Graph Extremities Defined by Search Algorithms

Anne Berry, Jean Blair, Jean-Paul Bordat, Geneviève Simonet

Graph extremities defined by search algorithms

A. Berry * J. Blair † J.-P. Bordat ‡ G. Simonet‡

Abstract

Graph search algorithms have exploited and in some cases identified graph extremities, such as the leaves of a tree and the simplicial vertices of a chordal graph. Recently, several well-known graph search algorithms have been collectively expressed as two generic algorithms called MLS and MLSM, each of which instantiates with parameters defining the set of labels to be used, a partial order on labels used to choose at each step a vertex of maximal label and the way a label is initialized and incremented. In this paper, we investigate the properties of the vertex that is numbered 1 by MLS on a chordal graph and by MLSM on an arbitrary graph. We show that the minimal separators included in the neighborhood of this vertex are totally ordered by inclusion and that this vertex is indeed an extremity, as it belongs to a moplex of the graph. When the order on labels is total, this extremity property is still stronger since all the vertices of this moplex are numbered consecutively. When MLS is run on a non-chordal graph with a total order on labels, vertex number 1 is a weaker kind extremity, called an OCF-vertex.

1 Introduction

Various properties that identify a vertex as outermost or as an *extremity* of a graph have long been exploited in both graph theory and the design of efficient graph algorithms. The endpoints of a path, for example, are its two extremities; leaves are the extremities of a tree. Because this simple notion has proved very useful in dealing with trees, graph theorists have endeavored to extend it to wider graph classes.

For chordal graphs, extremities were defined as the *simplicial vertices* (a vertex is simplicial if its neighborhood is a clique), concurrently by Dirac [9] and by Lekkerkerker and Boland [19]. This concept led to efficient recognition algorithms for chordal graphs, based on the characterization of Fulkerson and Gross [14], who showed that a graph is chordal if and only if it has a *peo* (perfect elimination ordering). A peo is an ordering of the vertices given by the *simplicial elimination scheme*, which repeatedly finds a simplicial vertex and removes it from the graph. This was efficiently implemented by **Algorithm LexBFS**, introduced by Rose, Tarjan and Lueker [21], which finds a peo in a single linear-time pass if the input graph is chordal, numbering the vertices from n to 1. Thus LexBFS ends on a simplicial vertex. Tarjan and Yannakakis [22] later simplified Lex BFS into **Algorithm MCS**.

*LIMOS, Ensemble scientifique des Cézeaux, F-63177 Aubière, France. berry@isima.fr

†Dept. of EE & CS, United States Military Academy, West Point, NY 10996, USA. jean.blair@usma.edu

‡LIRMM, 161, Rue Ada, F-34392 Montpellier, France. bordat@lirmm.fr, simonet@lirmm.fr

Other forms of extremities have been defined for special classes of graphs. Golumbic and Goss [16] defined an edge-elimination for the subclass of weakly chordal graphs called chordal bipartite graphs. This was later extended by Hayward, Spinrad and Sritharan [17] to an elimination scheme involving extremities of weakly chordal graphs called co-pairs. Dahlhaus, Hammer, Maffray and Olariu [13] used MCS to find a domination elimination ordering on HHD-free graphs. On AT-free graphs, Corneil, Olariu and Stewart [11] defined dominating pairs of vertices, and used LexBFS to find such a pair efficiently [12], as the vertex numbered 1 by LexBFS belongs to a dominating pair, and a second pass of LexBFS will find a second such vertex.

On arbitrary graphs, work was done on computing a *minimal triangulation* of a graph (a chordal graph obtained from this graph by adding an inclusion-minimal set of edges). Ohtsuki, Cheung and Fujisawa [20] used what is called the *simplicial elimination game*, which simulates a simplicial elimination scheme by repeatedly choosing a vertex, adding every edge whose absence violates the simpliciality condition and removing it from the current graph. The graph obtained from the original graph G by adding all edges added in this process is chordal, and [20] showed that it is a minimal triangulation of G if and only if no other ordering of the vertices can produce by the simplicial elimination game a strictly inclusion-smaller set of added edges. They called such an ordering an *meo* (minimal elimination ordering) of G, and showed that an ordering is a meo if and only if at each step of the simplicial elimination game, a special vertex (which we call an *OCF-vertex*) is chosen. Thus OCF-vertices can be viewed as extremities of an arbitrary graph.

Algorithm LEX M, introduced in the same seminal paper as LexBFS [21] (LexBFS is a streamlined version of LEX M), finds an meo efficiently, so vertex number 1 of LEX M is always an OCF-vertex. LEX M was extended to **Algorithm MCS-M** by Berry, Blair, Heggernes and Peyton [1] using the same label simplification as MCS to compute an meo efficiently.

Berry and Bordat [2] defined a new kind of extremity by refining the notion of a simplicial vertex into that of a simplicial *moplex*. This strengthened the notion of simplicial vertex in a chordal graph, as any vertex of a simplicial moplex is simplicial, but some simplicial vertices may not belong to any simplicial moplex. In the same way, the notion of moplex strengthens the notion of OCF-vertex in an arbitrary graph, as any vertex of a moplex is an OCF-vertex, whereas some OCF-vertices may not belong to any moplex. Thus emerged the notions of *weak* and *strong extremities* of a graph: the weak extremities are OCF-vertices, which are exactly the simplicial vertices when the graph is chordal, and the strong extremities are the vertices belonging to a moplex, which are exactly the vertices belonging to a simplicial moplex when the graph is chordal. Blair and Peyton [8] studied Algorithm MCS on a chordal graph in relation with the clique tree representation of chordal graphs. Though not explicitly stated in [8], it follows from their work that MCS run on a chordal graph ends on a moplex, i.e. number the vertices of a moplex last with the smallest consecutive numbers. In an arbitrary graph, LEX M was shown by Berry and Bordat to end on a moplex [3], whereas MCS ends on an OCF vertex that does not necessarily belong to a moplex [1]. Another relevant result is that LexBFS always ends on a moplex, even on a non-chordal graph [2].

Corneil and Krueger [10] introduced **MNS** (Maximal Neighborhood Search), as an algorithm that encompasses both LexBFS and MCS, and showed that it also computes a peo if the graph is chordal. Recently, Berry, Krueger and Simonet [7] extended the family of search algorithms by defining a generic algorithm **MLS** (Maximal Label Search). Algorithm MLS has two input variables: a graph and a *labelling structure* containing in particular a set of labels and a partial order on

this set. We obtain instances of MLS such as LexBFS, MCS or MNS by choosing a specific labelling structure. [7] showed that the set of orderings of the vertices of a given graph computable by MLS (with all possible labelling structures) is equal to the set of orderings computable by MNS, which ensures that MLS finds a peo if the graph is chordal. Its corresponding extension **MLSM** finds an meo and generalizes both LEX M and MCS-M.

In this paper, we investigate the properties of the vertex numbered 1 by MLS and MLSM, in order to determine in what measure these more general graph searches preserve some of the strong extremal properties exhibited by MCS, LexBFS and LEX M. We show that indeed they do. Figure 1 summarizes the extremity properties of Algorithms MLS and MLSM. It presents known results described in this Section as well as new ones given in this paper.

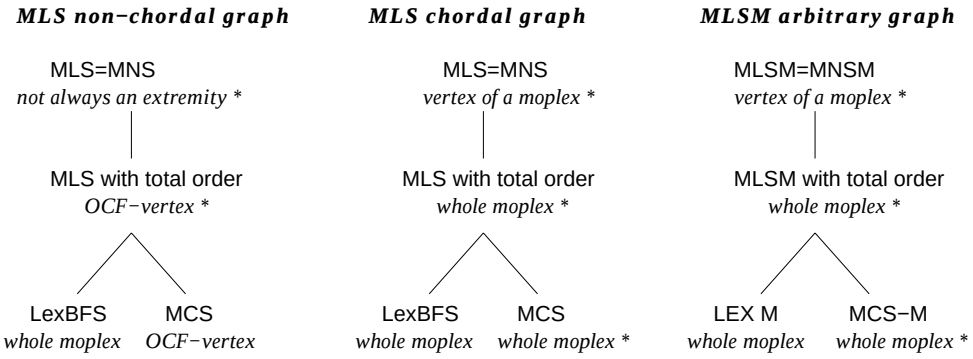

Figure 1: Extremity properties of Algorithms MLS and MLSM. These properties concern the vertex numbered 1 by the algorithms: it is an OCF-vertex (weak extremity), a vertex of a moplex (strong extremity) or a vertex of a moplex whose vertices are numbered consecutively, denoted by 'whole moplex'. The equality MLS = MNS (resp. MLSM = MNSM) means that both algorithms compute the same orderings on any graph, and therefore have the same extremity properties. 'with total order' means 'with total order on labels'. New results presented in this paper are indicated by *.

The paper is organized as follows: in Section 2 we give some notations and definitions, as well as some useful previous results. Section 3 describes the properties of the neighborhood of vertex 1. Section 4 describes the kind of graph extremities which vertex 1 belongs to for MLS and MLSM, and discusses the special properties of the related ordering.

2 Preliminaries

2.1 Basic definitions and notations

All graphs in this work are undirected and finite. The reader is referred to the classical work of [15] for any definitions not included in this section. The subgraph of G induced by the subset A of vertices is denoted $G(A)$. The *neighborhood* of a vertex x in G is denoted $N_G(x)$. The *closed neighborhood* is $N_G[x] = \{x\} \cup N_G(x)$. The neighborhood of a set of vertices A is $N_G(A) = \cup_{x \in A} N_G(x) \setminus A$, and $N_G[A] = A \cup N_G(A)$. We will omit subscript G when the graph we work on is clear from the context.

A *clique* is a set of pairwise adjacent vertices. A vertex or a set of vertices is *simplicial* if its neighborhood is a clique. A *module* is a subset X of vertices which share the same external neighborhood: $\forall x, y \in X, N(x) \setminus X = N(y) \setminus X$, so that a *clique module* is a subset X of vertices which share the same closed neighborhood: $\forall x, y \in X, N[x] = N[y]$.

An ordering α on the set V of vertices is a one-to-one mapping from $\{1, 2, ..., n\}$ to V. In every figure of this paper showing an ordering α on V, every vertex x will be named by its number $\alpha^{-1}(x)$. Be careful that all algorithms considered here number the vertices from n down to 1, so the first numbered vertex is $\alpha(n)$ and the last numbered one is $\alpha(1)$.

A *chordal* (or *triangulated*) graph is a graph with no chordless cycle of length greater or equal to 4. The notions of simplicial elimination scheme and game, minimal triangulation, peo and meo have been defined in Section 1. The graph obtained by running simplicial elimination game on graph G, choosing the vertices according to ordering α, is denoted G^+_α.

It remains to define the graph extremities we will work on, namely OCF-vertices and moplexes.

Definition 2.1 *A vertex x in $G = (V, E)$ is an* OCF-vertex *of G if, for any non-adjacent distinct neighbors y, z of x, there is a connected component C of $G(V \setminus N[x])$ such that y and z belong to $N(C)$.*

A subset S of vertices is a *minimal separator* iff there are at least two connected components of $G(V \setminus S)$ whose neighborhood in G is S (see [5] for extensive definitions on minimal separation).

Definition 2.2 *(Berry, Bordat [2]) A* moplex *of a graph G is a clique module of G whose neighborhood is a minimal separator.*

2.2 Algorithms MLS and MLSM

The definitions of Algorithms MLS and MLSM are based on the notion of *labelling structure*.

Definition 2.3 *A **labelling structure** is a structure $(L, \preceq, l_0, Inc)$, where:*

- *L is a set (the set of labels),*
- *$\preceq$ is a partial order on L (which may be total or not, $\prec$ denoting the corresponding strict order), which will be used to choose a vertex of maximal label,*
- *l_0 is an element of L which will be used to initialize the labels,*
- *Inc is a mapping from $L \times \{2, 3, ..., n\}$ to L, which will be used to increment a label, and such that for any integer i from 2 to n and any labels l and l' in L_i, the following properties hold:*

  (ls1) $l \prec Inc(l, i)$

  (ls2) if $l \prec l'$ then $Inc(l, i) \prec Inc(l', i)$

  where L_i is the subset of L defined by induction by:

  - $L_n = \{l_0\}$, and for any i from n down to 2

  - $L_{i-1} = L_i \cup \{l \in L \mid \exists l' \in L_i \mid l = Inc(l', i)\}$.

Algorithm MLSM (Maximal Label Search for Meo)

input : A graph $G = (V, E)$ and a labelling structure $\mathfrak{L} = (L, \preceq, l_0, Inc)$.

output : An meo α on V and a minimal triangulation $H = G^+_\alpha$ of G.

Initialize all labels as l_0; $E' \leftarrow E$; $G' \leftarrow G$;
for $i = n$ **downto** 1 **do**
  Choose a vertex x of G' of maximal label;
  $\alpha(i) \leftarrow x$;
  foreach *vertex* y *of* G' *different from* x **do**
    if *there is a path from* x *to* y *in* G' *such that every internal vertex on the path has a label strictly smaller than* $label(y)$ **then**
      $E' \leftarrow E' \cup \{xy\}$;
  foreach y *in* V *such that* $xy \in E'$ **do**
    $label(y) \leftarrow Inc(label(y), i)$;
  Remove x from G';
$H \leftarrow (V, E')$;

Figure 2: Algorithm MLSM.

Algorithm, MLSM, is given in Figure 2.

Algorithm MLS can be streamlined from MLSM, exactly as [21] streamlined LEX M into LexBFS, by removing the first *foreach loop*. Thus variables E' and H become useless ($E' = E$ and $H = G$) and the second *foreach loop* becomes: *foreach* neighbor y of x in G' *do*. MLS computes a peo of any chordal graph.

When using labelling structure $\mathfrak{L}$ with MLS (resp. MLSM), we will refer to the algorithm as $\mathfrak{L}$-MLS (resp. $\mathfrak{L}$-MLSM). LexBFS, MCS and MNS are instances of algorithm $\mathfrak{L}$-MLS and LEX M, MCS-M and MNSM (defined in [7] from MNS) are instances of algorithm $\mathfrak{L}$-MLSM, with the following respective labelling structure $\mathfrak{L} = (L, \preceq, l_0, Inc)$:

LexBFS and LEX M: L is the set of lists of elements of $\{2, 3, ..., n\}$ in strictly decreasing order, $\preceq$ is lexicographical order (a total order), l_0 is the empty list, $Inc(l, i)$ is obtained from l by adding i to the end of the list, provided that i is strictly smaller than the last integer in l.

MCS and MCS-M: $L = \{0, 1, 2, ..., n-1\}$, $\preceq$ is $\leq$ (a total order), $l_0 = 0$, $Inc(l, i) = l + 1$.

MNS and MNSM: L is the power set of $\{2, 3, ..., n\}$, $\preceq$ is $\subseteq$ (not a total order), $l_0 = \emptyset$, $Inc(l, i) = l \cup \{i\}$.

We call MLS (resp. MLSM, $\mathfrak{L}$-MLS, $\mathfrak{L}$-MLSM) ordering of a graph $G = (V, E)$ any ordering on V that can be computed by the given algorithm. The reader can easily convince himself of the following Property proved in [7].

Property 2.4 *([7]) For any graph G and any labelling structure $\mathfrak{L}$, Algorithms $\mathfrak{L}$-MLSM computing ordering α on G has the same behavior as $\mathfrak{L}$-MLS on G^+_α (they give the same labels and numbers to vertices, provided that they break ties in the same way).*

Property 2.4 has two consequences that will be largely used in this paper.

1. Any $\mathfrak{L}$-MLSM ordering α of G is a $\mathfrak{L}$-MLS ordering of G^+_α. Thus a property of MLS on chordal graphs can in some cases be extended to a property of

MLSM on arbitrary graphs since G^+_α is chordal (Corollaries 3.10 and 3.11 and Theorem 4.5).

2. MLS and MLSM have the same behavior on a chordal graph, since in that case, $G^+_\alpha = G$. Any execution of MLS on a chordal graph can be seen as an execution of MLSM, and conversely. Thus any property of MLSM on arbitrary graphs also holds for MLS on chordal graphs (Corollary 4.3).

3 The neighborhood of vertex 1

3.1 Super-components and slices

In this Section, we will study some properties of the neighborhood of the vertex numbered 1 by MLS or MLSM, which will lead to extremity properties in Section 4.

Notations 3.1 *For any graph $G = (V, E)$ and any ordering α on V,*
- a component of (G, α) is a connected component of $G(V \setminus N(\alpha(1)))$,
- a super-component of (G, α) is the union of all components of (G, α) having the same neighborhood as some component of (G, α),
- p denotes the number of super-components of (G, α) and $(D_1, D_2, ..., D_p)$ denotes the sequence of the super-components of (G, α) ordered as follows: $\forall i \in [1, p-1]$, $\alpha^{-1}(d_i) > \alpha^{-1}(d_{i+1})$, where d_i denotes the vertex of D_i with maximum number $(\alpha^{-1}(d_i) = max\{\alpha^{-1}(d), d \in D_i\})$,
- the slices of (G, α) are the subsets T_1, T_2, ..., T_p of V partitionning V as follows : $\forall i \in [1, p]$, $T_i = D_i \cup S_i$, where $S_i = N(D_i) \setminus (\cup_{1 \leq j < i} N(D_j)) = N(D_i) \setminus (\cup_{1 \leq j < i} S_j)$, and t_i denotes the vertex of T_i with maximum number.
We say that α numbers the slices in increasing order if $\forall i \in [1, p-1]$, $\forall t \in T_i$, $\alpha^{-1}(t) > \alpha^{-1}(t_{i+1})$.
The partition of V in slices can be refined into its partition in thin slices, where the thin slices of (G, α) are defined from the components of (G, α) in the same way as the slices are defined from the super-components of (G, α).

Example 3.2 *Figure 3 shows the slices defined by an execution of MCS on a chordal graph. $N(\alpha(1)) = \{8, 3, 2\}$ and the components of (G, α) are $\{7, 5\}$, $\{6\}$, $\{4\}$ and $\{1\}$, with $N(\{7, 5\}) = N(\{6\}) = \{8\}$, $N(\{4\}) = \{8, 3\}$ and $N(\{1\}) = \{8, 3, 2\}$, so $p = 3$ and*
- $D_1 = \{7, 5\} \cup \{6\} = \{7, 6, 5\}$, $d_1 = 7$, $S_1 = \{8\}$, $T_1 = \{8, 7, 6, 5\}$ and $t_1 = 8$,
- $D_2 = \{4\}$, $d_2 = 4$, $S_2 = \{3\}$, $T_2 = \{4, 3\}$ and $t_2 = 4$,
- $D_3 = \{1\}$, $d_3 = 1$, $S_3 = \{2\}$, $T_3 = \{2, 1\}$ and $t_3 = 2$.
We observe that:
- $N(D_1) \subset N(D_2) \subset N(D_3)$,
- α numbers the slices in increasing order (but not the thin slices, since slice T_1 is partitionned into the thin slices $T'_1 = \{8, 7, 5\}$ and $T'_2 = \{6\}$).
We will see how these properties can be generalized to MLS and MLSM orderings.

We will use the following notations and Lemmas, which are proved in the Appendix.

Notations 3.3 *For any execution of MLS on any graph G computing ordering α and any vertices x and y of G such that $\alpha^{-1}(x) \geq \alpha^{-1}(y)$,*
$l_x(y)$ denotes the label of y just before numbering x,
$NN_x(y)$ denotes the set of Numbered Neighbors of y just before numbering x, i.e.

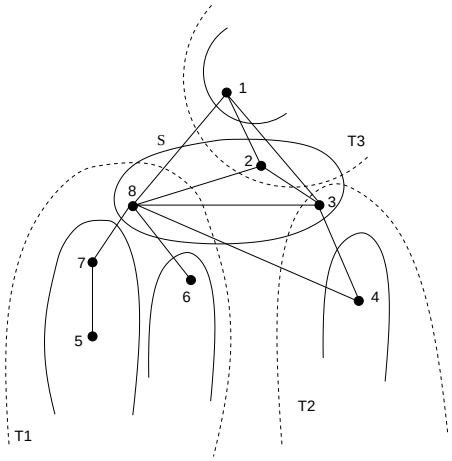

Figure 3: The slices defined by an execution of MCS on a chordal graph.

$NN_x(y) = \{z \in N(y) \mid \alpha^{-1}(z) > \alpha^{-1}(x)\}$.

Lemma 3.4 *For any MLS ordering α of a chordal graph, $\forall i \in [1..p]$, $NN_{d_i}(d_i) = NN_{d_i}(\alpha(1)) \subseteq N(D_i)$, $NN_{t_i}(t_i) = NN_{t_i}(\alpha(1))$, and if $i < p$ then $\alpha^{-1}(t_i) \geq \alpha^{-1}(d_i) > \alpha^{-1}(t_{i+1})$.*

Lemma 3.5 *For any MLS ordering α of a chordal graph, $\forall i \in [1..p-1], \forall s \in S_i$, $\alpha^{-1}(s) > \alpha^{-1}(t_{i+1})$.*

Lemma 3.6 *For any MLS ordering α of a chordal graph, if the order on labels is total then $\forall i \in [1..p-1]$, $\forall d \in D_i$, $\alpha^{-1}(d) > \alpha^{-1}(t_{i+1})$.*

We have the following Property.

Property 3.7 *For any chordal graph G and any MLS ordering α of G $\forall i \in [1, p-1]$, $N(D_i) \subset N(D_{i+1})$.*

Proof: Let G be a chordal graph, let α be a MLS ordering of G and $i \in [1, p-1]$. Let us show that $N(D_i) \subset N(D_{i+1})$.
Let $s \in N(D_i)$, and let $j \leq i$ such that $s \in S_j$. By Lemmas 3.4 and 3.5, $\alpha^{-1}(s) > \alpha^{-1}(t_{j+1}) \geq \alpha^{-1}(t_{i+1}) \geq \alpha^{-1}(d_{i+1})$. So $s \in NN_{d_{i+1}}(\alpha(1))$ and by Lemma 3.4 $s \in N(D_{i+1})$. Thus $N(D_i) \subseteq N(D_{i+1})$ and, as $N(D_i) \neq N(D_{i+1})$ by definition of the sequence (D_i), $N(D_i) \subset N(D_{i+1})$. □

For any graph G and any ordering α on V, the minimal separators of G included in $N(\alpha(1))$ are exactly the neighborhoods of the components of (G, α) different from $\{\alpha(1)\}$ ([4]), which are also the neighborhoods of the super-components of (G, α) different from $\{\alpha(1)\}$. So we have the following result.

Theorem 3.8 *For any chordal graph G and any MLS ordering α of G, the minimal separators of G included in $N(\alpha(1))$ are totally ordered by inclusion.*

The following Theorem immediately follows from Lemmas 3.5 and 3.6.

Theorem 3.9 *For any chordal graph G and any labelling structure $\mathcal{L}$, if the order on labels is total then any $\mathcal{L}$-MLS ordering of G numbers the slices in increasing order.*

By Property 2.4 any $\mathcal{L}$-MLSM ordering α of any graph G is a $\mathcal{L}$-MLS ordering of G^+_α, which verifies the preceding neighborhood properties since G^+_α is chordal. Moreover, the neighborhood of $\alpha(1)$ is the same in G and G^+_α (by definition of G^+_α) and since α is a meo of G, the components of (G, α) and of (G^+_α, α) are the same with the same neighborhoods (this follows for instance from Theorem 5.12, Invariant 4.9 and Lemma 6.2 in [5]). Thus the super-components, their neighborhoods and the slices of (G, α) and of (G^+_α, α) are the same, and Theorems 3.8 and 3.9 extend to MLSM on an arbitrary graph.

Corollary 3.10 *For any graph G and any MLSM ordering α of G, the minimal separators of G included in $N(\alpha(1))$ are totally ordered by inclusion.*

Corollary 3.11 *For any graph G and any labelling structure $\mathcal{L}$, if the order on labels is total then any $\mathcal{L}$-MLSM ordering of G numbers the slices in increasing order.*

If the order on labels is *not* total, an $\mathcal{L}$-MLS ordering α does not necessarily number slices in increasing order. Figure 4 gives a counterexample for this.

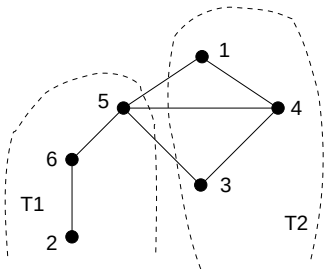

Figure 4: This MNS ordering does not number slices in increasing order on this chordal graph.

3.2 MLS on a non-chordal graph

We will end this section by a few remarks on running MLS on a non-chordal graph. [2] showed Theorems 3.8 and 3.9 for an execution of LexBFS on an arbitrary graph, with the even stronger property that it numbers the thin slices in increasing order instead of the slices.

Property 3.12 *([2]) For any graph G (chordal or not) and any LexBFS ordering α of G, the minimal separators of G included in $N(\alpha(1))$ are totally ordered by inclusion, and α numbers the thin slices in increasing order.*

Upon investigation, it turns out that in a non-chordal graph, MLS does *not* in general exhibit these properties, even when the order $\preceq$ is total. Figure 5 gives a counter-example of this.

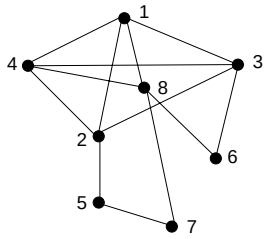

Figure 5: MCS on a non-chordal graph. The super-components of (G, α) (which are also its components), are: $D_1 = \{5, 7\}$, $D_2 = \{6\}$ and $D_3 = \{1\}$; $N(D_1) = \{2, 8\}$, $N(D_2) = \{3, 8\}$ and $N(D_3) = \{2, 3, 4, 8\}$. The neighborhoods are not pairwise inclusive, α does not number the slices in increasing order, vertex $\alpha(1)$ does not belong to a moplex of G. However, $\alpha(1)$ is an OCF-vertex of G.

4 Extremities

4.1 The extremities defined by MLS and MLSM

For any MLSM ordering α of a graph G, since α is an meo of G, by [20] $\alpha(1)$ is an OCF-vertex of G (weak extremity). To show that $\alpha(1)$ is a strong extremity, i.e. a vertex of a moplex of G, we need only use the properties of the neighborhood of $\alpha(1)$ described in Section 3 and the following Lemma proved in the Appendix.

Lemma 4.1 *For any graph G and any vertex x of G, x is a vertex of a moplex of G iff x is an OCF-vertex of G and the set of minimal separators of G included in $N(x)$ has a greatest element S_{max} for inclusion.*
In that case, the moplex containing x is $N[x] \setminus S_{max}$.

We will also show that if moreover the order on labels is total then the vertices of the moplex X containing $\alpha(1)$ are numbered consecutively, i.e. $X = \{\alpha(1), ...\alpha(|X|)\}$. We say that the algorithm ends on a moplex.

Theorem 4.2 *For any non-clique graph G and any labelling structure $\mathcal{L}$, $\mathcal{L}$-MLSM on input graph G ends on a vertex of a moplex. If moreover the order on labels is total then it ends on a moplex.*

Proof: Since α is a meo of G $\alpha(1)$ is an OCF-vertex of G [20]. Moreover, by Corollary 3.10 the set of minimal separators of G included in $N(\alpha(1))$ has a greatest element S_{max} for inclusion. If $D_p \neq \{\alpha(1)\}$ then $S_{max} = N(\alpha(1))$ else $S_{max} = N(D_{p-1})$ ($p > 1$ because otherwise G would be reduced to the closed neighborhood of OCF-vertex $\alpha(1)$ and would be a clique). It follows by Lemma 4.1 that $\alpha(1)$ is a vertex of a moplex X of G, with $X = N[\alpha(1)] \setminus S_{max}$. So X is either equal to $\{\alpha(1)\}$ or to D_p. Thus if the order on labels is total then by Corollary 3.11 $\mathcal{L}$-MLSM ends on X. □

As by Property 2.4, any chordal graph has the same $\mathcal{L}$-MLS and $\mathcal{L}$-MLSM orderings, we have the following Corollary.

Corollary 4.3 *For any non-clique chordal graph G and any labelling structure $\mathcal{L}$, $\mathcal{L}$-MLS on input graph G ends on a vertex of a moplex. If moreover the order on labels is total then it ends on a moplex.*

4.2 The orderings defined by MLS and MLSM

MLS defines a peo of any chordal graph, and MLSM defines an meo of any graph. We can strengthen these properties to *moplex orderings*, introduced by Berry and Bordat [3]. We define a moplex elimination scheme in a chordal graph by repeatedly choosing a moplex of the current graph, which is simplicial since this graph is chordal, and removing it from the current graph, until this graph is a clique. If the graph fails to be chordal, we define moplex elimination game in the same way as we defined simplicial elimination game: we simulate a moplex elimination scheme by repeatedly choosing a moplex of the current graph, adding every edge whose absence violates the simpliciality of this moplex, and removing it from the current graph, until this graph is a clique. Thus we obtain an ordered partition $(X_1, X_2, ..., X_k)$ where X_i is a moplex of the current graph at step i if $i < k$ and X_k is the final clique. A *moplex ordering* is an ordering α on V that is compatible with such an ordered partition, i.e. such that $\forall i \in [1, k-1]$, $\forall x \in X_i$, $\forall x' \in X_{i+1}$, $\alpha^{-1}(x) < \alpha^{-1}(x')$. Then G^+_α is exactly the graph obtained from G by adding all edges added in the moplex elimination game.

Berry and Bordat [3] showed that any moplex ordering of any graph is a meo of this graph. They also showed that any LexBFS ordering of a chordal graph, respectively any LEX M ordering of any graph, is a moplex ordering. We extend these results to MLS and MLSM. Following Property 4.4 easily follows from Corollary 4.3, and Theorem 4.5 follows from Property 2.4 and 4.4 and from some known results about minimal triangulation. Their proofs are given in the Appendix.

Property 4.4 *For any chordal graph G and any labelling structure $\mathcal{L}$, if the order on labels is total then any $\mathcal{L}$-MLS ordering of G is a moplex ordering of G.*

Theorem 4.5 *For any graph G and any labelling structure $\mathcal{L}$, if the order on labels is total then any $\mathcal{L}$-MLSM ordering of G is a moplex ordering of G.*

4.3 MLS on a non-chordal graph

Again, we will discuss the properties of MLS when run on a non-chordal graph. [2] showed that even on a non-chordal graph, LexBFS always ends on a moplex. In general, an MLS execution does not end on a moplex and does not even end on a vertex of a moplex, even if the order on labels is total. However, we show that if the order on labels is total then MLS ends on a weak extremity.

Theorem 4.6 *For any graph G, any labelling structure $\mathcal{L}$ and any $\mathcal{L}$-MLS ordering α of G, if the order on labels is total then $\alpha(1)$ is an OCF-vertex of G.*

This is the case in particular for MCS (as shown in [1]). The proof of Theorem 4.6 is given in the Appendix.

If the order on labels is *not* total then MLS does not necessarily end on an OCF-vertex. For instance, on the graph shown in Figure 6, $\alpha = (1, 2, 3, 4, 5, 6)$ is an MNS ordering of G and 1 is not an OCF-vertex of G since 2 and 4 are non-adjacent and 4 does not belong to the neighborhood of the unique connected component $\{3, 5\}$ of $G(V \setminus N[1])$.

4.4 The MLS-Terminal Vertex Problem on chordal graphs

We define the MLS-Terminal Vertex Problem as follows: given a graph G and a vertex x of G, is x MLS-terminal, i.e. is there a MLS execution which gives x the

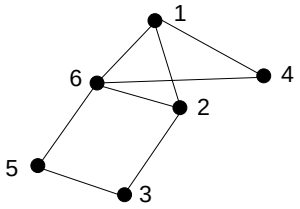

Figure 6: MNS on a non-chordal graph does not end on an OCF-vertex

number 1? We define the $\mathcal{L}$-MLS-Terminal Vertex Problem in the same way for any labelling structure $\mathcal{L}$.

Lanlignel [18] showed that LexBFS-Terminal Vertex Problem is NP-complete even in the restrictive classes of weakly chordal graphs and HDP (Heriditary Dominating Path) graphs, and left open its complexity on a chordal graph.

We will give a simple characterization of MNS-terminal vertices in a chordal graph.

Characterization 4.7 *For any chordal graph G and any vertex x of G, x is MNS-terminal iff x is simplicial and the neighborhoods of the connected components of $G(V \setminus N[x])$ are totally ordered by inclusion.*

The MLS orderings of a graph are exactly its MNS orderings [7], so its MLS-terminal vertices are exactly its MNS-terminal vertices and we have the following Corollary.

Corollary 4.8 *MNS-Terminal (resp. MLS-Terminal) Vertex Problem is polynomial in the class of chordal graphs.*

5 Conclusion

We have shown that Algorithms MLS on a chordal graph and MLSM on an arbitrary graph define as number 1 vertices which are well-defined as extremities of the graph, with special properties.

When run on a non-chordal graph, the properties of MLS are weaker, except for Algorithm LexBFS, which stands out as having properties in many ways similar to the meo-computing algorithms such as LEX M, MCS-M, and more generally MLSM. In view of this, perhaps it would be interesting to investigate experimentally whether the ordering α produced by an execution of LexBFS on a non-chordal graph defines a triangulation G^+_α which is close to minimal. MLS run on a non-chordal graph with a non-total order on labels does not always end on a weak extremity of the graph. It is an open question whether it ends in that case on a still weaker kind of extremity.

We also leave open the complexity of LexBFS-Terminal Vertex Problem on chordal graphs, as well as the complexity of MLS-Terminal Vertex Problem on arbitrary graphs.

References

[1] A. Berry, J. Blair, P. Heggernes and B. Peyton. Maximum Cardinality Search for computing minimal triangulations of graphs. *Algorithmica*, 2003.

[2] A. Berry and J.-P. Bordat. Separability generalizes Dirac's theorem. *Discrete Applied Mathematics*, 84:43–53, 1998.

[3] A. Berry and J.-P. Bordat. Moplex elimination orderings. *Electronic Notes in Discrete Mathematics* , Volume 8, 2001, Proceedings of First Cologne-Twente Workshop on Graphs and Combinatorial Optimization.

[4] A. Berry, J.-P. Bordat, and P. Heggernes. Recognizing weakly chordal graphs by edge separability. *Nordic Journal Computing*, 7:3 (2000), pp. 164–177.

[5] A. Berry, J-P. Bordat, P. Heggernes, G. Simonet, and Y. Villanger. A wide-range algorithm for minimal triangulation from an arbitrary ordering. *Journal of Algorithms*, to appear (in press).

[6] A. Berry, P. Heggernes and Yngve Villanger. A vertex incremental approach for dynamically maintaining chordal graphs. *Proceedings ISAAC 2003 - 14th International Symposium on Algorithms and Computation, Kyoto, Japan, December 2003.* Springer Verlag, Lecture Notes in Computer Science 2906, pages 47 - 57.

[7] A. Berry, R. Krueger, and G. Simonet. Ultimate generalizations of LexBFS and LEX M. Research Report ***, submitted.

[8] J. R. S. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In J. A. George, J. R. Gilbert, and J. W. H. Liu, editors, *Graph Theory and Sparse Matrix Computations*, pages 1–30. Springer Verlag, 1993. IMA Volumes in Mathematics and its Applications, Vol. 56.

[9] G. A. Dirac. On rigid circuit graphs. *PAbh. Math. Sem. Univ. Hamburg*, 25 (1961), pp. 71–76.

[10] D. Corneil and R. Krueger. A unified view of graph searching. Submitted.

[11] D. G Corneil, S. Olariu, and L. Stewart. Asteroidal triple free graphs. *SIAM Journal Discrete Mathematics*, 10:399–430, 1997.

[12] D.G Corneil, S. Olariu, and L. Stewart. Linear time algorithms for dominating pairs in asteroidal triple–free graphs. *Proceedings of 22nd ICALP Conference, LNCS 944 (Springer)*, pages 292–302, 1995.

[13] E. Dahlhaus, P. L. Hammer, F. Maffray, and S. Olariu. On Domination Elimination Orderings and Domination Graphs. *Proceedings of WG 1994*, 81-92, 1994.

[14] D.R. Fulkerson and O.A. Gross. Incidence matrixes and interval graphs. *Pacific Journal of Math.*, 15:835–855, 1965.

[15] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[16] Golumbic and Goss. V ***

[17] R. Hayward, J. Spinrad, and R. Sritharan. Weakly chordal graph algorithms via handles. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2000).*

[18] J-M. Lanlignel. Autour de la decomposition en coupes. PhD Thesis, LIRMM, Montpellier, France, 2001.

[19] C. Lekkerkerker and J. Boland. Representation of a finite graph by a set of intervals on real line. *Fund. Math.*, 51 (1962), pp. 45–64.

[20] T. Ohtsuki, L. K. Cheung, and T. Fujisawa. Minimal triangulation of a graph and optimal pivoting ordering in a sparse matrix. *J. Math. Anal. Appl.*, 54 (1976), pp. 622–633.

[21] D. Rose, R.E. Tarjan, and G. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journ. Comput*, 5:146–160, 1976.

[22] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13:566–579, 1984.

Appendix

The subscript in notations $l_x(y)$ and $NN_x(y)$ will be omitted when the considered step is clear from the context.

- We give the proofs of the Lemmas of Section 3 and of some other useful ones.

Lemma 5.1 *At any step of an execution of MLS on a graph G, for any unnumbered vertices u, v of G,*

(i) if $NN(u) = NN(v)$ then $l(u) = l(v)$,

(ii) if $NN(u) \subset NN(v)$ then $l(u) \prec l(v)$.

Proof: This Lemma is proved in [7]. □

Lemma 3.4 For any MLS ordering α of a chordal graph, $\forall i \in [1..p]$, $NN_{d_i}(d_i) = NN_{d_i}(\alpha(1)) \subseteq N(D_i)$, $NN_{t_i}(t_i) = NN_{t_i}(\alpha(1))$, and if $i < p$ then $\alpha^{-1}(t_i) \geq \alpha^{-1}(d_i) > \alpha^{-1}(t_{i+1})$. **Proof:** $NN_{d_i}(d_i) \subseteq NN_{d_i}(\alpha(1))$

by definition of d_i and $NN_{d_i}(d_i) \not\subset NN_{d_i}(\alpha(1))$ (otherwise by Lemma 5.1 $l_{d_i}(d_i) \prec l_{d_i}(\alpha(1))$, which contradicts the choice of d_i at that step). Hence $NN_{d_i}(d_i) = NN_{d_i}(\alpha(1)) \subseteq N[D_i] \cap N(\alpha(1)) = N(D_i)$.
We suppose that $i < p$. By definition of d_i and t_i, $\alpha^{-1}(t_i) \geq \alpha^{-1}(d_i)$. Let us show that $\alpha^{-1}(d_i) > \alpha^{-1}(t_{i+1})$. If $t_{i+1} \notin N(\alpha(1))$ then $t_{i+1} = d_{i+1}$ and we are done since by definition of d_i, $\alpha^{-1}(d_i) > \alpha^{-1}(d_{i+1})$. Otherwise, assume for contradiction that $\alpha^{-1}(t_{i+1}) > \alpha^{-1}(d_i)$. Then $t_{i+1} \in NN_{d_i}(\alpha(1))$, with $NN_{d_i}(\alpha(1)) \subseteq N(D_i) \subseteq V \setminus T_{i+1}$, so $t_{i+1} \notin T_{i+1}$, a contradiction.
Let us show that $NN_{t_i}(t_i) = NN_{t_i}(\alpha(1))$. Since $\forall j \geq i$, $\forall t \in T_j$, $\alpha^{-1}(t_i) \geq \alpha^{-1}(t_j) \geq \alpha^{-1}(t)$, $NN_{t_i}(t_i) \subseteq N(t_i) \cap (\cup_{j<i} T_j)$. Since $\forall j < i$, $N(t_i) \cap D_j = \emptyset$ (because $T_i \cap N(D_j) = \emptyset$) $NN_{t_i}(t_i) \subseteq \cup_{j<i} S_j \subseteq N(\alpha(1))$. So $NN_{t_i}(t_i) \subseteq NN_{t_i}(\alpha(1))$, and we show the equality as above for d_i. □

Lemma 5.2 *If G is a chordal graph, μ a chordless path in G and z a vertex of G not belonging to μ and seeing both extremities of μ, then z sees every vertex of μ.*

Proof: It is a well-known property of chordal graphs which follows for instance from results from [6]. □

Lemma 5.3 *Let G be a chordal graph, let α be a MLS ordering of G, let $i \in [1..p]$, $s \in S_i$ such that $\alpha^{-1}(d_i) > \alpha^{-1}(s)$, and let μ be a chordless $d_i s$-path such that every internal vertex of μ belongs to D_i. Then the following property P holds just after processing d_i and remains true until numbering s.*
P : there is a vertex v of μ such that every vertex of $\mu[v, s]$ is unnumbered, $NN(\alpha(1)) \subset NN(v)$ and $\forall t \in \mu[v, s] \setminus \{v\}$, $NN(t) = NN(\alpha(1))$.

Proof: Let us show that P holds just after processing d_i. $NN_{d_i}(d_i) = NN_{d_i}(\alpha(1))$ by Lemma 3.4, so $NN_{d_i}(\alpha(1)) \subseteq N(d_i)$. Also $NN_{d_i}(\alpha(1)) \subseteq N(s)$ since, as α is a peo of G, $N(\alpha(1))$ is a clique of G. It follows by

Lemma 5.2 that for any vertex t of μ, $NN_{d_i}(\alpha(1)) \subseteq N(t)$, so $NN_{d_i}(\alpha(1)) \subseteq NN_{d_i}(t)$, and as the reverse inclusion holds by definition of d_i and S_i, $NN_{d_i}(t) = NN_{d_i}(\alpha(1))$. After processing d_i, $NN(\alpha(1))$ is unchanged and P is true, taking as vertex v the vertex of μ next to d_i (which is the only vertex of μ seeing d_i since μ is chordless).
Let us show that P is preserved when processing some vertex w, $\alpha^{-1}(d_i) > \alpha^{-1}(w) > \alpha^{-1}(s)$.
First case: $w \notin N(\alpha(1))$. If no vertex of $\mu[v,s]$ sees w then $w \notin \mu[v,s]$ and P remains true with the same vertex v, otherwise P remains true, taking as new vertex v the last vertex of $\mu[v,s]$ seeing w.
Second case: $w \in N(\alpha(1))$. ws is an edge of G, as $N(\alpha(1))$ is a clique. We get from edge ws and path $\mu[v,s]$ a wv-path such that every internal vertex t is unnumbered and verifies $NN(t) = NN(\alpha(1)) \subset NN(v)$, so by Lemma 5.1 $l(t) \prec l(v)$. By Property 2.4, this execution of MLS on chordal graph G can be seen as an execution of MLSM, according to which wv will be an edge of H with $H = G^+_\alpha = G$. It follows that w sees v in G. By Lemma 5.2 w sees every vertex of $\mu[v,s]$. It follows that when processing w, w is added to $NN(\alpha(1))$ as well as to $NN(t)$ for every vertex t of $\mu[v,s]$, and P remains true with the same vertex v. □

Lemma 3.5 For any MLS ordering α of a chordal graph, $\forall i \in [1..p-1], \forall s \in S_i$, $\alpha^{-1}(s) > \alpha^{-1}(t_{i+1})$.

Proof: Suppose there is some $s \in S_i$ such that $\alpha^{-1}(t_{i+1}) > \alpha^{-1}(s)$. Then by Lemma 3.4 $\alpha^{-1}(d_i) > \alpha^{-1}(t_{i+1}) > \alpha^{-1}(s)$. Lemma 5.3 ensures the existence of some vertex v such that $NN_{t_{i+1}}(\alpha(1)) \subset NN_{t_{i+1}}(v)$, so by Lemma 3.4 $NN_{t_{i+1}}(t_{i+1}) \subset NN_{t_{i+1}}(v)$ and by Lemma 5.1 $l_{t_{i+1}}(t_{i+1}) \prec l_{t_{i+1}}(v)$, which contradicts the choice of t_{i+1} at that step. □

Lemma 3.6 For any MLS ordering α of a chordal graph, if the order on labels is total then $\forall i \in [1..p-1]$, $\forall d \in D_i$, $\alpha^{-1}(d) > \alpha^{-1}(t_{i+1})$.

Proof: Suppose there is some $d \in D_i$ such that $\alpha^{-1}(t_{i+1}) > \alpha^{-1}(d)$. We can choose d as the first vertex of D_i numbered after t_{i+1}. Just before processing t_{i+1}, $l(t_{i+1})$ is maximum (since the order $\preceq$ is total) with $l_{t_{i+1}}(t_{i+1}) = l_{t_{i+1}}(\alpha(1))$ by Lemmas 3.4 and 5.1, so $l_{t_{i+1}}(d) \preceq l_{t_{i+1}}(\alpha(1))$. As by Lemma 3.5, every vertex of $N(D_i)$ is numbered before t_{i+1}, $l(d)$ remains equal to $l_{t_{i+1}}(d)$ until d is numbered. We thus only have to prove that $l(\alpha(1))$ is incremented before numbering d. This will lead to a contradiction, since $\alpha(1)$ is numbered last.

If $t_{i+1} \in N(\alpha(1))$, $l(\alpha(1))$ is incremented when numbering t_{i+1} and we are done.

If $t_{i+1} \notin N(\alpha(1))$, $t_{i+1} = d_{i+1}$. By Property 3.7, S_{i+1} is not empty, so let $s \in S_{i+1}$. $\alpha^{-1}(d_{i+1}) > \alpha^{-1}(s)$, and by Lemma 5.3, until numbering s, there is some unnumbered vertex v such that $NN(\alpha(1)) \subset NN(v)$, so by Lemma 5.1 $l(\alpha(1)) \prec l(v)$, hence $l(d) \prec l(v)$. This implies that $\alpha^{-1}(s) > \alpha^{-1}(d)$ and as $l(\alpha(1))$ is incremented when processing s, it is incremented before numbering d. □

- Here is the proof of Lemma 4.1:

Lemma 4.1 For any graph G and any vertex x of G, x is a vertex of a moplex of G iff x is an OCF-vertex of G and the set of minimal separators of G included in $N(x)$ has a greatest element S_{max} for inclusion.
In that case, the moplex containing x is $N[x] \setminus S_{max}$.

Proof: We suppose that x is a vertex of a moplex X of G. Then $N[x] = N[X]$ and there is a connected component C of $G(V \setminus N[x])$ such that $N(C) = N(X)$. Since X is a clique module of G, any non-adjacent neighbors of x must belong to $N(X)$ i.e. to $N(C)$, so x is an OCF-vertex of G, and the neighborhood of any connected component of $G(V \setminus N[x])$ is a subset of $N(X)$ i.e. of $N(C)$, so $N(C)$ is the greatest minimal separator of G included in $N(x)$ for inclusion.
Conversely, we suppose that x is an OCF-vertex of G and the set of minimal separators of G included in $N(x)$ has a greatest element S_{max} for inclusion. Let $X = N[x] \setminus S_{max}$. We immediately verify that X is a clique module of G, with $N(X) = S_{max}$. Let C be the connected component of $G(V \setminus N[x])$ such that $S_{max} = N(C)$. X and C are distinct connected components of $G(V \setminus N(X))$ such that $N(X) = N(C)$, so $N(X)$ is a minimal separator of G and therefore X is a moplex of G. □

- We give the proofs of Property 4.4 and Theorem 4.5:

Property 4.4 For any chordal graph G and any labelling structure $\mathfrak{L}$, if the order on labels is total then any $\mathfrak{L}$-MLS ordering of G is a moplex ordering of G.

Proof: We have only to point out that by Corollary 4.3 any $\mathfrak{L}$-MLS ordering α of a non-clique chordal graph G ends on a moplex X of G and that the restriction of α to the chordal subgraph $G(V \setminus X)$ is itself a $\mathfrak{L}$-MLS ordering of this subgraph, and the proof follows by induction on $|V|$. □

Theorem 4.5 For any graph G and any labelling structure $\mathfrak{L}$, if the order on labels is total then any $\mathfrak{L}$-MLSM ordering of G is a moplex ordering of G.

Proof: Let α be a $\mathfrak{L}$-MLSM ordering of G. For any k from 1 to n, let G_α^k denote the current graph of the simplicial elimination game just before processing $\alpha(k)$, let V_α^k be its set of vertices, i.e. $V_\alpha^k = \{\alpha(k), ..., \alpha(n)\}$, and α_k be the restriction of α to V_α^k. To prove that α is a moplex ordering of G, it is sufficient to prove that for any k from 1 to n such that G_α^k is not a clique, α_k ends on a moplex of G_α^k. Let k, $1 \leq k \leq n$, such that G_α^k is not a clique and let us show that α_k ends on a moplex of G_α^k.
By Property 2.4 α is a $\mathfrak{L}$-MLS ordering of G_α^+, so α_k is a $\mathfrak{L}$-MLS ordering of $G_\alpha^+(V_\alpha^k)$, which is equal to $(G_\alpha^k)_{\alpha_k}^+$. As α is a meo of G, its restriction α_k is a meo of G_α^k (otherwise there would exist another ordering β_k on V_α^k that produces some better fill-in than α_k, and therefore an ordering β on V that produces some better fill-in than α, a contradiction). So, $(G_\alpha^k)_{\alpha_k}^+$ is a minimal triangulation of G_α^k and therefore, as G_α^k is not a clique, $(G_\alpha^k)_{\alpha_k}^+$ is not a clique either. It follows from Corollary 4.3 that α_k ends on a moplex of $(G_\alpha^k)_{\alpha_k}^+$, which is also a moplex of G_α^k since any moplex of a minimal triangulation of a graph is a moplex of this graph [2]. □

- We now present the proof of Theorem 4.6, which uses Lemma 5.4.

Lemma 5.4 *We consider a MLS execution on a graph G computing α and vertices u, v of G such that $\alpha^{-1}(u) > \alpha^{-1}(v)$. If $l_u(v) \preceq l_u(\alpha(1))$ and every neighbor t of v such that $\alpha^{-1}(u) \geq \alpha^{-1}(t) > \alpha^{-1}(v)$ is a neighbor of $\alpha(1)$ then $l_v(v) = l_v(\alpha(1))$ and every neighbor t of $\alpha(1)$ such that $\alpha^{-1}(u) \geq \alpha^{-1}(t) > \alpha^{-1}(v)$ is a neighbor of v.*

Proof: At every step between numbering u and numbering v, if the label of v is incremented then the label of $\alpha(1)$ is incremented too, so by (ls1) and (ls2) the inequality $l(v) \preceq l(\alpha(1))$ is preserved, and since $l(v)$ is maximal just before numbering v, $l_v(v) = l_v(\alpha(1))$. Now if some vertex t numbered before u and v was a neighbor of $\alpha(1)$ but not of v then the inequality would become $l(v) \prec l(\alpha(1))$, which would be preserved until numbering v, a contradiction. □

Theorem 4.6 For any graph G, any labelling structure $\mathcal{L}$ and any $\mathcal{L}$-MLS ordering α of G, if the order on labels is total then $\alpha(1)$ is an OCF-vertex of G.

Proof: We define the partition of V into subsets T'_i, $1 \leq i \leq p'$, which is a variant of the partition of V into thin slices. For any i in $[1..p']$,
- $t'_i = \alpha(n)$ if $i = 1$, otherwise t'_i is the first vertex numbered after c'_{i-1} not belonging to $N[C'_{i-1}]$,
- $c'_i = t'_i$ if $t'_i \notin N(\alpha(1))$, otherwise c'_i is the first vertex numbered after t'_i not belonging to $N(\alpha(1))$ ($c'_{p'} = \alpha(1)$),
- C'_i is the component of (G, α) containing c'_i ($C'_{p'} = \{\alpha(1)\}$),
- T'_i is the set of vertices t such that $\alpha^{-1}(t'_i) \geq \alpha^{-1}(t)$ and (if $i < p'$) $\alpha^{-1}(t) > \alpha^{-1}(t'_{i+1})$, so by definition, the subsets T'_i are numbered in increasing order.

Note that for any i in $[1..p']$ $T'_i \subseteq N(\alpha(1)) \cup C'_i$ and that there may be distinct i, j such that $C'_i = C'_j$, so p' can be greater than the number of components of (G, α).

We first show that for any i in $[1..p']$, $l(\alpha(1))$ is maximum among labels of unnumbered vertices just before numbering t'_i. The proof goes by induction on i. It obviously holds for $i = 1$. We suppose that it holds for some $i < p'$. $l_{t'_i}(t'_{i+1}) \preceq l_{t'_i}(\alpha(1))$. By Lemma 5.4 with $u = t'_i$ and $v = t'_{i+1}$ $l_{t'_{i+1}}(t'_{i+1}) = l_{t'_{i+1}}(\alpha(1))$, and since $l(t'_{i+1})$ is maximum at that step (because the order $\preceq$ is total) $l(\alpha(1))$ is maximum too.

Now let $y, z \in N(\alpha(1))$ such that $\alpha^{-1}(y) > \alpha^{-1}(z)$. Let us show that either $y \in N(z)$ or there is some $j < p'$ such that $y, z \in N(C'_j)$. Let $i \in [1..p']$ such that $y \in T'_i$.

First case: $\exists j \in [i..p'-1] \mid z \in N(C'_j)$

Let j' be the smallest integer such that $i \leq j'$ and $C'_{j'} = C'_j$. If $\alpha^{-1}(c'_{j'}) \geq \alpha^{-1}(y)$ then $\alpha^{-1}(c'_{j'}) \geq \alpha^{-1}(y) > \alpha^{-1}(t'_{i+1}) \geq \alpha^{-1}(t'_{j'+1})$, so by definition of $t'_{j'+1}$, $y \in N(C'_{j'})$. Otherwise $\alpha^{-1}(t'_i) \geq \alpha^{-1}(y) > \alpha^{-1}(c'_{j'})$ and by Lemma 5.4 with $u = t'_i$ and $v = c'_{j'}$, $y \in N(c'_{j'})$, so $y \in N(C'_{j'})$. Thus in both cases $y, z \in N(C'_j)$.

Second case: $\forall j \in [i..p'-1]$, $z \notin N(C'_j)$

$\alpha^{-1}(t'_i) \geq \alpha^{-1}(y) > \alpha^{-1}(z)$ so by Lemma 5.4 with $u = t'_i$ and $v = z$,

$y \in N(z)$.
Hence $\alpha(1)$ is an OCF-vertex of the input graph. $\square$

- We now present the proof of Characterization 4.7:

Characterization 4.7 For any graph G and any vertex x of G, x is MNS-terminal iff x is simplicial and the neighborhoods of connected components of $G(V \setminus N[x])$ are totally ordered by inclusion.

We will use the following well-known property of chordal graphs:

Property 5.5 *(confluence vertex property) Let $G = (V, E)$ be a chordal graph, let C be a connected subset such that $N(C)$ is a clique. Then $\exists z \in C \mid N(C) \subseteq N(z)$.*

Proof: (of Characterization 4.7) If x is MNS-terminal then x is simplicial since MNS computes a peo of G and by Theorem 3.8 neighborhoods are totally ordered by inclusion.
Conversely, we suppose that x is simplicial and neighborhoods of connected components of $G(V \setminus N[x])$ are totally ordered by inclusion. Let $(C_1, C_2, ..., C_q)$ be an ordering on the components of $G(V \setminus N(x))$ such that $\forall j \in [1..q-1]$, $N(C_j) \subseteq N(C_{j+1})$, with $C_q = \{x\}$. . For any $j \in [1..q]$, let $S'_j = N(C_j) \setminus N(C_{j-1})$, let $T'_j = S'_j \cup C_j$ and let $c_j \in C_j \mid N(C_j) \subseteq N(c_j)$ (c_j exists by Property 5.5 because as x is simplicial, C_j is simplicial too). Let us show that there is a MNS execution on G numbering the sets T'_j in increasing order. We prove by induction on j that there is a MNS execution on G numbering successively T'_1, T'_2, ..., T'_j first. It obviously holds for $j = 0$. We suppose that it holds for $j-1$ for some $j \leq q$. After numbering T'_1, T'_2, ..., T'_{j-1}, for every unnumbered vertex t, $l(t) = NN(t) \subseteq N(C_{j-1}) \subseteq N(C_j) \subseteq N(c_j)$. So c_j can be chosen at that step. It remains to show that at any step after numbering c_j and some, but not all, vertices of T'_j, it is possible to choose the next vertex in T'_j. Let y be an unnumbered vertex in T'_j, μ be a $c_j y$-path whose internal vertices are in C_j, let y' be the first unnumbered vertex of μ from c_j and z' be the numbered vertex preceding y' on μ. Then $z' \in l(y')$. Let y'' be an unnumbered vertex whose label is maximal among those of unnumbered vertices t such that $z' \in l(t)$. So the label of y'' is maximal among those of all unnumbered vertices and $y'' \in T'_j$ (since $z' \in C_j \cap N(y'')$). Thus y'' can be chosen at that step. Hence T'_1, T'_2, ..., T'_q can be numbered in increasing order. As the vertices of T'_q form a clique module (since x is simplicial) they can be numbered in an arbitrary order and x can be numbered last. $\square$