Combinatorica (2024) 44:417–427
https://doi.org/10.1007/s00493-023-00077-w

ORIGINAL PAPER

Tight Bound on Treedepth in Terms of Pathwidth and Longest Path

Meike Hatzel[1] · Gwenaël Joret[2] · Piotr Micek[3] · Marcin Pilipczuk[4] · Torsten Ueckerdt[5] · Bartosz Walczak[3]

Received: 6 February 2023 / Revised: 7 October 2023 / Accepted: 27 November 2023 /
Published online: 19 December 2023

Abstract

We show that every graph with pathwidth strictly less than a that contains no path on 2^b vertices as a subgraph has treedepth at most $10ab$. The bound is best possible up to a constant factor.

Keywords Treewidth · Pathwidth · Treedepth

1 Introduction

Treewidth (tw), pathwidth (pw), and treedepth (td) are among the best-known and most widely studied structural width parameters of graphs. They are related by the inequalities $\mathrm{tw}(G) + 1 \leqslant \mathrm{pw}(G) + 1 \leqslant \mathrm{td}(G)$ for every graph G. Moreover, trees have treewidth 1 and arbitrarily large pathwidth, while paths have pathwidth 1 and arbitrarily large treedepth.

Treedepth is approximated by the maximum length of a path[1]: every graph containing an ℓ-vertex path has treedepth greater than $\log_2 \ell$, and every graph with no such path has treedepth less than ℓ [7, Sect. 6]. Similarly, pathwidth is approximated by the

[1] In this paper, we are concerned only about non-induced paths.

M. Hatzel was supported by the Federal Ministry of Education and Research (BMBF) and by a fellowship within the IFI programme of the German Academic Exchange Service (DAAD). G. Joret is supported by a CDR grant from the Belgian National Fund for Scientific Research (FNRS), a PDR grant from FNRS, and by the Wallonia Brussels International (WBI) agency. P. Micek is supported by the National Science Center of Poland under grant no. 2018/31/G/ST1/03718. B. Walczak is partially supported by the National Science Center of Poland under grant no. 2019/34/E/ST6/00443. The research of M. Pilipczuk is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme Grant Agreement 714704.

✉ Gwenaël Joret
gwenael.joret@ulb.be

Extended author information available on the last page of the article

maximum height of a complete binary tree minor: every graph containing a complete binary tree of height h as a minor has pathwidth at least $\lfloor \frac{h}{2} \rfloor$ [9], and every graph with no such minor has pathwidth $\mathcal{O}(2^h)$ [1]. For both parameters, the exponential gap between the respective lower and upper bounds cannot be avoided, as witnessed by complete graphs. Treewidth is approximated by the maximum size of a grid minor, but here the gap is polynomial: while every graph containing a $k \times k$ grid as a minor has treewidth at least k [8], every graph with no such minor has treewidth polynomial in k [2].

Kawarabayashi and Rossman [6] showed that treedepth is approximated with polynomial gap by the three above-mentioned obstructions together: every graph with no $k \times k$ grid minor, no height k complete binary tree minor, and no 2^k-vertex path has treedepth polynomial in k. More specifically, they proved that every graph of treewidth less than k with no height k complete binary tree minor and no 2^k-vertex path has treedepth $\mathcal{O}(k^5 \log^2 k)$. Here are an improvement of this statement and an analogous result relating pathwidth and treewidth:

Theorem 1 (Czerwiński, Nadara, Pilipczuk [3][2]) *Every graph of treewidth less than t with no complete binary tree of height h as a minor and no 2^b-vertex path has treedepth $\mathcal{O}(thb)$.*

Theorem 2 (Groenland, Joret, Nadara, Walczak [5]) *Every graph of treewidth less than t with no complete binary tree of height h as a minor has pathwidth $\mathcal{O}(th)$.*

We complete the picture by proving an analogous result relating treedepth and pathwidth.

Theorem 3 *Every graph of pathwidth less than a containing no 2^b-vertex path has treedepth at most $10ab$.*

Clearly, Theorems 2 and 3 imply Theorem 1. On the other hand, Theorem 1 implies that every graph of pathwidth less than a containing no 2^b-vertex path has treedepth $\mathcal{O}(a^2 b)$. This is because every graph with pathwidth less than a has treewidth less than a and contains no complete binary tree of height $2a$ as a minor. In [5], it was conjectured that the bound on treedepth can be reduced to $\mathcal{O}(ab)$, and Theorem 3 provides a proof of this conjecture.

We remark that the bound in Theorem 3 is sharp up to a constant factor, which can be seen as follows. Let b and c be integers with $b > c \geqslant 1$, and let $a = 2^c$. Consider the graph G obtained from a path on 2^{b-c} vertices by replacing each vertex with a clique on $\frac{a}{2} = 2^{c-1}$ vertices and replacing each edge by a complete bipartite graph between the two cliques. Then $\mathrm{pw}(G) = a - 1$. Also, G has 2^{b-1} vertices, and thus it has no 2^b-vertex path. It can be checked that G has treedepth at least $\frac{a}{2}(b - c)$, which is roughly $ab/2$ when $b \gg c$. It is shown in [5] that the bound in Theorem 2 is also sharp up to a constant factor. Whether the bound in Theorem 1 can be improved remains an open problem.

[2] In [3], the bound is stated in the special case $t = h = b$, but the proof works in general.

2 Preliminaries

All graphs in this paper are finite and simple, that is, they have no loops or parallel edges. All logarithms in this paper are to the base 2.

A *rooted tree* is a tree with one vertex designated as the *root*. A *rooted forest* is a disjoint union of rooted trees. We define the *height* of a rooted forest F as the maximum number of vertices on a path from a root to a leaf in F. A vertex u is an *ancestor* of a vertex v in a rooted forest F if u lies on the (unique) path from a root to v in F. A rooted forest F is an *elimination forest* of a graph G if $V(F) = V(G)$ and for every edge uv of G, one of the vertices u and v is an ancestor of the other in F. The *treedepth* of a graph G, denoted by $\mathrm{td}(G)$, is the minimum height of an elimination forest of G.

A *tree decomposition* of a graph G is a pair $(T, \mathcal{B})$ such that T is a tree, the vertices of which are called *nodes*, and $\mathcal{B}$ is a collection $\{B_t\}_{t\in V(T)}$ of subsets of $V(G)$, called *bags*, indexed by the nodes of T, such that the following conditions are satisfied:

(1) for every edge $uv \in E(G)$, there is a bag containing both u and v;
(2) for every vertex $v \in V(G)$, the set of nodes $t \in V(T)$ with $v \in B_t$ induces a non-empty subtree of T.

The *width* of a tree decomposition is the maximum size of a bag minus 1. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width of a tree decomposition of G. The notions of *path decomposition* and *pathwidth* are defined analogously with the extra condition that the tree T is a path. The pathwidth of G is denoted by $\mathrm{pw}(G)$.

A *k-linkage* between two subsets A and B of the vertices of a graph G is a subgraph of G that consists of k vertex-disjoint paths each starting in A and ending in B. (If A and B intersect, then a path of a k-linkage between A and B may consist of a single vertex in $A \cap B$.) A path decomposition $(P, \mathcal{B})$ of a graph G is *linked* if for any two nodes $t, t' \in V(P)$, there is a k-linkage between B_t and $B_{t'}$ where k is the minimum size of a bag B_s for nodes s on the path from t to t' in P. We use the fact that there is always a path decomposition of minimum width that is linked.

Theorem 4 (Erde [4, Theorem 5.8]) *Every graph G has a path decomposition of width $\mathrm{pw}(G)$ that is linked.*

3 Proof

We proceed with the proof of Theorem 3, that every graph of pathwidth less than a containing no 2^b-vertex path has treedepth at most $10ab$.

Let G be a graph with $\mathrm{pw}(G) < a$ and with no path on 2^b vertices. If $2^b < 2a$, then the statement of the theorem is easily seen to hold by considering a depth-first search forest of G, which is an elimination forest of G. Its height is less than 2^b, which is less than $2a$. Hence, $\mathrm{td}(G) < 2a < 10ab$. Therefore, we may assume that $2^b \geqslant 2a$. This inequality will be used at the very end of the proof.

Fix a linked path decomposition $(P, \mathcal{B})$ of G with $\mathcal{B} = \{B_t\}_{t\in V(P)}$ and $|B_t| \leqslant a$ for every node $t \in V(P)$; such a linked path decomposition exists by Theorem 4. We think of the nodes as being laid out from left to right along P. For a set of nodes $X \subseteq V(P)$, let

$$B(X) = \bigcup_{t \in X} B_t.$$

We call the node set of any subpath of P an *interval*. For an interval I, we let

$$\text{level}(I) = \min\{|B_t| \mid t \in I\} \qquad \text{and} \qquad \text{int}(I) = B(I) - B(V(P) - I).$$

Thus, $\text{int}(I)$ (the "interior" of I) is the set of vertices of G that lie only in the bags of nodes in I.

For every $k \in \{1, \dots, a\}$ and every inclusion-maximal interval I^* with $\text{level}(I^*) \geqslant k$, we fix some k-linkage between the bags of the leftmost and the rightmost nodes in I^*, and we let $\mathcal{L}_k^*(I^*)$ be the vertex set of that k-linkage. For every $k \in \{1, \dots, a\}$ and every interval I with $\text{level}(I) \geqslant k$, we let $\mathcal{L}_k(I) = \mathcal{L}_k^*(I^*) \cap B(I)$ where I^* is the unique inclusion-maximal interval with $\text{level}(I^*) \geqslant k$ containing I. We note the following properties of the sets $\mathcal{L}_k(I)$ for further reference:

$$\mathcal{L}_k(I) \subseteq B(I), \tag{1}$$

$$\mathcal{L}_k(I') \subseteq \mathcal{L}_k(I) \qquad \text{for every interval } I' \subseteq I, \tag{2}$$

$$|\mathcal{L}_k(I) \cap B_t| \geqslant k \qquad \text{for every node } t \in I, \tag{3}$$

$$|\mathcal{L}_k(I)| < k \cdot 2^b. \tag{4}$$

We describe an iterative process in which we construct a rooted tree T whose vertices are contained in $V(G)$ except for the root, which is a special vertex $r^* \notin V(G)$. The initial tree T contains only the root r^*. We grow the tree T in rounds, in each round attaching new paths formed by some vertices of G that are not yet in T. We maintain the invariant that $T - r^*$ is an elimination forest of the corresponding induced subgraph of G, that is, for any two vertices in $V(T) - \{r^*\}$ that are adjacent in G, one is an ancestor of the other in T. The process ends when T contains all vertices of G, so that $T - r^*$ is an elimination forest of G.

A simple plan for a round would be to find a bag B_t whose removal from G would halve some measure that is proportional to the logarithm of the maximum path length. Then, after adding B_t to T (as a path), we could continue growing T independently on each of the two sides of $G - B_t$ starting from the vertex of B_t that is currently a leaf of T. This is too simple to work, but it motivates our actual approach.

For a tree T as above and an interval I, we use the following notation. Let $\ell = \text{level}(I)$. For every $k \in \{1, \dots, \ell\}$, we define

$$x_k(I, T) = |(\text{int}(I) \cap \mathcal{L}_k(I)) - V(T)| \qquad \text{and} \qquad w_k(I, T) = \sum_{i=1}^{k} \log(x_i(I, T) + 1).$$

The following "monotonicity" property is a direct consequence of (2):

$$\text{if } I' \subseteq I \text{ and } V(T') \supseteq V(T), \text{ then } x_k(I', T') \leqslant x_k(I, T) \text{ and } w_k(I', T') \leqslant w_k(I, T), \tag{5}$$

Furthermore, it follows from (4) that $x_i(I, T) + 1 \leqslant i \cdot 2^b$ for every $i \in \{1, \ldots, \ell\}$, which yields

$$w_\ell(I, T) \leqslant \sum_{i=1}^{\ell} \log(i \cdot 2^b) = b\ell + \log(\ell!), \tag{6}$$

$$w_\ell(I, T) - w_k(I, T) \leqslant \sum_{i=k+1}^{\ell} \log(i \cdot 2^b) = b(\ell - k) + \log\left(\frac{\ell!}{k!}\right) \quad \text{for every } k \in \{1, \ldots, \ell\}. \tag{7}$$

For notational convenience, we also define $w_i(\emptyset, T) = 0$.

For a vertex v of G that has been added to T at some time in the process, let depth(v) denote the number of vertices of G on the path from r^* to v in T (thus disregarding r^* in the count). Since we always augment the tree T by adding new vertices as leaves, depth(v) is determined when v is added to T and remains unchanged till the end of the process.

During the aforesaid iterative process of constructing the tree T, we maintain

- a set X of nodes of P, and the invariant that $B(X) \subseteq V(T)$;
- the family $\mathcal{I}$ of intervals contained in $V(P) - X$ that are inclusion-maximal;
- a designated vertex v_I in T for every interval $I \in \mathcal{I}$.

We also maintain the following invariants:

Inv. 1. For every interval $I \in \mathcal{I}$, the path from the root r^* to v_I in T contains every vertex of $B(I) \cap V(T)$.

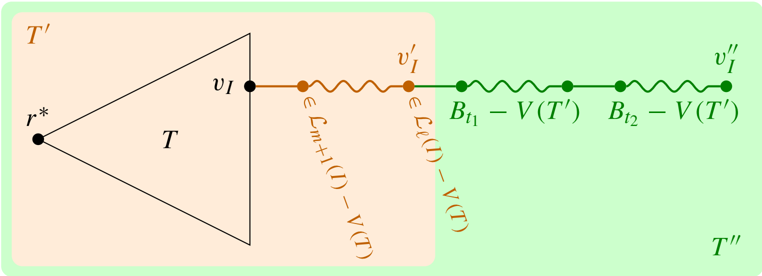
Inv. 2. For every interval $I \in \mathcal{I}$ and for $\ell = \text{level}(I)$, we have

$$\tfrac{1}{5}\,\text{depth}(v_I) + w_\ell(I, T) \leqslant (b+1)\ell + \log(\ell!).$$

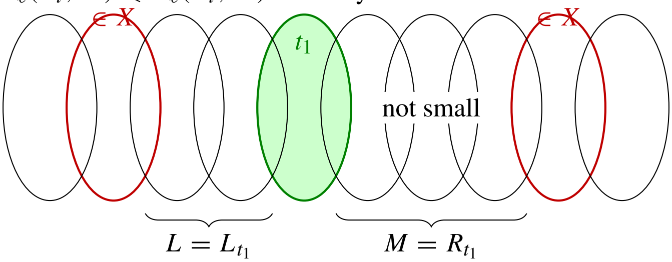
The former allows us to maintain the invariant that $T - r^*$ is an elimination forest of the corresponding induced subgraph of G, and the latter helps us bound the height of T.

Initially, the tree T contains only the root r^*, the set X is empty, $\mathcal{I} = \{I\}$ where $I = V(P)$, and $v_I = r^*$. For this initial setup, Inv. 1 holds by the fact that $B(I) \cap V(T) = \emptyset$, whereas Inv. 2 holds by (6) and the fact that depth$(v_I) = 0$.

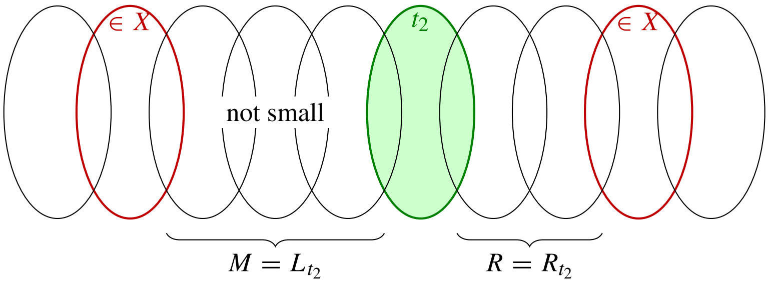
Every round consists in choosing an arbitrary interval $I \in \mathcal{I}$ and adding one or two nodes of I to X. As a result, the interval I is replaced in $\mathcal{I}$ by at most three of its proper subintervals.



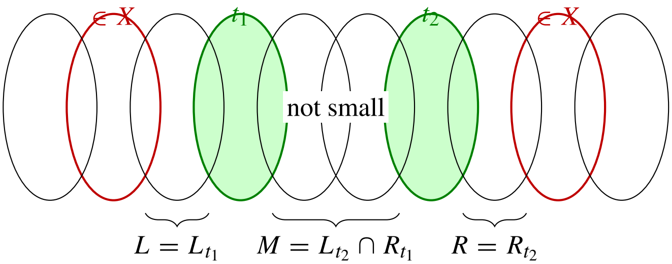
if $w_\ell(L_t, T') \leqslant w_\ell(R_t, T')$ for every small node $t \in I$:



if $w_\ell(L_t, T') > w_\ell(R_t, T')$ for every small node $t \in I$:



otherwise:



Now, we describe the details of a single round. Pick an interval $I \in \mathcal{I}$. Let

$$\ell = \text{level}(I) \qquad \text{and} \qquad m = \max(\{0\} \cup \{i \in \{1, \dots, \ell\} \mid \mathcal{L}_i(I) - V(T) = \emptyset\}).$$

It follows that

$$\text{for every node } t \in I, \text{ at least } m \text{ vertices of } B_t \text{ are already in } T. \tag{8}$$

Indeed, this is true if $m = 0$, and if $m > 0$, this follows from (3) and the fact that all vertices of $\mathcal{L}_m(I)$ are already in T. For every $i \in \{1, \dots, \ell\}$, since every vertex in $\mathcal{L}_i(I) - \text{int}(I)$ belongs to the bag of a neighbor of I in P, which belongs to X, we have

$$\mathcal{L}_i(I) - V(T) \subseteq \mathcal{L}_i(I) - B(X) \subseteq \text{int}(I). \tag{9}$$

Let $k = \ell - m$. (We may have $m = \ell$ and $k = 0$.) For every $i \in \{m + 1, \dots, \ell\}$, by (9), we have $x_i(I, T) = |\mathcal{L}_i(I) - V(T)| \geqslant 1$. It follows that

$$w_\ell(I, T) = \sum_{i=1}^{\ell} \log(x_i(I, T) + 1) \geqslant \sum_{i=m+1}^{\ell} \log(x_i(I, T) + 1) \geqslant \ell - m = k. \tag{10}$$

Choose one vertex from the set $\mathcal{L}_i(I) - V(T)$ for each $i \in \{m + 1, \dots, \ell\}$, and add these at most k vertices into T as a path with one end attached to v_I. That is, the first vertex is added as a child of v_I and every further vertex is added as a child of the previous one. Let v'_I be the last such vertex (i.e., the other end of the path) if $k \geqslant 1$, and let $v'_I = v_I$ if $k = 0$. Let T' denote the resulting augmented tree.

Call a node $t \in I$ *small* if $|B_t| \leqslant \ell + k$. By the definition of level(I), at least one node in I is small. It follows from (8) that

$$\text{for each small node } t \in I,\ \text{at most } \ell + k - m = 2k \text{ vertices of } B_t \text{ are not yet in } T'. \tag{11}$$

Recall that we think of I as ordered from left to right. For every node $t \in I$, let L_t and R_t denote the sets of nodes of I to the left and to the right of t, respectively, so that $I = L_t \cup \{t\} \cup R_t$. If $w_\ell(L_t, T') \leqslant w_\ell(R_t, T')$ for every small node $t \in I$, then let t_1 be the rightmost small node in I, and let $L = L_{t_1}$ and $M = R_{t_1}$. Similarly, if $w_\ell(L_t, T') > w_\ell(R_t, T')$ for every small node $t \in I$, then let t_2 be the leftmost small node in I, and let $M = L_{t_2}$ and $R = R_{t_2}$. Otherwise, let t_1 be the rightmost small node in I such that $w_\ell(L_{t_1}, T') \leqslant w_\ell(R_{t_1}, T')$ and t_2 be the leftmost small node in I such that $w_\ell(L_{t_2}, T') > w_\ell(R_{t_2}, T')$. In this case, by (5), t_1 and t_2 occur in this order from left to right, and there are no small nodes between them. Now, let $L = L_{t_1}$, $R = R_{t_2}$, and $M = R_{t_1} \cap L_{t_2}$ (i.e., M is the set of nodes strictly between t_1 and t_2). See the figure.

Whenever t_1 and t_2 are defined, we add them to X. We remove I from $\mathcal{I}$, and whenever L, M, and R are defined and non-empty, we add them as new intervals to $\mathcal{I}$.

Now, we add the vertices of $B_{t_1} - V(T')$ and $B_{t_2} - V(T')$ (whenever t_1 or t_2 are defined) to T' as one path with one end attached to v'_I. That is, the first such vertex is a child of v'_I, and every further vertex is a child of the previous. Note that possibly both sets $B_{t_1} - V(T')$ and $B_{t_2} - V(T')$ are empty; in particular, this happens when $k = 0$. Let v''_I be the last vertex added if at least one vertex was added, and let $v''_I = v'_I$ otherwise. Let T'' denote the new tree. By (11), we have added at most $4k$ additional vertices, so

$$\text{depth}(v_I'') \leqslant \text{depth}(v_I') + 4k = \text{depth}(v_I) + 5k. \tag{12}$$

Whenever L, M, or R is defined and non-empty, we set the corresponding vertex v_L, v_M, or v_R to be v_I''. By Inv. 1 for I, it follows that the path from r^* to v_I'' contains every vertex of $B(I) \cap V(T'')$, which yields Inv. 1 for L, M, and R (when they are defined and non-empty).

Before verifying Inv. 2, let us capture the key properties of L, M, and R. If L is defined and non-empty (so that $L = L_{t_1}$), then let $\overline{L} = R_{t_1}$. If R is defined and non-empty (so that $R = R_{t_2}$), then let $\overline{R} = L_{t_2}$. Whenever the respective sets are defined, we have

$$w_\ell(L, T') \leqslant w_\ell(\overline{L}, T') \quad \text{and} \quad w_\ell(R, T') \leqslant w_\ell(\overline{R}, T'), \tag{13}$$

$$w_\ell(M, T'') \leqslant w_\ell(I, T), \tag{14}$$

$$\text{level}(M) \geqslant \ell + k + 1, \tag{15}$$

where (14) follows from (5), and (15) follows as there are no small nodes in M.

While a bound analogous to (14) holds also for $w_\ell(L, T'')$ and $w_\ell(R, T'')$, we need a stronger one. First we focus on the interval L, and the argument is symmetric for the interval R. For $i \in \{1, \dots, \ell\}$, we compare $x_i(I, T)$ with $x_i(L, T')$ and $x_i(\overline{L}, T')$. Note that $\text{int}(L)$ and $\text{int}(\overline{L})$ are vertex-disjoint and are both contained in $\text{int}(I)$. For each $i \in \{1, \dots, \ell\}$, we have $\mathcal{L}_i(L) \subseteq \mathcal{L}_i(I)$ and $\mathcal{L}_i(\overline{L}) \subseteq \mathcal{L}_i(I)$, by (2). For each $i \in \{m+1, \dots, \ell\}$, we have put one vertex of $\mathcal{L}_i(I) - V(T)$ into T'; this vertex belongs to $\text{int}(I)$ by (9). This, the property (1), and the definition of x_i imply that for each $i \in \{1, \dots, \ell\}$, we have

$$x_i(I, T) \geqslant \begin{cases} x_i(L, T') + x_i(\overline{L}, T') & \text{if } i \leqslant m, \\ x_i(L, T') + x_i(\overline{L}, T') + 1 & \text{if } i > m. \end{cases}$$

Since $x_i(L, T')$ and $x_i(\overline{L}, T')$ are non-negative, the above implies

$$\begin{aligned} & x_i(I, T) + 1 \\ & \quad \geqslant \begin{cases} x_i(L, T') + x_i(\overline{L}, T') + 1 \geqslant \frac{1}{2}(x_i(L, T') + x_i(\overline{L}, T') + 2) & \text{if } i \leqslant m, \\ x_i(L, T') + x_i(\overline{L}, T') + 2 & \text{if } i > m. \end{cases} \end{aligned} \tag{16}$$

Recalling that $k = \ell - m$, we calculate

$$\begin{aligned} w_\ell(I, T) &= \sum_{i=1}^{\ell} \log(x_i(I, T) + 1) \\ &\geqslant \sum_{i=1}^{\ell} \log(x_i(L, T') + x_i(\overline{L}, T') + 2) - m && \text{by (16)} \end{aligned}$$

$$\begin{aligned}
&\geqslant \sum_{i=1}^{\ell} \frac{\log(x_i(L,T')+1)+\log(x_i(\overline{L},T')+1)}{2} + \ell - m && (*)\\
&= \tfrac{1}{2}(w_\ell(L,T') + w_\ell(\overline{L},T')) + k \\
&\geqslant w_\ell(L,T') + k && \text{by (13)}\\
&\geqslant w_\ell(L,T'') + k && \text{by (5)},
\end{aligned}$$

where in $(*)$, we use the inequality $\log(x+y) = \log\frac{x+y}{2} + 1 \geqslant \frac{1}{2}(\log x + \log y) + 1$ that follows from the concavity of log. From this and the analogous argument for R, we conclude that

$$w_\ell(L,T'') + k \leqslant w_\ell(I,T) \quad \text{and} \quad w_\ell(R,T'') + k \leqslant w_\ell(I,T). \tag{17}$$

Now, we are set to verify Inv. 2 for the intervals L, R, and M (when they are defined and non-empty). We have

$$\begin{aligned}
&\tfrac{1}{5}\operatorname{depth}(v_L) + w_{\text{level}(L)}(L,T'')\\
&\quad\leqslant \tfrac{1}{5}\operatorname{depth}(v_I) + k + w_\ell(L,T'') + b(\text{level}(L) - \ell)\\
&\qquad + \log\left(\frac{\text{level}(L)!}{\ell!}\right) && \text{by (12) and (7)}\\
&\quad\leqslant \tfrac{1}{5}\operatorname{depth}(v_I) + w_\ell(I,T) + b(\text{level}(L) - \ell)\\
&\qquad + \log\left(\frac{\text{level}(L)!}{\ell!}\right) && \text{by (17)}\\
&\quad\leqslant (b+1)\ell + \log(\ell!) + b(\text{level}(L) - \ell)\\
&\qquad + \log\left(\frac{\text{level}(L)!}{\ell!}\right) && \text{by Inv. 2 for } I\\
&\quad\leqslant (b+1)\,\text{level}(L) + \log(\text{level}(L)!).
\end{aligned}$$

The exact same bounds hold with L replaced by R. Finally, for M, we have

$$\begin{aligned}
&\tfrac{1}{5}\operatorname{depth}(v_M) + w_{\text{level}(M)}(M,T'')\\
&\quad\leqslant \tfrac{1}{5}\operatorname{depth}(v_I) + k + w_\ell(M,T'') + b(\text{level}(M) - \ell)\\
&\qquad + \log\left(\frac{\text{level}(M)!}{\ell!}\right) && \text{by (12) and (7)}\\
&\quad\leqslant \tfrac{1}{5}\operatorname{depth}(v_I) + k + w_\ell(I,T) + b(\text{level}(M) - \ell)\\
&\qquad + \log\left(\frac{\text{level}(M)!}{\ell!}\right) && \text{by (14)}\\
&\quad\leqslant (b+1)\ell + \log(\ell!) + k + b(\text{level}(M) - \ell)\\
&\qquad + \log\left(\frac{\text{level}(M)!}{\ell!}\right) && \text{by Inv. 2 for } I
\end{aligned}$$

$$
\begin{aligned}
&\leqslant (b+1)\ell + \operatorname{level}(M) - \ell + b(\operatorname{level}(M) - \ell) \\
&\quad + \log(\operatorname{level}(M)!) && \text{by (15)} \\
&= (b+1)\operatorname{level}(M) + \log(\operatorname{level}(M)!).
\end{aligned}
$$

This completes the round of our process for the interval I, with T'' becoming the new tree T. We have shown that both invariants, Inv. 1 and Inv. 2 are preserved.

The process ends when all vertices of G have been added to T. It remains to show that $T - r^*$ is an elimination forest of G with height at most $10ab$. To see that it is an elimination forest, observe that whenever a vertex $v \in \operatorname{int}(I)$ is added to T when considering an interval I, Inv. 1 guarantees that all neighbors of v in G that are already in T lie on the path from r^* to v in T, as the neighbors of v in G belong to $B(I)$ by the definition of path decomposition.

The height of the forest $T - r^*$ is equal to $\max\{\operatorname{depth}(v) \mid v \in V(G)\}$. Let v be a vertex of G. Consider the moment in the process when v has been added to T. Say, it happened when processing an interval I with $\operatorname{level}(I) = \ell$. Let m and k be the values fixed when processing I. Clearly, we have $\operatorname{depth}(v) \leqslant \operatorname{depth}(v_I'')$ and $\ell \leqslant a$. Therefore,

$$
\begin{aligned}
\operatorname{depth}(v) &\leqslant \operatorname{depth}(v_I'') \\
&\leqslant \operatorname{depth}(v_I) + 5k && \text{by (12)} \\
&\leqslant \operatorname{depth}(v_I) + 5w_\ell(I, T) && \text{by (10)} \\
&\leqslant 5(b+1)\ell + 5\log(\ell!) && \text{by Inv. 2} \\
&\leqslant 5(b+1)a + 5\log(a!) \\
&\leqslant 5ab + 5a\log a + 5a.
\end{aligned}
$$

Recall that $2^b \geqslant 2a$ and thus $b \geqslant \log(2a) = \log a + 1$. It follows that

$$\operatorname{depth}(v) \leqslant 5ab + 5a\log a + 5a \leqslant 10ab.$$

We conclude that $T - r^*$ is an elimination forest of G with height at most $10ab$, as desired.

Acknowledgements This research was carried out at the "Sparsity in Algorithms, Combinatorics, and Logic" workshop held in Dagstuhl in September 2021. We thank the organizers and the workshop participants for creating a productive working atmosphere. In particular, we thank Marthe Bonamy, Marcin Briański, Kord Eickmeyer, Wojciech Nadara, Ben Rossman, Blair D. Sullivan, and Alexandre Vigny for initial discussions on the topic of the paper. We also thank an anonymous reviewer for noticing an error in a previous version of the paper.

References

1. Bienstock, D., Robertson, N., Seymour, P., Thomas, R.: Quickly excluding a forest. J. Comb. Theory Ser. B **52**(2), 274–283 (1991)
2. Chekuri, C., Chuzhoy, J.: Polynomial bounds for the grid-minor theorem. J. ACM **63**(5), Article 40, (2016). arXiv:1305.6577

3. Czerwiński, W., Nadara, W., Pilipczuk, M.: Improved bounds for the excluded-minor approximation of treedepth. SIAM J. Discret. Math. **35**(2), 934–947 (2021). arXiv:1904.13077
4. Erde, J.: A unified treatment of linked and lean tree-decompositions. J. Comb. Theory Ser. B **130**, 114–143 (2018). arXiv:1703.03756
5. Groenland, C., Joret, G., Nadara, W., Walczak, B.: Approximating pathwidth for graphs of small treewidth. ACM Trans. Algorithms **19**(2), Article 16 (2023). arXiv:2008.00779
6. Kawarabayashi, K., Rossman, B.: A polynomial excluded-minor approximation of treedepth. J. Eur. Math. Soc. **24**(4), 1449–1470 (2022)
7. Nešetřil, J., de Mendez, P.O.: Sparsity, volume 28 of Algorithms and Combinatorics. Springer, Berlin (2012)
8. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. J. Comb. Theory Ser. B **41**(1), 92–114 (1986)
9. Scheffler, P.: Die Baumweite von Graphen als ein Maß für die Kompliziertheit algorithmischer Probleme. PhD thesis, Akademie der Wissenschaften der DDR, Berlin, (1989)

Authors and Affiliations

Meike Hatzel[1] · Gwenaël Joret[2] · Piotr Micek[3] · Marcin Pilipczuk[4] · Torsten Ueckerdt[5] · Bartosz Walczak[3]

Meike Hatzel
meikehatzel@nii.ac.jp

Piotr Micek
piotr.micek@uj.edu.pl

Marcin Pilipczuk
malcin@mimuw.edu.pl

Torsten Ueckerdt
torsten.ueckerdt@kit.edu

Bartosz Walczak
bartosz.walczak@uj.edu.pl

[1] National Institute of Informatics, Tokyo, Japan

[2] Computer Science Department, Université libre de Bruxelles, Brussels, Belgium

[3] Department of Theoretical Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

[4] Institute of Informatics, University of Warsaw, Warsaw, Poland

[5] Computer Science Department, Karlsruhe Institute of Technology, Karlsruhe, Germany