

A Simple Polynomial Algorithm for the Longest Path Problem on Cocomparability Graphs

George B. Mertzios*

Derek G. Corneil^{†‡}

Abstract

Given a graph G , the longest path problem asks to compute a simple path of G with the largest number of vertices. This problem is the most natural optimization version of the well known and well studied Hamiltonian path problem, and thus it is NP-hard on general graphs. However, in contrast to the Hamiltonian path problem, there are only few restricted graph families such as trees and some small graph classes where polynomial algorithms for the longest path problem have been found. Recently it has been shown that this problem can be solved in polynomial time on interval graphs by applying dynamic programming to a characterizing ordering of the vertices of the given graph [16], thus answering an open question. In the present paper, we provide the first polynomial algorithm for the longest path problem on a much greater class, namely on cocomparability graphs. Our algorithm uses a similar – but essentially simpler – dynamic programming approach, which is applied to a *Lexicographic Depth First Search (LDFS)* characterizing ordering of the vertices of a cocomparability graph. Therefore, our results provide evidence that this general dynamic programming approach can be used in a more general setting, leading to efficient algorithms for the longest path problem on greater classes of graphs. LDFS has recently been introduced in [8]. Since then, a similar phenomenon of extending an existing interval graph algorithm to cocomparability graphs by using an LDFS preprocessing step has also been observed for the minimum path cover problem [6]. Therefore, more interestingly, our results also provide evidence that cocomparability graphs present an interval graph structure when they are considered using an LDFS ordering of their vertices, which may lead to other new and more efficient combinatorial algorithms.

Keywords: Cocomparability graphs, longest path problem, lexicographic depth first search, dynamic programming.

1 Introduction

The Hamiltonian path problem, i.e. the problem of deciding whether a graph G contains a simple path that visits each vertex of G exactly once, is one of the most well known NP-complete problems, with numerous applications. The most natural optimization version of this problem is the *longest path* problem, that is, to compute a simple path of maximum length, or equivalently, to find a maximum induced subgraph which is Hamiltonian. Even if a graph itself is not Hamiltonian, it makes sense in several applications to search for a longest path. However, computing a longest path seems to be more difficult than deciding whether or not a graph admits a Hamiltonian path. Indeed, it has been proved that even if a graph is Hamiltonian, the problem of computing a path of length $n - n^\varepsilon$ for any $\varepsilon < 1$ is NP-hard, where n is the number of vertices

*Department of Computer Science, RWTH Aachen University, Aachen, Germany.
Email: mertzios@cs.rwth-aachen.de

[†]Department of Computer Science, University of Toronto, Toronto, Canada. Email: dgc@cs.utoronto.ca

[‡]The second author wishes to thank the Natural Sciences and Engineering Research Council of Canada for financial assistance.

of the input graph [18]. Moreover, there is no polynomial-time constant-factor approximation algorithm for the longest path problem unless $P=NP$ [18].

The Hamiltonian path problem (as well as many of its variants, e.g. the Hamiltonian cycle problem) is known to be NP-complete on general graphs [14]; furthermore, it remains NP-complete even when the input is restricted to some small classes of graphs, such as split graphs [15], chordal bipartite graphs, split strongly chordal graphs [22], directed path graphs [23], circle graphs [10], planar graphs [14], and grid graphs [17]. On other restricted families of graphs, however, considerable success has been achieved in finding polynomial time algorithms for the Hamiltonian path problem. In particular, this problem can be solved polynomially on proper interval graphs [2], interval graphs [1, 19], and cocomparability graphs [12] (see [3, 15] for definitions of these and other graph classes mentioned in the paper).

In contrast to the Hamiltonian path problem, there are only a few known polynomial algorithms for the longest path problem, and, until recently, these were restricted to trees [4], weighted trees and block graphs [28], bipartite permutation graphs [29], and ptolemaic graphs [27]. In [28] the question was raised whether the problem could be solved in polynomial time for a much larger class, namely interval graphs. Very recently such an algorithm has been discovered [16]. This algorithm applies dynamic programming to the vertex ordering of the given interval graph that is obtained after sorting the intervals according to their right endpoints. Another natural generalization of the Hamiltonian path problem is the *minimum path cover* problem, where the goal is to cover each vertex of the graph exactly once using the smallest number of simple paths. Clearly a solution to either the longest path or the minimum path cover problems immediately yields a solution to the Hamiltonian path problem. Unlike the situation for the longest path problem, polynomial time algorithms for the Hamiltonian cycle [19], Hamiltonian path, and minimum path cover [1] problems on interval graphs have been available since the 1980s and early 1990s.

Cocomparability graphs (i.e. graphs, the complements of which can be transitively oriented) strictly contain interval and permutation graphs [3] and have also been studied with respect to various Hamiltonian problems. In particular, it is well known that the Hamiltonian path and Hamiltonian cycle problems [13], as well as the minimum path cover problem (referred to as the Hamiltonian path completion problem [12]) are polynomially solvable on cocomparability graphs. On the other hand, the complexity status of the longest path problem on cocomparability graphs – and even on the smaller class of permutation graphs – has long been open.

Until recently, the only polynomial algorithms for the Hamiltonian path, Hamiltonian cycle [13], and minimum path cover [12] problems on cocomparability graphs exploited the relationship between these problems and the bump number of a poset representing the transitive orientation of the complement graph. Furthermore, it had long been an open question whether there are algorithms for these problems that, as with the interval graph algorithms, are based on the structure of cocomparability graphs. This question has been recently answered by the algorithm in [6], which solves the minimum path cover problem on cocomparability graphs by building off the corresponding algorithm for interval graphs [1] and using a preprocessing step based on the recently discovered *Lexicographic Depth First Search (LDFS)* [8].

In the present paper we provide the first polynomial algorithm for the longest path problem on cocomparability graphs (and thus also on permutation graphs). Our algorithm develops a similar – but much simpler – dynamic programming approach to that of [16], which is applied to a *Lexicographic Depth First Search (LDFS)* characterizing ordering of the vertices of a cocomparability graph (see [6, 8]). As a byproduct, this algorithm solves also the longest path problem on interval graphs in a much simpler way than that of [16] (the algorithm of [16] consists of three phases, during which it introduces several dummy vertices to construct a second auxiliary graph). Furthermore, these results provide evidence that this general dynamic programming

approach can be used in a more general setting, providing efficient algorithms for the longest path problem on greater classes of graphs. As already mentioned above, a similar phenomenon of extending an existing interval graph algorithm to cocomparability graphs by using an LDFS preprocessing step has also been observed for the minimum path cover problem [6]. Therefore, more interestingly, our results also provide evidence that cocomparability graphs present an interval graph structure when they are considered using an LDFS ordering of their vertices, which may lead to other new and more efficient combinatorial algorithms.

Organization of the paper In Section 2 we provide the necessary preliminaries and notation, including vertex ordering characterizations of interval graphs, cocomparability graphs, and LDFS orderings. In Section 3 we study the effect of an LDFS preprocessing step on the vertex ordering characterization of cocomparability graphs. This section provides much of the structural foundation for our longest path algorithm that is presented in Section 4. Finally, we discuss the presented results and further research in Section 5.

2 Preliminaries and notation

In this article we follow standard notation and terminology, see for instance [15]. We consider finite, simple, and undirected graphs. Given a graph $G = (V, E)$, we denote by n the cardinality of V . An edge between vertices u and v is denoted by uv , and in this case vertices u and v are said to be *adjacent*. \overline{G} denotes the *complement* of G , i.e. $\overline{G} = (V, \overline{E})$, where $uv \in \overline{E}$ if and only if $uv \notin E$. Let $S \subseteq V$ be a set of vertices of G . Then, the cardinality of S is denoted by $|S|$ and the subgraph of G *induced* by S is denoted by $G[S]$, i.e. $G[S] = (S, F)$, where for any two vertices $u, v \in S$, $uv \in F$ if and only if $uv \in E$. The set $N(v) = \{u \in V \mid uv \in E\}$ is called the *neighborhood* of the vertex $v \in V$ in G .

A *simple path* P of a graph G is a sequence of distinct vertices v_1, v_2, \dots, v_k such that $v_i v_{i+1} \in E$, for each $i \in \{1, 2, \dots, k-1\}$, and is denoted by $P = (v_1, v_2, \dots, v_k)$; throughout the paper all paths considered are simple. Furthermore, v_1 (resp. v_k) is called the *first* (resp. *last*) vertex of P . We denote by $V(P)$ the set of vertices of the path P , and define the *length* $|P|$ of P to be the number of vertices in P , i.e. $|P| = |V(P)|$. Additionally, if $P = (v_1, v_2, \dots, v_{i-1}, v_i, \dots, v_j, v_{j+1}, \dots, v_k)$ is a path of a graph and $P_0 = (v_i, \dots, v_j)$ is a subpath of P , we sometimes equivalently use the notation $P = (v_1, v_2, \dots, v_{i-1}, P_0, v_{j+1}, \dots, v_k)$.

Recall that *interval* graphs are the intersection graphs of closed intervals on the real line. Furthermore, a *comparability* graph is a graph whose edges can be transitively oriented (i.e. if $x \rightarrow y$ and $y \rightarrow z$, then $x \rightarrow z$); a *cocomparability* graph G is a graph whose complement \overline{G} is a comparability graph. *Permutation* graphs are exactly the intersection of comparability and cocomparability graphs. Moreover, cocomparability graphs strictly contain interval graphs and permutation graphs, as well as other families of graphs such as trapezoid graphs and cographs [3].

2.1 Vertex ordering characterizations

We now state vertex ordering characterizations of interval graphs, of cocomparability graphs and of any ordering of the vertex set V that can result from an LDFS search of an arbitrary graph $G = (V, E)$. The following ordering characterizes interval graphs and has appeared in a number of papers including [24].

Lemma 1 ([24]) $G = (V, E)$ is an interval graph if and only if there is an ordering (called an I-ordering) of V such that for all $x < y < z$, if $xz \in E$ then also $xy \in E$.

Note that the characterization of Lemma 1 can also result after sorting the intervals of an interval representation of G according to their left endpoints. Furthermore, note that some papers on interval graphs (for instance [1, 11, 16]) used the equivalent “reverse” vertex ordering, which results after sorting the intervals of an interval representation according to their right endpoints.

A similar characterization of *unit interval* graphs (also known as *proper interval* graphs) requires that if $xz \in E$ then both $xy, yz \in E$. It was observed in [20] that the following generalization of the interval order characterization captures cocomparability graphs.

Lemma 2 ([20]) *$G = (V, E)$ is a cocomparability graph if and only if there is an ordering (called an umbrella-free ordering, or a CO-ordering) of V such that for all $x < y < z$, if $xz \in E$ then $xy \in E$ or $yz \in E$ (or both).*

Observation 1 *An I-ordering of an interval graph G is also an umbrella-free ordering.*

In the following, we present the notion of the recently introduced Lexicographic Depth First Search (LDFS) ordering (see [8]).

Definition 1 *Let $G = (V, E)$ be a graph and σ be any ordering of V . Let (a, b, c) be a triple of vertices of G such that $a <_\sigma b <_\sigma c$, $ac \in E$, and $ab \notin E$. If there exists a vertex $d \in V$ such that $a <_\sigma d <_\sigma b$, $db \in E$, and $dc \notin E$, then (a, b, c) is a good triple; otherwise it is a bad triple. Furthermore, if the triple (a, b, c) is good, then vertex d is called a d -vertex of this triple.*

Definition 2 *Let $G = (V, E)$ be a graph. An ordering σ of V is a Lexicographic Depth First Search (LDFS) ordering if and only if σ has no bad triple.*

An example of a good triple (a, b, c) and a d -vertex of it is depicted in Figure 1. In this example, the edges ac and db are indicated with solid lines, while the non-edges ab and dc are indicated with dashed lines. Furthermore, the d -vertex is drawn gray for better visibility.

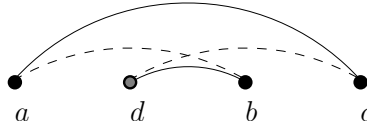


Figure 1: A good triple (a, b, c) and a d -vertex of this triple, in the vertex ordering $\sigma = (a, d, b, c)$.

2.2 Algorithms

In the following we present the generic LDFS algorithm (Algorithm 1) that starts at a distinguished vertex u . This algorithm has been recently introduced in [8]. It looks superficially similar to the well known and well studied *Lexicographic Breadth First Search (LBFS)* [25] (for a survey, see [5]); nevertheless, it appears that vertex orderings computed by the LDFS and by the LBFS have inherent structural differences. Briefly, the generic LDFS algorithm proceeds as follows. Initially, the label ε is assigned to all vertices. Then, iteratively, an unvisited vertex v with lexicographically maximum label is chosen and removed from the graph. If v is chosen as the i th vertex, then each of its neighbors that are still unnumbered have their label updated by having the digit i prepended to their label. The digits in the label of any vertex are always in decreasing order, which ensures that all neighbors of the last chosen vertex have a lexicographically greater label than its non-neighbors. By extension, this argument ensures that all vertices are visited in a depth-first search order. When applied to a graph with n vertices and m edges,

Algorithm 1 LDFS(G, u) [8]

Input: A connected graph $G = (V, E)$ with n vertices and a distinguished vertex u of G

Output: An LDFS ordering σ_u of the vertices of G

- 1: Assign the label ε to all vertices
 - 2: $label(u) \leftarrow \{0\}$
 - 3: **for** $i = 1$ to n **do**
 - 4: Pick an unnumbered vertex v with the lexicographically largest label
 - 5: $\sigma_u(i) \leftarrow v$ {assign to v the number i }
 - 6: **for** each unnumbered vertex $w \in N(v)$ **do**
 - 7: prepend i to $label(w)$
 - 8: **return** the ordering $\sigma_u = (\sigma_u(1), \sigma_u(2), \dots, \sigma_u(n))$
-

Algorithm 1 can be implemented to run in $O(\min\{n^2, n + m \log n\})$ time [21]; however, the current fastest implementation runs in $O(\min\{n^2, n + m \log \log n\})$ [26].

The execution of the LDFS algorithm is captured in the example shown in Figure 2. In this example, suppose that the LDFS algorithm starts at vertex e . Suppose that LDFS chooses vertex d next. Now, ordinary DFS could choose either a or c next, but LDFS has to choose c , since it has a greater label (c was a neighbor of the previously visited vertex e). The vertex following c in the LDFS ordering σ_e must be a rather than b , since a has a greater label than b (a was a neighbor of the previously visited vertex d). The LDFS then backtracks to b completing the LDFS ordering as $\sigma_e = (e, d, c, a, b)$.

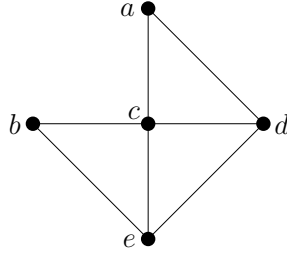


Figure 2: Illustrating LDFS.

It is important here to connect the vertex ordering σ_u that is returned by the LDFS algorithm (i.e. Algorithm 1) with the notion of an LDFS ordering, as defined in Definition 2. The following theorem shows that a vertex ordering σ of an arbitrary graph G can be returned by an application of the LDFS algorithm to G (starting at some vertex u of G) if and only if σ is an LDFS ordering.

Theorem 1 ([8]) *For an arbitrary graph $G = (V, E)$, an ordering σ of V can be returned by an application of Algorithm 1 to G if and only if σ is an LDFS ordering.*

In the generic LDFS, there could be some choices to be made at line 4 of Algorithm 1; in particular, at some iteration there may be a set S of vertices that have the same label and the algorithm must choose one vertex from S . Generic LDFS (i.e. Algorithm 1) allows an arbitrary choice here. We present in the following a special kind of an LDFS algorithm, called LDFS⁺ (cf. Algorithm 2 below), which makes a specific choice of a vertex in such a case of equal labels, as follows. Along with the graph $G = (V, E)$, an ordering π of V is also given as input. The algorithm LDFS⁺ (see Algorithm 2 for a formal description) operates exactly as a generic LDFS that starts at the *rightmost* vertex of V in the ordering π , with the only difference that, in the

case where at some iteration at least two unvisited vertices have the same label, it chooses the *rightmost* vertex among them in the ordering π .

Algorithm 2 LDFS⁺ (G, π)

Input: A connected graph $G = (V, E)$ with n vertices and an ordering π of V

Output: An LDFS ordering σ of the vertices of G

- 1: Assign the label ε to all vertices
 - 2: **for** $i = 1$ to n **do**
 - 3: Pick the rightmost vertex v in π among the unnumbered vertices with the lexicographically largest label
 - 4: $\sigma(i) \leftarrow v$ {assign to v the number i }
 - 5: **for** each unnumbered vertex $w \in N(v)$ **do**
 - 6: prepend i to $label(w)$
 - 7: **return** the ordering $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n))$
-

In the following, we present the *RightMost-Neighbor* (RMN) algorithm. Although the name RMN has not been used, this algorithm essentially has been introduced in [1] in order to find a minimum path cover in a given interval graph*. The RMN algorithm is a very simple “greedy” algorithm that starts at the rightmost vertex of a given ordering σ of V and traces each path by repeatedly proceeding to the rightmost unvisited neighbor of the current vertex. If the current vertex has no unvisited neighbors, then the rightmost unvisited vertex is chosen as the first vertex in the next path.

Algorithm 3 RMN(σ)

Input: A graph $G = (V, E)$ with n vertices and an ordering σ of V

Output: An ordering $\hat{\sigma}$ of the vertices of G

- 1: Label all vertices as “unvisited”; $i \leftarrow 1$
 - 2: **while** there are unvisited vertices **do**
 - 3: Pick the rightmost unvisited vertex x in σ
 - 4: $\hat{\sigma}(i) \leftarrow x$ {add vertex x to the ordering $\hat{\sigma}$ }
 - 5: Mark x as “visited”; $i \leftarrow i + 1$
 - 6: **while** x has at least one unvisited neighbor **do**
 - 7: Pick the x ’s rightmost unvisited neighbor y in σ
 - 8: $\hat{\sigma}(i) \leftarrow y$ {add vertex y to the ordering $\hat{\sigma}$ }
 - 9: Mark y as “visited”; $i \leftarrow i + 1$
 - 10: $x \leftarrow y$
 - 11: **return** $\hat{\sigma} = (\hat{\sigma}(1), \hat{\sigma}(2), \dots, \hat{\sigma}(n))$
-

Note that in Algorithm 2 we denote the input vertex ordering by π and the output ordering by σ , while in Algorithm 3, σ denotes the input vertex ordering. The reason for this notation is that we will often consider an arbitrary umbrella-free vertex ordering π of a cocomparability graph G , apply Algorithm 2 (i.e. LDFS⁺) to π to compute the ordering σ , and then apply Algorithm 3 (i.e. RMN) to σ to compute the ordering $\hat{\sigma}$. Then, as proved in [6], the LDFS vertex ordering σ remains umbrella-free. Moreover, the final ordering $\hat{\sigma}$ defines a minimum path cover of G [6].

*Actually, the algorithm of [1] uses the “reverse” vertex ordering of an I-ordering (as defined in Lemma 1), which results after sorting the intervals of an interval representation according to their right endpoints, and thus they presented an equivalent *LeftMost-Neighbor* (LMN) algorithm for the case of interval graphs. A similar observation applies to the algorithms in [11] and [16].

3 Normal paths in cocomparability graphs

In this section we investigate the structure of the vertex ordering σ that is obtained after applying an LDFS⁺ preprocessing step to an arbitrary umbrella-free ordering π of a cocomparability graph G . On such an LDFS umbrella-free ordering σ , we define a special type of paths, called *normal* paths (cf. Definition 5), which is a crucial notion for our algorithm for the longest path problem on cocomparability graphs (cf. Algorithm 4). In the following definition we introduce the notion of a *maximal* path in a graph, which extends that of a longest path.

Definition 3 *A path P of a graph G is maximal if there exists no path P' of G , such that $V(P) \subset V(P')$.*

The main result of this section is that for any maximal path P of a cocomparability graph G (and thus also for any longest path), there exists a normal path P' on the same vertices (cf. Theorem 2). Due to this result, it is sufficient for our algorithm that computes the longest path problem on cocomparability graphs (cf. Algorithm 4) to search only among the normal paths of the given cocomparability graph, in order to compute a longest path. The next lemma will be used in the sequel.

Lemma 3 *Let $G = (V, E)$ be a cocomparability graph and σ be an LDFS umbrella-free ordering of V . Let $P = (v_1, v_2, \dots, v_k)$ be a path of G and $v_\ell \notin V(P)$ be a vertex of G , such that $v_k <_\sigma v_\ell <_\sigma v_1$ and $v_\ell v_k \notin E$. Then, there exist two consecutive vertices v_{i-1} and v_i in P , $2 \leq i \leq k$, such that $v_{i-1}v_\ell \in E$ and $v_i <_\sigma v_\ell$.*

Proof. Since $v_k <_\sigma v_\ell <_\sigma v_1$ and $v_\ell \notin V(P)$, there exists at least one edge $e = xy$ of P , which straddles v_ℓ in σ . Thus, at least one of x and y is adjacent to v_ℓ , since σ is umbrella-free. Recall that $v_\ell v_k \notin E$; let v_{i-1} , $2 \leq i \leq k$, be the last vertex of P such that $v_{i-1}v_\ell \in E$. If $v_\ell <_\sigma v_i$, then there exists similarly at least one vertex v_j , $i \leq j \leq k$, such that $v_j v_\ell \in E$, which is a contradiction by the assumption on v_{i-1} . Thus, $v_i <_\sigma v_\ell$. This completes the proof of the lemma. ■

Definition 4 *Let $G = (V, E)$ be a cocomparability graph, σ be an LDFS umbrella-free ordering of V , and σ' be an induced subordering of σ . An LDFS closure σ'' of σ' (within σ) is an induced subordering of σ with the smallest number of vertices, such that σ'' is an LDFS ordering that includes σ' .*

Observe that any induced subordering σ' of an umbrella-free ordering σ also remains an umbrella-free ordering (cf. Lemma 2). An example of a cocomparability graph $G = (V, E)$, as well as an LDFS umbrella-free ordering $\sigma = (u_1, u_2, \dots, u_9)$ of V is illustrated in Figure 3. In this example, $\sigma' = (u_1, u_3, u_4, u_5, u_7, u_8)$ is an induced subordering of σ (and thus also umbrella-free). Furthermore, the ordering $\sigma'' = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8)$ is an LDFS closure of σ' (within σ), where the vertices u_2 and u_6 are the d -vertices of the triples (u_1, u_3, u_4) and (u_5, u_7, u_8) , respectively.

Observation 2 *Let σ be an LDFS umbrella-free ordering, σ' an arbitrary induced subordering of σ , and σ'' any LDFS closure of σ' (within σ). Then, every vertex v of $\sigma'' \setminus \sigma'$ is a d -vertex of some good triple (a, b, c) in σ'' .*

The next lemma follows easily by Observation 2.

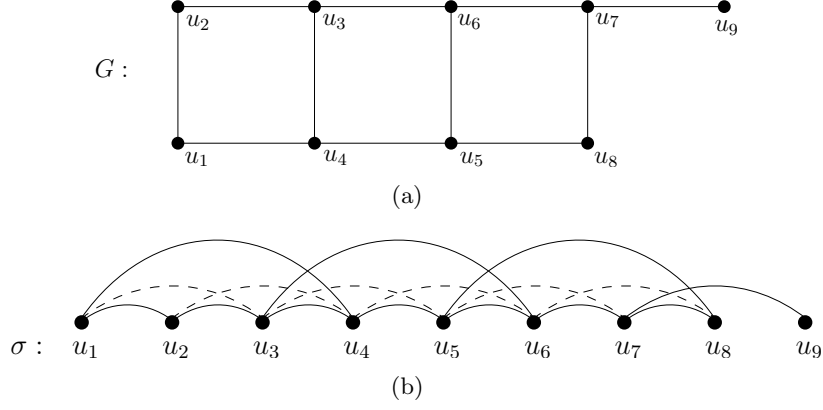


Figure 3: (a) A cocomparability graph $G = (V, E)$ and (b) an LDFS umbrella-free ordering $\sigma = (u_1, u_2, \dots, u_9)$ of V .

Lemma 4 *Let σ be an LDFS umbrella-free ordering, σ' an arbitrary induced subordering of σ , and σ'' any LDFS closure of σ' (within σ). Let v be a vertex of $\sigma'' \setminus \sigma'$. Then, there exists at least one vertex v' in σ' , such that $v <_\sigma v'$ and $vv' \notin E$.*

Proof. Suppose otherwise that for every vertex v' of σ'' , for which $v <_\sigma v'$ and $vv' \notin E$, the vertex v' belongs to $\sigma'' \setminus \sigma'$. Note by Observation 2 that v is a d -vertex of some good triple in σ'' ; let this triple be (a_0, b_0, c_0) . Then, $v <_\sigma c_0$ and $vc_0 \notin E$ by definition of a good triple. Thus, c_0 is a vertex of $\sigma'' \setminus \sigma'$ by our assumption on v . Then, c_0 is a d -vertex of some good triple in σ'' by Observation 2; let this triple be (a_1, b_1, c_1) . Then, in particular, $c_0 <_\sigma c_1$ and $c_0c_1 \notin E$ by definition of a good triple. Therefore $v <_\sigma c_0 <_\sigma c_1$. Furthermore $vc_1 \notin E$, since otherwise the vertices v, c_0, c_1 build an umbrella in σ , which is a contradiction. That is, $v <_\sigma c_1$ and $vc_1 \notin E$, and thus c_1 is a vertex of $\sigma'' \setminus \sigma'$ by our assumption on v . Now, for every $i \geq 2$, we can inductively construct a sequence c_2, c_3, \dots, c_i of vertices in $\sigma'' \setminus \sigma'$, such that $v <_\sigma c_0 <_\sigma c_1 <_\sigma c_2 <_\sigma \dots <_\sigma c_i$ and the vertices $v, c_0, c_1, c_2, \dots, c_i$ build an independent set. This is a contradiction, since σ'' is finite. Therefore, there exists at least one vertex v' in σ' , such that $v <_\sigma v'$ and $vv' \notin E$. This completes the proof of the lemma. ■

Corollary 1 *Let σ be an LDFS umbrella-free ordering, σ' an arbitrary induced subordering of σ , and σ'' any LDFS closure of σ' (within σ). Then, the rightmost vertex of σ' is also the rightmost vertex of σ'' .*

Proof. Let v' and v'' be the rightmost vertices of σ' and of σ'' , respectively. If $v' \neq v''$, then v'' is a vertex of $\sigma'' \setminus \sigma'$ and $v' <_\sigma v''$, since σ' is a subset of σ'' . Then, there exists by Lemma 4 at least one vertex v''' in σ' , such that $v'' <_\sigma v'''$, i.e. $v' <_\sigma v'' <_\sigma v'''$, which is a contradiction to our assumption on v' . ■

In the following we introduce the notion of a typical and a normal path in a cocomparability graph $G = (V, E)$ (with respect to an LDFS umbrella-free ordering σ of V), which will be used in the remainder of the paper.

Definition 5 *Let $G = (V, E)$ be a cocomparability graph and σ be an LDFS umbrella-free ordering of V . Then,*

- (a) *a path $P = (v_1, v_2, \dots, v_k)$ of G is called typical if v_1 is the rightmost vertex of $V(P)$ in σ and v_2 is the rightmost vertex of $N(v_1) \cap V(P)$ in σ , and*

- (b) a typical path $P = (v_1, v_2, \dots, v_k)$ of G is called *normal* if v_i is the rightmost vertex of $N(v_{i-1}) \cap \{v_i, v_{i+1}, \dots, v_k\}$ in σ , for every $i = 2, \dots, k$.

For example, in the cocomparability graph G of Figure 3, the path $P = (u_8, u_5, u_6, u_3, u_4)$ is a normal path. The next observation follows from Definition 5.

Observation 3 Let $G = (V, E)$ be a cocomparability graph and σ be an LDFS umbrella-free ordering of V . Let P be a normal path of G (with respect to the ordering σ) and $\sigma|_{V(P)}$ be the restriction of σ on the vertices of P . Then, the ordering of the vertices of $V(P)$ in P coincides with the ordering $\text{RMN}(\sigma|_{V(P)})$.

A similar notion of a normal (i.e. RMN) path for the special case of interval graphs has appeared in [11] (referred to as a *straight* path), as well as in [16]. We now state the following two auxiliary lemmas.

Lemma 5 ([6]) Let $G = (V, E)$ be a cocomparability graph and π be an umbrella-free ordering of V . Let $\pi' = \text{RMN}(\pi)$ and $\pi'' = \text{LDFS}^+(\pi)$. Furthermore, let $x, y \in V$ such that $xy \notin E$. If $y <_\pi x$, then $x <_{\pi'} y$ and $x <_{\pi''} y$.

Lemma 6 Let $G = (V, E)$ be a cocomparability graph, π be an umbrella-free ordering of V , and $\pi' = \text{RMN}(\pi)$. Let $x, y \in V$ such that $y <_\pi x$ and $y <_{\pi'} x$. Then, y is not the first vertex of π' and for the previous vertex z of y in π' , $y <_\pi x <_\pi z$, $zy \in E$, and $zx \notin E$.

Proof. First, note that the first vertex of the ordering $\pi' = \text{RMN}(\pi)$ is the rightmost vertex of π . Thus y is not the first vertex of π' , since $y <_\pi x$. Let z be the previous vertex of y in π' . Then, x is unvisited, when z is being visited by π' , since $z <_{\pi'} y <_{\pi'} x$. Suppose that $zx \in E$. Then, y could not be the next vertex of z in π' , since x is unvisited and $y <_\pi x$, which is a contradiction. Thus $zx \notin E$. Furthermore, Lemma 5 implies that $x <_{\pi'} z$, i.e. $y <_\pi x <_{\pi'} z$, since $z <_{\pi'} x$ and $zx \notin E$. Suppose now that $zy \notin E$. In the case where no neighbor of z is unvisited, when z is being visited by π' , then x is the next vertex of z in π' instead of y , since $y <_\pi x$, which is a contradiction. In the case where at least one neighbor w of z is unvisited, when z is being visited by π' , then one of the unvisited neighbors w of z is the next vertex of z in π' instead of y , which is again a contradiction. Thus, $zy \in E$. This completes the proof of the lemma. ■

Notation 1 In the remainder of this section, we consider a cocomparability graph $G = (V, E)$ and an LDFS umbrella-free ordering σ of G . Furthermore, we consider a maximal path P of G , the restriction $\sigma' = \sigma|_{V(P)}$ of σ on the vertices of P and an arbitrary LDFS closure σ'' of σ' (within σ). Finally, we consider the orderings $\hat{\sigma} = \text{LDFS}^+(\sigma')$ and $\hat{\hat{\sigma}} = \text{RMN}(\hat{\sigma})$.

The next structural lemma will be used in the sequel, in order to prove in Theorem 2 that for every maximal path P there exists a normal path P' of G , such that $V(P') = V(P)$.

Lemma 7 Let x, y, z be three vertices of σ' , such that $x <_{\hat{\sigma}} y <_{\hat{\sigma}} z$ and $z <_{\sigma'} y <_{\sigma'} x$, where $xy, xz \in E$ and $yz \notin E$. Then, x is not the next vertex of z in $\hat{\hat{\sigma}}$.

Proof. The proof will be done by contradiction. We will exploit the facts that P is a maximal path (cf. Notation 1) and that, given a Hamiltonian cocomparability graph H and an LDFS umbrella-free ordering π of H , the ordering $\text{RMN}(\pi)$ gives a Hamiltonian path of H [6]. Suppose that there exists a triple (x, y, z) of vertices in σ' that satisfy the conditions of the lemma, such that x is the next vertex of z in $\hat{\hat{\sigma}}$. Among all those triples, let (x, y, z) be the one, where z

is the rightmost possible in $\hat{\sigma}$ and y is the rightmost possible in $\hat{\sigma}$ among those with equal z . Note that always $z <_{\hat{\sigma}} y$ by Lemma 5, since $\hat{\sigma} = \text{RMN}(\hat{\sigma})$, and since $yz \notin E$ and $y <_{\hat{\sigma}} z$ by assumption.

Since $z <_{\sigma'} y <_{\sigma'} x$, $xz \in E$, and $yz \notin E$, and since σ'' is an LDFS-closure of σ' (within σ), there exists by Observation 2 a vertex d in σ'' , such that $z <_{\sigma''} d <_{\sigma''} y <_{\sigma''} x$, $dy \in E$, and $dx \notin E$. Thus, since $dx \notin E$ and σ is an umbrella-free ordering, it follows that $zd \in E$. In the following we will distinguish the cases where $d \in \sigma'$ and $d \in \sigma'' \setminus \sigma'$.

Case 1. Suppose first that $d \in \sigma'$, i.e. $d \in V(P)$, and thus $d \in \hat{\sigma}$. Then, since $\hat{\sigma} = \text{LDFS}^+(\sigma')$, and since $dx \notin E$ and $d <_{\sigma'} x$, it follows by Lemma 5 that $x <_{\hat{\sigma}} d$.

Case 1a. Suppose that $d <_{\hat{\sigma}} z$, i.e. $x <_{\hat{\sigma}} d <_{\hat{\sigma}} z$. If d is unvisited when z is being visited in $\hat{\sigma}$, then d would be the next vertex of z in $\hat{\sigma}$ instead of x , since $zd \in E$, which is a contradiction. Thus, d has been visited before z in $\hat{\sigma}$, i.e. $d <_{\hat{\sigma}} z$. Therefore, Lemma 6 implies that d is not the first vertex in $\hat{\sigma}$, while $d <_{\hat{\sigma}} z <_{\hat{\sigma}} a$, $ad \in E$, and $az \notin E$ for the previous vertex a of d in $\hat{\sigma}$. Then, in particular, $a <_{\sigma'} z$ by Lemma 5, since $z <_{\hat{\sigma}} a$, $az \notin E$, and $\hat{\sigma} = \text{LDFS}^+(\sigma')$. Summarizing, $d <_{\hat{\sigma}} z <_{\hat{\sigma}} a$ and $a <_{\sigma'} z <_{\sigma'} d$, where $dz, da \in E$ and $za \notin E$, while d is the next vertex of a in $\hat{\sigma}$. This comes in contradiction to the choice of the triple (x, y, z) , for which z is the rightmost possible in $\hat{\sigma}$.

Case 1b. Suppose that $z <_{\hat{\sigma}} d$, i.e. $y <_{\hat{\sigma}} z <_{\hat{\sigma}} d$. Recall that $yd \in E$ and $yz \notin E$. Thus, since $\hat{\sigma}$ is an LDFS ordering, there exists a vertex d' in $\hat{\sigma}$, such that $y <_{\hat{\sigma}} d' <_{\hat{\sigma}} z <_{\hat{\sigma}} d$, $d'z \in E$, and $d'd \notin E$. Note that $x <_{\hat{\sigma}} y <_{\hat{\sigma}} d'$. Similarly to the previous paragraph, if d' is unvisited when z is being visited in $\hat{\sigma}$, then d' would be the next vertex of z in $\hat{\sigma}$ instead of x , since $d'z \in E$, which is a contradiction. Thus, d' has been visited before z in $\hat{\sigma}$, i.e. $d' <_{\hat{\sigma}} z$. Therefore, Lemma 6 implies that d' is not the first vertex in $\hat{\sigma}$, while $d' <_{\hat{\sigma}} z <_{\hat{\sigma}} a'$, $a'd' \in E$, and $a'z \notin E$ for the previous vertex a' of d' in $\hat{\sigma}$. Then, in particular, $a' <_{\sigma'} z$ by Lemma 5, since $z <_{\hat{\sigma}} a'$, $a'z \notin E$, and $\hat{\sigma} = \text{LDFS}^+(\sigma')$. Similarly, $d <_{\sigma'} d'$, since $d' <_{\hat{\sigma}} d$ and $d'd \notin E$. Therefore, since $z <_{\sigma'} d$, it follows that $z <_{\sigma'} d <_{\sigma'} d'$. Summarizing, $d' <_{\hat{\sigma}} z <_{\hat{\sigma}} a'$ and $a' <_{\sigma'} z <_{\sigma'} d'$, where $d'z, d'a' \in E$ and $za' \notin E$, while d' is the next vertex of a' in $\hat{\sigma}$. This comes in contradiction to the choice of the triple (x, y, z) , such that z is the rightmost possible in $\hat{\sigma}$.

Case 2. Suppose now that $d \in \sigma'' \setminus \sigma'$, i.e. $d \notin V(P)$, and thus $d \notin \hat{\sigma}$. Consider the set of vertices w of $\hat{\sigma}$, such that $y <_{\hat{\sigma}} w <_{\hat{\sigma}} z$. We partition this set into the (possibly empty) sets $A = \{w \mid y <_{\hat{\sigma}} w <_{\hat{\sigma}} z, wz \notin E\}$ and $B = \{w \mid y <_{\hat{\sigma}} w <_{\hat{\sigma}} z, wz \in E\}$. First observe that $xw \in E$ for every $w \in A$, since $xz \in E$ and $zw \notin E$, and since $\hat{\sigma}$ is an umbrella-free ordering. We will now prove that $yw \in E$ for every vertex $w \in A$. Suppose otherwise that $yw, zw \notin E$ for a vertex w , for which $y <_{\hat{\sigma}} w <_{\hat{\sigma}} z$. Then, $z <_{\sigma'} w <_{\sigma'} y$ by Lemma 5, since $y <_{\hat{\sigma}} w <_{\hat{\sigma}} z$, $yw, zw \notin E$, and $\hat{\sigma} = \text{LDFS}^+(\sigma')$. Recall that $y <_{\sigma'} x$ by assumption in the statement of the lemma. Thus, $x <_{\hat{\sigma}} w <_{\hat{\sigma}} z$ and $z <_{\sigma'} w <_{\sigma'} x$, where $xw, xz \in E$ and $wz \notin E$, while x is the next vertex of z in $\hat{\sigma}$. Therefore, since $y <_{\hat{\sigma}} w$, this comes in contradiction to the choice of the triple (x, y, z) , such that y is the rightmost possible (with respect to z) in $\hat{\sigma}$. Therefore, $yw \in E$ for every vertex $w \in A$.

If a vertex $w \in B$ is unvisited when z is being visited in $\hat{\sigma}$, then w would be the next vertex of z in $\hat{\sigma}$ instead of x , which is a contradiction to the assumption. Thus, w has been visited before z in $\hat{\sigma}$, i.e. $w <_{\hat{\sigma}} z$, for every $w \in B$. On the other hand, Lemma 5 implies that $z <_{\hat{\sigma}} w$ for every $w \in A$, i.e. w is being visited after z in $\hat{\sigma}$, since $\hat{\sigma} = \text{RMN}(\hat{\sigma})$, $w <_{\hat{\sigma}} z$, and $wz \notin E$ for every $w \in A$. Let v be a vertex that is visited after z in $\hat{\sigma}$, i.e. $z <_{\hat{\sigma}} v$. Then, $v <_{\hat{\sigma}} z$. Indeed, suppose otherwise that $z <_{\hat{\sigma}} v$ for such a vertex v . If $zv \in E$, then v is the next vertex of z in $\hat{\sigma}$ instead of x , since in this case $x <_{\hat{\sigma}} z <_{\hat{\sigma}} v$, which is a contradiction. If $zv \notin E$, then $v <_{\hat{\sigma}} z$ by

Lemma 5, since $\widehat{\sigma} = \text{RMN}(\widehat{\sigma})$, which is again a contradiction. Thus, $v <_{\widehat{\sigma}} z$ for every vertex v that is visited after z in $\widehat{\sigma}$. In the following we distinguish the cases $A \neq \emptyset$ and $A = \emptyset$. The case where $A = \emptyset$ can be handled similarly to the case where $A \neq \emptyset$, as we will see in the sequel.

Case 2a. $A \neq \emptyset$. Recall that $z <_{\widehat{\sigma}} w$ for every $w \in A$, i.e. every $w \in A$ is visited after z in $\widehat{\sigma}$, as we proved above. Thus, since x is the next vertex of z in $\widehat{\sigma}$ by assumption, all vertices $w \in A$ are unvisited, when x is being visited by $\widehat{\sigma}$. Therefore, the next vertex of x in $\widehat{\sigma}$ is some $w_1 \in A$, since $xw \in E$ and $y <_{\widehat{\sigma}} w$ for every $w \in A$. Now recall by the previous paragraph that $v <_{\widehat{\sigma}} z$ for every vertex v that is visited after z in $\widehat{\sigma}$, and that all vertices of B have been visited before z in $\widehat{\sigma}$. Furthermore recall that $yw \in E$ for every $y \in A$. Therefore, $\widehat{\sigma}$ visits after w_1 only vertices $w \in A$, until it reaches vertex y . Denote by P' the path on the vertices of $V(P)$ produced by $\widehat{\sigma}$. Suppose that not all vertices of A have been visited before y in P' , i.e. in $\widehat{\sigma}$. Then the next vertex of y in $\widehat{\sigma}$ is again some $w_2 \in A$. That is, $P' = (P_0, z, x, P_1, y, w_2, P_2)$ for some subpaths P_0 , P_1 , and P_2 of P' , where $V(P_1) \subseteq A$ and $w_2 \in A$. Thus, since $xw \in E$ for every $w \in A$, there exists the path $P'' = (P_0, z, d, y, P_1, x, w_2, P_2)$, where $V(P'') = V(P) \cup \{d\}$, which is a contradiction, since P is a maximal path.

Thus we may assume for the sequel that all vertices of A have been visited before y in P' , i.e. in $\widehat{\sigma}$. Then, $V(P_1) = A$. If y is the last vertex in $\widehat{\sigma}$, then $P' = (P_0, z, x, P_1, y)$ for some subpaths P_0 and P_1 of P' , where $V(P_1) = A$. In this case, there exists the path $P'' = (P_0, z, d, y, P_1, x)$, where $V(P'') = V(P) \cup \{d\}$, which is a contradiction, since P is a maximal path. Suppose that y is not the last vertex in $\widehat{\sigma}$ and denote by $q \notin A$ the next vertex of y in $\widehat{\sigma}$. Then, $P' = (P_0, z, x, P_1, y, q, P_2)$ for some subpaths P_0 , P_1 , and P_2 of P' , where $V(P_1) = A$, and let $P_1 = (w_1, w_2, \dots, w_\ell)$. If $w_\ell q \in E$, there exists the path $P'' = (P_0, z, d, y, x, P_1, q, P_2)$, which contradicts the maximality of P . If $xq \in E$, then there exists the path $P'' = (P_0, z, d, y, P_1, x, q, P_2)$, which again contradicts the maximality of P .

To complete the proof of Case 2a, we now assume that $w_\ell q, xq \notin E$. First we proof that $q <_{\widehat{\sigma}} x$. Otherwise, suppose that $y <_{\widehat{\sigma}} q$. Then $q <_{\widehat{\sigma}} z$, since $v <_{\widehat{\sigma}} z$ for every vertex v that is visited after z in $\widehat{\sigma}$, as we proved above, and thus $y <_{\widehat{\sigma}} q <_{\widehat{\sigma}} z$. Furthermore $q \notin B$, since all vertices of B have been visited before z in $\widehat{\sigma}$, as we proved above. Therefore $y \in A$, which is a contradiction, since we assumed that all vertices of A have been visited before y in $\widehat{\sigma}$. Now suppose $x <_{\widehat{\sigma}} q <_{\widehat{\sigma}} y$, i.e. $x <_{\widehat{\sigma}} q <_{\widehat{\sigma}} y <_{\widehat{\sigma}} w_\ell$. Then the vertices x, q, w_ℓ build an umbrella in $\widehat{\sigma}$, which is again a contradiction, since $\widehat{\sigma}$ is umbrella-free. Thus, $q <_{\widehat{\sigma}} x$.

Let s be a vertex, such that $x <_{\widehat{\sigma}} s <_{\widehat{\sigma}} y$ and s is visited after y in $\widehat{\sigma}$. Then, $s \neq q$, since $q <_{\widehat{\sigma}} x <_{\widehat{\sigma}} s$. If $ys \in E$, then s is the next vertex of y in $\widehat{\sigma}$ instead of q , since $\widehat{\sigma} = \text{RMN}(\widehat{\sigma})$, which is a contradiction. Thus $ys \notin E$ for every vertex s , such that $x <_{\widehat{\sigma}} s <_{\widehat{\sigma}} y$ and s is visited after y in $\widehat{\sigma}$.

We now construct a new ordering ρ of $V(P) \cup \{v\}$, where v is a new vertex. This new ordering ρ is based on the LDFS umbrella-free ordering $\widehat{\sigma}$ and the structure of ρ will allow us to show that G has a path on the vertices of $V(P) \cup \{d\}$, thereby contradicting the maximality of path P . The ordering ρ is constructed by adding the new vertex v immediately to the right of vertex y in $\widehat{\sigma}$. The adjacencies between the vertices of $V(P)$ in $\widehat{\sigma}$ remain the same in ρ , while the adjacencies between the new vertex v and the vertices of $V(P)$ in ρ are defined as follows. First, v is made adjacent in ρ to y and to all neighbors of y . Second, v is made adjacent also to z and to every vertex $w \in B$. Note that v is adjacent in ρ to all vertices w of $\widehat{\sigma}$, for which $v <_{\rho} w \leq_{\rho} z$. Therefore, if $wy \notin E$ and $wv \in E$ in ρ for some vertex $w \in V(P) \setminus \{y\}$, then $y <_{\widehat{\sigma}} w \leq_{\widehat{\sigma}} z$ (in particular, $w \in B$). Let H be the graph induced by the ordering ρ .

We will prove that ρ remains an LDFS umbrella free ordering of the vertices of $V(P) \cup \{v\}$. Since $G[V(P)]$ (i.e. the subgraph of G induced by $\widehat{\sigma}$) is an induced subgraph of H , if there is

an umbrella or a bad triple in ρ , then the new vertex v must belong to this umbrella or bad triple, since $\hat{\sigma} = \rho|_{V(P)}$ is an LDFS umbrella-free ordering of $V(P)$. Suppose that v belongs to an umbrella in ρ with vertices a, b, v , where either $v <_\rho a <_\rho b$, or $a <_\rho v <_\rho b$, or $a <_\rho b <_\rho v$.

Suppose first that $v <_\rho a <_\rho b$. Then, since $va \notin E$, it follows by the construction of ρ that $z <_\rho a$, i.e. $z <_\rho a <_\rho b$, and thus also $ya \notin E$ and $yb \in E$. That is, the vertices y, a, b build an umbrella in $\hat{\sigma}$, which is a contradiction. Suppose now that $a <_\rho v <_\rho b$. Then, $a \neq y$, since v is adjacent to y in ρ . Thus, since $va \notin E$, it follows by the construction of ρ that also $ay \notin E$. Furthermore, since $vb \notin E$, it follows by the construction of ρ that $z <_\rho b$, and thus also $yb \notin E$. That is, the vertices a, y, b build an umbrella in $\hat{\sigma}$, which is a contradiction. Suppose finally that $a <_\rho b <_\rho v$. Then, $b \neq y$, since v is adjacent to y in ρ . Furthermore, $a \neq y$, since y lies immediately to the left of v in ρ . Thus, since $av \in E$ and $bv \notin E$, it follows by the construction of ρ that also $ay \in E$ and $by \notin E$, i.e. the vertices a, b, y build an umbrella in $\hat{\sigma}$, which is a contradiction. Thus, ρ is umbrella-free.

Suppose now that v belongs to a bad triple in ρ with vertices a, b, v , where either $v <_\rho a <_\rho b$ or $a <_\rho v <_\rho b$ or $a <_\rho b <_\rho v$. First let $v <_\rho a <_\rho b$, where $vb \in E$ and $va \notin E$. Since $va \notin E$, it follows by the construction of ρ that $z <_{\hat{\sigma}} a <_{\hat{\sigma}} b$, and thus also $y <_{\hat{\sigma}} a <_{\hat{\sigma}} b$, $yb \in E$, and $ya \notin E$. Since $\hat{\sigma}$ is an LDFS ordering, there exists a vertex v' between y and a in $\hat{\sigma}$, such that $v'a \in E$ and $v'b \notin E$. Note that $v' \neq v$, since $vb \in E$ and $v'b \notin E$. Thus, the vertices v, a, b do not build a bad triple in ρ , which is a contradiction. Now let $a <_\rho v <_\rho b$, where $ab \in E$ and $av \notin E$. Note that $a \neq y$, since $av \notin E$, and thus also $a <_{\hat{\sigma}} y <_{\hat{\sigma}} b$ and $ay \notin E$. Since $\hat{\sigma}$ is an LDFS ordering, there exists a vertex v' between a and y in $\hat{\sigma}$, such that $v'b \notin E$ and $v'y \in E$, and thus also $v'v \in E$. Thus, the vertices a, v, b do not build a bad triple in ρ , which is a contradiction. Finally let $a <_\rho b <_\rho v$, where $av \in E$ and $ab \notin E$. By the construction of ρ , note that $b \neq y$, since $av \in E$ and $ab \notin E$. Thus $a <_{\hat{\sigma}} b <_{\hat{\sigma}} y$. Furthermore, $ay \in E$ by the construction of ρ , since $av \in E$. Since $\hat{\sigma}$ is an LDFS ordering, there exists a vertex v' between a and b in $\hat{\sigma}$, such that $v'b \in E$ and $v'y \notin E$, and thus also $v'v \notin E$. Thus, the vertices a, b, v do not build a bad triple in ρ , which is a contradiction. Summarizing, ρ is an LDFS umbrella-free ordering.

Since $\hat{\sigma}$ is an LDFS umbrella-free ordering of the vertices of a path P , the ordering $\hat{\hat{\sigma}} = \text{RMN}(\hat{\sigma})$ gives a Hamiltonian path P' of the subgraph of G induced by $V(P)$ [6]. Recall that $P' = (P_0, z, x, P_1, y, q, P_2)$ for some subpaths P_0, P_1 , and P_2 of P' , where $V(P_1) = A$. Thus, the graph H induced by the ordering ρ of the vertices of $V(P) \cup \{v\}$ is again Hamiltonian, since we can just insert in P' the new vertex v of ρ between z and x . Therefore, since ρ is an LDFS umbrella-free ordering, the ordering $\hat{\rho} = \text{RMN}(\rho)$ gives a Hamiltonian path of H [6], i.e. of the graph induced by ρ . We will compare now the orderings $\hat{\hat{\sigma}}$ and $\hat{\rho}$.

First, we will prove that both orderings $\hat{\hat{\sigma}}$ and $\hat{\rho}$ coincide until vertex z is visited. Indeed, since $\hat{\hat{\sigma}}$ and $\hat{\rho}$ differ only at the vertex v , the only difference of these orderings before z is visited could be that v is visited before z in $\hat{\rho}$. Suppose that v is visited before z in $\hat{\rho}$. Note that the first vertex of the ordering $\hat{\rho} = \text{RMN}(\rho)$ is the rightmost vertex of ρ . Therefore, v is not the first vertex of $\hat{\rho}$, since $v <_\rho z$. Let a be the previous vertex of v in $\hat{\rho}$. Then, a is adjacent to v in ρ , since $\hat{\rho}$ is a path. If a is adjacent to z in ρ , then z is the next vertex of a in $\hat{\rho}$ instead of v , since $v <_\rho z$, which is a contradiction. Thus, a is not adjacent to z in both ρ and $\hat{\sigma}$. Note that both orderings $\hat{\hat{\sigma}}$ and $\hat{\rho}$ coincide at least until the visit of a , which is visited before z in both $\hat{\hat{\sigma}}$ and $\hat{\rho}$, and thus $a \neq y$. If $a <_\rho y$ or $z <_\rho a$, it follows by the construction of ρ that a is adjacent also to y in $\hat{\sigma}$. Thus, since v is the next vertex of a in $\hat{\rho} = \text{RMN}(\rho)$, it follows that y is the next vertex of a in $\hat{\hat{\sigma}} = \text{RMN}(\hat{\sigma})$, i.e. that y is visited before z in $\hat{\hat{\sigma}}$, which is a contradiction. Suppose that $v <_\rho a <_\rho z$. Then, since $az \notin E$, it follows that $a \in A$, and thus $ay \in E$ in the ordering $\hat{\sigma}$, as we proved above. Therefore, since v is the next vertex of a in $\hat{\rho}$, it follows that y is the next vertex of a in $\hat{\hat{\sigma}}$, i.e. that y is visited before z in $\hat{\hat{\sigma}}$, which is a contradiction. Therefore, v is not

visited before z in $\hat{\rho}$, and thus both orderings $\hat{\sigma}$ and $\hat{\rho}$ coincide until vertex z is visited.

Now, v is the rightmost unvisited neighbor of z in ρ at the time that vertex z is being visited by $\hat{\rho}$, since by our initial assumption x is the next vertex of z in $\hat{\sigma}$. Furthermore, similarly to $\hat{\sigma}$, the ordering $\hat{\rho}$ visits the vertices of P_1 after v , where $V(P_1) = A$. In the sequel, after visiting all vertices of P_1 , $\hat{\rho}$ visits y as the rightmost unvisited neighbor of the last vertex of P_1 . Recall that $ys \notin E$ for every unvisited vertex s , such that $x <_{\hat{\sigma}} s <_{\hat{\sigma}} y$, and that the next vertex of y in $\hat{\sigma}$ is $q <_{\hat{\sigma}} x$. Therefore, x is the rightmost unvisited neighbor of y in ρ at the time that y is being visited by $\hat{\rho}$, and thus $\hat{\rho}$ visits x after y . Summarizing, the Hamiltonian path P_{ρ} of the graph H (i.e. the graph induced by ρ) that is computed by $\hat{\rho}$ is $P_{\rho} = (P_0, z, v, P_1, y, x, Q)$ for some subpath Q of P_{ρ} , where $P' = (P_0, z, x, P_1, y, q, P_2)$. Note that $V(Q) = V(P_2) \cup \{q\}$, since $V(P_{\rho}) = V(P') \cup \{v\} = V(P) \cup \{v\}$. Furthermore, note that Q is also a path of $G[V(P)]$, since $v \notin V(Q)$. Then, there exists the path $P'' = (P_0, z, d, y, P_1, x, Q)$ of G , where $V(P'') = V(P) \cup \{d\}$, which is a contradiction, since P is a maximal path.

Case 2b. $A = \emptyset$. Then y is the next vertex of x in $\hat{\sigma}$, since $xy \in E$ and all vertices to the right of y in $\hat{\sigma}$ have been already visited before x in $\hat{\sigma}$. That is, the path P' of the vertices of $V(P)$ constructed by $\hat{\sigma}$ is $P' = (P_0, z, x, y, P_3)$, for some subpaths P_0 and P_3 of P' . Consider the ordering ρ , which obtained by adding a new vertex v to $\hat{\sigma}$, as described in Case 2a. Then, similarly to Case 2a, the graph H induced by ρ is Hamiltonian and the ordering $\hat{\rho} = \text{RMN}(\rho)$ gives a Hamiltonian path P_{ρ} of H , where $P_{\rho} = (P_0, z, v, y, Q)$. Note that $V(Q) = V(P_3) \cup \{x\}$ and that Q is also a path of $G[V(P)]$, since $v \notin V(Q)$. Thus, there exists the path $P'' = (P_0, z, d, y, Q)$ of G , where $V(P'') = V(P) \cup \{d\}$, which is a contradiction, since P is a maximal path. This completes the proof of the lemma. ■

The next lemma now follows by Lemma 7.

Lemma 8 *Let x be the rightmost vertex in σ' and y be the rightmost neighbor of x in σ' . Then, x is the last vertex of $\hat{\sigma}$ and y is the previous vertex of x in $\hat{\sigma}$.*

Proof. First note that, if σ' has at least two vertices, x is not the first vertex of $\hat{\sigma}$, since $\hat{\sigma} = \text{RMN}(\hat{\sigma})$ and x is the leftmost vertex of $\hat{\sigma}$. Suppose that x is not the last vertex of $\hat{\sigma}$, i.e. x is an intermediate vertex. Let a and b be the previous and the next vertices of x in $\hat{\sigma}$, respectively. Then, $a <_{\hat{\sigma}} b$. If $ab \in E$, then b is the next vertex of a in $\hat{\sigma}$ instead of x , since $x <_{\hat{\sigma}} a$, which is a contradiction. Therefore $ab \notin E$, and thus $b <_{\hat{\sigma}} a$ by Lemma 5, since $a <_{\hat{\sigma}} b$. Furthermore $a <_{\sigma'} b$ by Lemma 5, since $b <_{\hat{\sigma}} a$ and $ab \notin E$, and thus $a <_{\sigma'} b <_{\sigma'} x$, since x is the rightmost vertex in σ' . That is, $x <_{\hat{\sigma}} b <_{\hat{\sigma}} a$ and $a <_{\sigma'} b <_{\sigma'} x$, where $xb, xa \in E$ and $ba \notin E$, while x is the next vertex of a in $\hat{\sigma}$, which is a contradiction by Lemma 7. Therefore, x is the last vertex of $\hat{\sigma}$.

Note now that y is the second leftmost vertex in $\hat{\sigma}$, since x is the rightmost vertex of σ' and $\hat{\sigma} = \text{LDFS}^+(\sigma')$. Suppose that y is not the previous vertex of x in $\hat{\sigma}$ and let $a \neq y$ be the previous vertex of x in $\hat{\sigma}$. Then, $x <_{\hat{\sigma}} y <_{\hat{\sigma}} a$ and $xy, xa \in E$. Furthermore, y has been visited before a in $\hat{\sigma}$, i.e. $y <_{\hat{\sigma}} a$, since x is the last vertex of $\hat{\sigma}$. Suppose that $ya \notin E$. Then, since $y <_{\hat{\sigma}} a$, it follows by Lemma 5 that $a <_{\hat{\sigma}} y$, which is a contradiction, since $y <_{\hat{\sigma}} a$. Therefore $ya \in E$. Thus, since $y <_{\hat{\sigma}} a$ and $y <_{\hat{\sigma}} a$, Lemma 6 implies that y is not the first vertex of $\hat{\sigma}$ and that $y <_{\hat{\sigma}} a <_{\hat{\sigma}} z$, $yz \in E$, and $az \notin E$ for the previous vertex z of y in $\hat{\sigma}$. Furthermore $z <_{\sigma'} a$ by Lemma 5, since $a <_{\hat{\sigma}} z$ and $az \notin E$. On the other hand $a <_{\sigma'} y$, since $xa \in E$ and y is the rightmost neighbor of x in σ' . That is, $y <_{\hat{\sigma}} a <_{\hat{\sigma}} z$ and $z <_{\sigma'} a <_{\sigma'} y$, where $ya, yz \in E$ and $az \notin E$, while y is the next vertex of z in $\hat{\sigma}$, which is a contradiction by Lemma 7. Therefore, y is the previous vertex of x in $\hat{\sigma}$. ■

The next corollary follows easily by Definition 5(a) and Lemma 8.

Corollary 2 *Let $G = (V, E)$ be a cocomparability graph, σ be an LDFS umbrella-free ordering of G , and P be a maximal path of G . Then there exists a typical path P' of G , such that $V(P') = V(P)$.*

Proof. Consider the restriction $\sigma' = \sigma|_{V(P)}$ of σ on the vertices of P ; note that σ' is an induced subordering of σ . Furthermore, consider the orderings $\hat{\sigma} = \text{LDFS}^+(\sigma')$ and $\hat{\hat{\sigma}} = \text{RMN}(\hat{\sigma})$ (cf. Notation 1). Note that, since σ' is an ordering of the vertices of $V(P)$, the ordering $\hat{\hat{\sigma}}$ defines a minimum path cover of $G[V(P)]$ [6]. Therefore, since $G[V(P)]$ has P as a Hamiltonian path, it follows that the ordering $\hat{\hat{\sigma}}$ defines a single path Q on the vertices of $V(P)$ (note that this path Q may be P itself or a different path on the same vertices). Let now x be the rightmost vertex in σ' and y be the rightmost neighbor of x in σ' . Then, since $\hat{\hat{\sigma}}$ defines the path Q , Lemma 8 implies that x is the last vertex of Q and y is the previous vertex of x in Q . Therefore, the reverse path P' of Q is a typical path of G with $V(P') = V(P)$. ■

We are now ready to present the main theorem of this section.

Theorem 2 *Let $G = (V, E)$ be a cocomparability graph, σ be an LDFS umbrella-free ordering of G , and P be a maximal path of G . Then there exists a normal path P' of G , such that $V(P') = V(P)$.*

Proof. Let the maximal path P be denoted (v_1, v_2, \dots, v_k) . If $k \leq 2$, the lemma clearly holds. Suppose in the sequel that $k \geq 3$ and that there exists no normal path P' of G , such that $V(P') = V(P)$. We may assume without loss of generality that G has the smallest number of vertices among all cocomparability graphs that have such a maximal path P . Furthermore, we may assume by Corollary 2 that P is typical, i.e. that v_1 is the rightmost vertex of $V(P)$ in σ and that v_2 is the rightmost vertex of $N(v_1) \cap \{v_2, v_3, \dots, v_k\}$ in σ .

Let $i \in \{2, 3, \dots, k-1\}$ be the greatest index, such that v_j is the rightmost vertex of $N(v_{j-1}) \cap \{v_j, v_{j+1}, \dots, v_k\}$ in σ for every $j = 2, \dots, i$. Such an index i exists by the assumption that there exists no normal path P' of G , for which $V(P') = V(P)$. Let $P_1 = (v_1, v_2, \dots, v_i)$ and $P_2 = (v_{i+1}, v_{i+2}, \dots, v_k)$ be the subpaths of P until the vertex v_i and after the vertex v_i , respectively. Then, in particular, P_1 is normal by the assumption on i , i.e. P_1 has the first i vertices of an RMN when applied on the restriction $\sigma|_{V(P)}$ of the ordering σ on the vertices of P . We will construct a path $P^* = (v_1^*, v_2^*, \dots, v_k^*)$, such that $V(P^*) = V(P)$, v_1^* is the rightmost vertex of $V(P^*)$ in σ , and v_ℓ^* is the rightmost vertex of $N(v_{\ell-1}^*) \cap \{v_\ell^*, v_{\ell+1}^*, \dots, v_k^*\}$ in σ for every $\ell = 2, \dots, i+1$, thus arriving to a contradiction by the assumption on the index i .

Consider a vertex $v_\ell \in \{v_{i+1}, v_{i+2}, \dots, v_k\}$, such that $v_i <_\sigma v_\ell$. Then, $v_i <_\sigma v_\ell <_\sigma v_1$, since P is typical. We will prove that $v_i v_\ell \in E$. Suppose otherwise that $v_i v_\ell \notin E$. Then, since $v_\ell \notin V(P_1)$, it follows by Lemma 3 that there exist two consecutive vertices v_{j-1} and v_j in P_1 , where $2 \leq j \leq i$, such that $v_{j-1} v_\ell \in E$ and $v_j <_\pi v_\ell$. Thus, v_j is not the rightmost vertex of $N(v_{j-1}) \cap \{v_j, v_{j+1}, \dots, v_k\}$ in σ , which is a contradiction. Therefore, $v_i v_\ell \in E$ for every $v_\ell \in \{v_{i+1}, v_{i+2}, \dots, v_k\}$, such that $v_i <_\sigma v_\ell$.

In the sequel let v_j be the rightmost vertex of $N(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_k\}$ in σ , where $j > i+1$ by the assumption on the index i . Now we distinguish the cases where $v_i <_\sigma v_j$ and $v_j <_\sigma v_i$.

Case 1. $v_i <_\sigma v_j$. Suppose that there exists a vertex $v_\ell \in \{v_{i+1}, v_{i+2}, \dots, v_k\}$, such that $v_j <_\sigma v_\ell$. Then, as we proved above, $v_i v_\ell \in E$, which is a contradiction, since v_j is the rightmost vertex of $N(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_k\}$ in σ and $v_j <_\sigma v_\ell$. Thus, v_j is the rightmost vertex of $\{v_{i+1}, v_{i+2}, \dots, v_k\}$ in σ . Let σ' be the induced subordering of σ on the vertices of $V(P_2) = \{v_{i+1}, v_{i+2}, \dots, v_k\}$, and σ'' be an LDFS closure of σ' (within σ). Then, by definition $V(P_1) \cap V(\sigma') = \emptyset$. Furthermore, v_j is the rightmost vertex in σ' , and thus v_j remains the rightmost vertex in σ'' by Corollary 1.

First we will prove that $V(P_1) \cap V(\sigma'') = \emptyset$. Suppose otherwise that $V(P_1) \cap V(\sigma'') \neq \emptyset$, and let v be the rightmost vertex of $V(P_1) \cap V(\sigma'')$ in σ . Then $v \in V(\sigma'' \setminus \sigma')$, since $V(P_1) \cap V(\sigma') = \emptyset$. Thus, there exists by Lemma 4 at least one vertex v' in σ' , such that $v <_\sigma v'$ and $vv' \notin E$. Then, $v' \in V(P_2) \subset V(P)$ by definition of the ordering σ' , i.e. v' is a vertex of $\sigma|_{V(P)}$. Furthermore, since $v <_\sigma v'$ and $vv' \notin E$, Lemma 5 implies that v' is visited before v in P_1 (that is, by applying RMN on $\sigma|_{V(P)}$), i.e. $v' \in V(P_1)$, which is a contradiction, since $v' \in V(P_2)$. Thus, $V(P_1) \cap V(\sigma'') = \emptyset$.

Now we will prove that the subpath $P_2 = (v_{i+1}, v_{i+2}, \dots, v_k)$ of P is maximal in $G|_{\sigma''}$. Indeed, suppose otherwise that P_2 is not maximal in $G|_{\sigma''}$, i.e. there exists a path P'_2 of $G|_{\sigma''}$, such that $V(P_2) \subset V(P'_2)$. Thus, since $G|_{\sigma''}$ has strictly fewer vertices than G , there exists (by the assumption on G) a normal path P''_2 of $G|_{\sigma''}$, such that $V(P'_2) = V(P''_2)$. Therefore, in particular, P''_2 has strictly more vertices than P_2 and v_j is the first vertex of P''_2 . Thus, since $v_i v_j \in E$, the path $(v_1, v_2, \dots, v_i, P''_2)$ of G has strictly more vertices than P , which is a contradiction to the assumption that P is maximal. Therefore, the subpath P_2 of P is maximal in $G|_{\sigma''}$, and thus there exists a normal path Q of $G|_{\sigma''}$, such that $V(Q) = V(P_2)$. Then, in particular, v_j is the first vertex of Q , and thus

$$P^* = (v_1^*, v_2^*, \dots, v_k^*) = (v_1, v_2, \dots, v_i, Q) \quad (1)$$

is as requested.

Case 2. $v_j <_\sigma v_i$. Consider an arbitrary vertex $v_\ell \in \{v_{i+1}, v_{i+2}, \dots, v_k\}$ and suppose that $v_i <_\sigma v_\ell$. Then, $v_\ell \neq v_j$, since $v_j <_\sigma v_i$. Furthermore, as we proved above, $v_i v_\ell \in E$, which is a contradiction, since v_j is the rightmost vertex of $N(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_k\}$ in σ and $v_j <_\sigma v_i <_\sigma v_\ell$. Therefore, $v_\ell <_\sigma v_i$ for every $v_\ell \in \{v_{i+1}, v_{i+2}, \dots, v_k\}$, i.e. v_i is the rightmost vertex of $V(P_2) \cup \{v_i\} = \{v_i, v_{i+1}, \dots, v_k\}$ in σ . Consider the induced subordering σ' of σ on the vertices $V(P_2) \cup \{v_i\}$ and an LDFS closure σ'' of σ' (within σ). Then, similarly to Case 1, the subpath $(v_i, P_2) = (v_i, v_{i+1}, \dots, v_k)$ of P is a maximal path of $G|_{\sigma''}$, and thus there exists a normal path Q of $G|_{\sigma''}$, such that $V(Q) = \{v_i, v_{i+1}, \dots, v_k\}$. Then, in particular, v_i is the first vertex of Q , and thus

$$P^* = (v_1^*, v_2^*, \dots, v_k^*) = (v_1, v_2, \dots, v_{i-1}, Q) \quad (2)$$

is as requested. This completes the proof of the lemma. ■

4 The longest path problem on cocomparability graphs

In this section we present the first polynomial algorithm that computes a longest path of a cocomparability graph G . This dynamic programming algorithm is based on Theorem 2; in particular, this algorithm computes a *longest normal* path of G . For the rest of this section we consider an LDFS umbrella-free ordering σ of a given cocomparability graph $G = (V, E)$, which can be obtained by executing an LDFS⁺ on an arbitrary umbrella-free ordering π of G [6]. We consider that the vertices of V , where $|V| = n$, are numbered in σ increasingly from left to right, i.e. $\sigma = (u_1, u_2, \dots, u_n)$. Furthermore, for simplicity of the presentation, we add to σ a dummy isolated vertex u_{n+1} to the right of all other vertices of V , i.e. we consider without loss of generality that $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$. It is easy to see that σ remains an LDFS umbrella-free ordering after the addition of the dummy vertex u_{n+1} .

Definition 6 Let $G = (V, E)$ be a cocomparability graph with $|V| = n$ and let $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. For every pair of indices $i, j \in \{1, 2, \dots, n\}$,

- if $i > j$, then $G(i, j) = \emptyset$,

- if $i \leq j$, then $G(i, j)$ is the subgraph $G[S]$ of G induced by the vertex set $S = \{u_i, u_{i+1}, \dots, u_j\} \setminus N(u_{j+1})$.

It is easy to see by Definition 6 that for every pair of indices $i, j \in \{1, 2, \dots, n\}$, the vertices u_i and u_j may or may not belong to $G(i, j)$, since they may or may not be adjacent to u_{j+1} in G . Furthermore note that $G(1, n) = G$ and that $G(i, n) = G[\{u_i, u_{i+1}, \dots, u_n\}]$ for every $i \in \{1, 2, \dots, n\}$, since u_{n+1} is an isolated vertex.

As an example of Definition 6, the subgraph $G(3, 8)$ of the cocomparability graph G of Figure 3 is illustrated in Figure 4. In this figure the dummy isolated vertex u_{10} is also depicted, while the vertices $V(G(3, 8)) = \{u_3, u_4, u_5, u_6, u_8\}$ of $G(3, 8)$, as well as the edges of $G(3, 8)$, are drawn darker than the others for better visibility. Furthermore, note that the path $P = (u_8, u_5, u_6, u_3, u_4)$ is a normal path of $G(3, 8)$.

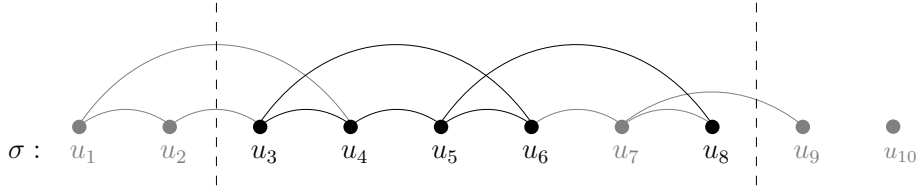


Figure 4: The subgraph $G(3, 8)$ of the cocomparability graph G of Figure 3.

Observation 4 For every pair of indices $i, j \in \{1, 2, \dots, n\}$, $G(i + 1, j) = G(i, j) \setminus \{u_i\}$.

Observation 5 Let $P = (P_1, u_i)$ be a normal path of $G(i, j)$, for some pair of indices $i, j \in \{1, 2, \dots, n\}$. Then P_1 is a normal path of both $G(i + 1, j)$ and $G(i, j)$.

Observation 6 Let $P_1 = (P_0, u_x)$ be a normal path of $G(i + 1, j)$, for some pair of indices $i, j \in \{1, 2, \dots, n\}$, and let $u_i \in V(G(i, j))$ and $u_x \in N(u_i)$. Then $P = (P_1, u_i)$ is a normal path of $G(i, j)$.

Lemma 9 Let $G = (V, E)$ be a cocomparability graph and $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. Suppose that $u_i <_\sigma u_x$ and $u_x \in N(u_i)$. Then $u_k \in N(u_i)$ for every $u_k \in V(G(i + 1, x - 1))$.

Proof. Let $u_k \in V(G(i + 1, x - 1))$. Then $u_i <_\sigma u_k <_\sigma u_x$ and $u_k \notin N(u_x)$ by Definition 6. Therefore, since σ is an umbrella-free ordering and $u_x \in N(u_i)$ by assumption, it follows that $u_k \in N(u_i)$. ■

Lemma 10 Let $G = (V, E)$ be a cocomparability graph and $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. Then $V(G(i + 1, x - 1)) \subseteq V(G(i, j))$ for every $u_x \in V(G(i + 1, j))$.

Proof. Consider a vertex $u_y \in V(G(i + 1, x - 1))$. Then, since also $u_x \in V(G(i + 1, j))$, it follows by Definition 6 that $u_y \notin N(u_x)$ and $u_x \notin N(u_{j+1})$. Suppose that $u_y \in N(u_{j+1})$. Then, since $u_y <_\sigma u_x <_\sigma u_{j+1}$, the vertices u_y, u_x, u_{j+1} build an umbrella in σ , which is a contradiction. Therefore $u_y \notin N(u_{j+1})$, and thus $u_y \in V(G(i, j))$ by Definition 6. ■

In the following we state two lemmas that are crucial for the proof of the main Theorem 3 of this section.

Lemma 11 *Let $G = (V, E)$ be a cocomparability graph and $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. Let $u_i \in V(G(i, j))$, $u_x \in V(G(i+1, j))$, $u_y \in V(G(i+1, x-1))$, and $u_x \in N(u_i)$. Furthermore, let P_1 be a normal path of $G(i+1, j)$ with u_x as its last vertex and P_2 be a normal path of $G(i+1, x-1)$ with u_y as its last vertex. Then $P = (P_1, u_i, P_2)$ is a normal path of $G(i, j)$ with u_y as its last vertex.*

Proof. We will first prove that $V(P_1) \subseteq V(G(i+1, j)) \setminus V(G(i+1, x-1))$. Suppose otherwise that $V(P_1) \cap V(G(i+1, x-1)) \neq \emptyset$, and let u_k be the first vertex of P_1 , such that $u_k \in V(G(i+1, x-1))$. Then u_k is not the rightmost vertex of P_1 in σ , since $u_k <_\sigma u_x$. Therefore, since P_1 is a normal path by assumption, u_k is not the first vertex of P_1 , and thus there exists a previous vertex u_ℓ of u_k in P_1 , i.e. $u_\ell \in N(u_k)$. Suppose first that $u_\ell \in N(u_x)$. Then, since $u_k <_\sigma u_x$ and u_x is unvisited by P_1 when u_ℓ is visited, it follows that u_k is not the rightmost unvisited vertex of $N(u_\ell) \cap V(P_1)$ in σ , when P_1 visits u_ℓ . This is a contradiction by Definition 5, since u_k is the next vertex of u_ℓ in P_1 and P_1 is a normal path by assumption. Suppose now that $u_\ell \notin N(u_x)$. Let $u_\ell <_\sigma u_x$. Then $u_\ell \in V(G(i+1, x-1))$ by Definition 6. This is a contradiction to the assumption that u_k is the first vertex of P_1 , such that $u_k \in V(G(i+1, x-1))$. Let $u_x <_\sigma u_\ell$, i.e. $u_k <_\sigma u_x <_\sigma u_\ell$. Note that $u_k \notin N(u_x)$ by Definition 6, since $u_k \in V(G(i+1, x-1))$. Thus the vertices u_k, u_x, u_ℓ build an umbrella in σ , since $u_\ell \in N(u_k)$, $u_k \notin N(u_x)$, and $u_\ell \notin N(u_x)$, which is a contradiction. Therefore $V(P_1) \cap V(G(i+1, x-1)) = \emptyset$, i.e. $V(P_1) \subseteq V(G(i+1, j)) \setminus V(G(i+1, x-1))$.

Since $V(P_1) \subseteq V(G(i+1, j)) \setminus V(G(i+1, x-1))$ by the previous paragraph and $V(P_2) \subseteq V(G(i+1, x-1))$ by assumption, it follows that $V(P_1) \cap V(P_2) = \emptyset$. Recall now that $u_k \in N(u_i)$ for every $u_k \in V(P_2) \subseteq V(G(i+1, x-1))$ by Lemma 9. Furthermore, recall that $V(P_1) \subseteq V(G(i+1, j)) \subseteq V(G(i, j))$ by Observation 4 and that $V(P_2) \subseteq V(G(i+1, x-1)) \subseteq V(G(i, j))$ by Lemma 10. Therefore, since $u_i \in V(G(i, j))$ and $u_x \in N(u_i)$ by assumption, it follows that $P = (P_1, u_i, P_2)$ is a path of $G(i, j)$. Moreover u_y is the last vertex of P , since u_y is the last vertex of P_2 by assumption.

In the following we prove that P is normal. To this end, first let $\sigma_1 = \sigma|_{P_1}$ be the restriction of the ordering σ on the vertices of the path P_1 and let $\sigma'_1 = \text{RMN}(\sigma_1)$. Then the ordering of the vertices of $V(P_1)$ in P_1 coincides with the ordering σ'_1 by Observation 3. Note that σ_1 is an umbrella-free ordering, as a restriction of the umbrella-free ordering σ .

Note now that the first vertex u_ℓ of P is also the first vertex of P_1 , since $P = (P_1, u_i, P_2)$. Moreover, u_ℓ is the rightmost vertex of P_1 in σ , since P_1 is normal by assumption. Furthermore, note that $u_k <_\sigma u_x \leq_\sigma u_\ell$ for every $u_k \in V(P_2) \cup \{u_i\}$. Therefore, u_ℓ is also the rightmost vertex of P in σ . Let u_r and $u_{r'}$ be two consecutive vertices of P_1 , i.e. $u_{r'}$ is the rightmost unvisited vertex of $N(u_r) \cap V(P_1)$ in σ , when P_1 visits u_r . We will prove that $u_{r'}$ is also the rightmost unvisited vertex of $N(u_r) \cap V(P)$ in σ , when P visits u_r . Suppose otherwise that $u_k \neq u_{r'}$ is the rightmost unvisited vertex of $N(u_r) \cap V(P)$ in σ , when P visits u_r . Then in particular $u_{r'} <_\sigma u_k$ and $u_k \in N(u_r)$. If $u_k \in V(P_1)$, then u_k would be also the rightmost unvisited vertex of $N(u_r) \cap V(P_1)$ in σ , when P_1 visits u_r , which is a contradiction.

Therefore $u_k \in V(P_2) \cup \{u_i\} \subseteq \{u_i, u_{i+1}, \dots, u_{x-1}\}$, and thus in particular $u_k <_\sigma u_x$. Suppose that $u_r \in N(u_x)$. Then, since $u_k <_\sigma u_x$ and u_x is unvisited when P visits u_r , it follows that u_k is not the rightmost unvisited vertex of $N(u_r) \cap V(P)$ in σ when P visits u_r , which is a contradiction to the assumption on u_k . Thus $u_r \notin N(u_x)$. Recall that u_x is the last vertex of P_1 by assumption. Therefore, u_r appears before u_x in P_1 , and thus $u_r <_{\sigma'_1} u_x$ as we proved above, where $\sigma_1 = \sigma|_{P_1}$ and $\sigma'_1 = \text{RMN}(\sigma_1)$. Therefore, since $u_r \notin N(u_x)$ and σ_1 is an umbrella-free ordering, it follows by Lemma 5 that $u_x <_{\sigma_1} u_r$, i.e. $u_x <_\sigma u_r$. That is, $u_k <_\sigma u_x <_\sigma u_r$. Recall that $u_k \in V(P_2) \cup \{u_i\}$. First let $u_k \in V(P_2) \subseteq V(G(i+1, x-1))$. Then $u_k \notin N(u_x)$ by Definition 6. Therefore, since also $u_r \notin N(u_x)$ and $u_k \in N(u_r)$, the vertices u_k, u_x, u_r

build an umbrella in σ , which is a contradiction. Now let $u_k = u_i$. Then $u_k = u_i <_\sigma u_{r'}$, since $u_{r'} \in V(P_1) \subseteq V(G(i+1, j))$. Thus u_k is not the rightmost unvisited vertex of $N(u_r) \cap V(P)$ in σ , when P visits u_r , which is a contradiction to the assumption on u_k . Therefore, for any two consecutive vertices $u_r, u_{r'}$ of P_1 , $u_{r'}$ is the rightmost unvisited vertex of $N(u_r) \cap V(P)$ in σ , when P visits u_r .

Recall that $V(P_2) \subseteq V(G(i+1, x-1))$ by assumption, and thus $u_k \notin N(u_x)$ for every vertex $u_k \in V(P_2)$. Therefore, u_i is the rightmost unvisited vertex of $N(u_x) \cap V(P)$ in σ , when P visits u_x (i.e. the last vertex of P_1). Note that exactly the vertices of $V(P_2)$ are the unvisited vertices of $V(P)$, when P visits u_i . Moreover, recall that P_2 is a normal path and that $u_k \in N(u_i)$ for every $u_k \in V(P_2) \subseteq V(G(i+1, x-1))$ by Lemma 9. Therefore, the first vertex of P_2 is also the rightmost unvisited vertex of $N(u_i) \cap V(P)$ in σ , when P visits u_i . Consider now any pair of consecutive vertices $u_r, u_{r'}$ of P_2 . Then, $u_{r'}$ is the rightmost unvisited vertex of $N(u_r) \cap V(P_2)$ in σ (resp. of $N(u_r) \cap V(P)$ in σ), when P_2 (resp. P) visits u_r . Therefore, P is a normal path. This completes the proof of the lemma. ■

Notation 2 Let $G = (V, E)$ be a cocomparability graph and $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. Let $i, j \in \{1, 2, \dots, n\}$ be a pair of indices, let $u_k \in V(G(i, j))$, and let P be a normal path of $G(i, j)$. For simplicity of presentation, we will say in the following that “ P is a longest normal path of $G(i, j)$ with u_k as its last vertex” if P has the greatest number of vertices among those normal paths of $G(i, j)$ that have u_k as their last vertex.

Lemma 12 Let $G = (V, E)$ be a cocomparability graph and $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. Let P be a longest normal path of $G(i, j)$ with $u_y \neq u_i$ as its last vertex and let $P = (P_1, u_i, P_2)$. Let u_x be the last vertex of P_1 . Then P_1 is a longest normal path of $G(i+1, j)$ with u_x as its last vertex and P_2 is a longest normal path of $G(i+1, x-1)$ with u_y as its last vertex.

Proof. Note that P has at least two vertices, since $u_y, u_i \in V(P)$. Therefore, since $u_i <_\sigma u_k$ for every $u_k \in V(P) \setminus \{u_i\}$, it follows that u_i is not the first vertex of P , and thus $P_1 \neq \emptyset$. Note that $V(P_1) \subseteq V(G(i+1, j))$, i.e. $V(P_1) \subseteq V(G(i, j)) \setminus \{u_i\}$ by Observation 4, since $u_i \notin V(P_1)$. Furthermore, since P is a normal path by assumption and P_1 is a subpath of P , it follows that P_1 is a normal path of $G(i+1, j)$ with u_x as its last vertex.

Let $\sigma' = \sigma|_P$ be the restriction of the ordering σ on the vertices of the path P and let $\sigma'' = \text{RMN}(\sigma')$. Then, since P is a normal path by assumption, the ordering of the vertices of $V(P)$ in P coincides with the ordering σ'' by Observation 3.

We will now prove that $V(P_2) \subseteq V(G(i+1, x-1))$. Consider an arbitrary vertex $u_k \in V(P_2)$ and note that $u_i <_\sigma u_k$. Note that both u_i and u_k are unvisited by P when u_x is visited. Suppose that $u_k \in N(u_x)$. Then, since $u_i <_\sigma u_k$, it follows that u_i is not the rightmost unvisited vertex of $N(u_x) \cap V(P)$ in σ , when P visits u_x . Thus, since P is normal by assumption, it follows that u_i is not the next vertex of u_x in P , which is a contradiction. Therefore $u_k \notin N(u_x)$ for every $u_k \in V(P_2)$. Recall by the previous paragraph that the ordering of the vertices of $V(P)$ in P coincides with the ordering $\sigma'' = \text{RMN}(\sigma')$, where $\sigma' = \sigma|_P$. Therefore, since $u_k \in V(P_2)$ appears after u_x in P , it follows that $u_x <_{\sigma''} u_k$. Thus, since $u_k \notin N(u_x)$, Lemma 5 implies that $u_k <_{\sigma'} u_x$, i.e. $u_k <_\sigma u_x$. Summarizing, $u_k \notin N(u_x)$ and $u_i <_\sigma u_k <_\sigma u_x$ for every $u_k \in V(P_2)$, and thus $V(P_2) \subseteq V(G(i+1, x-1))$ by Definition 6.

Since $u_i <_\sigma u_x$ and $u_x \in N(u_i)$, Lemma 9 implies that $u_k \in N(u_i)$ for every $u_k \in V(P_2) \subseteq V(G(i+1, x-1))$. Therefore, since $P = (P_1, u_i, P_2)$ is a normal path by assumption, the first vertex of P_2 is the rightmost vertex of $V(P_2)$ in σ . Consider now any two consecutive vertices $u_r, u_{r'}$ of P_2 . Then, since $P = (P_1, u_i, P_2)$ is a normal path, it follows

that $u_{r'}$ is the rightmost unvisited vertex of $N(u_r) \cap V(P)$ (resp. of $N(u_r) \cap V(P_2)$) in σ , when P (resp. P_2) visits u_r . Therefore, since also u_y is the last vertex of P by assumption, P_2 is a normal path of $V(G(i+1, x-1))$ with u_y as its last vertex.

Suppose now that there exists a normal path P'_1 (resp. P'_2) of $G(i+1, j)$ (resp. of $G(i+1, x-1)$) with u_x (resp. with u_y) as its last vertex, such that $|P'_1| > |P_1|$ (resp. $|P'_2| > |P_2|$). Then, Lemma 11 implies that $P' = (P'_1, u_i, P_2)$ (resp. $P' = (P_1, u_i, P'_2)$) is a normal path of $G(i, j)$ with u_y as its last vertex, such that $|P'| > |P|$. This is a contradiction to the assumption that P is a longest normal path of $G(i, j)$ with u_y as its last vertex. Therefore, there exists no such path P'_1 (resp. P'_2), and thus P_1 (resp. P_2) is a longest normal path of $G(i+1, j)$ (resp. of $G(i+1, x-1)$) with u_x (resp. with u_y) as its last vertex. This completes the proof of the lemma. ■

4.1 The algorithm

In the following we present our Algorithm 4 that computes a longest path of a given cocomparability graph G . For simplicity of the presentation of this algorithm, we make the following convention.

Notation 3 Let $G = (V, E)$ be a cocomparability graph and $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$ be an LDFS umbrella-free ordering of $V \cup \{u_{n+1}\}$, where u_{n+1} is a dummy isolated vertex. For every pair of indices $i, j \in \{1, 2, \dots, n\}$ and for every vertex $u_k \in V(G(i, j))$, we denote by $P(u_k; i, j)$ a longest normal path of $G(i, j)$ with u_k as its last vertex and by $\ell(u_k; i, j)$ the length $|P(u_k; i, j)|$ of $P(u_k; i, j)$, i.e. the number of vertices of $P(u_k; i, j)$.

We first give a brief overview of Algorithm 4. It takes as input a cocomparability graph $G = (V, E)$ and an umbrella-free ordering π of V . As a preprocessing step, the algorithm applies LDFS⁺ (i.e. Algorithm 2) to the ordering π in order to compute an LDFS umbrella-free ordering σ of V . In the sequel, the dynamic programming part of Algorithm 4 builds a 3-dimensional table where for every pair of indices $i, j \in \{1, 2, \dots, n\}$ and for every vertex $u_k \in V(G(i, j))$, the entry $P(u_k; i, j)$ stores the ordered vertices of a longest normal path of $G(i, j)$ with u_k as its last vertex; the length of this path (i.e. $|P(u_k; i, j)|$) is stored in $\ell(u_k; i, j)$. Thus a longest normal path of $G = G(1, n)$ will be stored in $P(u_k; 1, n)$ for a u_k that maximizes $\ell(u_k; 1, n)$ among all $u_k \in V$ (cf. line 18). Note that from the *for*-loops in lines 3 and 4 of the algorithm and the obvious inductive hypothesis, it may be assumed during the $\{i, j\}$ th iteration of the body of the dynamic programming (cf. lines 5-17), that the values $P(u_{k'}; i', j')$ and $\ell(u_{k'}; i', j')$ have been correctly computed at previous iterations of the algorithm, for every $i' > i$.

On entry to the initialization phase for a particular $\{i, j\}$ (cf. lines 5-8), we want initial paths that do not use vertex u_i as an intermediate vertex. For a path with $u_y \in V(G(i+1, j))$ as its last vertex, such a path is stored in $P(u_y; i+1, j)$. For a path with u_i itself as its last vertex, we are only interested in the case where $u_i \in V(G(i, j))$ and, if so, we initialize $P(u_i; i, j) = (u_i)$.

Then, we enter the induction step phase of the algorithm (cf. lines 9-17) and determine how the entries of the table can be extended with the inclusion of vertex u_i (in the case where $u_i \in V(G(i, j))$). First we note that, if a normal path P of $G(i, j)$ that includes u_i has at least two vertices, then P must involve a vertex $u_x \in V(G(i+1, j))$ with $u_x u_i \in E$. For such a vertex u_x , there are two different roles that it can play in getting a possibly longer normal path to be stored in the table. First, adding the edge $u_x u_i$ to a longest normal path of $G(i+1, j)$ with u_x as its last vertex, might create a normal path with u_i as its last vertex, which is longer than the one currently stored in $P(u_i; i, j)$. This situation is covered in lines 11-13. The other role that vertex u_x might play is to serve as the “glue” between a normal path P_1 of $G(i+1, j)$

with u_x as its last vertex and a normal path P_2 of $G(i+1, x-1)$ with some vertices $u_{y'}$ and u_y as its first and last vertex, respectively ($u_{y'}$ and u_y are not necessarily distinct). Note that these two paths would be “glued” together via the two edges $u_x u_i$ and $u_i u_{y'}$. This situation is covered in lines 14-17.

The next main theorem of this section proves that Algorithm 4 computes in $O(n^4)$ time a longest path of a cocomparability graph with n vertices.

Algorithm 4 Computing a longest path of a cocomparability graph

Input: A cocomparability graph $G = (V, E)$ with $|V| = n$ and an umbrella-free ordering π of V

Output: A longest path of G

```

1: Run an LDFS+ preprocessing step to  $\pi$  to obtain the LDFS umbrella-free ordering  $\sigma$ 
2: Add an isolated dummy vertex  $u_{n+1}$  to  $\sigma$ ; denote  $\sigma = \{u_1, u_2, \dots, u_n, u_{n+1}\}$ 
3: for  $i = n$  downto 1 do
4:   for  $j = i$  to  $n$  do
5:     for every  $u_y \in V(G(i+1, j))$  do
6:        $P(u_y; i, j) \leftarrow P(u_y; i+1, j)$ ;  $\ell(u_y; i, j) \leftarrow \ell(u_y; i+1, j)$  {initialization}
7:     if  $u_i \in V(G(i, j))$  then
8:        $P(u_i; i, j) \leftarrow (u_i)$ ;  $\ell(u_i; i, j) \leftarrow 1$  {initialization}
9:     for every  $u_x \in V(G(i+1, j))$  do
10:      if  $u_i \in V(G(i, j))$  and  $u_x \in N(u_i)$  then
11:        if  $\ell(u_i; i, j) < \ell(u_x; i+1, j) + 1$  then
12:           $P(u_i; i, j) \leftarrow (P(u_x; i+1, j), u_i)$ 
13:           $\ell(u_i; i, j) \leftarrow \ell(u_x; i+1, j) + 1$ 
14:        for every  $u_y \in V(G(i+1, x-1))$  do
15:          if  $\ell(u_y; i, j) < \ell(u_x; i+1, j) + \ell(u_y; i+1, x-1) + 1$  then
16:             $P(u_y; i, j) \leftarrow (P(u_x; i+1, j), u_i, P(u_y; i+1, x-1))$ 
17:             $\ell(u_y; i, j) \leftarrow \ell(u_x; i+1, j) + \ell(u_y; i+1, x-1) + 1$ 
18: return a path  $P(u_k; 1, n)$  with  $\ell(u_k; 1, n) = \max\{\ell(u_y; 1, n) \mid u_y \in V\}$ 

```

Theorem 3 For a given cocomparability graph $G = (V, E)$ with n vertices, Algorithm 4 computes a longest path P of G in $O(n^4)$ time.

Proof. In the first line, Algorithm 4 applies an LDFS⁺ preprocessing step to the given umbrella-free ordering π of V . The resulting LDFS ordering σ is again umbrella-free [6]. In the second line, the algorithm adds a dummy isolated vertex u_{n+1} to σ to the right of all other vertices of V , i.e. we consider without loss of generality that $\sigma = (u_1, u_2, \dots, u_n, u_{n+1})$. Note that σ remains an LDFS umbrella-free ordering, also after the addition of u_{n+1} to it. Furthermore, note that any longest path of G is also maximal (cf. Definition 3). Therefore, in order to compute a *longest* path of G , it suffices by Theorem 2 to compute a *longest normal* path of G (with respect to the ordering σ), i.e. a longest path among the normal ones.

In lines 3-17, Algorithm 4 iterates for every pair of indices $i, j \in \{1, 2, \dots, n\}$ and computes a path $P(u_k; i, j)$ and a value $\ell(u_k; i, j)$ for every vertex $u_k \in V(G(i, j))$. We will prove by induction on i that $P(u_k; i, j)$ is indeed a longest normal path of $G(i, j)$ with u_k as its last vertex and that $\ell(u_k; i, j) = |P(u_k; i, j)|$.

For the induction basis, let $i = n$; in this case also $j = n$ (cf. line 4). Furthermore $u_i \notin N(u_{i+1})$ for $i = n$, since u_{n+1} is an isolated vertex, and thus the algorithm exe-

cutes line 8. In this line, the algorithm computes the path $P(u_n; n, n) = (u_n)$, which is clearly the only (and thus also the longest) normal path of $G(n, n)$ with u_n as its last vertex. Then, since $G(n + 1, n) = \emptyset$ (cf. Definition 6), lines 6 and 10-17 are not executed at all. This proves the induction basis.

For the induction step, let $i \leq n - 1$. Consider the iteration of the algorithm for any $j \in \{i, i + 1, \dots, n\}$. First, the algorithm initializes in lines 5-8 the values $P(u_k; i, j)$ and $\ell(u_k; i, j)$ for every $u_k \in V(G(i, j))$. Then, it updates these values if necessary in lines 9-17. For every vertex $u_y \in V(G(i + 1, j))$, the induction hypothesis implies that $P(u_y; i + 1, j)$ is a longest normal path of $G(i + 1, j)$ with u_y as its last vertex and that $\ell(u_y; i + 1, j) = |P(u_y; i + 1, j)|$. Recall by Observation 4 that $G(i + 1, j) = G(i, j) \setminus \{u_i\}$. Therefore, for every $u_y \in V(G(i + 1, j))$, the value $\ell(u_y; i + 1, j)$ is the greatest length of a normal path P of $G(i, j)$ with u_y as its last vertex, such that P does not include u_i . The algorithm initializes in line 6 for every $u_y \in V(G(i + 1, j))$ the values $P(u_y; i, j)$ and $\ell(u_y; i, j)$ as $P(u_y; i + 1, j)$ and $\ell(u_y; i + 1, j)$, respectively. Furthermore, in the case where $u_i \in V(G(i, j))$, the algorithm initializes in line 8 the values $P(u_i; i, j) = (u_i)$ and $\ell(u_i; i, j) = 1$. Otherwise, in the case where $u_i \notin V(G(i, j))$, the algorithm does not execute line 8, since the values $P(u_i; i, j)$ and $\ell(u_i; i, j)$ can not be defined (cf. Notation 3).

Suppose that $u_i \in V(G(i, j))$; then the path $P(u_i; i, j)$ is well defined (cf. Notation 3). Recall by Observation 6 that for any normal path P_1 of $G(i + 1, j)$ with a vertex u_x as its last vertex, such that $u_x \in N(u_i)$, the path (P_1, u_i) is a normal path of $G(i, j)$. Conversely, recall by Observation 5 that the path $P(u_i; i, j) \setminus \{u_i\}$ (if not empty) is a normal path of $G(i + 1, j)$. Therefore, in order to update the value of $P(u_i; i, j)$, the algorithm correctly computes in lines 11-13 the paths $(P(u_x; i + 1, j), u_i)$ for every $u_x \in V(G(i + 1, j))$, such that $u_x \in N(u_i)$, and keeps the longest of them.

Recall now that for every $u_y \in V(G(i + 1, j))$, the value $\ell(u_y; i + 1, j)$ is the greatest length of a normal path P of $G(i, j)$ with u_y as its last vertex, such that P does not include u_i . Furthermore, recall that for every $u_y \in V(G(i + 1, j))$ the values $P(u_y; i, j)$ and $\ell(u_y; i, j)$ have been initialized in line 6 as $P(u_y; i + 1, j)$ and $\ell(u_y; i + 1, j)$, respectively. In the case where $u_i \in V(G(i, j))$ (cf. line 10), the algorithm executes lines 15-17 for every $u_x \in V(G(i + 1, j))$ with $u_x \in N(u_i)$ and for every $u_y \in V(G(i + 1, x - 1))$. For such a pair of vertices u_x, u_y , recall by Lemma 11 that $(P(u_x; i + 1, j), u_i, P(u_y; i + 1, x - 1))$ is a normal path of $G(i, j)$ with u_y as its last vertex. Conversely, let P be a normal path of $G(i, j)$ with $u_y \neq u_i$ as its last vertex, let $P = (P_1, u_i, P_2)$, and let u_x be the last vertex of P_1 . Then Lemma 12 implies that $P_1 = P(u_x; i + 1, j)$ and $P_2 = P(u_y; i + 1, x - 1)$. Therefore, the algorithm correctly computes during the multiple executions of lines 15-17 the greatest length ℓ of a normal path P of $G(i, j)$ with u_y as its last vertex, such that P includes u_i . If at least one of these paths has greater length than the initial value $\ell(u_y; i, j)$ that has been computed in line 6, the algorithm keeps in $P(u_y; i, j)$ the longest among these paths. This completes the induction step.

Therefore, for every pair of indices $i, j \in \{1, 2, \dots, n\}$ (such that $G(i, j) \neq \emptyset$) and every $u_k \in V(G(i, j))$, the algorithm correctly computes after the execution of lines 1-17 a longest normal path $P(u_k; i, j)$ of $G(i, j)$ with u_k as its last vertex and its length $\ell(u_k; i, j) = |P(u_k; i, j)|$. Finally, the algorithm computes and returns in line 18 the longest among the paths $P(u_y; 1, n)$, where $u_y \in V(G(1, n))$. Since $G(1, n) = G$, the returned path is a longest normal path of G , and thus also a longest path of G by Theorem 2.

Before establishing the running time of the algorithm, we discuss some implementation details. First of all, to avoid the search of the table indicated in line 18, the length and location of the current longest path would be maintained throughout the algorithm. Secondly, we have to state exactly what is stored in each entry of the table. Following standard dynamic programming techniques, we do not store the path itself, but rather, an indication of how the path is built.

In particular, each of lines 6, 8, 12, and 16 gives “instructions” on how to build the current longest path using information that has already been computed. At the end of the algorithm a simple recursive unwinding of these “instructions” yields a longest path in the given graph.

Regarding the running time of Algorithm 4, we first examine the dynamic programming part of the algorithm. Lines 15-17 lie in four loops of $O(n)$ iterations each. Following the implementation details described above, each step in lines 15-17 can be executed in constant time, yielding an $O(n^4)$ bound on the dynamic programming portion of the algorithm. Since the other parts of the algorithm, even if we have to confirm that we have an umbrella-free ordering of V , can easily be implemented to run in $O(n^3)$ time, the total running time of Algorithm 4 is $O(n^4)$. This completes the proof of the theorem. ■

Remark 1 Recall by Observation 1 that an I -ordering σ of any interval graph G is also an umbrella-free ordering. Furthermore, it is easy to see that σ is also an LDFS ordering. Thus, since lines 2-17 of Algorithm 4 are applied to such an ordering σ , and since interval graphs are strictly included in cocomparability graphs [3], Theorem 3 implies that Algorithm 4 (which is essentially simpler than the algorithm presented in [16]) also computes with the same time complexity a longest path of an interval graph.

5 Conclusion and further research

In this paper we provided the first polynomial algorithm for the longest path problem on cocomparability graphs. This algorithm is based on a dynamic programming approach that is applied to a Lexicographic Depth First Search (LDFS) characterizing ordering of the vertices of cocomparability graphs. Our results provide hope that this general dynamic programming approach can be used in a more general setting, leading to efficient algorithms for the longest path problem on even greater classes of graphs. Furthermore, more interestingly, in addition to the recent results presented in [6], our results also provide evidence that cocomparability graphs present an interval graph structure when they are considered using an LDFS characterization ordering of their vertices, which may lead to other new and more efficient combinatorial algorithms. Many interesting open questions are raised by the results in this paper:

- There are now two path problems where the interval graph algorithm can be modified by the addition of an LDFS⁺ preprocessing sweep to solve the same problem on cocomparability graphs. Are there other such problems? (Note that the Hamiltonian cycle algorithm for interval graphs does not seem to extend to cocomparability graphs in this way.)
- More importantly, is there an underlying “interval structure” in cocomparability graphs exposed by an LDFS⁺ sweep of an umbrella-free ordering?
- There are many applications of multi-sweeping of LBFS (see [9] for a recent result; for a survey see [5]). Is anything gained by multi-sweeping LDFS?
- Are there other applications of LDFS?
- Can the new Hamiltonian path, minimum path cover, and longest path algorithms for cocomparability graphs be extended to asteroidal triple-free (AT-free) graphs, or failing that to graph classes that lie between cocomparability graphs and AT-free graphs [7]? The complexity of all Hamiltonicity problems is still open for AT-free graphs.
- Can LDFS be implemented to run in linear time?

References

- [1] S. R. Arikati and C. P. Rangan. Linear algorithm for optimal path cover problem on interval graphs. *Information Processing Letters*, 35(3):149–153, 1990.
- [2] A. A. Bertossi. Finding Hamiltonian circuits in proper interval graphs. *Information Processing Letters*, 17(2):97–101, 1983.
- [3] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: a survey*. Society for Industrial and Applied Mathematics (SIAM), 1999.
- [4] R. W. Bulterman, F. W. van der Sommen, G. Zwaan, T. Verhoeff, A. J. M. van Gasteren, and W. H. J. Feijen. On computing a longest path in a tree. *Information Processing Letters*, 81(2):93–96, 2002.
- [5] D. G. Corneil. Lexicographic breadth first search a survey. In *Proceedings of the 30rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 1–19, 2004.
- [6] D. G. Corneil, B. Dalton, and M. Habib. The application of Lexicographic Depth First Search to minimum path cover problems in cocomparability graphs. In preparation.
- [7] D. G. Corneil, E. Köhler, S. Olariu, and L. Stewart. Linear orderings of subfamilies of AT-free graphs. *SIAM Journal on Discrete Mathematics*, 20(1):105–118, 2006.
- [8] D. G. Corneil and R. M. Krueger. A unified view of graph searching. *SIAM Journal on Discrete Mathematics*, 22(4):1259–1276, 2008.
- [9] D. G. Corneil, S. Olariu, and L. Stewart. The LBFS structure and recognition of interval graphs. *SIAM Journal of Discrete Mathematics*, 23(4):1905–1953, 2009.
- [10] P. Damaschke. The Hamiltonian circuit problem for circle graphs is NP-complete. *Information Processing Letters*, 32(1):1–2, 1989.
- [11] P. Damaschke. Paths in interval graphs and circular arc graphs. *Discrete Mathematics*, 112(1-3):49–64, 1993.
- [12] P. Damaschke, J. S. Deogun, D. Kratsch, and G. Steiner. Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm. *Order*, 8:383–391, 1992.
- [13] J. S. Deogun and G. Steiner. Polynomial algorithms for Hamiltonian cycle in cocomparability graphs. *SIAM Journal on Computing*, 23(3):520–552, 1994.
- [14] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5:704–714, 1976.
- [15] M. C. Golumbic. *Algorithmic graph theory and perfect graphs (Annals of Discrete Mathematics, Vol. 57)*. North-Holland Publishing Co., 2 edition, 2004.
- [16] K. Ioannidou, G. B. Mertzios, and S. D. Nikolopoulos. The longest path problem has a polynomial solution on interval graphs. *Algorithmica*. To appear. A technical report can be found in <http://sunsite.informatik.rwth-aachen.de/Publications/AIB/2009/2009-11.pdf>.
- [17] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. Hamiltonian paths in grid graphs. *SIAM Journal on Computing*, 11:676–686, 1982.
- [18] D. R. Karger, R. Motwani, and G. D. S. Ramkumar. On approximating the longest path in a graph. *Algorithmica*, 18:82–98, 1997.
- [19] J. M. Keil. Finding Hamiltonian circuits in interval graphs. *Information Processing Letters*, 20:201–206, 1985.
- [20] D. Kratsch and L. Stewart. Domination on cocomparability graphs. *SIAM Journal on Discrete Mathematics*, 6(3):400–417, 1993.
- [21] R. Krueger. *Graph Searching*. PhD thesis, Department of Computer Science, University of Toronto, 2005.
- [22] H. Müller. Hamiltonian circuits in chordal bipartite graphs. *Discrete Mathematics*, 156(1-3):291–298, 1996.
- [23] G. Narasimhan. A note on the Hamiltonian circuit problem on directed path graphs. *Information Processing Letters*, 32(4):167–170, 1989.

- [24] S. Olariu. An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37(1):21–25, 1991.
- [25] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.
- [26] J. Spinrad. Efficient implementation of lexicographic depth first search. Submitted for publication.
- [27] Y. Takahara, S. Teramoto, and R. Uehara. Longest path problems on ptolemaic graphs. *IEICE - IEICE - Transactions on Information and Systems*, E91-D(2):170–177, 2008.
- [28] R. Uehara and Y. Uno. Efficient algorithms for the longest path problem. In *15th Annual International Symposium on Algorithms and Computation (ISAAC)*, pages 871–883, 2004.
- [29] R. Uehara and G. Valiente. Linear structure of bipartite permutation graphs and the longest path problem. *Information Processing Letters*, 103(2):71–77, 2007.