The Longest Path Problem is Polynomial on Cocomparability Graphs

Kyriaki Ioannidou and Stavros D. Nikolopoulos

Department of Computer Science, University of Ioannina
P.O.Box 1186, GR-45110 Ioannina, Greece
{kioannid, stavros}@cs.uoi.gr

Abstract

The longest path problem is the problem of finding a path of maximum length in a graph. As a generalization of the Hamiltonian path problem, it is NP-complete on general graphs and, in fact, on every class of graphs that the Hamiltonian path problem is NP-complete. Polynomial solutions for the longest path problem have recently been proposed for weighted trees, ptolemaic graphs, bipartite permutation graphs, interval graphs, and some small classes of graphs. Although the Hamiltonian path problem on cocomparability graphs was proved to be polynomial almost two decades ago, the complexity status of the longest path problem on cocomparability graphs has remained open until now; actually, the complexity status of the problem has remained open even on the smaller class of permutation graphs. In this paper, we present a polynomial-time algorithm for solving the longest path problem on the class of cocomparability graphs. Our result resolves the open question for the complexity of the problem on such graphs, and since cocomparability graphs form a superclass of both interval and permutation graphs, extends the polynomial solution of the longest path problem on interval graphs and provides polynomial solution to the class of permutation graphs.

Keywords: Longest path problem, cocomparability graphs, permutation graphs, polynomial algorithm, complexity.

1 Introduction

The problem of finding a path of maximum length in a graph (Longest Path Problem) generalizes the Hamiltonian path problem and thus it is NP-complete on general graphs; in fact, it is NP-complete on every class of graphs that the Hamiltonian path problem is NP-complete. It is thus interesting to study the longest path problem on classes of graphs $\mathcal{C}$ where the Hamiltonian path problem is polynomial, since if a graph $G \in \mathcal{C}$ is not Hamiltonian, it makes sense in several applications to search for a longest path of G. Although the Hamiltonian path problem has been extensively studied in the past two decades, only recently did the longest path problem start receiving attention.

Additionally, recently the longest path problem has also received attention in the direction of approximation results, some of which imply that finding 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 has a Hamiltonian path, the problem of finding a path of length $n - n^{\varepsilon}$ for any $\varepsilon < 1$ is NP-hard, where n is the number of vertices of the graph [21]. Moreover, there is no polynomial-time constant-factor approximation algorithm for the longest path problem unless P=NP [21]. For related results see also [12–14, 29, 30].

The Hamiltonian path problem is known to be NP-complete in general graphs [15,16], and remains NP-complete even when restricted to some small classes of graphs such as split graphs [17], chordal bipartite graphs, split strongly chordal graphs [23], directed path graphs [24], circle graphs [7], planar graphs [16], and grid graphs [20, 26]. On the other hand, it admits polynomial time solutions on some known classes of graphs; such classes include interval graphs [1, 8], circular-arc graphs [8], biconvex graphs [2], and cocomparability graphs [9]. Note that the problem of finding a longest path on proper interval graphs is easy, since all connected proper interval graphs have a Hamiltonian path which can be computed in linear time [3].

Polynomial time solutions for the longest path problem are known only for small classes of graphs. Specifically, a linear-time algorithm for finding a longest path in a tree was proposed by Dijkstra early in 1960, a formal proof of which can be found in [5]. Recently, through a generalization of Dijkstra's algorithm for trees, Uehara and Uno [27] solved the longest path problem for weighted trees and block graphs in linear time and space, and for cacti in $O(n^2)$ time and space, where n is the number of vertices of the input graph. Polynomial algorithms for the longest path problem have been also proposed on bipartite permutation and ptolemaic graphs having $O(n)$ and $O(n^5)$ time complexity, respectively [25, 28]. Furthermore, Uehara and Uno in [27] solved the longest path problem on a subclass of interval graphs in $O(n^3(m + n \log n))$ time, and as a corollary they showed that a longest path on threshold graphs can be found in $O(n + m)$ time and space. Recently, Ioannidou *et al.* [19] showed that the longest path problem has a polynomial solution on interval graphs by proposing an algorithm that runs in $O(n^4)$ time, answering thus the question left open in [27] concerning the complexity of the problem on interval graphs.

Although the Hamiltonian path problem on cocomparability graphs was proved to be polynomial almost two decades ago [9], the complexity status of the longest path problem on cocomparability graphs has remained open until now; actually, the complexity status of the problem has remained open even on the smaller class of permutation graphs. Note that, the hamiltonian cycle problem as well has been proved to be polynomial on permutation graphs [10] and cocomparability graphs [11].

In this paper we present a polynomial-time algorithm for solving the longest path problem on the class of cocomparability graphs, an important and well-known class of perfect graphs [17]. Thus, our result resolves the open question for the complexity of the problem on cocomparability graphs, and since cocomparability graphs form a superclass of both interval and permutation graphs, extends the polynomial solution of the longest path problem on interval graphs [19], and also provides polynomial solution to the class of permutation graphs.

The rest of this paper is organized as follows. In Section 2, we first review some properties of partial orders, comparability and cocomparability graphs and, then, introduce the notion of a normal antipath on a comparability graph, which is needed for our algorithm. In Section 3, we present our algorithm for solving the longest path problem on a cocomparability graph, and in Section 4 we prove the correctness and compute the time complexity of our algorithm. Finally, some concluding remarks follow in Section 5.

2 Theoretical Framework

We consider finite undirected graphs with no loops or multiple edges. For a graph G, we denote its vertex and edge set by $V(G)$ and $E(G)$, respectively. An undirected edge is a pair of distinct vertices $u, v \in V(G)$, and is denoted by uv. We say that the vertex u is adjacent to the vertex v or, equivalently, the vertex u sees the vertex v, if there is an edge uv in G. If $uv \notin E(G)$ then we say that vertices u and v are *antineighbors* in G. Let S be a set of vertices of a graph G; then, the cardinality of the set S is denoted by $|S|$ and the subgraph of G induced by S is denoted by $G[S]$. Sometimes we denote by $G \setminus S$ the graph $G[V(G) \setminus S]$. The set $N(v) = \{u \in V(G) : uv \in E(G)\}$

is called the *neighborhood* of the vertex $v \in V(G)$ in G, sometimes denoted by $N_G(v)$ for clarity reasons. The set $N[v] = N(v) \cup \{v\}$ is called the *closed neighborhood* of the vertex $v \in V(G)$. By $N_{\overline{G}}(v)$ we denote the set of the antineighbors of the vertex v in the graph G.

For basic definitions in graph theory refer to [4,17,22]. A *simple path* (resp. *antipath*) of a graph G is a sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $v_i v_{i+1} \in E(G)$ (resp. $v_i v_{i+1} \notin E(G)$), for each i, $1 \leq i \leq k-1$, and is denoted by $(v_1, v_2, \ldots, v_k)$; throughout the paper all paths and antipaths are considered to be simple. We denote by $V(P)$ the set of vertices in the path (antipath) P, and define the *length* of the path (antipath) P to be the number of vertices in P, i.e., $|P| = |V(P)|$. We call *right endpoint* of a path (antipath) $P = (v_1, v_2, \ldots, v_k)$ the last vertex v_k of P. Moreover, if $P = (v_1, v_2, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, v_j, v_{j+1}, v_{j+2}, \ldots, v_k)$ is a path (antipath) of a graph and $P_0 = (v_i, v_{i+1}, \ldots, v_j)$ is a subpath (subantipath) of P, we shall denote the path (antipath) P by $P = (v_1, v_2, \ldots, v_{i-1}, P_0, v_{j+1}, v_{j+2}, \ldots, v_k)$.

2.1 Partial Orders and Cocomparability Graphs

A *partial order* will be denoted by $\mathcal{P} = (V, <_{\mathcal{P}})$, where V is the finite ground set of elements or vertices and $<_{\mathcal{P}}$ is an irreflexive, antisymmetric, and transitive binary relation on V. Two elements $a, b \in V$ are comparable in $\mathcal{P}$ (denoted by $a \sim_{\mathcal{P}} b$) if $a <_{\mathcal{P}} b$ or $b <_{\mathcal{P}} a$; otherwise, they are said to be incomparable (denoted by $a \parallel b$). An *extension* of a partial order $\mathcal{P} = (V, <_{\mathcal{P}})$ is a partial order $L = (V, <_L)$ on the same ground set that *extends* $\mathcal{P}$, i.e., $a <_{\mathcal{P}} b \Rightarrow a <_L b$, for all $a, b \in V$. The *dual partial order* $\mathcal{P}^d$ of $\mathcal{P} = (V, <_{\mathcal{P}})$ is a partial order $\mathcal{P}^d = (V, <_{\mathcal{P}^d})$ such that for any two elements $a, b \in V$, $a <_{\mathcal{P}^d} b$ if and only if $b <_{\mathcal{P}} a$.

The graph G, edges of which are exactly the comparable pairs of a partial order $\mathcal{P}$ on $V(G)$, is called the *comparability graph* of $\mathcal{P}$, and is denoted by $G(\mathcal{P})$. The complement graph $\overline{G}$, whose edges are the incomparable pairs of $\mathcal{P}$, is called the *cocomparability graph* of $\mathcal{P}$, and is denoted by $\overline{G}(\mathcal{P})$. Alternatively, a graph G is a cocomparability graph if its complement graph $\overline{G}$ has a transitive orientation, corresponding to the comparability relations of a partial order $\mathcal{P}_{\overline{G}}$. Note that a partial order $\mathcal{P}$ uniquely determines its comparability graph $G(\mathcal{P})$ and its cocomparability graph $\overline{G}(\mathcal{P})$, but the reverse is not true, i.e., a cocomparability graph G has as many partial orders $\mathcal{P}_{\overline{G}}$ as the number of the transitive orientations of $\overline{G}$. Also, the class of cocomparability graphs is hereditary, that is if G is a cocomparability graph, then every induced subgraph of G is a cocomparability graph.

Let G be a comparability graph, and let $\mathcal{P}_G$ be a partial order which corresponds to G. The graph G can be represented by a directed covering graph with layers $H_1, H_2, \ldots, H_h$, in which each vertex is on the highest possible layer. That is, the maximal vertices of the partial order $\mathcal{P}_G$ are on the highest layer H_h, and for every vertex v on layer H_{i-1} there exists a vertex u on layer H_i such that $v <_{\mathcal{P}_G} u$; such a layered representation of G (respectively $\mathcal{P}_G$) is a called the *Hasse diagram* of G (respectively $\mathcal{P}_G$) [9].

Let $\sigma = (V(G), <_\sigma)$ be a partial order on the vertices of a comparability graph G, such that for any two vertices $v, u \in V(G)$, $v <_\sigma u$ if and only if $v \in H_i$, $u \in H_j$, and $i < j$; hereafter, we equivalently denote $v <_\sigma u$ by $u >_\sigma v$. For simplicity sometimes we shall write $v =_\sigma u$, for vertices $v, u \in V(G)$ which belong to the same layer H_i; we write $v \neq_\sigma u$ to denote that vertices $v, u \in V(G)$ belong to different layers. Also, $v \leq_\sigma u$ implies that either $v <_\sigma u$ or $v =_\sigma u$; again, we equivalently denote $v \leq_\sigma u$ by $u \geq_\sigma v$. Throughout the paper, such an ordering σ is called a *layered ordering* of G. Note that, the partial order σ is an extension of the partial order $\mathcal{P}_G$; in particular, it holds $v <_{\mathcal{P}_G} u$ if and only if $v <_\sigma u$ and $vu \in E(G)$, for any two vertices $u, v \in V(G)$.

Since a comparability graph G does not uniquely determine a partial order, hereafter we will represent a comparability graph G by its Hasse diagram and we will denote the partial order $(V(G), <_{\mathcal{P}_G})$ to which the Hasse diagram of G corresponds by $\mathcal{P}_G$. Thus, we will say that $\mathcal{P}_G$ is the partial order which *corresponds* to the comparability graph G. Note that vertices in the Hasse diagram satisfy the following property: for any three vertices $v, u, w \in V(G)$ such that $v \in H_i$, $u \in H_j$, $w \in H_k$, and $i < j < k$ (or, equivalently, $v <_\sigma u <_\sigma w$), if $vu \in E(G)$ and $uw \in E(G)$, then $vw \in E(G)$.

The following definition and results were given by Damaschke *et al.* in [9], based on which they prove the correctness of their algorithm for finding a Hamiltonian path of a cocomparability graph; note that their algorithm uses the bump number algorithm which is presented in [18].

Definition 2.1 *(Damaschke et al. [9]): Let G be a comparability graph, and let $\mathcal{P}_G$ be the partial order which corresponds to G. A path $P = (v_1, v_2, \ldots, v_k)$ of the cocomparability graph $\overline{G}$ is monotone if $v_i <_{\mathcal{P}_G} v_j$ implies $i < j$.*

Lemma 2.1 *(Damaschke et al. [9]): Let G be a comparability graph, and let $\mathcal{P}_G$ be the partial order which corresponds to G. Let $P = (v_1, v_2, \ldots, v_k)$ be a Hamiltonian path of the cocomparability graph $\overline{G}$ such that v_1 is a minimal element of $\mathcal{P}_G$. Then there exists a monotone Hamiltonian path P' of $\overline{G}$ starting with v_1.*

Theorem 2.1 *(Damaschke et al. [9]): Let G be a cocomparability graph. Then, G has a Hamiltonian path if and only if G has a monotone Hamiltonian path.*

It appears that the above two results hold not only for Hamiltonian paths of a cocomparability graph $\overline{G}$, but also for any path of $\overline{G}$. Indeed, let P be a path of $\overline{G}$ and let $\overline{G'} = \overline{G}[V(P)]$ be the subgraph of $\overline{G}$ induced by the vertices of P (recall that cocomparability graphs have the hereditary property). Also, let $\mathcal{P}_{G'}$ be the partial order which corresponds to G' such that $\mathcal{P}_G$ is an extension of $\mathcal{P}_{G'}$, i.e., for any two vertices $u, v \in V(\overline{G})$, if $u <_{\mathcal{P}_G} v$ and $u, v \in V(\overline{G'})$, then $u <_{\mathcal{P}_{G'}} v$. Then, since P is a Hamiltonian path of $\overline{G'}$, from Theorem 2.1 there exists a monotone path P' of $\overline{G'}$ (with respect to $\mathcal{P}_{G'}$) such that $V(P') = V(P)$. From Definition 2.1 it is easy to see that P' is also a monotone path of $\overline{G}$ (with respect to $\mathcal{P}_G$), since $\mathcal{P}_G$ is an extension of $\mathcal{P}_{G'}$.

Additionally, since a path P of a cocomparability graph $\overline{G}$ is an antipath of the comparability graph G, and since our algorithm for computing a longest path of a cocomparability graph $\overline{G}$ computes in fact a longest antipath of the comparability graph G, we restate the above definition and results and whenever P denotes a path of a cocomparability graph $\overline{G}$, we refer to P as an antipath of the comparability graph G.

We first restate Definition 2.1 as follows: an antipath $P = (v_1, v_2, \ldots, v_k)$ of a comparability graph G is monotone if $v_i <_{\mathcal{P}_G} v_j$ implies $i < j$, where $\mathcal{P}_G$ is the partial order which corresponds to G. We next restate Lemma 2.1 and Theorem 2.1 in a form stronger than the one stated in [9].

Lemma 2.2 *Let G be a comparability graph, and let $\mathcal{P}_G$ be the partial order which corresponds to G. Let $P = (v_1, v_2, \ldots, v_k)$ be an antipath of G such that v_1 is a minimal element of $V(P)$ in $\mathcal{P}_G$. Then there exists a monotone antipath P' of G starting with vertex v_1 such that $V(P') = V(P)$.*

Theorem 2.2 *Let G be a comparability graph. If P is an antipath of G, then there exists a monotone antipath P' of G such that $V(P') = V(P)$.*

The following lemma holds.

Lemma 2.3 *Let G be a comparability graph, and let σ be the layered ordering of G. Let $P = (v_1, v_2, \ldots, v_k)$ be an antipath of G, and let $v_\ell \notin V(P)$ be a vertex of G such that $v_1 \leq_\sigma v_\ell <_\sigma v_k$ and $v_\ell v_k \in E(G)$. Then there exist two consecutive vertices v_{i-1} and v_i in P, $2 \leq i \leq k$, such that $v_{i-1} v_\ell \notin E(G)$ and $v_\ell <_\sigma v_i$.*

Proof. Let $P = (v_1, v_2, \ldots, v_k)$ be an antipath of G, and let $v_\ell \notin V(P)$ be a vertex of G such that $v_1 \leq_\sigma v_\ell <_\sigma v_k$ and $v_\ell v_k \in E(G)$. We first show that at least one vertex of P does not see v_ℓ. In the case where $v_1 =_\sigma v_\ell$, then v_1 is such a vertex, i.e., $v_1 v_\ell \notin E(G)$. Consider now that case where $v_1 <_\sigma v_\ell <_\sigma v_k$, and assume that $v_\ell v_i \in E(G)$ for every vertex $v_i \in V(P)$, $1 \leq i \leq k$. Then for every vertex $v_i \in V(P)$, $1 \leq i \leq k$, it follows that $v_\ell \neq_\sigma v_i$, since vertices belonging to the same layer of the Hasse diagram of G form an independent set. If $v_2 <_\sigma v_1$, then obviously $v_2 <_\sigma v_\ell$. Assume now that $v_1 <_\sigma v_2$; recall that $v_1 <_\sigma v_\ell$. If $v_1 <_\sigma v_\ell <_\sigma v_2$, from the transitivity property it follows that $v_2 v_1 \in E(G)$, since $v_2 v_\ell \in E(G)$ and $v_\ell v_1 \in E(G)$; this is a contradiction to our assumption that v_1 and v_2 are consecutive in the antipath P. Thus, $v_2 <_\sigma v_\ell$. Similarly, we can easily show by induction that for every pair v_{x-1}, v_x of consecutive vertices in P, $2 \leq x \leq k-1$, if $v_{x-1} <_\sigma v_\ell$ then $v_x <_\sigma v_\ell$, otherwise $v_{x-1} v_x \in E(G)$ due to the transitivity property. In particular, the same holds for the pair v_{k-2} and v_{k-1}, i.e., from $v_{k-2} <_\sigma v_\ell$, we obtain $v_{k-1} <_\sigma v_\ell$. Recall that $v_\ell <_\sigma v_k$; thus, $v_{k-1} <_\sigma v_\ell <_\sigma v_k$, and since $v_k v_\ell \in E(G)$ and $v_\ell v_{k-1} \in E(G)$, from the transitivity property we obtain that $v_k v_{k-1} \in E(G)$. This comes to a contradiction to our assumption that P is an antipath of G. Thus, there exists at least one vertex of P which does not see v_ℓ.

Let v_{i-1} be the last vertex from left to right in P (i.e., $i-1$ is the greatest index) such that $v_{i-1} v_\ell \notin E(G)$, $2 \leq i \leq k$. Therefore, for every index j, $i \leq j \leq k$, we have $v_j v_\ell \in E(G)$ and, thus, $v_j \neq_\sigma v_\ell$. If $i = k$, then v_{k-1}, v_k is a pair of consecutive vertices in P such that $v_{k-1} v_\ell \notin E(G)$ and $v_\ell <_\sigma v_k$, and the lemma holds. Assume that $2 \leq i \leq k-1$. We will show that $v_\ell <_\sigma v_j$ for every j, $i \leq j \leq k$. For $j = k$, $v_\ell <_\sigma v_k$ holds by assumption. Consider now the case where $i \leq j \leq k-1$. Assume that there exists a vertex v_p, $i \leq p \leq k-1$, such that $v_p <_\sigma v_\ell$; let v_p be the last such vertex from left to right in P. Thus, $v_\ell <_\sigma v_{p+1}$, by the choice of v_p. Then, $v_p <_\sigma v_\ell <_\sigma v_{p+1}$, and since $v_{p+1} v_\ell \in E(G)$ and $v_\ell v_p \in E(G)$, we obtain that $v_{p+1} v_p \in E(G)$. This is a contradiction to our assumption that v_p and v_{p+1} are consecutive in the antipath P of G. Therefore, there exists no vertex v_p, $i \leq p \leq k-1$, such that $v_p <_\sigma v_\ell$. Thus, we have shown that $v_\ell <_\sigma v_j$ for every j, $i \leq j \leq k$. In particular, $v_\ell <_\sigma v_i$. Therefore, the vertices v_{i-1} and v_i are a pair of consecutive vertices in P such that $v_{i-1} v_\ell \notin E(G)$ and $v_\ell <_\sigma v_i$. ∎

2.2 Normal Antipaths on Comparability Graphs

Our algorithm computes a longest path P of a cocomparability graph G by computing a specific type of antipaths of the comparability graph $\overline{G}$, which we call *normal* antipaths.

Definition 2.2 *Let G be a comparability graph, and let σ be a layered ordering of G. The antipath $P = (v_1, v_2, \ldots, v_k)$ of G is called normal, if v_1 is a leftmost (i.e., minimal) vertex of $V(P)$ in σ, and for every i, $2 \leq i \leq k$, the vertex v_i is a leftmost vertex of $N_{\overline{G}}(v_{i-1}) \cap \{v_i, v_{i+1}, \ldots, v_k\}$ in σ.*

Note that in Definition 2.2, vertex v_1 is a leftmost (minimal) vertex of $V(P)$ in σ, and not necessarily a leftmost (minimal) vertex of $V(G)$ in σ. Based on Lemma 2.3 and Definition 2.2, we prove the following result.

Lemma 2.4 *Let G be a comparability graph, and let σ be the layered ordering of G. Let $P = (v_1, v_2, \ldots, v_k)$ be a normal antipath of G, and let v_ℓ, and v_j be two vertices of P such that $v_\ell <_\sigma v_j$ and $v_\ell v_j \in E(G)$. Then $\ell < j$, i.e., v_ℓ appears before v_j in P.*

Proof. Let $P = (v_1, v_2, \ldots, v_k)$ be a normal antipath of a comparability graph G, and let v_ℓ and v_j be two vertices of P such that $v_\ell <_\sigma v_j$ and $v_\ell v_j \in E(G)$. Assume that $j < \ell$, i.e., $P = (v_1, \ldots, v_j, \ldots, v_\ell, \ldots, v_k)$. Since P is a normal antipath, then v_1 is a leftmost vertex of $V(P)$ in σ; thus, $v_1 \leq_\sigma v_\ell <_\sigma v_j$. Since $P' = (v_1, v_2, \ldots, v_j)$ is an antipath, $v_\ell \notin V(P')$, $v_1 \leq_\sigma v_\ell <_\sigma v_j$, and $v_\ell v_j \in E(G)$, then from Lemma 2.3, we obtain that there exist two consecutive vertices v_{i-1} and v_i in P', $2 \leq i \leq j$, such that $v_{i-1} v_\ell \notin E(G)$ and $v_\ell <_\sigma v_i$. However, this comes to a contradiction to our assumption that P is a normal antipath, since from Definition 2.2 we obtain that v_ℓ should be the next vertex of v_{i-1} in P, instead of v_i. Therefore, we obtain $\ell < j$. ∎

Recall that, if $\mathcal{P}_G$ is the partial order corresponding to a comparability graph G, and σ is the layered ordering of G, then $v_\ell <_{\mathcal{P}_G} v_j$ if and only if $v_\ell <_\sigma v_j$ and $v_\ell v_j \in E(G)$, for any two vertices $v_\ell, v_j \in V(G)$. Therefore, the definition of a monotone antipath can be paraphrased as follows: an antipath $P = (v_1, v_2, \ldots, v_k)$ of a comparability graph G is monotone if $v_\ell <_\sigma v_j$ and $v_\ell v_j \in E(G)$ implies that v_ℓ appears before v_j in P. Then, from Lemma 2.4 we obtain the following result.

Corollary 2.1 *Let G be a comparability graph. If P is a normal antipath of G, then P is a monotone antipath of G.*

Note that the inverse of Corollary 2.1 is not always true; for example, see the antipath P in Figure 1. In [9], for proving that for any Hamiltonian path P of a cocomparability graph $\overline{G}$ there exists a monotone Hamiltonian path of $\overline{G}$, Damaschke *et al.* first show that there exists a path $P' = (v_1, v_2, \ldots, v_{|V(\overline{G})|})$ of $\overline{G}$ such that v_1 is a minimal vertex of either $\mathcal{P}_G$ or $\mathcal{P}_G^d$; recall that, $\mathcal{P}_G^d$ is the dual partial order of $\mathcal{P}_G$. Using the same arguments, we obtain the following lemma.

Lemma 2.5 *Let G be a comparability graph, and let $\mathcal{P}_G$ be the partial order which corresponds to G. If P is an antipath of G, then there exists an antipath P' of G such that $V(P') = V(P)$ which starts with a minimal vertex of $V(P)$ in $\mathcal{P}_G$.*

Proof. Let $P = (v_1, v_2, \ldots, v_{k-1}, v_k, v_{k+1}, \ldots, v_x)$ be an antipath of a comparability graph G. Let k be the smallest index such that v_k is either a minimal or a maximal vertex of $V(P)$ in $\mathcal{P}_G^d$.

Case (I): Consider first the case where v_k is a minimal vertex of $V(P)$ in $\mathcal{P}_G^d$. We apply Lemma 2.2 to antipath $P_1 = (v_k, v_{k+1}, \ldots, v_x)$ and obtain a monotone antipath $P_1' = (v_k', \ldots, v_x')$ with respect to $\mathcal{P}_G^d$ such that $V(P_1') = V(P_1)$ and $v_k' = v_k$. Therefore, $P_2 = (v_1, v_2, \ldots, v_{k-1}, v_k', \ldots, v_x')$ is an antipath of G such that $V(P_2) = V(P)$. Since $(v_1, v_2, \ldots, v_{k-1})$ contains no maximal vertex of $V(P)$ in $\mathcal{P}_G^d$ and $(v_k', \ldots, v_x')$ is a monotone antipath with respect to $\mathcal{P}_G^d$, it follows that v_x' is a maximal vertex of $V(P) = \{v_1, v_2, \ldots, v_{k-1}, v_k', \ldots, v_x'\}$ in $\mathcal{P}_G^d$, or, equivalently, v_x' is a minimal vertex of $V(P)$ in $\mathcal{P}_G$. Thus, the reversed antipath $P' = (v_x', v_{x-1}', \ldots, v_k', v_{k-1}, \ldots, v_1)$ of P_2 is an antipath of G such that $V(P') = V(P)$ which starts with a minimal vertex of $V(P)$ in $\mathcal{P}_G$.

Case (II): Consider now the case there v_k is a maximal vertex of $V(P)$ in $\mathcal{P}_G^d$. Thus, v_k is a minimal vertex of $V(P)$ in $\mathcal{P}_G$. Following the Case (I), we can obtain an antipath $P_1' = (v_1', v_2', \ldots, v_x')$ of G such that $V(P_1') = V(P)$ which starts with a minimal vertex $v_1' = v_k$ of $V(P)$ in $\mathcal{P}_G^d$. Based on Lemma 2.2, we can obtain a monotone antipath $P_2' = (u_1', u_2', \ldots, u_x')$ of G such that $V(P_2') = V(P_1') = V(P)$ which starts with a minimal vertex $u_1' = v_1'$ of $V(P)$ in $\mathcal{P}_G^d$. Since P_2' is a monotone antipath, u_x' is a maximal vertex of $V(P_2') = V(P)$ in $\mathcal{P}_G^d$, or, equivalently, u_x' is a minimal vertex of $V(P)$ in $\mathcal{P}_G$. Thus, the reversed antipath $P' = (u_x', u_{x-1}', \ldots, u_1)$ of P_2' is an antipath of G such that $V(P') = V(P)$ which starts with a minimal vertex of $V(P)$ in $\mathcal{P}_G$. ∎

The following result is central for the correctness of our algorithm.

Lemma 2.6 *Let P be a longest antipath of a comparability graph G. Then, there exists a normal antipath P' of G such that $V(P') = V(P)$.*

Proof. Let G be a comparability graph, $\mathcal{P}_G$ be the partial order that corresponds to G, σ be the layered ordering of G, and let $P = (v_1, v_2, \ldots, v_k)$ be a longest antipath of G. If $k = 1$, the lemma holds. Suppose that $k \geq 2$. We will prove that for every index i, $2 \leq i \leq k$, there exists an antipath $P_i = (v_1', v_2', \ldots, v_k')$, such that $V(P_i) = V(P)$, v_1' is a leftmost vertex of $V(P_i)$ in σ, and for every index j, $2 \leq j \leq i$, the vertex v'_j is a leftmost vertex of $N_{\overline{G}}(v'_{j-1}) \cap \{v'_j, v'_{j+1}, \ldots, v'_k\}$ in σ. The proof will be done by induction on i.

From Lemma 2.5, we may assume that v_1 is a minimal vertex of $V(P)$ in $\mathcal{P}_G$, and then from Lemma 2.2 we may assume that P is a monotone antipath of G. Thus, for every vertex v_i, $2 \leq i \leq k$, such that $v_i <_\sigma v_1$, we have $v_iv_1 \notin E(G)$. If v_1 is a leftmost vertex of $V(P)$ in σ, then $P_1 = P$. Consider now the case where v_1 is not a leftmost vertex of $V(P)$ in σ. Let j, $2 \leq j \leq k$, be the greatest index such that v_j is a leftmost vertex of $V(P)$ in σ. If $v_1v_{j+1} \notin E(G)$ then $P_1 = (v_j, v_{j-1}, \ldots, v_1, v_{j+1}, \ldots, v_k)$ is an antipath of G such that $V(P_1) = V(P)$ and v_1 is a leftmost vertex of $V(P_1)$ in σ.

Consider now the case where $v_1v_{j+1} \in E(G)$. Since P is monotone and v_1 appears in P before v_{j+1}, we obtain that $v_1 <_\sigma v_{j+1}$. Since $v_j <_\sigma v_1 <_\sigma v_{j+1}$, $v_jv_{j+1} \notin E(G)$, and $v_1v_{j+1} \in E(G)$, from the transitivity property it follows that $v_1v_j \notin E(G)$. Therefore, by the construction of the Hasse diagram of G (and, thus, of σ), there exists a vertex v_x in G, such that $v_x =_\sigma v_1$ and $v_jv_x \in E(G)$; thus, $v_{j+1}v_x \notin E(G)$ due to the transitivity property. If $v_x \notin V(P)$, then $P' = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{j+1}, \ldots, v_k)$ is an antipath of G longer than P. This is a contradiction to our assumption that P is a longest antipath of G, thus, $v_x \in V(P)$. Since P is monotone, $v_jv_x \in E(G)$, and $v_j <_\sigma v_x =_\sigma v_1$, it follows that v_j appears in P before v_x, i.e., $j + 1 \leq x \leq k$. In fact, $j + 2 \leq x \leq k$, since $v_x =_\sigma v_1 <_\sigma v_{j+1}$. Then $P' = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{x-1}, \ldots, v_{j+1})$ is an antipath of G such that $V(P') = V(P) \setminus \{v_{x+1}, v_{x+2}, \ldots, v_k\}$. If $v_{j+1}v_{x+1} \notin E(G)$ then $P_1 = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{x-1}, \ldots, v_{j+1}, v_{x+1}, \ldots, v_k)$ is an antipath of G such that $V(P_1) = V(P)$ and v_j is a leftmost vertex of $V(P_1)$ in σ.

Consider now the case where $v_{j+1}v_{x+1} \in E(G)$. Since P is monotone, $v_{j+1}v_{x+1} \in E(G)$ and v_{j+1} appears in P before v_{x+1}, we have that $v_{j+1} <_\sigma v_{x+1}$; thus, $v_x <_\sigma v_{j+1} <_\sigma v_{x+1}$. Since $v_xv_{j+1} \notin E(G)$, it follows by the construction of the Hasse diagram, that there exists a vertex v_y in G such that $v_y =_\sigma v_{j+1}$ and $v_xv_y \in E(G)$; thus, $v_{x+1}v_y \notin E(G)$ due to the transitivity property. Similarly to the above, $v_y \in V(P)$, since P is a longest antipath of G. Since P is monotone, $v_xv_y \in E(G)$ and $v_x <_\sigma v_y =_\sigma v_{j+1}$, it follows that v_x appears in P before v_y, i.e., $x + 1 \leq y \leq k$ and, in fact, $x + 2 \leq y \leq k$. Therefore, $P' = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{x-1}, \ldots, v_{j+1}, v_y, v_{y-1}, \ldots, v_{x+1})$ is an antipath of G such that $V(P') = V(P) \setminus \{v_{y+1}, v_{y+2}, \ldots, v_k\}$. Again, if $v_{x+1}v_{y+1} \notin E(G)$, then using the above transformation we obtain an antipath P_1. If $v_{x+1}v_{y+1} \in E(G)$, then we can repeat the above procedure until we find a pair of vertices v_{x+1} and v_{y+1} in P such that $v_y <_\sigma v_{x+1}$, $x + 2 \leq y \leq k$, and $v_{x+1}v_{y+1} \notin E(G)$.

Assume that such a pair of vertices v_{x+1} and v_{y+1} does not exists in P, i.e., v_{y+1} is the last vertex v_k of P, $v_y <_\sigma v_{x+1}$, $x + 2 \leq y = k - 1$, and $v_{x+1}v_{y+1} \in E(G)$. Therefore, $P' = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{x-1}, \ldots, v_{j+1}, v_y, v_{y-1}, \ldots, v_{x+1})$ is an antipath of G such that $V(P') = V(P) \setminus \{v_{y+1}\}$ and $y + 1 = k$. Since P is monotone, $v_{x+1}v_{y+1} \in E(G)$, and v_{x+1} appears in P before v_{y+1}, it follows that $v_{x+1} <_\sigma v_{y+1}$; thus, $v_y <_\sigma v_{x+1} <_\sigma v_{y+1}$. Then, similarly to the above, it follows that $v_yv_{x+1} \notin E(G)$, and thus there exists a vertex v_ℓ in G such that $v_{x+1} =_\sigma v_\ell$ and $v_yv_\ell \in E(G)$; thus $v_\ell v_{y+1} \notin E(G)$. Since P is monotone, $v_y <_\sigma v_\ell$ and $v_yv_\ell \in E(G)$, it follows that if $v_\ell \in V(P)$, then v_ℓ appears in P after v_y and, in fact, after v_{y+1}, i.e., $y + 1 < \ell \leq k$. This comes to a contradiction to our assumption that $y+1 = k$, i.e., v_{y+1} is the last vertex v_k of P. Thus, $v_\ell \notin V(P)$ and, therefore, $P' = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{x-1}, \ldots, v_{j+1}, v_y, v_{y-1}, \ldots, v_{x+1}, v_\ell, v_{y+1})$ is an antipath of G longer that P, since $y + 1 = k$ and, thus, $V(P') = V(P) \cup \{v_\ell\}$. This comes to a contradiction to our assumption that P is a longest antipath of G. Therefore, there exists a pair of vertices v_{x+1} and v_{y+1} in P such that $v_y <_\sigma v_{x+1}$, $x + 2 \leq y \leq k$, and $v_{x+1}v_{y+1} \notin E(G)$. Then, $P_1 = (v_j, v_{j-1}, \ldots, v_1, v_x, v_{x-1}, \ldots, v_{j+1}, v_y, v_{y-1}, \ldots, v_{x+1}, v_{y+1}, v_{y+2}, \ldots, v_k)$ is an antipath such that $V(P_1) = V(P)$ and v_j is a leftmost vertex of $V(P_1)$ in σ. This completes the proof for the induction basis.

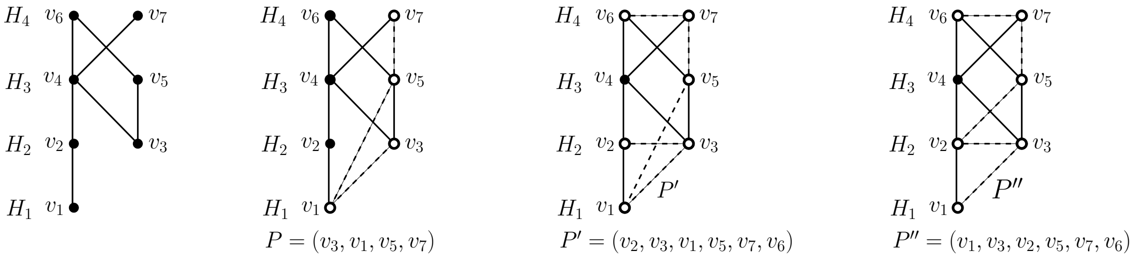

Figure 1: Illustrating a Hasse diagram of a comparability graph G, an antipath P of G which is neither normal nor longest, an antipath P' of G such that $|P'| > |P|$ which is not normal, and a normal antipath P'' of G such that $V(P'') = V(P')$.

Consider now an arbitrary index i, $2 \leq i \leq k-1$, and let $P_i = (v'_1, v'_2, \ldots, v'_i, v'_{i+1}, \ldots, v'_k)$ be an antipath of G, such that $V(P_i) = V(P)$, v'_1 is a leftmost vertex of $V(P_i)$ in σ, and for every index j, $2 \leq j \leq i$, the vertex v'_j is a leftmost vertex of $N_{\overline{G}}(v'_{j-1}) \cap \{v'_j, v'_{j+1}, \ldots, v'_k\}$ in σ. Therefore, the antipath $(v'_1, v'_2, \ldots, v'_i)$ is normal. We now show that v'_i is a minimal vertex of $\{v'_i, v'_{i+1}, \ldots, v'_k\}$ in $\mathcal{P}_G$. Assume otherwise that there exists a vertex $v'_x \in \{v'_{i+1}, v'_{i+2}, \ldots, v'_k\}$, such that $v'_x <_{\mathcal{P}_G} v'_i$ or, equivalently, $v'_x <_\sigma v'_i$ and $v'_x v'_i \in E(G)$. By the induction hypothesis, v'_1 is a leftmost vertex of $V(P)$ in σ and, thus, $v'_1 \leq_\sigma v'_x <_\sigma v'_i$. Since $P' = (v'_1, v'_2, \ldots, v'_i)$ is an antipath of G, $v'_x \notin V(P')$, $v'_x v'_i \in E(G)$, and $v'_1 \leq_\sigma v'_x <_\sigma v'_i$, from Lemma 2.3 we obtain that there exist two consecutive vertices v'_{y-1} and v'_y in P', $2 \leq y \leq i$, such that $v'_{y-1} v'_x \notin E(G)$ and $v'_x <_\sigma v'_y$. This comes to a contradiction to our assumptions, since by the induction hypothesis v'_y is a leftmost vertex of $N_{\overline{G}}(v'_{y-1}) \cap \{v'_y, v'_{y+1}, \ldots, v'_i, \ldots, v'_x, \ldots, v'_k\}$, while $v'_x \in N_{\overline{G}}(v'_{y-1}) \cap \{v'_y, v'_{y+1}, \ldots, v'_i, \ldots, v'_x, \ldots, v'_k\}$ and $v'_x <_\sigma v'_y$. Therefore, we conclude that v'_i is a minimal vertex of $\{v'_i, v'_{i+1}, \ldots, v'_k\}$ in $\mathcal{P}_G$. From Lemma 2.2, for any antipath P of a comparability graph G which starts with a minimal element v of $V(P)$ in $\mathcal{P}_G$, there exists a monotone antipath P'' of G starting with the same vertex v such that $V(P'') = V(P)$. Therefore, without loss of generality we may assume that $\{v'_i, v'_{i+1}, \ldots, v'_k\}$ is a monotone antipath of G. Therefore, by the induction hypothesis it is easy to obtain that the path P_i is a monotone path.

If v'_{i+1} is a leftmost vertex of $N_{\overline{G}}(v'_i) \cap \{v'_{i+1}, v'_{i+2}, \ldots, v'_k\}$ in σ, then $P_{i+1} = P_i$. Consider now the case where v_{i+1} is not a leftmost vertex of $N_{\overline{G}}(v'_i) \cap \{v'_{i+1}, v'_{i+2}, \ldots, v'_k\}$ in σ. Let j, $i+2 \leq j \leq k$, be the greatest index for which v'_j is a leftmost vertex of $N_{\overline{G}}(v'_i) \cap \{v'_{i+1}, v'_{i+2}, \ldots, v'_k\}$ in σ. Then, $P' = (v'_1, v'_2, \ldots, v'_i, v'_j, v'_{j-1}, \ldots, v'_{i+1})$ is an antipath of G such that $V(P') = V(P) \setminus \{v'_{j+1}, v'_{j+2}, \ldots, v'_k\}$. If $v'_{i+1} v'_{j+1} \notin E(G)$, then $P_{i+1} = (v''_1, v''_2, \ldots, v''_i, v''_{i+1}, \ldots, v''_k) = (v'_1, v'_2, \ldots, v'_i, v'_j, v'_{j-1}, \ldots, v'_{i+1}, v'_{j+1}, v'_{j+2}, \ldots, v'_k)$ is an antipath of G such that $V(P_{i+1}) = V(P)$, v''_1 is a leftmost vertex of $V(P_{i+1})$ in σ, and for every index ℓ, $2 \leq \ell \leq i+1$, the vertex v''_ℓ is a leftmost vertex of $N_{\overline{G}}(v''_{\ell-1}) \cap \{v''_\ell, v''_{\ell+1}, \ldots, v''_k\}$ in σ. In the case where $v'_{i+1} v'_{j+1} \in E(G)$, then we repeat exactly the same procedure described in the induction basis until we find a pair of vertices v'_{x+1} and v'_{y+1} in P such that $v'_y <_\sigma v'_{x+1}$, $x+2 \leq y \leq k$, and $v'_{x+1} v'_{y+1} \notin E(G)$; such a pair of vertices exists, as we have proven in the induction basis. Then, $P_{i+1} = (v''_1, v''_2, \ldots, v''_i, v''_{i+1}, \ldots, v''_k) = (v'_1, v'_2, \ldots, v'_i, v'_j, v'_{j-1}, \ldots, v'_{i+1}, \ldots, v'_{x+1}, v'_{y+1}, v'_{y+2}, \ldots, v'_k)$ is an antipath of G such that $V(P_{i+1}) = V(P)$, v''_1 is a leftmost vertex of $V(P_{i+1})$ in σ, and for every index ℓ, $2 \leq \ell \leq i+1$, the vertex v''_ℓ is a leftmost vertex of $N_{\overline{G}}(v''_{\ell-1}) \cap \{v''_\ell, v''_{\ell+1}, \ldots, v''_k\}$ in σ. This completes the proof for the induction step.

Thus, the antipath $P' = P_k$ is a normal antipath of G such that $V(P') = V(P)$. ∎

Figure 1 illustrates a Hasse diagram of a comparability graph G. The antipath $P = (v_3, v_1, v_5, v_7)$ of G is not normal, and there exists no normal antipath $\widehat{P}$ of G such that $V(\widehat{P}) = V(P)$; however, note that P is monotone. Also, P is not a longest antipath of G, since there exists an antipath $P' = (v_2, v_3, v_1, v_5, v_7, v_6)$ of G such that $|P'| > |P|$. Also, P' is not a normal antipath of G and there exists a normal antipath $P'' = (v_1, v_3, v_2, v_5, v_7, v_6)$ of G such that $V(P'') = V(P')$; note that it is easy to see that P'' is a longest antipath of G.

3 The Algorithm

Our algorithm, which we call Algorithm LP_Cocomparability, computes a longest path P of a cocomparability graph G by computing a longest antipath P of the comparability graph $\overline{G}$.

Let G be a comparability graph and let $H_1, H_2, \ldots, H_k$ be the layers of its Hasse diagram. For simplifying our description, we add a dummy vertex u_0 to G such that u_0 belongs to a layer H_0 and $u_0u_i \in E(G)$ for every i, $1 \leq i \leq n$; let G' be the resulting graph. Note that, G' is a comparability graph having a Hasse diagram with layers $H_0, H_1, H_2, \ldots, H_k$, and let σ be a layered ordering of G', where $V(G') = \{u_0, u_1, u_2, \ldots, u_n\}$. It is easy to see that u_0 does not participate in any longest antipath P of G' such that $|P| \geq 2$. In general, a longest antipath P of G' which does not contain the vertex u_0 is also a longest antipath of G. Algorithm LP_Cocomparability computes a longest antipath of G' which is a longest antipath of the original graph G as well. Hereafter, we consider comparability graphs G having assumed that we have already added the dummy vertex u_0. Thus, the antipaths we compute in G are also antipaths of the graph $G \setminus \{u_0\}$.

We next give some definitions and notations necessary for the description of the algorithm. Let $L_j = (v_1, v_2, \ldots, v_k)$ be an arbitrary ordering of the vertices $v_1, v_2, \ldots, v_k$. We denote by $V(L_j)$ the set $\{v_1, v_2, \ldots, v_k\}$ and by $|L_j|$ the cardinality of the set $V(L_j)$. For every vertex $v_z \in L_j$, we denote by $L_j(v_z)$ the ordering $(v_1, v_2, \ldots, v_{z-1}, v_{z+1}, v_{z+2}, \ldots, v_{|L_j|}, v_z)$, and for every index r, $0 \leq r \leq |L_j|$, we denote by $L_j^r(v_z)$ the ordering containing the first r vertices of $L_j(v_z)$; thus:

- $L_j = (v_1, v_2, \ldots, v_k)$,
- $L_j(v_z) = (v_1, v_2, \ldots, v_{z-1}, v_{z+1}, v_{z+2}, \ldots, v_{|L_j|}, v_z)$,
- $L_j^r(v_z) = (v_1, v_2, \ldots, v_r)$ if $1 \leq r \leq z-1$,
- $L_j^r(v_z) = (v_1, v_2, \ldots, v_{z-1}, v_{z+1}, v_{z+2}, \ldots, v_{r+1})$ if $z \leq r \leq |L_j| - 1$,
- $L_j^0(v_z) = \emptyset$, and $L_j^{|L_j|}(v_z) = L_j(v_z)$.

Definition 3.1 *Let G be a comparability graph, let $H_0, H_1, H_2, \ldots, H_k$ be the layers of its Hasse diagram, let $V(G) = \{u_0, u_1, u_2, \ldots, u_n\}$, and let σ be the layered ordering of G. For every triple p, i, and j, where $1 \leq i \leq j \leq k$ and $u_p \in H_{i-1}$, we define the graph $G(u_p, i, j)$ to be the subgraph $G[S]$, where $S = \{u_x : u_x \in H_\ell, i \leq \ell \leq j,$ and $u_pu_x \in E(G)\}$.*

In other words, graph $G(u_p, i, j)$ is the subgraph of G induced by those vertices which belong to the layers $H_\ell, i \leq \ell \leq j$, and which are neighboring vertices of $u_p \in H_{i-1}$.

Definition 3.2 *Let L_j be an ordering of the set $H_j \cap V(G(u_p, i, j))$. We define the graph $G^r_{u_z}(u_p, i, j)$, where $u_z \in L_j$ and $0 \leq r \leq |L_j|$, to be the subgraph $G[S]$, where $S = V(G(u_p, i, j-1)) \cup L_j^r(u_z)$ if $i < j$, and $S = L_j^r(u_z)$ if $i = j$.*

Note that, since the dummy vertex u_0 is adjacent to every other vertex of G, the graph $G(u_0, 1, j)$, $1 \leq j \leq k$, is the subgraph $G[S]$ of G induced by the set $S = \{u_x : u_x \in H_\ell, 1 \leq \ell \leq j\}$. Additionally, $G^{|L_j|}_{u_z}(u_p, i, j) = G(u_p, i, j)$, and if $i < j$, then $G^0_{u_z}(u_p, i, j) = G(u_p, i, j-1)$.

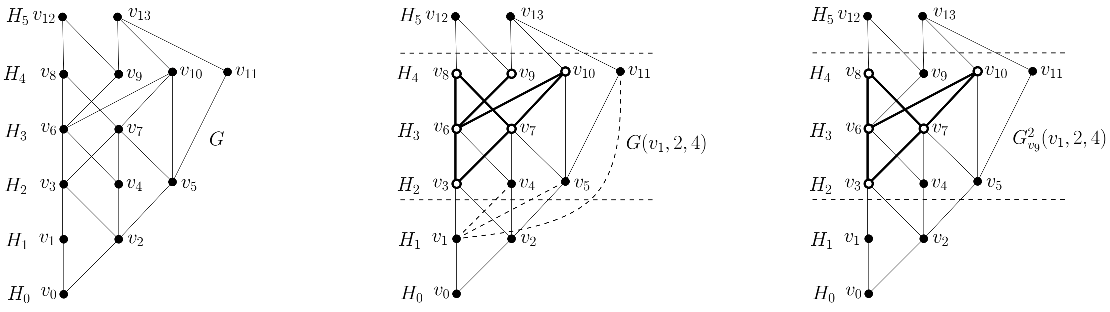

Figure 2: Illustrating a Hasse diagram of a comparability graph G and the induced subgraphs $G(v_1, 2, 4)$ and $G^2_{v_9}(v_1, 2, 4)$ of G.

Figure 2 illustrates examples of the graphs defined in Definitions 3.1 and 3.2. In particular, the figure to the left illustrates a Hasse diagram of a comparability graph G with layers $H_0, H_1, \ldots, H_5$. The figure in the middle illustrates the subgraph $G(v_1, 2, 4)$ of G induced by the vertices $\{v_3, v_6, v_7, v_8, v_9, v_{10}\}$. The figure to the right illustrates the subgraph $G^2_{v_9}(v_1, 2, 4)$ of G, if we consider the ordering $L_4 = (v_8, v_9, v_{10})$ for the vertices of $H_4 \cap V(G(v_1, 2, 4))$. The subgraph $G^2_{v_9}(v_1, 2, 4)$ of G is induced by the vertices $\{v_3, v_6, v_7, v_8, v_{10}\}$, and it is actually an induced subgraph of $G(v_1, 2, 4)$.

Notation 3.1 *For every vertex $u_t \in V(G^r_{u_z}(u_p, i, j))$, if $u_t \in H_j$, then we denote by $f(u_t)$ the smallest index such that $f(u_t) < j$, for which there exists a vertex u_x of $G^r_{u_z}(u_p, i, j)$ such that $u_x \in H_{f(u_t)}$ and $u_x u_t \notin E(G)$; in the case where no such index $f(u_t)$ exists, we set $f(u_t) = j$.*

Notation 3.2 *For every vertex $u_y \in V(G^r_{u_z}(u_p, i, j))$ we denote by $P(u_y; G^r_{u_z}(u_p, i, j))$ a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, and by $\ell(u_y; G^r_{u_z}(u_p, i, j))$ the length of $P(u_y; G^r_{u_z}(u_p, i, j))$.*

Before describing Algorithm LP_Cocomparability in detail, we first give a high level description of our algorithm. A detailed description of Algorithm LP_Cocomparability is presented in Figures 3 and 4.

Algorithm LP_Cocomparability. Given a comparability graph G and its Hasse diagram with $H_0, H_1, \ldots, H_k$, our algorithm computes for every induced subgraph $G(u_p, i, j)$ and for every vertex u_y of $G(u_p, i, j)$, the length $\ell(u_y; G(u_p, i, j))$ and the corresponding antipath $P(u_y; G(u_p, i, j))$, and outputs the maximum among the values $\{\ell(u_y; G(u_0, 1, k)) : u_y \in V(G(u_0, 1, k))\}$ and the corresponding normal antipath $P(u_y; G(u_0, 1, k))$. In particular, our algorithm LP_Cocomparability works as follows:

(A) For every vertex $u_y \in V(G(u_0, 1, k))$

> `compute` a longest normal antipath of $G(u_0, 1, k)$ with right endpoint the vertex u_y, where $G(u_0, 1, k) = G \setminus \{u_0\}$.

(B) `Compute` the longest antipath among the n antipaths computed in (A).

(A.1) A longest normal antipath of $G(u_0, 1, k)$ with right endpoint the vertex u_y can be computed as follows:

► `compute` a longest normal antipath of $G(u_p, i, j)$ with right endpoint u_y, for every subgraph $G(u_p, i, j)$ and for every vertex $u_y \in V(G(u_p, i, j))$, $1 \le i \le j \le k$ and $u_p \in H_{i-1}$, as follows:

let L_j be an ordering of $H_j \cap V(G(u_p, i, j))$;
for every subgraph $G^r_{u_z}(u_p, i, j)$, $1 \le r \le |L_j|$ and $u_z \in L_j$, and for every vertex $u_y \in V(G^r_{u_z}(u_p, i, j))$ such that $u_y \notin L_j \setminus \{u_t\}$ (where u_t is the last vertex of $L^r_j(u_z)$),

`compute` a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, where $G^{|L_j|}_{u_z}(u_p, i, j) = G(u_p, i, j)$, $\forall u_z \in L_j$;

(A.1.1) A longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y can be computed from the normal antipaths of the graph $G^{r-1}_{u_z}(u_p, i, j)$ as follows:

- `compute` $w_1 = \ell(u_x; G^{r-1}_{u_z}(u_p, i, j))$ of path $P'_1 = P(u_x; G^{r-1}_{u_z}(u_p, i, j))$;
- `compute` $w_2 = \ell(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j))$ of path $P'_2 = P(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j))$;
- `if` $w_1 + w_2 + 1 > \ell(u_y; G^r_{u_z}(u_p, i, j))$ `then`

  $\ell(u_y; G^r_{u_z}(u_p, i, j)) \leftarrow w_1 + w_2 + 1$ and $P(u_y; G^r_{u_z}(u_p, i, j)) \leftarrow (P'_1, u_t, P'_2)$;

  where $\ell \ge f(u_t)$, $u_x \in H_\ell \cap V(G^{r-1}_{u_z}(u_p, i, j))$ and $u_x u_t \notin E(G)$;

Step (B) is trivial.

In Section 4.1, we prove that $P(u_y; G(u_0, 1, k))$ is a longest antipath of G. Note that, if P is a longest normal antipath of $G(u_p, i, j)$ with right endpoint the vertex u_y, i.e., $P = P(u_y; G(u_p, i, j))$, then P is not necessarily a longest antipath of $G(u_p, i, j)$. However, if P is a longest antipath of $G(u_p, i, j)$, then from Lemma 2.6 there exists in $G(u_p, i, j)$ a normal antipath P' such that $V(P') = V(P)$; let u_y be the right endpoint of the normal antipath P'. Thus, there exists a longest normal antipath $P' = P(u_y; G(u_p, i, j))$ which is also a longest antipath in $G(u_p, i, j)$ for some vertex $u_y \in V(G(u_p, i, j))$.

4 Correctness and Time Complexity

In this section we prove the correctness of our algorithm and compute its time complexity. In particular, in Section 4.1 we show that Algorithm LP_Cocomparability correctly computes a longest normal antipath P of the comparability graph $\overline{G}$, while in Section 4.2 we analyze and compute the time complexity of our algorithm.

4.1 Correctness of Algorithm LP_Cocomparability

Let G be a comparability graph, let $H_0, H_1, H_2, \ldots, H_k$ be the layers of its Hasse diagram, and let σ be the layered ordering of G. We prove the following results.

Lemma 4.1 *Let L_j be an ordering of the set $H_j \cap V(G(u_p, i, j))$, let $P = (P_1, v_\ell, P_2)$ be a normal antipath of $G^r_{u_z}(u_p, i, j)$, and let v_ℓ be the last vertex of $L^r_j(u_z)$. Then, P_1 and P_2 are normal antipaths of $G^r_{u_z}(u_p, i, j)$.*

Proof. Let $P = (v_1, v_2, \ldots, v_{\ell-1}, v_\ell, v_{\ell+1}, \ldots, v_y)$ be a normal antipath of $G^r_{u_z}(u_p, i, j)$. Then, from Definition 2.2, v_1 is a leftmost vertex of $V(P)$ in σ, and for every index x, $2 \le x \le y$, the vertex v_x is a leftmost vertex of $N_{\overline{G}}(v_{x-1}) \cap \{v_x, v_{x+1}, \ldots, v_y\}$ in σ. It is easy to see that $P_1 = (v_1, v_2, \ldots, v_{\ell-1})$ is a normal antipath of $G^r_{u_z}(u_p, i, j)$. Indeed, since $V(P_1) \subset V(P)$, then v_1 is also a leftmost vertex of $V(P_1)$ in σ and, additionally, v_x is a leftmost vertex of $N_{\overline{G}}(v_{x-1}) \cap \{v_x, v_{x+1}, \ldots, v_{\ell-1}\}$ in σ, for every index x, $2 \le x \le \ell - 1$.

Algorithm LP_Cocomparability

Input: a comparability graph G where $V(G) = \{u_0, u_1, u_2, \dots, u_n\}$, the layers $H_0, H_1, H_2, \dots, H_k$ of its Hasse diagram, and a layered ordering σ of G.

Output: a longest normal antipath of G.

```
for j = 1 to k
   for i = j downto 1
      for every vertex u_p \in H_{i-1}
         let L_j be an ordering of H_j \cap V(G(u_p, i, j))
         for every vertex u_z \in L_j
            for r = 1 to |L_j|
               let u_t be the last vertex of L_j^r(u_z)
               for every vertex u_y \in V(G_{u_z}^r(u_p, i, j)) and y \neq t   {initialization}
                  if r = 1 then
                     \ell(u_y; G_{u_z}^0(u_p, i, j)) \leftarrow \ell(u_y; G(u_p, i, j-1));
                     P(u_y; G_{u_z}^0(u_p, i, j)) \leftarrow P(u_y; G(u_p, i, j-1));
                  \ell(u_y; G_{u_z}^r(u_p, i, j)) \leftarrow \ell(u_y; G_{u_z}^{r-1}(u_p, i, j));
                  P(u_y; G_{u_z}^r(u_p, i, j)) \leftarrow P(u_y; G_{u_z}^{r-1}(u_p, i, j));
               end_for
               if i = j then                                          {case i = j}
                  \ell(u_t; G_{u_z}^r(u_p, j, j)) \leftarrow |L_j^r(u_z)|;
                  P(u_t; G_{u_z}^r(u_p, j, j)) \leftarrow L_j^r(u_z);
               if i \neq j then
                  \ell(u_t; G_{u_z}^r(u_p, i, j)) \leftarrow 1;            {initialization for u_y = u_t}
                  P(u_t; G_{u_z}^r(u_p, i, j)) \leftarrow (u_t);
                  execute process(G_{u_z}^r(u_p, i, j));
            end_for
            \ell(u_z; G(u_p, i, j)) \leftarrow \ell(u_z; G_{u_z}^{|L_j|}(u_p, i, j));   {for the vertex u_z \in L_j}
            P(u_z; G(u_p, i, j)) \leftarrow P(u_z; G_{u_z}^{|L_j|}(u_p, i, j));
         end_for
         for every vertex u_y \in V(G(u_p, i, j)) and u_y \notin L_j          {for u_y \notin L_j}
            \ell(u_y; G(u_p, i, j)) \leftarrow \ell(u_y; G_{u_z}^{|L_j|}(u_p, i, j));
            P(u_y; G(u_p, i, j)) \leftarrow P(u_y; G_{u_z}^{|L_j|}(u_p, i, j));
         end_for
      end_for
   end_for
end_for

compute the max{\ell(u_y; G(u_0, 1, k)) : u_y \in G(u_0, 1, k)} and the corresponding
      antipath P(u_y; G(u_0, 1, k));
```

Figure 3: The algorithm for finding a longest antipath of G.

```
process(G^r_{u_z}(u_p, i, j))

procedure bridge(G^r_{u_z}(u_p, i, j))
if f(u_t) < j then          {u_t is the last vertex of L^r_j(u_z)}
    for h = f(u_t) + 1 to j
        for ℓ = f(u_t) to h − 1
            for every vertex u_x ∈ H_ℓ ∩ V(G^{r−1}_{u_z}(u_p, i, j)) and u_x u_t ∉ E(G)
                for every vertex u_y ∈ H_h ∩ V(G^{r−1}_{u_z}(u_x, ℓ + 1, j))
                    w_1 ← ℓ(u_x; G^{r−1}_{u_z}(u_p, i, j));   P'_1 ← P(u_x; G^{r−1}_{u_z}(u_p, i, j));
                    w_2 ← ℓ(u_y; G^{r−1}_{u_z}(u_x, ℓ + 1, j));   P'_2 ← P(u_y; G^{r−1}_{u_z}(u_x, ℓ + 1, j));
                    if w_1 + w_2 + 1 > ℓ(u_y; G^r_{u_z}(u_p, i, j)) then
                        ℓ(u_y; G^r_{u_z}(u_p, i, j)) ← w_1 + w_2 + 1;
                        P(u_y; G^r_{u_z}(u_p, i, j)) ← (P'_1, u_t, P'_2);

procedure append(G^r_{u_z}(u_p, i, j))
for ℓ = f(u_t) to j          {u_t is the last vertex of L^r_j(u_z)}
    for every vertex u_x ∈ H_ℓ ∩ (V(G^{r−1}_{u_z}(u_p, i, j)) and u_x u_t ∉ E(G)
        w_1 ← ℓ(u_x; G^{r−1}_{u_z}(u_p, i, j));   P'_1 ← P(u_x; G^{r−1}_{u_z}(u_p, i, j));
        if w_1 + 1 > ℓ(u_t; G^r_{u_z}(u_p, i, j)) then
            ℓ(u_t; G^r_{u_z}(u_p, i, j)) ← w_1 + 1;
            P(u_t; G^r_{u_z}(u_p, i, j)) ← (P'_1, u_t);

return (the value ℓ(u_y; G^r_{u_z}(u_p, i, j)) and the antipath P(u_y; G^r_{u_z}(u_p, i, j)), for every
        vertex u_y ∈ V(G^r_{u_z}(u_p, f(u_t) + 1, j)) if f(u_t) < j, and for u_y = u_t if f(u_t) = j);
```

Figure 4: The procedure `process()`.

Consider now the antipath $P_2 = (v_{\ell+1}, v_{\ell+2}, \ldots, v_y)$ of $G^r_{u_z}(u_p, i, j)$. We first prove that $v_{\ell+1}$ is a leftmost vertex of $V(P_2)$ in σ. By assumption $v_\ell \in L_j$, thus, $v_x \leq_\sigma v_\ell$ for every index x, $\ell+1 \leq x \leq y$. We will show that $v_x v_\ell \notin E(G)$, for every index x, $\ell+1 \leq x \leq y$. Let v_x be a vertex of $V(P_2)$. Consider first the case where $v_x =_\sigma v_\ell$; then it is straightforward that $v_x v_\ell \notin E(G)$. Consider now the case where $v_x <_\sigma v_\ell$. Since P is a normal antipath, $v_x <_\sigma v_\ell$, and v_ℓ appears before v_x in P, from Lemma 2.4 we obtain that $v_x v_\ell \notin E(G)$. Thus, we have proved that $v_x v_\ell \notin E(G)$ for every vertex $v_x \in V(P_2)$. Since $v_{\ell+1}$ is a leftmost vertex of $N_{\overline{G}}(v_\ell) \cap \{v_{\ell+1}, v_{\ell+2}, \ldots, v_y\}$ in σ, and since $N_{\overline{G}}(v_\ell) \cap \{v_{\ell+1}, v_{\ell+2}, \ldots, v_y\} = V(P_2)$, it follows that $v_{\ell+1}$ is a leftmost vertex of $V(P_2)$ in σ. Additionally, since P is a normal antipath, it is straightforward that v_x is a leftmost vertex of $N_{\overline{G}}(v_{x-1}) \cap \{v_x, v_{x+1}, \ldots, v_y\}$ in σ, for every index x, $\ell+2 \leq x \leq y$. Therefore, from Definition 2.2 it follows that P_2 is a normal antipath of $G^r_{u_z}(u_p, i, j)$. ∎

The following lemma is important and shows how our algorithm LP_Cocomparability constructs normal antipaths in $G^r_{u_z}(u_p, i, j)$ using normal antipaths of the smaller graph $G^{r-1}_{u_z}(u_p, i, j)$.

Lemma 4.2 *Let L_j be an ordering of the set $H_j \cap V(G(u_p, i, j))$, and let u_t be the last vertex of $L^r_j(u_z)$. Let P_1 be a normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint a vertex u_x such that $u_x \in H_\ell$, $f(u_t) \leq \ell \leq j-1$, and $u_t u_x \notin E(G)$. Let P_2 be a normal antipath of $G^{r-1}_{u_z}(u_x, \ell+1, j)$ with right endpoint a vertex u_y such that $u_y \in H_h$, $\ell+1 \leq h \leq j$, and $V(P_1) \cap V(P_2) = \emptyset$. Then, $P = (P_1, u_t, P_2)$ is a normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y.*

Proof. Let P_1 be a normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint a vertex u_x such that $u_x \in H_\ell$, $f(u_t) \leq \ell \leq j-1$, and $u_t u_x \notin E(G)$, and let P_2 be a normal antipath of $G^{r-1}_{u_z}(u_x, \ell+1, j)$ with right endpoint a vertex u_y such that $u_y \in H_h$, $\ell + 1 \leq h \leq j$, and $V(P_1) \cap V(P_2) = \emptyset$. Since $u_t u_x \notin E(G)$, $u_x <_\sigma u_s \leq_\sigma u_t$ and $u_s u_x \in E(G)$ for every vertex $u_s \in V(P_2)$, it follows that $u_t u_s \notin E(G)$ for every vertex $u_s \in V(P_2)$. Thus, the first vertex of P_2 is an antineighbor of u_t. Therefore, since $V(P_1) \cap V(P_2) = \emptyset$, it follows that $P = (P_1, u_t, P_2)$ is an antipath of G. Additionally, since $u_p <_\sigma u_x <_\sigma u_s$, $u_p u_x \in E(G)$, and $u_x u_s \in E(G)$ for every vertex $u_s \in V(G^{r-1}_{u_z}(u_x, \ell+1, j))$, from the transitivity property we obtain that $u_p u_s \in E(G)$, for every vertex $u_s \in V(P_2)$; thus, for every vertex $u_s \in V(P_2)$, we obtain $u_s \in V(G^{r-1}_{u_z}(u_p, i, j))$. Therefore, since $G^{r-1}_{u_z}(u_p, i, j)$ and $G^{r-1}_{u_z}(u_x, \ell+1, j)$ are induced subgraphs of $G^{r}_{u_z}(u_p, i, j)$, it follows that P is a antipath of $G^{r}_{u_z}(u_p, i, j)$. Hereafter, in the rest of this proof $P_1 = (v_1, v_2, \ldots, v_{q-1})$, $P_2 = (v_{q+1}, v_{q+2}, \ldots, v_s)$, $u_x = v_{q-1}$, $u_y = v_s$, and $u_t = v_q$.

We first show that $P = (v_1, v_2, \ldots, v_q, \ldots, v_s)$ is a normal antipath. Since v_1 is a leftmost vertex of $V(P_1)$ in σ, it follows that $v_1 \leq_\sigma u_x$. Furthermore, since for every vertex $v_k \in V(P_2)$ it holds $u_x <_\sigma v_k$, it follows that v_1 is a leftmost vertex of $V(P)$ in σ. We next show that for every k, $2 \leq k \leq s$, the vertex v_k is a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_s\}$ in σ.

Consider first the case where $2 \leq k \leq q-1$, i.e., $v_k \in V(P_1)$. Since P_1 is a normal antipath, it follows that v_k is a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_{q-1}\}$ in σ. Consider first the case where $v_k \leq_\sigma u_x$. Since $u_x <_\sigma v_{k'}$ for every vertex $v_{k'}$, $q \leq k' \leq s$, it follows that $v_k <_\sigma v_{k'}$. Therefore, in the case where $v_k \leq_\sigma u_x$, we obtain that v_k is also a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_s\}$ in σ. Consider now the case where $u_x <_\sigma v_k$. Since v_q is a rightmost vertex of $V(P)$ is σ, it follows that v_k is a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_{q-1}, v_q\}$ in σ. Now, since $u_x <_\sigma v_k$, and v_k is the next vertex of v_{k-1} in P_1, it follows that $v_{k-1} u_x \in E(G)$. Also, since P_1 is normal, $v_{k-1} u_x \in E(G)$, and v_{k-1} appears before u_x in P_1, from Lemma 2.4 it follows that $v_{k-1} <_\sigma u_x$. Now, since $v_{k-1} <_\sigma u_x <_\sigma v_{k'}$ for every vertex $v_{k'} \in V(P_2)$, $v_{k-1} u_x \in E(G)$, and $u_x v_{k'} \in E(G)$, from the transitivity property it follows that $v_{k-1} v_{k'} \in E(G)$. Thus, for every vertex $v_{k'}$ of P_2, it follows that $v_{k-1} v_{k'} \in E(G)$. Therefore, in the case where $u_x <_\sigma v_k$, we obtain again that v_k is a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_s\}$ in σ. Therefore, in the case where $2 \leq k \leq q-1$, we have proved that v_k is a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_s\}$ in σ.

Consider now the case where $k = q$. Since P_1 is a normal antipath, and for every vertex $v_{k'} \in V(P_2)$ we have that $v_{k'} \in V(G^{r-1}_{u_z}(u_x, \ell+1, j))$, it follows that $v_{k'} u_x \in E(G)$. Therefore, v_q is the only antineighbor of v_{q-1} in $\{v_q, v_{q+1}, \ldots, v_s\}$ and, thus, v_q is a leftmost vertex of $N_{\overline{G}}(v_{q-1}) \cap \{v_q, v_{q+1}, \ldots, v_s\}$ in σ. Now, in the case where $k = q+1$, we have that v_{q+1} is a leftmost vertex of $V(P_2) = \{v_{q+1}, v_{q+2}, \ldots, v_s\}$ in σ, since P_2 is a normal antipath. Therefore, it easily follows that v_{q+1} is a leftmost vertex of $N_{\overline{G}}(v_q) \cap \{v_{q+1}, v_{q+2}, \ldots, v_s\}$ in σ. Finally, in the case where $q+2 \leq k \leq s$, since P_2 is a normal antipath it directly follows that v_k is a leftmost vertex of $N_{\overline{G}}(v_{k-1}) \cap \{v_k, v_{k+1}, \ldots, v_s\}$ in σ. ∎

We next prove the correctness of Algorithm LP_Cocomparability. Let G be a comparability graph, let $H_0, H_1, \ldots, H_k$ be the layers of its Hasse diagram, and let σ be the layered ordering of G.

Hereafter, we distinguish the notation we use for the values computed by Algorithm LP_Cocomparability, from the notation we use for the optimum values. In particular, by $\ell(u_y; G^{r}_{u_z}(u_p, i, j))$ we denote the value computed by Algorithm LP_Cocomparability for the length of a longest normal antipath of $G^{r}_{u_z}(u_p, i, j)$ which has u_y as its right endpoint, and by $P(u_y; G^{r}_{u_z}(u_p, i, j))$ the corresponding computed antipath. On the other hand, by $\mathcal{L}(u_y; G^{r}_{u_z}(u_p, i, j))$ we denote the optimum value of the length of a longest normal antipath of $G^{r}_{u_z}(u_p, i, j)$ which has u_y as its right endpoint, and by $\mathcal{P}(u_y; G^{r}_{u_z}(u_p, i, j))$ the corresponding antipath.

Lemma 4.3 *For every induced subgraph $G(u_p, i, j)$ of G, and for every vertex $u_y \in V(G(u_p, i, j))$, the value $\ell(u_y; G(u_p, i, j))$ computed by Algorithm LP_Cocomparability is equal to the length of a longest normal antipath of $G(u_p, i, j)$ with right endpoint the vertex u_y and, also, the corresponding computed antipath $P(u_y; G(u_p, i, j))$ is a longest normal antipath of $G(u_p, i, j)$ with right endpoint the vertex u_y.*

Proof. The proof of the lemma for every subgraph $G(u_p, i, j)$, $1 \leq i \leq j \leq k$, will be done by induction on the index j, $1 \leq j \leq k$.

Induction basis. We first prove the lemma for $j = 1$, i.e., for the subgraph $G(u_p, 1, 1)$, where $u_p = u_0$ in this case. Let L_1 be an ordering of the set $H_1 \cap V(G(u_p, 1, 1))$. It is easy to see that the length of a longest normal antipath of $G(u_p, 1, 1)$ with right endpoint a vertex $u_z \in L_1$ is equal to $|L_1|$, i.e., $\mathcal{L}(u_z; G(u_p, 1, 1)) = |L_1|$.

Let us now compare this value to the value $\ell(u_z; G(u_p, 1, 1))$ computed by Algorithm LP_Cocomparability in this case. Since $i = j$, it easy to see that for every graph $G^r_{u_z}(u_p, 1, 1)$, $1 \leq r \leq |L_1|$, Algorithm LP_Cocomparability correctly computes and sets (in lines 15-17) $\ell(u_t; G^r_{u_z}(u_p, 1, 1)) = |L^r_1(u_z)|$ and $P(u_t; G^r_{u_z}(u_p, 1, 1)) = L^r_1(u_z)$, where u_t is the last vertex of $L^r_1(u_z)$. In particular, for $r = |L_1|$, we have $\ell(u_t; G^{|L_1|}_{u_z}(u_p, 1, 1)) = |L_1(u_z)|$ and $P(u_t; G^{|L_1|}_{u_z}(u_p, 1, 1)) = L_1(u_z)$. Moreover, Algorithm LP_Cocomparability sets (in lines 23-24) $\ell(u_z; G(u_p, 1, 1)) = \ell(u_z; G^{|L_1|}_{u_z}(u_p, 1, 1))$ and $P(u_z; G(u_p, 1, 1)) = P(u_z; G^{|L_1|}_{u_z}(u_p, 1, 1))$, for every vertex u_z of L_1. Thus, Algorithm LP_Cocomparability correctly computes $\ell(u_z; G(u_p, 1, 1)) = |L_1(u_z)| = |L_1|$ and $P(u_z; G(u_p, 1, 1)) = L_1(u_z)$, for every vertex u_z of L_1. Therefore, $\ell(u_z; G(u_p, 1, 1)) = \mathcal{L}(u_z; G(u_p, 1, 1))$ and $P(u_z; G(u_p, 1, 1)) = \mathcal{P}(u_z; G(u_p, 1, 1))$, for every vertex u_z of L_1; thus, the lemma holds for every subgraph $G^r_{u_z}(u_p, 1, 1)$, $1 \leq r \leq |L_1|$. This proves the induction basis.

Induction hypothesis. Assume now that the lemma holds for every index j', $1 \leq j' \leq j - 1 \leq k - 1$. That is, assume that for every induced subgraph $G(u_p, i', j')$ of G, $1 \leq i' \leq j' \leq j - 1 \leq k - 1$, and for every vertex $u_y \in V(G(u_p, i', j'))$, the value $\ell(u_y; G(u_p, i', j'))$ computed by Algorithm LP_Cocomparability is equal to the length $\mathcal{L}(u_y; G(u_p, i', j'))$ of a longest normal antipath of $G(u_p, i', j')$ with right endpoint the vertex u_y and, also, the corresponding computed antipath $P(u_y; G(u_p, i', j'))$ is a longest normal antipath of $G(u_p, i', j')$ with right endpoint the vertex u_y.

Induction step. We will next show that the lemma holds for $j' = j$, $1 \leq i \leq j \leq k$, i.e., for every induced subgraph $G(u_p, i, j)$ of G.

Case A. Consider first the case where $1 \leq i = j \leq k$. Let L_j be an ordering of the set $H_j \cap V(G(u_p, j, j))$. It is easy to see that the length of a longest normal antipath of $G(u_p, j, j)$ with right endpoint a vertex $u_z \in L_j$ is equal to $|L_j|$, i.e., $\mathcal{L}(u_z; G(u_p, j, j)) = |L_j|$.

Let us now compare this value to the value $\ell(u_z; G(u_p, j, j))$ computed by Algorithm LP_Cocomparability. Let u_t be the last vertex of $L^r_j(u_z)$. We first show that for every graph $G^r_{u_z}(u_p, j, j)$, $1 \leq r \leq |L_j|$, Algorithm LP_Cocomparability correctly computes the values $\ell(u_t; G^r_{u_z}(u_p, j, j))$ and $P(u_t; G^r_{u_z}(u_p, j, j))$. It is easy to see that $\mathcal{L}(u_t; G^r_{u_z}(u_p, j, j)) = |L^r_j(u_z)|$. In the case where $i = j$, Algorithm LP_Cocomparability correctly computes and sets (in lines 15-17) $\ell(u_t; G^r_{u_z}(u_p, j, j)) = |L^r_j(u_z)|$ and $P(u_t; G^r_{u_z}(u_p, j, j)) = L^r_j(u_z)$; note that for $r = |L_j|$, we have $|L^r_j(u_z)| = |L_j|$. Since Algorithm LP_Cocomparability computes these values for every vertex $u_z \in L_j$, i.e., for every subgraph $G^r_{u_z}(u_p, j, j))$, and since $G(u_p, j, j) = G^{|L_j|}_{u_z}(u_p, j, j)$ for any vertex $u_z \in L_j$, it follows that Algorithm LP_Cocomparability correctly computes and sets (in lines 23-24) $\ell(u_z, G(u_p, j, j)) = \ell(u_z, G^{|L_j|}_{u_z}(u_p, j, j))$ and $P(u_z, G(u_p, j, j)) = P(u_z, G^{|L_j|}_{u_z}(u_p, j, j))$, for every vertex $u_z \in L_j$. Thus, the lemma holds for every subgraph $G(u_p, i, j)$ of G such that $1 \leq i = j \leq k$.

Case B. Consider now the case where $1 \leq i < j \leq k$. To prove that the lemma holds in this case, we use the following claim.

Claim 1 *For every induced subgraph $G^r_{u_z}(u_p, i, j)$ of G, $1 \leq i < j \leq k$ and $0 \leq r \leq |L_j|$, and for every vertex $u_y \in V(G^r_{u_z}(u_p, i, j))$ such that $u_y \notin L_j \setminus \{u_t\}$, where u_t is the last vertex of $L^r_j(u_z)$, Algorithm LP_Cocomparability correctly computes $\ell(u_y; G^r_{u_z}(u_p, i, j))$ and $P(u_y; G^r_{u_z}(u_p, i, j))$.*

Recall that $G(u_p, i, j) = G^{|L_j|}_{u_z}(u_p, i, j)$ for any vertex $u_z \in L_j$. Then, on the one hand, for the length of a longest normal antipath of $G(u_p, i, j)$ with right endpoint a vertex $u_y \in V(G(u_p, i, j))$ such that $u_y \notin L_j$, we have that $\mathcal{L}(u_y; G(u_p, i, j)) = \mathcal{L}(u_y; G^{|L_j|}_{u_z}(u_p, i, j))$, where u_z is any vertex of L_j. Thus, from Claim 1 we obtain that $\ell(u_y; G^{|L_j|}_{u_z}(u_p, i, j)) = \mathcal{L}(u_y; G(u_p, i, j))$, where u_z is any vertex of L_j. It is easy to see that, for every vertex u_y of $G(u_p, i, j)$ such that $u_y \notin L_j$, Algorithm LP_Cocomparability sets (in lines 26-28) $\ell(u_y; G(u_p, i, j)) = \ell(u_y; G^{|L_j|}_{u_z}(u_p, i, j))$ and $P(u_y; G(u_p, i, j)) = P(u_y; G^{|L_j|}_{u_z}(u_p, i, j))$, where u_z is any vertex of L_j.

On the other hand, for the length of a longest normal antipath of $G(u_p, i, j)$ with right endpoint a vertex $u_z \in L_j$, from Claim 1 we obtain that $\ell(u_z; G^{|L_j|}_{u_z}(u_p, i, j)) = \mathcal{L}(u_z; G(u_p, i, j))$. Since the procedure `process()` is executed for every vertex $u_z \in L_j$, i.e., for every subgraph $G^r_{u_z}(u_p, i, j))$, it follows that Algorithm LP_Cocomparability correctly computes and sets (in lines 23-24) $\ell(u_z; G(u_p, i, j)) = \ell(u_z; G^{|L_j|}_{u_z}(u_p, i, j))$ and $P(u_z; G(u_p, i, j)) = P(u_z; G^{|L_j|}_{u_z}(u_p, i, j))$ for every vertex $u_z \in L_j$.

It is now clear that Algorithm LP_Cocomparability correctly computes the length of a longest normal antipath of $G(u_p, i, j)$ with right endpoint a vertex u_y, for every vertex $u_y \in V(G(u_p, i, j))$. This proves the lemma. ∎

We next prove by induction that the `process()` of algorithm LP_Cocomparability, i.e., procedures `bridge()` and `append()`, correctly computes the length $\ell(u_y; G^r_{u_z}(u_p, i, j))$ of a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, and the corresponding antipath $P(u_y; G^r_{u_z}(u_p, i, j))$.

Proof of Claim 1.
For proving the claim we use the induction hypothesis of Lemma 4.3. That is, we assume that for every induced subgraph $G(u_p, i', j')$ of G, $1 \leq i' \leq j' \leq j - 1 \leq k - 1$, and for every vertex $u_y \in V(G(u_p, i', j'))$, the value $\ell(u_y; G(u_p, i', j'))$ computed by Algorithm LP_Cocomparability is equal to the length $\mathcal{L}(u_y; G(u_p, i', j'))$ of a longest normal antipath of $G(u_p, i', j')$ with right endpoint the vertex u_y and, also, the corresponding computed antipath $P(u_y; G(u_p, i', j'))$ is a longest normal antipath of $G(u_p, i', j')$ with right endpoint the vertex u_y.

Let $G^r_{u_z}(u_p, i, j)$ be an induced subgraph of G such that $1 \leq i < j \leq k$ and $0 \leq r \leq |L_j|$. We prove the claim by induction on the index r, $0 \leq r \leq |L_j|$, i.e., we prove that for every induced subgraph $G^r_{u_z}(u_p, i, j)$ of G, $1 \leq i < j \leq k$ and $0 \leq r \leq |L_j|$, and for every vertex $u_y \in V(G^r_{u_z}(u_p, i, j))$ such that $u_y \notin L_j \setminus \{u_t\}$, where u_t is the last vertex of $L^r_j(u_z)$, Algorithm LP_Cocomparability correctly computes $\ell(u_y; G^r_{u_z}(u_p, i, j))$ and $P(u_y; G^r_{u_z}(u_p, i, j))$.

To this end, we distinguish three cases concerning the position of the vertex u_y in the ordering σ. In each case, we examine first the optimum value of the length $\mathcal{L}(u_y; G^r_{u_z}(u_p, i, j))$ of a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, and then we compare this value to the value $\ell(u_y; G^r_{u_z}(u_p, i, j))$ computed by Algorithm LP_Cocomparability; we also compare the corresponding antipaths.

Induction basis. Consider first the case where $r = 0$, i.e., $L^0_j = \emptyset$. Since in this claim we examine the case where $i \neq j$, from Definition 3.2 we obtain that $G^0_{u_z}(u_p, i, j) = G(u_p, i, j - 1)$. Therefore, it is easy to see that for every subgraph $G^0_{u_z}(u_p, i, j)$, and for every vertex $u_y \in V(G^0_{u_z}(u_p, i, j))$, the length $\mathcal{L}(u_y; G^0_{u_z}(u_p, i, j))$ is equal to $\mathcal{L}(u_y; G(u_p, i, j-1))$.

Algorithm LP_Cocomparability sets (in lines 8-11) $\ell(u_y; G^0_{u_z}(u_p, i, j)) = \ell(u_y; G(u_p, i, j-1))$ and $P(u_y; G^0_{u_z}(u_p, i, j)) = P(u_y; G(u_p, i, j-1))$, for every vertex $u_y \in V(G^0_{u_z}(u_p, i, j))$. Since by the induction hypothesis of Lemma 4.3, Algorithm LP_Cocomparability correctly computes the values of $\ell(u_y; G(u_p, i, j-1))$ and $P(u_y; G(u_p, i, j-1))$, it follows that the algorithm also correctly computes the values of $\ell(u_y; G^0_{u_z}(u_p, i, j))$ and $P(u_y; G^0_{u_z}(u_p, i, j))$. Therefore, the claim holds for $r = 0$.

Induction hypothesis. Suppose now that the claim holds for every index ℓ, $0 \leq \ell \leq r-1 \leq |L_j| - 1$.

Induction step. We will now prove that the claim holds for the subgraph $G^r_{u_z}(u_p, i, j)$ of G, $1 \leq r \leq |L_j|$. To this end, we distinguish three cases concerning the position of the vertex u_y in the ordering σ:

Case 1: $u_y \in H_\ell$, $i \leq \ell \leq f(u_t)$,
Case 2: $u_y \in H_\ell$, $f(u_t) + 1 \leq \ell \leq j - 1$, and
Case 3: $u_y = u_t$.

Note that Cases 1 and 2 correspond to the procedure `bridge()` of algorithm LP_Cocomparability, while Case 3 corresponds to procedure `append()`.

Case 1. We consider first the case where $u_y \in H_\ell$ and $i \leq \ell \leq f(u_t)$. In this case we have $\mathcal{L}(u_y; G^r_{u_z}(u_p, i, j)) = \mathcal{L}(u_y; G^{r-1}_{u_z}(u_p, i, j))$, since from Definition 2.2 and Notation 3.1 we obtain that u_t does not belong to any normal antipath with right endpoint a vertex $u_y \in H_\ell$, $i \leq \ell \leq f(u_t)$.

In this case, Algorithm LP_Cocomparability computes and sets (in lines 8-14) $\ell(u_y; G^r_{u_z}(u_p, i, j)) = \ell(u_y; G^{r-1}_{u_z}(u_p, i, j))$ for the length of a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint a vertex $u_y \in H_\ell$, $i \leq \ell \leq f(u_t)$; the algorithm also computes the corresponding antipath. Note that, this computation is done during the initialization (in lines 8-14), and these values do not change during the `process()` of the algorithm, since $u_y \in H_\ell$ and $\ell < f(u_t) + 1$. Since by the induction hypothesis Algorithm LP_Cocomparability correctly computes and sets $\ell(u_y; G^{r-1}_{u_z}(u_p, i, j) = \mathcal{L}(u_y; G^{r-1}_{u_z}(u_p, i, j)$, for every vertex $u_y \in G^{r-1}_{u_z}(u_p, i, j)$ such that $u_y \notin L_j$, it follows that Algorithm LP_Cocomparability correctly computes the values of $\ell(u_y; G^r_{u_z}(u_p, i, j))$ and $P(u_y; G^r_{u_z}(u_p, i, j))$.

Case 2. We consider next the case where $u_y \in H_h$ and $f(u_t)+1 \leq h \leq j-1$. Let $P = (u_{x'}, \ldots, u_y)$ be a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ which has u_y as its right endpoint. We now distinguish two cases: (I) the case where P contains the vertex u_t, and (II) the case where P does not contain the vertex u_t.

2(I) Consider first the case where P contains the vertex u_t. Assume first that $P = (u_t, u_y)$ is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y. From Definition 2.2 we obtain that P is not normal; this is a contradiction to our assumption on P.

Assume now that $P = (u_{x'}, \ldots, u_x, u_t, u_{y'}, \ldots, u_y) = (P_1, u_t, P_2)$ is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y. From Lemma 4.1, we obtain that $P_1 = (u_{x'}, \ldots, u_x)$ and $P_2 = (u_{y'}, \ldots, u_y)$ are normal antipaths of $G^r_{u_z}(u_p, i, j)$ and, in fact, of $G^{r-1}_{u_z}(u_p, i, j)$. We next prove the following results (Claims 2, 3, and 4) for the antipaths P, P_1, and P_2.

Claim 2 *Let P, P_1, and P_2 be the antipaths of Case 2(I). Then, for every vertex $u_s \in V(P_2)$ we have $u_x <_\sigma u_s$ and $u_x u_s \in E(G)$, where u_x is the right endpoint of P_1.*

Proof of Claim 2. Let $P = (P_1, u_t, P_2)$ and let u_x be the right endpoint of P_1. Since $u_t \in L_j$ and P is an antipath of $G^r_{u_z}(u_p, i, j)$, it follows that $u_s \leq_\sigma u_t$ for every vertex $u_s \in V(P_2)$.

(i) Consider first the case where u_s is a vertex of P_2 such that $u_s <_\sigma u_t$. Since P is normal and u_t is the next vertex of u_x in P, it follows that $u_x u_s \in E(G)$ for every vertex $u_s \in V(P_2)$ such that $u_s <_\sigma u_t$. Since P is normal, $u_x u_s \in E(G)$, and u_x appears before u_s in P, from Lemma 2.4 we obtain that $u_x <_\sigma u_s$, for every vertex $u_s \in V(P_2)$ such that $u_s <_\sigma u_t$. Therefore, we have proved that for every vertex $u_s \in V(P_2)$ such that $u_s <_\sigma u_t$, we have $u_x <_\sigma u_s$ and $u_x u_s \in E(G)$.

(ii) Consider now the case where u_s is a vertex of P_2 such that $u_s =_\sigma u_t$. Since u_y is a vertex of P_2 such that $u_y <_\sigma u_t$, from case (i) of Claim 2 we obtain that $u_x <_\sigma u_y$. Since $u_x <_\sigma u_y <_\sigma u_t =_\sigma u_s$, it follows that $u_x <_\sigma u_s$, for every vertex $u_s \in V(P_2)$ such that $u_s =_\sigma u_t$. It is left to show that the property $u_x u_s \in E(G)$ holds for every vertex $u_s \in V(P_2)$ such that $u_s =_\sigma u_t$. Assume that P is an antipath for which this property does not hold. We next show that there exists a longest normal antipath P' of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, such that $P' = (P_1, P'_2)$ and $V(P') = V(P)$, for which the property $u_x u_s \in E(G)$ holds for every vertex $u_s \in V(P_2)$ such that $u_s =_\sigma u_t$.

Let u_s be a vertex of P_2 such that $u_s =_\sigma u_t$ and $u_x u_s \notin E(G)$. Let $P = (P_1, u_t, P_2) = (P_1, u_t, u_{y'} \ldots, u_{s'}, u_s, u_{s''}, \ldots, u_y)$, and let u_s be the last such vertex in P. Then $P' = (P_1, P'_2) = (P_1, u_s, u_{y'} \ldots, u_{s'}, u_t, u_{s''}, \ldots, u_y)$ is an antipath, since we next prove that both u_t and u_s are connected with an antiedge to every vertex of P_2. To this end, let u_q be a vertex of P_2 such that $q \neq s$. If $u_q =_\sigma u_t$, then indeed $u_q u_t \notin E(G)$ and $u_q u_s \notin E(G)$. If $u_q <_\sigma u_t$, then from case (i) of Claim 2 we obtain that $u_x <_\sigma u_q$ and $u_x u_q \in E(G)$. Since $u_x <_\sigma u_q <_\sigma u_t$, $u_x u_q \in E(G)$, and $u_x u_t \notin E(G)$, from the transitivity property we obtain $u_q u_t \notin E(G)$; using the same arguments we obtain that $u_q u_s \notin E(G)$. Therefore, since $u_{y'}$, $u_{s'}$, and $u_{s''}$ are vertices of P_2, we obtain that $P' = (P_1, u_s, u_{y'} \ldots, u_{s'}, u_t, u_{s''}, \ldots, u_y)$ is a longest antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y. It is easy to see that P' is normal, since P is normal and $u_t =_\sigma u_s$. Additionally, since P' is normal, it is easy to obtain that the property of case (i) of Claim 2 holds for P' as well, where $P' = (P_1, u_s, \widetilde{P_2})$. By repeating the above procedure we can obtain a longest normal antipath $P' = (P'_1, u_t, P'_2)$ with right endpoint the vertex u_y such that $u_x u_s \in E(G)$ for every vertex $u_s \in V(P'_2)$ such that $u_s =_\sigma u_t$, where u_x is the last vertex of P'_1.

Therefore, we may assume without loss of generality that $P = (u_{x'}, \ldots, u_x, u_t, u_{y'}, \ldots, u_y) = (P_1, u_t, P_2)$ is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, with the property that $u_x u_s \in E(G)$ for every vertex $u_s \in V(P_2)$ such that $u_s =_\sigma u_t$. Thus, we have proved that $u_x <_\sigma u_s$ and $u_x u_s \in E(G)$, for every vertex $u_s \in V(P_2)$; *QED of Claim 2.* ∎

Claim 3 *Let P, P_1, and P_2 be the antipaths of Case 2(I). Then, P_1 is a normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ which has u_x as its right endpoint, and P_2 is a normal antipath of $G^{r-1}_{u_z}(u_x, \ell+1, j)$ which has u_y as its right endpoint.*

Proof of Claim 3. By assumption, for every vertex $u_q \in V(P_1)$ we have $u_q \in V(G^{r-1}_{u_z}(u_p, i, j))$, and u_x is the right endpoint of P_1.

Since $u_x <_\sigma u_y$ and by assumption $u_y \in H_h$, $f(u_t) + 1 \leq h \leq j - 1$, we obtain that $u_x \in H_\ell$, where $f(u_t) \leq \ell \leq j-2$. Then, from Claim 2 we obtain that $u_s \in H_h$, $\ell+1 \leq h \leq j$, for every vertex $u_s \in V(P_2)$. Additionally, since from Claim 2 we have $u_x u_s \in E(G)$ for every vertex $u_s \in V(P_2)$, it follows from Definition 3.2 that $u_s \in V(G^{r-1}_{u_z}(u_x, \ell+1, j))$ for every vertex $u_s \in V(P_2)$. Note that, every vertex u_s of $G^{r-1}_{u_z}(u_x, \ell+1, j)$ is also a vertex of $G^{r-1}_{u_z}(u_p, i, j)$. Indeed, since $i < \ell+1 \leq j$, and since $u_p <_\sigma u_x <_\sigma u_s$, $u_p u_x \in E(G)$, and $u_x u_s \in E(G)$, from the transitivity property we obtain that $u_p u_s \in E(G)$. Therefore, we have shown that every vertex of P_2 belongs to $G^{r-1}_{u_z}(u_x, \ell+1, j)$; *QED of Claim 3.* ∎

Claim 4 *If P_1 is a normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ which has u_x as its right endpoint, and P_2 is a normal antipath of $G^{r-1}_{u_z}(u_x, \ell+1, j)$ which has u_y as its right endpoint, then $V(P_1) \cap V(P_2) = \emptyset$.*

Proof of Claim 4. Let $\mathcal{H}_2$ be the subgraph of $G^r_{u_z}(u_p, i, j)$ induced by $V(\mathcal{H}_2) = V(G^{r-1}_{u_z}(u_x, \ell+1, j))$; recall from Claim 3 that every vertex of P_2 belongs to $\mathcal{H}_2$.

Let u_q be a vertex of P_1. If $u_q \leq_\sigma u_x$, then $u_q \in H_d$ and $d \leq \ell$; thus, from Definition 3.2 we obtain that $u_q \notin V(\mathcal{H}_2) = V(G^{r-1}_{u_z}(u_x, \ell+1, j))$. Consider now the case where u_q is a vertex of P_1 such that $u_x <_\sigma u_q$. Since P_1 is a normal antipath, $u_x <_\sigma u_q$, and u_q appears before u_x in P_1, from Lemma 2.4 we obtain that $u_x u_q \notin E(G)$. Therefore, from Definition 3.2 we obtain again that $u_q \notin V(\mathcal{H}_2) = V(G^{r-1}_{u_z}(u_x, \ell+1, j))$. Therefore, we have proved that no vertex of P_1 belongs to $\mathcal{H}_2$. Let $\mathcal{H}_1$ be the subgraph of $G^r_{u_z}(u_p, i, j)$ induced by $V(\mathcal{H}_1) = V(G^{r-1}_{u_z}(u_p, i, j)) \setminus V(G^{r-1}_{u_z}(u_x, \ell+1, j))$. Thus, we have shown that every vertex of P_1 belongs to $\mathcal{H}_1$.

Therefore, we have shown that $V(P_1) \subseteq V(\mathcal{H}_1)$, $V(P_2) \subseteq V(\mathcal{H}_2)$, and $V(\mathcal{H}_1) \cap V(\mathcal{H}_2) = \emptyset$. It is easy to see that $V(P_1) \cap V(P_2) = \emptyset$; *QED of Claim 4.* ∎

Since $P = (P_1, u_t, P_2)$ is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, and since the antipaths P_1 and P_2 belong to two disjoint induced subgraphs of $G^r_{u_z}(u_p, i, j)$, it follows that P_1 is a longest normal antipath of H_1 with right endpoint the vertex u_x, and that P_2 is a longest normal antipath of H_2 with right endpoint the vertex u_y. Note that, u_t is connected with an antiedge to every vertex u_s of H_2 and, thus, also of P_2. Indeed, in the case where $u_s =_\sigma u_t$ this is straightforward. In the case where $u_s <_\sigma u_t$, then from Claim 2 we have and $u_x <_\sigma u_s <_\sigma u_t$ and $u_x u_s \in E(G)$ for every vertex u_s of P_2; since $u_x u_t \notin E(G)$, from the transitivity property we obtain that $u_s u_t \notin E(G)$ for every vertex u_s of P_2.

Therefore, since $H_2 = G^{r-1}_{u_z}(u_x, \ell+1, j)$, we obtain that $|P_2| = \mathcal{L}(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j))$. We will now show that $|P_1| = \mathcal{L}(u_x; G^{r-1}_{u_z}(u_p, i, j))$. To this end, let P_0 be a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint the vertex u_x. Assume that there exists a vertex $u_s \in V(P_0)$ such that $u_s \in V(\mathcal{H}_2) = V(G^{r-1}_{u_z}(u_x, \ell+1, j))$. Since $u_x \in H_\ell$, it follows that $u_x <_\sigma u_s$ and $u_x u_s \in E(G)$. Since P_0 is normal, from Lemma 2.4 we obtain that u_x appears before u_s in P_0. This comes to a contradiction to our assumption that u_x is the right endpoint of P_0. Thus, no vertex of P_0 belongs to $\mathcal{H}_2$. Thus, $V(P_0) \subseteq V(\mathcal{H}_1)$, and since P_1 is a longest normal antipath of H_1 with right endpoint the vertex u_x, we obtain that $|P_0| \subseteq |P_1|$. Additionally, since $\mathcal{H}_1$ is an induced subgraph of $G^{r-1}_{u_z}(u_p, i, j)$, we obtain that $|P_1| \subseteq |P_0|$. Thus, $|P_0| = |P_1|$ and, therefore, P_1 is a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint the vertex u_x. Thus, $|P_1| = \mathcal{L}(u_x; G^{r-1}_{u_z}(u_p, i, j))$.

Therefore, if $P = (P_1, u_t, P_2)$ is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint a vertex $u_y \in H_h$, $f(u_t) + 1 \leq h \leq j - 1$, we have shown that $|P| = \mathcal{L}(u_y; G^r_{u_z}(u_p, i, j)) = \mathcal{L}(u_x; G^{r-1}_{u_z}(u_p, i, j)) + \mathcal{L}(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j)) + 1$ and $P = \mathcal{P}(u_y; G^r_{u_z}(u_p, i, j)) = (\mathcal{P}(u_x; G^{r-1}_{u_z}(u_p, i, j)), u_t, \mathcal{P}(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j)))$.

We next examine the results computed by Algorithm LP_Cocomparability in Case 2(I). Let $P(u_y; G^r_{u_z}(u_p, i, j))$ be the antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint a vertex u_y computed by Algorithm LP_Cocomparability, in the case where $u_y \in H_h$, $f(u_t) + 1 \leq h \leq j - 1$. Note that the antipath $P(u_y; G^r_{u_z}(u_p, i, j))$, which is computed by the algorithm with the procedure `bridge()`, contains the vertex u_t. In fact, Algorithm LP_Cocomparability computes and sets $P(u_y; G^r_{u_z}(u_p, i, j)) = (P'_1, u_t, P'_2)$, where u_t is the last vertex of $L^r_j(u_z)$, and for the two antipaths P'_1 and P'_2 we have: $P'_1 = P(u_x; G^{r-1}_{u_z}(u_p, i, j))$, where $u_x \in H_\ell$, $f(u_t) \leq \ell \leq j - 2$, and $u_x u_t \notin E(G)$, and $P'_2 = P(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j))$, where $u_y \in H_h$, $\ell + 1 \leq h \leq j - 1$. Actually, in this case, Algorithm LP_Cocomparability computes with the procedure `bridge()` the value $w_1 + w_2 + 1 = |P'_1| + |P'_2| + 1$, for every vertex u_x such that $u_x \in H_\ell$, $f(u_t) \leq \ell \leq j-2$, and $u_x u_t \notin E(G)$, and sets $\ell(u_y; G^r_{u_z}(u_p, i, j))$ to be equal to the maximum among these values. Also, Algorithm LP_on_H computes the corresponding antipath $P(u_y; G^r_{u_z}(u_p, i, j)) = (P'_1, u_t, P'_2)$.

By the induction hypothesis, we obtain that Algorithm LP_Cocomparability has correctly computed the values $P'_1 = P(u_x; G^{r-1}_{u_z}(u_p, i, j))$ and $P'_2 = P(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j))$, i.e., $P'_1 = \mathcal{P}(u_x; G^{r-1}_{u_z}(u_p, i, j))$ and $P'_2 = \mathcal{P}(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j))$. Therefore, from Claim 3 we obtain that $V(P'_1) \cap V(P'_2) = \emptyset$. Then, from Lemma 4.2 we obtain that the antipath $P(u_y; G^r_{u_z}(u_p, i, j)) =$

(P'_1, u_t, P'_2) computed by Algorithm LP_Cocomparability is a normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y. Moreover, since Algorithm LP_Cocomparability computes with the procedure `bridge()` the value $\ell(u_x; G^{r-1}_{u_z}(u_p, i, j)) + \ell(u_y; G^{r-1}_{u_z}(u_x, \ell+1, j)) + 1$, for every vertex u_x such that $u_x \in H_\ell$, $f(u_t) \leq \ell \leq j-2$, and $u_x u_t \notin E(G)$, and sets $\ell(u_y; G^r_{u_z}(u_p, i, j))$ to be equal to the maximum among these values, it follows that $\ell(u_y; G^r_{u_z}(u_p, i, j)) = \mathcal{L}(u_y; G^r_{u_z}(u_p, i, j))$. Also, the corresponding antipath $P(u_y; G^r_{u_z}(u_p, i, j) = (P(u_x; G^{r-1}_{u_z}(u_p, i, j)), u_t, P(u_y; G^{r-1}_{u_z}(u_x, \ell + 1, j)))$ computed by Algorithm LP_Cocomparability is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y.

2(II) Consider now the case where there exists a longest normal antipath P of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y which does not contain the vertex u_t. Then $V(P) \subseteq V(G^{r-1}_{u_z}(u_p, i, j))$ and, thus, P is a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y, i.e., $P = \mathcal{P}(u_y; G^{r-1}_{u_z}(u_p, i, j))$.

We next examine the results computed by Algorithm LP_Cocomparability in Case 2(II). By the induction hypothesis, we obtain that Algorithm LP_Cocomparability correctly computes $\ell(u_y; G^{r-1}_{u_z}(u_p, i, j)) = \mathcal{L}(u_y; G^{r-1}_{u_z}(u_p, i, j))$ for every vertex $u_y \in G^{r-1}_{u_z}(u_p, i, j)$ such that $u_y \notin L_j \setminus \{u_t\}$. Observe first that during the initialization (in lines 8-14) the algorithm sets $\ell(u_y; G^r_{u_z}(u_p, i, j)) = \ell(u_y; G^{r-1}_{u_z}(u_p, i, j))$ for every vertex $u_y \in H_h$, $f(u_t) + 1 \leq h \leq j-1$. Next, it suffices to show that these values do not change during the execution of the `process()`.

From Lemma 4.2 (since from Claim 4 we have $V(P'_1) \cap V(P'_2) = \emptyset$), we obtain that the antipaths (P'_1, u_t, P'_2) constructed by Algorithm LP_Cocomparability, during the execution of the procedure `bridge()`, are normal antipaths of $G^r_{u_z}(u_p, i, j)$ with right endpoint a vertex u_y. Therefore, since we have assumed that the longest normal antipath P of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_y does not contain the vertex u_t, it directly follows that no (normal) antipath (P'_1, u_t, P'_2) with right endpoint the vertex u_y which is constructed with the procedure `bridge()` is longer than P. Thus, since $|P|$ is the initial value given to $\ell(u_y; G^r_{u_z}(u_p, i, j))$ (during the initialization in lines 8-14), it follows that the statement $w_1 + w_2 + 1 > \ell(u_y; G^r_{u_z}(u_p, i, j))$ (in the procedure `bridge()`) is false for every vertex $u_x \in H_\ell$ such that $f(u_t) \leq \ell \leq h-1$ and $u_t u_x \notin E(G)$. Therefore, the initial value of $\ell(u_y; G^r_{u_z}(u_p, i, j))$ does not change during the execution of the `process()`.

Thus, Algorithm LP_Cocomparability correctly computes $\ell(u_y; G^r_{u_z}(u_p, i, j)) = \ell(u_y; G^{r-1}_{u_z}(u_p, i, j))$ and $P(u_y; G^r_{u_z}(u_p, i, j)) = P(u_y; G^{r-1}_{u_z}(u_p, i, j))$; recall that, by the induction hypothesis, $\ell(u_y; G^{r-1}_{u_z}(u_p, i, j)) = \mathcal{L}(u_y; G^{r-1}_{u_z}(u_p, i, j))$ and $P(u_y; G^{r-1}_{u_z}(u_p, i, j)) = \mathcal{P}(u_y; G^{r-1}_{u_z}(u_p, i, j))$.

Concluding, in both Cases 2(I) and 2(II), we have proved that the antipath $P(u_y; G^r_{u_z}(u_p, i, j))$ computed by Algorithm LP_Cocomparability is a longest normal antipath $\mathcal{P}(u_y; G^r_{u_z}(u_p, i, j))$ of $G^r_{u_z}(u_p, i, j)$ with u_y as its right endpoint, and $\ell(u_y; G^r_{u_z}(u_p, i, j)) = \mathcal{L}(u_y; G^r_{u_z}(u_p, i, j))$. Thus, the claim holds in Case 2.

Case 3. Consider now the case where $u_y = u_t$.

3(I) Assume first that u_t has no antineighbors in $G^r_{u_z}(u_p, i, j)$. Then $\mathcal{P}(u_t; G^r_{u_z}(u_p, i, j)) = (u_t)$ is a longest normal antipath of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_t.

Since we examine the case where $i \neq j$, it is easy to see that Algorithm LP_Cocomparability sets (in lines 19-20) $\ell(u_t; G^r_{u_z}(u_p, i, j)) = 1$ and $P(u_t; G^r_{u_z}(u_p, i, j)) = (u_t)$. Since u_t has no antineighbors in $G^r_{u_z}(u_p, i, j)$, it follows that $r = 1$ and $f(u_t) = j$. Thus, the initial value of $\ell(u_t; G^r_{u_z}(u_p, i, j))$ does not change during the execution of the `process()`. Therefore, Algorithm LP_Cocomparability correctly computes the values of $\ell(u_t; G^r_{u_z}(u_p, i, j))$ and $P(u_t; G^r_{u_z}(u_p, i, j))$ in the case where u_t has no antineighbors in $G^r_{u_z}(u_p, i, j)$.

3(II) Assume now that u_t has at least one antineighbor in $G^r_{u_z}(u_p, i, j)$. Let $P = (u_{x'}, \ldots, u_x, u_t) = (P', u_t)$ be a longest normal antipath $\mathcal{P}(u_t; G^r_{u_z}(u_p, i, j))$ of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_t. Then, it is easy to see that P' is a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint the vertex u_x, i.e., $P' = \mathcal{P}(u_x; G^{r-1}_{u_z}(u_p, i, j))$.

In Case 3(II), Algorithm LP_Cocomparability computes (with the procedure `append()`) the value $w_1 + 1 = \ell(u_x; G^{r-1}_{u_z}(u_p, i, j)) + 1$, for every vertex $u_x \in H_\ell \cap V(G^{r-1}_{u_z}(u_p, i, j))$ such that $f(u_t) \leq \ell \leq j$, $x \neq t$, and $u_x u_t \notin E(G)$, and sets $\ell(u_t; G^r_{u_z}(u_p, i, j))$ to be equal to the maximum among these values. We next show that the algorithm correctly computes $\ell(u_t; G^r_{u_z}(u_p, i, j)) = \mathcal{L}(u_t; G^r_{u_z}(u_p, i, j))$ and $P(u_t; G^r_{u_z}(u_p, i, j)) = \mathcal{P}(u_t; G^r_{u_z}(u_p, i, j))$.

3(II.a) Assume first that $u_x \notin L_j$, where u_x is the right endpoint of P'. Since by the induction hypothesis the algorithm correctly computes the values $\ell(u_s; G^{r-1}_{u_z}(u_p, i, j)$ for every vertex $u_s \in G^{r-1}_{u_z}(u_p, i, j)$ such that $u_s \notin L_j$, it follows that Algorithm LP_Cocomparability computes, among other, the value $\ell(u_x; G^{r-1}_{u_z}(u_p, i, j)) + 1 = |P'| + 1$, and sets $\ell(u_t; G^r_{u_z}(u_p, i, j))$ to be equal to $|P'| + 1 = |P|$ which is equal to the length $\mathcal{L}(u_t; G^r_{u_z}(u_p, i, j))$ of a longest normal antipath P of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_t. Also, the corresponding computed antipath $P(u_t; G^r_{u_z}(u_p, i, j)) = (P(u_x; G^{r-1}_{u_z}(u_p, i, j)), u_t)$ is a longest normal antipath $\mathcal{P}(u_t; G^r_{u_z}(u_p, i, j))$ of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_t.

3(II.b) Consider now the case where for any longest normal antipath $P = (u_{x'}, \ldots, u_x, u_t) = (P', u_t)$ of $G^r_{u_z}(u_p, i, j)$ with right endpoint the vertex u_t we have $u_x \in L_j$. Then P' is a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint any vertex of L_j, i.e., $|P'| \geq |P''|$ for any normal antipath P'' of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint a vertex of L_j. Let u_x be the rightmost vertex of $L^{r-1}_j(u_z)$ for which such an antipath P' exists. Since Algorithm LP_Cocomparability computes (with the procedure `append()`) the value $w_1 + 1 = \ell(u_x; G^{r-1}_{u_z}(u_p, i, j)) + 1$ for every vertex $u_x \in L^{r-1}_j(u_z)$, and sets $\ell(u_t; G^r_{u_z}(u_p, i, j))$ to be equal to the maximum among these values, it follows that it suffices to show that there exists at least one vertex $u_x \in L^{r-1}_j(u_z)$ for which Algorithm LP_Cocomparability correctly computes the value $\ell(u_x; G^{r-1}_{u_z}(u_p, i, j))$ and sets it to be equal to $|P'| = \mathcal{L}(u_x; G^{r-1}_{u_z}(u_p, i, j))$.

3(II.b.1) Consider first the case where u_x is the last vertex of $L^{r-1}_j(u_z)$, i.e., there exists such a longest normal antipath P' of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint a vertex of L_j, for which the right endpoint u_x of P' is the last vertex of $L^{r-1}_j(u_z)$. Then by the induction hypothesis, Algorithm LP_Cocomparability correctly computes the length $\ell(u_x; G^{r-1}_{u_z}(u_p, i, j))$ of a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint the vertex u_x. Therefore, in this case the last vertex u_x of $L^{r-1}_j(u_z)$ is such a vertex for which $\ell(u_x; G^{r-1}_{u_z}(u_p, i, j)) = \mathcal{L}(u_x; G^{r-1}_{u_z}(u_p, i, j)) = |P'|$; thus, Claim 1 holds.

3(II.b.2) Consider now the case where u_x is not the last vertex of $L^{r-1}_j(u_z)$, i.e., $u_x \in L^{r-2}_j(u_z)$. Let u_q be the last vertex of $L^{r-1}_j(u_z)$. Since P' is a longest normal antipath of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint any vertex of L_j, it follows that $u_q \in V(P')$, since otherwise $\widetilde{P} = (P', u_q)$ is such a normal antipath longer than P'. Let $P' = (u_{x'}, \ldots, u_{q'}, u_q, u_{q''}, \ldots, u_x) = (P_1, u_q, P_2)$. We now prove that $u_{q'} <_\sigma u_q$. Since $u_q \in L_j$, it follows that $u_{q'} \leq_\sigma u_q$. Assume that $u_{q'} =_\sigma u_q$. Then using Lemma 2.4 and Definition 2.2, we can easily prove that $u_s =_\sigma u_q$ for every vertex $u_s \in V(P_2)$. Thus, $P'' = (P_1, u_{q''}, \ldots, u_x, u_q)$ is a normal antipath such that $V(P'') = V(P')$ with right endpoint the vertex u_q which appears after u_x in $L^{r-1}_j(u_z)$; this is a contradiction to our choice of u_x. Therefore, we have proved that $u_{q'} <_\sigma u_q$. Therefore, using the same arguments as in Claim 2 and the property that $u_{q'} <_\sigma u_q$, we can prove that $u_{q'} <_\sigma u_s$ and $u_{q'} u_s \in E(G)$, for every vertex $u_s \in V(P_2)$.

Since $u_{q'} u_q \notin E(G)$ and $u_{q'} <_\sigma u_q$, we assume that $u_{q'} \in H_\ell$, $f(u_q) \leq \ell \leq j - 1$. Let $\mathcal{H}_2$ be the subgraph of $G^{r-1}_{u_z}(u_p, i, j)$ induced by $V(\mathcal{H}_2) = V(G^{r-2}_{u_z}(u_{q'}, \ell + 1, j))$ and let $\mathcal{H}_1$ be the subgraph of $G^{r-1}_{u_z}(u_p, i, j)$ induced by $V(\mathcal{H}_1) = V(G^{r-2}_{u_z}(u_p, i, j)) \setminus V(G^{r-2}_{u_z}(u_{q'}, \ell + 1, j))$. Using the same arguments as in Claim 3 and the property that $u_{q'} <_\sigma u_q$, we can show that every vertex of P_2 belongs to $\mathcal{H}_2$, and also that every vertex of P_1 belongs to $\mathcal{H}_1$. Therefore, we have that $V(P_1) \subseteq V(\mathcal{H}_1)$, $V(P_2) \subseteq V(\mathcal{H}_2)$, and $V(\mathcal{H}_1) \cap V(\mathcal{H}_2) = \emptyset$; thus, $V(P_1) \cap V(P_2) = \emptyset$. Finally, using the same arguments as in Case 2(I) we can obtain that P_1 is a longest normal antipath of $G^{r-2}_{u_z}(u_p, i, j)$ with right endpoint the vertex $u_{q'}$, i.e., $|P_1| = \mathcal{L}(u_{q'}; G^{r-2}_{u_z}(u_p, i, j))$, and P_2 is a longest normal antipath of $G^{r-2}_{u_z}(u_{q'}, \ell+1, j)$ with right endpoint the vertex u_x, i.e., $|P_2| = \mathcal{L}(u_x; G^{r-2}_{u_z}(u_{q'}, \ell+1, j))$.

Since $u_{q'} <_\sigma u_q$, it follows that $u_{q'} \notin L_j$. Therefore, from the induction hypothesis Algorithm LP_Cocomparability correctly computes the length $\ell(u_{q'}; G^{r-2}_{u_z}(u_p, i, j)) = |P_1|$. Now it is left to show that the value $\ell(u_x; G^{r-2}_{u_z}(u_{q'}, \ell+1, j)) = |P_2|$ computed by the algorithm is equal to $\mathcal{L}(u_x; G^{r-2}_{u_z}(u_{q'}, \ell+1, j))$. Observe that now P_2 is a longest normal antipath of $G^{r-2}_{u_z}(u_{q'}, \ell+1, j)$ with right endpoint any vertex of $L^{r-2}_j(u_z)$ and, actually, u_x is the rightmost vertex of $L^{r-2}_j(u_z)$ for which such an antipath P_2 exists, otherwise we come to a contradiction to the choice of P' (note that u_q is connected with an antiedge to every vertex of P_2). If u_x is the last vertex of $L^{r-2}_j(u_z)$ then, similarly to the above, by the induction hypothesis the algorithm correctly computes the value $\ell(u_x; G^{r-2}_{u_z}(u_{q'}, \ell+1, j))$. If u_x is not the last vertex of $L^{r-2}_j(u_z)$, then we repeat the above same procedure of Case 3(II.b.2), where now by u_q we denote the last vertex of $L^{r-2}_j(u_z)$.

We repeat the above procedure until u_x is the last vertex of the ordering $L^{r'}_j(u_z)$, $1 \leq r' \leq r-2$, i.e., we repeat the above procedure $(r-2-r')$ times in total. Let $P' = (P_1, P_2, \ldots, P_{r-2-r'+1})$ be the longest normal antipath P' of $G^{r-1}_{u_z}(u_p, i, j)$ with right endpoint the vertex u_x, such that at the s^{th} iteration of the above procedure we prove that Algorithm LP_Cocomparability correctly computes the antipath P_s, $1 \leq s \leq r-2-r'$. Finally, at the $(r-2-r')^{th}$ iteration we also obtain by the induction hypothesis that the algorithm correctly computes the antipath $P_{r-2-r'+1}$ which has u_x as its right endpoint, since at that iteration u_x is the last vertex of the ordering $L^{r'}_j(u_z)$. Concluding, Algorithm LP_Cocomparability correctly computes the length $\ell(u_x; G^{r-1}_{u_z}(u_p, i, j)) = |P'|$ and, thus, the length $\ell(u_t; G^{r}_{u_z}(u_p, i, j)) = |P'| + 1 = |P|$; the algorithm also correctly computes the corresponding antipaths.

Concluding, we have proved that Claim 1 holds for the subgraph $G^{r}_{u_z}(u_p, i, j)$ of G, where $1 \leq r \leq |L_j|$; *QED of Claim 1.* ∎

Let P be a longest antipath of G such that $|P| \geq 2$. From Lemma 2.6 we may assume that P is a longest normal antipath of G and let u_y be its right endpoint. Also, P belongs to the graph $G \setminus \{u_0\}$. Since $G(u_0, 1, k) = G \setminus \{u_0\}$ and since Algorithm LP_Cocomparability computes the maximum among the lengths $\{\ell(u_y; G(u_0, 1, k)) : u_y \in V(G(u_0, 1, k))\}$ and the corresponding antipath P', from Lemma 4.3 we obtain that $|P'| = |P|$. Therefore, we obtain the following.

Theorem 4.1 *Algorithm LP_Cocomparability computes a longest path of a cocomparability graph in polynomial time.*

4.2 Time Complexity

Let G be a comparability graph on $|V(G)| = n$ vertices and $|E(G)| = m$ edges. Given a Hasse diagram of G, the time complexity of our algorithm is as follows.

Algorithm LP_Cocomparability executes the subroutine `process()` for every induced subgraph $G^{r}_{u_z}(u_p, i, j)$ of G. In particular, the procedure `process()` contains two procedures namely `bridge()` and `append()`. The execution of the procedure `bridge()` for the subgraph $G^{r}_{u_z}(u_p, i, j)$ takes $O(n^2)$ time, due to the $O(n^2)$ pairs of antineighbors u_x and u_y of the vertex u_t in the graph $G^{r}_{u_z}(u_p, i, j)$. The execution of the procedure `append()` for the subgraph $G^{r}_{u_z}(u_p, i, j)$ takes $O(n)$ time, due to the $O(n)$ antineighbors u_x of the vertex u_t in the graph $G^{r}_{u_z}(u_p, i, j)$. Therefore, the execution of the procedure `process()` for the subgraph $G^{r}_{u_z}(u_p, i, j)$ takes $O(n^2)$ time.

Additionally, `process()` is executed at most once for each subgraph $G^{r}_{u_z}(u_p, i, j)$ of G. Since $1 \leq i \leq j \leq k$, $u_p \in H_{i-1}$, $u_z \in L_j$, and $1 \leq r \leq |L_j|$, it follows that there exist $O(n^5)$ such subgraphs $G^{r}_{u_z}(u_p, i, j)$ of G. Thus, Algorithm LP_Cocomparability takes $O(n^7)$ time.

In order to compute the length of a longest antipath, we need to store one value for every vertex u_y of $G^{r}_{u_z}(u_p, i, j)$, for every induced subgraph $G^{r}_{u_z}(u_p, i, j)$ of G. Thus, since there are in total $O(n^5)$ such subgraphs $G^r_{u_z}(u_p, i, j)$, and since each one has at most $O(n)$ vertices, we can compute the length of a longest antipath in $O(n^6)$ space. Furthermore, in order to compute and report a longest antipath, instead of its length only, we have to store an antipath of at most n vertices for each one of the $O(n^6)$ computed values. Therefore, the space complexity of Algorithm LP_Cocomparability is $O(n^7)$.

5 Concluding Remarks

In this work we presented a polynomial-time algorithm for solving the longest path problem on cocomparability graphs, resolving thus the open question on the complexity status of the problem on cocomparability and, thus, on permutation graphs. We also help to shed some light on the borderline between P and NP, since the longest path problem is known to be NP-complete on comparability graphs and quasi-parity graphs, while it polynomial on permutation and cocomparability graphs.

It would be interesting to study the complexity of the longest path problem on distance-hereditary and bipartite distance-hereditary graphs, since they admit polynomial solutions for the Hamiltonian path problem, and also since the longest path problem has been proved to be NP-complete on chordal bipartite graphs, HHD-free graphs, and parity graphs, while it is polynomial on ptolemaic graphs and trees. Additionally, the same holds for the classes of convex and biconvex graphs, since the longest path problem has been proved to be NP-complete on chordal bipartite graphs and polynomial on bipartite permutation graphs.

References

[1] S.R. Arikati and C. Pandu Rangan, Linear algorithm for optimal path cover problem on interval graphs, *Inform. Proc. Lett.* **35** (1990) 149–153.

[2] K. Asdre and S.D. Nikolopoulos, The 1-fixed-endpoint path cover problem is polynomial on interval graphs, *Algorithmica*, **58** (2010) 679–710.

[3] A.A. Bertossi, Finding Hamiltonian circuits in proper interval graphs, *Inform. Proc. Lett.* **17** (1983) 97–101.

[4] A. Brandstädt, V.B. Le and J.P. Spinrad, *Graph Classes: A Survey*, SIAM, Philadelphia, PA, 1999.

[5] R. Bulterman, F. van der Sommen, G. Zwaan, T. Verhoeff, A. van Gasteren, and W. Feijen, On computing a longest path in a tree, *Inform. Proc. Lett.* **81** (2002) 93–96.

[6] M.S. Chang, S.L. Peng, and J.L. Liaw, Deferred-query: An efficient approach for some problems on interval graphs, *Networks* **34** (1999) 1–10.

[7] P. Damaschke, The Hamiltonian circuit problem for circle graphs is NP-complete, *Inform. Proc. Lett.* **32** (1989) 1–2.

[8] P. Damaschke, Paths in interval graphs and circular arc graphs. *Discrete Math.* **112** (1993) 49–64.

[9] P. Damaschke, J.S. Deogun, D. Kratsch, and G. Steiner, Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, *Order* **8** (1992) 383–391.

[10] J.S. Deogun and C. Riedesel, Hamiltonian cycles in permutation graphs, *J. Comb. Math. and Comb. Computing* **27** (1998) 161–200.

[11] J.S. Deogun and G. Steiner, Polynomial algorithms for hamiltonian cycle in cocomparability graphs, *SIAM J. Computing* **23** (1994) 520–552.

[12] T. Feder and R. Motwani, Finding large cycles in Hamiltonian graphs, *Discrete Appl. Math.* **158** (2010) 882–893.

[13] H.N. Gabow, Finding paths and cycles of superpolylogarithmic length, *SIAM J. Computing* **36** (2007) 1648–1671.

[14] H.N. Gabow and S. Nie, Finding long paths, cycles and circuits, *Proc. of the 19th annual International Symp. on Algorithms and Computation (ISAAC)*, LNCS **5369** (2008) 752–763.

[15] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, New York, 1979.

[16] M.R. Garey, D.S. Johnson, and R.E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, *SIAM J. Computing* **5** (1976) 704–714.

[17] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Annals of Discrete Mathematics, Vol. 57), North-Holland Publishing Co., Amsterdam, The Netherlands, 2004.

[18] M. Habib, R.H. Mörhing, and G. Steiner, Computing the bump number is easy, *Order* **5** (1988) 107–129.

[19] K. Ioannidou, G.B. Mertzios, and S.D. Nikolopoulos, The longest path problem has a polynomial solution on interval graphs, *Algorithmica*, doi: 10.1007/s00453-010-9411-3.

[20] A. Itai, C.H. Papadimitriou, and J.L. Szwarcfiter, Hamiltonian paths in grid graphs, *SIAM J. Computing* **11** (1982) 676–686.

[21] D. Karger, R. Motwani, and G.D.S. Ramkumar, On approximating the longest path in a graph, *Algorithmica* **18** (1997) 82–98.

[22] T.A. McKee and F.R. McMorris, *Topics in Intersection Graph Theory*, Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[23] H. Müller, Hamiltonian circuits in chordal bipartite graphs, *Discrete Math.* **156** (1996) 291–298.

[24] G. Narasimhan, A note on the Hamiltonian circuit problem on directed path graphs, *Inform. Proc. Lett.* **32** (1989) 167–170.

[25] Y. Takahara, S. Teramoto, and R. Uehara, Longest path problems on ptolemaic graphs, *IEICE Trans. Inf. and Syst.* **91-D** (2008) 170–177.

[26] R. Uehara, Simple geometrical intersection graphs, *Proc. of the 2nd annual Workshop on Algorithms and Computation (WALCOM'08)*, LNCS **4921** (2004) 25–33.

[27] R. Uehara and Y. Uno, Efficient algorithms for the longest path problem, *Proc. of the 15th annual International Symp. on Algorithms and Computation (ISAAC)*, LNCS **3341** (2004) 871–883.

[28] R. Uehara and G. Valiente, Linear structure of bipartite permutation graphs and the longest path problem, *Inform. Proc. Lett.* **103** (2007) 71–77.

[29] S. Vishwanathan, An approximation algorithm for finding a long path in Hamiltonian graphs, *Proc. of the 11th annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, ACM (2000) 680–685.

[30] Z. Zhang, and H. Li, Algorithms for long paths in graphs, *Theoret. Comput. Sci.* **377** (2007) 25–34.