Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Discrete Mathematics 306 (2006) 1637 – 1650
DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Linear layouts measuring neighbourhoods in graphs

Frank Gurski

Heinrich-Heine-Universität Düsseldorf, Institute of Computer Science, D-40225 Düsseldorf, Germany

Received 22 November 2005; received in revised form 2 March 2006; accepted 28 March 2006

Abstract

In this paper we introduce the graph layout parameter neighbourhood-width as a variation of the well-known cut-width. The cut-width of a graph $G = (V, E)$ is the smallest integer k, such that there is a linear layout $\varphi : V \to \{1, \ldots, |V|\}$, such that for every $1 \leqslant i < |V|$ there are at most k edges $\{u, v\}$ with $\varphi(u) \leqslant i$ and $\varphi(v) > i$. The neighbourhood-width of a graph is the smallest integer k, such that there is a linear layout φ, such that for every $1 \leqslant i < |V|$ the vertices u with $\varphi(u) \leqslant i$ can be divided into at most k subsets each members having the same neighbourhood with respect to the vertices v with $\varphi(v) > i$.

We show that the neighbourhood-width of a graph differs from its linear clique-width or linear NLC-width at most by one. This relation is used to show that the minimization problem for neighbourhood-width is NP-complete.

Furthermore, we prove that simple modifications of neighbourhood-width imply equivalent layout characterizations for linear clique-width and linear NLC-width.

We also show that every graph of path-width k or cut-width k has neighbourhood-width at most $k+2$ and we give several conditions such that graphs of bounded neighbourhood-width have bounded path-width or bounded cut-width.

Keywords: Graph layout; Neighbourhoods in graphs; Cut-width; Linear clique-width; Linear NLC-width

1. Introduction

A *linear layout* (a *layout*, or an *arrangement*) of an undirected graph $G = (V, E)$ is a bijective function $\varphi : V \to \{1, \ldots, |V|\}$. A graph layout problem on a graph G seeks for a layout for G such that a certain function on the graph is optimized. For a survey on graph layout problems see e.g. [12,25]. We will use the following notations for graph layout problems given in [12].

For a graph G, we denote by $\Phi(G)$ the set of all layouts for G. Given a layout $\varphi \in \Phi(G)$ we define for $1 \leqslant i \leqslant |V|$ the vertex sets

$$L(i, \varphi, G) = \{u \in V \mid \varphi(u) \leqslant i\}$$

and

$$R(i, \varphi, G) = \{u \in V \mid \varphi(u) > i\}.$$

The *reverse layout* φ^R, for $\varphi \in \Phi(G)$, is defined by $\varphi^R(u) = |V| - \varphi(u) + 1$, $u \in V$.

E-mail address: gurski-dm@acs.uni-duesseldorf.de.

0012-365X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2006.03.048

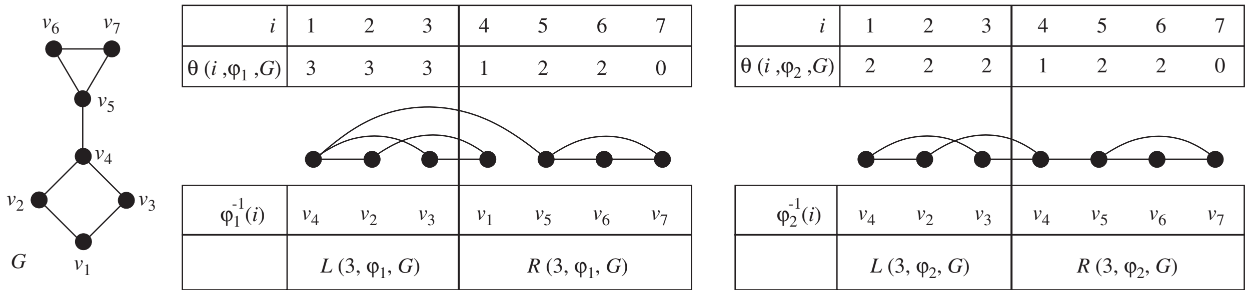

Fig. 1. The figure shows a graph G, a layout φ_1 for G with $\theta(3, \varphi_1, G)$=cut-width$(\varphi_1, G) = 3$, and a layout φ_2 for G with $\theta(3, \varphi_2, G)$= cut-width(φ_2, G)= 2.

A *layout cost function* is a function that defines for a graph G and a layout $\varphi \in \Phi(G)$ an integer $F(\varphi, G)$. For a layout cost function we define the corresponding layout problem F by determining a layout $\varphi^* \in \Phi(G)$, such that $F(\varphi^*, G) = F(G)$ where

$$F(\varphi^*, G) = \min_{\varphi \in \Phi(G)} F(\varphi, G).$$

Next, we illustrate these notations by the well-known graph layout parameter cut-width.

The *edge cut* for a graph G, $\varphi \in \Phi(G)$, $1 \leqslant i \leqslant |V|$, is defined as

$$\theta(i, \varphi, G) = |\{\{u, v\} \mid u \in L(i, \varphi, G), v \in R(i, \varphi, G)\}|.$$

In Fig. 1, we show two layouts φ_j, $j = 1, 2$, of a graph G, by aligning vertex v at position $\varphi_j(v)$ on a horizontal line. Each vertical line between two consecutive vertices $\varphi_j^{-1}(i)$ and $\varphi_j^{-1}(i + 1)$ separates the vertex set of G into $L(i, \varphi_j, G)$ and $R(i, \varphi_j, G)$. $\theta(i, \varphi_j, G)$ is the number of edges crossing the vertical line between vertex $\varphi_j^{-1}(i)$ and vertex $\varphi_j^{-1}(i + 1)$.

The layout cost function for cut-width is defined by

$$\text{cut-width}(\varphi, G) = \max_{1 \leqslant i \leqslant |V|} \theta(i, \varphi, G).$$

In Fig. 1, cut-width(φ_1, G)=3 and cut-width(φ_2, G)=2 holds true, this can easily be verified by counting the maximum number of edges crossing a vertical line between two consecutive vertices $\varphi_j^{-1}(i)$ and $\varphi_j^{-1}(i+1)$ in layout φ_j, $j=1, 2$.

The *cut-width problem* seeks for a given graph G a linear layout $\varphi^* \in \Phi(G)$, such that

$$\text{cut-width}(\varphi^*, G) = \min_{\varphi \in \Phi(G)} \text{cut-width}(\varphi, G),$$

the *cut-width* of graph G is defined by cut-width(G) = cut-width(φ^*, G).

In this paper we introduce the neighbourhood-width which leads, in comparison to cut-width, a more powerful complexity measure. Graph $G = (V, E)$ has neighbourhood-width at most k, if there is a linear layout $\varphi \in \Phi(G)$, such that for every $1 \leqslant i < |V|$ the vertices in $L(i, \varphi, G)$ can be divided into at most k subsets $L_1, \ldots, L_k$, such that the vertices of set L_j, $1 \leqslant j \leqslant k$, have the same neighbourhood with respect to the vertices in $R(i, \varphi, G)$.

One motivation for defining neighbourhood-width is to characterize graphs of bounded clique-width and graphs of bounded NLC-width. The clique-width and NLC-width of a graph G is defined as the minimum number of labels needed to define G by expressions consisting of single labelled vertices, union, edge insertion, and relabelling operations [11,26]. Clique-width and NLC-width bounded graphs are particularly interesting from an algorithmic point of view. A lot of NP-complete graph problems can be solved in linear time for graphs of bounded clique-width if a corresponding decomposition for the graph is given as an input [9]. For example, all graph properties which are expressible in monadic second-order logic with quantifications over vertices and vertex sets (MSO_1-logic) are decidable in linear time on clique-width bounded graphs if a corresponding decomposition for the graph is given as an input [9]. Recently, Oum and Seymour have shown that such a decomposition can be found in polynomial time [23]. In this paper we consider graphs defined by linear clique-width and linear NLC-width expressions, i.e. in every union operation one of the two involved graphs consists of a single labelled vertex. Restricted versions of clique-width and NLC-width are sometimes very useful. This shows for example the proof of the NP-completeness of minimizing clique-width in [14,15].

This paper is organized as follows. In Section 2, we recall the definition of linear NLC-width, linear clique-width, and path-width. In Section 3, we introduce the neighbourhood-width of a graph and we show that every graph of neighbourhood-width k has linear NLC-width and linear clique-width k or $k+1$. The class of graphs of neighbourhood-width 1 is characterized as the set of threshold graphs. In Sections 4 and 5, we modify the layout parameter neighbourhood-width to show equivalent layout characterizations for linear NLC-width and linear clique-width, independently from vertex labelled graphs. In Section 6, we give upper bounds for the linear NLC-width, linear clique-width, and neighbourhood-width of graphs of bounded path-width and graphs of bounded cut-width. Further we show that under several conditions graphs of bounded neighbourhood-width even have bounded cut-width. In Section 7 these bounds are used to show that minimizing the neighbourhood-width of a given graph is an NP-complete problem, but for graphs of bounded tree-width the neighbourhood-width can be approximated with constant difference guarantee.

2. Preliminaries

Let $\mathscr{G}$ be the set of all *graphs* $G=(V_G, E_G)$, where V_G is a finite set of *vertices* and $E_G \subseteq \{\{u, v\} \mid u, v \in V_G,\ u \neq v\}$ is a finite set of *edges*. Let $[k] := \{1, \ldots, k\}$ be the set of all integers between 1 and k. We also work with *labelled graphs* $G=(V_G, E_G, \text{lab}_G)$, where $\text{lab}_G : V_G \to [k]$ is a mapping. A labelled graph $J=(V_J, E_J, \text{lab}_J)$ is a *subgraph* of G if $V_J \subseteq V_G$, $E_J \subseteq E_G$, and $\text{lab}_J(u) = \text{lab}_G(u)$ for all $u \in V_J$. J is an *induced subgraph* of G if additionally $E_J = \{\{u, v\} \in E_G \mid u, v \in V_J\}$. For $U \subseteq V_G$, we define by $G[U]$ the subgraph of G induced by the vertices of U. The labelled graph consisting of a single vertex labelled by $a \in [k]$ is denoted by $\bullet_a$. For the definition of special graph classes we refer to the survey of Brandstädt et al. [6].

Next, we recall the definitions of linear NLC-width, linear clique-width and path-width.

Definition 1 (*linear NLC-width, [20]*). Let k be a positive integer. The class lin-NLC_k of labelled graphs is recursively defined as follows:

(1) The single vertex graph $\bullet_a$ for $a \in [k]$ is in lin-NLC_k.
(2) Let $G=(V_G, E_G, \text{lab}_G) \in$ lin-NLC_k be a vertex labelled graph, $v \notin V_G$ be a single vertex labelled by $a \in [k]$, and $S \subseteq [k]^2$ be a relation, then $G \times_S \bullet_a$ defined by $V' := V_G \cup \{v\}$, $E' := E_G \cup \{\{u, v\} \mid u \in V_G, (\text{lab}_G(u), a) \in S\}$, and

$$\text{lab}'(u) := \begin{cases} \text{lab}_G(u) & \text{if } u \in V_G \\ a & \text{if } u = v \end{cases} \quad \forall u \in V'$$

is in lin-NLC_k.
(3) Let $G=(V_G, E_G, \text{lab}_G) \in$ lin-NLC_k be a labelled graph and $R : [k] \to [k]$ be a function, then $\circ_R(G) := (V_G, E_G, \text{lab}')$ defined by $\text{lab}'(u) := R(\text{lab}_G(u))$ is in lin-NLC_k.

The *linear NLC-width* of a labelled graph G (linear NLC-width(G)) is the least integer k such that $G \in$ lin-NLC_k.

Graphs of linear NLC-width 1 are exactly $(C_4, P_4, 2K_2)$-free graphs and thus exactly threshold graphs [17]. For every fixed $k \geqslant 2$ the recognition problem for graphs of linear NLC-width at most k is still open. If k is given to the input the recognition problem is NP-complete, see Section 7.

Definition 2 (*linear clique-width, [20]*). Let k be a positive integer. The class lin-CW_k of labelled graphs is recursively defined as follows:

(1) The single vertex graph $\bullet_a$ for $a \in [k]$ is in lin-CW_k.

(2) Let $G=(V_G, E_G, \mathrm{lab}_G) \in \text{lin-CW}_k$ be a vertex labelled graph and $v \notin V_G$ be a single vertex labelled by $a \in [k]$, then $G \oplus \bullet_a := (V', E_G, \mathrm{lab}')$ defined by $V' := V_G \cup \{v\}$ and

$$\mathrm{lab}'(u) := \begin{cases} \mathrm{lab}_G(u) & \text{if } u \in V_G \\ a & \text{if } u = v \end{cases} \quad \forall u \in V'$$

is in lin-CW_k.

(3) Let $a, b \in [k]$ be two distinct integers and $G=(V_G, E_G, \mathrm{lab}_G) \in \text{lin-CW}_k$ be a labelled graph, then
 (a) $\rho_{a\to b}(G) := (V_G, E_G, \mathrm{lab}')$ defined by

$$\mathrm{lab}'(u) := \begin{cases} \mathrm{lab}_G(u) & \text{if } \mathrm{lab}_G(u) \neq a \\ b & \text{if } \mathrm{lab}_G(u) = a \end{cases} \quad \forall u \in V_G$$

 is in lin-CW_k and
 (b) $\eta_{a,b}(G) := (V_G, E', \mathrm{lab}_G)$ defined by $E' := E_G \cup \{\{u, v\} \mid u, v \in V_G,\ u \neq v,\ \mathrm{lab}(u) = a,\ \mathrm{lab}(v) = b\}$ is in lin-CW_k.

The *linear clique-width* of a labelled graph G (linear clique-width(G)) is the least integer k such that $G \in \text{lin-CW}_k$.

Graphs of linear clique-width at most 2 are exactly (co-$2P_3$, P_4, $2K_2$)-free graphs and can thus be recognized in polynomial time [17]. For every fixed $k \geqslant 3$ the recognition problem for graphs of linear clique-width at most k is still open. If k is given to the input the recognition problem is NP-complete, see [14].

An expression X built with the operations $\bullet_a$, $\times_S$, $\circ_R$ for $a \in [k]$, $S \subseteq [k]^2$, and $R : [k] \to [k]$ as defined above is called a *linear NLC-width k-expression*. An expression X built with the operations $\bullet_a$, $\oplus$, $\rho_{a\to b}$, $\eta_{a,b}$ for integers $a, b \in [k]$ as defined above is called a *linear clique-width k-expression*. Note that every expression defines a layout by the order in which the vertices are inserted in the corresponding graph. Every such expression has by its recursive definition a tree structure which we call the *linear NLC-width expression tree* or *linear clique-width expression tree*, respectively.

The *linear NLC-width* (*linear clique-width*) of an unlabelled graph $G=(V, E)$ is the smallest integer k, such that there is a mapping $\mathrm{lab} : V \to [k]$ such that the labelled graph (V, E, lab) has linear NLC-width (linear clique-width) at most k. The graph defined by expression X is denoted by $\mathrm{val}(X)$.

There is a very close relation between the linear NLC-width and linear clique-width of a graph.

Lemma 3 (*Gurski and Wanke [20]*). *Let G be a graph of linear NLC-width k, then G has linear clique-width k or $k+1$.*

For example every *path* $P_n = (\{v_1, \ldots, v_n\}, \{\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}\})$ has linear NLC-width at most 3 and linear clique-width at most 3. This can easily be shown by the following 3-expressions X_{P_n} and Y_{P_n}:

$$X_{P_3} = (\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\{(2,3)\}} \bullet_3,$$

$$X_{P_n} = \circ_{\{(1,1),(2,1),(3,2)\}}(X_{P_{n-1}}) \times_{\{(2,3)\}} \bullet_3, \quad n \geqslant 4,$$

$$Y_{P_3} = \eta_{2,3}(\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_3),$$

$$Y_{P_n} = \eta_{2,3}(\rho_{3\to 2}(\rho_{2\to 1}(Y_{P_{n-1}})) \oplus \bullet_3), \quad n \geqslant 4.$$

Further results on graph classes of bounded linear NLC-width can be found in [20].

Last we want to recall the definition of path-width.

Definition 4 (*path-width, [24]*). A *path decomposition* of a graph $G=(V_G, E_G)$ is a pair $(\mathcal{X}, T)$ where $T=(V_T, E_T)$ is a path and $\mathcal{X} = \{X_u \mid u \in V_T\}$ is a family of subsets $X_u \subseteq V_G$, one for each node u of T, such that

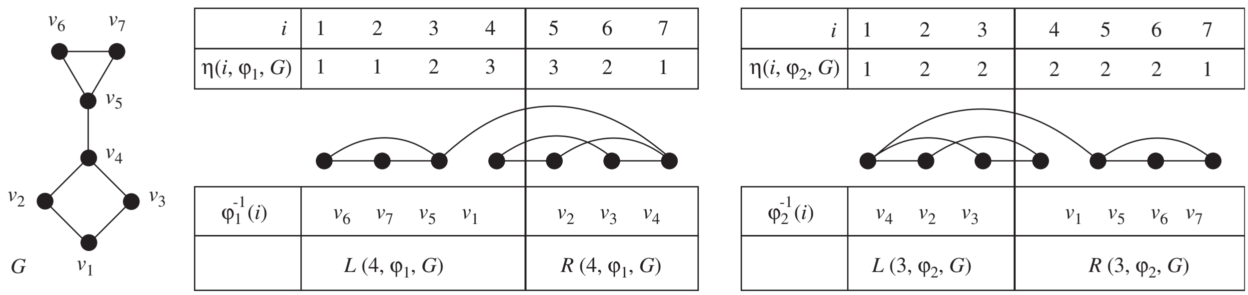

Fig. 2. The figure shows a graph G, a layout φ_1 for G with $\eta(4, \varphi_1, G) = \mathrm{nw}(\varphi_1, G) = 3$, and a layout φ_2 for G with $\eta(3, \varphi_2, G) = \mathrm{nw}(\varphi_2, G) = 2$.

(1) $\bigcup_{u \in V_T} X_u = V_G$,
(2) for every edge $\{v_1, v_2\} \in E_G$, there is a node $u \in V_T$ such that $v_1 \in X_u$ and $v_2 \in X_u$, and
(3) for every vertex $v \in V_G$ the subgraph of T induced by the nodes $u \in V_T$ with $v \in X_u$ is connected.

The *width* of a path decomposition $(\mathcal{X} = \{X_u \mid u \in V_T\}, T = (V_T, E_T))$ is $\max_{u \in V_T} |X_u| - 1$. The *path-width* of a graph G is the smallest integer k such that there is a path decomposition $(\mathcal{X}, T)$ of width k for G.

The path-width of a graph $G = (V, E)$ equals its *vertex separation number* (*vsn*), which is a well-known graph layout parameter [25,21,22] defined as follows:

$$\mathrm{vsn}(G) = \min_{\varphi \in \Phi(G)} \max_{1 \leqslant i \leqslant |V|} |\{u \in L(i, \varphi, G) \mid \exists v \in R(i, \varphi, G) : \{u, v\} \in E\}|.$$

3. A linear layout measuring neighbourhoods in graphs

In this section we introduce the layout parameter neighbourhood-width as a variation of the well-known cut-width. Let $G = (V, E)$ be a graph and $U, W \subseteq V$ two disjoint vertex sets, by

$$N_W(u) = \{v \in W \mid \{u, v\} \in E\}$$

we denote the neighbourhood of vertex u into set W, i.e. the vertices of W which are adjacent to u. By

$$N(U, W) = \{N_W(u) \mid u \in U\}$$

we denote the set of all neighbourhoods of the vertices of set U into set W.

For a graph G, $\varphi \in \Phi(G)$, $1 \leqslant i \leqslant |V|$, we define the *neighbourhood cut* as the number of distinct neighbourhoods of the vertices of set $L(i, \varphi, G)$ into set $R(i, \varphi, G)$ by

$$\eta(i, \varphi, G) = |N(L(i, \varphi, G), R(i, \varphi, G))|.$$

In Fig. 2, we show two layouts φ_j, $j = 1, 2$, of a graph G, by aligning vertex v at position $\varphi_j(v)$ on a horizontal line. $\eta(i, \varphi_j, G)$ is the number of disjoint subsets $L_{i,j}$ of the vertices left of the vertical line between vertex $\varphi_j^{-1}(i)$ and $\varphi_j^{-1}(i+1)$, such that all vertices in $L_{i,j}$ have the same neighbourhood with respect to the vertices right of the vertical line. For example the vertices of set $L(4, \varphi_1, G) = \{v_1, v_5, v_6, v_7\}$ have the following three neighbourhoods into set $R(4, \varphi_1, G) = \{v_2, v_3, v_4\}$, $N_{R(4,\varphi_1,G)}(v_6) = N_{R(4,\varphi_1,G)}(v_7) = \emptyset$, $N_{R(4,\varphi_1,G)}(v_5) = \{v_4\}$, $N_{R(4,\varphi_1,G)}(v_1) = \{v_2, v_3\}$, thus $N(L(4, \varphi_1, G), R(4, \varphi_1, G)) = \{\emptyset, \{v_4\}, \{v_2, v_3\}\}$ and $\eta(4, \varphi_1, G) = 3$.

Further we define a layout cost function by

$$\mathrm{nw}(\varphi, G) = \text{neighbourhood-width}(\varphi, G) = \max_{1 \leqslant i \leqslant |V|} \eta(i, \varphi, G).$$

In Fig. 2 two layouts φ_1, φ_2 for a graph G are shown, where $\mathrm{nw}(\varphi_1, G) = 3$ and $\mathrm{nw}(\varphi_2, G) = 2$.

The *neighbourhood-width problem* searches for a given graph G a layout $\varphi^* \in \Phi(G)$, such that

$$\text{neighbourhood-width}(\varphi^*, G) = \min_{\varphi \in \Phi(G)} \text{neighbourhood-width}(\varphi, G),$$

the *neighbourhood-width* of graph G is defined by neighbourhood-width(G) = neighbourhood-width(φ^*, G).

Note that for every graph G and every $\varphi \in \Phi(G)$, the value of cut-width(G, φ) does not change if we exchange layout φ by its reverse layout φ^R, while this property is not true for the value of nw(G, φ) in general.

Next, we show a very tight connection between the neighbourhood-width, linear NLC-width, and linear clique-width.

Theorem 5. *Let G be a graph of neighbourhood-width k, then G has linear NLC-width k or $k+1$ and linear clique-width k or $k+1$.*

Proof. Let G be a graph with n vertices of neighbourhood-width k and $\varphi : V \to [n]$ be a layout such that neighbourhood-width$(\varphi, G) \leqslant k$. We next recursively define linear NLC-width $(k+1)$-expressions X_i, $i = 1, \ldots, n$, such that X_i defines subgraph $G[L(i, \varphi, G)]$ and in graph val(X_i) two vertices with the same neighbourhood with respect to the vertices in $R(i, \varphi, G)$ are labelled equal by a label from $[k]$.

(1) For $i = 1$. We define $X_1 = \bullet_1$ representing vertex $\varphi^{-1}(1)$.
(2) For $i > 1$. By induction expression X_{i-1} defines subgraph $G[L(i-1, \varphi, G)]$. X_i can be defined by $X_i = \circ_R(X_{i-1} \times_S \bullet_{k+1})$ as follows. For vertex $\varphi^{-1}(i)$ we define an expression $\bullet_{k+1}$. Therefore, we use label $k+1$ which is not used in graph val(X_{i-1}), since vertex $\varphi^{-1}(i)$ does not necessarily belong to one of the $\leqslant k$ neighbourhoods of the vertices in $L(i-1, \varphi, G)$ into set $R(i, \varphi, G)$. The edges between vertex $\varphi^{-1}(i)$ and vertices of graph val(X_{i-1}) will be inserted by operation $\times_S$, which is possible by our assumption. In the resulting expression we relabel the vertices in $L(i, \varphi, G)$ by function $R : [k+1] \to [k]$ with respect to their neighbourhoods into $R(i, \varphi, G)$. This is possible by our assumption that neighbourhood-width$(\varphi, G) \leqslant k$.

The resulting $(k+1)$-expression X_n defines graph G by construction, which implies that linear NLC-width$(G) \leqslant$ neighbourhood-width$(G) + 1$.

On the other hand, if G has linear NLC-width k, then there exists a k-expression X defining G. Expression X defines a layout $\varphi : V \to [n]$ for the vertices of G by the order the vertices of G are inserted in graph G. By the definition of linear NLC-width, for every $1 \leqslant i < n$, vertices of the same label in graph $G[L(i, \varphi, G)]$ will be treated in the same way by all further operations. Since there are k possible vertex labels, for every $1 \leqslant i < n$, the vertices in $L(i, \varphi, G)$ define at most k neighbourhoods with respect to set $R(i, \varphi, G)$. Thus, φ defines a layout for G such that neighbourhood-width$(\varphi, G) \leqslant k$. This implies that neighbourhood-width$(G) \leqslant$ linear NLC-width(G).

Similar arguments can be used to show the bounds for linear clique-width. □

The proof of Theorem 5 even shows that the set of graphs of neighbourhood-width 1 is equal to the set of graphs of linear NLC-width 1 and by the results shown in [17], we conclude the following characterization.

Corollary 6. *For every graph G the following statements are equivalent*:

(1) *G has neighbourhood-width* 1.
(2) *G contains no C_4, no P_4, and no $2K_2$ as induced subgraph.*
(3) *G is a threshold graph.*
(4) *G has linear NLC-width* 1.

By the expressions given in Section 2 and a very simple observation we conclude that every path P_n, $n \geqslant 4$, has linear clique-width 3, every path P_n, $n \geqslant 7$, has linear NLC-width 3, and every path P_n, $n \geqslant 4$, has neighbourhood-width 2, which implies that the bounds of Theorem 5 cannot be improved.

4. A layout characterization for linear NLC-width

Let $G = (V, E) \in$ lin-NLC$_k$ be a graph, $\varphi \in \Phi(G)$ be a layout defined by a k-expression for G, and $1 \leqslant i < |V|$. We next show that a modification of the neighbourhood cut given in Section 3 leads to an equivalent layout definition for linear NLC-width, independently from vertex labelled graphs. As stated in the proof of Theorem 5, such a layout definition has to consider, besides counting the number of neighbourhoods of the vertices in set $L(i, \varphi, G)$ into set $R(i, \varphi, G)$, whether vertex $\varphi^{-1}(i+1)$, $1 \leqslant i \leqslant |V|-2$, has the same neighbourhood as a vertex of $L(i, \varphi, G)$ into set $R(i+1, \varphi, G)$. Therefore, we define a boolean graph property $\pi_1 : \mathbb{N} \times \Phi(\mathscr{G}) \times \mathscr{G} \to \{0, 1\}$ as follows:

$$\pi_1(i, \varphi, G) = 0 \quad \Leftrightarrow \quad \exists u \in L(i, \varphi, G) : N_{R(i+1,\varphi,G)}(u) = N_{R(i+1,\varphi,G)}(\varphi^{-1}(i+1)).$$

Obviously for $i = |V| - 1$, $\pi_1(i, \varphi, G)$ is equal to 0, which fits to the definition of linear NLC-width, since the last inserted vertex $\varphi^{-1}(|V|)$ in an linear NLC-width expression can always get an arbitrary label.

We define the *modified neighbourhood cut* for a graph G, $\varphi \in \Phi(G)$, $1 \leqslant i < |V|$, by

$$\nu(i, \varphi, G) = |N(L(i, \varphi, G), R(i, \varphi, G))| + \pi_1(i, \varphi, G).$$

We next show that for every graph G a minimal value for $\max_{1 \leqslant i < |V|} \nu(i, \varphi, G)$, $\varphi \in \Phi(G)$, is equal to its linear NLC-width.

Theorem 7. *Let $G = (V, E)$ be a graph with at least two vertices, then*

$$\text{linear NLC-width}(G) = \min_{\varphi \in \Phi(G)} \max_{1 \leqslant i < |V|} \nu(i, \varphi, G).$$

Proof. Let G be a graph with n vertices and $\varphi : V \to [n]$ be a layout for graph G such that $\max_{1 \leqslant i < |V|} \nu(i, \varphi, G) = k$. We next define linear NLC-width k-expressions X_i, $i = 1, \ldots, n$, such that X_i defines subgraph $G[L(i, \varphi, G)]$ and in $\text{val}(X_i)$ two vertices with the same neighbourhood into set $R(i, \varphi, G)$ have the same label.

(1) For $i = 1$. We define $X_1 = \bullet_1$ representing vertex $\varphi^{-1}(1)$.
(2) For $i > 1$. By induction there is a linear NLC-width k-expression X_{i-1} that defines graph $G[L(i-1, \varphi, G)]$. We now insert vertex $\varphi^{-1}(i)$ into graph $\text{val}(X_{i-1})$. In order to choose the initial label of vertex $\varphi^{-1}(i)$ we distinguish between the following two cases:
   (a) If there is a vertex u in set $L(i-1, \varphi, G)$ with the same neighbourhood as $\varphi^{-1}(i)$ into set $R(i, \varphi, G)$, then we will label vertex $\varphi^{-1}(i)$ by the label $l \in [k]$ of vertex u in graph $\text{val}(X_{i-1})$.
   (b) Otherwise, $\pi_1(i-1, \varphi, G) = 1$ and thus $|N(L(i-1, \varphi, G), R(i-1, \varphi, G))| \leqslant k-1$ holds true. This implies that graph $\text{val}(X_{i-1})$ uses at most $k-1$ labels. We can choose one free label $l \in [k]$ which will be given to vertex $\varphi^{-1}(i)$.

   Then we can define X_i by $X_i = \circ_R(X_{i-1} \times_S \bullet_l)$. The edges between vertex $\varphi^{-1}(i)$ and vertices of graph $\text{val}(X_{i-1})$ will be inserted by operation $\times_S$, which is possible by our assumption. In the resulting expression we relabel the vertices in $L(i, \varphi, G)$ by function $R : [k] \to [k]$ with respect to their neighbourhoods into $R(i, \varphi, G)$, i.e. two vertices of $L(i, \varphi, G)$ are labelled equal in graph $G[L(i, \varphi, G)]$ if and only if they have the same neighbourhood with respect to $R(i, \varphi, G)$.

Graph $\text{val}(X_n)$ defines graph G by construction. Thus, for every layout $\varphi \in \Phi(G)$, linear NLC-width$(G) \leqslant \max_{1 \leqslant i < |V|} \nu(i, \varphi, G)$.

On the other hand, if G has linear NLC-width k, then there is a k-expression X defining G. Expression X defines a layout $\varphi_1 : V \to [n]$ for the vertices of G by the order the vertices of G are inserted in the graph. Since there are only k possible vertex labels, for every $1 \leqslant i < n$, the vertices in $L(i, \varphi_1, G)$ define at most k neighbourhoods with respect to vertex set $R(i, \varphi_1, G)$, and thus we know that $|N(L(i, \varphi_1, G), R(i, \varphi_1, G))| \leqslant k$. Next, we consider two possible cases for vertex $\varphi_1^{-1}(i+1)$.

(1) If vertex $\varphi_1^{-1}(i+1)$ has the same neighbourhood as at least one vertex of $L(i, \varphi_1, G)$ with respect to set $R(i+1, \varphi_1, G)$, then $\pi_1(i, \varphi_1, G) = 0$.
(2) Otherwise, vertex $\varphi_1^{-1}(i+1)$ has to be labelled in expression X different from vertices in $L(i, \varphi_1, G)$. This implies that there are only $k-1$ possible labels for the vertices of $L(i, \varphi_1, G)$ and thus at most $k-1$ neighbourhoods from $L(i, \varphi_1, G)$ into $R(i, \varphi_1, G)$ and thus $|N(L(i, \varphi_1, G), R(i, \varphi_1, G))| + \pi_1(i, \varphi_1, G) \leqslant (k-1) + 1 = k$.

Thus, we have shown that a layout φ_1 exists, such that $\max_{1 \leqslant i < |V|} \nu(i, \varphi_1, G) \leqslant$ linear NLC-width(G).

Table 1
Examples for values used in layout definition of linear NLC-width for graph P_4

i	1	2	3	4	i	1	2	3	4
$\varphi_1^{-1}(i)$	v_1	v_2	v_3	v_4	$\varphi_2^{-1}(i)$	v_1	v_2	v_4	v_3
$\eta(i, \varphi_1, P_4)$	1	2	2	–	$\eta(i, \varphi_2, P_4)$	1	2	2	–
$\pi_1(i, \varphi_1, P_4)$	1	1	0	–	$\pi_1(i, \varphi_2, P_4)$	1	0	0	–

By the first part of the proof we can conclude that

$$\min_{\varphi \in \Phi(G)} \max_{1 \leqslant i < |V|} v(i, \varphi, G) = \max_{1 \leqslant i < |V|} v(i, \varphi_1, G) = \text{linear NLC-width}(G),$$

which completes our proof. □

In Table 1, we give the values for η and π_1 for two layouts of graph $P_4 = (\{v_1, v_2, v_3, v_4\}, \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}\})$.

5. A layout characterization for linear clique-width

Let $G = (V, E) \in \text{lin-CW}_k$ be a graph, $\varphi \in \Phi(G)$ a layout defined by a k-expression for G, $1 \leqslant i < |V|$. We next give a further modification of the neighbourhood cut given in Section 4 which leads to an equivalent layout definition for linear clique-width, independently from vertex labelled graphs. Therefore, we use three boolean graph properties $\pi_i : \mathbb{N} \times \Phi(\mathcal{G}) \times \mathcal{G} \to \{0, 1\}$, $1 \leqslant i \leqslant 3$.

Similar as in the case of linear NLC-width, if there is no vertex in $L(i, \varphi, G)$, $1 \leqslant i \leqslant |V| - 2$, with the same neighbourhood as vertex $\varphi^{-1}(i+1)$ with respect to $R(i+1, \varphi, G)$, we need one additional label for vertex $\varphi^{-1}(i+1)$ which is not used in graph $G[L(i, \varphi, G)]$. Therefore, we again use graph property π_1 of Section 4.

We now assume that there exists a non-empty subset L_1 of $L(i, \varphi, G)$, such that vertex $\varphi^{-1}(i+1)$ has the same neighbourhood as the vertices of L_1 with respect to $R(i+1, \varphi, G)$.

Unfortunately, linear clique-width operations do not allow to connect vertices of the same label by an edge, thus if $\{\varphi^{-1}(i+1), u\} \in E$ for some $u \in L_1$, $1 \leqslant i \leqslant |V| - 2$, we have to label vertex $\varphi^{-1}(i+1)$ differently from the label used for the vertices of L_1 and thus differently from all vertices in $G[L(i, \varphi, G)]$. For $i = |V| - 1$, every vertex in set $L(i, \varphi, G)$ has the same (empty) neighbourhood as vertex $\varphi^{-1}(i+1)$ into set $R(i+1, \varphi, G)$, we need one additional label for vertex $\varphi^{-1}(i+1)$, if $\varphi^{-1}(i+1)$ is adjacent to all vertices of $L(i, \varphi, G)$. We denote this property by $\pi_2(i, \varphi, G)$, which can be expressed as follows:

$$\pi_2(i, \varphi, G) = 1 \quad \Leftrightarrow \quad \begin{cases} \exists\, u \in L(i, \varphi, G) : \\ N_{R(i+1,\varphi,G)}(u) = N_{R(i+1,\varphi,G)}(\varphi^{-1}(i+1)) \\ \wedge \{u, \varphi^{-1}(i+1)\} \in E & \text{if } i \leqslant |V| - 2, \\ \forall\, u \in L(i, \varphi, G) : \{u, \varphi^{-1}(i+1)\} \in E & \text{if } i = |V| - 1. \end{cases}$$

Further, since an edge insertion in graph $G[L(i, \varphi, G)] \oplus \varphi^{-1}(i+1)$ may have an effect on the adjacencies in graph $G[L(i, \varphi, G)]$, we have to verify whether every vertex in $N_{L(i,\varphi,G)}(\varphi^{-1}(i+1))$ is also contained in $N_{L(i,\varphi,G)}(u)$ for every $u \in L_1$. For $i = |V| - 1$, we need one additional label for vertex $\varphi^{-1}(i+1)$, if $\varphi^{-1}(i+1)$ cannot be labelled as its non-neighbours in set $L(i, \varphi, G)$. We denote this property by $\pi_3(i, \varphi, G)$, which can be defined as follows:

$$\pi_3(i, \varphi, G) = 1 \quad \Leftrightarrow \quad \begin{cases} \exists\, u \in L(i, \varphi, G) : \\ N_{R(i+1,\varphi,G)}(u) = N_{R(i+1,\varphi,G)}(\varphi^{-1}(i+1)) \\ \wedge N_{L(i,\varphi,G)}(\varphi^{-1}(i+1)) - \{u\} \not\subseteq N_{L(i,\varphi,G)}(u) & \text{if } i \leqslant |V| - 2, \\ \exists\, u \in L(i, \varphi, G) - N_{L(i,\varphi,G)}(\varphi^{-1}(i+1)) \\ \wedge v \in N_{L(i,\varphi,G)}(\varphi^{-1}(i+1)) : \{u, v\} \notin E & \text{if } i = |V| - 1. \end{cases}$$

The logical or of the defined properties π_i, $1 \leqslant i \leqslant 3$, allows us to define a *modified neighbourhood cut* for $1 \leqslant i < |V|$ as follows:

$$v_2(i, \varphi, G) = \eta(i, \varphi, G) + (\pi_1(i, \varphi, G) \vee \pi_2(i, \varphi, G) \vee \pi_3(i, \varphi, G)).$$

Table 2
Examples for values used in layout definition of linear clique-width for graph P_4

i	1	2	3	4	i	1	2	3	4
$\varphi_1^{-1}(i)$	v_1	v_2	v_3	v_4	$\varphi_2^{-1}(i)$	v_1	v_2	v_4	v_3
$\eta(i, \varphi_1, P_4)$	1	2	2	–	$\eta(i, \varphi_2, P_4)$	1	2	2	–
$\pi_1(i, \varphi_1, P_4)$	1	1	0	–	$\pi_1(i, \varphi_2, P_4)$	1	0	0	–
$\pi_2(i, \varphi_1, P_4)$	0	0	0	–	$\pi_2(i, \varphi_2, P_4)$	0	0	0	–
$\pi_3(i, \varphi_1, P_4)$	0	0	1	–	$\pi_3(i, \varphi_2, P_4)$	0	0	1	–

Analogical to the proof of Theorem 7 we conclude that for every graph G a minimal value for $\max_{1 \leqslant i < |V|} v_2(i, \varphi, G)$, $\varphi \in \Phi(G)$, is equal to its linear clique-width.

Theorem 8. *Let $G = (V, E)$ be a graph with at least two vertices, then*

$$\text{linear clique-width}(G) = \min_{\varphi \in \Phi(G)} \max_{1 \leqslant i < |V|} v_2(i, \varphi, G).$$

In Table 2, we give the values for η, π_1 π_2, and π_3 for two layouts of graph P_4.

6. Relations between the layout measures

In this section we summarize relations between the cut-width, path-width, linear NLC-width, linear clique-width, and neighbourhood-width.

Theorem 9. *Let G be a graph of path-width or cut-width k, then G has linear NLC-width at most $k+2$, linear clique-width at most $k+2$, and neighbourhood-width at most $k+2$.*

Proof. Let G be a graph of path-width k and $(\mathscr{X} = \{X_u \mid u \in V_P\}, P = (\{1, \ldots, n'\}, \{\{1,2\}, \ldots, \{n'-1, n'\}\}))$ be a path-decomposition of width k for G, i.e. $|X_i| \leqslant k+1$, $1 \leqslant i \leqslant n'$. We next recursively define for $i = 1, \ldots, n'$ a linear clique-width $(k+2)$-expression Y_i which defines subgraph $G\left[\bigcup_{j=1}^{i} X_j\right]$. We will use label $k+2$ exclusively for vertices which will not get any further incident edges.

(1) For $i = 1$. Subgraph $G[X_1]$ contains at most $k+1$ vertices and can be defined by a linear clique-width $(k+1)$-expression Y_1 such that each vertex gets a different label.
(2) For $i > 1$. We first relabel in expression Y_{i-1}, which defines subgraph $G\left[\bigcup_{j=1}^{i-1} X_j\right]$, the vertices of $X_{i-1} - X_i$ into label $k+2$, these vertices will not get any further edges during the composition. Since set $X_i = (X_i \cap X_{i-1}) \cup (X_i - X_{i-1})$ contains at most $k+1$ vertices we can insert the vertices of $X_i - X_{i-1}$, each with a label unequal to all other used labels in graph $\text{val}(Y_{i-1})$, and connect them to their neighbours in X_i.

Graph $\text{val}(Y_{n'})$ defines graph G by construction, which implies that the linear clique-width of a graph G is always at most path-width$(G)+2$.

Since the path-width of a graph is always at most its cut-width [2], the neighbourhood-width of a graph is always at most its linear NLC-width (Theorem 5), and the linear NLC-width of a graph is always at most its linear clique-width (Lemma 3), the remaining results follow. $\square$

The proofs of this bounds also show how to define from a given layout φ, a layout φ', such that neighbourhood-width $(\varphi', G) = \text{cut-width}(\varphi, G) + 2$. Thus, the definition of neighbourhood-width leads an extension for the definition of cut-width.

Further we conclude that every graph class of bounded path-width, also has bounded linear NLC-width. On the other hand, graph classes of bounded linear NLC-width do not have bounded path-width in general, e.g. a clique K_n

(complete graph on n vertices) has linear NLC-width 1 and path-width $n-1$. In the next theorem we consider the path-width of linear NLC-width bounded graphs under certain conditions.

For a graph G we want to denote by $\Delta(G)$ the maximum vertex degree of graph G.

Theorem 10. *Let G be a graph of linear NLC-width k or linear clique-width k*

(1) *and the complete bipartite graph $K_{n,n}$ for some $n>1$ is not a subgraph of G, then G has path-width at most* $2k(n-1)$.
(2) *and G is uniformly l-sparse,*[1] *then G has path-width at most* $4kl$.
(3) *and there is a graph with n vertices which is not a minor of G, then G has path-width at most* $2k(n-1)$.
(4) *and G is planar, then G has path-width at most* $4k$.
(5) *then G has path-width at most* $k \cdot \Delta(G)$.

Proof.

(1) We refer to the proof of Theorem 2 of [18] which constructs for a graph of NLC-width k, which does not contain the complete bipartite graph $K_{n,n}$ as a subgraph, from a given NLC-width k-expression-tree T a tree-decomposition $(\mathscr{X}=\{X_u \mid u \in V_T\},\ T=(V_T, E_T))$ of width $3k(n-1)-1$. In decomposition $(\mathscr{X}, T)$ every vertex set X_u either consists of a single vertex or is the union of at most $3k$ sets of size at most $n-1$.
Given a linear NLC-width k-expression-tree T the same proof constructs a tree-decomposition $(\mathscr{X}, T)$ of width at most $2k(n-1)$, since every union node (every node labelled by $\times_S$) in expression tree T has at least one child which is a leaf of T. In this case it is easy to see that T is a caterpillar[2] with hairlength one and vertex degree at most 3 and all sets X_u, for leaves u of T contain exactly one vertex of G. If we remove all vertices of T which do not belong to its backbone, we obtain by $(\mathscr{X}, T)$ a path-decomposition of width $2k(n-1)$, which defines G.
(2) If a graph G is uniformly l-sparse, then graph G contains no complete bipartite graph $K_{2l+1,2l+1}$ as a subgraph.
(3) If there is a graph with n vertices which is not a minor of G then the complete graph K_n is not a minor of G, and thus the $K_{n,n}$ is not a minor of G, and thus the complete bipartite graph $K_{n,n}$ is not a subgraph of G.
(4) Planar graphs do not contain the $K_{3,3}$ as a subgraph.
(5) Let T be a linear NLC-width k-expression tree defining G. If for a node u of T and a label $l \in [k]$, there are more than $\Delta(G)$ vertices in the subgraph defined by the subtree of T with root u, then there will be no further edge insertion to label l. Thus, we can easily define a path-decomposition of width $k \cdot \Delta(G)$ from T. □

The last theorem immediately implies bounds for the path-width of specific graphs. These bounds also hold for the tree-width of a graph which is always less or equal its path-width, in the case of threshold graphs (which are co-graphs) even tree-width equals the path-width [5]. Since threshold are exactly graphs of linear NLC-width 1 [17], and incidence graphs[3] contain no $K_{2,2}$ as a subgraph, the bounds of Theorem 10 imply the following corollary.

Corollary 11.

(1) *Incidence graphs of threshold graphs have path-width at most* 2.
(2) *Planar threshold graphs have path-width at most* 4.
(3) *The path-width of threshold graphs is at most its maximum vertex degree.*

Although graph classes of bounded linear clique-width do not have bounded path-width, in general, there is a very close relation between path-width and the linear clique-width of the corresponding line graphs.[4]

[1] A graph $G=(V_G, E_G)$ is *l-sparse* if $|E_G| \leqslant l \cdot |V_G|$. It is *uniformly l-sparse* if every subgraph of G is l-sparse [8].

[2] A *caterpillar C* is a tree where all vertices of degree at least 3 lie on a path, called the backbone of C. The *hairlength* of a caterpillar is the maximum distance of a non-backbone vertex to the backbone.

[3] The incidence graph $I(G)$ of a graph G is the graph with vertices $V_G \cup E_G$ and an edge joining $v \in V_G$ and $e \in E_G$ if and only if v is incident to e in G.

[4] The line graph $L(G)$ of a graph G has a vertex for every edge of G and an edge between two vertices if the corresponding edges of G are adjacent.

Table 3
Overview on the inclusions of graph families

Arbitrary graphs
$\mathscr{F}(\text{CUTW}) \subsetneq \mathscr{F}(\text{PW}) \subsetneq \mathscr{F}(\text{NW}) = \mathscr{F}(\text{lin-NLC}) = \mathscr{F}(\text{lin-CW})$
Graphs without arbitrary large $K_{n,n}$
$\mathscr{F}(\text{CUTW}) \subsetneq \mathscr{F}(\text{PW}) = \mathscr{F}(\text{NW}) = \mathscr{F}(\text{lin-NLC}) = \mathscr{F}(\text{lin-CW})$
Graphs of bounded vertex degree
$\mathscr{F}(\text{CUTW}) = \mathscr{F}(\text{PW}) = \mathscr{F}(\text{NW}) = \mathscr{F}(\text{lin-NLC}) = \mathscr{F}(\text{lin-CW})$

Theorem 12 (*Gurski and Wanke [19]*). *A set of graphs has bounded path-width if and only if its set of line graphs has bounded linear clique-width.*

A similar relation can be obtained for incidence graphs by Theorems 9 and 10(1) and the fact that incidence graphs contain no $K_{2,2}$ as a subgraph.

Corollary 13. *A set of graphs has bounded path-width if and only if its set of incidence graphs has bounded linear clique-width.*

In order to get bounds for the cut-width of a graph G of bounded linear NLC-width, we can use the conditions of Theorem 10 and the following relation shown in [7].

$$\text{cut-width}(G) \leqslant \text{path-width}(G) \cdot \Delta(G). \tag{1}$$

In fact, the path-width and cut-width of a graph can differ very much, e.g. a $K_{1,n}$ has path-width 1 and cut-width $\lceil n/2 \rceil$.

Notice that a similar result as shown in Theorem 12 and Corollary 13 does not hold for cut-width and line graphs of bounded linear NLC-width or incidence graphs of bounded linear NLC-width. A simple counter-example is the set $\mathscr{L}$ of all $K_{1,n}$ which has unbounded cut-width, but the set of all line graphs of graphs in $\mathscr{L}$ has bounded linear NLC-width and the set of all incidence graphs of graphs in $\mathscr{L}$ has bounded linear NLC-width as well.

Last we want to mention that we can use the conditions of Theorem 10 to bound the cut-width of a graph by its neighbourhood-width by Theorem 5 and the relation (1) shown in [7]. This re-proves the following bound shown in [20].

Lemma 14. *Let G be a graph of neighbourhood-width k, then G has cut-width at most* $(k+1) \cdot \Delta(G)^2$.

Thus, graphs of bounded vertex degree have bounded cut-width if and only if they have bounded neighbourhood-width.

In order to summarize the results, we denote by $\mathscr{F}(\text{CUTW})$ ($\mathscr{F}(\text{PW})$, $\mathscr{F}(\text{NW})$, $\mathscr{F}(\text{lin-NLC})$, $\mathscr{F}(\text{lin-CW})$) the family of all graph classes of bounded cut-width (path-width, neighbourhood-width, linear NLC-width, linear clique-width, respectively), see Table 3.

7. Complexity results

In this section we summarize results on the complexity of minimizing the layout parameters considered in this paper.

The problems of minimizing cut-width and minimizing path-width of a given graph are well known to be NP-complete [16,1]. Recently, Fellows et al. have shown the NP-completeness of minimizing linear clique-width by a reduction from path-width minimization [14].

The reduction for linear NLC-width minimization can be done by the ideas shown in [19,14].

Theorem 15. *Given a graph G and a positive integer k, the problem to decide whether G has linear NLC-width at most k is NP-complete, even for co-bipartite graphs.*

Proof. In [14] for an arbitrary graph G a new graph G' is constructed by substituting every edge $\{u, v\}$ of G by three paths $(\{u, u_i, v_i, v\}, \{\{u, u_i\}, \{u_i, v_i\}, \{v_i, v\}\})$, $1 \leqslant i \leqslant 3$, such that it holds

$$\text{path-width}(G) \leqslant \text{linear clique-width}(G') \leqslant \text{path-width}(G) + 4.$$

By Lemma 3 we can deduce

$$\text{path-width}(G) - 1 \leqslant \text{linear NLC-width}(G') \leqslant \text{path-width}(G) + 4.$$

This inequality can be used to show the NP-completeness of minimizing linear NLC-width.

The problem to decide whether a given graph has linear NLC-width at most k is obviously in NP.

For a graph $G = (V, E)$ and an integer $r > 1$ let G^r be the graph G in that every vertex u is replaced by a clique C_u with r vertices and every edge $\{u, v\}$ is replaced by all edges between the vertices of C_u and C_v. That is, $G^r = (V_r, E_r)$ has vertex set $V_r = \{u_{i,j} \mid u_i \in V, j \in \{1, \ldots, r\}\}$ and edge set

$$E_r = \{\{u_{i,j}, u_{i',j'}\} \mid j, j' = 1, \ldots, r \text{ and } i = i' \vee \{u_i, u_{i'}\} \in E)\}.$$

Bodlaender et al. have shown in [4] that G has path-width k if and only if G^r has path-width $r(k + 1) - 1$.

Arnborg et al. have shown in [1] that path-width minimization is NP-complete, even for co-bipartite graphs. That is, given a graph G and an integer k, the problem to decide whether G has path-width at most k, is NP-complete.

For a given co-bipartite graph G, we first construct graph G^6 and then graph $(G^6)'$, which is still co-bipartite. This can be done in polynomial time. If G has path-width k, then $(G^6)'$ has path-width $6k+5$. By the inequality above $(G^6)'$ has linear NLC-width at least $6k + 4$ and linear NLC-width at most $6k + 9$. That is,

$$\text{path-width}(G) = \left\lfloor \frac{\text{linear NLC-width}((G^6)') - 4}{6} \right\rfloor.$$

Thus, a graph G has path-width at most k if and only if $(G^6)'$ has linear NLC-width at most $6k + 9$, which completes our proof. □

The results of Theorem 5 can be used to show the NP-completeness of finding a minimum layout with respect to neighbourhood cuts.

Corollary 16. *Given a graph G and an integer k, the problem to decide whether G has neighbourhood-width most k is NP-complete, even for co-bipartite graphs.*

Proof. The proof runs similar to the proof of Theorem 15. We again use the construction of [14], which defines for an arbitrary co-bipartite graph G a new graph G', such that it holds

$$\text{path-width}(G) \leqslant \text{linear clique-width}(G') \leqslant \text{path-width}(G) + 4.$$

By Theorem 5 we can conclude

$$\text{path-width}(G) - 1 \leqslant \text{neighbourhood-width}(G') \leqslant \text{path-width}(G) + 4.$$

This inequality implies the NP-completeness of minimizing neighbourhood-width similar as shown in proof of Theorem 15 for linear NLC-width. □

In [14] the following non-approximability result for linear clique-width is shown.

Lemma 17 (*Fellows et al. [14]*). *For every ε, $0 < \varepsilon < 1$, there is no polynomial time algorithm that computes for a given graph G a linear clique-width k-expression such that k-linear clique-width $(G) \leqslant |V_G|^{\varepsilon}$, unless $P = NP$.*

The bounds shown in Lemma 3 and Theorem 5 and the results of [14] imply the following non-approximability result for linear NLC-width and for neighbourhood-width.

Corollary 18.

(1) *For every* ε, $0 < \varepsilon < 1$, *there is no polynomial time algorithm that computes for a given graph G a linear NLC-width k-expression such that k-linear NLC-width* $(G) \leqslant |V_G|^{\varepsilon}$, *unless* $P = NP$.
(2) *For every* ε, $0 < \varepsilon < 1$, *there is no polynomial time algorithm that computes for a given graph G a layout* φ *such that* $\mathrm{nw}(\varphi, G) - nw\ (G) \leqslant |V_G|^{\varepsilon}$, *unless* $P = NP$.

These results also imply that there is no polynomial time approximation algorithm for linear clique-width, linear NLC-width and neighbourhood-width with constant difference guarantee.

The ideas shown in [13] can also be used to compute the linear clique-width of a given graph of bounded tree-width in linear time.

Theorem 19. *Given a graph G of bounded tree-width and a positive integer k, the problem to decide whether G has linear clique-width at most k is decidable in linear time.*

Lemma 3 and Theorem 5 imply the following approximation result for graphs of bounded tree-width of difference guarantee 1.

Corollary 20.

(1) *Given a graph G of bounded tree-width and a positive integer k, then there exists a linear time algorithm that computes for a given graph G a linear NLC-width k-expression such that k-linear NLC-width* $(G) \leqslant 1$.
(2) *Given a graph G of bounded tree-width and a positive integer k, then there exists a linear time algorithm that computes for a given graph G a layout* φ *such that* $nw\ (\varphi, G) - nw\ (G) \leqslant 1$.

8. Conclusions

In this paper we introduced the neighbourhood-width of graphs as a new layout measure for graphs. Our results show that neighbourhood-width leads an extension of the layout parameters vertex separation number and cut-width. For example arbitrary large complete graphs K_n have neighbourhood-width 1, vertex separation number $n-1$ [3], and cut-width $\lceil n/2 \rceil \cdot \lfloor n/2 \rfloor$ [12], complete bipartite graphs $K_{n,m}$ have neighbourhood-width 2, vertex separation number $\min\{n, m\}$ [3], and cut-width $\lceil n \cdot m/2 \rceil$ [12].

The close relation of neighbourhood-width and linear clique-width shown in Section 3, implies that all graph properties which are expressible in monadic second-order logic with quantifications over vertices and vertex sets (MSO_1-logic) are decidable in linear time on neighbourhood-width bounded graphs if a layout for the graph is given as an input [9]. On graph classes of bounded vertex separation number, and thus also on graph classes of bounded cut-width, even all graph properties which are expressible in monadic second-order logic with quantifications over vertices, vertex sets, edges, and edge sets (MSO_2-logic) are decidable in linear time [10].

Simple modifications in the definition of neighbourhood-width lead the first equivalent layout definitions for linear NLC-width and linear clique-width, independently from vertex labelled graphs. These layouts imply simple but exponential algorithms for determining the linear NLC-width and linear clique-width of a given graph.

One of the main open questions is the complexity of the recognition problem for graphs of linear NLC-width at most k, linear clique-width at most k, and neighbourhood-width at most k, for every fixed k, $k \geqslant 2$, $k \geqslant 3$, and $k \geqslant 2$, respectively.

Acknowledgements

The author wants to thank the two anonymous referees for their helpful remarks.

References

[1] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embeddings in a *k*-tree, SIAM J. Algebraic Discrete Methods 8 (2) (1987) 277–284.

[2] H. Bodlaender, Classes of graphs with bounded tree-width, in: Bulletin of the European Association for Theoretical Computer Science, EATCS, vol. 36, 1988, pp. 116–125.

[3] H.L. Bodlaender, A partial *k*-arboretum of graphs with bounded treewidth, Theoret. Comput. Sci. 209 (1998) 1–45.

[4] H.L. Bodlaender, J.R. Gilbert, H. Hafsteinsson, T. Kloks, Approximating treewidth, pathwidth, frontsize, and shortest elimination tree, J. Algorithms 18 (2) (1995) 238–255.

[5] H.L. Bodlaender, R.H. Möhring, The pathwidth and treewidth of cographs, SIAM J. Discrete Math. 6 (2) (1993) 181–188.

[6] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[7] F.R.K. Chung, P.D. Seymour, Graphs with small bandwidth and cutwidth, Discrete Math. 75 (1989) 113–119.

[8] B. Courcelle, The monadic second-order logic of graphs XIV: uniformly sparse graphs and edge set quantifications, Theoret. Comput. Sci. 299 (2003) 1–36.

[9] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, Theory Comput. Systems 33 (2) (2000) 125–150.

[10] B. Courcelle, M. Mosbah, Monadic second-order evaluations on tree-decomposable graphs, Theoret. Comput. Sci. 109 (1993) 49–82.

[11] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, Discrete Appl. Math. 101 (2000) 77–114.

[12] J. Díaz, J. Petit, M. Serna, A survey of graph layout problems, ACM Comput. Surveys (CSUR) 34 (3) (2002) 313–356.

[13] W. Espelage, F. Gurski, E. Wanke, Deciding clique-width for graphs of bounded tree-width, J. Graph Algorithms Appl.—Special Issue of JGAA on WADS 2001 7 (2) (2003) 141–180.

[14] M.R. Fellows, F.A. Rosamund, U. Rotics, S. Szeider, Proving NP-hardness for clique-width I: non-approximability of sequential clique-width, Technical Report TR05-080, ECCC, 2005.

[15] M.R. Fellows, F.A. Rosamund, U. Rotics, S. Szeider, Proving NP-hardness for clique-width II: non-approximability of clique-width, Technical Report TR05-081, ECCC, 2005.

[16] F. Gavril, Some NP-complete problems on graphs, in: Proceedings of the 11th Conference on Information Sciences and Systems, Johns Hopkins University, Baltimore, MD, 1977, pp. 91–95.

[17] F. Gurski, Characterizations for co-graphs defined by restricted NLC-width or clique-width operations, Discrete Math. 306 (2) (2006) 271–277.

[18] F. Gurski, E. Wanke, The tree-width of clique-width bounded graphs without $K_{n,n}$, in: Proceedings of Graph-Theoretical Concepts in Computer Science, Lecture Notes in Computer Science, vol. 1938, Springer, Berlin, 2000, pp. 196–205.

[19] F. Gurski, E. Wanke, Minimizing NLC-width is NP-complete (Extended Abstract), in: Proceedings of Graph-Theoretical Concepts in Computer Science, Lecture Notes in Computer Science, vol. 3787, Springer, Berlin, 2005, pp. 69–80.

[20] F. Gurski, E. Wanke, On the relationship between NLC-width and linear NLC-width, Theoret. Comput. Sci. 347 (1–2) (2005) 76–89.

[21] N.G. Kinnersley, The vertex separation number of a graph equals its path width, Inform. Process. Lett. 42 (1992) 345–350.

[22] R.H. Möhring, Graph problems related to gate matrix layout and PLA folding, Computing 7 (1990) 17–51.

[23] S.-I. Oum, P.D. Seymour, Approximating clique-width and branch-width, J. Combin. Theory Ser. B, 2006, in press.

[24] N. Robertson, P.D. Seymour, Graph minors I. Excluding a forest, J. Combin. Theory Ser. B 35 (1983) 39–61.

[25] M. Serna, D.M. Thilikos, Parameterized complexity for graph layout problems, in: Bulletin of the European Association for Theoretical Computer Science, EATCS, vol. 86, 2005, pp. 41–65.

[26] E. Wanke, *k-NLC* graphs and polynomial algorithms, Discrete Appl. Math. 54 (1994) 251–266.