On powers of graphs of bounded NLC-width (clique-width)*

Karol Suchan, Ioan Todinca

LIFO - Université d'Orléans
BP 6759 F-45067 France
Karol.Suchan,Ioan.Todinca@lifo.univ-orleans.fr

Abstract. Given a graph G, the graph G^l has the same vertex set and two vertices are adjacent in G^l if and only if they are at distance at most l in G. The l-coloring problem consists in finding an optimal vertex coloring of the graph G^l, where G is the input graph. We show that, for any fixed value of l, the l-coloring problem is polynomial when restricted to graphs of bounded NLC-width (or clique-width), if an expression of the graph is also part of the input. We also prove that the NLC-width of G^l is at most $2(l+1)^{\mathrm{nlcw}(G)}$.

1 Introduction

Many generalizations of the coloring problem of graphs were proposed as models for real-life applications. One of them is the radio-coloring problem (also called λ-coloring or $L(2,1)$-coloring), in which the vertices of the graph represent transmitters, and two vertices are adjacent if and only if the two transmitters are very close. Each vertex must receive an integer frequency, such that adjacent vertices (very close transmitters) get frequencies differing by at least two units, and vertices at distance two in the graph (close transmitters) get different frequencies. The aim is to minimize the range of the used frequencies, i.e. we search for the minimum λ such that G admits a radio-coloring with frequencies between 0 and λ. Let us call $\lambda_{2,1}$ this minimum. Another generalization of the classical coloring is the l-coloring problem, i. e. the coloring of G^l, where G is the input graph. Here $G^l = (V, E^l)$ denotes the l^{th} power of G: two vertices are adjacent in G^l if and only if they are at distance at most l in G. The radio-coloring and the l-coloring problems are related. Every radio-coloring is also a 2-coloring. On the other hand, by multiplying the values of colors in a 2-coloring by two, we easily obtain a radio-coloring. Thus $\lambda_{1,1} \leq \lambda_{2,1} \leq 2\lambda_{1,1}$, where $\lambda_{1,1}$ denotes the minimum number of colors necessary for a 2-coloring. Hence an algorithm solving the 2-coloring problem for a class of graphs also provides a 2-approximation for the radio-coloring problem. Both problems are NP-hard for general graphs [1, 6]. The l-coloring problem is polynomial for graphs of tree-width at most k, for any fixed k and l [12].

In this paper we study the NLC-width of G^l and the l-coloring problem on graphs of bounded NLC-width (or, equivalently, of bounded clique-width).

The class of k-node label controlled graphs (denoted NLC_k) is recursively defined using k vertex-labels and three operators: $\bullet_i$, $\circ_R$, $\times_S$. The operation $\bullet_i$ creates a graph with a single vertex labelled $i \in [k]$, where $[k] = \{1, 2, \ldots, k\}$. The operation $\circ_R$, with R being a total function $[k] \to [k]$, replaces every instance of label i with $R(i)$. The binary operator $\times_S$, with $S \subseteq [k] \times [k]$, creates the disjoint union of given graphs G, H and adds edges between the vertices of G labelled i and the vertices of H labelled j, for every $(i, j) \in S$. The NLC-width of an unlabelled graph G is the minimum value of k such that G, with some appropriate labelling, is in NLC_k. The clique-width parameter is defined in a very similar way. The class of graphs CWD_k is recursively defined using four operators. The operator $\bullet_i$, like above, creates a graph with a unique vertex labelled i. The unary operator $\rho_{i \to j}$ relabels all instances of label i with label j and the unary operator $\eta_{i,j}$ adds all the edges xy with x (resp. y) of label i (resp. j). The binary operator $\oplus$ constructs the disjoint union of two graphs. The clique-width and the NLC-width of

* A preliminary version of this paper appeared as [10] in Proceedings 29th Workshop on Graph-theoretic Concepts in Computer Science (WG 2003).

a graph differ by at most a factor of two. All the results of this paper could be stated either in terms of clique-width or using NLC-width. Nevertheless, we consider that in our case it was more convenient to use NLC-width, in order to simplify some notations.

Many NP-hard problems become polynomially solvable for graphs of bounded NLC-width (clique-width), if an expression of the graph, using the operators above, is part of the input. The classical coloring problem is one of them [5, 8]. Let us note that graphs of bounded tree-width are a particular case of graphs of bounded clique-width, more precisely if G has tree-width at most k then $\mathrm{cwd}(G) \leq 2^{k+1} + 1$ [4]. Several other classes of graphs (distance-hereditary graphs, P_4-sparse and P_4-tidy graphs, ...) are known to have bounded clique-width.

Up to now, one of the limits for these decomposition techniques was due to the fact that there were no good algorithms for computing or approximating the clique-width or the NLC-width of a graph. Except for very small values of the parameters ($\mathrm{cwd} \leq 3$ or $\mathrm{nlcw} \leq 2$), there was only an $\mathcal{O}(\lg n)$ approximation algorithm which is not sufficient for most of the problems. Recently, Sang-il Oum and Paul Seymour announced a 3-approximation algorithm for the rank-width of a graph, and this parameter is strongly related to clique-width: $\text{rank-width} \leq \mathrm{cwd} \leq 2^{\text{rank-width}+1}$ [9]. The new result might strongly increase the interest for graphs of bounded clique-width.

The initial motivation for this work was the radio-coloring problem for graphs of bounded clique-width, but it turns out that the problem is NP-complete even for graphs of clique-width at most 3 [2]. We show in this article that the l-coloring problem is polynomial when restricted to graphs of bounded NLC-width, if an expression of the graph is part of the input. This result extends the one on graphs of bounded tree-width [12]. As mentioned above, it provides a 2-approximation for the radio-coloring problem on graphs of bounded NLC-width.

After introducing the basic notions in Section 2, we show in Section 4 that, for any graph G of NLC-width k, the NLC-width of G^l is at most $2(l+1)^k$. We think that this result is interesting in its own right, since it shows that, when considering a class of graphs of bounded NLC-width, the NLC-width stays bounded when taking powers. Note that a similar result does not hold for tree-width.

Using the coloring algorithm for graphs of bounded cliquewidth proposed in [8], this result directly implies that, for fixed k and l, the l-coloring problem is solvable in $\mathcal{O}(n^{2^{4\cdot 2^{k\lg(l+1)}+1}})$ time when restricted to graphs of clique-width at most k if an expression of G is part of the input. In Section 5 we give an algorithm for the l-coloring problem with a $\mathcal{O}(n^{3\cdot 2^{k\lg(l+1)}} n^4)$ time bound. Although the complexity remains high, it is considerably lower than the previous one. Moreover, this time bound is to be compared with the existing time bounds for classical coloring problem on graphs with clique-width at most k ($\mathcal{O}(n^{2^{2k+1}})$, see [8]) or with the time-bound of the l-coloring problem on partial k-trees ($\mathcal{O}(n^{2^{2(k+1)(l+2)+1}})$, see [12]).

2 Basic definitions

Throughout this paper we consider simple, finite, undirected graphs. Given a graph $G = (V, E)$, we denote by n the number of vertices of G. An edge $\{x, y\}$ will be simply denoted by xy. A path $\mu = [x_1, \ldots, x_l]$ is a sequence of vertices such that $x_i x_{i+1} \in E$, $\forall i, 1 \leq i < l$. The length $|\mu|$ of this path is $l - 1$.

The distance between two vertices x and y of G, denoted by $\mathrm{dist}_G(x, y)$, is the length of the shortest path from x to y. Also, $\mathrm{dist}_G(W_1, W_2)$ denotes the distance between two sets of vertices $W_1, W_2 \in V$. We define by $G^l = (V, E^l)$ the l^{th} power of G, i.e. $E^l = \{xy \mid \mathrm{dist}_G(x, y) \leq l\}$.

Let $[k] = \{1, \ldots, k\}$ be the set of integers from 1 to k. A k-labelled graph $\hat{G} = (V, E, \mathrm{lab})$ is a graph (V, E) whose vertices are labelled by some mapping $\mathrm{lab} : V \to [k]$. Given a labelled graph $\hat{G} = (V, E, \mathrm{lab})$, we denote by G the unlabeled graph (V, E).

There are two very similar graph parameters, namely the clique-width (definde by Courcelle and Olariu [4]) and the NLC-width (defined by Wanke [11]). Both express the fact that a graph can be constructed using a simple grammar, and the two parameters differ by at most a factor of two. The results of this paper could be expressed using either clique-width or NLC-width. For technical convenience, we rather use the notion of NLC-width.

Definition 1 (NLC-width [11]). *Let k be some positive integer. The class NLC_k of k-labeled graphs is defined as follows.*

1. *For any $i \in [k]$, the graph $\bullet_i$ with a single vertex labeled i is in NLC_k.*
2. *Consider two disjoint graphs $\hat{G} = (V_G, E_G, \mathrm{lab}_G)$ and $\hat{H} = (V_H, E_H, \mathrm{lab}_H)$ in NLC_k and let $S \subseteq [k]^2$ be a relation. Then the graph $\hat{G} \times_S \hat{H} = (V', E', \mathrm{lab}')$ defined by $V' = V_G \cup V_H$, $E' = E_G \cup E_H \cup \{xy \mid x \in V_G, y \in V_H, (\mathrm{lab}_G(x), \mathrm{lab}_H(y)) \in S\}$, and*

$$\mathrm{lab}'(x) = \begin{cases} \mathrm{lab}_G(x) & \textit{if } x \in V_G \\ \mathrm{lab}_H(x) & \textit{if } x \in V_H \end{cases}, \forall x \in V'$$

   is in NLC_k.
3. *Let $\hat{G} = (V, E, \mathrm{lab}) \in NLC_k$ and $R : [k] \to [k]$ a function, then $\circ_R(G) = (V, E, \mathrm{lab}')$ defined by $\mathrm{lab}'(x) = R(\mathrm{lab}(x)), \forall x \in V$ is in NLC_k.*

An unlabeled graph $G = (V, E)$ is in NLC_k if there is a labeling lab *of G such that $(V, E, \mathrm{lab}) \in NLC_k$. The* NLC-width *of a graph $G = (V, E)$, denoted* $\mathrm{nlcw}(G)$*, is the smallest positive integer k such that $G \in NLC_k$.*

3 Parsing the expression tree

To any graph $\hat{G} \in NLC_k$ we can associate an expression tree T using the operators $\bullet_i$, $\times_S$ and $\circ_R$. We say that T is a k-expression tree and that $\hat{G}$ is the value $\mathrm{val}(T)$ of T. For simplicity, we also say that T *corresponds* to the labelled graph $\hat{G}$ and to the unlabelled graph G.

As usual in the decomposition techniques, our results will be based on a bottom-up parsing of the expression tree of the graph. In this section we give the notations and the main technical lemmas that are used both for computing the NLC-width and the optimal coloring of G^l.

Consider a node u of the expression tree T. We denote by $T[u]$ the subtree of T rooted in u and let $\hat{G}[u] = (V[u], E[u], \mathrm{lab}_u)$ be the graph $\mathrm{val}(T[u])$. $V_i[u]$ denotes the vertices of label i in $\hat{G}[u]$.

If u is a $\times_S$ node and v, w are its sons, we denote by $\mathrm{dist}_G^{\mathrm{ext}}(V_i[v], V_j[w])$ the length of the shortest path μ of G such that μ goes from $V_i[v]$ to $V_j[w]$ and μ does not intersect $V[v]$ or $V[w]$, except in its extremities.

Lemma 1. *Consider an expression tree of the graph G, let u be a $\times_S$ node of the tree and v, w be the two sons of u. Let $x \in V_i[v]$ and $y \in V_j[w]$. A shortest external x, y-path (with only its ends in $V[v] \cup V[w]$) is of length $\mathrm{dist}_G^{\mathrm{ext}}(V_i[v], V_j[w])$.*

Proof. Let $\mu = [x_1, x_2, \ldots, x_p]$ be a shortest path from $V_i[v]$ to $V_j[w]$ in G, such that the interior of μ does not meet $V_i[v]$ or $V_j[w]$. Then $x_1 \in V_i[v]$ and $x_p \in V_j[w]$. The second vertex x_2 of μ is not in $V[v]$, and since it is adjacent to x_1 in G it also adjacent to all vertices having the same label as x_1 in $G[v]$, by definition of the class NLC_k. In particular, x_2 is adjacent to x. For similar reasons, x_{p-1} is adjacent to y in G. So $\mu' = [x'_1 = x, x_2, \ldots, x_{p-1}, x'_p = y]$ is a path of G of length $\mathrm{dist}_G^{\mathrm{ext}}(V_i[v], V_j[w])$. Clearly every external x, y-path is of length at least $\mathrm{dist}_G^{\mathrm{ext}}(V_i[v], V_j[w])$ and the conclusion follows. □

Proposition 1. *Consider an expression tree of the graph G, let u be a $\times_S$ node of the tree and v, w be the two sons of u. Let $x \in V[v]$ and $y \in V[w]$. Then*

$$\mathrm{dist}_G(x, y) = min\{\mathrm{dist}_G(x, V_p[v]) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w]) + \mathrm{dist}_G(y, V_q[w]) \mid (p, q) \in [k]^2\}$$

Proof. Let $\mu = [x = x_1, x_2, \dots, x_s = y]$ be a shortest path from x to y in G. Let $r, 1 < r \leq s$ be the smallest index such that $x_r \in V[w]$ and $l, 1 \leq l < s$ be the biggest index such that $x_l \in V[v]$. Then:

$$\mathrm{dist}_G(x,y) = \mathrm{dist}_G(x,x_l) + \mathrm{dist}_G(x_l,x_r) + \mathrm{dist}_G(x_r,y)$$

Suppose $x_l \in V_i[v]$ and $x_r \in V_j[w]$, so:

$$\mathrm{dist}_G(x,y) \geq \mathrm{dist}_G(x,V_i[v]) + \mathrm{dist}_G^{\mathrm{ext}}(V_i[v],V_j[w]) + \mathrm{dist}_G(y,V_j[w])$$

and in particular

$$\mathrm{dist}_G(x,y) \geq min\{\mathrm{dist}_G(x,V_p[v]) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v],V_q[w]) + \mathrm{dist}_G(y,V_q[w]) \mid (p,q) \in [k]^2\}$$

On the other hand, for every pair $(p,q) \in [k]^2$ such that the distance $\mathrm{dist}_G^{\mathrm{ext}}(V_p[v],V_q[w]) < \infty$, there is an x,y-path in G of length $\mathrm{dist}_G(x,V_p[v]) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v],V_q[w]) + \mathrm{dist}_G(y,V_q[w])$. Indeed, suppose that $x'_l \in V_p[v]$ is a vertex for which the distance $\mathrm{dist}_G(x,V_p[v])$ is achieved. Similarly for $x'_r \in V_q[w]$. By Lemma 1, $\mathrm{dist}_G^{\mathrm{ext}}(x'_l,x'_r) = \mathrm{dist}_G^{\mathrm{ext}}(V_p[v],V_q[w])$. Therefore, concatenating the subpaths $x - x'_l$, $x'_l - x'_r$, $x'_r - y$ that realize the respective distances, we obtain an x,y-path as needed. Hence:

$$\mathrm{dist}_G(x,y) \leq min\{\mathrm{dist}_G(x,V_p[v]) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v],V_q[w]) + \mathrm{dist}_G(y,V_q[w]) \mid (p,q) \in [k]^2\}$$

and the equality is proved. □

The next result gives us a tool to control when two vertices, one in $V[v]$ and the other in $V[w]$, are at distance at most l in G.

Proposition 2. *Consider an expression tree of the graph G, let u be a $\times_S$ node of the tree and v, w be the two sons of u. Let $x \in V[v]$. Then*

$$\mathrm{dist}_G(x,V_j[w]) = min\{\mathrm{dist}_G(x,V_p[u]) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v],V_q[w]) + \mathrm{dist}_G(V_q[w],V_j[w]) \mid (p,q) \in [k]^2\}$$

$$\mathrm{dist}_G(x,V_j[u]) = min\{\mathrm{dist}_G(x,V_j[v]), \mathrm{dist}_G(x,V_j[w])\}$$

Proof. For the first equation, we simply apply Proposition 1 over all vertices $y \in V_j[w]$. The second part is obvious.

Since, given a graph G, we deal with the graph G^l, we are only interested in the distance between two vertices (or two vertex subsets) of G if this distance is at most l. Hence the distance function that we use will only have values in the set $\{0, 1, \dots, l-1, \infty\}$, denoted by $[\![l]\!]$. (For technical reasons even the value l is not necessary.) Let us introduce some other notations that we use all throughout the paper.

Notation 1 *Let $\hat{G} \in NLC_k$ and T be a k-expression tree of G. For some node u of T and any vertex x of G, let*

$$d_i^u(x) = \begin{cases} \mathrm{dist}_G(x,V_i[u]) & \textit{if } \mathrm{dist}_G(x,V_i[u]) < l \\ \infty & \textit{otherwise} \end{cases}$$

The similar notation $d_i^u(W)$ is used for a set of vertices $W \subseteq V$.

We restate now the results of Propositions 1 and 2 using the new notations.

Corollary 1. *Let $\hat{G} \in NLC_k$ and T be a k-expression tree corresponding to G. Let u be a $\times_S$ node of the tree and v, w be the two sons of u. Consider two vertices $x \in V[v]$ and $y \in V[w]$.*

$$\mathrm{dist}_G(x,y) \leq l \textit{ iff } \exists p,q \in k \textit{ s.t. } d_p^v(x) + d_q^w(y) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w]) \leq l. \tag{1}$$

$$\begin{aligned} d_i^w(x) &= \begin{cases} \min\{\ d_p^v(x) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w]) + \\ \qquad \mathrm{dist}_G(V_q[w], V_i[w]) \mid (p,q) \in [k]^2\} & \textit{if this minimum is } < l \\ \infty & \textit{otherwise} \end{cases} \\ d_i^u(x) &= \min(d_i^v(x), d_i^w(x)) \end{aligned} \tag{2}$$

Proof. Equation 2 is obtained by simply plugging Notation 1 in Proposition 2.

For Equation 1, observe that in Lemma 1, the value of $\mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w])$ is at least one. Hence Lemma 1 can be rewritten in this new form. ☐

4 NLC-width and powers of graphs

We show in this section that, for any a graph $\hat{G} \in NLC_k$, the NLC-width of G^l is at most $2(l+1)^k$. The following theorem constructs, from a k expression tree of G, an expression tree corresponding to G^l whose labels are (roughly speaking) in $[\![l]\!]^k$.

The construction proceeds by parsing bottom-up the expression T of $\hat{G}$. One of the key points is that, when we are in the node u, we construct an expression tree $f(u)$ corresponding to the graph $G^l[u]$ induced by $V[u]$ in G^l: two vertices x and y are adjacent in $G^l[u]$ if and only if they are at distance at most l in the *whole* graph G, in particular the shortest path from x to y in G may not be contained in $V[u]$.

Theorem 1. *Let k and l be some positive integers, $\hat{G} = (V, E, \mathrm{lab}) \in NLC_k$ and T be a k-expression tree of G. There is a function f associating to each node u of T an expression tree $f(u)$ with labels in $[\![l]\!]^k \cup ([\![l]\!]^k \times \{\perp\})$ such that:*

1. *$f(u)$ is an expression tree corresponding to $G^l[u] = (V[u], \{xy \mid x, y \in V[u]$ and $\mathrm{dist}_G(x,y) \leq l\})$;*
2. *for any $x \in V$, its label in $\mathrm{val}(f(u))$ is exactly $(d_1^u(x), \ldots, d_k^u(x))$.*

Proof. The expression trees $f(u)$ are obtained by parsing the tree T, from bottom to top. In the construction of $f(u)$, we distinguish three cases, depending whether u is a $\bullet$ node, a $\circ$ node, or a $\times$ node.

The node u is a $\bullet_i$ node. In this case $T[u]$ has a unique node $\bullet_i$, let $f(u) = \bullet_{(d_1, d_2, \ldots, d_k)}$ with $d_j = 0$ if $j = i$ and $d_j = \infty$ otherwise. Clearly $f(u)$ satisfies our conditions.

The node u is a $\circ_R$ node Let v be the unique son of u. The expression tree $f(v)$ also corresponds to the unlabelled graph $G^l[u]$. We have to perform a relabelling on $f(v)$ in order to satisfy the second condition of our theorem.

A vertex x of G is at distance at most d from $V_i[u]$ if and only if it is at distance at most d from $V_j[v]$ for some $j \in [k]$ such that $R(j) = i$. Hence the ith coordinate of the label of x in $f(u)$ is $d_i^u(x) = \min\{d_j^v(x) \mid R(j) = i\}$, where $d_j^v(x)$ is the jth coordinate of the label of x in $f(v)$.

Consider now the function $R' : [\![l]\!]^k \to [\![l]\!]^k$ defined by $R'(d_1^v, d_2^v, \ldots d_k^v) = (d_1^u, d_2^u, \ldots, d_k^u)$ with $d_i^u = \min\{d_j^v \mid R(j) = i\}$. The expression-tree $\circ_{R'}(f(v))$ satisfies the conditions of our theorem.

The node u is a $\times_S$ node Let v, w be the two sons of u. The expression-trees $f(v)$ and $f(w)$ correspond to $G^l[v]$ and $G^l[w]$ respectively. The expression tree $f(u)$ will be constructed by gluing the two subtrees, and we also have to complete the following tasks:

- add all the edges xy with $x \in V[v]$ and $y \in V[w]$ and $\mathrm{dist}_G(x, y) \leq l$ (all the other edges are already present);
- update the labels for each $x \in V[u]$ in order to represent the distances from x to the sets $V_i[u]$ (this distance becomes $\min\{d_i^v(x), d_i^w(x)\}$).

In order to achieve these two goals, we shall use the Equations 1 and 2 of Corollary 1. We shall add the missing edges usind a $\times_{S'}$ nodes, than perform the relabellings (one can easily convince that it is not possible to swap the two operations). Nevertheless different relabellings are needed for $f(v)$ and $f(w)$ and we should be able to distinguish the vertices coming from the two subtrees. Therefore, we first mark the vertices of $f(w)$ as follows. Consider a function $R_m : [\![l]\!]^k \to [\![l]\!]^k \times \{\perp\}$ defined by $R_m(d_1, \ldots, d_k) = (d_1, \ldots, d_k, \perp)$. First, we apply the operator $\circ_{R_m}$ to $f(w)$.

Let $x \in V[v]$ and $y \in V[w]$ two vertices of G. The label of x (resp. y) in $f(v)$ (resp. $f(w)$) is $(d_1^v(x), \ldots, d_k^v(x))$ (resp. $(d_1^w(y), \ldots, d_k^w(y))$). According to Equation 1, x and y are at distance at most l in G if and only if

$$\exists p, q \in k \text{ s.t. } d_p^v(x) + d_q^w(y) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w]) \leq l$$

Therefore we construct a binary relation $S' \subseteq [\![l]\!]^k \times ([\![l]\!]^k \times \{\perp\})$ such that $(d_1^v, \ldots, d_k^v), (d_1^w, \ldots, d_k^w, \perp)) \in S$ if and only if there exist $p, q \in k$ such that $d_p^v(x) + d_q^w(y) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w]) \leq l$. Then the expression tree $f(v) \times_{S'} (\circ_{R_m}(f(w)))$ corresponds exactly to $G^l[u]$.

To conform with the second condition of the theorem, we prepare the relabelling. We use Equation 2 to define the function $R^v : [\![l]\!]^k \to [\![l]\!]^k$ updating the labels of $V[v]$ as $R^v(d_1^v, \ldots, d_k^v) = (d_1^u, \ldots, d_k^u)$, where

$$d_j^u = min(\{d_j^v\} \cup \{d_p^v + d^{ext}(V_p[v], V_q[w]) + d(V_q[w], V_j[w]) | (p, q) \in [k]^2\})$$

(all the values greater than $l - 1$ are rounded to ∞). Symmetrically, we define R_w to be the function updating the labels of $V[w]$, with the differeence that R_w is defined on marked labels.

The expression tree $f(u) = \circ_{R_m^{-1}}(\circ_{R^w}(\circ_{R^v}(f(v) \times_{S'} (\circ_{R_m}(f(w))))))$ achieves our two tasks listed above (the operator $\circ_{R_m^{-1}}$ is used to erase the temporary marks). Hence it satisfies the conditions of the theorem. □

From Theorem 1 we deduce:

Theorem 2. *For any unlabeled graph G of NLC-width k and any positive integer l, we have that* $\mathrm{nlcw}(G^l) \leq 2(l + 1)^k$.

Proof. Consider an k-expression tree T corresponding to G. Theorem 1 constructs an expression tree T' corresponding to G^l. Moreover, T' uses at most $2(l + 1)^k$ different labels. □

Remark 1. The proof of Theorem 1 can easily be turned into a polynomial time algorithm that, given a graph G of NLC-width at most k and a k-expression tree corresponding to G, constructs a $2(l + 1)^k$ expression tree corresponding to G^l.

The upper bounds that we give for the NLC-width of G^l are exponential in k, the NLC-width of G. It is natural to ask wether this bounds could be improved, in particular transformed into polynomial bounds. We think that on the contrary, for any k, there is a graph G of NLC-width k such that $\mathrm{nlcw}(G^2)$ is at least $2^{\alpha k}$, where $\alpha > 0$ is a constant not depending on k. Unfortunately we were not able to prove this result up to now. Let us mention that the existing techniques used to establish lower bounds for the NLC-width of particular graphs are relatively difficult [3, 7].

5 The l-coloring problem for graphs of bounded NLC-width

Given a graph $G \in NLC_k$ and a k-expression of G, one can compute a coloring of G in polynomial time. Two different approaches have been proposed for this problem [5, 8]. We can conclude that, for graphs of bounded NLC-width and for bounded l, the chromatic number of G^l can be computed in $\mathcal{O}(n^{2^{4\cdot 2^{k\lg(l+1)}+1}})$ time, if the NLC_k-expression tree of G is part of the input.

In this section, we give an algorithm for the l-coloring problem, with a $\mathcal{O}(n^{3\cdot 2^{k\lg(l+1)}}n^4)$ time bound. The technique used here generalizes the coloring algorithm for graphs of bounded NLC-width proposed by Espelage et al. [5]. Let us note that we could also adapt the algorithm of Kobler and Rotics [8] (which was actually done in the preliminary version of this paper [10]), but the latter version is more complicated.

5.1 The encoding

The l-coloring on $G = (V, E)$ is considered here as a partition problem. We say that a partition of V into r sets $W_1, \dots, W_a$ is *admissible* if no set W_α has two vertices at distance at most l in G. The l-coloring problem consists in computing the minimum number a such that there exist an admissible partition of V into a sets.

Let $\hat{G} = (V, E, \mathrm{lab})$ and T be a k-expression tree of G. Let u be a node of T with corresponding vertex set $V[u]$ and its partition $\mathcal{W} = \{W_1^u, \dots, W_a^u\}$.

In our algorithm, we encode partitions by attributing to each of its partition-sets a label. This label is similar to what we use in Section 4, i.e. is a k-tuple of distances. In this case, they are computed as the minimum over all the vertices of the considered partition set.

In detail, for the partition $\mathcal{W}$ we define the encoding $\mathcal{M} = \mathrm{enc}(\mathcal{W})$ to be a multiset $\langle M_1^u, \dots, M_a^u \rangle$. Each M_α^u is the encoding $\mathrm{enc}(W_\alpha^u)$ of the corresponding partition-set. We define M_α^u to be a k-tuple $(d_1^u(W_\alpha^u), \dots, d_k^u(W_\alpha^u))$, where (compare Notation 1)

$$d_j^u(W_\alpha^u) = \min\{d_j^u(x) \mid x \in W_\alpha^u\}$$

Notice that we use $\mathcal{M}$ typesetting to denote a multiset and M for a single label. Similarly, we use $\mathcal{W}$ for a partition and W for its constituting partition-sets. For the sake of simplicity, we omit the name of considered partition-set whenever this is clear from the context, i.e. write $(d_1^u, \dots, d_k^u)$ in place of $(d_1^u(W_\alpha^u), \dots, d_k^u(W_\alpha^u))$.

Let $F(u)$ be the set of all multisets $\mathcal{M}$, for all admissible partitions $\mathcal{W}$ of $V[u]$ (here again, a partition of $V[u]$ is admissible if no part has two vertices at distance at most l in the whole graph G).

Note that a single multiset $\mathcal{M} \in F[u]$ may correspond to several admissible partitions. Nevertheless, all these partitions have the same number of parts, which is precisely the cardinality of $\mathcal{M}$. Furthermore, in our discussion it is useful to consider the partition $\mathcal{W}$ corresponding to $\mathcal{M}$ as any of them, still having all the properties crucial for the algorithm. (For formal exactitude we might define equivalence classes, on the set of partitions, of the relation: having the same encoding.)

Proposition 3. *For any node u of T, the size of $F(u)$ is at most $(n+1)^{2^{k\lg(l+1)}}$.*

Proof. To count all the possible multisets, we consider distinct set encodings $\mathrm{enc}(W)$ and the number of times they appear in the multiset. Since $M \in [\![l]\!]^k$, there are $(l+1)^k = 2^{k\lg(l+1)}$ possibilities. For each of them we map a number $o, 0 \le o \le n$ of occurences, as there may be at most n partition sets. That gives $(n+1)^{2^{k\lg(l+1)}}$ mappings. □

5.2 The algorithm

The algorithm computes the set $F[u]$ for every node u of the tree, from bottom to top. In this way, we keep track of all the admissible partitions of $V[u]$. In particular if $root$ is the root of the tree, let $\mathcal{M}_{opt}$ be an element of $F[root]$ of minimum cardinality. The chromatic number of G^l is $|\mathcal{M}_{opt}|$.

The algorithm can be easily adapted in order to store an admissible partition corresponding to each multiset, in particular we can obtain an optimal l-coloring of G.

Let u be the current node that is parsed in the tree, consider that the sets $F[v]$ have been computed for all the descendants v of u, and let us compute $F[u]$ by distinguishing the three types of nodes: $\bullet$, $\circ$ and $\times$. The technique is quite similar to the computation of $\mathrm{nlcw}(G^l)$.

u is a $\bullet_t$ node. We create partition $\mathcal{W}$ of single partition-set. Obviously it is admissible. Define $F[u] = \{\langle(d_1^u, d_2^u, \ldots, d_k^u)\rangle\}$, where $d_i^u = 0$ for $i = t$ and $d_i = \infty$ otherwise.

u is a $\circ_R$ node. All the partitions stay unchanged, we only need to do a relabelling. Let v be the unique son of u. We obtain $F[u]$ by applying an update function R' to all the labels in multisets $\mathcal{M} \in F[v]$.

As in the proof of Theorem 1, we use the given function R to construct the function $R' : [\![l]\!]^k \to [\![l]\!]^k$. Define $R'(d_1^v, d_2^v, \ldots d_k^v) = (d_1^u, d_2^u, \ldots, d_k^u)$ with $d_i^u = \min\{d_j^v \mid R(j) = i\}$.

Lets start with $F[u]$ being an empty set. Then for every $\mathcal{M}^v = \langle M_1^v, \ldots, M_a^v\rangle \in F[v]$ we construct the corresponding $\mathcal{M}^u = \langle M_1^u, \ldots, M_a^u\rangle$, where $M_\alpha^u = R'(M_\alpha^v)$, and add it to $F[u]$.

u is a $\times_S$ node. We construct new partitions, making sure that they are admissible and we proceed with relabelling. Let v, w denote the two sons of u in T. We create $F[u]$ by considering all the pairs in $(\mathcal{M}^v, \mathcal{M}^w) \in F[v] \times F[w]$ and adding to $F[u]$ the multisets that correspond to admissible partitions obtained by recombining $\mathcal{W}^v$ and $\mathcal{W}^w$.

Let $(M_1^v, \ldots, M_a^v) \in F(v)$ be the encoding of an admissible partition of $V[v]$ and $(M_1^w, \ldots, M_b^w) \in F(w)$ be the encoding of an admissible partition of $V[w]$. For such a pair, consider a new partition $(W_1^u, \ldots, W_c^u)$ of $V[u]$, where every W_γ^u is either

a) W_α^v, for some $\alpha \in [a]$ or
b) W_β^w, for some $\beta \in [b]$ or
c) $W_\alpha^v \cup W_\beta^w$ for some couple $(\alpha, \beta) \in [a] \times [b]$

It is enough to process such recombinations, since in $F[v]$ and $F[w]$ there are all admissible partitions encoded. That is to say, computing joins of at most two partition-sets is enough to cover all the possibilities. The others come as 2-joins for different $(\mathcal{W}^v, \mathcal{W}^w)$.

The first two cases are not a threat for the admissibility of the resulting partition, but the third is. We need to check if there is a pair (x, y) of vertices in $W_\alpha^v \times W_\beta^w$ that are neighbors in G^l (i.e. $\mathrm{dist}_G(x, y) \le l$). By Equation 1 of Corollary 1 and since our labels store the minimal values of distances, that is the case if and only if

$$\min\{d_p^v(W_\alpha^v) + \mathrm{dist}_G^{\mathrm{ext}}(V_p[v], V_q[w]) + d_q^w(W_\beta^w) | (p, q) \in [k]^2\} \le l \tag{3}$$

If there is no such pair of adjacent vertices, we can merge the partition-sets.

In the case $W_\gamma^u = W_\alpha^v, \alpha \in [p]$ we use Equation 2 of Corollary 1 to update the $enc(W_\alpha^v)$ to reflect as the distances change when passing from $V[v]$ to $V[u]$. In detail, replace each distance $d_q^v(W_\alpha^v)$ with $d_q^u(W_\alpha^v)$, for $1 \le q \le k$. Let us use $\star$ to denote this operation. We proceed similarly for the case $W_\beta^w, \beta \in [q]$.

In the case $W_\gamma^u = W_\alpha^v \cup W_\beta^w$, for every element of the label M_γ^u we take

$$d_q^u(W_\gamma^u) = \min\{d_q^u(W_\alpha^v), d_q^u(W_\beta^w)\}$$

Notice that this is the minimum of corresponding elements in the labels M^v_α and M^w_β after the update. Let us use $\otimes$ to denote this operator working on updated labels.

To sum up the $\times_S$ node processing we give the following algorithm calculating $F[u]$ (see also the similar algorithm of [5]):

initialize $\mathcal{D}$ with $\{\langle\rangle\} \times F[v] \times F[w]$
for each $(\mathcal{M}, \mathcal{M}', \mathcal{M}'')$ unprocessed in $\mathcal{D}$
    for each M' in $\mathcal{M}'$
        for each M'' in $\mathcal{M}''$
            if no_neighbors (M', M'') then
                $D = (\mathcal{M} \cup (\star(M') \otimes \star(M'')), \mathcal{M}' - M', \mathcal{M}'' - M'')$
                if not $D \in \mathcal{D}$ then $\mathcal{D} = \mathcal{D} \cup \{D\}$
    for each M' in $\mathcal{M}'$
        $D = (\mathcal{M} \cup \star(M'), \mathcal{M}' - M', \mathcal{M}'')$
        if not $D \in \mathcal{D}$ then $\mathcal{D} = \mathcal{D} \cup \{D\}$
    for each M'' in $\mathcal{M}''$
        $D = (\mathcal{M} \cup \star(M''), \mathcal{M}', \mathcal{M}'' - M'')$
        if not $D \in \mathcal{D}$ then $\mathcal{D} = \mathcal{D} \cup \{D\}$
    mark as processed $(\mathcal{M}, \mathcal{M}', \mathcal{M}'')$
set $F[u] = \{\mathcal{M} \mid (\mathcal{M}, \langle\rangle, \langle\rangle) \in \mathcal{D}\}$

Theorem 3. *Given a graph $G \in NLC_k$ and a k-expression tree of G, the l-coloring problem is solvable in $\mathcal{O}(n^{3 \cdot 2^{k \lg(l+1)}} n^4)$ time.*

Proof. The complexity of the algorithm is obviously dominated by the processing of $\times_S$ nodes, so we get into details of the algorithm above. By Proposition 3, the size of the set $\mathcal{P}$ of all possible multisets of labels is equal $f(n,k,l) = (n+1)^{2^{k \lg(l+1)}}$. The main loop is evaluated at most once for every element of $\mathcal{P} \times \mathcal{P} \times \mathcal{P}$, thus at most $(f(n,k,l))^3$ times. Its content is dominated by the first loop, evaluated for every pair of labels in $\mathcal{M}' \times \mathcal{M}''$, thus at most $(n+1)^2$ times. The function 'no_neighbors', checking the condition (3) for every pair of distances in the processed labels, can be evaluated in k^4 time. Notice that, using appropriate data structures, one can check if $D = (\mathcal{M}, \mathcal{M}', \mathcal{M}'')$ belongs to $\mathcal{D}$ in $\mathcal{O}(|\mathcal{M}| + |\mathcal{M}'| + |\mathcal{M}''|) = \mathcal{O}(n)$ time. The expression tree T contains at most n nodes of type $\times_S$. Putting these together we get the result.

□

References

1. H. Bodlaender, T. Kloks, R. Tan, and J. van Leeuwen. λ-colorings of graphs. In *Proceedings 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2000)*, volume 1770 of *Lecture Notes in Computer Science*, pages 395–406. Springer-Verlag, 2000.
2. H. Bodlaender and D. Kratsch, 2002. Private communication.
3. D. Corneil and U. Rotics. On the relationship between clique-width and treewidth. In *Workshop on Graph-theoretic Concepts in Computer Science (WG 2001)*, volume 2204 of *Lecture Notes in Computer Science*, pages 78–90. Springer, 2001.
4. B. Courcelle and S. Olariu. Upper bounds to the clique-width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.
5. W. Espelage, F. Gurski, and E. Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Workshop on Graph-theoretic Concepts in Computer Science (WG 2001)*, volume 2204 of *Lecture Notes in Computer Science*, pages 117–128. Springer, 2001.
6. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completness*. Freeman, 1979.
7. M.C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *Int. J. Found. Comput. Sci.*, 11(3):423–443, 2000.

8. D. Kobler and U.Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003.
9. S. Oum and P. Seymour. Approximating clique-width and branch-width. Unpublished. See the web page `http://www.princeton.edu/ sangil/`.
10. I. Todinca. Coloring powers of graphs of bounded clique-width. In *Proceedings 29th Workshop on Graph-theoretic Concepts in Computer Science (WG 2003)*, volume 2880 of *Lecture Notes in Computer Science*, pages 370–382. Springer, 2003.
11. E. Wanke. k-nlc graphs and polynomial algorithms. *Discrete Applied Mathematics*, 54(2-3):251–266, 1994.
12. X. Zhou, Y. Kanari, and T. Nishizeki. Generalized vertex-colorings for partial k-trees. *IEICE Trans. Fundamentals*, E83-A(4):671 – 677, april 2000.