REPORTS
IN
INFORMATICS

ISSN 0333-3590

Lex M versus MCS-M

Yngve Villanger

REPORT NO 261 **January 2004**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2004-261.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Lex M versus MCS-M

Yngve Villanger*

Abstract

We study the problem of minimal triangulation of graphs. One of the first algorithms to solve this problem is Lex M, which was presented in 1976. A new algorithm, and a simplification of Lex M, called MCS-M was presented in 2002. In this paper we compare these two algorithms and show that they produce the same set of chordal graphs, answering an open question mentioned by the authors of MCS-M.

1 Introduction

Graph theory has several important problems that rely on creating a chordal supergraph from a given graph by adding a set of edges. The set of added edges is called *fill*, and this process is called a *triangulation* of the given graph. Several different goals may be desired; one is to introduce as few new edges as possible called *minimum fill*, and another is to create a triangulation such that the largest clique is as small as possible, which corresponds to the *treewidth* of the graph. Both these problems are NP-hard [1, 13].

Minimal fill, also called *minimal triangulation*, is a triangulation such that there exists no chordal subgraph where only fill edges are removed from the graph. There exist several practical algorithms that solve this problem [2, 3, 5, 6, 7, 9, 11, 12]. Since minimum fill is hard to compute, minimal fill may be used as an alternative, even though the difference in the number of fill edges may be quite large. One of the algorithms that solve the minimal triangulation problem is Lex M (Rose, Tarjan and Lueker [12]), which is a classical algorithm based on a special breadth first search, and lexicographic labeling of the vertices. Recently (Berry, Blair, Heggernes and Peyton [4]) introduced a new algorithm called MCS-M. This is a simplification of Lex M so that cardinality labels are used instead of lexicographic labels.

Triangulation of a graph can also be obtained by using *the elimination game*[10] algorithm. This algorithm takes the graph and an ordering of the vertices as input. The ordering of the vertices given to the elimination game is also called an *elimination ordering*. This ordering uniquely defines the set of fill edges for a given graph, but many different elimination orderings introduce the same set of fill edges. If an ordering produces a minimal triangulation, then the ordering is called a *minimal elimination ordering*(meo).

Both Lex M and MCS-M produce a minimal elimination ordering(meo). The user of Lex M and MCS-M can select the last vertex in the ordering, and has some choices during the execution of the algorithm. Because of these choises, both algorithms produce a set of minimal orderings from a given graph. The number of orderings in each of these sets can be quite different, and some orderings may only occure in one of the sets. However in this paper we show that for every Lex M ordering, there exists an MCS-M ordering which create the exact same fill edges, and for every MCS-M ordering there exists a Lex M ordering. It follows that Lex M and MCS-M create the exact same set of triangulations.

*Department of Informatics, University of Bergen, N-5020 Bergen, Norway. `yngvev@ii.uib.no`

2 Preliminaries

In this paper we consider finite, simple, undirected and connected graphs. Given a graph $G = (V, E)$, we denote the number of vertices as $n = |V|$ and number of edges as $m = |E|$. The *neighborhood* of a vertex $u \in V$ is denoted by $N_G(u) = \{v \mid (u, v) \in E\}$, and $N_G[u] = N_G(u) \cup \{u\}$. In the same way we define the neighborhood of a set $A \subseteq V$ of vertices by $N_G(A) = \cup_{u \in A} N_G(u) \backslash A$. A vertex sequence $v_1 - v_2 - ... - v_k$ describes a *path* if (v_i, v_{i+1}) is an edge for $1 \leq i < k$. The *length* of a path is the number of edges in it. A *cycle* is a path that starts and ends with the same vertex. If there is an edge between every pair of vertices in a set $A \subseteq V$, then the set A is called a *clique*. A vertex whose neighbors induce a clique is called *simplicial*.

Chordal graphs are the family of graphs where every cycle of length greater than three has a chord. A *chord* is an edge between two non-consecutive vertices of a cycle. Chordal graphs can be computed from non chordal graphs by introducing new edges, called *fill* edges. This process is called a *triangulation*. Given a graph G and an ordering α of the vertices in G, then *the elimination game*[10] can be used to obtain a triangulation G^+_α of the given graph G. The triangulation is obtained by picking the first vertex from the ordering, make its neighborhood into a clique, and then remove the vertex from the graph. This is repeated until no vertex remains. The ordering α is called an *elimination ordering*. The vertex at position i is given by $\alpha(i)$, and $\alpha^{-1}(u)$ gives us the position of the vertex u in the ordering. Theorem 2.1 gives an exact description of what edges that exist in the resulting graph, given a graph and an ordering of the vertices.

Theorem 2.1 *(Rose, Tarjan and Lueker [12]) Given a graph $G = (V, E)$ and an elimination ordering α of G, (y, z) is an edge in G^+_α if and only if $(y, z) \in E$ or there exists a path $y, x_1, x_2, ..., x_k, z$ in G where $\alpha^{-1}(x_i) < min\{\alpha^{-1}(y), \alpha^{-1}(z)\}$, for $1 \leq i \leq k$.*

Lex M computes a minimal elimination ordering given a graph and a starting vertex. The elimination order is produced back to front, and the last vertex in the ordering can be selected by the user. Each vertex in Lex M is assigned a label. This label is a sequence of number ordered in decreasing order. Let $L(u)$ be the label of vertex u, and let $L_k(u)$ be the number at position k in the sequence $L(u)$. The labels can be compared in the following way: $L(u) = L(v)$ if $|L(u)| = |L(v)|$ and $L_i(u) = L_i(v) \mid 1 \leq i \leq |L(u)|$. And $L(u) < L(v)$ if $L_k(u) < L_k(v)$, where k is the smallest number such that $L_k(u) \neq L_k(v)$, or $L_i(u) = L_i(v) \mid 1 \leq i \leq |L(u)|$ and $|L(u)| < |L(v)|$.

Lex M (Rose, Tarjan and Lueker [12])

input : $G = (V, E)$.

output : A minimal elimination ordering α and G_α^+.

begin
  $G_\alpha^+ = G$;
  for *all vertices* u *in* G **do**
    $L(u) = \emptyset$;
  for $i = n$ *to* 1 **do**
    Let v be one of the unnumbered vertices with largest label;
    $\alpha^{-1}(v) = i$;
    for *each unnumbered vertex* u *such that there is a path between* u *and* v *in* G *containing only unnumbered vertices that all have lower labels than* $L(u)$ **do**
      add i to $L(u)$;
      add fill edge (v, u) to G_α^+;
end

MCS-M is also an algorithm that produces a minimal elimination ordering. In the same way as Lex M, MCS-M produces an elimination ordering back to front, where the last vertex in the ordering can be selected by the user. MCS-M differs from Lex M by using cardinality weight $w(x)$ of a vertex x instead of a label. MCS-M basically uses the same technique as Lex M when searching the graph.

MCS-M (Berry, Blair and Heggernes [4])

input : $G = (V, E)$.

output : A minimal elimination ordering α and G_α^+.

begin
  $G_\alpha^+ = G$;
  for *all vertices* u *in* G **do**
    $L(u) = \emptyset$;
  for $i = n$ *to* 1 **do**
    Let v be one of the unnumbered vertices with largest weight;
    $\alpha^{-1}(v) = i$;
    for *each unnumbered vertex* u *such that there is a path between* u *and* v *in* G *containing only unnumbered vertices that all have lower weight than* $w(u)$ **do**
      $w(u) = w(u) + 1$;
      add fill edge (v, u) to G_α^+;
end

Both Lex M and MCS-M have the possibility to let the user select the next vertex from the set of unnumbered vertices with the largest label or weight. These choices are not necessarily the same for the two algorithms. In Figure 1, there is an example where Lex M and MCS-M do not have the same choices.

To make it easier to discuss Lex M and MCS-M we give an exact description of the label and weight for each vertex at each step of the algorithm. Let $L_{z-}(x)$ be the label of vertex x in Lex M, right before z has been assigned the number $\alpha^{-1}(z)$ and added this number to the labels described by Lex M, and let $L_{z+}(x)$

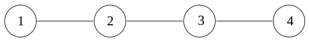

Figure 1: Let 2 be the starting vertex in the given graph. In this situation Lex M is capable of creating the following set of elimination orderings $[\{4,3,1,2\},\{4,1,3,2\}]$, while MCS-M is capable of creating the following set of orderings $[\{4,3,1,2\},\{4,1,3,2\},\{1,4,3,2\}]$. Observe that every one of these orderings is a perfect elimination ordering(peo)[8] for the given graph.

be the label of x right after. For MCS-M we do the same, let $w_{z-}(x)$ be the weight of vertex x in MCS-M, right before z has been assigned the number $\alpha^{-1}(z)$ and increase the weight of other vertices as described by MCS-M, and let $w_{z+}(x)$ be the weight of x right after. Given a set $A \subseteq V$ of vertices then $hW_{z-}(A)$ is the set of vertices in A with the highest weight assigned by MCS-M, and $hL_{z-}(A)$ is the set of vertices in A with the largest labels assigned by Lex M right before z has been assigned a number.

3 Orderings and labeling in Lex M

Before starting on comparing Lex M and MCS-M, it is necessary to look at elimination orderings, and the labels in Lex M. Let us consider the elimination ordering.

The set of vertices *monotonely adjacent* to a vertex is the set of higher numbered neighbors, and is defined as follows. Given a graph $G = (V, E)$ and an ordering α of the vertices, then $madj_{G^+_\alpha}(z) = \{w \mid (z,w) \in E(G^+_\alpha), \alpha^{-1}(z) < \alpha^{-1}(w)\}$. We are interested in knowing which changes that can be done to the ordering without altering the resulting triangulation. Our approach is to consider two adjacent vertices in the ordering, and decide if they can switch places in the ordering.

Lemma 3.1 *Given a graph $G = (V, E)$, and an ordering $\alpha = [x_1, x_2, ..., x_k, u, v, x_{k+3}, ..., x_n]$ of $V(G)$, let $\beta = [x_1, x_2, ..., x_k, v, u, x_{k+3}, ..., x_n]$ (u and v are swapped). If $(u,v) \notin E(G^+_\alpha)$ then $G^+_\alpha = G^+_\beta$.*

Proof: We want to show that $madj_{G^+_\alpha}(z) = madj_{G^+_\beta}(z)$ for each $z \in V$, since it then follows that $G^+_\alpha = G^+_\beta$. Let z be any vertex in $V \setminus \{u,v\}$. The set of vertices appearing prior to z in α and in β is exactly the same. It follows from Theorem 2.1 that $madj_{G^+_\alpha}(z) = madj_{G^+_\beta}(z)$ for any $z \in V \setminus \{u,v\}$. Let us now consider the vertices u and v. The edge $(u,v) \notin E(G^+_\alpha)$, and due to Theorem 2.1, there exists no path from u to v in G that only passes through the vertices $x_1, x_2, ..., x_k$. We show that $madj_{G^+_\alpha}(u) = madj_{G^+_\beta}(u)$. In order to do this we will show that both $madj_{G^+_\alpha}(u) \setminus madj_{G^+_\beta}(u)$ and $madj_{G^+_\beta}(u) \setminus madj_{G^+_\alpha}(u)$ are empty sets. Let us first on the contrary assume that there exists a vertex $z \in madj_{G^+_\beta}(u) \setminus madj_{G^+_\alpha}(u)$. Then there must exist a path from u to z in G that only passes through vertices $x_1, x_2, ..., x_k, v$, and this path must contain v, since there does not exist any path in G between u and z that only uses vertices $x_1, x_2, ..., x_k$, because $z \notin madj_{G^+_\alpha}(u)$. This gives a contradiction since there exists no path in G from u to v that only uses vertices $x_1, x_2, ..., x_k$, and thus no such path between u and z through v can exist. Now let us on the contrary assume that there exists a vertex $z \in madj_{G^+_\alpha}(u) \setminus madj_{G^+_\beta}(u)$. This is a contradiction since there must exist a path from u to z in G that only passes through the vertices $x_1, x_2, ..., x_k$, but no such path that only passes through the vertices $x_1, x_2, ..., x_k, v$. It follows that $madj_{G^+_\alpha}(u) = madj_{G^+_\beta}(u)$. It remains to show that $madj_{G^+_\alpha}(v) = madj_{G^+_\beta}(v)$. This is proved in the same way as for u. Let us first on the contrary assume that there exists a vertex $z \in madj_{G^+_\alpha}(v) \setminus madj_{G^+_\beta}(v)$. Then there must exist a path from v to z in G, which only passes through the vertices $x_1, x_2, ..., x_k, u$. This path must contain u because there does not exist any path in G that only passes through the vertices $x_1, x_2, ..., x_k$, since $z \in madj_{G^+_\alpha}(v) \setminus madj_{G^+_\beta}(v)$. This is a contradiction since there does not exist a path from u to v in G, that only passes through the vertices $x_1, x_2, ..., x_k$. Let us on the contrary assume that there exists a vertex $z \in madj_{G^+_\beta}(v) \setminus madj_{G^+_\alpha}(v)$. This is a contradiction because there must exist a path from v to z in G that only passes through the vertices $x_1, x_2, ..., x_k$, but there do not exist any path in G that only passes through the vertices $x_1, x_2, ..., x_k, u$, since $z \in madj_{G^+_\beta}(v) \setminus madj_{G^+_\alpha}(v)$. It follows that $madj_{G^+_\alpha}(v) = madj_{G^+_\beta}(v)$. $\square$

It is also necessary to obtain some more details regarding the labels in Lex M and the length of these labels. We will look at the length of labels where there is an unnumbered path between the vertices in the graph.

Lemma 3.2 *Let there exist an unnumbered path $x_0, x_1, ..., x_k$ where $k \geq 1$ and $u = x_0, v = x_k$ right before step $\alpha^{-1}(z)$ of Lex M, and let $L_{z-}(x_i) \leq L_{z-}(u)$ where $0 < i < k$. Then $|L_{z-}(u)| > |L_{z-}(v)|$ if and only if $L_{z-}(u) > L_{z-}(v)$.*

Proof: $\Rightarrow$ Let us first on the contrary assume that $|L_{z-}(u)| \leq |L_{z-}(v)|$ and $L_{z-}(u) > L_{z-}(v)$. Let v' be the vertex such that $\alpha^{-1}(v')$ is the largest number in $L_{z-}(v) \setminus L_{z-}(u)$ that does exists since $|L_{z-}(v)| \geq |L_{z-}(u)|$ and $L_{z-}(v) < L_{z-}(u)$. It follows that $\alpha^{-1}(v') > \alpha^{-1}(z)$ since $\alpha^{-1}(v') \in L_{z-}(v)$. Let q be the smallest number such that $0 < q \leq k$ and $\alpha^{-1}(v') \in L_{v'+}(x_q)$. Then $L_{v'-}(x_q) \geq L_{v'-}(x_i) \mid 0 \leq i < q$. We show this by contradiction. Let p be the largest number such that $0 \leq p < q$ and $L_{v'-}(x_p) > L_{v'-}(x_q)$. Now we have a path from x_p to v' through vertex x_q where $L_{v'-}(v') \geq L_{v'-}(x_p)$ and $L_{v'-}(x_p) > L_{v'-}(x_i) \mid p < i \leq q$, and since $\alpha^{-1}(v') \in L_{v'+}(x_q)$ there exists a path from x_q to v' where the labels of all the vertices in the path are smaller than both the labels of x_q and v'. This is a contradiction since $\alpha^{-1}(v') \notin L_{v'+}(x_p)$. Now back to our main proof. A result of this is that $L_{v'+}(x_q) > L_{v'+}(x_i) \mid 0 \leq i < q$. Now we have a contradiction since $\alpha^{-1}(v') \notin L_{z-}(u)$ and $L_{v'+}(x_q) > L_{v'+}(u = x_0)$ thus $L_{z-}(x_q) > L_{z-}(u = x_0)$.
$\Leftarrow$ Let us next on the contrary assume that $|L_{z-}(u)| > |L_{z-}(v)|$ and $L_{z-}(u) \leq L_{z-}(v)$. Let u' be the vertex such that $\alpha^{-1}(u')$ is the largest number in $L_{z-}(u) \setminus L_{z-}(v)$, which does exist since $|L_{z-}(u)| > |L_{z-}(v)|$. Let p be the largest number such that $0 \leq p < k$ and $\alpha^{-1}(u') \in L_{u'+}(x_p)$. It follows that $L_{u'-}(x_p) \geq L_{u'-}(x_i) \mid p < i \leq k$, since none of the labels of these vertices are assigned the number $\alpha^{-1}(u')$, thus $L_{u'+}(x_p) > L_{u'+}(x_i) \mid p < i \leq k$. Now we have a contradiction since $\alpha^{-1}(u') \notin L_{z-}(v)$ and $L_{u'+}(x_p) > L_{u'+}(v = x_k)$ thus $L_{z-}(x_p) > L_{z-}(v = x_k) \geq L_{z-}(u)$. $\square$

Lemma 3.2 can also be used in the case where $|L_{z-}(u)| < |L_{z-}(v)|$ if and only if $L_{z-}(u) < L_{z-}(v)$. This is described in Lemma 3.3.

Lemma 3.3 *Let there exist an unnumbered path $x_0, x_1, ..., x_k$ where $k \geq 1$ and $u = x_0, v = x_k$ right before step $\alpha^{-1}(z)$ of Lex M, and let $L_{z-}(x_i) \leq L_{z-}(u)$ where $0 < i < k$. Then $|L_{z-}(u)| > |L_{z-}(v)|$ if and only if $L_{z-}(u) > L_{z-}(v)$,*
$|L_{z-}(u)| < |L_{z-}(v)|$ if and only if $L_{z-}(u) < L_{z-}(v)$,
$|L_{z-}(u)| = |L_{z-}(v)|$ if and only if $L_{z-}(u) = L_{z-}(v)$.

Proof: The first case is Lemma 3.2. Second case is proved using Lemma 3.2. Let us first assume that $L_{z-}(u) < L_{z-}(v)$ and then prove that $|L_{z-}(u)| < |L_{z-}(v)|$. It follows that $L_{z-}(x_i) < L_{z-}(v) \mid 0 < i < k$ since $L_{z-}(u) < L_{z-}(v)$. Lemma 3.2 can now be used on the path $x_k, x_{k-1}, ..., x_0 \mid k \geq 1$ where $u = x_k, v = x_0$, thus $|L_{z-}(u)| < |L_{z-}(v)|$. But Lemma 3.2 also proves the only if part, and thus $L_{z-}(u) < L_{z-}(v)$ given that $|L_{z-}(u)| < |L_{z-}(v)|$. The third case follows, since no alternatives are left. $\square$

4 Lex M versus MCS-M

Lex M and MCS-M are not that different when it comes to altering labels and weights. Given that a vertex z is selected as the next vertex to be numbered for both algorithms, both Lex M and MCS-M do a search among unnumbered vertices that can be reached from z. In order to compare them, these unnumbered vertices are defined as a component.

Definition 4.1 *Let S be the set of numbered vertices, at some step of Lex M or MCS-M on $G = (V, E)$. Then an* unum component *is a connected component of $G(V \setminus S)$.*

The basic idea when proving that Lex M and MCS-M create the exact same set of triangulations, is as follows. First step is to show that an unum component can not affect other unum components at the same step. Then prove that Lex M and MCS-M have the same choices, in each unum component when a vertex is numbered.

Observe that no edges exist between vertices in different unum components. We want to prove that only the elimination order within each unum component affects the resulting graph, and thus the unum components can be treated individually.

Lemma 4.2 *Given a graph G and an ordering α of G and a vertex u. Remove every vertex v and the set of edges incident to v in G from the graph, where $\alpha^{-1}(v) > \alpha^{-1}(u)$. Then a set of unum components C_u will remain, where $0 \leq |C_u| < n$. Any ordering of the remaining vertices of α will create the same graph G_α^+, as long as the vertices of every unum component $C \in C_u$ are in the partial same order as in α.*

Proof: There are no edges between vertices in different unum components, since the numbered and removed vertices separate them, and no edges will be introduced either since all remaining vertices are eliminated before those who are numbered. Lemma 3.1 ensures that the same graph is obtained if two following vertices in the ordering α is swapped and there does not exist an edge between these vertices in the triangulated graph. The result is that we can always swap two following vertices from different unum components. Thus the same graph G_α^+ is obtained as long as the vertices of every unum component $C \in C_u$ is in the same order as in α. $\square$

The next step is to prove that Lex M and MCS-M treat an unum component in the same way. By treat we mean the following: Lex M can select z as the next vertex to be assigned a number in an unum component if and only if MCS-M can select it. The label of a vertex x in the unum component is changed by Lex M after z is numbered if and only if MCS-M change the weight of x after z is numbered. And finally that the same set of unum components remains when Lex M and MCS-M have numbered a vertex z in the unum component.

It follows quite easy that the third requirement is correct, since the same set of edges and vertices remains, when the vertex z and the edges incident to z are removed for the graph. Thus the same set of connected unum components will remain on both algorithms.

We now prove that $hW_{z-}(C) = hL_{z-}(C)$, and thus the vertex x is in $hW_{z-}(C)$ if and only if x is in $hL_{z-}(C)$. The proof is based on the following assumption. Given an unum component C, where $|L_{z-}(x)| = w_{z-}(x)$ for every $x \in C$ and z is the next vertex to be assigned a number. This assumption will follow from the final proof.

Lemma 4.3 *Given an unum component C, where $|L_{z-}(u)| = w_{z-}(u)$ for every $u \in C$, then $hW_{z-}(C) = hL_{z-}(C)$.*

Proof: We want to show that $hW_{z-}(C) \subseteq hL_{z-}(C)$ and $hL_{z-}(C) \subseteq hW_{z-}(C)$ and thus $hW_{z-}(C) = hL_{z-}(C)$. First step is to prove that $hW_{z-}(C) \subseteq hL_{z-}(C)$. Let us on the contrary assume that there exists a vertex $m \in hW_{z-}(C) \setminus hL_{z-}(C)$, and let l be any vertex in $hL_{z-}(C)$. The unum component C is connected, and every vertex in C is unnumbered. Thus there exists an unnumbered path $x_0, x_1, ..., x_k \mid 0 < k \leq |C|$, where $l = x_0, m = x_k$ and $x_i \in C \mid 0 \leq i \leq k$. Then $L_{z-}(x_i) \leq L_{z-}(l) \mid 0 < i \leq k$ since $l \in hL_{z-}(C)$. $w_{z-}(m) > w_{z-}(l)$ since $m \in hW_{z-}(C)$ and $l \notin hW_{z-}(C)$. Thus the path $x_0, x_1, ..., x_k$ is a contradiction to Lemma 3.3. Next we want to prove that $hL_{z-}(C) \subseteq hW_{z-}(C)$, and thus $hL_{z-}(C) = hW_{z-}(C)$. Let us on the contrary assume that there exists a vertex $l \in hL_{z-}(C) \setminus hW_{z-}(C)$, and let m be any vertex in $hW_{z-}(C)$. Then there exists a path $x_0, x_1, ..., x_k \mid 0 < k \leq |C|$, where $m = x_0, l = x_k$ and $x_i \in C \mid 0 \leq i \leq k$. This is also a contradiction to Lemma 3.3 since $L_{z-}(x_i) \leq L_{z-}(l) \mid 0 < i < k$, and $L_{z-}(l) > L_{z-}(m)$ and $w_{z-}(l) \leq w_{z-}(m)$. $\square$

Next step is to prove that Lex M and MCS-M alter the label and the weight of the same set of vertices in an unum component. The assumption that $|L_{z-}(x)| = w_{z-}(x)$ for every vertex x in the unum component is also used here.

Lemma 4.4 *Given a graph $G = (V, E)$, and let C be an unum component where $z \in C$ and $z \in hL_{z-}(C)$, and let $|L_{z-}(u)| = w_{z-}(u)$ for every $u \in C$, then $|L_{z+}(u)| = w_{z+}(u)$ for every $u \in C \setminus \{z\}$.*

Proof: Let us on the contrary assume that $|L_{z+}(u)| \neq w_{z+}(u)$ for some $u \in C \setminus \{z\}$. Two cases are possible. First case is if $|L_{z+}(u)| = w_{z+}(u) + 1$. The unum component C induces a connected subgraph in G, thus there exist at least one path $x_0, x_1, ..., x_k \mid k \geq 1$ where $u = x_0, z = x_k$, $x_i \in C$ and $L_{z-}(x_i) < L_{z-}(u) \leq L_{z-}(z) \mid 0 < i < k$, since $L_{z+}(u) = L_{z-}(u) + 1$. Then for every such path there exists a vertex $x_j \mid 0 < j < k$ such that $w_{z-}(x_j) \geq w_{z-}(u)$, since $w_{z+}(u) = w_{z-}(u)$.The path $x_0, x_1, ..., x_j$ is a contradiction to Lemma 3.3 since $L_{z-}(x_i) < L_{z-}(u = x_0) \mid 0 < i \leq j$ and $w_{z-}(x_j) \geq w_{z-}(u)$. The second case is when $|L_{z+}(u)| + 1 = w_{z+}(u)$ for some vertex $u \in C \setminus \{z\}$. Then there has to exist at least one path $x_0, x_1, ..., x_k \mid k \geq 1$ where $u = x_0, z = x_k$, $x_i \in C \mid 0 \leq i \leq k$ and $w_{z-}(x_i) < w_{z-}(u) \leq w_{z-}(z) \mid 0 < i < k$, since $w_{z+}(u) = w_{z-}(u)+1$. Then for every such path there exists a vertex $x_j \mid 0 < j < k$ such that $L_{z-}(x_j) \geq L_{z-}(u)$, since $|L_{z+}(u)| = |L_{z-}(u)|$. Let j be the smallest number such that $L_{z-}(x_j) \geq L_{z-}(u)$. The path $x_0, x_1, ..., x_j$ is a contradiction to Lemma 3.3 since $L_{z-}(x_i) < L_{z-}(u = x_0) \mid 0 < i < j$ and $L_{z-}(x_j) \geq L_{z-}(u = x_0)$ and $w_{z-}(x_j) < w_{z-}(u)$. $\square$

It is now time for the final proof, which is done by induction. We have to ensure that $|L_{z-}(u)| = w_{z-}(u)$ for every unnumbered vertex u in each unum component. Observe that Lex M and MCS-M do not select the next unum component in the same order, and thus $|L_{z-}(u)| \neq w_{z-}(u)$ when a given number of vertices are numbered.

Theorem 4.5 *Lex M and MCS-M create the exact same set of graphs.*

Proof: We prove this by induction.
Induction hypothesis: Given an unum component C where $z \in hW_{z-}(C) = hL_{z-}(C)$ and $|L_{z-}(u)| = w_{z-}(u) \,\forall\, u \in C$. Let C_z be the set of remaining unum components from the vertices in $C \setminus \{z\}$ when z and the edges incident to z are removed from the graph. Then $|L_{z+}(u)| = w_{z+}(u)$ for every $u \in C \setminus \{z\}$ and $hW_{z+}(C') = hL_{z+}(C')$ for every $C' \in C_z$.
Base case: Let the given graph be an unum component C, where $L_{z-}(u) = \emptyset$ and $w_{z-}(u) = 0$ for every $u \in V(G)$. Let z be the first vertex to be selected, then z is assigned the number $n-1$. Lex M will add $n-1$ to the label of every neighbor of z, and MCS-M will set the weight of every neighbor of z to 1. Thus $|L_{z+}(u)| = w_{z+}(u)$ for every $u \in V(G) \setminus \{z\}$. Removing the vertex z and the edges $(z, u) \mid u \in V(G)$ from G will leave a set C_z of unum components. Then due to Lemma 4.3 $hW_{z+}(C') = hL_{z+}(C')$ for every $C' \in C_z$, since $|L_{z+}(u)| = w_{z+}(u)$ where $u \in C'$.
Induction step: Given an unum component C where $|L_{z-}(u)| = w_{z-}(u)$ for every $u \in C$, and $z \in hW_{z-}(C) = hL_{z-}(C)$. Let C_z be the set of unum components when z is removed from C. Then we have to prove that $|L_{z+}(u)| = w_{z+}(u)$ for every $u \in C \setminus \{z\}$ and $hW_{z+}(C') = hL_{z+}(C')$ for every $C' \in C_z$. Lemma 4.4 ensures that $|L_{z+}(u)| = w_{z+}(u)$ for every $u \in C \setminus \{z\}$ and Lemma 4.3 ensures that $hW_{z+}(C') = hL_{z+}(C')$ for every $C' \in C_z$. $\square$

5 Conclusion

Even though MCS-M and Lex M can create different orderings, we prove that they create the same set of triangulations, and thereby answer an open question given in [4]. This is shown by defining an unum component, which is a connected subgraph when the numbered vertices are removed from the graph. Then we show that Lex M and MCS-M have the same choices when the size of the labels in Lex M is the same as the weight of the vertices in MCS-M.

We may also observe that each of the unum components can be computed individually since they do not affect each other. This property may probably be used to improve the practical running time for both algorithms.

Acknowledgment:

The author thank Pinar Heggernes for her useful comments.

References

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

[2] A. Berry. A wide-range efficient algorithm for minimal triangulation. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[3] A. Berry, J. Bordat, P. Heggernes, G. Simonet, and Y. Villanger. A wide-range algorithm for minimal triangulation from an arbitrary ordering, 2003.

[4] Anne Berry, Jean Blair, Pinar Heggernes, and Barry W. Peyton. Maximum cardinality search for computing minimal triangulations of graphs. *To appear in Algorithmica.*, 2003.

[5] Anne Berry, Pinar Heggernes, and Villanger Yngve. A vertex incremental approach for dynamically maintaining chordal graphs, 2003.

[6] Jean R. S. Blair, Pinar Heggernes, and Jan Arne Telle. A practical algorithm for making filled graphs minimal. *Theoret. Comput. Sci.*, 250(1-2):125–141, 2001.

[7] Elias Dahlhaus. Minimal elimination ordering inside a given chordal graph. In *Graph-theoretic concepts in computer science (Berlin, 1997)*, volume 1335 of *Lecture Notes in Comput. Sci.*, pages 132–143. Springer, Berlin, 1997.

[8] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965.

[9] Tatsuo Ohtsuki. A fast algorithm for finding an optimal ordering for vertex elimination on a graph. *SIAM J. Comput.*, 5(1):133–145, 1976.

[10] S. Parter. The use of linear graphs in Gauss elimination. *SIAM Rev.*, 3:119–130, 1961.

[11] Barry W. Peyton. Minimal orderings revisited. *SIAM J. Matrix Anal. Appl.*, 23(1):271–294 (electronic), 2001.

[12] Donald J. Rose, R. Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.

[13] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Algebraic Discrete Methods*, 2(1):77–79, 1981.