf{lca}(x, x')| = |\mathbf{lca}(y, y')| = \ell - 1$. Let $a, b \in \{0, 1\}^{\gamma-\ell}$ be such that

$$a = x[\ell + 1, \gamma] = x'[\ell + 1, \gamma] \neq y[\ell + 1, \gamma] = y'[\ell + 1, \gamma] = b.$$

In particular, we have $a \neq b$ (implying $\mathbf{rev}_{\gamma-\ell}(a) \neq \mathbf{rev}_{\gamma-\ell}(b)$). Note that $\mathbf{rev}_\gamma(a)$ represents the $\gamma - \ell$ most significant bits of $\mathbf{rev}_\gamma(x)$ and $\mathbf{rev}_\gamma(x')$; similarly, $\mathbf{rev}_\gamma(b)$ represents the $\gamma - \ell$ most significant bits of $\mathbf{rev}_\gamma(y)$ and $\mathbf{rev}_\gamma(y')$.

If $\mathbf{rev}_{\gamma-\ell}(a) < \mathbf{rev}_{\gamma-\ell}(b)$ then $\mathbf{rev}_\gamma(x') < \mathbf{rev}_\gamma(y)$; and if $\mathbf{rev}_{\gamma-\ell}(b) < \mathbf{rev}_{\gamma-\ell}(a)$ then $\mathbf{rev}_\gamma(y') < \mathbf{rev}_\gamma(x)$. In either case, $[\mathbf{rev}_\gamma(x), \mathbf{rev}_\gamma(x')]$ and $[\mathbf{rev}_\gamma(y), \mathbf{rev}_\gamma(y')]$ are disjoint, proving part (a).

Next, consider part (b). Suppose $\lfloor \mathbf{lca}(x, x) \rfloor, \lfloor \mathbf{lca}(y, y') \rfloor \in [x, x') \cap [y, y')$. Since every $p \in [x, x')$ is a descendant of $\mathbf{lca}(x, x')$, we conclude that $\mathbf{lca}(y, y')$ is a descendant of $\mathbf{lca}(x, x')$. Similarly, $\mathbf{lca}(x, x')$ is a descendant of $\mathbf{lca}(y, y')$. But then $\mathbf{lca}(x, x') = \mathbf{lca}(y, y')$, and part (b) follows from part (a). ◀

4.2 Manhattan Graphs

In this section, we describe a directed grid graph G_k^{bit} (which we refer to as the Manhattan graph) with $3k$ terminals. We show that any distance-preserving subgraph of G_k^{bit} has $\Omega(k \log k)$ branching vertices. The graph has $k^2 + 2k$ vertices arranged in a square grid. The vertices and edges of G_k^{bit} are defined as follows. (Figure 2 makes this definition easier to understand.)

1. $V(G_k^{\text{bit}}) = \{0, 1, 2, \ldots, k - 1\} \times \{-1, 0, 1, \ldots, k\}$.
2. There are three kinds of edges: horizontal, upward and downward; the edge set is given by $E(G_k^{\text{bit}}) = E_{\text{hor}} \cup E_{\text{up}} \cup E_{\text{down}}$, where

$$E_{\text{hor}} = \{((i, j), (i, j + 1)) : i = 0, 1, \ldots, k - 1 \text{ and } j = -1, 0, \ldots, k - 1\};$$
$$E_{\text{up}} = \{((i_1, j), (i_2, j)) : 0 \leq i_2 < i_1 \leq k - 1 \text{ and } j = -1, 0, \ldots, k\};$$
$$E_{\text{down}} = \{((i_1, j), (i_2, j)) : 0 \leq i_1 < i_2 \leq k - 1 \text{ and } j = -1, 0, \ldots, k\}.$$

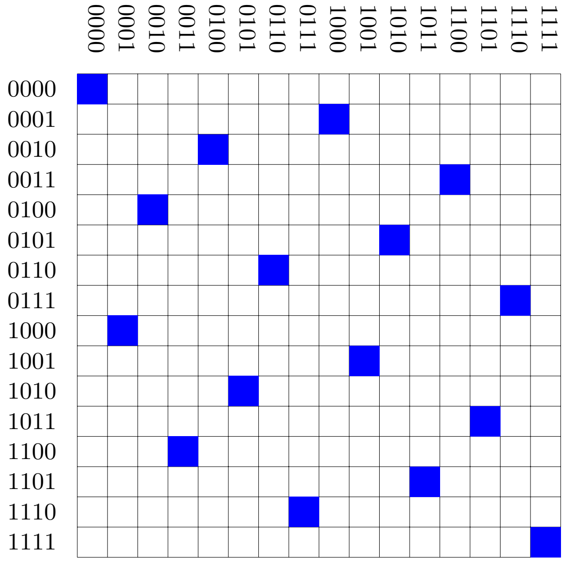

Figure 2 The bit-reversal permutation matrix for $k = 16$. Each cell represents a vertex: the blue cells represent the terminal vertices of T_{mid}; all the other vertices are non-terminals. Edges are named horizontal, upward and downward in the natural way.

3. The edge weights are given by the function $w : E(G_k^{\text{bit}}) \to \{0, 1\}$, defined as follows: $w(e) = 1$ if $e \in E_{\text{hor}} \cup E_{\text{up}}$, and $w(e) = 0$ if $e \in E_{\text{down}}$.

The set of terminals are of the form $T = T_{\text{left}} \cup T_{\text{mid}} \cup T_{\text{right}}$, where

$$\begin{aligned} T_{\text{left}} &= \{0, 1, \dots, k-1\} \times \{-1\}, \\ T_{\text{right}} &= \{0, 1, \dots, k-1\} \times \{k\}; \\ T_{\text{mid}} &= \{(\mathbf{rev}_\gamma(i), i) : i = 0, 1, \dots, k-1\}. \end{aligned}$$

This completes the definition of G_k^{bit}.

Fix an optimal distance-preserving subgraph H_k^{bit} of G_k^{bit}. We shall show that H_k^{bit} has $\Omega(k \log k)$ vertices of degree at least 3.

▶ **Lemma 13.** *$V(H_k^{\text{bit}}) = V(G_k^{\text{bit}})$ and $E_{\text{hor}} \subseteq E(H_k^{\text{bit}})$.*

Proof. Note that the *unique* shortest path between the terminals $(i, -1)$ and (i, k) is precisely $((i, -1), (i, 0), \dots, (i, k))$. Thus, all vertices and all horizontal edges in the i-th row of G_k^{bit} must be part of H_k^{bit}. ◀

It follows from Theorem 13 that every non-terminal vertex in H_k^{bit} has degree at least two, namely the two horizontal edges incident on it.

Special edges: From now on, we rely solely on the fact that H_k^{bit} is distance-preserving for every pair of terminals in T_{mid}, i.e. we prove the stronger statement that just preserving terminal distances in T_{mid} requires $\Omega(k \log k)$ branching vertices.

Order the vertices in T_{mid} as $t_0, t_1, \dots, t_{k-1}$, where $t_i = (\mathbf{rev}_\gamma(i), i)$. Note that these terminals appear in different rows and columns. Consider the following pairs of terminals.

$$T_{\text{twins}} = \{(t_i, t_j) : (i, j) \in \mathcal{H}_\gamma\}.$$

For each twin (t_i, t_j), fix $P(i, j)$, a path of minimum distance between t_i and t_j in H_k^{bit}. We are now set to formally define special edges.

▶ **Definition 14.** Let $\mathbf{spcl}(i,j) = ((r_{ij}, \lfloor \mathbf{lca}(i,j) \rfloor), (r_{ij}, \lceil \mathbf{lca}(i,j) \rceil))$ be an edge of $P(i,j)$, where $\mathbf{rev}_\gamma(i) \le r_{ij} \le \mathbf{rev}_\gamma(j)$. (By Theorem 15, such an edge exists.) Let $\mathbf{spcl} = \{\mathbf{spcl}(i,j) : (t_i, t_j) \in T_{\text{twins}}\}$.

▶ **Lemma 15.** *Let $(t_i, t_j) \in T_{\text{twins}}$; let $\ell = \lfloor \mathbf{lca}(i,j) \rfloor$. Then, there is an $r_{ij} \in [\mathbf{rev}_\gamma(i), \mathbf{rev}_\gamma(j)]$ such that $P(i,j)$ contains the edge $((r_{ij}, \ell), (r_{ij}, \ell+1))$.*

Proof. We have $i < j$, $t_i = (\mathbf{rev}_\gamma(i), i)$ and $t_j = (\mathbf{rev}_\gamma(j), j)$. Also note that since $(i,j) \in \mathcal{H}_\gamma$, $\mathbf{rev}_\gamma(i) < \mathbf{rev}_\gamma(j)$. Thus, $d(i,j) = j - i$, and the shortest path $P(t_i, t_j)$ goes from column i to column j and never skips a column. Since $\ell \in [i,j)$, there must be an edge in $P(i,j)$ of the form $((r_{ij}, \ell), (r_{ij}, \ell+1))$ (say, the edge of $P(i,j)$ that leaves column ℓ for the last time). We claim that $r_{ij} \in [\mathbf{rev}_\gamma(i), \mathbf{rev}_\gamma(j)]$. For otherwise, $P(i,j)$ would contain an edge in E_{up}. Then, apart from the $j-i$ edges from E_{hor}, $P(i,j)$ would contain an additional edge from E_{up} of weight 1; that is, the length of $P(i,j)$ would be at least $j-i+1$—contradicting the fact that $d(i,j) = j-i$. ◀

▶ **Lemma 16** (Key lemma). *Suppose $(t_x, t_{x'})$ and $(t_y, t_{y'})$ are distinct pairs in T_{twins} such that their special edges are in the same row r, that is,*

$$\mathbf{spcl}(x,x') = ((r,\alpha),(r,\alpha+1))$$
$$\mathbf{spcl}(y,y') = ((r,\beta),(r,\beta+1)),$$

where $\alpha = \lfloor \mathbf{lca}(x,x') \rfloor$ and $\beta = \lfloor \mathbf{lca}(y,y') \rfloor$.

(a) *Then, $\alpha \ne \beta$. In particular, $\mathbf{spcl}(x,x') \ne \mathbf{spcl}(y,y')$.*

(b) *Suppose $\alpha < \beta$. Then, there exists an $\ell \in [\alpha+1, \beta]$ such that (r,ℓ) is either a branching vertex or a terminal in H_k^{bit}.*

Proof. Part (a) follows from Claim 12 (a). Consider part (b). By our definition of special edge, $r \in [\mathbf{rev}_\gamma(x), \mathbf{rev}_\gamma(x')]$ and $r \in [\mathbf{rev}_\gamma(y), \mathbf{rev}_\gamma(y')]$. So, $[\mathbf{rev}_\gamma(x), \mathbf{rev}_\gamma(x')] \cap [\mathbf{rev}_\gamma(y), \mathbf{rev}_\gamma(y')] \ne \emptyset$, and by Claim 12 (b) (in the contrapositive) either $\alpha \notin [y,y')$ or $\beta \notin [x,x')$. If $\alpha \notin [y,y')$, $\mathbf{spcl}(x,x')$ is not on $P(y,y')$. The first vertex in row r that is part of $P(y,y')$ is in a column $\ell \in [\alpha+1,\beta]$. Then, (r,ℓ) is either a branching vertex or the terminal t_y. On the other hand, if $\beta \notin [x,x')$, then the last vertex of $P(t_x, t_{x'})$ in row r lies in a column $\ell \in [\alpha+1,\beta]$, so (r,ℓ) is either a branching vertex or the terminal $t_{x'}$. ◀

▶ **Corollary 17.**

(a) $|\mathbf{spcl}| = |T_{\text{twins}}| = k \log k/2$ *(since $|T_{\text{twins}}| = |\mathcal{H}_\gamma| = k\log k/2$).*

(b) *If two edges in* **spcl** *fall in the same row, then there is a branching vertex or a terminal separating them.*

▶ **Theorem 18.** *H_k^{bit} has $\Omega(k \log k)$ branching vertices.*

Proof. For each $i \in \{0, 1, \dots, k-1\}$, let δ_i be the number of distinct edges in **spcl** in row i. Then, by Theorem 17 (a), we have

$$\sum_{i=0}^{k-1} \delta_i = |\mathbf{spcl}| = \left(\frac{k \log k}{2}\right).$$

Furthermore, Theorem 17 (b) implies that there are at least $\delta_i - 2$ many branching vertices of the form (i,x) in H_k^{bit}, where $0 \le x \le k-1$. Thus, the total number of branching vertices in H_k^{bit} is at least

$$(\delta_0 - 2) + (\delta_1 - 2) + \cdots + (\delta_{k-1} - 2) = \left(\sum_{i=0}^{k-1} \delta_i\right) - 2k = \left(\frac{k \log k}{2}\right) - 2k.$$

Since this quantity is $\Omega(k \log k)$, this completes the proof. ◀

4.3 Translating the Lower Bound to Interval Graphs

In this section, we present an interval graph G_{int} with $O(k)$ terminals, for which every distance-preserving subgraph has $\Omega(k \log k)$ branching vertices. Our lower bound relies on the lower bound for the Manhattan graph shown in the previous section. Let us describe the interval graph. Let $\mathcal{J}$ be the set of intervals.

$$\mathcal{J} = \{[x, x+1] : x = -1, -1+1/k, \ldots, -1/k, 0, \ldots, k, k+1/k, \ldots, k+1-1/k\}.$$

Thus, we have unit intervals starting at all integral multiples of $1/k$ in the range $[-1, k+1-1/k]$; in all we have $k(k+2)$ intervals in $\mathcal{J}$. These intervals naturally define an interval graph. Furthermore, the edges of G_{int} are *directed* as follows. Orient the edges of G_{int} *from* an earlier interval *to* a later interval, i.e. $([x, x+1], [y, y+1])$ is a directed edge *from* $[x, x+1]$ *to* $[y, y+1]$ if and only if $x < y \leq x+1$. Note that this orientation does not affect shortest paths. Any shortest path from $[i, i+1]$ to $[j, j+1]$ (where $i < j$) in the undirected interval graph is also a valid directed shortest path in G_{int}. Also, G_{int} has $k^2 + 2k$ vertices, which (surprisingly?) is the number of vertices in the Manhattan graph of the previous section. In fact, the connection is deeper. Let us arrange the intervals in a two-dimensional array

$$\mathbf{A} = \langle a_{i,j} : i = 0, \ldots, k-1 \text{ and } j = -1, 0, \ldots, k \rangle,$$

where a_{ij} corresponds to the interval $[j + (k-1-i)/k, j+1+(k-1-i)/k]$. Thus, the first k intervals of $\mathcal{J}$ occupy the left most column of the array $\mathbf{A}$ (from bottom to top); the next k intervals occupy the next column (again from bottom to top), and so on. It is easy to check that, after this arrangement, the directed edges of G_{int} are of three types: horizontal, upward and slanting.

$$\begin{aligned}
E_{\text{hor}}(G_{\text{int}}) &= \{(a_{i,j}, a_{i,j+1}) : 0 \leq i \leq k-1 \text{ and } -1 \leq j \leq k-1\}; \\
E_{\text{up}}(G_{\text{int}}) &= \{(a_{i,j}, a_{i',j}) : 1 \leq i \leq k-1 \ \text{ and } 0 \leq i' < i \ \text{ and } -1 \leq j \leq k \ \}; \\
E_{\text{slant}}(G_{\text{int}}) &= \{(a_{i,j}, a_{i',j+1}) : 0 \leq i \leq k-2 \text{ and } i < i' \leq k-1 \text{ and } -1 \leq j \leq k-1\}.
\end{aligned}$$

Thus, $E(G_{\text{int}}) = E_{\text{hor}}(G_{\text{int}}) \cup E_{\text{up}}(G_{\text{int}}) \cup E_{\text{slant}}(G_{\text{int}})$. All edges in $E(G_{\text{int}})$ have weight 1. This 2d array can be viewed as a $k \times (k+2)$ grid, and we place terminals in this graph at the same $3k$ locations as in the Manhattan graph. This completes the description of G_{int}. Using the lower bound shown for Manhattan graphs in the previous section (Theorem 18), we complete the proof of Theorem 11. (Details appear in the full version of the paper.)

Acknowledgments. We are grateful to Nithin Varma and Rakesh Venkat for introducing us to the problem and helping with the initial analysis of shortest paths in interval graphs, and for their comments at various stages of this work. We would also like to thank the anonymous reviewers of this paper for their helpful suggestions and comments.

References

1 Yun Kuen Cheung, Gramoz Goranci, and Monika Henzinger. Graph Minors for Preserving Terminal Distances Approximately - Lower and Upper Bounds. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2016.131.

2 James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

3 Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM Journal on Discrete Mathematics*, 20(2):463–501, 2006.

4 Karl Däubel, Yann Disser, Max Klimm, Torsten Mütze, and Frieder Smolny. Distance-preserving graph contractions. *CoRR*, abs/1705.04544, 2017. URL: http://arxiv.org/abs/1705.04544.

5 Tomás Feder and Rajeev Motwani. Clique partitions, graph compression and speeding-up algorithms. *J. Comput. System Sci.*, 51(2):261–272, 1995. doi:10.1006/jcss.1995.1065.

6 Greg N Frederickson and Nancy A Lynch. Electing a leader in a synchronous ring. *Journal of the ACM (JACM)*, 34(1):98–115, 1987.

7 Anupam Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 220–227. Society for Industrial and Applied Mathematics, 2001.

8 Lior Kamma, Robert Krauthgamer, and Huy L Nguyên. Cutting corners cheaply, or how to remove steiner points. *SIAM Journal on Computing*, 44(4):975–995, 2015.

9 Robert Krauthgamer, Huy Nguyên, and Tamar Zondiner. Preserving terminal distances using minors. *SIAM Journal on Discrete Mathematics*, 28(1):127–141, 2014. doi:10.1137/120888843.

10 Robert Krauthgamer and Tamar Zondiner. Preserving terminal distances using minors. In *Automata, Languages, and Programming*, volume 7391 of *Lecture Notes in Computer Science*, pages 594–605. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-31594-7_50.

11 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. doi:10.1002/jgt.3190130114.

12 Mihai Pătraşcu and Erik D Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing*, 35(4):932–963, 2006.