Algorithmica (2017) 78:342–377
DOI 10.1007/s00453-016-0164-5

Linear Rank-Width of Distance-Hereditary Graphs I. A Polynomial-Time Algorithm

Isolde Adler[1] · **Mamadou Moustapha Kanté**[2] · **O-joung Kwon**[3,4]

Received: 21 August 2015 / Accepted: 7 May 2016 / Published online: 24 May 2016
© Springer Science+Business Media New York 2016

Abstract Linear rank-width is a linearized variation of rank-width, and it is deeply related to matroid path-width. In this paper, we show that the linear rank-width of every n-vertex distance-hereditary graph, equivalently a graph of rank-width at most 1, can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$, and a linear layout witnessing the linear rank-width can be computed with the same time complexity. As a corollary, we show that the path-width of every n-element matroid of branch-width at most 2 can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$, provided that the matroid is given by its binary representation. To establish this result, we present a characterization of the linear rank-width of distance-hereditary graphs in terms of their canonical split decompositions. This characterization is similar to the known characterization of the path-width of forests

The first author is supported by the German Research Council, Project GalA, AD 411/1-1. The third author is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2011-0011653). A preliminary version appeared in the proceedings of WG'14.

✉ O-joung Kwon
ojoungkwon@gmail.com

Isolde Adler
I.M.Adler@leeds.ac.uk

Mamadou Moustapha Kanté
mamadou.kante@isima.fr

1 School of Computing, University of Leeds, Leeds, UK

2 Université Clermont Auvergne, Université Blaise Pascal, LIMOS, CNRS, Aubière, France

3 Department of Mathematical Sciences, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, South Korea

4 Present Address: Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Kende u.13-17, Budapest, Hungary

given by Ellis, Sudborough, and Turner [The vertex separation and search number of a graph. *Inf. Comput.*, 113(1):50–79, 1994]. However, different from forests, it is non-trivial to relate substructures of the canonical split decomposition of a graph with some substructures of the given graph. We introduce a notion of 'limbs' of canonical split decompositions, which correspond to certain vertex-minors of the original graph, for the right characterization.

Keywords Rank-width · Linear rank-width · Distance-hereditary graphs · Vertex-minors · Matroid branch-width · Matroid path-width

1 Introduction

Rank-width [28] is a graph parameter introduced by Oum and Seymour with the goal of efficient approximation of the *clique-width* [6] of a graph. *Linear rank-width* can be seen as the linearized variant of rank-width, and it is similar to path-width, which can be seen as the linearized variant of tree-width. While path-width is a well-studied notion, much less is known about linear rank-width. Vertex-minor is a graph containment relation where rank-width and linear rank-width do not increase when taking this operation.

Rank-width is related to matroid branch-width, which has an important role in structural theory on matroids. We refer to the series of papers by Geelen, Gerards, and Whittle on the Matroid Minors Project [14,15] and Rota's Conjecture [16] for more information on matroid branch-width. It is known that the matroid branch-width (matroid path-width) of a binary matroid is equal to the rank-width (linear rank-width) of its fundamental graph plus one [27]. This equality can be further generalized to matroids over a fixed finite field with the finite field version of rank-width [24,25]. Hence new results on (linear) rank-width will immediately yield new results on matroid branch-width or on matroid path-width. In this paper, we will derive a complexity result for computing matroid path-width from linear rank-width.

Kashyap [22] showed that it is NP-hard to compute matroid path-width on binary matroids. By reducing from matroid path-width, we can show that computing linear rank-width is NP-hard in general. But, fixed parameter tractable algorithms, parametrized by the linear rank-width, for testing whether the linear rank-width of a graph is at most k exist. First, as the class of graphs of linear rank-width at most k, for fixed k, is closed under taking vertex-minors, from the well-quasi-ordering theorem by Oum [27], the class of graphs of linear rank-width at most k is characterized by a finite set of forbidden vertex-minors. Since one can check in time $\mathcal{O}(f(\ell, h) \cdot n^3)$ whether a fixed graph H on h vertices is isomorphic to a vertex-minor of a given graph of rank-width at most ℓ [7], we can deduce that, for any fixed k, one can check whether a given graph has linear rank-width at most k in time $\mathcal{O}(g(k) \cdot n^3)$. But, as we need to construct the set of forbidden vertex-minors and we do not know a bound on their size, the above algorithm is non-constructive. Recently, Jeong et al. [19] showed that, for fixed k, there is a constructive algorithm to test whether a given graph has linear rank-width at most k in time $\mathcal{O}(f(k) \cdot n^3)$.

It is natural to ask which graph classes allow for an efficient computation. Adler and Kanté [18] first showed that it is possible to compute the linear rank-width of forests in linear time. As Bodlaender and Kloks [4] showed that it is possible to compute the path-width of graphs of bounded tree-width in polynomial time, one can ask whether it is also possible to compute the linear rank-width of graphs of bounded rank-width in polynomial time. This question was very recently settled by Jeong et al. [19], but the exponent on n in the running time is not realistic and depends on the rank-width. To the best of our knowledge, the existence of such an algorithm that runs in time $\mathcal{O}(n^3)$ or even better is an open question.[1]

Our main result is an $\mathcal{O}(n^2 \cdot \log_2 n)$-time algorithm to compute the linear rank-width of a distance-hereditary graph, and a linear layout witnessing its linear rank-width. A graph G is *distance-hereditary* if for every pair of two vertices u and v of G, the distance between u and v in any connected induced subgraph of G containing both u and v, is the same as the distance between u and v in G. Distance-hereditary graphs are exactly graphs of rank-width at most 1 [27], and include all forests and cographs.

Theorem 6.1 *The linear rank-width of every n-vertex distance-hereditary graph can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$. Moreover, a linear layout of the graph witnessing the linear rank-width can be computed with the same time complexity.*

In contrast, computing the path-width of distance-hereditary graphs is known to be NP-hard [23].

A direct consequence of Theorem 6.1 is the possibility to compute the path-width of matroids with branch-width at most 2 in polynomial time. It is known that every matroid of branch-width at most 2 is a binary matroid [10,30,31].

Corollary 7.4 *The path-width of every n-element matroid of branch-width at most 2 can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$, provided that the matroid is given by its binary representation. Moreover, a linear layout of the matroid witnessing the path-width can be computed with the same time complexity.*

The main ingredient of our algorithm is a new characterization of the linear rank-width of distance-hereditary graphs (Theorem 4.1). Our characterization makes use of the special structure of canonical split decompositions [8] of distance-hereditary graphs. Roughly, a canonical split decomposition decomposes a distance-hereditary graph in a tree-like fashion into complete graphs and stars, and our characterization is recursive along the sub-decompositions of the split decomposition.

While a similar idea has been exploited in [18,21,26] for other parameters, here we encounter a new problem. When we take a subgraph of a given split decomposition, the obtained split decomposition may have vertices that do not represent vertices of the original graph. It is not at all obvious how to deal with these vertices in the recursive step. We handle this by introducing *limbs* of canonical split decompositions, that correspond to certain vertex-minors of the original graphs, and have the desired properties to allow our characterization. We think that the notion of limbs may be useful in other contexts, too, and hopefully, it can be extended to other graph classes and allow for further new efficient algorithms.

The paper is structured as follows. Section 2 introduces the basic notions, in particular linear rank-width, vertex-minors, and split decompositions. In Sect. 3, we define

[1] At the time this paper was submitted, the algorithm in [19] was not even known.

limbs and its canonical decompositions, called canonical limbs, and show some basic properties. We use them in Sect. 4 for our characterization of the linear rank-width of distance-hereditary graphs. In Sect. 5, we establish essential properties of canonical limbs, which will be used to obtain the running time of our algorithm. Section 6 presents the $\mathcal{O}(n^2 \cdot \log_2 n)$-time algorithm for computing the linear rank-width of distance-hereditary graphs, and in Sect. 7, we obtain an algorithm for computing the path-width of matroids of branch-width at most 2 as a corollary. To obtain the running time, we need the fact that every n-vertex distance-hereditary graph G has linear rank-width at most $\log_2 n$. Generally, we prove in Sect. 8 that every graph of rank-width k has linear rank-width at most $k\lfloor \log_2 n \rfloor$.

2 Preliminaries

In this paper, graphs are finite, simple and undirected, unless stated otherwise. Our graph terminology is standard, see for instance [11]. Let G be a graph. We denote the vertex set of G by $V(G)$ and the edge set by $E(G)$. An edge between x and y is written xy (equivalently yx). For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G induced by X, and let $G\backslash X := G[V(G)\backslash X]$. For shortcut we write $G\backslash x$ for $G\backslash\{x\}$. For a vertex x of G, let $N_G(x)$ be the set of *neighbors* of x in G and we call $|N_G(x)|$ the *degree* of x in G. An edge e of G is called a *cut-edge* if its removal increases the number of connected components of G.

A *tree* is a connected acyclic graph. A *leaf* of a tree is a vertex of degree one. A *path* is a tree where every vertex has degree at most two. The *length* of a path is the number of its edges. A *star* is a tree with a distinguished vertex, called its *center*, adjacent to all other vertices. A *complete graph* is a graph with all possible edges. A graph G is called *distance-hereditary* if for every pair of two vertices x and y of G the distance of x and y in G equals the distance of x and y in any connected induced subgraph containing both x and y [2].

2.1 Linear Rank-Width and Vertex-Minors

For sets R and C, an (R, C)-*matrix* is a matrix whose rows and columns are indexed by R and C, respectively. For an (R, C)-matrix M, $X \subseteq R$, and $Y \subseteq C$, let $M[X, Y]$ be the submatrix of M whose rows and columns are indexed by X and Y, respectively.

Linear Rank-Width

Let G be a graph. We denote by A_G the *adjacency matrix* of G over the binary field. The *cut-rank function* of G is a function $\text{cutrk}_G : 2^{V(G)} \to \mathbb{Z}$ where for each $X \subseteq V(G)$,

$$\text{cutrk}_G(X) := \text{rank}(A_G[X, V(G)\backslash X]).$$

A sequence $(x_1, \ldots, x_n)$ of the vertex set $V(G)$ is called a *linear layout* of G. If $|V(G)| \geq 2$, then the *width* of a linear layout $(x_1, \ldots, x_n)$ of G is defined as

$$\max_{1 \leq i \leq n-1} \{\text{cutrk}_G(\{x_1, \ldots, x_i\})\}.$$

The *linear rank-width* of G, denoted by $\text{lrw}(G)$, is defined as the minimum width over all linear layouts of G if $|V(G)| \geq 2$, and otherwise, let $\text{lrw}(G) := 0$.

Caterpillars and complete graphs have linear rank-width at most 1. Ganian [12] characterized the graphs of linear rank-width at most 1, and called them *thread graphs*. Adler and Kanté [18] showed that linear rank-width and path-width coincide on forests, and therefore, there is a linear-time algorithm to compute the linear rank-width of forests. It is easy to see that the linear rank-width of a graph is the maximum over the linear rank-widths of its connected components.

To obtain the bound presented in Theorem 6.1, we will need the fact that the linear rank-width of an n-vertex distance-hereditary graph G is at most $\log_2 n$. In fact, we generally show that the linear rank-width of a graph with rank-width k is at most $k\lfloor \log_2 n \rfloor$. The proof scheme is similar to the one for path-width [3].

A tree is *subcubic* if it has at least two vertices and every internal vertex has degree 3. A *rank-decomposition* of a graph G is a pair (T, L), where T is a subcubic tree and L is a bijection from the vertices of G to the leaves of T. For an edge e in T, $T \setminus e$ induces a partition (X_e, Y_e) of the leaves of T. The *width* of an edge e is defined as $\text{cutrk}_G(L^{-1}(X_e))$. The *width* of a rank-decomposition (T, L) is the maximum width over all edges of T. The *rank-width* of G, denoted by $\text{rw}(G)$, is the minimum width over all rank-decompositions of G if $|V(G)| \geq 2$, and otherwise, let $\text{rw}(G) := 0$.

Theorem 2.1 (Oum *[27]*). *A graph is distance-hereditary if and only if it has rank-width at most 1.*

Lemma 2.2 *Let k be a positive integer and let G be a graph of rank-width k. Then* $\text{lrw}(G) \leq k\lfloor \log_2 |V(G)| \rfloor$.

Lemma 2.2 will be proved in Sect. 8.

Vertex-Minors

For a graph G and a vertex x of G, the *local complementation at x of G* is an operation to replace the subgraph induced by the neighbors of x with its complement. The resulting graph is denoted by $G * x$. If H can be obtained from G by applying a sequence of local complementations, then G and H are called *locally equivalent*. A graph H is called a *vertex-minor* of a graph G if H can be obtained from G by applying a sequence of local complementations and deletions of vertices.

Lemma 2.3 (Oum *[27]*). *Let G be a graph and let x be a vertex of G. Then for every subset X of $V(G)$, we have* $\text{cutrk}_G(X) = \text{cutrk}_{G*x}(X)$. *Therefore, every vertex-minor H of G satisfies that* $\text{lrw}(H) \leq \text{lrw}(G)$.

For an edge xy of G, let $W_1 := N_G(x) \cap N_G(y)$, $W_2 := (N_G(x) \setminus N_G(y)) \setminus \{y\}$, and $W_3 := (N_G(y) \setminus N_G(x)) \setminus \{x\}$. The *pivoting on xy* of G, denoted by $G \wedge xy$, is the operation to complement the adjacencies between distinct sets W_i and W_j, and swap the vertices x and y. It is known that $G \wedge xy = G * x * y * x = G * y * x * y$ [27]. See Fig. 1 for an example.

We introduce some basic lemmas on local complementations, which will be used in several places.

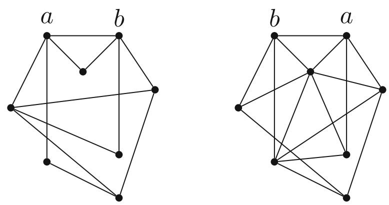


Fig. 1 Pivoting an edge *ab*

Lemma 2.4 *Let G be a graph and* $x, y \in V(G)$ *such that* $xy \notin E(G)$. *Then* $G*x*y = G*y*x$.

Proof It is straightforward as applying a local complementation at x or y does not change the neighbor sets of x and y. □

Lemma 2.5 *Let G be a graph and* $x, y, z \in V(G)$ *such that* $xy, xz \notin E(G)$ *and* $yz \in E(G)$. *Then* $G * x \wedge yz = G \wedge yz * x$.

Proof By the definition of pivoting, $G * x \wedge yz = G * x * y * z * y$. Note that $xy \notin E(G)$, $xz \notin E(G * y)$, and $xy \notin E(G * y * z)$. Therefore, by Lemma 2.4, $G*x*y*z*y = (G*y)*x*z*y = (G*y*z)*x*y = (G*y*z*y)*x = G \wedge yz*x$. □

Lemma 2.6 (Oum *[27]*). *Let G be a graph and* $x, y, z \in V(G)$ *such that* $xy, yz \in E(G)$. *Then* $G \wedge xy \wedge xz = G \wedge yz$.

2.2 Split Decompositions and Local Complementations

We will follow the definition of split decompositions in [5]. We notice that split decompositions are usually defined on connected graphs. For computing the linear rank-width of a distance-hereditary graph, it is enough to compute the linear rank-width of its connected components and take the maximum over all those values. Thus we will mostly assume that the given graph is connected in this paper, and use split decompositions in usual sense.

Let G be a connected graph. A *split* in G is a vertex partition (X, Y) of G such that $|X|, |Y| \geq 2$ and $\text{rank}(A_G[X, Y]) = 1$. In other words, (X, Y) is a split in G if $|X|, |Y| \geq 2$ and there exist non-empty sets $X' \subseteq X$ and $Y' \subseteq Y$ such that $\{xy \in E(G) \mid x \in X, y \in Y\} = \{xy \mid x \in X', y \in Y'\}$. Notice that not all connected graphs have a split, and those that do not have a split are called *prime* graphs.

A *marked graph* D is a connected graph D with a set of edges $M(D)$, called *marked edges*, that form a matching such that every edge in $M(D)$ is a cut-edge. The ends of the marked edges are called *marked vertices*, and the components of $(V(D), E(D)\setminus M(D))$ are called *bags* of D. The edges in $E(D)\setminus M(D)$ are called *unmarked edges*, and the vertices that are not marked vertices are called *unmarked vertices*. If (X, Y) is a split in G, then we construct a marked graph D that consists of the vertex set $V(G) \cup \{x', y'\}$ for two distinct new vertices $x', y' \notin V(G)$ and the edge set $E(G[X]) \cup E(G[Y]) \cup \{x'y'\} \cup E'$ where we define $x'y'$ as marked and

$$\begin{aligned} E' :=& \{x'x \mid x \in X \text{ and there exists } y \in Y \text{ such that } xy \in E(G)\} \\ & \cup \{y'y \mid y \in Y \text{ and there exists } x \in X \text{ such that } xy \in E(G)\}. \end{aligned}$$

The marked graph D is called a *simple decomposition of* G.

A *split decomposition* of a connected graph G is a marked graph D defined inductively to be either G or a marked graph defined from a split decomposition D' of G by replacing a component H of $(V(D'), E(D')\setminus M(D'))$ with a simple decomposition of H. For a marked edge xy in a split decomposition D, the *recomposition of* D *along* xy is the split decomposition $D' := (D \wedge xy)\setminus\{x, y\}$. For a split decomposition D, let $\mathcal{G}[D]$ denote the graph obtained from D by recomposing all marked edges. By definition, if D is a split decomposition of G, then $\mathcal{G}[D] = G$. Since each marked edge of a split decomposition D is a cut-edge and all marked edges form a matching, if we contract all unmarked edges in D, then we obtain a tree. We call it the *decomposition tree of* G *associated with* D and denote it by T_D. To distinguish the vertices of T_D from the vertices of G or D, the vertices of T_D will be called *nodes*. Obviously, the nodes of T_D are in bijection with the bags of D. Two bags of D are called *neighbor bags* if their corresponding nodes in T_D are adjacent. A subgraph of a split decomposition is called a *sub-decomposition*.

A split decomposition D of G is called a *canonical split decomposition* (or *canonical decomposition* for short) if each bag of D is either a prime graph, a star, or a complete graph, and D is not the refinement of a decomposition with the same property. The following is due to Cunningham and Edmonds [8], and Dahlhaus [9].

Theorem 2.7 (Cunningham and Edmonds *[8];* Dahlhaus *[9]*). *Every connected graph G has a unique canonical decomposition, up to isomorphism, and it can be computed in time $\mathcal{O}(|V(G)| + |E(G)|)$.*

From Theorem 2.7, we can talk about only one canonical decomposition of a connected graph G because all canonical decompositions of G are isomorphic.

Let D be a split decomposition of a connected graph G with bags that are either prime graphs, complete graphs or stars (it is not necessarily a canonical decomposition). The *type of a bag* of D is either P, K, or S depending on whether it is a prime graph, a complete graph, or a star. The *type of a marked edge* uv is AB where A and B are the types of the bags containing u and v respectively. If $A = S$ or $B = S$, then we can replace S by S_p or S_c depending on whether the end of the marked edge is a leaf or the center of the star.

Theorem 2.8 (Bouchet *[5]*). *Let D be a split decomposition of a connected graph with bags that are either complete graphs or stars. Then D is a canonical decomposition if and only if it has no marked edge of type KK or S_pS_c.*

We will use the following characterization of distance-hereditary graphs.

Theorem 2.9 (Bouchet *[5]*). *A connected graph is distance-hereditary if and only if each bag of its canonical decomposition is of type K or S.*

We now relate the split decompositions of a graph and the ones of its locally equivalent graphs. Let D be a split decomposition of a connected graph. A vertex v of D *represents* an unmarked vertex x (or is a *representative* of x) if either $v = x$ or there is a path of even length from v to x in D starting with a marked edge such that marked edges and unmarked edges appear alternately in the path. Two unmarked vertices x and y are *linked* in D if there is a path from x to y in D such that unmarked edges and marked edges appear alternately in the path.

Lemma 2.10 *Let D be a split decomposition of a connected graph. Let v' and w' be two vertices in a same bag of D, and let v and w be two unmarked vertices of D represented by v' and w', respectively. The following are equivalent.*

(1) v and w are linked in D.
(2) $vw \in E(\mathcal{G}[D])$.
(3) $v'w' \in E(D)$.

Proof It is not hard to show that v' and w' are adjacent in D if and only if there is an alternating path from v to w in D from the definition of representativity. Note that recomposing a marked edge in a split decomposition does not change the property that two unmarked vertices are linked, and the adjacency of two vertices in $\mathcal{G}[D]$. It implies that v and w are linked in D if and only if $vw \in E(\mathcal{G}[D])$. □

A *local complementation* at an unmarked vertex x in a split decomposition D, denoted by $D * x$, is the operation to replace each bag B containing a representative w of x with $B * w$. Observe that $D * x$ is a split decomposition of $\mathcal{G}[D] * x$, and $M(D) = M(D * x)$. Two split decompositions D and D' are *locally equivalent* if D can be obtained from D' by applying a sequence of local complementations at unmarked vertices.

Lemma 2.11 (Bouchet [5]). *Let D be the canonical decomposition of a connected graph. If x is an unmarked vertex of D, then $D * x$ is the canonical decomposition of $\mathcal{G}[D] * x$.*

Remark If D is a canonical decomposition and $D' = D * x$ for some unmarked vertex v of D, then $T_{D'}$ and T_D are isomorphic because $M(D) = M(D')$. Thus, for every node v of T_D associated with a bag B of D, its corresponding node v' in $T_{D'}$ is associated in D' with either

(1) B if x has no representative in B, or
(2) $B * w$ if B has a representative w of v.

For easier arguments in several places, if T_D is given for D, then we assume that $T_{D'} = T_D$ for every split decomposition D' locally equivalent to D. For a canonical decomposition D and a node v of its decomposition tree, we write $\mathsf{b}_D(v)$ to denote the bag of D with which it is in correspondence.

Let x and y be linked unmarked vertices in a split decomposition D, and let P be the alternating path in D linking x and y. Observe that each bag contains at most one unmarked edge in P. Notice also that if B is a bag of type S containing an unmarked edge of P, then the center of B is a representative of either x or y. The *pivoting on xy*

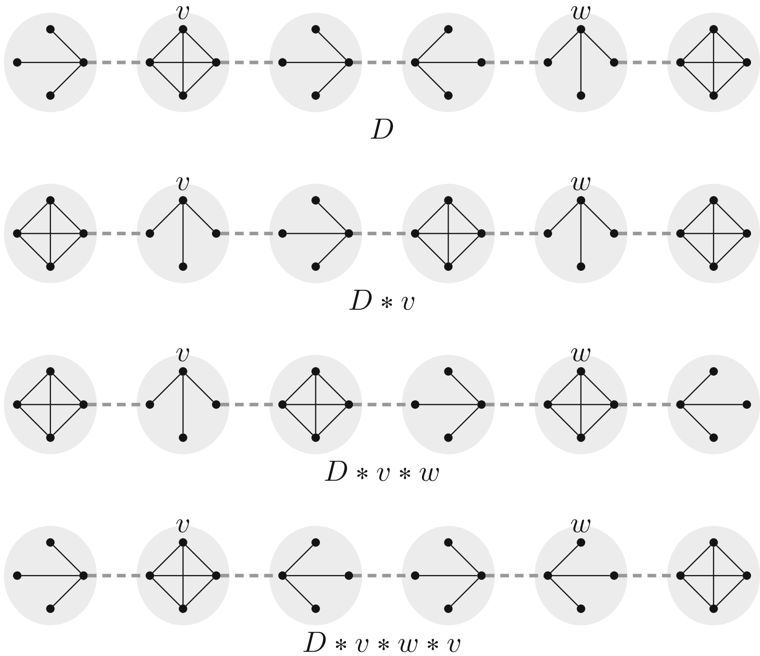


Fig. 2 The split decomposition $D * v * w * v$, which is the same as $D \wedge vw$

of D, denoted by $D \wedge xy$, is the split decomposition obtained as follows: for each bag B containing an unmarked edge of P, if $v, w \in V(B)$ represent respectively x and y in D, then we replace B with $B \wedge vw$. (It is worth noticing that by Lemma 2.10, we have $vw \in E(B)$, hence $B \wedge vw$ is well-defined.)

Lemma 2.12 *Let D be a split decomposition of a connected graph. If $xy \in E(\mathcal{G}[D])$, then $D \wedge xy = D * x * y * x$.*

Proof Since $xy \in E(\mathcal{G}[D])$, by Lemma 2.10, x and y are linked in D. It is easy to see that by the operation $D * x * y * x$, only the bags in the path from x to y are modified, and they are modified according to the definition of $D \wedge xy$. See Fig. 2 for an example of this procedure. □

As a corollary of Lemmas 2.11 and 2.12, we get the following.

Corollary 2.13 *Let D be the canonical decomposition of a connected graph. If $xy \in E(\mathcal{G}[D])$, then $D \wedge xy$ is the canonical decomposition of $\mathcal{G}[D] \wedge xy$.*

The following are split decomposition versions of Lemmas 2.4, 2.5, 2.6, and they can be easily verified in a same way.

Lemma 2.14 *Let D be the canonical decomposition of a connected graph. The following are satisfied.*

(1) If x, y are unmarked vertices of D that are not linked, then $D * x * y = D * y * x$.

(2) If x, y, z are unmarked vertices of D such that x is linked to neither y nor z, and y and z are linked, then $D * x \wedge yz = D \wedge yz * x$.
(3) If x, y, z are unmarked vertices of D such that y is linked to both x and z, then $D \wedge xy \wedge xz = D \wedge yz$.

For a bag B of D and a component T of $D \backslash V(B)$, let us denote by $\zeta_b(D, B, T)$ and $\zeta_c(D, B, T)$ the adjacent marked vertices of D that are in B and in T respectively. The subscripts b and c stand for 'bag' and 'component', respectively. Observe that $\zeta_c(D, B, T)$ is not incident with any marked edge in T. So, when we take a sub-decomposition T from D, we regard $\zeta_c(D, B, T)$ as an unmarked vertex of T.

3 Limbs in Canonical Decompositions

We define the notion of *limb* that is the key ingredient in our characterization. The linear-time algorithm for computing the path-width of trees (and hence their linear rank-width by [18]) is based on the following characterization.

Proposition 3.1 (Ellis, Sudborough, and Turner *[21]*). *A tree T has path-width at most k if and only if for every vertex v of T at most two components of $T \backslash v$ have path-width at most k, and all the other components have path-width at most $k - 1$.*

We want to have a similar characterization for distance-hereditary graphs using their canonical split decompositions, and the notion of limbs is intended to satisfy the following property.

> A distance-hereditary graph has linear rank-width at most k if and only if for every bag B of its canonical decomposition, among the limbs obtained by removing B, there are at most two limbs whose corresponding graphs have linear rank-width at most k, and for other limbs, the corresponding graphs have linear rank-width at most $k - 1$.

Limbs of a canonical decomposition are roughly some of its proper vertex-minors. Before defining it, let us first explain why taking sub-decompositions is not sufficient.

Let B be a bag of a canonical decomposition D. When defining sub-decompositions of components of $D \backslash V(B)$ as limbs we have to deal with the marked vertices having a neighbor in B. If limbs are decompositions obtained by removing these vertices from the components of $D \backslash V(B)$, then we may lose the adjacency information between sub-decompositions, and we cannot get such a characterization indeed; See Fig. 3 for an example. On the other hand, if we regard these marked vertices as new vertices in the sub-decompositions, then we still cannot obtain such a characterization. We give an example in Fig. 4 where three sub-decompositions induce graphs of linear rank-width 2, but the original graph also has linear rank-width 2.

It turns out that by applying local complementations on the marked vertices having a neighbor in B, in the right way depending on the type of B, we can avoid the difficulties showed in Figs. 3 and 4, and indeed obtain the wanted characterization. We now define the notion of limb and prove some technical lemmas that will be used in the subsequent sections. In this section let us fix D the canonical decomposition of

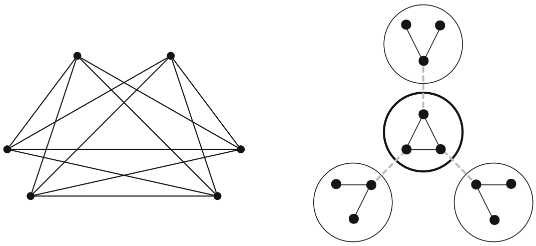
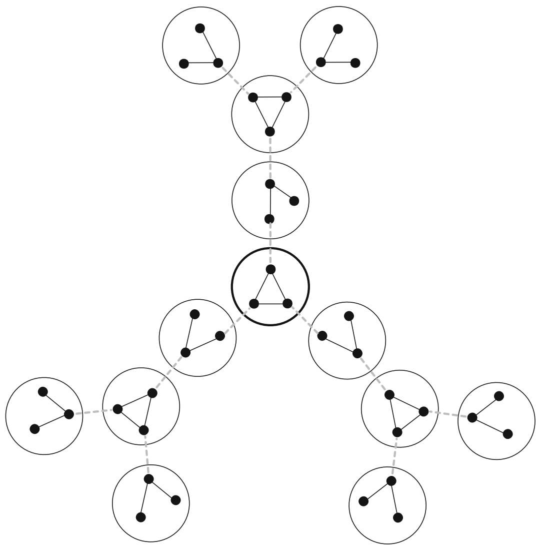


Fig. 3 A graph of linear rank-width 2 and its canonical decomposition. If we regard the marked vertices incident with vertices in the middle bag B as vertices not contained in sub-decompositions after removing B, then each sub-decomposition corresponds to a graph without edges, which has linear rank-width 0

Fig. 4 The canonical decomposition of a graph of linear rank-width 2. If we regard the marked vertices incident with vertices in the middle bag B as vertices of sub-decompositions after removing B, then each sub-decomposition corresponds to a graph of linear rank-width 2



a connected distance-hereditary graph G. We recall from Theorems 2.8 and 2.9 that each bag of D is of type K or S, and marked edges of types KK or $S_p S_c$ do not occur.

For an unmarked vertex y in D and a bag B of D containing a marked vertex that represents y, let T be the component of $D \backslash V(B)$ containing y, and let $v := \zeta_c(D, B, T)$ and $w := \zeta_b(D, B, T)$ be adjacent marked vertices of D. (Recall that $v \in V(T)$ and $w \in V(B)$.) We define the *limb* $\mathcal{L} := \mathcal{L}_D[B, y]$ with respect to B and y as follows:

(1) if B is of type K, then $\mathcal{L} := T * v \backslash v$,
(2) if B is of type S and w is a leaf, then $\mathcal{L} := T \backslash v$,
(3) if B is of type S and w is the center, then $\mathcal{L} := T \wedge vy \backslash v$.

Since v becomes an unmarked vertex in T, the limb is well-defined and it is a split decomposition. While T is a canonical decomposition, $\mathcal{L}$ may not be a canonical decomposition at all, because deleting v may create a bag of size 2. We analyze the cases when such a bag appears, and describe how to transform it into a canonical decomposition.

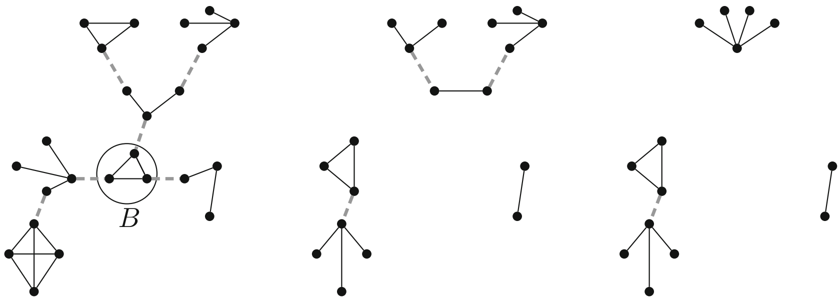


Fig. 5 In **a** we have a canonical decomposition D of a distance-hereditary graph with a bag B. The *dashed edges* are marked edges of D. In **b** we have limbs associated with the components of $D\backslash V(B)$. The canonical limbs associated with limbs are shown in (**c**)

Suppose that a bag B' of size 2 appears in $\mathcal{L}$ by deleting v. If B' has no adjacent bags in $\mathcal{L}$, then B' itself is a canonical decomposition. Otherwise we have two cases.

(1) (B' has one neighbor bag B_1.)
If $v_1 \in V(B_1)$ is the marked vertex adjacent to a vertex of B' and r is the unmarked vertex of B' in $\mathcal{L}$, then we can transform the limb into a canonical decomposition by removing the bag B' and replacing v_1 with r. In other words, we recompose along the marked edge connecting B' and B_1.

(2) (B' has two neighbor bags B_1 and B_2.)
If $v_1 \in V(B_1)$ and $v_2 \in V(B_2)$ are the two marked vertices that are adjacent to the two marked vertices of B', then we can first transform the limb into another decomposition by removing B' and adding a marked edge v_1v_2. If the new marked edge v_1v_2 is of type KK or S_pS_c, then by recomposing along v_1v_2, we finally transform the limb into a canonical decomposition.

Let $\mathcal{LC}_D[B, y]$ be the canonical decomposition obtained from $\mathcal{L}_D[B, y]$, and we call it the *canonical limb*. Let $\mathcal{LG}_D[B, y]$ be the graph obtained from $\mathcal{L}_D[B, y]$ by recomposing all marked edges. See Fig. 5 for an example of canonical limbs.

Lemma 3.2 *Let B be a bag of D. If an unmarked vertex y of D is represented by a marked vertex of B, then $\mathcal{L}_D[B, y]$ is connected.*

Proof Let T be the component of $D\backslash V(B)$ containing y, and $v := \zeta_c(D, B, T)$, and B' be the bag of D containing v. Since local complementations maintain connectedness, it suffices to verify that $V(B')\backslash\{v\}$ induces a connected subgraph in $\mathcal{L}_D[B, y]$. This is not hard to see for each of the three cases. □

Lemma 3.3 *Let B be a bag of D. If two unmarked vertices x and y are represented by a marked vertex w in B, then $\mathcal{L}_D[B, x]$ is locally equivalent to $\mathcal{L}_D[B, y]$.*

Proof Since x and y are represented by the same vertex w of B, they are contained in the same component of $D\backslash V(B)$, say T. Let $v := \zeta_c(D, B, T)$.

If B is a complete bag or a star bag having w as a leaf, then by the definition of limbs, $\mathcal{L}_D[B,x] = \mathcal{L}_D[B,y]$. So, we may assume that B is a star bag and w is its center. Since v is linked to both x and y in T, by Lemma 2.14, $T \wedge vx \wedge xy = T \wedge vy$. So, we obtain that $(T \wedge vx \backslash v) \wedge xy = T \wedge vx \wedge xy \backslash v = T \wedge vy \backslash v$. Therefore $\mathcal{L}_D[B,x]$ is locally equivalent to $\mathcal{L}_D[B,y]$. □

For a bag B of D and a component T of $D\backslash V(B)$, we define $f_D(B,T)$ as the linear rank-width of $\mathcal{LG}_D[B,y]$ for some unmarked vertex $y \in V(T)$. By Lemma 3.3, $f_D(B,T)$ does not depend on the choice of y. Furthermore, by the following proposition, it does not change when we replace D with some decomposition locally equivalent to D.

Proposition 3.4 *Let B be a bag of D and let y be an unmarked vertex of D represented by a vertex w in B. Let $x \in V(\mathcal{G}[D])$. If an unmarked vertex y' is represented by w in $D * x$, then $\mathcal{LG}_D[B,y]$ is locally equivalent to $\mathcal{LG}_{D*x}[(D*x)[V(B)],y']$. Therefore, $f_D(B,T) = f_{D*x}((D*x)[V(B)],T_x)$ where T and T_x are the components of $D\backslash V(B)$ and $(D*x)\backslash V(B)$ containing y, respectively. Moreover, $\mathcal{LC}_D[B,y]$ and $\mathcal{LC}_{D*x}[(D*x)[V(B)],y']$ are locally equivalent as canonical decompositions.*

Before proving it, let us recall the following by Geelen and Oum.

Lemma 3.5 (Geelen and Oum [17, Lemma3.1]). *Let G be a graph and x, y be two distinct vertices in G. Let $xw \in E(G*y)$ and $xz \in E(G)$.*

(1) If $xy \notin E(G)$, then $(G*y)\backslash x$, $(G*y*x)\backslash x$, and $(G*y) \wedge xw\backslash x$ are locally equivalent to $G\backslash x$, $G*x\backslash x$, and $G \wedge xz\backslash x$, respectively.
(2) If $xy \in E(G)$, then $(G*y)\backslash x$, $(G*y*x)\backslash x$, and $(G*y) \wedge xw\backslash x$ are locally equivalent to $G\backslash x$, $G \wedge xz\backslash x$, and $(G*x)\backslash x$, respectively.

Proof of Proposition 3.4 *Let $v := \zeta_c(D,B,T)$ and $B' := (D*x)[V(B)]$. Let T and T_x be the components of $D\backslash V(B)$ and $(D*x)\backslash V(B')$ containing y, respectively. Note that $V(T) = V(T_x)$.*

We claim that $\mathcal{LG}_D[B,y]$ is locally equivalent to $\mathcal{LG}_{D*x}[B',y']$ for some unmarked vertex y' represented by w in $D*x$. We divide into cases depending on the type of the bag B and whether $x \in V(T)$.

Case 1. *$x \in V(T)$ and x is not linked to v in T.*

Since x is not linked to v in T, applying a local complementation at x does not change the bag B. Thus, $B' = B$ and $vx \notin E(\mathcal{G}[T])$. In this case, let $y' := y$.

(1) (B is of type S and w is a leaf of B.) $\mathcal{L}_D[B,y] = T\backslash v$ and $\mathcal{L}_{D*x}[B',y'] = T*x\backslash v$. Since $(T\backslash v)*x = T*x\backslash v$, $\mathcal{L}_D[B,y]$ and $\mathcal{L}_{D*x}[B',y']$ are locally equivalent, and thus $\mathcal{LG}_D[B,y]$ and $\mathcal{LG}_{D*x}[B',y']$ are locally equivalent.
(2) (B is of type S and w is the center of B.) $\mathcal{L}_D[B,y] = T \wedge vy\backslash v$ and $\mathcal{L}_{D*x}[B',y'] = (T*x) \wedge vy\backslash v$, and we have

$$\mathcal{LG}_D[B,y] = \mathcal{G}[T \wedge vy\backslash v] = \mathcal{G}[T] \wedge vy\backslash v.$$

Since $vx \notin E(\mathcal{G}[T])$, by Lemma 3.5, $\mathcal{LG}_D[B,y]$ is locally equivalent to

$$\mathcal{LG}_{D*x}[B',y'] = \mathcal{G}[(T*x) \wedge vy\backslash v] = \mathcal{G}[T]*x \wedge vy\backslash v.$$

(3) (B is of type K.) $\mathcal{L}_D[B, y] = T * v \backslash v$ *and* $\mathcal{L}_{D*x}[B', y'] = T * x * v \backslash v$*, and we have*

$$\mathcal{LG}_D[B, y] = \mathcal{G}[T * v \backslash v] = \mathcal{G}[T] * v \backslash v.$$

Since $vx \notin E(\mathcal{G}[T])$*, by Lemma 3.5,* $\mathcal{LG}_D[B, y]$ *is locally equivalent to*

$$\mathcal{LG}_{D*x}[B', y'] = \mathcal{G}[T * x * v \backslash v] = \mathcal{G}[T] * x * v \backslash v.$$

Case 2. $x \in V(T)$ and x is linked to v in T.

Since x *is linked to* v *in* T*,* $vx \in E(\mathcal{G}[T])$*. Let* $y' := x$ *for this case.*

(1) (B is of type S and w is a leaf of B.) Applying a local complementation at x does not change the type of the bag B. So, $\mathcal{L}_D[B, y] = T \backslash v$ *and* $\mathcal{L}_{D*x}[B', y'] = T * x \backslash v$*. Since* $(T \backslash v) * x = T * x \backslash v$*,* $\mathcal{LG}_D[B, y]$ *and* $\mathcal{LG}_{D*x}[B', y']$ *are locally equivalent.*

(2) (B is of type S and w is the center of B.) Applying a local complementation at x changes the bag B into a bag of type K, and the component T into $T * x$*. Thus,* $\mathcal{L}_D[B, y] = T \wedge vy \backslash v$ *and* $\mathcal{L}_{D*x}[B', y'] = (T * x) * v \backslash v$*, and*

$$\mathcal{LG}_D[B, y] = \mathcal{G}[T \wedge vy \backslash v] = \mathcal{G}[T] \wedge vy \backslash v.$$

Since $vx \in E(\mathcal{G}[T])$*, by Lemma 3.5,* $\mathcal{LG}_D[B, y]$ *is locally equivalent to*

$$\mathcal{LG}_{D*x}[B', y'] = \mathcal{G}[(T * x) * v \backslash v] = \mathcal{G}[T] * x * v \backslash v.$$

(3) (B is of type K.) Applying a local complementation at x changes the bag B into a bag of type S whose center is w. $\mathcal{L}_D[B, y] = T * v \backslash v$ *and* $\mathcal{L}_{D*x}[B', y'] = T * x \wedge vx \backslash v$*, and we have*

$$\mathcal{LG}_D[B, y] = \mathcal{G}[T * v \backslash v] = \mathcal{G}[T] * v \backslash v.$$

Since $vx \in E(\mathcal{G}[T])$*, by Lemma 3.5,* $\mathcal{LG}_D[B, y]$ *is locally equivalent to*

$$\mathcal{LG}_{D*x}[B', y'] = \mathcal{G}[T * x \wedge vx \backslash v] = \mathcal{G}[T] * x \wedge vx \backslash v.$$

Case 3. $x \notin V(T)$.

If x has no representative in the bag B, then applying a local complementation at x does not change the bag B and the component T. Therefore, we may assume that x is represented by some vertex in B, that is adjacent to w. In this case, v is still a representative of y in $D * x$*. Let* $y' := y$*.*

(1) (B is of type S and w is a leaf of B.) If the representative of x in B is a leaf of B, then it is not adjacent to w. Thus, the representative of x in B is a center of B, and applying a local complementation at x changes B into a bag of type K, and T into $T * v$*. We have* $\mathcal{L}_{D*x}[B', y'] = (T * v) * v \backslash v = T \backslash v = \mathcal{L}_D[B, y]$*.*

(2) (B is of type S and w is the center of B.) Since w is the center of B, x is represented by a leaf of the bag B. Applying a local complementation at x does not change the bag B, but it changes T into T ∗ v. So we have $\mathcal{L}_D[B, y] = T \wedge vy \backslash v$ *and* $\mathcal{L}_{D*x}[B', y'] = (T * v) \wedge vy \backslash v = T * y * v \backslash v$*, and we have*

$$\mathcal{LG}_D[B, y] = \mathcal{G}[T \wedge vy \backslash v] = \mathcal{G}[T] \wedge vy \backslash v.$$

Since $vy \in E(\mathcal{G}[T])$*, by Lemma 3.5,* $\mathcal{LG}_D[B, y]$ *is locally equivalent to*

$$\mathcal{LG}_{D*x}[B', y'] = \mathcal{G}[T * y * v \backslash v] = \mathcal{G}[T] * y * v \backslash v.$$

(3) (B is of type K.) After applying a local complementation at x in D, B becomes a star with a leaf w, and T becomes T ∗ v. Therefore, we have $\mathcal{L}_{D*x}[B', y'] = T * v \backslash v = \mathcal{L}_D[B, y]$*.*

We conclude that $\mathcal{LG}_D[B, y]$ *and* $\mathcal{LG}_{D*x}[B', y']$ *are locally equivalent, and by Lemma 3.3, we have* $f_D(B, T) = f_{D*x}(B', T_x)$*. Also, by construction* $\mathcal{LC}_D[B, y]$ *and* $\mathcal{LC}_{D*x}[B', y']$ *are canonical decompositions of* $\mathcal{LG}_D[B, y]$ *and* $\mathcal{LG}_{D*x}[B', y']$*, respectively. By Lemma 2.11, we can conclude that* $\mathcal{LC}_D[B, y]$ *and* $\mathcal{LC}_{D*x}[B', y']$ *are locally equivalent as canonical decompositions.* □

The following lemma is useful to reduce cases in several proofs.

Lemma 3.6 *Let* B_1 *and* B_2 *be two distinct bags of D and for each* $i \in \{1, 2\}$*, let* T_i *be the components of* $D \backslash V(B_i)$ *such that* T_1 *contains the bag* B_2 *and* T_2 *contains the bag* B_1*. Then there exists a canonical decomposition* D' *locally equivalent to D such that for each* $i \in \{1, 2\}$*,* $D'[V(B_i)]$ *is a star and* $\zeta_b(D, B_i, T_i)$ *is a leaf of* $D'[V(B_i)]$*.*

Proof Let $v_i := \zeta_b(D, B_i, T_i)$ for $i = 1, 2$. It is easy to make B_1 into a star bag having v_1 as a leaf by applying local complementations. We may assume that v_1 is a leaf of B_1 in D. If v_2 is a leaf of B_2, then we are done. If B_2 is a complete bag, then choose an unmarked vertex w_2 of D that is represented by a vertex of B_2 other than v_2. Then applying a local complementation at w_2 makes B_2 into a star bag having v_2 as a leaf without changing B_1. Therefore, we may assume that v_2 is the center of the star bag B_2. If B_1 and B_2 are neighbor bags in D, then the marked edge connecting B_1 and B_2 is of type $S_p S_c$, contradicting to the assumption that D is a canonical decomposition. Thus, B_1 and B_2 are not neighbor bags in D.

Let $T := D[V(T_1) \cap V(T_2)]$ and $w_2 := \zeta_c(D, B_2, T_2)$. By the definition of a canonical decomposition, w_2 is not a leaf of a star bag in D. Therefore, there exists an unmarked vertex $y \in V(T)$ of D such that y is linked to w_2 in T. Choose an unmarked vertex y' of D represented by w_2 in D. Since y is linked to y' and the alternating path from y to y' in D passes through B_2 but not B_1, pivoting yy' in D makes B_2 into a star bag having v_2 as a leaf without changing B_1. Thus, each v_i is a leaf of $(D \wedge yy')[V(B_i)]$ in $D \wedge yy'$, as required. □

We conclude the section with the following.

Proposition 3.7 *Let B_1 and B_2 be two distinct bags of D, and T_1 be a component of $D\backslash V(B_1)$ not containing B_2, and T_2 be the component of $D\backslash V(B_2)$ containing B_1. If $y_1 \in V(T_1)$ and $y_2 \in V(T_2)$ are two unmarked vertices of D that are represented by some vertices in B_1 and B_2, respectively, then $\mathcal{LG}_D[B_1, y_1]$ is a vertex-minor of $\mathcal{LG}_D[B_2, y_2]$. Therefore $f_D(B_1, T_1) \le f_D(B_2, T_2)$.*

Proof Let $u_2 := \zeta_c(D, B_2, T_2)$ and $v_2 := \zeta_b(D, B_2, T_2)$. By Lemma 3.6, there exists a canonical decomposition D' locally equivalent to D such that B_2 is a star bag in D' with a leaf v_2. For each $i \in \{1, 2\}$, let $T_i' := D'[V(T_i)]$, $B_i' := D'[V(B_i)]$ and let y_i' be an unmarked vertex of D' represented by $\zeta_b(D', B_i', T_i')$.

Since v_2 is a leaf of B_2' in D', we have $\mathcal{L}_{D'}[B_2', y_2'] = T_2'\backslash v_2$. Because T_1' is a subgraph of $T_2'\backslash v_2$, we can easily observe that $\mathcal{LG}_{D'}[B_1', y_1']$ is a vertex-minor of $\mathcal{LG}_{D'}[B_2', y_2']$. Since $\mathcal{L}_D[B_i, y_i]$ is locally equivalent to $\mathcal{L}_{D'}[B_i', y_i']$ for each $i \in \{1, 2\}$, $\mathcal{LG}_D[B_1, y_1]$ is a vertex-minor of $\mathcal{LG}_D[B_2, y_2]$. We conclude that $f_D(B_1, T_1) \le f_D(B_2, T_2)$. □

4 Characterizing the Linear Rank-Width of Distance-Hereditary Graphs

In this section, we prove the main structural result of this paper, which characterizes the linear rank-width of distance-hereditary graphs.

Theorem 4.1 *Let k be a positive integer and let D be the canonical decomposition of a connected distance-hereditary graph G. Then $\mathrm{lrw}(G) \le k$ if and only if for each bag B of D, D has at most two components T of $D\backslash V(B)$ such that $f_D(B, T) = k$, and every other component T' of $D\backslash V(B)$ satisfies that $f_D(B, T') \le k - 1$.*

Let D be the canonical decomposition of a connected distance-hereditary graph G, and we fix a positive integer k. For simpler arguments, we remove D from the notation $f_D(B, T)$ in this section. We first prove the forward direction.

Proof of the forward direction of Theorem 4.1 *Suppose that there exists a bag B of D such that $D\backslash V(B)$ has at least three components T which induce limbs L where $\mathcal{G}[L]$ has linear rank-width k.*

We claim that $\mathrm{lrw}(G) \ge k + 1$. We may assume that $D\backslash V(B)$ has exactly three components T_1, T_2 and T_3, where each component T_i satisfies $f(B, T_i) = k$. Since removing a vertex from a graph does not increase the linear rank-width, we may further assume that $V(B) = \{\zeta_b(D, B, T_i) \mid 1 \le i \le 3\}$. Now, every unmarked vertex of D is contained in one of T_1, T_2, and T_3. For each $1 \le i \le 3$, let $w_i := \zeta_c(D, B, T_i)$, and let N_i be the set of the unmarked vertices of T_i that are linked to w_i in T_i. Choose a vertex u_i in N_i and let $D_i := \mathcal{L}_D[B, u_i]$ and $G_i := \mathcal{G}[D_i]$. We remark that N_i is exactly the set of the vertices in G_i that have a neighbor in $V(G)\backslash V(G_i)$. By Proposition 3.4 and Lemmas 2.3 and 2.11, for any canonical decomposition D' locally equivalent to D, we have $\mathrm{lrw}(\mathcal{G}[D]) = \mathrm{lrw}(\mathcal{G}[D'])$ and $f(B, T_i)$ does not change. So, we may assume that B is a complete bag of D.

We first claim that $D_2 = (D * u_1)[V(T_2)\backslash w_2]$. Since B is a complete bag, by the definition of limbs, $D_2 = T_2 * w_2\backslash w_2$. Since u_1 is linked to w_1 in T_1 and there is an *alternating path from* w_1 *to* w_2 *in* D*, by concatenating alternating paths it is easy to see that* $(D * u_1)[V(T_2)\backslash w_2] = T_2 * w_2\backslash w_2 = D_2$*, as claimed.*

Towards a contradiction, suppose that G *has a linear layout* L *of width* k*. Let* a *and* b *be the first and last vertices of* L*, respectively. Since* B *has no unmarked vertices, without loss of generality, we may assume that* $a, b \in V(G_1) \cup V(G_3)$*. With this assumption, we claim that* G_2 *has linear rank-width at most* $k-1$*.*

Let $v \in V(G_2)$ *and* $S_v := \{x \in V(G) \mid x \leq_L v\}$ *and* $T_v := V(G)\backslash S_v$*. Since* v *is arbitrary, it is sufficient to show that* $\mathrm{cutrk}_{G_2}(S_v \cap V(G_2)) \leq k-1$*.*

We divide into three cases. We first check two cases that are (1) ($N_1 \cap S_v \neq \emptyset$ *and* $N_3 \cap T_v \neq \emptyset$*), and (2) (*$N_1 \cap T_v \neq \emptyset$ *and* $N_3 \cap S_v \neq \emptyset$*). If both of them are not satisfied, then we can easily deduce that* $N_1 \cup N_3 \subseteq S_v$ *or* $N_1 \cup N_3 \subseteq T_v$*.*

Case 1. $N_1 \cap S_v \neq \emptyset$ and $N_3 \cap T_v \neq \emptyset$.

Let $x_1 \in N_1 \cap S_v$ *and* $x_3 \in N_3 \cap T_v$*. We claim that*

$$\mathrm{cutrk}_{G_2}(S_v \cap V(G_2)) = \mathrm{cutrk}_{G[V(G_2)\cup\{x_1,x_3\}]}((S_v \cap V(G_2)) \cup \{x_1\}) - 1.$$

Because $\mathrm{cutrk}_{G[V(G_2)\cup\{x_1,x_3\}]}((S_v \cap V(G_2)) \cup \{x_1\}) \leq \mathrm{cutrk}_G(S_v) \leq k$*, the claim implies that* $\mathrm{cutrk}_{G_2}(S_v \cap V(G_2)) \leq k-1$*.*

Note that x_1 *and* x_3 *have the same neighbors in* $G[V(G_2) \cup \{x_1, x_3\}]$ *because* B *is a complete bag. Since* x_1 *is adjacent to* x_3 *in* $G[V(G_2) \cup \{x_1, x_3\}]$*,* x_3 *becomes a leaf in* $G[V(G_2) \cup \{x_1, x_3\}] * x_1$ *whose neighbor is* x_1*. Since* $(D * x_1)[V(T_2)\backslash w_2] = D_2$*, we have*

$$G[V(G_2) \cup \{x_1, x_3\}] * x_1\backslash x_1\backslash x_3 = (G * x_1)[V(G_2)] = G_2.$$

Therefore,

$$\begin{aligned}
&\mathrm{cutrk}_{G[V(G_2)\cup\{x_1,x_3\}]}((S_v \cap V(G_2)) \cup \{x_1\}) \\
&\quad = \mathrm{cutrk}_{G[V(G_2)\cup\{x_1,x_3\}]*x_1}((S_v \cap V(G_2)) \cup \{x_1\}) \\
&\quad = \mathrm{rank}\left(\begin{array}{c|cc} & x_3 & T_v \cap V(G_2) \\ \hline x_1 & 1 & * \\ S_v \cap V(G_2) & 0 & * \end{array} \right) \\
&\quad = \mathrm{rank}\left(\begin{array}{c|cc} & x_3 & T_v \cap V(G_2) \\ \hline x_1 & 1 & 0 \\ S_v \cap V(G_2) & 0 & * \end{array} \right) \\
&\quad = \mathrm{cutrk}_{G[V(G_2)\cup\{x_1,x_3\}]*x_1\backslash x_1\backslash x_3}(S_v \cap V(G_2)) + 1 \\
&\quad = \mathrm{cutrk}_{(G_2)}(S_v \cap V(G_2)) + 1,
\end{aligned}$$

as claimed.

Case 2. $N_1 \cap T_v \neq \emptyset$ and $N_3 \cap S_v \neq \emptyset$.

In the same way as ***Case 1****, we can prove* $\mathrm{cutrk}_{G_2}(S_v \cap V(G_2)) \leq k-1$*.*

Case 3. $N_1 \cup N_3 \subseteq S_v$ or $N_1 \cup N_3 \subseteq T_v$.

We can assume without loss of generality that $N_1 \cup N_3 \subseteq S_v$ because the case when $N_1 \cup N_3 \subseteq T_v$ is similar. Since $a, b \in V(G_1) \cup V(G_3)$ and the graph $G[V(G_1) \cup V(G_3)]$ is connected, there exist vertices $s \in S_v \cap (V(G_1) \cup V(G_3))$ and $t \in T_v \cap (V(G_1) \cup V(G_3))$ such that

(1) $st \in E(G)$,
(2) t has no neighbors in N_2.

We have

$$\begin{aligned}\text{cutrk}_G(S_v) &\geq \text{rank}\left(\begin{array}{cc} & \begin{array}{cc} t & T_v \cap V(G_2)\end{array} \\ \begin{array}{c} s \\ S_v \cap V(G_2)\end{array} & \left(\begin{array}{c|c} 1 & * \\ \hline 0 & * \end{array}\right)\end{array}\right) \\ &= \text{rank}\left(\begin{array}{cc} & \begin{array}{cc} t & T_v \cap V(G_2)\end{array} \\ \begin{array}{c} s \\ S_v \cap V(G_2)\end{array} & \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & * \end{array}\right)\end{array}\right) \\ &= \text{cutrk}_{G_2}(S_v \cap V(G_2)) + 1.\end{aligned}$$

Therefore, we conclude $\text{cutrk}_{G_2}(S_v \cap V(G_2)) \leq k - 1$.

Thus, G_2 has linear rank-width at most $k - 1$, which yields a contradiction. □

The proof of the converse direction can be summarized as follows.

(1) There is a path P in T_D such that for each node v in P and each component T of $D \backslash V(\mathsf{b}_D(v))$ not containing a bag $\mathsf{b}_D(w)$ with $w \in V(P)$, $f(B, T) \leq k - 1$ (Lemmas 4.4 and 4.5).
(2) We then follow the linear order induced by the path P to construct a linear layout of width k by concatenating the linear layouts of the graphs induced by the limbs associated with the nodes of P (Lemmas 4.2 and 4.3).

For two linear layouts $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_m)$, we define

$$(x_1, \ldots, x_n) \oplus (y_1, \ldots, y_m) := (x_1, \ldots, x_n, y_1, \ldots, y_m).$$

Lemma 4.2 *Let B be a bag of D of type S with two unmarked vertices x and y such that x is the center and y is a leaf of B. If for every component T of $D \backslash V(B)$, $f(B, T) \leq k - 1$, then the graph $\mathcal{G}[D]$ has a linear layout of width at most k whose first and last vertices are x and y, respectively.*

Proof Let $T_1, T_2, \ldots, T_\ell$ be the components of $D \backslash V(B)$ and for each $1 \leq i \leq \ell$, let $w_i := \zeta_c(D, B, T_i)$ and let y_i be a vertex in T_i represented by a vertex of B. Since each w_i is adjacent to a leaf of B, $T_i \backslash w_i$ is the limb of D with respect to B and y_i. Let $A := V(B) \backslash \left(\bigcup_{1 \leq j \leq \ell} \{\zeta_b(D, B, T_i)\}\right) \backslash \{x, y\}$, and let L_A be a sequence of A.

Suppose that for every component T of $D \backslash V(B)$, $f(B, T) \leq k - 1$. For each $1 \leq i \leq \ell$, let L_i be a linear layout of $\mathcal{G}[T_i \backslash w_i]$ of width at most $k - 1$. We claim that

$$L := (x) \oplus L_1 \oplus L_2 \oplus \cdots \oplus L_\ell \oplus L_A \oplus (y)$$

is a linear layout of $\mathcal{G}[D]$ of width at most k. It is sufficient to prove that for every $w \in V(\mathcal{G}[D])\setminus\{x, y\}$, $\text{cutrk}_{\mathcal{G}[D]}(\{v \mid v \le_L w\}) \le k$.

Let $w \in V(\mathcal{G}[D])\setminus(A \cup \{x, y\})$, and let $S_w := \{v : v \le_L w\}$ and $T_w := V(\mathcal{G}[D])\setminus S_w$. Let j be the integer such that L_j contains w. Then

$$\begin{aligned}
&\text{cutrk}_{\mathcal{G}[D]}(S_w)\\
&= \text{rank}\left(\begin{array}{c|c|c|c|c}
 & y & A & T_w \cap V(\mathcal{G}[T_j]) & T_w\setminus\{y\}\setminus A\setminus V(\mathcal{G}[T_j]) \\
x & 1 & 1 & * & * \\ \hline
S_w \cap V(\mathcal{G}[T_j]) & 0 & 0 & * & 0 \\ \hline
S_w\setminus\{x\}\setminus V(\mathcal{G}[T_j]) & 0 & 0 & 0 & 0
\end{array}\right)\\
&= \text{rank}\left(\begin{array}{c|c|c|c|c}
 & y & A & T_w \cap V(\mathcal{G}[T_j]) & T_w\setminus\{y\}\setminus A\setminus V(\mathcal{G}[T_j]) \\
x & 1 & 0 & 0 & 0 \\ \hline
S_w \cap V(\mathcal{G}[T_j]) & 0 & 0 & * & 0 \\ \hline
S_w\setminus\{x\}\setminus V(\mathcal{G}[T_j]) & 0 & 0 & 0 & 0
\end{array}\right)\\
&= \text{cutrk}_{\mathcal{G}[T_j\setminus w_j]}(S_w \cap V(\mathcal{G}[T_j])) + 1 \le (k-1) + 1 = k.
\end{aligned}$$

If $w \in A$, then it is easy to show that $\text{cutrk}_{\mathcal{G}[D]}(\{v \mid v \le_L w\}) \le 1$. Therefore, L is a linear layout of $\mathcal{G}[D]$ of width k whose first and last vertices are x and y, respectively. □

We can remove the assumption on the shape of B in Lemma 4.2.

Lemma 4.3 *Let B be a bag of D with two unmarked vertices x and y. If for every component T of $D\setminus V(B)$, $f(B, T) \le k - 1$, then the graph $\mathcal{G}[D]$ has a linear layout of width at most k whose first and last vertices are x and y, respectively.*

Proof Suppose that $f(B, T) \le k - 1$ for every component T of $D\setminus V(B)$. We obtain a decomposition D' from D as follows:

(1) If B is a complete graph, then let $D' := D * x$.
(2) If B is a star whose center is x, then let $D' := D$.
(3) Otherwise let $D' := D \wedge xz$ where z is an unmarked vertex represented by the center of B.

It is clear that $D'[V(B)]$ is a star whose center is x. By Proposition 3.4, for each component T of $D\setminus V(B)$, $f(B, T) = f_{D'}(D'[V(B)], D'[V(T)])$. Thus, by Lemma 4.2, $\mathcal{G}[D']$ has a linear layout of width at most k whose first and last vertices are x and y, respectively. Since $\mathcal{G}[D']$ is locally equivalent to $\mathcal{G}[D]$, we conclude that $\mathcal{G}[D]$ also has such a linear layout. □

Lemma 4.4 *If*

(1) for each bag B of D, there are at most two components T of $D\setminus V(B)$ satisfying $f(B, T) = k$, and
(2) for every other component T' of $D\setminus V(B)$, $f(B, T') \le k - 1$, and

(3) P is the set of nodes v in T_D such that exactly two components T of $D \setminus V(\mathsf{b}_D(v))$ satisfy $f(\mathsf{b}_D(v), T) = k$,

then either $P = \emptyset$ or $T_D[P]$ is a path.

Proof Suppose that $P \neq \emptyset$. If P has two distinct nodes v_1 and v_2, then there exists a component T_1 of $D \setminus V(\mathsf{b}_D(v_1))$ not containing $V(\mathsf{b}_D(v_2))$ such that $f(\mathsf{b}_D(v_1), T_1) = k$, and there exists a component T_2 of $D \setminus V(\mathsf{b}_D(v_2))$ not containing $V(\mathsf{b}_D(v_1))$ such that $f(\mathsf{b}_D(v_2), T_2) = k$. By Proposition 3.7, for every node v on the path from v_1 to v_2 in T_D, v must be contained in P. So P induces a tree in T_D.

Suppose now that P contains a node v having three neighbor bags v_1, v_2, and v_3 in P. Then, again by Proposition 3.7, D must have three components T of $D \setminus V(\mathsf{b}_D(v))$ such that $f(\mathsf{b}_D(v), T) = k$, which contradicts the assumption. Therefore, P induces a path in T_D. □

Lemma 4.5 *If*

(1) for each bag B of D, there are at most two components T of $D \setminus V(B)$ satisfying $f(B, T) = k$, and
(2) $f(B, T') \leq k - 1$ for all the other components T' of $D \setminus V(B)$,

then T_D has a path P such that for each node v in P and each component T of $D \setminus V(\mathsf{b}_D(v))$ not containing a bag $\mathsf{b}_D(w)$ with $w \in V(P)$, $f(\mathsf{b}_D(v), T) \leq k - 1$.

Proof Let P' be the set of nodes v in T_D such that exactly two components T of $D \setminus V(\mathsf{b}_D(v))$ satisfy $f(\mathsf{b}_D(v), T) = k$. By Lemma 4.4, either $P' = \emptyset$ or $T_D[P']$ is a path.

We first assume that $P' \neq \emptyset$. Let $T_D[P'] = v_1 v_2 \cdots v_n$, and for each $1 \leq i \leq n$, let $B_i := \mathsf{b}_D(v_i)$. By the definition, there exists a component T_1 of $D \setminus V(B_1)$ such that T_1 does not contain a bag $\mathsf{b}_D(w)$ with $w \in V(P')$ and $f(B_1, T_1) = k$. Let v_0 be the node of T_D such that $\mathsf{b}_D(v_0)$ is the bag of T_1 that is the neighbor bag of B_1 in D. Similarly, there exists a component T_n of $D \setminus V(B_n)$ such that T_n does not contain a bag $\mathsf{b}_D(w)$ with $w \in V(P')$ and $f(B_n, T_n) = k$. Let v_{n+1} be the node of T_D such that $\mathsf{b}_D(v_{n+1})$ is the bag of T_n that is the neighbor bag of B_n in D. Then $P := v_0 v_1 v_2 \cdots v_n v_{n+1}$ is the required path.

Now we assume that $P' = \emptyset$. We choose a node v_0 in T_D and let $B_0 := \mathsf{b}_D(v_0)$. If D has no component T of $D \setminus V(B_0)$ such that $f(B_0, T) = k$, then $P := v_0$ satisfies the condition. If not, we take a maximal path $P := v_0 v_1 \cdots v_{n+1}$ in T_D such that (with $B_i := \mathsf{b}_D(v_i)$)

– for each $0 \leq i \leq n$, $D \setminus V(B_i)$ has one component T_i such that $f(B_i, T_i) = k$, and B_{i+1} is the bag of T_i that is the neighbor bag of B_i in D.

By the maximality of P, P is a path in T_D such that for each node v of P and a component T of $D \setminus V(\mathsf{b}_D(v))$ not containing a bag $\mathsf{b}_D(w)$ with $w \in V(P)$, $f(\mathsf{b}_D(v), T) \leq k - 1$. □

We are now ready to prove the converse direction of the proof of Theorem 4.1.

Proof of the Backward Direction of Theorem 4.1 *Suppose that for each bag B of D, at most two components T of $D \setminus V(B)$ induce limbs L where $\mathcal{G}[L]$ has linear* *rank-width exactly k, and all other component T' of $D\backslash V(B)$ induce limbs L' where $\mathcal{G}[L']$ has linear rank-width at most $k-1$. We claim that* $\mathrm{lrw}(G) \le k$.

Let $P := v_0v_1 \cdots v_nv_{n+1}$ be the path in T_D such that

- *for each node v in P and a component T of $D\backslash V(\mathsf{b}_D(v))$ not containing a bag $\mathsf{b}_D(w)$ with $w \in V(P)$, $f(\mathsf{b}_D(v), T) \le k-1$ (such a path exists by Lemma 4.5).*

For each $0 \le i \le n+1$, let $B_i := \mathsf{b}_D(v_i)$. If P consists of one node, then by Lemma 4.3, $\mathrm{lrw}(G) = \mathrm{lrw}(\mathcal{G}[D]) \le k$. *Thus, we may assume that $n \ge 0$.*

By adding unmarked vertices in B_0 and B_{n+1} if necessary, we assume that B_0 and B_{n+1} have unmarked vertices a_0 and b_{n+1} in D, respectively.

For each $0 \le i \le n$, let b_i be a marked vertex of B_i and let a_{i+1} be a marked vertex B_{i+1} such that b_ia_{i+1} is the marked edge connecting B_i and B_{i+1}. Let D_0 be the component of $D\backslash V(B_1)$ containing the bag B_0. Let D_{n+1} be the component of $D\backslash V(B_n)$ containing the bag B_{n+1}. For each $1 \le i \le n$, let D_i be the component of $D\backslash(V(B_{i-1}) \cup V(B_{i+1}))$ containing the bag B_i. Notice that the vertices a_i and b_i are unmarked vertices in D_i.

Since every component T of $D_i\backslash V(B_i)$ satisfies that $f_{D_i}(B_i, T) \le k-1$, by Lemma 4.3, G_i has a linear layout L'_i of width k whose first and last vertices are a_i and b_i, respectively. For each $1 \le i \le n$, let L_i be the linear layout obtained from L'_i by removing a_i and b_i. Let L_0 and L_{n+1} be obtained from L'_0 and L'_{n+1} by removing b_0 and a_{n+1}, respectively. Then we can easily check that $L := L_0 \oplus L_1 \oplus \cdots \oplus L_{n+1}$ is a linear layout of $\mathcal{G}[D]$ having width at most k. Therefore $\mathrm{lrw}(\mathcal{G}[D]) \le k$. □

5 Canonical Limbs

The objective now is to design an algorithm to compute the linear rank-width of distance-hereditary graphs based on our characterization in Theorem 4.1. The scheme of this algorithm is actually the same as the algorithm for computing the linear rank-width (or path-width) of trees. Since our algorithm for distance-hereditary graphs needs more notations, before describing it, we briefly describe, for easier understanding, the algorithm for trees [21].

Let F be a rooted tree. The algorithm from [21] computes the linear rank-width of F bottom-up, *i.e.*, it computes for each internal node u the linear rank-width of the subtree $F(u)$ rooted at u. Let $k := \max\{\mathrm{lrw}(F(v)) \mid v \text{ is a child of } u\}$. If there is a descendant v of u, called a *k-critical node*, that has two children v_1 and v_2 such that $\mathrm{lrw}(F(v)) = \mathrm{lrw}(F(v_1)) = \mathrm{lrw}(F(v_2)) = k$, then by Proposition 3.1 in order to decide the linear rank-width of $F(u)$ we need to know the linear rank-width of $F(u)\backslash V(F(v))$. We can recursively call the algorithm on $F(u)\backslash V(F(v))$, but this would not give a linear-time algorithm, and similar situations can happen in $F(u)\backslash V(F(v))$. The idea introduced in [21] to cope with this difficulty was to keep in u the linear rank-width of the subtrees that may cause a recursive call to the algorithm because of the presence of ℓ-critical nodes for $\ell \le k$. For instance, in $F_0 := F(u)\backslash V(F(v))$ we may have a k_0-critical node w with $k_0 := \max\{\mathrm{lrw}(F_0(v)) \mid v \text{ is a child of } u \text{ in } F_0\}$, and then we may need the linear rank-width of $F_0\backslash V(F_0(w))$ to answer, and so on.

Similar to trees, in the case of a distance-hereditary graph G, we will start by rooting the canonical decomposition D of G, and for each bag B with the parent bag B' and the component T of $D \setminus V(B')$ containing B, we compute $f_D(B', T)$. For this, we define a *k-critical bag* in the same fashion. Let D' be the canonical limb with respect to B' and an unmarked vertex $y \in V(T)$ where y is represented by some vertex in B'. Now, if B'' is a k-critical bag in D', as in the case of trees we need to compute $f_{D'}(B'', T')$ where T' is the component of $D' \setminus V(B'')$ containing the parent of B''. However, contrary to the case of trees, the canonical limb $\mathcal{LC}[D', B'', y']$, for some unmarked vertex y' in $V(T')$, is not necessarily an induced subgraph of D. We overcome this difficulty by showing that the order in which we can recursively compute canonical limbs is not important, which enables us to store information similar to the cases of trees.

As we explained above, we investigate useful properties of canonical limbs which are related to the orders from which canonical limbs are taken. Note that for recursively taking limbs, we need to transform an obtained limb into a canonical limb because limbs are only defined on canonical decompositions. Let D be the canonical decomposition of a connected distance-hereditary graph.

Proposition 5.1 *Let B_1 and B_2 be two distinct bags of D and for each $i \in \{1, 2\}$, let T_i be the component of $D \setminus V(B_i)$, $w_i := \zeta_b(D, B_i, T_i)$ and y_i be an unmarked vertex of D represented by w_i such that*

- *T_1 contains the bag B_2 and T_2 contains the bag B_1, and*
- *$V(B_1)$ induces a bag in $\mathcal{LC}_D[B_2, y_2]$, and $V(B_2)$ induces a bag in $\mathcal{LC}_D[B_1, y_1]$.*

We define that

- *$B_1' := (\mathcal{LC}_D[B_2, y_2])[V(B_1)]$,*
- *$B_2' := (\mathcal{LC}_D[B_1, y_1])[V(B_2)]$,*
- *y_1' is an unmarked vertex of $\mathcal{LC}_D[B_2, y_2]$ represented by w_1, and*
- *y_2' is an unmarked vertex of $\mathcal{LC}_D[B_1, y_1]$ represented by w_2.*

Then $\mathcal{LC}_{\mathcal{LC}_D[B_1,y_1]}[B_2', y_2']$ is locally equivalent to $\mathcal{LC}_{\mathcal{LC}_D[B_2,y_2]}[B_1', y_1']$.

Proof For each $i \in \{1, 2\}$, let $v_i := \zeta_c(D, B_i, T_i)$. By Lemma 3.6, there exists a canonical decomposition D' locally equivalent to D such that for each $i \in \{1, 2\}$, w_i is a leaf of $D'[V(B_i)]$ in D'. For each $i \in \{1, 2\}$, let $P_i := D'[V(B_i)]$, $T_i' := D'[V(T_i)]$, and z_i be an unmarked vertex of D' represented by w_i. We define that

- $T' := D'[V(T_1') \cap V(T_2')]$,
- $P_1' := (\mathcal{LC}_{D'}[P_2, z_2])[V(P_1)]$,
- $P_2' := (\mathcal{LC}_{D'}[P_1, z_1])[V(P_2)]$,
- z_1' is an unmarked vertex of $\mathcal{LC}_{D'}[P_2, z_2]$ represented by w_1,
- z_2' is an unmarked vertex of $\mathcal{LC}_{D'}[P_1, z_1]$ represented by w_2.

Since D is locally equivalent to D', by Proposition 3.4, $\mathcal{LC}_D[B_1, y_1]$ is locally equivalent to $\mathcal{LC}_{D'}[P_1, z_1]$. Again, since $\mathcal{LC}_D[B_1, y_1]$ is locally equivalent to $\mathcal{LC}_{D'}[P_1, z_1]$, by Proposition 3.4,

$$\mathcal{LC}_{\mathcal{LC}_D[B_1,y_1]}[B_2', y_2'] \text{ is locally equivalent to } \mathcal{LC}_{\mathcal{LC}_{D'}[P_1,z_1]}[P_2', z_2'].$$

Similarly, we obtain that

$$\mathcal{LC}_{\mathcal{LC}_D[B_2,y_2]}[B_1', y_1'] \text{ is locally equivalent to } \mathcal{LC}_{\mathcal{LC}_{D'}[P_2,z_2]}[P_1', z_1'].$$

Since each v_i is a leaf of P_i in D', $\mathcal{LC}_{\mathcal{LC}_{D'}[P_1,z_1]}[P_2', z_2']$ and $\mathcal{LC}_{\mathcal{LC}_{D'}[P_2,z_2]}[P_1', z_1']$ are canonical decompositions obtained from $T'\backslash v_1\backslash v_2$ by recomposing if neceesary. From the assumption that $V(B_1)$ induces a bag in $\mathcal{LC}_D[B_2, y_2]$, and $V(B_2)$ induces a bag in $\mathcal{LC}_D[B_1, y_1]$, $V(B_1)$ and $V(B_2)$ also induce bags in $\mathcal{LC}_{D'}[P_2, z_2]$ and $\mathcal{LC}_{D'}[P_1, z_1]$, respectively. Thus the order of taking canonical limbs with respect to P_1 and P_2 does not affect on the resulting decompositions, and it implies that

$$\mathcal{LC}_{\mathcal{LC}_{D'}[P_1,z_1]}[P_2', z_2'] = \mathcal{LC}_{\mathcal{LC}_{D'}[P_2,z_2]}[P_1', z_1'].$$

Therefore, $\mathcal{LC}_{\mathcal{LC}_D[B_1,y_1]}[B_2', y_2']$ is locally equivalent to $\mathcal{LC}_{\mathcal{LC}_D[B_2,y_2]}[B_1', y_1']$. □

Proposition 5.2 *Let B_1 and B_2 be two distinct bags of D. Let T_1 be a component of $D\backslash V(B_1)$ that does not contain B_2, and T_2 be the component of $D\backslash V(B_2)$ containing the bag B_1. For $i \in \{1, 2\}$, let $w_i := \zeta_b(D, B_i, T_i)$, and y_i be an unmarked vertex of D represented by w_i. If $V(B_1)$ induces a bag B_1' of $\mathcal{LC}_D[B_2, y_2]$, then $\mathcal{LC}_D[B_1, y_1]$ is locally equivalent to $\mathcal{LC}_{\mathcal{LC}_D[B_2,y_2]}[B_1', y_1']$, where y_1' is an unmarked vertex of $\mathcal{LC}_D[B_2, y_2]$ represented by w_1.*

Proof Suppose $V(B_1)$ induces a bag B_1' of $\mathcal{LC}_D[B_2, y_2]$ and y_1' is an unmarked vertex represented in $\mathcal{LC}_D[B_2, y_2]$ by w_1. By Lemma 3.6, there exists a canonical decomposition D' locally equivalent to D such that w_2 is a leaf of a star bag $D'[V(B_2)]$. We define

- $P_1 := D'[V(B_1)]$,
- $P_2 := D'[V(B_2)]$,
- for each $i \in \{1, 2\}$, z_i is an unmarked vertex of D' represented by w_i,
- $P_1' := (\mathcal{LC}_{D'}[P_2, z_2])[V(B_1)]$, and
- z_1' is an unmarked vertex of $\mathcal{LC}_{D'}[P_2, z_2]$ represented by w_1.

Since D is locally equivalent to D', by Proposition 3.4, $\mathcal{LC}_D[B_1, y_1]$ is locally equivalent to $\mathcal{LC}_{D'}[P_1, z_1]$. Similarly, we obtain that $\mathcal{LC}_D[B_2, y_2]$ is locally equivalent to $\mathcal{LC}_{D'}[P_2, z_2]$. Since $\mathcal{LC}_D[B_2, y_2]$ is locally equivalent to $\mathcal{LC}_{D'}[P_2, z_2]$, by Proposition 3.4,

$$\mathcal{LC}_{\mathcal{LC}_D[B_2,y_2]}[B_1', y_1'] \text{ is locally equivalent to } \mathcal{LC}_{\mathcal{LC}_{D'}[P_2,z_2]}[P_1', z_1'].$$

Since w_2 is a leaf of P_2 in D', $\mathcal{LC}_{D'}[P_1, z_1] = \mathcal{LC}_{\mathcal{LC}_{D'}[P_2,z_2]}[P_1', z_1']$, and therefore, $\mathcal{LC}_D[B_1, y_1]$ is locally equivalent to $\mathcal{LC}_{\mathcal{LC}_D[B_2,y_2]}[B_1', y_1']$, as required. □

6 Computing the Linear Rank-Width of Distance-Hereditary Graphs

We describe an algorithm to compute the linear rank-width of distance-hereditary graphs. Since the linear rank-width of a graph is the maximum linear rank-width over all its connected components, we will focus on connected distance-hereditary graphs.

Theorem 6.1 *The linear rank-width of every connected distance-hereditary graph with n vertices can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$. Moreover, a linear layout* *of the graph witnessing the linear rank-width can be computed with the same time complexity.*

As explained in Sect. 5, the main idea consists of rooting the canonical decomposition D of a connected distance-hereditary graph and associating each bag B of D with a canonical limb $\mathcal{LC}_D[B', y]$ where B' is the parent of B and y is an unmarked vertex in some descendant bag of B, and computing the linear rank-width of $\mathcal{LG}_D[B', y]$. Following Theorem 4.1, in order to compute the linear rank-width of $\mathcal{LG}_D[B', y]$, we need to check the linear rank-width of proper limbs obtained from $\mathcal{LC}_D[B', y]$ by removing some bags of $\mathcal{LC}_D[B', y]$. Basically, we need to take canonical limbs recursively from this reason. In contrast to the case of forests for computing linear rank-width, the associated canonical limbs here are not necessarily sub-decompositions of the original decomposition, and thus, it is not at all trivial how to store values to use in the next steps. The crucial point of achieving our running time is to overcome this problem using the results in Sect. 5.

Rooted Decomposition Trees. We define the notion of *rooted decomposition trees*. A decomposition tree is *rooted* if we distinguish either a node or an edge and call it the *root* of the tree. Let T be a rooted decomposition tree with the root r. A node v is a descendant of a node v' if v' is in the unique path from the root to v, and when r is an edge, this path contains both end nodes of r. If v is a descendant of v' and v and v' are adjacent, then we call v a *child* of v' and v' the *parent* of v. Observe from the definition of descendants that if $r = vv'$, then v is the parent of v' and also v' is the parent of v. We allow this tricky part for a technical reason. A node in T is called a *non-root node* if it is not the root node.

Two nodes v and v' are called *comparable* if one node is a descendant of the other one. Otherwise, they are called *incomparable*. Recall that for each node v of T and each canonical decomposition D with T as its decomposition tree we write $\mathsf{b}_D(v)$ to denote the bag of D with which it is in correspondence. For convenience, let $\mathsf{pb}_D(v) := \mathsf{b}_D(v')$ with v' the parent of v.

Let D be the canonical decomposition of a connected distance-hereditary graph G and let T be its decomposition tree rooted at r. Let $B := \mathsf{b}_D(v)$ for some non-root node v of T, and let y be an unmarked vertex of D that is represented by a vertex of B. We define the root of the decomposition tree $\widetilde{T}$ of $\mathcal{LC}_D[B, y]$ as follows. We assume that $\widetilde{T}$ is obtained from T by removing v, and possibly adding an edge or identifying two nodes following the definition of canonical limbs. If two comparable nodes w and w' with w the parent of w' are identified, then let w be the identified node. Otherwise, we give a new label for the identified node.

(1) If r exists in $\widetilde{T}$, then we assign r as the root of $\widetilde{T}$. In the other cases, we can observe that either
   - r is the root node and $\mathsf{b}_D(r)$ is removed when taking the canonical limb or
   - r is the root edge, and a bag $\mathsf{b}_D(r')$ is removed where r' is a node incident with the root edge, when taking the canonical limb.

(2) If the removed node has one neighbor in $T \backslash r$, then we assign this neighbor as the root of $\widetilde{T}$.

(3) If the removed node has two neighbors in $T \backslash r$ and they are linked by a new edge in $\widetilde{T}$, then we assign the new edge as the root of $\widetilde{T}$.

(4) If the removed node has two neighbors in $T\backslash r$ and they are identified in $\widetilde{T}$, then we assign the new node as the root of $\widetilde{T}$.

The following observation is easy to check from the definition of rooted decomposition trees of canonical limbs.

Fact 6.2 *If w is a non-root node of the rooted decomposition tree $\widetilde{T}$ of a canonical limb $\mathcal{LC}_D[B, y]$, then w is also a non-root node of T with the property that $V(\mathsf{b}_D(w)) = V(\mathsf{b}_{\mathcal{LC}_D[B,y]}(w))$.*

For a non-root node v, we will frequently take two types of canonical limbs; one is with respect to $\mathsf{pb}_D(v)$ and the component of $D\backslash V(\mathsf{pb}_D(v))$ containing $\mathsf{b}_D(v)$, and the other is with respect to $\mathsf{b}_D(v)$ and the component of $D\backslash V(\mathsf{b}_D(v))$ containing $\mathsf{pb}_D(v)$. For convenience, we define the following notations. For every non-root node v of T with the parent node v', we define that

- $\mathcal{T}_1[D, v]$ is the component of $D\backslash V(\mathsf{b}_D(v'))$ containing $\mathsf{b}_D(v)$,
- $\mathcal{T}_2[D, v]$ is the component of $D\backslash V(\mathsf{b}_D(v))$ containing $\mathsf{b}_D(v')$,
- $f_1(D, v) := f_D(\mathsf{pb}_D(v), \mathcal{T}_1[D, v])$,
- $f_2(D, v) := f_D(\mathsf{b}_D(v), \mathcal{T}_2[D, v])$,
- $\zeta_1(D, v) := \zeta_b(D, \mathsf{b}_D(v'), \mathcal{T}_1[D, v])$, and
- $\zeta_2(D, v) := \zeta_b(D, \mathsf{b}_D(v), \mathcal{T}_2[D, v])$.

k-**Critical Nodes.** A node v of T is called *k-critical* if $f_1(D, v) = k$ and v has two children v_1 and v_2 such that $f_1(D, v_1) = f_1(D, v_2) = k$.

From now on, we define some sequences of canonical limbs, which will be taken sequentially in our algorithm. We recall that $\text{lrw}(G) \le \log_2|V(G)|$ by Theorem 2.1 and Lemma 2.2. For convenience, let

$$\eta := \lfloor \log_2|V(G)| \rfloor.$$

For each non-root node v of T, we define recursively the following. We first choose an unmarked vertex y of D represented by $\zeta_1(D, v)$, and

- let D^v_η be any canonical limb $\mathcal{LC}_D[\mathsf{pb}_D(v), y]$, and let T^v_η be the rooted decomposition tree of D^v_η.

For each $1 \le j \le \eta$, let $\alpha^v_j := \max\{f_1(D^v_j, w) \mid w \text{ is a non-root node of } T^v_j\}$, and we define D^v_{j-1} and T^v_{j-1} as follows:

(1) If $\alpha^v_j \ne j$, then let $D^v_{j-1} := D^v_j$ and $T^v_{j-1} := T^v_j$.
(2) If $\alpha^v_j = j$ and one of the following is satisfied, then let $D^v_{j-1} := D^v_j$ and $T^v_{j-1} := T^v_j$.
   - T^v_j has a node with at least 3 children w such that $f_1(D^v_j, w) = j$.
   - T^v_j has two incomparable nodes v_1 and v_2 where v_1 is a j-critical node v_1 and $f_1(D^v_j, v_2) = j$.
   - T^v_j has no j-critical nodes.
(3) Otherwise, T^v_j has the unique j-critical node v_c. In this case, we choose an unmarked vertex y of D^v_j represented by $\zeta_2(D^v_j, v_c)$ and let $D^v_{j-1} := \mathcal{LC}_{D^v_j}[\mathsf{b}_{D^v_j}(v_c), y]$ and let T^v_{j-1} be the rooted decomposition tree of D^v_{j-1}.

Lastly for each $0 \leq j \leq \eta$, let $\beta_j^v := \mathrm{lrw}(\mathcal{G}[D_j^v])$.

Roughly, for a non-root node v with parent v', and $w_1, w_2, \ldots, w_p$ as children, we define a sequence of 4-tuples $(D_j^v, T_j^v, \alpha_j^v, \beta_j^v)$, for each $1 \leq j \leq \eta = \lfloor \log_2 |V(G)| \rfloor$ where D_j^v is some rooted decomposition, T_j^v is its rooted decomposition tree, β_j^v is the linear rank-width of $\mathcal{G}[D_j^v]$, and α_j^v is the maximum over $\{\beta_j^{w_i} \mid 1 \leq i \leq p\}$. These 4-tuples are the information needed to avoid the recursive calls to the algorithm (as already explained in Sect. 5). D_η^v is any limb of $D \backslash V(\mathsf{b}_D(v'))$ associated with $\mathcal{T}_1[D, v]$. These 4-tuples are motivated by the following. Let k be the maximum over the linear rank-width of the $\mathcal{G}[D_\eta^{w_i}]$'s. If any of the conditions in (2) above is verified by T_η^v, then we can decide easily the linear rank-width of $\mathcal{G}[D_\eta^v]$. Otherwise, there is exactly one critical node v_c in one of the $T_\eta^{w_i}$'s. By Theorem 4.1 we need to compute the linear rank-width of $\mathcal{G}[D']$ where D' is defined as one limb of $D_\eta^v \backslash V(\mathsf{b}_{D_\eta^v}(v_c))$ associated with $\mathcal{T}_2[D_\eta^v, v_c]$. We define D_{k-1}^v as this D', and D_j^v as D_η^v for all $k \leq j \leq \eta - 1$, as we do not know whether we will need some of these D_j^vs in the future. Indeed, for instance the same situation can happen in D_{k-1}^v with some other ℓ-critical node w with $\ell := \max\{\beta_{k-1}^{w'} \mid w' \text{ a child of the root of } D_{k-1}^v\}$, hence we need again to compute the linear rank-width of $\mathcal{G}[D'']$ with D'' defined as one limb of $D_{k-1}^v \backslash V(\mathsf{b}_{D_{k-1}^v}(w))$ associated with $\mathcal{T}_2[D_{k-1}^c, w]$, and this D'' is denoted as D_j^v for $k - 2 \leq j \leq \ell - 1$.

The existence of the unique j-critical node in (3) is verified in the next proposition.

Proposition 6.3 *Let $0 \leq j \leq \eta$ and let v be a non-root node of T such that $\alpha_j^v \leq j$ and T_j^v contains neither*

- *a node having at least 3 children w with $f_1(D_j^v, w) = \alpha_j^v$, nor*
- *two incomparable nodes v_1 and v_2 having the property that v_1 is an α_j^v-critical node and $f_1(D_j^v, v_2) = \alpha_j^v$.*

Let w be an α_j^v-critical node of T_j^v. Then w is the unique α_j^v-critical vertex of T_j^v. Moreover, $\mathrm{lrw}(\mathcal{G}[D_j^v]) = \alpha_j^v + 1$ *if and only if* $\mathrm{lrw}(\mathcal{G}[D_{j-1}^v]) = f_2(D_j^v, w) = \alpha_j^v$.

Proof Let $k := \alpha_j^v$. We first show that w is the unique k-critical node of T_j^v. Let w' be a k-critical node of T_j^v that is distinct from w. From the second assumption, w and w' must be comparable in T_j^v. Without loss of generality, we may assume that w is a descendant of w' in T_j^v. Then by the definition of k-criticality, w' has a child w'' such that $f_1(D_j^v, w'') = k$ and w is not a descendant of w'' in T_j^v, contradicting to the second assumption.

Now we claim that $\mathrm{lrw}(\mathcal{G}[D_j^v]) = k + 1$ if and only if $f_2(D_j^v, w) = k$. By the assumption on k and by Theorem 4.1, $\mathrm{lrw}(\mathcal{G}[D_j^v]) \leq k + 1$. Let w_1 and w_2 be the two children of w such that $f_1(D_j^v, w_1) = f_1(D_j^v, w_2) = k$. By assumption, every other child w' of w satisfies that $f_1(D_j^v, w') \leq k - 1$.

If $f_2(D_j^v, w) = k$, then clearly we have $\mathrm{lrw}(\mathcal{G}[D_j^v]) \geq k + 1$ by Theorem 4.1. For the forward direction, suppose that $\mathrm{lrw}(\mathcal{G}[D_j^v]) \geq k + 1$. Since T_j^v contains no node having at least three children w such that $f_1(D_j^v, w) = k$, by Theorem 4.1, there should exist a k-critical node v_c of T_j^v such that $f_2(D_j^v, v_c) = k$. Since w is the unique k-critical node of T_j^v, $w = v_c$ and $f_2(D_j^v, w) = \mathrm{lrw}(\mathcal{G}[D_{j-1}^v]) = k$, as required. □

Let v be a non-root node of T. From Theorem 4.1, we can easily observe that $\alpha^v_\eta \leq \text{lrw}(\mathcal{G}[D^v_\eta)]) \leq \alpha^v_\eta + 1$. By Proposition 6.3, if T^v_η has no unique critical node, then it is easy to determine β^v_η, and otherwise the computation of β^v_η can be reduced to the computation of $f_2(D^v_\eta, v_c)$ where v_c is the unique α^v_η-critical node of T^v_η. In order to compute it, we can recursively call the algorithm on $\mathcal{G}[D^v_{\alpha^v_\eta - 1}]$. However, we will prove that these recursive calls are not needed if we store the values β^v_j.

Lemma 6.4 *Let v be a non-root node of T. Let i be an integer such that $0 \leq i < \eta$. If $\alpha^v_i \leq i$, then $\alpha^v_{i+1} \leq i + 1$.*

Proof Suppose that $\alpha^v_{i+1} \geq i + 2$. By the definition of D^v_i, $D^v_i = D^v_{i+1}$ and therefore, $\alpha^v_i \geq i + 2$, which yields a contradiction. □

Our Algorithm. Now we are ready to present and analyze our algorithm. We describe the algorithm explicitly in Algorithm 2. First, we modify the given decomposition as follows. For the canonical decomposition D' of a connected distance-hereditary graph G, we modify D' into a canonical decomposition D by adding a root bag R and making it adjacent to a bag R' of D' so that $f_1(D, v) = \text{lrw}(G)$, where v is the node corresponding to the bag R'. We call (D, R) a *modified canonical decomposition of G*. Let T be the decomposition tree of the new canonical decomposition D. Algorithm 2 computes $\beta^v_i = \text{lrw}(\mathcal{G}[D^v_i])$ for all non-root nodes v of T and all integers i such that $\alpha^v_i \leq i$. We recall that $\eta = \lfloor \log_2 |V(G)| \rfloor$. We refer to the correctness proof for the exact description of the algorithm.

We present the subroutine **Limb** which computes a canonical limb associated with $\mathcal{T}_i[D, w]$ for $i \in \{1, 2\}$ in Algorithm 1.

Algorithm 1: Limb($D, T, \{\gamma(v) \mid v \in V(T\backslash r)\}, w \in V(T\backslash r), i \in \{1, 2\}$).

Input: A canonical decomposition D of a connected distance-hereditary graph, its rooted decomposition tree T with the root r, $\{\gamma(v) \in \mathbb{N} \mid v \in V(T\backslash r)\}$, a non-root node w of T, and $i \in \{1, 2\}$.

Output: A canonical decomposition D' of D associated with $\mathcal{T}_i[D, w]$, its rooted decomposition tree T' with the root r', $\{\gamma(v) \mid v \in V(T'\backslash r')\}$, and α.

Let w' be the parent of w;
if $i = 1$ **then** choose an unmarked vertex y of D represented by $\zeta_1(D, w)$ and $v \leftarrow w'$;
else choose an unmarked vertex y of D represented by $\zeta_2(D, w)$ and $v \leftarrow w$;
$D' \leftarrow \mathcal{LC}_D[\mathsf{b}_D(v), y]$ and obtain T' from T and assign the root r' of T';
$\alpha \leftarrow \max\{\gamma(v) \mid v \in V(T'\backslash r')\}$;
return $(D', T', \{\gamma(v) \mid v \in V(T'\backslash r')\}, \alpha)$;

Correctness of the Algorithm. The following proposition has a key role in the algorithm. It mainly uses the results in Sect. 5.

Proposition 6.5 *Let v be a non-root node of T and let $0 \leq i \leq \eta$ such that $\alpha^v_i \leq i$. If w is a non-root node of T^v_i, then, $\beta^w_i = f_1[D^v_i, w]$.*

Proof Let w be a non-root node of T^v_i. By Fact 6.2, for each $i + 1 \leq j \leq \eta$, $w \in V(T^v_j)$ and hence $w \in V(T)$. Moreover, since $\alpha^v_i \leq i$, by Lemma 6.4, $\alpha^v_j \leq j$ for all $i + 1 \leq j \leq \eta$. For each $i \leq j \leq \eta$, we define that

Algorithm 2: COMPUTE LINEAR RANK- WIDTH OF CONNECTED DISTANCE-HEREDITARY GRAPHS

Input: A connected distance-hereditary graph G.
Output: The linear rank-width of G.

Compute a modified canonical decomposition (D, R) of G, and the decomposition tree T of D with the root node r;
Let $\beta_i^v \leftarrow 0$ for each non-root node v and each $0 \le i \le \eta$;
For each non-root leaf node v in T and each $0 \le i \le \eta$, let $\beta_i^v \leftarrow 1$;
$\Gamma \leftarrow \{\beta_i^v \mid v \in V(T \backslash r), 0 \le i \le \eta\}$;
while *T has a non-root node v where β_η^v is not computed* **do**
    Let v be a non-root node in T where $\beta_\eta^v = 0$, but $\beta_\eta^{v'} \neq 0$ for each child v' of v;
        /* Compute canonical limbs necessary to compute $f_1(D, v)$/*
    $(D_\eta^v, T_\eta^v, \Gamma_\eta^v, \alpha_\eta^v) \leftarrow$ **Limb**$(D, T, \Gamma, v, 1)$;
    Let S be a stack; $i \leftarrow \alpha_\eta^v$; $k \leftarrow \alpha_\eta^v$;
    while *(true)* **do**
        if *(T_i^v has a node having at least 3 children v' with $\beta_i^{v'} = i$) or (T_i^v has two incomparable nodes v_1 and v_2 having the property that v_1 is an i-critical node and $\beta_i^{v_2} = i$) or (T_i^v has no i-critical nodes)* **then**
            Stop this loop
        Find the unique i-critical node v_c of T_i^v;
        $(D_{i-1}^v, T_{i-1}^v, \Gamma_{i-1}^v, \alpha_{i-1}^v) \leftarrow$ **Limb**$(D_i^v, T_i^v, \Gamma_i^v, v_c, 2)$;
        push(S, i) and $i \leftarrow \alpha_{i-1}^v$;
        /* Recursively compute β_i^v for all i with $\alpha_i^v \le i$/*
    if *(T_i^v has a node having at least 3 children v' with $\beta_i^{v'} = i$) or (T_i^v has two incomparable nodes v_1 and v_2 with the property that v_1 is an i-critical node and $\beta_i^{v_2} = i$)* **then** $\beta_i^v \leftarrow i + 1$;
    else $\beta_i^v \leftarrow i$;
    while $(S \neq \emptyset)$ **do**
        $j \leftarrow$ pull(S);
        if $\beta_i^v = j$ **then** $\beta_j^v \leftarrow j + 1$;
        else $\beta_j^v \leftarrow j$;
        for $\ell \leftarrow i + 1$ ***to*** $j - 1$ **do**
            $\beta_\ell^v \leftarrow \beta_i^v$;
        $i \leftarrow j$;
    for $j \leftarrow k + 1$ ***to*** η **do**
        $\beta_j^v \leftarrow \beta_k^v$;
Let r' be the unique neighbor of the root and **return** $\beta_\eta^{r'}$;

- y_j is an unmarked vertex of D_j^v represented by the marked vertex $\zeta_1(D_j^v, w)$.

Now, we claim that for each $i \le j \le \eta$,

- $\mathcal{LC}_{D_j^v}[\mathsf{pb}_{D_j^v}(w), y_j]$ is locally equivalent to D_j^w.

If it is true, then we obtain that $\mathcal{LC}_{D_i^v}[\mathsf{pb}_{D_i^v}(w), y_i]$ is locally equivalent to D_i^w, which implies that $\beta_i^w = f_1[D_i^v, w]$. We prove it by induction on $\eta - j$.

If $j = \eta$, then both D^v_η and D^w_η are canonical limbs of D. Since w is a non-root node of T^v_η, $V(\mathsf{b}_D(w))$ induces a bag in D^v_η, and hence by Proposition 5.2, D^w_η is locally equivalent to $\mathcal{LC}_{D^v_\eta}[\mathsf{pb}_{D^v_\eta}(w), y_\eta]$.

Now let us assume that $i \leq j < \eta$. By induction hypothesis D^w_{j+1} is locally equivalent to $\mathcal{LC}_{D^v_{j+1}}[\mathsf{pb}_{D^v_{j+1}}(w), y_{j+1}]$. Assume first that $\alpha^v_{j+1} \leq j$. Then, by Proposition 5.2, we have that $\alpha^w_{j+1} \leq j$. In that case, by the definition, we have $D^v_j = D^v_{j+1}$ and $D^w_j = D^w_{j+1}$, and we conclude the statement.

Assume now that $\alpha^v_{j+1} = j+1$. Since $\alpha^v_{j+1} = j+1$ and $\alpha^v_j \leq j$, T^v_{j+1} should have a unique $(j+1)$-critical node v_c such that $D^v_j = \mathcal{LC}_{D^v_{j+1}}[\mathsf{b}_{D^v_{j+1}}(v_c), y_c]$ for some unmarked vertex y_c of D^v_{j+1} represented by $\zeta_2(D^v_{j+1}, v_c)$. We distinguish two cases: either v_c is incomparable with w in T^v_{j+1}, or v_c is a descendant of w in T^v_{j+1}. Since w is a node of T^v_j, w cannot be a descendant of v_c.

Case 1. *v_c is incomparable with w in T^v_{j+1}.*

Since v_c is incomparable with w in T^v_{j+1} and v_c is the unique $(j+1)$-critical node in T^v_{j+1}, there is no $(j+1)$-critical node in T^w_{j+1}. Hence, $D^w_j = D^w_{j+1}$ by definition. Also, by Proposition 5.2,

- $\mathcal{LC}_{D^v_j}[\mathsf{pb}_{D^v_j}(w), y_j]$ is locally equivalent to $\mathcal{LC}_{D^v_{j+1}}[\mathsf{pb}_{D^v_{j+1}}(w), y_{j+1}]$.

Hence, we can conclude that D^w_j is locally equivalent to $\mathcal{LC}_{D^v_j}[\mathsf{pb}_{D^v_j}(w), y_j]$ because D^w_{j+1} is locally equivalent to $\mathcal{LC}_{D^v_{j+1}}[\mathsf{pb}_{D^v_{j+1}}(w), y_{j+1}]$.

Case 2. *v_c is a descendant of w in T^v_{j+1}.*

If v_c is a child of w in T^v_{j+1} and the bag $\mathsf{b}_{D^v_{j+1}}(w)$ has size 3, then T^v_j cannot contain w as a node, and this contradicts the assumption that w is a node of T^v_j. Therefore, we may assume that either

(1) $|\mathsf{b}_{D_{j+1}}(w)| \geq 4$, or
(2) $|\mathsf{b}_{D_{j+1}}(w)| = 3$ and v_c is not a child of w in T^v_{j+1}.

This implies that v_c is a node of the decomposition tree of $\mathcal{LC}_{D^v_{j+1}}[\mathsf{pb}_{D^v_{j+1}}(w), y_{j+1}]$. Let $D' := \mathcal{LC}_{D^v_{j+1}}[\mathsf{pb}_{D^v_{j+1}}(w), y_{j+1}]$. By induction hypothesis, we know that D^w_{j+1} is locally equivalent to D'. Note that, by definition v_c is also the unique critical node of T^w_{j+1}, and

- $D^w_j = \mathcal{LC}_{D^w_{j+1}}[\mathsf{b}_{D^w_{j+1}}(v_c), z]$ for some unmarked vertex z of D^w_{j+1} represented by $\zeta_2(D^w_{j+1}, v_c)$.

Also, by Proposition 5.1,

- $\mathcal{LC}_{D^v_j}[\mathsf{pb}_{D^v_j}(w), y_j]$ is locally equivalent to $\mathcal{LC}_{D'}[\mathsf{b}_{D'}(v_c), z']$ where z' is an unmarked vertex of D' represented by $\zeta_2(D', v_c)$.

Since D' is locally equivalent to D^w_{j+1}, $\mathcal{LC}_{D^v_j}[\mathsf{pb}_{D^v_j}(w), y_j]$ is locally equivalent to D^w_j, and this concludes the proof. □

Proof of Theorem 6.1 *We first show that Algorithm 2 correctly computes the linear rank-width of G. If $|V(G)| \leq 1$, then* $\mathrm{lrw}(G) = 0$ *from the definition. We may assume that $|V(G)| \geq 2$. Let (D, R) be a modified canonical decomposition of G and let T*

be the canonical decomposition tree of D and let r' be the unique neighbor of the root of T. As we observed, we have that $\mathrm{lrw}(G) = \mathrm{lrw}(\mathcal{G}[D_\eta^{r'}]) = \beta_\eta^{r'}$*, and want to prove that Algorithm 2 correctly outputs $\beta_\eta^{r'}$. We claim that for each non-root node v of T and $0 \le i \le \eta$ such that $\alpha_i^v \le i$, Algorithm 2 correctly computes β_i^v.*

Suppose v is a non-root leaf node of T. Since every canonical limb is connected by Lemma 3.2 and $|V(G)| \ge 2$, D_η^v is isomorphic to either a complete graph or a star with at least two vertices. Thus, $\mathrm{lrw}(\mathcal{G}[D_\eta^v]) = 1$*, and by construction for each $0 \le i \le \eta$, $D_i^v = D_\eta^v$, and Line 3 correctly puts these values.*

*We assume that v is a non-root node in T that is not a leaf, and for all its descendants v' and integers $0 \le \ell \le \eta$ with $\alpha_\ell^{v'} \le \ell$, $\beta_\ell^{v'}$ is computed (*i.e. $\beta_\ell^{v'} \ne 0$*). We claim that Line 10-15 recursively computes D_i^v for each i where $\alpha_i^v \le i$. We first remark that for computing α_i^v of T_i^v, we use the fact that for each non-root node w of T_i^v, $\beta_i^w = f_1[D_i^v, w]$ from Proposition 6.5. So, $\alpha_i^v = \max\{\beta_i^w \mid w$ a non-root node w of $T_i^v\}$.*

Let $i \in \{0, 1, \dots, \eta\}$ such that $\alpha_i^v \le i$. If $\alpha_i^v < i$, then by the definition, $T_{i-1}^v = T_i^v$ and thus, we take $D_{i-1}^v = D_i^v$. We may assume that $\alpha_i^v = i$. If either T_i^v has a node with at least 3 children v' such that $\beta_i^{v'} = i$, or T_i^v has two incomparable nodes v_1 and v_2 with v_1 an i-critical node and $\beta_i^{v_2} = i$, then from the definition of D_i^v, we have that $D_{i-1}^v = D_i^v$ and for all $0 \le \ell \le i-1$, $\alpha_\ell^v = i > \ell$. Since we do not need to evaluate β_ℓ^v when $\alpha_\ell^v > \ell$, we stop the loop. If T_i^v has no i-critical node, then $\beta_i^v = \alpha_i^v = i$, that is, the β_i^v value cannot be increased by one. In this case, we also stop the loop. These 3 cases are the conditions in Line 11.

Suppose neither of the conditions in Line 11 occur. Then by Proposition 6.3, T_i^v has a unique i-critical bag v_c and D_{i-1}^v is equal to a canonical limb $\mathcal{LC}_{D_i^v}[\mathsf{b}_{D_i^v}(v_c), y]$ where y is some unmarked vertex of D_i^v represented by $\zeta_2(D_i^v, v_c)$. So, we compute D_{i-1}^v from D_i^v, the rooted decomposition tree T_{i-1}^v of D_{i-1}^v and compute subsequently α_{i-1}^v. Notice that for all $\alpha_{i-1}^v \le \ell \le i-1$, $D_\ell^v = D_{i-1}^v$ and thus it is sufficient in the next iteration to deal with $D_{\alpha_{i-1}^v}^v$ directly. Thus, Line 10-15 correctly computes canonical decompositions D_i^v for each i where $\alpha_i^v = i$.

Now we verify the procedure of computing β_j^v in Line 17. Let $0 \le \ell \le \eta$ be the minimum integer such that $\alpha_\ell^v = \ell$. If $\ell = 0$, then $\beta_\ell^v = 1$. Suppose $\ell \ge 1$. Then since $\alpha_{\ell-1}^v > \ell - 1$, by Theorem 4.1, we have that

(1) $\beta_\ell^v = \ell + 1$ if either T_ℓ^v has a node having at least 3 children v' with $\beta_\ell^{v'} = \ell$, or two incomparable nodes v_1 and v_2 with the property that v_1 is an i-critical node and $\beta_i^{v_2} = i$,
(2) $\beta_\ell^v = \ell$ if otherwise.

Thus, Line 17 correctly computes it.

In the loop in Line 10, we use a stack to pile up the integers i such that T_i^v has the unique i-critical node. When T_i^v has the unique i-critical node, by Proposition 6.3,

(1) $\beta_i^v = i + 1$ if $\beta_{i-1}^v = i$, and
(2) $\beta_i^v = i$ if $\beta_{i-1}^v \le i - 1$.

So, from the lower value in the stack we can compute β_i^v recursively. From Line 19 to Line 26, Algorithm 2 computes all β_i^v correctly where $\alpha_i^v \le i$, and in particular, it *computes β^v_η. Therefore, at the end of the algorithm, it computes $\beta^{r'}_\eta$ that is equal to the linear rank-width of G.*

The Running Time of the Algorithm. *Let us now analyze its running time. Let n and m be the number of vertices and edges of G. Its canonical decomposition can be computed in time $\mathcal{O}(n+m)$ by Theorem 2.7, and one can compute a modified canonical decomposition (D, R) in constant time. Note that the number of bags in D is bounded by $\mathcal{O}(n)$ (see [13, Lemma 2.2]).*

We first remark that Algorithm 1 runs in time $\mathcal{O}(n)$. This is because when we take a limb from a canonical decomposition, we need to take a local complementation or a pivoting on a sub-decomposition, and in the worst case, we may visit each bag to apply these operations. The decomposition tree and α, β values can be obtained in linear time.

Now we observe the running time of Algorithm 2. The number of iterations of the whole loop from Line 6 to Line 27 is at most $\mathcal{O}(n)$ because it runs in as many as the number of bags in D. Lines 6-9 can be implemented in time $\mathcal{O}(n)$. The loop in Line 10 runs $\log_2(n)$ times because $\mathrm{lrw}(G) \le \log_2(n)$, and all the steps in Line 10 can be implemented in time $\mathcal{O}(n)$. Also, Lines 17-26 can be done in time $\mathcal{O}(n)$. We conclude that this algorithm runs in time $\mathcal{O}(n^2 \cdot \log_2 n)$.

Finding an Optimal Linear Layout. *We finally establish how to find a linear layout witnessing $\mathrm{lrw}(G)$. We may assume that G has at least 3 vertices. We can assume that for each non-root node v of T and $0 \le i \le \eta$ with $\alpha^v_i \le i$, T^v_i and β^v_i are computed. We inductively obtain optimal linear layouts of $\mathcal{G}[D^v_i]$ using those values. If v is a non-root leaf node of T^v_i, then $\mathcal{G}[D^v_i]$ is either a complete graph or a star for all i, and thus, any ordering of $V(\mathcal{G}[D^v_i])$ is a linear layout of width 1. We may assume that v is a not a leaf node.*

We will search for the path depicted in Lemma 4.5 to apply the same technique used in the proof of Theorem 4.1. What we have shown in Theorem 4.1 is that for a canonical decomposition D of a distance-hereditary graph with its decomposition tree T_D, if T_D has a path $P := v_0 v_1 \cdots v_n v_{n+1}$ such that

- *for each node v in P and a component T of $D \backslash V(\mathsf{b}_D(v))$ not containing a bag $\mathsf{b}_D(w)$ with $w \in P$, $f(\mathsf{b}_D(v), T) \le k-1$,*

then we can generate a linear layout of $\mathcal{G}[D]$ having width at most k. But it assumed that we have linear layouts of graphs corresponding to pending limbs . So, for our purpose, it is necessary to find such a path with $k = \beta^v_i$ such that

- *for each node v in P and a component T of $D \backslash V(\mathsf{b}_D(v))$ not containing a bag $\mathsf{b}_D(w)$ with $w \in P$, a linear layout of $\mathcal{LG}_D[\mathsf{b}_D(v), T]$ with an optimal width is already computed.*

Let us assume that $k = \beta^v_i$. There are two cases; either T^v_i has the k-critical node or not.

Case 1. T^v_i has no k-critical node.

In this case, we take a path P from the root node of T^v_i (or both end nodes of the root edge) to a node w where $\beta^w_i = k$ but for every descendant w' of w, $\beta^{w'}_i < k$. Since T^v_i has no k-critical node, every node outside of this path has β value less than k. Thus, the graphs corresponding to limbs pending to this path have linear rank-width *at most $k-1$, and moreover, by induction hypothesis, we already obtained an optimal linear layout for each graph. This path can be computed in linear time.*

Case 2. T_i^v has a k-critical node.

First note that T_i^v cannot have two k-critical nodes, otherwise, $\beta_i^v = k+1$, which contradicts to our assumption. Let x be the unique k-critical node of T_i^v, and let x_1, x_2 be two children of x where $\beta_i^{x_j} = k$ for each $j \in \{1, 2\}$. For each $j \in \{1, 2\}$, we choose a descendant w_j of x_j where $\beta_i^{w_j} = k$ but for every descendant w_j' of w_j, $\beta_i^{w_j} < k$. Let P be the path from w_1 to w_2 in T_i^v. This path can be computed in linear time.

Since x is the unique k-critical node of T_i^v, every node below of this path has β value less than k, and the graphs corresponding to subtrees pending to the path are computed in advance. Moreover, since this case is exactly when $\alpha_i^v = k$ and $\beta_i^v = k$ and T_i^v has a unique critical node, the canonical decomposition corresponding to the subtree of $T_i^v \backslash x$ containing the parent of x is exactly D_{k-1}^v, and $\mathcal{G}[D_{k-1}^v]$ should have linear rank-width at most $k-1$ as $\beta_i^v = k$. By the induction hypothesis, the optimal linear layout of $\mathcal{G}[D_{k-1}^v]$ is also computed before, as required.

We conclude that we can compute an optimal layout of G in time $\mathcal{O}(n^2 \cdot \log_2 n)$. □

7 Path-Width of Matroids with Branch-Width 2

As a corollary of Theorem 6.1, we can compute the path-width of matroids of branch-width at most 2. We first recall the necessary materials about matroids. We refer to the book written by Oxley [29] for our matroid notations.

Matroids. A pair $(E(M), \mathcal{I}(M))$ is called a *matroid M* if $E(M)$, called the *ground set of M*, is a finite set and $\mathcal{I}(M)$, called the set of *independent sets of M*, is a nonempty collection of subsets of $E(M)$ satisfying the following conditions:

(I1) if $I \in \mathcal{I}(M)$ and $J \subseteq I$, then $J \in \mathcal{I}(M)$,
(I2) if $I, J \in \mathcal{I}(M)$ and $|I| < |J|$, then $I \cup \{z\} \in \mathcal{I}(M)$ for some $z \in J \backslash I$.

A maximal independent set in M is called a *base of M*. It is known that, if B_1 and B_2 are bases of M, then $|B_1| = |B_2|$.

For a matroid M and a subset X of $E(M)$, we let $(X, \{I \subseteq X : I \in \mathcal{I}(M)\})$ be the matroid denoted by $M_{|X}$. The size of a base of $M_{|X}$ is called the *rank* of X in M and the *rank function* of M is the function $r_M : 2^{E(M)} \to \mathbb{N}$ that maps every $X \subseteq E(M)$ to its rank. The rank of $E(M)$ is called the rank of M.

If M is a matroid, then we define λ_M, called the *connectivity function of M*, such that for every subset X of $E(M)$,

$$\lambda_M(X) = r_M(X) + r_M(E(M) \backslash X) - r_M(E(M)) + 1.$$

It is known that the function λ_M is symmetric and submodular.

Let A be a binary matrix and let E be the column labels of A. Let $\mathcal{I}$ be the collection of all those subsets I of E such that the columns of A with index in I are linearly independent. Then $(E, \mathcal{I})$ is a matroid. We denote it by $M(A)$. Every matroid isomorphic to $M(A)$ for some matrix A is called a *binary matroid* and A is called a *representation of M over the binary field*.

We now define fundamental graphs of binary matroids. Let G be a bipartite graph with a bipartition (A, B). We define $M(G, A, B)$ as the binary matroid represented by the $(A \times V)$-matrix $(I_A \quad A_G[A, B])$ where I_A is the $(A \times A)$ identity matrix; and we call G a *fundamental graph* of $M(G, A, B)$. We remark that $|E(M)| = |V(G)|$.

Branch-Width and Path-Width of Matroids. A *branch-decomposition* of a matroid M is a pair (T, L), where T is a subcubic tree and L is a bijection from the elements of $E(M)$ to the leaves of T. For an edge e in T, $T \setminus e$ induces a partition (X_e, Y_e) of the leaves of T. The *width* of an edge e is defined as $\lambda_M(L^{-1}(X_e))$. The *width* of a branch-decomposition (T, L) is the maximum width over all edges of T. The *branch-width* of M, denoted by $\mathrm{bw}(M)$, is the minimum width over all branch-decompositions of M. If $|E(M)| \leq 1$, then M admits no branch-decomposition and $\mathrm{bw}(M) = 0$.

A sequence $e_1, \ldots, e_n$ of the ground set $E(M)$ is called a *linear layout* of M. The *width* of a linear layout $e_1, \ldots, e_n$ of M is

$$\max_{1 \leq i \leq n-1} \{\lambda_M(\{e_1, \ldots, e_i\})\}.$$

The *path-width* of M, denoted by $\mathrm{pw}(M)$, is defined as the minimum width over all linear layouts of M.

The following relation is established by Oum [27].

Proposition 7.1 (Oum *[27]*). *Let G be a bipartite graph with a bipartition (A, B) and let $M := M(G, A, B)$. For every $X \subseteq V(G)$, $\mathrm{cutrk}_G(X) = \lambda_M(X) - 1$. Thus, $\mathrm{rw}(G) = \mathrm{bw}(M) - 1$ and $\mathrm{lrw}(G) = \mathrm{pw}(M) - 1$.*

Here, we observe that every matroid of branch-width at most 2 is binary. This can be observed from the known minor characterizations for binary matroids and matroids of branch-width at most 2. For the definition of matroid minors, we refer to [29].

Theorem 7.2 (Tutte *[30,31]*) *A matroid is binary if and only if it has no minor isomorphic to $U_{2,4}$.*

Theorem 7.3 (Dharmatilake *[10]*). *A matroid has branch-width at most 2 if and only if it has no minor isomorphic to $U_{2,4}$ and $M(K_4)$.*

Corollary 7.4 *The path-width of every n-element matroid of branch-width at most 2 can be computed in time $\mathcal{O}(n^2 \cdot \log_2 n)$, provided that the matroid is given by its binary representation. Moreover, a linear layout of the matroid witnessing the path-width can be computed with the same time complexity.*

Proof Let M be a matroid of branch-width at most 2 and assume that a binary representation A of M is given. We first run a greedy algorithm to find a base B of M [29, Sect. 1.8] in time $\mathcal{O}(|E(M)|^2)$. After choosing one base B, for each $e \in B$ and $e' \in E(M) \setminus B$, we test whether $(B \setminus \{e\}) \cup \{e'\}$ is again a base using the binary representation, which can be done in time $\mathcal{O}(|E(M)|)$ if we first pre-compute the sums of vectors in $B \setminus \{e\}$ for each $e \in B$. The fundamental graph G with respect to M is then the bipartite graph with bipartition $(B, E(M) \setminus B)$ and ee' is an edge if $(B \setminus \{e\}) \cup \{e'\}$ is a base [29]. From what precedes G can be constructed in time

$\mathcal{O}(|E(M)|^2)$. Since M has branch-width at most 2, by Proposition 7.1, the rank-width of G is at most 1. Using Theorem 6.1, we can compute the linear rank-width of G in time $\mathcal{O}(|E(M)|^2 \cdot \log_2 |E(M)|)$, which is the same as $\text{pw}(M) - 1$. Moreover, we can compute a linear layout witnessing $\text{lrw}(G)$ in the same time, that corresponds to the linear layout of M witnessing $\text{pw}(M)$. □

8 An Upper Bound on Linear Rank-Width

As we promised, we prove the following lemma here. We remark that Bodlaender, Gilbert, Hafsteinsson, and Kloks [3] proved a similar relation between tree-width and path-width.

Lemma 8.1 *Let k be a positive integer and let G be a graph of rank-width k. Then* $\text{lrw}(G) \le k\lfloor \log_2 |V(G)| \rfloor$.

Proof Since k is a positive integer, we have $|V(G)| \ge 2$. Let (T, L) be a rank-decomposition of G having width k. For convenience, we choose an edge e of T and subdivide it with introducing a new vertex x, and regard x as the root of T. For each internal vertex t of T with two subtrees T_1 and T_2 of $T \backslash t$ not containing x, let $\ell(t) := T_1$ and $r(t) := T_2$ if the number of leaves of T in T_1 is at least the number of leaves of T in T_2. Let S be a linear layout of G satisfying that

- for each $v_1, v_2 \in V(G)$ with the first common ancestor w of v_1 and v_2 in T, $v_1 <_S v_2$ if $L(v_1) \in V(\ell(w))$.

We can construct such a linear layout inductively.

We show that S has width at most $k\lfloor \log_2 |V(G)| \rfloor$. Let w be a vertex of G that is not the first vertex of S and let $S_w := \{v : v <_S w\}$. Let P_w be the path from $L(w)$ to the root x in T. Note that for each $t \in V(P_w) \backslash \{L(w)\}$ and the subtree T' of $T \backslash t$ not containing x and $L(w)$,

- if $r(t) = T'$, then all leaves of T in T' are not contained in S_w, and
- if $\ell(t) = T'$, then all leaves of T in T' are contained in S_w.

Let Q be the set of all vertices t in P_w such that the subtree $\ell(t)$ does not contain $L(w)$.

Let $q_1, q_2, \ldots, q_m$ be the sequence of all vertices in Q such that for each $1 \le j \le m-1$, q_j is a descendant of q_{j+1} in T. For $1 \le j \le m$, let Q_i be the set of all leaves of T contained in $\ell(q_i)$. Clearly, $S_w = Q_1 \cup Q_2 \cup \cdots \cup Q_m$ and $V(G) \backslash S_w \ne \emptyset$. Therefore, we have

$$\begin{aligned} |V(G)| &= |Q_1| + \cdots + |Q_m| + |V(G) \backslash S_w| \\ &\ge 1 + 2 + 4 + \cdots 2^{m-1} + 1 \\ &= 2^m. \end{aligned}$$

Thus, $m \le \lfloor \log_2 |V(G)| \rfloor$.

Note that for each $1 \leq j \leq m$, $\text{rank}(A_G[Q_i, V(G) \backslash S_w]) \leq k$. Therefore, we have that

$$\text{cutrk}_G(S_w) = \text{rank}(A_G[(Q_1 \cup \cdots \cup Q_m, V(G) \backslash S_w)]) \leq km \leq k\lfloor \log_2 |V(G)| \rfloor.$$

Since w was arbitrarily chosen, it implies that $\text{lrw}(G) \leq k\lfloor \log_2 |V(G)| \rfloor$. □

9 Concluding Remarks

We have provided a characterization of the linear rank-width of distance-hereditary graphs in terms of their canonical decompositions, and use this characterization to derive a polynomial-time algorithm to compute their linear rank-width. An easy consequence of this is also a polynomial-time algorithm for computing the path-width of matroids of branch-width at most 2, which was not addressed in the past.

In the second part of this work [1], we will discuss structural properties of distance-hereditary graphs related to linear rank-width. Note that Jeong et al. [20] provided a lower bound on the size of the vertex-minor obstruction set for graphs with bounded linear rank-width, by providing a set of pairwise locally non-equivalent vertex-minor obstructions for graphs of linear rank-width at most k for each k. Their graphs are indeed distance-hereditary graphs, and we will give a more general way to generate all distance-hereditary vertex-minor obstructions using the characterization given in this paper. Also, we prove that for a fixed tree T, every distance-hereditary graph of sufficiently large linear rank-width contains T as a vertex-minor.

Acknowledgements The authors would like to thank Sang-il Oum for pointing out that the computation of the path-width of matroids of branch-width at most 2 can be obtained as a corollary of our main result.

References

1. Adler, I., Kanté, Mamadou Moustapha, O-joung, K.: Linear rank-width of distance-hereditary graphs II. Vertex-minor obstructions. *Submitted*, (2015). http://arxiv.org/abs/1508.04718
2. Bandelt, H.J., Mulder, H.M.: Distance-hereditary graphs. J. Comb. Theory Ser. B **41**(2), 182–208 (1986)
3. Bodlaender, H.L., Gilbert, J.R., Hafsteinsson, H., Kloks, T.: Approximating treewidth, pathwidth, and minimum elimination tree height. J. Algorithms **18**, 238–255 (1991)
4. Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. J. Algorithms **21**(2), 358–402 (1996)
5. Bouchet, A.: Transforming trees by successive local complementations. J. Graph Theory **12**(2), 195–207 (1988)
6. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discrete Appl. Math. **101**(1–3), 77–114 (2000)
7. Courcelle, Bo, Oum, S.: Vertex-minors, monadic second-order logic, and a conjecture by Seese. J. Comb. Theory, Ser. B **97**(1), 91–126 (2007)
8. Cunnigham, W.H., Edmonds, J.: A combinatorial decomposition theory. Can. J. Math. **32**, 734–765 (1980)
9. Dahlhaus, E.: Parallel algorithms for hierarchical clustering, and applications to split decomposition and parity graph recognition. J. Graph Algorithms **36**(2), 205–240 (2000)
10. Dharmatilake, J.S.: A min-max theorem using matroid separations. In: Matroid theory (Seattle, WA, 1995), volume 197 of Contemp. Math., pp. 333–342. Amer. Math. Soc., Providence, RI, (1996)

11. Diestel, R.: Graph theory, volume 173 of Graduate Texts in Mathematics. Springer, Heidelberg, fourth edition, (2010)
12. Ganian, R.: Thread graphs, linear rank-width and their algorithmic applications. In: Iliopoulos, Costas S., Smyth, William F. (eds.), IWOCA, volume 6460 of Lecture Notes in Computer Science, pp. 38–42. Springer, (2010)
13. Gavoille, C., Paul, C.: Distance labeling scheme and split decomposition. Discrete Math. **273**(1–3), 115–130 (2003)
14. Geelen, J.F., Gerards, A.M.H., Whittle, G.: Branch-width and well-quasi-ordering in matroids and graphs. J. Combin. Theory Ser. B **84**(2), 270–290 (2002)
15. Geelen, J., Gerards, B., Whittle, G.: Excluding a planar graph from GF(q) representable matroids. J. Combin. Theory Ser. B **97**(6), 971–998 (2007)
16. Geelen, J., Gerards, B., Whittle, G.: Solving Rota's conjecture. Notices Amer. Math. Soc. **61**(7), 736–743 (2014)
17. Geelen, J.F., Oum, S.: Circle graph obstructions under pivoting. J. Graph Theory **61**(1), 1–11 (2009)
18. Isolde Adler and Mamadou Moustapha Kanté: Linear rank-width and linear clique-width of trees. Theoret. Comput. Sci. **589**, 87–98 (2015)
19. Jeong, J., Kim, E.J., Oum, S.: Constructive algorithm for path-width of matroids. SODA: 1695–1704 (2016)
20. Jeong, J., Kwon, O., Oum, S.: Excluded vertex-minors for graphs of linear rank-width at most k. European J. Combin. **41**, 242–257 (2014)
21. Jonathan, A., Ellis, J.A., Sudborough, I.H., Turner, J.S.: The vertex separation and search number of a graph. Inf. Comput. **113**(1), 50–79 (1994)
22. Kashyap, N.: Matroid pathwidth and code trellis complexity. SIAM J. Discrete Math. **22**(1), 256–272 (2008)
23. Kloks,T., Bodlaender, H.L., Müller, H., Kratsch, D.: Computing treewidth and minimum fill-in: All you need are the minimal separators. In: Lengauer, Thomas (ed.), ESA, volume 726 of Lecture Notes in Computer Science, pp. 260–271. Springer, (1993)
24. Mamadou Moustapha Kanté and Michael Rao: The rank-width of edge-coloured graphs. Theory Comput. Syst. **52**(4), 599–644 (2013)
25. Mamadou Moustapha Kanté: Well-quasi-ordering of matrices under Schur complement and applications to directed graphs. European J. Combin. **33**(8), 1820–1841 (2012)
26. Megiddo, N., Hakimi, S.L., Garey, M.R., Johnson, D.S., Papadimitriou, C.H.: The complexity of searching a graph. J. ACM **35**(1), 18–44 (1988)
27. Oum, S.: Rank-width and vertex-minors. J. Comb. Theory, Ser. B **95**(1), 79–100 (2005)
28. Oum, S., Seymour, P.: Approximating clique-width and branch-width. J. Comb. Theory, Ser. B **96**(4), 514–528 (2006)
29. Oxley, J.: Matroid Theory. Number 21 in Oxford Graduate Texts in Mathematics. Oxford University Press, second edition, (2012)
30. Tutte, W.T.: A homotopy theorem for matroids. I, II. Trans. Amer. Math. Soc. **88**, 144–174 (1958)
31. Tutte, W.T.: Lectures on matroids. J. Res. Nat. Bur. Standards Sect. B **69B**, 1–47 (1965)