RECOGNITION AND ORIENTATION ALGORITHMS FOR P_4-COMPARABILITY GRAPHS

S.D. Nikolopoulos - L. Palios

23-2000

Preprint no. 23-00/2000

Department of Computer Science
University of Ioannina
451 10 Ioannina, Greece

Recognition and Orientation Algorithms for P_4-comparability Graphs

Stavros D. Nikolopoulos and Leonidas Palios
Department of Computer Science, University of Ioannina
GR-45110 Ioannina, Greece
e-mail: {stavros,palios}@cs.uoi.gr

Abstract: We consider two problems pertaining to P_4-comparability graphs, namely, the problem of recognizing whether a simple undirected graph is a P_4-comparability graph and the problem of producing an acyclic P_4-transitive orientation of a P_4-comparability graph. These problems have been considered by Hoàng and Reed who described $O(n^4)$ and $O(n^5)$-time algorithms for their solution respectively, where n is the number of vertices of the given graph. Faster algorithms have recently been presented by Raschle and Simon; the time complexity of their algorithms for either problem is $O(n+m^2)$, where m is the number of edges of the graph.

In this paper, we describe different $O(n+m^2)$-time algorithms for the recognition and the acyclic P_4-transitive orientation problems on P_4-comparability graphs. Instrumental in these algorithms are structural relationships of the P_4-components of a graph, which we establish and which are interesting in their own right. Our algorithms are simple, use simple data structures, and have the advantage over those of Raschle and Simon in that they are non-recursive, require linear space and admit efficient parallelization. Additionally, we describe an algorithm which computes a maximum clique of a P_4-comparability graph in $O(n+m)$ time, assuming that an acyclic P_4-transitive orientation of the graph is given; in fact, the algorithm is applicable to all perfectly orderable graphs, a superclass of the P_4-comparability graphs.

Keywords: Perfectly orderable graphs, comparability graphs, P_4-comparability graphs, P_4-components, recognition, P_4-transitive orientation, maximum clique, coloring.

1. Introduction

Let $G = (V, E)$ be a simple non-trivial undirected graph. An *orientation* of the graph G is an antisymmetric directed graph obtained from G by assigning a direction to each edge of G. An orientation (V, F) of G is called *transitive* if it satisfies the following condition: if abc is a chordless path on 3 vertices in G, then F contains the directed edges $\overrightarrow{ab}$ and $\overleftarrow{bc}$, or $\overleftarrow{ab}$ and $\overrightarrow{bc}$, where by $\overrightarrow{uv}$ or $\overleftarrow{vu}$ we denote an edge directed from u to v [12]. An orientation of a graph G is called P_4-*transitive* if the orientation of every chordless path on 4 vertices of G is transitive; an orientation of such a path $abcd$ is transitive if and only if the path's

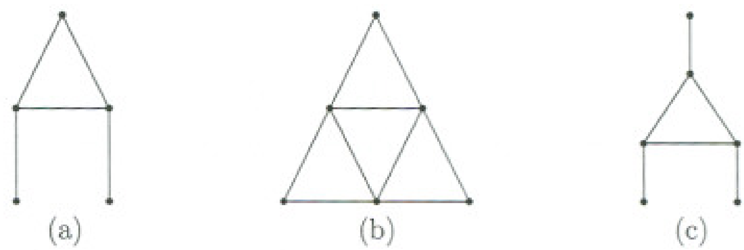

Figure 1: (a) a comparability graph; (b) a P_4-comparability graph (which is not comparability); (c) a graph which is not P_4-comparability.

edges are oriented in one of the following two ways: $\overrightarrow{ab}$, $\overleftarrow{bc}$ and $\overrightarrow{cd}$, or $\overleftarrow{ab}$, $\overrightarrow{bc}$ and $\overleftarrow{cd}$. The term borrows from the fact that a chordless path on 4 vertices is denoted by P_4.

A graph which admits an acyclic transitive orientation is called a *comparability graph* [10, 11, 12, 13]; Figure 1(a) depicts a comparability graph. A graph is a P_4-*comparability graph* if it admits an acyclic P_4-transitive orientation [16, 17]. In light of these definitions, every comparability graph is a P_4-comparability graph. Moreover, there exist P_4-comparability graphs which are not comparability; Figure 1(b) depicts such a graph, which is often referred to as a pyramid. The graph shown in Figure 1(c) is not a P_4-comparability graph.

In the early 1980s, Chvátal introduced the class of *perfectly orderable* graphs [5]; see also [16, 22, 25]. These are the graphs for which there exists a *perfect order* on the set of their vertices. An order on the vertex set of a graph G is called *perfect* if for each subgraph H of G, the greedy algorithm computes an optimal coloring of H by processing the vertices of G in that order. A *coloring* (or proper coloring) of a graph is an assignment of colors to its vertices so that no two adjacent vertices have the same color. The *greedy* algorithm, sometimes called the *first-fit* algorithm, receives the vertices of a graph G in some order $v_1 < v_2 < \ldots < v_n$ and works by assigning the smallest available color to the vertex v_i looking at the subgraph of G induced by the vertex set $\{v_1, v_2, \ldots, v_i\}$, $1 \leq i \leq n$; that is, it assigns the smallest color not yet assigned to any vertex adjacent to v_i among the previously colored vertices and does not change the assigned color afterwards.

An order on the vertex set of a graph implies an orientation on the graph's edges: if $u < v$, then the edge connecting u and v is directed from u to v, i.e., $\overrightarrow{uv}$. Chvátal proved that

(i) a graph is perfectly orderable if and only if there exists an acyclic orientation such that no P_4 $abcd$ of the graph has $\overrightarrow{ab}$ and $\overleftarrow{cd}$ (called *obstruction*), and

(ii) all perfectly orderable graphs are perfect; a graph G is said to be perfect if for each induced subgraph H of G, the chromatic number of H equals the clique number of H.

The class of perfectly orderable graphs is very important since a number of problems which are NP-complete in general can be solved in polynomial time on its members [1, 11, 12, 15]; unfortunately, it is NP-complete to decide whether a graph admits a perfect order or, equivalently, an acyclic obstruction-free orientation [22]. Chvátal showed that the class of perfectly orderable graphs contains the comparability and the triangulated graphs [5]; thus, it also contains important subclasses of these graphs, such as the bipartite, permutation, interval, split, P_4-reducible, cographs, quasi-threshold and threshold graphs [3, 6, 9, 12, 30].

Hoàng and Reed introduced the classes of the P_4-comparability, the P_4-indifference, the P_4-simplicial and the Raspail graphs, and proved that they are all perfectly orderable [17]; the fact that the P_4-comparability graphs are perfectly orderable easily follows from property (i) above, as the P_4-comparability graphs admit acyclic orientations that do not contain obstructions. Moreover, the class of perfectly orderable graphs also includes a number of other classes of graphs which are characterized by important algorithmic and structural properties; we mention the classes of 2-threshold, brittle, co-chordal, weak bipolarizable, distance hereditary, Meyniel $\cap$ co-Meyniel, P_4-sparse [12]. Finally, since every perfectly orderable graph is strongly perfect [5], the class of perfectly orderable graphs is a subclass of the well-known class of perfect graphs.

Algorithms for many different problems on all the above mentioned subclasses of perfectly orderable graphs are available in the literature; for example, recognition algorithms [7, 14, 18, 19, 27], coloring algorithms [11, 15, 24], algorithms for finding vertex and edge sets with specific properties (such as, maximum cliques, maximum weighted cliques, maximum independent sets, P_4-chains, and hamiltonian paths and cycles) [2, 4,14, 23, 26], algorithms for testing graph isomorphism [12], etc. The comparability graphs in particular have been the focus of much research which culminated into efficient recognition and orientation algorithms [12, 20, 21, 29]. On the other hand, the P_4-comparability graphs have not received as much attention, despite the fact that the definitions of the comparability and the P_4-comparability graphs rely on the same principles [8, 16, 17, 28].

Our main objective is to study the recognition and acyclic P_4-transitive orientation problems on the class of P_4-comparability graphs. These problems have been addressed by Hoàng and Reed who described polynomial time algorithms for their solution [16, 17]. The algorithms are based on detecting whether the input graph G contains a "homogeneous set" or a "good partition" and recursively solve the same problem on the graph that results from the input graph after contraction of one or two vertex sets into a single vertex each. The recognition and the orientation algorithms require $O(n^4)$ and $O(n^5)$ time respectively, where n is the number of vertices of G. Recently, newer results on these problems were provided by Raschle and Simon [28]. Their algorithms work along the same lines, but they focus on the P_4-components of the graph. In particular, for a non-trivial P_4-component $\mathcal{C}$ of the input graph G, they compute the set R of vertices adjacent to some but not all the vertices of $\mathcal{C}$; depending on whether R is empty or not, they contract $\mathcal{C}$ into one or two (non-adjacent) vertices and they recursively solve the problem on the resulting graph. The time complexity of their algorithms for either problem is $O(n+m^2)$, where m is the number of edges of G, as it is dominated by the time to compute the P_4-components of G. Raschle and Simon also described recognition and orientation algorithms for P_4-indifference graphs [28]; their algorithms run within the same time complexity, i.e., $O(n+m^2)$. We note that Hoàng and Reed [16, 17] also presented algorithms which solve the recognition problem for P_4-indifference graphs in $O(n^6)$ time.

In this paper, we present different $O(n+m^2)$-time recognition and acyclic P_4-transitive orientation algorithms for P_4-comparability graphs of n vertices and m edges. Our technique relies on the computation of the P_4-components of the input graph and takes advantage of structural relationships of these components. Note that our algorithms employ neither contraction nor recursion. Our algorithms are simple, use simple data structures, and have the advantage over those of Raschle and Simon in that they are non-recursive, require linear space and admit efficient parallelization.

We are also interested in the coloring and the maximum clique problems on the P_4-comparability graphs. According to the definition, for a perfectly orderable graph G there exists a perfect order on its vertex set $V(G)$; if a perfect order is given then the greedy algorithm produces an optimal coloring of G in linear time. Chvátal [5] proved the following result: Let U be a set of pairwise adjacent vertices of a graph G such that each $w \in U$ has a neighbor $p(w) \notin U$ and the vertices $p(w)$ are pairwise non-adjacent; if there exists a perfect order $<$ such that $p(w) < w$ for all $w \in U$ then some $p(w)$ is adjacent to all the vertices in U. Based on this, he also observed that, for a perfectly orderable graph with chromatic number k, if H is a clique consisting of vertices with colors $c, c+1, \ldots, k$ then there exists a vertex with color $c-1$ which is adjacent to all the vertices of H. This observation directly leads into an algorithm, which, given a graph G and a perfect order on $V(G)$, finds a maximum clique of G. As mentioned in [17], it is easy to see that this algorithm can be made to run in $O(n^2)$ time. Here, we show how Chvátal's observation can be used to yield an $O(n+m)$-time algorithm for the maximum clique problem on a perfectly orderable graph G if a perfect order on the vertices of G is given.

The paper is structured as follows. In Section 2 we review the terminology that we will be using throughout the paper and we establish the theoretical framework on which our algorithms are based. We describe and analyze the recognition and orientation algorithms in Sections 3 and 4 respectively. In Section 5 we present the maximum clique algorithm. We conclude with Section 6 which summarizes our results and addresses extensions and open problems.

2. Theoretical Framework

Let $G = (V, E)$ be a simple non-trivial connected graph on n vertices and m edges. A *path* in G is a sequence of vertices $(v_0, v_1, \ldots, v_k)$ such that $v_{i-1}v_i \in E$ for $i = 1, 2, \ldots, k$; we say that this is a path from v_0 to v_k and that its *length* is k. A path is undirected or directed depending on whether G is an undirected or a directed graph. A path is called *simple* if none of its vertices occurs more than once; it is called *trivial* if its length is equal to 0. A path (simple path) $(v_0, v_1, \ldots, v_k)$ is called a *cycle* (*simple cycle*) of length $k+1$ if $v_0v_k \in E$. A simple path (cycle) $(v_0, v_1, \ldots, v_k)$ is *chordless* if $v_iv_j \notin E$ for any two non-consecutive vertices v_i, v_j in the path (cycle). Throughout the paper, the chordless path (chordless cycle, respectively) on n vertices is denoted by P_n (C_n, respectively). In particular, a chordless path on 4 vertices is denoted by P_4.

Let $abcd$ be a P_4 of a graph G. The vertices b and c are called *midpoints* and the vertices a and d *endpoints* of the P_4 $abcd$. The edge connecting the midpoints of a P_4 is called the *rib*; the other two edges (which are incident to the endpoints) are called the *wings*. For example, the edge bc is the rib and the edges ab and cd are the wings of the P_4 $abcd$. Two P_4s are called *adjacent* if they have an edge in common. The transitive closure of the adjacency relation is an equivalence relation on the set of P_4s of a graph G; the subgraphs of G spanned by the edges of the P_4s in the equivalence classes are the P_4-*components* of G. With slight abuse of terminology, we consider that an edge which does not belong to any P_4 belongs to a P_4-component by itself; such a component is called *trivial*. A P_4-component which is not trivial is called *non-trivial*; clearly a non-trivial P_4-component contains at least one P_4. If the set of midpoints and the set of endpoints of the P_4s of a non-trivial P_4-component $\mathcal{C}$ define a partition of the vertex set $V(\mathcal{C})$, then the P_4-component $\mathcal{C}$ is called *separable*. One can show that:

Lemma 2.1. *Let $G = (V, E)$ be a graph and let $\mathcal{C}$ be a non-trivial P_4-component of G. Then,*

(i) If ρ and ρ' are two P_4s which both belong to $\mathcal{C}$, then there exists a sequence $\rho, \rho_1, \ldots, \rho_k, \rho'$ of adjacent P_4s in $\mathcal{C}$;

(ii) $\mathcal{C}$ is connected;

(iii) If $\mathcal{C}$ is separable and if V_1 and V_2 are the sets of the midpoints and of the endpoints of the P_4s in $\mathcal{C}$, then for every vertex $v \in V_1$ there exists a vertex $v' \in V_2$ such that $vv' \notin E$, and for every vertex $u \in V_2$ there exists a vertex $u' \in V_1$ such that $uu' \notin E$.

Proof: (i) True, because the P_4-components of G are defined in terms of the equivalence classes of the transitive closure of the adjacency relation on the P_4s of G. (ii) Follows directly from (i). (iii) Since $v \in V_1$, there exists a P_4 ρ in $\mathcal{C}$ with v as one of its midpoints. Then, v' is the endpoint of ρ which is not adjacent to v. Similarly, if $u \in V_2$, there exists a P_4 ρ' in $\mathcal{C}$ with u as one of its endpoints; then, u' is the midpoint of ρ' which is not adjacent to u. ∎

The definition of a P_4-comparability graph requires that such a graph admit an acyclic P_4-transitive orientation. However, Hoàng and Reed [17] showed that in order to determine whether a graph is a P_4-comparability graph one can restrict one's attention to the P_4-components of the graph. In particular, what they proved ([17], Theorem 3.1) can be paraphrased in terms of the P_4-components as follows:

Lemma 2.2. **([17])** *Let G be a graph such that each of its P_4-components admits an acyclic P_4-transitive orientation. Then G is a P_4-comparability graph.*

Although determining that each of the P_4-components of a graph admits an acyclic P_4-transitive orientation suffices to establish that the graph is P_4-comparability, the directed graph produced by placing the oriented P_4-components together may contain cycles. However, an acyclic P_4-transitive orientation of the entire graph can be obtained by inversion of the orientation of some of the P_4-components. Therefore, if one wishes to compute an acyclic P_4-transitive orientation of a P_4-comparability graph, one needs to detect directed cycles (if they exist) formed by edges belonging to more than one P_4-component and appropriately invert the orientation of one or more of these P_4-components. Fortunately, one does not need to consider arbitrarily long cycles as shown in the following lemma [17].

Lemma 2.3. ([17], Lemma 3.5) *If a proper orientation of an interesting graph is cyclic, then it contains a directed triangle.*[1]

Given a non-trivial P_4-component $\mathcal{C}$ of a graph $G = (V, E)$, the set of vertices $V - V(\mathcal{C})$ can be partitioned into three sets:

(i) R contains the vertices of $V - V(\mathcal{C})$ which are adjacent to some (but not all) of the vertices in $V(\mathcal{C})$,

---

[1] An orientation is proper if the orientation of every P_4 is transitive. A graph is interesting if the orientation of every P_4-component is acyclic.

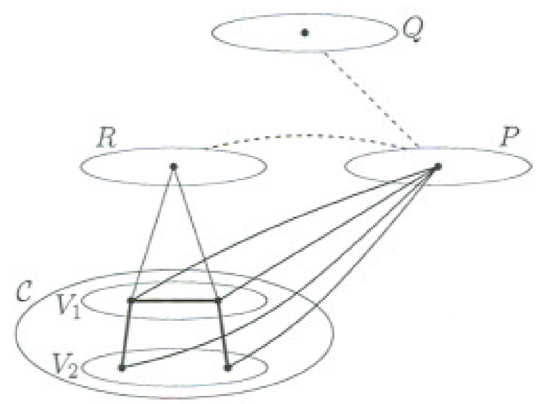

Figure 2: Partition of the vertex set with respect to a separable P_4-component $\mathcal{C}$.

(ii) P contains the vertices of $V - V(\mathcal{C})$ which are adjacent to all the vertices in $V(\mathcal{C})$, and

(iii) Q contains the vertices of $V - V(\mathcal{C})$ which are not adjacent to any of the vertices in $V(\mathcal{C})$.

The adjacency relation is considered in terms of the given graph G.

In [28], Raschle and Simon showed that, given a non-trivial P_4-component $\mathcal{C}$ and a vertex $v \notin V(\mathcal{C})$, if v is adjacent to the midpoints of a P_4 of $\mathcal{C}$ and is not adjacent to its endpoints, then v does so with respect to every P_4 in $\mathcal{C}$ (that is, v is adjacent to the midpoints and not adjacent to the endpoints of every P_4 in $\mathcal{C}$). This implies that any vertex of G, which does not belong to $\mathcal{C}$ and is adjacent to at least one but not all the vertices in $V(\mathcal{C})$, is adjacent to the midpoints of all the P_4s in $\mathcal{C}$. Based on that, Raschle and Simon showed that:

Lemma 2.4. **([28], Corollary 3.3)** *Let $\mathcal{C}$ be a non-trivial P_4-component and $R \neq \emptyset$. Then, $\mathcal{C}$ is separable and every vertex in R is V_1-universal and V_2-null*[2]*. Moreover, no edge between R and Q exists.*

The set V_1 is the set of the midpoints of all the P_4s in $\mathcal{C}$, whereas the set V_2 is the set of endpoints. Figure 2 shows the partition of the vertices of a graph with respect to a separable P_4-component $\mathcal{C}$; the dashed segments between R and P and P and Q indicate that there may be edges between pairs of vertices in the corresponding sets. Then, a P_4 with at least one but not all its vertices in $V(\mathcal{C})$ must be a P_4 of one of the following types:

type (1)	vpq_1q_2	where $v \in V(\mathcal{C}),\ p \in P,\ q_1, q_2 \in Q$
type (2)	p_1vp_2q	where $p_1 \in P,\ v \in V(\mathcal{C}),\ p_2 \in P,\ q \in Q$
type (3)	$p_1v_2p_2r$	where $p_1 \in P,\ v_2 \in V_2,\ p_2 \in P,\ r \in R$
type (4)	$v_2pr_1r_2$	where $v_2 \in V_2,\ p \in P,\ r_1, r_2 \in R$
type (5)	rv_1pq	where $r \in R,\ v_1 \in V_1,\ p \in P,\ q \in Q$

[2] For a set A of vertices, we say that a vertex v is A-universal if v is adjacent to every element of A; a vertex v is A-null if v is adjacent to no element of A.

type (6)	rv_1pv_2	where $r \in R,\ v_1 \in V_1,\ p \in P,\ v_2 \in V_2$
type (7)	$rv_1v_2v_2'$	where $r \in R,\ v_1 \in V_1,\ v_2, v_2' \in V_2$
type (8)	$v_1'rv_1v_2$	where $r \in R,\ v_1, v_1' \in V_1,\ v_2 \in V_2$

Raschle and Simon proved that neither a P_3 abc with $a \in V_1$ and $b, c \in V_2$ nor a $\overline{P_3}$ abc with $a, b \in V_1$ and $c \in V_2$ exists ([28], Lemma 3.4), which implies that:

Lemma 2.5. *Let $\mathcal{C}$ be a non-trivial P_4-component of a graph $G = (V, E)$. Then, no P_4s of type (7) or (8) with respect to $\mathcal{C}$ exist.*

Additionally, Raschle and Simon proved the following interesting result regarding the P_4-components.

Lemma 2.6. ([28], Theorem 3.6) *Two different P_4-components have different vertex sets.*

Moreover, we can show the following:

Lemma 2.7. *Let $\mathcal{A}$ and $\mathcal{B}$ be two non-trivial P_4-components of the graph G. If the component $\mathcal{A}$ contains an edge e both endpoints of which belong to the vertex set $V(\mathcal{B})$ of $\mathcal{B}$, then $V(\mathcal{A}) \subseteq V(\mathcal{B})$.*

Proof: Suppose for contradiction that there exists a vertex v of $\mathcal{A}$ which does not belong to $V(\mathcal{B})$. Let us consider the P_4 ρ of $\mathcal{A}$ which contains the edge e. If ρ has a vertex which does not belong to $V(\mathcal{B})$, then it has at least one but not all its vertices in $V(\mathcal{B})$, and it thus is a P_4 of type (1)-(6) with respect to $\mathcal{B}$ (according to Lemma 2.5, no P_4s of type (7) or (8) exist); this is impossible, however, since no P_4 of type (1)-(6) with respect to $\mathcal{B}$ has an edge both endpoints of which belong to $V(\mathcal{B})$. Therefore, all the vertices of ρ belong to $V(\mathcal{B})$. Next, we consider P_4s adjacent to ρ, and, for as long as these P_4s have all their vertices in $V(\mathcal{B})$, we keep considering adjacent P_4s. Since $\mathcal{A}$ contains the vertex v which does not belong to $V(\mathcal{B})$, eventually we will find a P_4 of $\mathcal{A}$ with a vertex not in $V(\mathcal{B})$. Let us consider the first such P_4 that we find. By definition, this P_4 has a vertex not in $V(\mathcal{B})$; moreover, since it is the first such P_4, it is adjacent to a P_4 all of whose vertices belong to $V(\mathcal{B})$, and it thus contains an edge both endpoints of which belong to $V(\mathcal{B})$. This, however, leads to a contradiction, since this P_4 should be of type (1)-(6) with respect to $\mathcal{B}$, and yet no such P_4 has an edge both endpoints of which belong to $V(\mathcal{B})$. ∎

Let us consider a non-trivial P_4-component $\mathcal{C}$ of the graph G such that $V(\mathcal{C}) \subset V$, and let $S_\mathcal{C}$ be the set of non-trivial P_4-components of G which have a vertex in $V(\mathcal{C})$ and a vertex in $V - V(\mathcal{C})$. Then, each component in $S_\mathcal{C}$ contains a P_4 of type (1)-(8), and thus, by taking Lemma 2.5 into account, we can partition the elements of $S_\mathcal{C}$ into two sets as follows:

- *P_4-components of type A:* the P_4 components, each of which contains at least one P_4 of type (1)-(5) with respect to $\mathcal{C}$;
- *P_4-components of type B:* the P_4-components which contain only P_4s of type (6) with respect to $\mathcal{C}$.

The following lemmata establish properties of P_4-components of type A and of type B.

Lemma 2.8. *Let $\mathcal{C}$ be a non-trivial P_4-component of a P_4-comparability graph $G = (V, E)$ and suppose that the vertices in $V - V(\mathcal{C})$ have been partitioned into sets R, P, and Q as described earlier in this section. Then, if there exists an edge xv (where $x \in R \cup P$ and $v \in V(\mathcal{C})$) that belongs to a P_4-component $\mathcal{A}$ of type A, then all the edges, which connect the vertex x to a vertex in $V(\mathcal{C})$, belong to $\mathcal{A}$. Moreover, these edges are all oriented towards x or they are all oriented away from x.*

Proof: Let xu be an edge of G connecting the vertex x to the vertex u in $V(\mathcal{C})$. Since $u \in V(\mathcal{C})$, there exists a vertex w in $V(\mathcal{C})$ such that u and w are not adjacent and they do not both belong to V_1 or V_2 (Lemma 2.1, statement (iii)). We show below that xu belongs to $\mathcal{A}$ and has the same orientation as xv.

(a) $x \in R$: Then, $u \in V_1$, $w \in V_2$ and the edge xv participates in a P_4 of type (5) or (6). If it participates in a P_4 of type (5), say, in $xvpq$ ($p \in P$, $q \in Q$), then the path $xupq$ is also a P_4 and therefore the edge xu belongs to $\mathcal{A}$ as well and has the same orientation as xv. Suppose now that xv participates in a P_4 of type (6), say, in $xvpv'$, where $p \in P$ and $v' \in V_2$ (Figure 3). Then, since xv belongs to the P_4-component $\mathcal{A}$, which is of type A and therefore contains a P_4 of type (1)-(5), there exists a sequence S of adjacent P_4s from the P_4 $xvpv'$ to a P_4 of type (1)-(5) (Lemma 2.1, statement (i)). Without loss of generality, we may assume that all the P_4s in the sequence S except for the last one are P_4s of type (6); otherwise, we consider the prefix of the sequence up to the first P_4 of type (1)-(5). Let the sequence S be

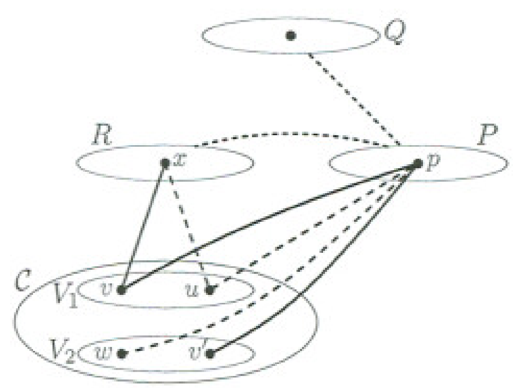

Figure 3

$$xvpv' = r_1v_1p_1v'_1,\ r_2v_2p_2v'_2,\ \ldots,\ r_kv_kp_kv'_k,\ \rho,$$

where $r_i \in R$, $v_i \in V_1$, $p_i \in P$, $v'_i \in V_2$, and ρ is a P_4 of type (1)-(5) adjacent to $r_kv_kp_kv'_k$. Clearly, all these P_4s belong to the component $\mathcal{A}$. Because each P_4 $r_iv_ip_iv'_i$ has one vertex from each one of four disjoint sets, the P_4s $r_iv_ip_iv'_i$ and $r_{i+1}v_{i+1}p_{i+1}v'_{i+1}$, which are adjacent, share an edge which is either a rib or a wing to both of them. So, the adjacency of $r_iv_ip_iv'_i$ and $r_{i+1}v_{i+1}p_{i+1}v'_{i+1}$ implies that $r_i = r_{i+1}$ and $v_i = v_{i+1}$, or $v_i = v_{i+1}$ and $p_i = p_{i+1}$, or $p_i = p_{i+1}$ and $v'_i = v'_{i+1}$. Let us now consider the sequence S' of paths

$$xupw = r_1up_1w,\ r_2up_2w,\ \ldots,\ r_kup_kw.$$

It is not difficult to see that each of these paths is a P_4: $r_iu \in E$, $p_iu \in E$, $p_iw \in E$, $uw \notin E$, $r_iw \notin E$, and, from the sequence S, $r_ip_i \notin E$. Moreover, any two consecutive paths in S' are adjacent; note that the adjacency of $r_iv_ip_iv'_i$ and $r_{i+1}v_{i+1}p_{i+1}v'_{i+1}$ in

S implies that $r_i = r_{i+1}$ or $p_i = p_{i+1}$ or both, which in turn implies that the P_4s $r_i u p_i w$ and $r_{i+1} u p_{i+1} w$ are adjacent. Finally, the fact that every element of P is $(V_1 \cup V_2)$-universal and that every element of R is V_1-universal and V_2-null implies that the path ρ', which results from ρ if we replace v_k by u and v'_k by w, is a P_4 as well. Moreover, ρ' is adjacent to $r_k u p_k w$ (since ρ is adjacent to $r_k v_k p_k v'_k$), and ρ and ρ' are P_4s of the same type and thus they have three vertices in common, as it follows from the general form of the P_4s of type (1)-(5). Therefore, ρ and ρ' are adjacent, they belong to the same P_4-component $\mathcal{A}$ and they have corresponding orientations; then, the edges $r_k v_k$ and $r_k u$ of their adjacent P_4s $r_k v_k p_k v'_k$ and $r_k u p_k w$ are oriented either both towards r_k or both away from it. In turn, the sequences S and S' of P_4s imply that the edges xv and xu belong to the same P_4-component and they are oriented either both towards x or both away from it, as desired.

(b) $x \in P$ and v, u both belong to V_1 or both belong to V_2: Then, xv participates in a P_4 of type (1)-(6). If it participates in a P_4, say, ρ, of type (1)-(5), then the path which results from ρ after replacing v by u is a P_4, is of the same type as ρ, and is adjacent to ρ. Therefore, the edges xv and xu belong to the same component $\mathcal{A}$ and have the same orientation. Suppose now that xv participates in a P_4 of type (6). We consider first the case where $v, u \in V_1$. Since $v \in V_1$, there exists a vertex v' such that $v' \in V_2$ and $vv' \notin E$ (Lemma 2.1, statement (iii)). Then, the path $rvxv'$ is a P_4 and belongs to $\mathcal{A}$ (edge xv). Case (a) applies for the edges rv and ru, implying that they belong to $\mathcal{A}$ and they are oriented either both towards their common endpoint or both away from it. Then, so do the edges xv and xu because of the P_4s $rvxv'$ and $ruxw$. We work similarly in the second case, where $v, u \in V_2$; this time we consider the P_4s $rv'xv$ and $rwxu$.

(c) $x \in P$, $v \in V_1$ and $u \in V_2$: Then, xv participates in a P_4, say, ρ, of type (1), (2), (5) or (6). If ρ is of type (1) or (2), then we work as in the first subcase of Case (b): replacing v by u in ρ yields a P_4, which together with ρ ensures that the edges xv and xu belong to the same P_4-component $\mathcal{A}$ and have the same orientation. If ρ is of type (5) or (6), i.e., of the form $rvxq$ or $rvxv'$ respectively ($r \in R$, $q \in Q$, $v' \in V_2$), then we consider the path $rwxu$ which is a P_4; note that $w \in V_1$ since $u \in V_2$. The lemma follows if we show that the edges rv and rw belong to the same P_4-component $\mathcal{A}$ and have the same orientation; this is established in Case (a) above.

(d) $x \in P$, $v \in V_2$ and $u \in V_1$: Then, xv participates in a P_4, say, ρ, of type (1)-(4) or (6). If ρ is of type (1) or (2), then we work as in the first subcase of Case (b): we replace v by u in ρ and we get a P_4 ρ' adjacent to ρ; then, the edge xw belongs to ρ' and has the same orientation as xv. Suppose now that ρ is of type (3), (4), or (6), i.e., $xvpr$ or $pvxr$, $vxrr'$, $rv'xv$ respectively ($r, r' \in R$, $p \in P$, $v' \in V_2$). In each of these cases, the path $ruxw$ is a P_4; note that $w \in V_2$ since $u \in V_1$. Therefore, the edges xu and xw are oriented either both towards x or both away from it. The lemma follows by noting that Case (b) implies that the edges xv and xw belong to $\mathcal{A}$ and are oriented either both towards x or both away from it. ∎

Lemma 2.9. *Let $\mathcal{B}$ and $\mathcal{C}$ be two non-trivial P_4-components of the graph G such that $\mathcal{B}$ is of type B with respect to $\mathcal{C}$. Then,* *(i) both $\mathcal{B}$ and $\mathcal{C}$ are separable;*

(ii) every edge of $\mathcal{B}$ has exactly one endpoint in $V(\mathcal{C})$;

(iii) if an edge of $\mathcal{B}$ is oriented towards its endpoint that belongs to $V(\mathcal{C})$, then so do all the edges of $\mathcal{B}$;

(iv) the edges of $\mathcal{B}$ incident upon the same vertex v are all oriented either towards v or away from it.

Proof: (i) Since $\mathcal{B}$ is of type B with respect to $\mathcal{C}$, then $R \neq \emptyset$; thus, $\mathcal{C}$ is separable in accordance with Lemma 2.4. Additionally, since all the P_4s of $\mathcal{B}$ are of type (6) with respect to $\mathcal{C}$, then the midpoints of all these P_4s are either midpoints of $\mathcal{C}$ or belong to P, whereas the endpoints are either endpoints of $\mathcal{C}$ or belong to R; thus, $\mathcal{B}$ is separable as well.

(ii) Clearly true, because of the general form of the P_4s of type (6).

(iii) Let e be the edge of $\mathcal{B}$ which is oriented towards its endpoint that belongs to $V(\mathcal{C})$. Clearly, all the edges of the P_4 to which e belongs are oriented towards their endpoint which belongs to $V(\mathcal{C})$ as well. The truth of the statement follows from the fact that a P_4 of type (6) has one vertex from each one of four disjoint sets and therefore two adjacent P_4s share an edge that is a rib or a wing to both of them.

(iv) Follows easily from statement (iii): if $v \in V(\mathcal{C})$, then all the edges of $\mathcal{B}$ incident upon v are oriented towards v; otherwise, they are oriented away from v. ∎

Lemma 2.10. *Let $\mathcal{B}$ and $\mathcal{C}$ be two non-trivial P_4-components of the graph G such that $|V(\mathcal{B})| \geq |V(\mathcal{C})|$ and let $\beta = \sum_{v \in V(\mathcal{C})} d_{\mathcal{B}}(v)$, where $d_{\mathcal{B}}(v)$ denotes the number of edges of $\mathcal{B}$ which are incident upon v. Then, $\mathcal{B}$ is of type B with respect to $\mathcal{C}$ if and only if $\beta = |E(\mathcal{B})|$.*

Proof: Clearly, if $\mathcal{B}$ is of type B with respect to $\mathcal{C}$, then $\beta = |E(\mathcal{B})|$; note that each edge of a P_4 of type (6) with respect to $\mathcal{C}$ has exactly one of its endpoints in $V(\mathcal{C})$ (Lemma 2.9, statement(ii)). Suppose now that $\beta = |E(\mathcal{B})|$; we will show that $\mathcal{B}$ is of type B with respect to $\mathcal{C}$. Since $\beta = |E(\mathcal{B})|$, the P_4-component B contains at least one vertex not in $V(\mathcal{C})$; otherwise, $V(\mathcal{B}) = V(\mathcal{C})$ and β would be equal to $2\,|E(\mathcal{B})|$. Then, $\mathcal{B}$ may contain P_4s of type (1)-(6) (recall that Lemma 2.5 excludes P_4s of type (7) and (8)) and P_4s none of whose vertices is a vertex in $V(\mathcal{C})$. The edges of the latter set of P_4s contribute nothing to the quantity β. On the other hand, the general form of the P_4s of type (1)-(6) indicates that the edges of such P_4s have at most one of their endpoints in $V(\mathcal{C})$, and thus contribute at most 1 to β each. Therefore, each edge of $\mathcal{B}$ contributes at most 1 to β. In order that $\beta = |E(\mathcal{B})|$, it is required that each edge contributes exactly 1. This is possible only if the edges participate in P_4s of type (6) with respect to $\mathcal{C}$; note that each P_4 of type (1)-(5) with respect to $\mathcal{C}$ contains at least one edge which is not incident upon any vertex of $\mathcal{C}$. Therefore, $\mathcal{B}$ has to be of type B with respect to $\mathcal{C}$. ∎

Lemma 2.11. *Let $\mathcal{C}$ be a non-trivial P_4-component of a P_4-comparability graph $G = (V, E)$ and let the edge uv be a rib of a P_4 in $\mathcal{C}$. Moreover, suppose that the vertices in $V - V(\mathcal{C})$ have been partitioned into sets R, P, and Q as described earlier in this section, and let $r \in R$. If the edges ru and rv belong to the non-trivial P_4-components $\mathcal{A}$ and $\mathcal{B}$ respectively, such that $\mathcal{A} \neq \mathcal{B}$ and both $\mathcal{A}$ and $\mathcal{B}$ are of type B with respect to $\mathcal{C}$, then:*

(i) *For every edge yz, which is the rib of a P_4 of $\mathcal{C}$, either $ry \in \mathcal{A}$ and $rz \in \mathcal{B}$, or $ry \in \mathcal{B}$ and $rz \in \mathcal{A}$.*

(ii) *The set $V_1(\mathcal{C})$ of midpoints of the P_4s in $\mathcal{C}$ can be partitioned into sets $M_{\mathcal{A}}$ and $M_{\mathcal{B}}$ such that $M_{\mathcal{A}}$ ($M_{\mathcal{B}}$ respectively) is a nonempty subset of the set of midpoints of the P_4s in $\mathcal{A}$ ($\mathcal{B}$ respectively). Similarly, the set $V_2(\mathcal{C})$ of endpoints of the P_4s in $\mathcal{C}$ can be partitioned into sets $N_{\mathcal{A}}$ and $N_{\mathcal{B}}$ such that $N_{\mathcal{A}}$ ($N_{\mathcal{B}}$ respectively) is a nonempty subset of the set of endpoints of the P_4s in $\mathcal{A}$ ($\mathcal{B}$ respectively).*

(iii) *Let $abcd$ be a P_4 of $\mathcal{C}$. If $b \in M_{\mathcal{A}}$, then $d \in N_{\mathcal{A}}$ and $a, c \notin V(\mathcal{A})$; if $a \in N_{\mathcal{A}}$, then $c \in M_{\mathcal{A}}$ and $b, d \notin V(\mathcal{A})$. Similarly, for $\mathcal{B}$.*

(iv) *$\mathcal{C}$ is of type B with respect to $\mathcal{A}$ and with respect to $\mathcal{B}$.*

(v) *$\mathcal{A}$ is of type B with respect to $\mathcal{B}$ and vice versa.*

Proof: (Note that since the P_4-components $\mathcal{A}$ and $\mathcal{B}$ are of type B with respect to $\mathcal{C}$, Lemma 2.9 (statement (i)) implies that all three P_4-components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are separable, and therefore their sets of midpoints and endpoints are well defined.) Below, the sets $V_1(\mathcal{K})$ and $V_2(\mathcal{K})$ pertain to the partition of the vertices of a separable P_4-component $\mathcal{K}$ into a set of midpoints and a set of endpoints of the P_4s of $\mathcal{K}$, and the sets $R(\mathcal{K})$ and $P(\mathcal{K})$ to the partition of the vertices of $V - V(\mathcal{K})$. The edge uv is the rib of a P_4 of $\mathcal{C}$; let that P_4 be $suvt$, where $s, t \in V_2(\mathcal{C})$. Furthermore, since the edge ru belongs to the P_4-component $\mathcal{A}$ and $\mathcal{A}$ is of type B with respect to $\mathcal{C}$, then ru belongs to a P_4 of type (6) with respect to $\mathcal{C}$; let that P_4 be $rupq$, where $p \in P(\mathcal{C})$ and $q \in V_2(\mathcal{C})$. Then, $rupt$ is also a P_4 and belongs to $\mathcal{A}$.

(i) Clearly, the proposition holds for the rib uv. We will show that if it holds for the rib bc of a P_4 $abcd$ of $\mathcal{C}$, then it also holds for the rib of any P_4 $a'b'c'd'$ adjacent to $abcd$. Because $\mathcal{C}$ is separable, the two P_4s $abcd$ and $a'b'c'd'$ share an edge which is a rib or a wing to both of them; hence, without loss of generality, $a = a'$ and $b = b'$, or $b = b'$ and $c = c'$, or $c = c'$ and $d = d'$. We will show that the edge rb' belongs to the same P_4-component as the edge rb, and that the edge rc' belongs to the same P_4-component as the edge rc. We distinguish the following cases:

- ▷ $a' = a$ and $b' = b$: Trivially, rb' and rb belong to the same P_4-component. We consider the paths $rcpa$ and $rc'pa'$; these are P_4s and because $a' = a$ they share the edge pa. Therefore, the edges rc and rc' belong to the same component.
- ▷ $b' = b$ and $c' = c$: Trivially true.
- ▷ $c' = c$ and $d' = d$: Similar to the case where $a' = a$ and $b' = b$.

Since for every P_4 ρ of $\mathcal{C}$, there exists a sequence of adjacent P_4s from the P_4 with rib uv to ρ (Lemma 2.1, statement (i)), the lemma follows.

(ii) The proposition for the midpoints of the P_4s of $\mathcal{C}$ follows directly from statement (i) given that $\mathcal{A} \neq \mathcal{B}$. In order to prove the proposition for the endpoints of the P_4s of $\mathcal{C}$, we consider a P_4 $abcd$ of $\mathcal{C}$. The paths $rbpd$ and $rcpa$ are P_4s, and thus the endpoints a and d of the P_4 belong to the components containing the edges rc and rb respectively. Therefore, in light of statement (i), every endpoint of $\mathcal{C}$ belongs to either $\mathcal{A}$ or $\mathcal{B}$. No endpoint may belong to both components, for otherwise the edge connecting p to that endpoint would belong to both $\mathcal{A}$ and $\mathcal{B}$, in contradiction to the fact that $\mathcal{A} \neq \mathcal{B}$.

(iii) Recall that the edge ru belongs to a P_4 $rupq$ of $\mathcal{A}$. If $b \in M_{\mathcal{A}}$, then, according to statement (i), the edge rb belongs to the P_4-component $\mathcal{A}$. Then, the path $rbpd$, which is a P_4, belongs to $\mathcal{A}$, and thus $d \in N_{\mathcal{A}}$. Statement (i) of the lemma also implies that $rc \in \mathcal{B}$, which in turn implies that $c \in M_{\mathcal{B}}$ because of the P_4 $rcpa$; the same P_4 implies that $a \in N_{\mathcal{B}}$. The general form of the P_4s of type (6) implies that if a midpoint (endpoint, respectively) of $\mathcal{C}$ belongs to a component which is of type B with respect to $\mathcal{C}$, then it is a midpoint (endpoint, respectively) of that component. Therefore, $c \notin V(\mathcal{A})$; otherwise, c would be a midpoint of $\mathcal{A}$ and would thus belong to $M_{\mathcal{A}}$, in contradiction to the fact that $c \in M_{\mathcal{B}}$. Similarly, $a \notin V(\mathcal{A})$; otherwise, a would belong to $N_{\mathcal{A}}$; a contradiction, since $a \in N_{\mathcal{B}}$.
A similar approach establishes that $c \in M_{\mathcal{A}}$ and $b, d \notin V(\mathcal{A})$ if $a \in N_{\mathcal{A}}$.

(iv) Let $abcd$ be a P_4 of $\mathcal{C}$. According to statements (i) and (ii), one of the midpoints b, c belongs to $M_{\mathcal{A}}$; let us suppose without loss of generality that $b \in M_{\mathcal{A}}$. Then, the general form of the P_4s of type (6) implies that b is a midpoint of $\mathcal{A}$, i.e., $b \in V_1(\mathcal{A})$. Moreover, according to statement (iii), $b \in M_{\mathcal{A}}$ implies that $d \in N_{\mathcal{A}}$ and $a, c \notin V(\mathcal{A})$; that is, $d \in V_2(\mathcal{A})$. Since $a \notin V(\mathcal{A})$ and a is adjacent to the vertex b and not adjacent to the vertex d of $\mathcal{A}$, then $a \in R(\mathcal{A})$. On the other hand, since $c \notin V(\mathcal{A})$ and c is adjacent to both the midpoint b and the endpoint d of $\mathcal{A}$, then $c \in P(\mathcal{A})$. Therefore, the P_4 $abcd$ is of type (6) with respect to the P_4-component $\mathcal{A}$. Since this holds for any P_4 of $\mathcal{C}$, the P_4-component $\mathcal{C}$ is of type B with respect to $\mathcal{A}$. Proving that $\mathcal{C}$ is of type B with respect to $\mathcal{B}$ is done in a similar way.

(v) Let $xyzw$ be a P_4 of $\mathcal{A}$ and suppose without loss of generality that y is a midpoint of $\mathcal{C}$; thus, $x \in R(\mathcal{C})$, $y \in V_1(\mathcal{C})$, $z \in P(\mathcal{C})$, and $w \in V_2(\mathcal{C})$. More specifically, $y \in M_{\mathcal{A}}$. Then, statements (i) and (ii) imply that $z \in M_{\mathcal{B}}$, which in turn implies that $y, w \notin V(\mathcal{B})$ according to statement (iii). On the other hand, since y is a midpoint of $\mathcal{C}$, there exists a P_4 of $\mathcal{C}$ with y as a midpoint; let it be $aycd$; that is, $c \in V_1(\mathcal{C})$ and $a, d \in V_2(\mathcal{C})$. Then, the path $xcza$ is a P_4 and belongs to $\mathcal{B}$, which implies that $x \in V_2(\mathcal{B})$ and $z \in V_1(\mathcal{B})$. Since $y \notin V(\mathcal{B})$, and y is adjacent to both the endpoint x and the midpoint z of $\mathcal{B}$, then $y \in P(\mathcal{B})$. Moreover, since $w \notin V(\mathcal{B})$, and w is adjacent to z and not adjacent to x, then $w \in R(\mathcal{B})$. Therefore, the P_4 $xyzw$ is of type (6) with respect to the P_4-component $\mathcal{B}$, which implies that the P_4-component $\mathcal{A}$ is of type B with respect to $\mathcal{B}$. Proving that $\mathcal{B}$ is of type B with respect to $\mathcal{A}$ is done in a similar way. ∎

Note that statement (i) of Lemma 2.11 implies that, for a P_4-component $\mathcal{C}$ meeting the conditions of the lemma, the subgraph spanned by the ribs of the P_4s in $\mathcal{C}$ is bipartite.

Lemma 2.12. *Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ be three distinct non-trivial P_4-components of a graph G such that $\mathcal{A}$ is of type B with respect to $\mathcal{B}$, $\mathcal{B}$ is of type B with respect to $\mathcal{C}$, and $|V(\mathcal{A})| \geq |V(\mathcal{C})|$. Then, if there exists a vertex which is a midpoint of all three components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, the P_4-component $\mathcal{A}$ is of type B with respect to $\mathcal{C}$.*

Proof: The conditions in the lemma and Lemma 2.9 (statement(i)) imply that all three P_4-components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are separable and therefore their sets of midpoints and endpoints are well defined. Below, for a separable P_4-component $\mathcal{K}$, the sets $V_1(\mathcal{K})$ and $V_2(\mathcal{K})$ denote the sets of midpoints and endpoints of the P_4s of $\mathcal{K}$, and the sets $R(\mathcal{K})$ and $P(\mathcal{K})$ the partition sets of the vertices of $V - V(\mathcal{K})$, as described earlier.

Let b be the vertex which is a midpoint of all three components. Since b is a midpoint of $\mathcal{A}$, there exists a P_4 of $\mathcal{A}$ with b as a midpoint; let that P_4 be $abcd$. Since b belongs to the set $V_1(\mathcal{B})$ and $\mathcal{A}$ is of type B with respect to $\mathcal{B}$, the P_4 $abcd$ is of type (6) with respect to $\mathcal{B}$; that is, $a \in R(\mathcal{B})$, $b \in V_1(\mathcal{B})$, $c \in P(\mathcal{B})$, and $d \in V_2(\mathcal{B})$, where the sets $V_1(\mathcal{K})$ and $V_2(\mathcal{K})$ pertain to the partition of the vertices of a separable P_4-component $\mathcal{K}$ into a set of midpoints and a set of endpoints of the P_4s of $\mathcal{K}$, and the sets $R(\mathcal{K})$ and $P(\mathcal{K})$ to the partition of the vertices of $V - V(\mathcal{K})$. Since $d \in V_2(\mathcal{B})$ and given that $\mathcal{B}$ is of type B with respect to $\mathcal{C}$, then either $d \in R(\mathcal{C})$ or $d \in V_2(\mathcal{C})$. The former case is not possible, since $b \in V_1(\mathcal{C})$ and b and d are not adjacent in G (recall that the path $abcd$ is a P_4). Therefore, $d \in V_2(\mathcal{C})$. On the other hand, $a \notin V(\mathcal{C})$. Otherwise, $\mathcal{A}$ would contain the edge ab, whose both endpoints would belong to $V(\mathcal{C})$; then, according to Lemma 2.7, $V(\mathcal{A}) \subseteq V(\mathcal{C})$. Since $|V(\mathcal{A})| \geq |V(\mathcal{C})|$, we have that $V(\mathcal{A}) = V(\mathcal{C})$, and then Lemma 2.6 would imply that $\mathcal{A} = \mathcal{C}$, a contradiction since the three P_4-components are distinct. Since $a \notin V(\mathcal{C})$ and given that a is adjacent to the midpoint b of $\mathcal{A}$ and not adjacent to the endpoint d, we conclude that $a \in R(\mathcal{C})$. Finally, $c \notin V(\mathcal{C})$ in a fashion similar to the one that we used for a. Since c is adjacent to both the midpoint b and the endpoint d of $\mathcal{A}$, we conclude that $c \in P(\mathcal{C})$. Therefore, the P_4 $abcd$ of $\mathcal{A}$ is of type (6) with respect to $\mathcal{C}$.

What we need to show is that all the P_4s of $\mathcal{A}$ are of type (6) with respect to $\mathcal{C}$. This will follow if we show that if ρ is a P_4 of $\mathcal{A}$ such that ρ is of type (6) with respect to $\mathcal{C}$ and one of ρ's midpoints is a midpoint of all three components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, then any P_4 adjacent to ρ also satisfies these conditions, that is, it is of type (6) with respect to $\mathcal{C}$ and it has a midpoint which is a midpoint of all three components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$.

Let us consider a P_4 $xyzw$ of $\mathcal{A}$ which is of type (6) with respect to $\mathcal{C}$ and suppose that its midpoint y is a midpoint of all three components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Let $x'y'z'w'$ be a P_4 adjacent to $xyzw$; then, $x' = x$ and $y' = y$, or $y' = y$ and $z' = z$, or $z' = z$ and $w' = w$. We consider these three cases separately:

(i) $x' = x$ and $y' = y$: Because $xyzw$ is a P_4 of type (6) with respect to $\mathcal{C}$, and y is a midpoint of $\mathcal{C}$, then $x \in R(\mathcal{C})$ and $y \in V_1(\mathcal{C})$, or equivalently, $x' \in R(\mathcal{C})$ and $y' \in V_1(\mathcal{C})$ since $x' = x$ and $y' = y$. Moreover, since the P_4-component $\mathcal{A}$ is of type B with respect to $\mathcal{B}$, the P_4 $x'y'z'w'$ of $\mathcal{A}$ is of type (6) with respect to $\mathcal{B}$; then, $w' \in V_2(\mathcal{B})$ due to the form of a P_4 of type (6) and the fact that y' (which coincides with y) is a midpoint of $\mathcal{B}$. Additionally, the fact that the P_4-component $\mathcal{B}$ is of type B with respect to $\mathcal{C}$ implies that the endpoint w' is an endpoint of a P_4 of type (6) with respect to $\mathcal{C}$ and thus belongs either to $R(\mathcal{C})$ or to $V_2(\mathcal{C})$. The former is not possible, since y' is a midpoint of $\mathcal{C}$ and w' is not adjacent to it; recall the P_4 $x'y'z'w'$. Therefore, $w' \in V_2(\mathcal{C})$. On the other hand, $z' \notin V(\mathcal{C})$. Otherwise, both endpoints of the edge $y'z'$ (which belongs to $\mathcal{A}$) would belong to $\mathcal{C}$, and then according to Lemma 2.7, $V(\mathcal{A}) \subseteq V(\mathcal{C})$; since $|V(\mathcal{A})| \geq |V(\mathcal{C})|$, we would have that $V(\mathcal{A}) = V(\mathcal{C})$, which leads to a contradiction since Lemma 2.6 would imply that $\mathcal{A} = \mathcal{C}$. Because $z' \notin V(\mathcal{C})$, and z' is adjacent to both the midpoint y' and the endpoint w' of $\mathcal{C}$, we conclude that $z' \in P(\mathcal{C})$.

(ii) $y' = y$ and $z' = z$: We work in a fashion similar to the one used in the previous case. Clearly, $y' \in V_1(\mathcal{C})$ and $z' \in P(\mathcal{C})$. As in the previous case, $w' \in V_2(\mathcal{C})$. On the other hand, $x' \notin V(\mathcal{C})$, which implies that $x' \in R(\mathcal{C})$, since x' is adjacent to the midpoint y' and not adjacent to the endpoint w' of $\mathcal{C}$.

(iii) $z' = z$ and $w' = w$: From $z' = z$ and $w' = w$, and from the fact that the P_4 $xyzw$ of $\mathcal{A}$ is of type (6) with respect to $\mathcal{C}$, where y is a midpoint of $\mathcal{C}$, we conclude that $z' \in P(\mathcal{C})$

and $w' \in V_2(\mathcal{C})$. Moreover, $\mathcal{A}$ is of type B with respect to $\mathcal{B}$; then, since the P_4 $xyzw$ belongs to $\mathcal{A}$, where y is a midpoint of $\mathcal{B}$, $w \in V_2(\mathcal{B})$. The P_4 $x'y'z'w'$ is a P_4 of $\mathcal{A}$ too, and thus is of type (6) with respect to $\mathcal{B}$; then, $y' \in V_1(\mathcal{B})$ because of the form of a P_4 of type (6) and of the fact that w' (which coincides with w) belongs to $V_2(\mathcal{B})$. In turn, because the P_4-component $\mathcal{B}$ is of type B with respect to $\mathcal{C}$, the midpoint y' is a midpoint of a P_4 of type (6) with respect to $\mathcal{C}$ and thus belongs either to $V_1(\mathcal{C})$ or to $P(\mathcal{C})$. The latter is not possible, since y' is not adjacent to the endpoint w' of $\mathcal{C}$. Therefore, $y' \in V_1(\mathcal{C})$. Finally, as in the previous case, $x' \in R(\mathcal{C})$.

In all three cases, we conclude that the P_4 $x'y'z'w'$ is of type (6) with respect to $\mathcal{C}$, and that its midpoint y' is a midpoint of all three components $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. The lemma follows from this result, the fact that the P_4 $abcd$ satisfies these conditions and that for any pair ρ and ρ' of P_4s of the same P_4-component there exists a sequence of adjacent P_4s from ρ to ρ' (Lemma 2.1, statement (i)). ∎

We close this section by showing that the assignment of compatible directions in all the P_4s of a P_4-component does not imply that the component is necessarily acyclic. We first give an example of a graph that has a P_4-component with a directed cycle of length 3, and then we generalize it to P_4-components with directed cycles of arbitrary length. Consider the graph of Figure 4(a); each vertex is adjacent to all but two other vertices so that the paths $x_0y_0y_1z_0$, $x_1y_1y_2z_1$, and $x_2y_2y_0z_2$ are all P_4s. Additionally, the paths $y_1z_0z_1x_0$ and $z_1x_0x_1y_1$ are P_4s, are adjacent since they share the edge z_1x_0 and belong to the same P_4-component as $x_0y_0y_1z_0$ and $x_1y_1y_2z_1$ because they form the following sequence of adjacent P_4s: $x_0y_0y_1z_0$, $y_1z_0z_1x_0$, $z_1x_0x_1y_1$, $x_1y_1y_2z_1$. Moreover, assuming (without loss of generality) that the edge y_0y_1 is oriented towards y_1, this sequence of P_4s implies that the edge y_1y_2 is oriented towards y_2. In a similar fashion, the P_4 $x_2y_2y_0z_2$ belongs to the same P_4-component and the edge y_2y_0 is oriented towards y_0. Thus, a directed cycle of length 3 is formed. In fact, this is not the only directed cycle of length 3 in the P_4-component; two more are formed by the directed edges in $x_0x_1x_2$ and in $z_0z_1z_2$.

The previous example can be easily generalized to yield a graph with a P_4-component exhibiting an arbitrarily long directed cycle. Let k be an integer at least equal to 3, and let $X_k = \{x_i \,|\, 0 \le i < k\}$, $Y_k = \{y_i \,|\, 0 \le i < k\}$, and $Z_k = \{z_i \,|\, 0 \le i < k\}$ be three sets of distinct vertices. We consider the graph $G_k = (V_k, E_k)$ where

$$\begin{aligned} V_k &= X_k \cup Y_k \cup Z_k \\ \text{and} \quad E_k &= V_k \times V_k - \Big(\{x_iy_{i+1} \,|\, 0 \le i < k\} \cup \{x_iz_i \,|\, 0 \le i < k\} \cup \{y_iz_i \,|\, 0 \le i < k\} \Big). \end{aligned}$$

The addition in the subscripts is assumed to be done mod k. Figures 4(a) and 4(b) depict G_3 and G_4 respectively. Then, the following lemma holds.

Lemma 2.13. *The graph G_k has the following properties:*

(i) The only P_4s of G_k are the paths $x_iy_iy_{i+1}z_i$, $y_{i+1}z_iz_{i+1}x_i$, and $y_{i+1}x_{i+1}x_iz_{i+1}$ for $0 \le i < k$.

(ii) The graph G_k has a single non-trivial P_4-component.

(iii) The directed edges y_iy_{i+1} $(0 \le i < k)$ form a directed cycle of length k in the non-trivial P_4-component of G_k.

(iv) No directed cycle of length less than k exists in the non-trivial P_4-component of G_k.

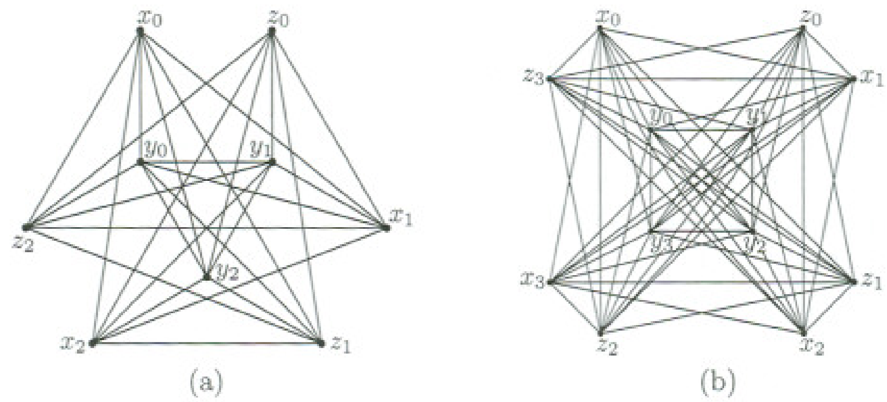

Figure 4: Graphs that have P_4-components with cyclic P_4-transitive orientation.

Proof: (i) Let $abcd$ be a P_4 of the graph G_k. First, suppose that the vertex a is y_{i+1} (for some $i = 0, \ldots, k-1$). Then the vertices c and d can only be x_i and z_{i+1}, since these are the only vertices of G_k not adjacent to y_{i+1}: if $d = x_i$, then $b = z_i$, and the P_4 is $y_{i+1}z_iz_{i+1}x_i$; if $d = z_{i+1}$, then $b = x_{i+1}$, and the P_4 is $y_{i+1}x_{i+1}x_iz_{i+1}$. These are the last two P_4s in the statement of the lemma. Now, suppose that $a \notin Y_k$; we may also assume without loss of generality that $d \notin Y_k$, thus avoiding to get the P_4s of the previous case again (traversed from back to front). But then $a, d \in X_k \cup Z_k$; since a and d are not adjacent, they can only be x_i and z_i for some $i = 0, \ldots, k-1$. Moreover, the remaining two vertices b and c, which are not adjacent to d and a respectively, can only be y_i and y_{i+1}. Therefore the P_4s in this case are the paths $x_iy_iy_{i+1}z_i$.

(ii) This property follows from the fact that the P_4s $x_iy_iy_{i+1}z_i$, $y_{i+1}z_iz_{i+1}x_i$, $z_{i+1}x_i$-$x_{i+1}y_{i+1}$, and $x_{i+1}y_{i+1}y_{i+2}z_{i+1}$ are adjacent and therefore belong to the same P_4-component for all i such that $0 \leq i < k$.

(iii) The sequence of P_4s in the proof of property (ii) implies that if the edge y_iy_{i+1} is oriented towards y_{i+1} then the edge $y_{i+1}y_{i+2}$ will be oriented towards y_{i+2}. The property follows.

(iv) From the P_4s of the graph G_k (see property (i)), we note that all their edges connect vertices whose subscripts differ by at most 1. Let us assume without loss of generality that the edges y_iy_{i+1} are oriented towards y_{i+1}. Then, from the P_4 $x_iy_iy_{i+1}z_i$, the edge x_iy_i is oriented towards x_i and the edge $y_{i+1}z_i$ towards y_{i+1}. Since the edge $y_{i+1}z_i$ is oriented towards y_{i+1}, the edges z_iz_{i+1} and $z_{i+1}x_i$ of the P_4 $y_{i+1}z_iz_{i+1}x_i$ are both oriented towards z_{i+1}. Finally, since the edge x_iz_{i+1} is oriented towards z_{i+1}, the edges $y_{i+1}x_{i+1}$ and $x_{i+1}x_i$ of the P_4 $y_{i+1}x_{i+1}x_iz_{i+1}$ are both oriented towards x_{i+1}. In other words, all the edges of the form a_ib_{i+1} are oriented from a_i to b_{i+1}, whereas the only edges connecting vertices with the same subscript are the edges x_iy_i which are oriented towards x_i; this implies that the length of a directed cycle of the P_4-component cannot be less than k. ∎

3. Recognition of P_4-comparability Graphs

The main idea of the algorithm is to build the P_4-components of the given graph G by considering all the P_4s of G; this is achieved by unioning in a single P_4-component the P_4-components of the edges of each such path, while it is made sure that the edges are compatibly oriented. It is important to note that the orientation of two edges in the same

P_4-component is not free to change relative to each other; either the orientation of all the edges in the component stays the same or it is inverted for all the edges. If no compatible orientation can be found or if the resulting P_4-components contain directed cycles, then the graph is not a P_4-comparability graph. The P_4s are produced by means of BFS-traversals of the graph G starting from each of G's vertices.

The algorithm is described in more detail below. Initially, each edge of G belongs to a P_4-component by itself.

Recognition Algorithm.

1. For each vertex v of the graph, we construct the BFS-tree T_v rooted at v and we update the level $level(x)$[3] and the parent p_x of each vertex x in T_v; before the construction of each of the BFS-trees, $level(x) = -1$ for each vertex x of the graph. Then, we process the edges of the graph as follows:
   (i) for each edge $e = uw$ where $level(u) = 1$ and $level(w) = 2$, we check whether there exist edges from w to a vertex in the 3rd level of T_v. If not, then we do nothing. Otherwise, we orient the edges vu, uw, vp_w, and $p_w w$ in a compatible fashion; for example, we orient vu and vp_w away from v, and uw and $p_w w$ away from w (note that if $u = p_w$, we end up processing the edges vu and uw only). If any two of these edges belong to the same P_4-component and have incompatible orientations, then we conclude that the graph G is not a P_4-comparability graph. If any two of these edges belong to different P_4-components, then we union these components into a single component; if the edges do not have compatible orientations, then we invert (during the unioning) the orientation of all the edges of one of the unioned P_4-components.
   (ii) for each edge $e = uw$ where $level(u) = i$ and $level(w) = i + 1$ for $i \geq 2$, we consider the edges $p_u u$ and uw. As in the previous case, if the two edges belong to the same P_4-component and they are not both oriented towards u or away from u, then there is no compatible orientation assignment and the graph is not a P_4-comparability graph. If the two edges belong to different P_4-components, we union the corresponding P_4-components in a single component, while making sure that the edges are oriented in a compatible fashion.
   (iii) for each edge $e = uw$ where $level(u) = level(w) = 2$, we go through all the vertices of the 1st level of T_v. For each such vertex x, we check whether x is adjacent to u or w. If x is adjacent to u but not to w, then the edges vx, xu, and uw form a P_4; we therefore union the corresponding P_4-components while orienting their edges compatibly. We work similarly for the case where x is adjacent to w but not to u, since the edges vx, xw, and wu form a P_4.
2. After all the vertices have been processed, we check whether the resulting non-trivial P_4-components contain directed cycles. This is done by applying topological sorting independently in each of the P_4components; if the topological sorting succeeds then the corresponding component is acyclic, otherwise there is a directed cycle. If any of the P_4-components contains a cycle, then the graph is not a P_4-comparability graph.

[3] The level of the root of a tree is equal to 0.

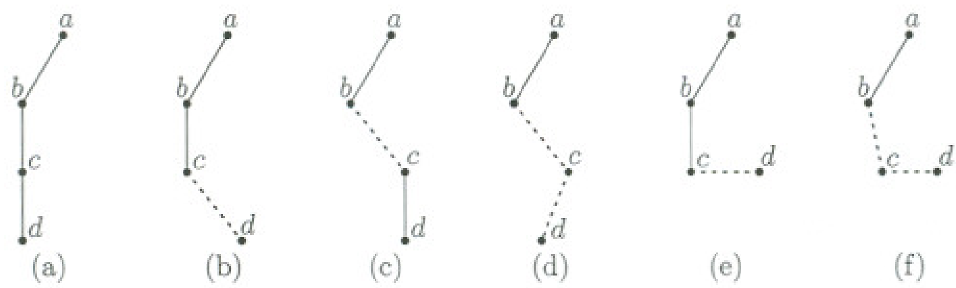

Figure 5: The different positions of a P_4 *abcd* in the BFS tree T_a.

For each P_4-component, we maintain a linked list of the records of the edges in the component, and the total number of these edges. Each edge record contains a pointer to the header record of the component to which the edge belongs; in this way, we can determine in constant time the component to which an edge belongs and the component's size. Unioning two P_4-components is done by updating the edge records of the smallest component and by linking them to the edge list of the largest one, which implies that the union operation takes time linear in the size of the smallest component. As mentioned above, in the process of unioning, we may have to invert the orientation in the edge records that we link, if the current orientations are not compatible.

The correctness of the algorithm follows from the fact that all the P_4s of the given graph are taken into account (see Lemma 3.1 below), from the correct orientation assignment on the edges of these paths, and from Lemma 2.2 in conjunction with Step 2 of the algorithm.

Lemma 3.1. *The algorithm takes into account all the P_4s of the given graph G.*

Proof: Let *abcd* be a P_4 of the graph G. Since the algorithm works with the BFS-trees of all the vertices of G, it will work with the BFS-tree T_a of the vertex a. Let us investigate the different positions that this path may assume in T_a. Clearly, the vertices a, b, and c have to belong to the 0th, 1st, and 2nd level respectively; the vertex d may belong to the 2nd or 3rd level, but not to the 1st level since d is not adjacent to a. All the possible positions of the path are shown in Figure 5; the solid lines, the slanting dashed lines, and the horizontal lines represent tree edges, cross edges, and level edges respectively. The first four cases of Figure 5 are covered by the combination of Steps 1(i) and 1(ii) of the algorithm: no matter which of the four cases is the case for *abcd*, the edges *ab* and *bc* are placed in the same P_4-component with the edge $p_c c$ in Step 1(i) and they are oriented compatibly; the edge *cd* is placed in the same component with the other two in Step 1(ii) when it is unioned and oriented compatibly with the edge $p_c c$ as well. The final two cases of Figure 5 are covered by the Step 1(iii) of the algorithm. ∎

Time and Space Complexity. Computing the BFS-tree T_v of the vertex v of G takes $O(1 + m(v)) = O(1 + m)$ time, where $m(v)$ is the number of edges in the connected component of G to which v belongs. Processing the tree T_v includes processing the edges and checking for compatible orientation, and unioning P_4-components. If we ignore P_4-component unioning, then, each of the Steps 1(i) and 1(ii) takes constant time per processed edge; the parent of a vertex in the tree can be determined in constant time with the use of an auxiliary array, and the P_4-component of an edge is determined in constant time by means of the pointer to the component head record (these pointers are updated during unioning). The Step 1(iii) of the algorithm takes time $O(\deg(v))$ for each edge in the 2nd level of the tree, where by $\deg(v)$ we denote the degree of the vertex v; this implies a total of $O(m \deg(v))$ time for the Step 1(iii) for the tree T_v. Now, the time required for all the P_4-component union operations during the processing of all the BFS-trees is $O(m \log m)$; there cannot be more than $m - 1$ such operations (we start with m P_4-components and we may end up with only one), and each one of them takes time linear in the size of the smallest of the two components that are unioned. Finally, checking whether a non-trivial P_4-component is acyclic takes $O(1 + m_i)$, where m_i is the number of edges of the component. Thus, the total time taken by Step 2 is $O(\sum_i (1 + m_i)) = O(m)$, since there are at most m P_4-components and $\sum_i m_i = m$. Thus, the overall time complexity is $O\big(\sum_v (1 + m + m \deg(v)) \; + m \log m + m) = O(n + m^2)$, since $\sum_v \deg(v) = 2m$.

The space complexity is linear in the size of the graph G; the information stored in order to help processing each BFS-tree is constant per vertex, and the handling of the P_4-components requires one record per edge and one record per component. Thus, the space required is $O(n + m)$.

Therefore, we have proved the following result:

Theorem 3.1. *It can be decided whether a simple graph on n vertices and m edges is a P_4-comparability graph in $O(n + m^2)$ time and $O(n + m)$ space.*

Remark. It must be noted that there are simpler ways of producing the P_4s of a graph in $O(n + m^2)$ time. However, such approaches require $\Theta(n^2)$ space. For example, Raschle and Simon note that a P_4 is uniquely determined by its wings [28]; this implies that the P_4s can be determined by considering all $\Theta(m^2)$ pairs of edges and by checking if the edges in each such pair are the wings of a P_4. In order not to exceed the $O(m^2)$ time complexity, the information on whether two vertices are adjacent should be available in constant time, something that necessitates a $\Theta(n^2)$-space adjacency matrix.

4. Acyclic P_4-transitive Orientation

Although each of the P_4-components of the given graph G which is produced by the recognition algorithm is acyclic, directed cycles may arise when all the P_4-components are placed together; obviously, these cycles will include edges from more than one P_4-component. Appropriate inversion of the orientation of some of the components will yield the desired acyclic P_4-transitive orientation.

Our algorithm to compute the acyclic P_4-transitive orientation of a P_4-comparability graph relies on the processing of the P_4-components of the given graph G and focuses on edges incident upon the vertices of the non-trivial P_4-component which is currently being processed. It assigns orientations in a greedy fashion, and avoids both the contraction step and the recursive call of the orientation algorithms of Hoàng and Reed [17], and Raschle and Simon [28]. More specifically, the algorithm works as follows:

Orientation Algorithm.

1. We apply the recognition algorithm of the previous section on the given graph G, which produces the P_4-components of G and an acyclic P_4-transitive orientation of each component.

2. We sort the non-trivial P_4-components of G by non-decreasing number of vertices; let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_h$ be the resulting ordered sequence. We associate with each $\mathcal{C}_i$ a `mark` and a `counter` field which are initialized to 0.

3. For each P_4-component $\mathcal{C}_i$ $(1 \leq i < h)$ in order, we do:
   By going through the vertices in $V(\mathcal{C}_i)$, we collect the edges which are incident upon a vertex in $V(\mathcal{C}_i)$ and belong to a P_4-component $\mathcal{C}_j$ where $j > i$. Then, for each such edge e, we increment the `counter` field associated with the P_4-component to which e belongs. Next, we go through the collected edges once more. This time, for such an edge e, we check whether the P_4-component to which e belongs has its `mark` field equal to 0 and its `counter` field equal to the total number of edges of the component; if yes, then we set the `mark` field of the component to 1, and, in case e is not oriented towards its endpoint in $V(\mathcal{C}_i)$, we flip the component's orientation (by updating a corresponding boolean variable). After that, we set the `counter` field of the component to which e belongs to 0; in this way, the `counter` fields of all the non-trivial P_4-components are equal to 0 every time a P_4-component starts getting processed in Step 3.

4. We orient the edges which belong to the trivial P_4-components: this can be easily done by topologically sorting the vertices of G using only the oriented edges of the non-trivial components, and orienting the remaining edges in accordance with the topological order of their incident vertices.

Note that in Step 3 we process all the non-trivial P_4-components of the given graph G except for the largest one. This implies that the vertex set $V(\mathcal{C}_i)$ of each P_4-component $\mathcal{C}_i$ $(1 \leq i < h)$ that we process is a proper subset of the vertex set V of G; if $V(\mathcal{C}_i) = V$, then $V(\mathcal{C}_h) = V$ as well, which implies that $\mathcal{C}_i = \mathcal{C}_h$ (Lemma 2.6), a contradiction. Thus, the discussion in Section 2 regarding the P_4-components of type A and type B applies to each such $\mathcal{C}_i$. Moreover, according to Lemma 2.10, the P_4-components whose `mark` field is set to 1 in Step 3 are components which are of type B with respect to the currently processed component $\mathcal{C}_i$. Each edge of these components has exactly one endpoint in $V(\mathcal{C}_i)$ (see Lemma 2.9, statement (ii)), so that it is valid to try to orient such an edge towards that endpoint. Furthermore, Lemma 2.9 (statement (iii)) implies that if such an edge gets oriented towards its endpoint which belongs to $V(\mathcal{C}_i)$, then so do all the edges of the same P_4-component. In the case that the set R in the partition of the vertices in $V - V(\mathcal{C}_i)$ (as described in Section 2) is empty, there are no P_4-components of type B with respect to $\mathcal{C}_i$. While processing $\mathcal{C}_i$, our algorithm updates the `counter` fields of the components that contain an edge incident upon a vertex in $V(\mathcal{C}_i)$, finds that none of these components ends up having its `counter` field equal to the number of its edges, and thus does nothing further.

The orientation algorithm does not compute the sets R, P, and Q with respect to the currently processed P_4-component $\mathcal{C}_i$. These sets can be computed in $O(n)$ time for each $\mathcal{C}_i$ as follows. We use an array with one entry per vertex of the graph G; we initialize the array entries corresponding to vertices in $V(\mathcal{C}_i)$ to 0 and all the remaining ones to -1. Let v_1 and v_2 be an arbitrary midpoint and an arbitrary endpoint of a P_4 in $\mathcal{C}_i$. We go through the vertices adjacent to v_1 and if the vertex does not belong to $V(\mathcal{C}_i)$, we set the corresponding entry to 1. Next, we go through the vertices adjacent to v_2; this time, if the vertex does not belong to $V(\mathcal{C}_i)$, we increment the corresponding entry. Then, the vertices in $\mathcal{C}_i$, R, and Q are the vertices whose corresponding array entries are equal to 0, 1, and -1 respectively, while the remaining vertices belong to P and their corresponding entries are larger than 1; recall that every vertex in $V - V(\mathcal{C}_i)$ which is adjacent to an endpoint of a P_4 of $\mathcal{C}_i$ is also adjacent to any midpoint.

Correctness of the Algorithm. The acyclicity of the directed graph produced by our orientation algorithm relies on the following two lemmata.

Lemma 4.1. *Let $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_h$ be the sequence of the non-trivial P_4-components of the given graph G ordered by non-decreasing vertex number. Consider the set $S_i = \{\mathcal{C}_j \mid j < i$ and $\mathcal{C}_i$ is of type B with respect to $\mathcal{C}_j\}$ and suppose that $S_i \neq \emptyset$. If $\hat{\imath} = \min\{j \mid \mathcal{C}_j \in S_i\}$, then our algorithm orients the edges of the component $\mathcal{C}_i$ towards their endpoint which belongs to $V(\mathcal{C}_{\hat{\imath}})$.*

Proof: The P_4-component $\mathcal{C}_i$ receives an arbitrary P_4-transitive orientation in Step 1 of the orientation algorithm. Since $\hat{\imath} = \min\{j \mid \mathcal{C}_j \in S_i\}$, then the P_4-component $\mathcal{C}_i$ is not of type B with respect to any of the components $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_{\hat{\imath}-1}$; thus, its `mark` field retains its 0 value in the first $\hat{\imath} - 1$ iterations of the `for`-loop in Step 3, since the value of the `counter` field of $\mathcal{C}_i$ will not be equal to the number of its edges for any of $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_{\hat{\imath}-1}$ (Lemma 2.10). Then, in the $\hat{\imath}$-th iteration (during which the component $\mathcal{C}_{\hat{\imath}}$ is processed), the `mark` field of $\mathcal{C}_i$ is set to 1 and $\mathcal{C}_i$ is oriented so that one of its edges points towards its endpoint which belongs to $V(\mathcal{C}_{\hat{\imath}})$. According to Lemma 2.9 (statement (iii)), the latter implies that all the edges of $\mathcal{C}_i$ are oriented towards their endpoint which belongs to $V(\mathcal{C}_{\hat{\imath}})$. This orientation will not change in subsequent iterations of the `for`-loop of Step 3, since the `mark` field of $\mathcal{C}_i$ has been set to 1; nor will it change in Step 4. ∎

Lemma 4.2. *Let $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_h$ be the non-trivial P_4-components of a graph G ordered by non-decreasing vertex number and suppose that each component has received an acyclic P_4-transitive orientation. Consider the set $S_i = \{\mathcal{C}_j \mid j < i$ and $\mathcal{C}_i$ is of type B with respect to $\mathcal{C}_j\}$, for $i = 1, 2, \ldots, h$. If the edges of each P_4-component $\mathcal{C}_i$ such that $S_i \neq \emptyset$ get oriented towards their endpoint which belongs to $V(\mathcal{C}_{\hat{\imath}})$, where $\hat{\imath} = \min\{j \mid \mathcal{C}_j \in S_i\}$, then the resulting directed subgraph of G spanned by the edges of the $\mathcal{C}_i$s $(1 \leq i \leq h)$ does not contain a directed cycle.*

Proof: Clearly, if the graph G has only one non-trivial P_4-component, then there cannot exist a directed cycle, since each P_4-component is acyclic. Let us now consider the case in which the graph has at least two non-trivial P_4-components, and let us suppose for contradiction that the orientation algorithm produces a directed graph that has a directed cycle. Then, in light of Lemma 2.3, there will exist an oriented triangle which forms a directed cycle; let the triangle have vertices v, u, and w. The edges of the triangle belong to non-trivial P_4-components. Let these components be $\mathcal{C}_j$, $\mathcal{C}_k$, and $\mathcal{C}_\ell$, containing the edges uw, vw, and uv respectively, and let us assume without loss of generality that $\ell = \min\{j, k, \ell\}$. Then, the vertex w does not belong to $V(\mathcal{C}_\ell)$. If $w \in V(\mathcal{C}_\ell)$, then the edge uw which belongs to $\mathcal{C}_j$ would have both endpoints in $V(\mathcal{C}_\ell)$. This would imply that $V(\mathcal{C}_j) \subseteq V(\mathcal{C}_\ell)$, according to Lemma 2.7. Moreover, since $\ell \leq j$, we have that $|V(\mathcal{C}_\ell)| \leq |V(\mathcal{C}_j)|$. Therefore, $V(\mathcal{C}_j) = V(\mathcal{C}_\ell)$, which implies that $\mathcal{C}_j = \mathcal{C}_\ell$ (Lemma 2.6). Similarly, $\mathcal{C}_k = \mathcal{C}_\ell$. But then, all three edges of the triangle belong to the same P_4-component, in contradiction to the fact that every P_4-component is acyclic. Thus, $w \notin V(\mathcal{C}_\ell)$.

Since every P_4-component is acyclic, at least two of the P_4-components $\mathcal{C}_j$, $\mathcal{C}_k$, and $\mathcal{C}_\ell$ must be different. In fact, they are all different. Note that if the three edges of the triangle participated in two distinct P_4-components, then $j = k$, since $\ell \neq j$ and $\ell \neq k$ because $w \notin V(\mathcal{C}_\ell)$. Moreover, $\mathcal{C}_j$ would be either of type A or of type B with respect to $\mathcal{C}_\ell$. In the former case, Lemma 2.8 would imply that the edges uw and vw would be oriented either both towards w or both away from it, and thus the triangle with vertices u, v, and w could not form a directed cycle. In the latter case, according to Lemma 2.9 (statement (iv)), the edges uw and vw would again be oriented either both towards w or both away from it, and thus the triangle with vertices u, v, and w could not form a directed cycle in this case either. Therefore, the three edges of the triangle belong to three distinct P_4-components.

Let us consider the P_4-component $\mathcal{C}_\ell$. Because $w \notin V(\mathcal{C}_\ell)$ while $u, v \in V(\mathcal{C}_\ell)$, the other two components $\mathcal{C}_j$ and $\mathcal{C}_k$ are of type A or of type B with respect to $\mathcal{C}_\ell$. If any one of them is of type A, then, according to Lemma 2.8, the edges uv and uw belong to the same P_4-component, in contradiction to the fact that $\mathcal{C}_j \neq \mathcal{C}_k$. Therefore, both $\mathcal{C}_j$ and $\mathcal{C}_k$ are of type B with respect to $\mathcal{C}_\ell$. Let $\hat{\jmath} = \min\{i \mid i < j \text{ and } \mathcal{C}_j \text{ is of type B with respect to } \mathcal{C}_i\}$ and $\hat{k} = \min\{i \mid i < k \text{ and } \mathcal{C}_k \text{ is of type B with respect to } \mathcal{C}_i\}$; note that $\hat{\jmath}$ and $\hat{k}$ are well defined and do not exceed ℓ, since $\ell < j$, $\ell < k$ and both $\mathcal{C}_j$ and $\mathcal{C}_k$ are of type B with respect to $\mathcal{C}_\ell$. Then, according to the statement of the lemma, the orientation convention implies that the edges of the P_4-components $\mathcal{C}_j$ and $\mathcal{C}_k$ are oriented towards their endpoint which belongs to $V(\mathcal{C}_{\hat{\jmath}})$ and $V(\mathcal{C}_{\hat{k}})$ respectively. Then, $\hat{\jmath} \neq \hat{k}$; if $\hat{\jmath} = \hat{k}$, the triangle with vertices u, v, and w could not form a directed cycle, since, according to the orientation convention, the edges uw and vw, which belong to $\mathcal{C}_j$ and $\mathcal{C}_k$ respectively, would be oriented both towards w if $w \in V(\mathcal{C}_{\hat{\jmath}})$, or both away from w if $w \notin V(\mathcal{C}_{\hat{\jmath}})$. Since $\hat{\jmath} \neq \hat{k}$, we may assume without loss of generality that $\hat{\jmath} < \hat{k}$. Then, $\hat{\jmath} < \ell$, since $\hat{\jmath} < \hat{k}$ and $\hat{k} \leq \ell$. We distinguish two cases:

(i) *the P_4-component $\mathcal{C}_\ell$ is not of type B with respect to any component $\mathcal{C}_i$ for $1 \leq i < \ell$*: If the P_4-components $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_{\hat{\jmath}}$ have a common midpoint, then Lemma 2.12 applies: note that $\mathcal{C}_\ell$ is of type B with respect to $\mathcal{C}_j$ (Lemma 2.11, statement (iv)), $\mathcal{C}_j$ is of type B with respect to $\mathcal{C}_{\hat{\jmath}}$, and $|V(\mathcal{C}_\ell)| \geq |V(\mathcal{C}_{\hat{\jmath}})|$ since $\ell > \hat{\jmath}$. Lemma 2.12 implies that the component $\mathcal{C}_\ell$ is of type B with respect to $\mathcal{C}_{\hat{\jmath}}$, which contradicts the fact that $\mathcal{C}_\ell$ is not of type B with respect to any component $\mathcal{C}_i$ $(1 \leq i < \ell)$. If the P_4-components $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_{\hat{\jmath}}$ do not have a common midpoint, then the P_4-components $\mathcal{C}_k$, $\mathcal{C}_j$, and $\mathcal{C}_{\hat{\jmath}}$ do. Suppose for contradiction that they do not, i.e., $V_1(\mathcal{C}_k) \cap V_1(\mathcal{C}_j) \cap V_1(\mathcal{C}_{\hat{\jmath}}) = \emptyset$, where by $V_1(\mathcal{K})$ we denote the set of midpoints of a separable P_4-component $\mathcal{K}$. Moreover, from the assumption that the P_4-components $\mathcal{C}_\ell$, $\mathcal{C}_j$, and $\mathcal{C}_{\hat{\jmath}}$ do not have a common midpoint, we have that $V_1(\mathcal{C}_\ell) \cap V_1(\mathcal{C}_j) \cap V_1(\mathcal{C}_{\hat{\jmath}}) = \emptyset$. Therefore, by taking the union of these two set intersections, we find that

$$
\begin{aligned}
&(\ V_1(\mathcal{C}_k) \cap V_1(\mathcal{C}_j) \cap V_1(\mathcal{C}_{\hat{\jmath}})\ ) \ \cup\ (\ V_1(\mathcal{C}_\ell) \cap V_1(\mathcal{C}_j) \cap V_1(\mathcal{C}_{\hat{\jmath}})\ ) \ =\ \emptyset \\
&\qquad \iff \quad \Big(\ (V_1(\mathcal{C}_k) \cap V_1(\mathcal{C}_j)) \ \cup\ (V_1(\mathcal{C}_\ell) \cap V_1(\mathcal{C}_j))\ \Big) \ \cap\ V_1(\mathcal{C}_{\hat{\jmath}}) \ =\ \emptyset.
\end{aligned}
$$

Since the P_4-components $\mathcal{C}_k$, $\mathcal{C}_j$, and $\mathcal{C}_\ell$ are of type B with respect to one another, Lemma 2.11 (statement (ii)) implies that the sets $V_1(\mathcal{C}_k) \cap V_1(\mathcal{C}_j)$ and $V_1(\mathcal{C}_\ell) \cap V_1(\mathcal{C}_j)$ partition the set $V_1(\mathcal{C}_j)$ of midpoints of $\mathcal{C}_j$; that is, $(\ V_1(\mathcal{C}_k) \cap V_1(\mathcal{C}_j)\ ) \cup (\ V_1(\mathcal{C}_\ell) \cap V_1(\mathcal{C}_j)\ ) = V_1(\mathcal{C}_j)$. Thus, the previous equality is equivalent to $V_1(\mathcal{C}_j) \cap V_1(\mathcal{C}_{\hat{\jmath}}) = \emptyset$. However, this comes into contradiction with the fact that $\mathcal{C}_j$ is of type B with respect to $\mathcal{C}_{\hat{\jmath}}$; therefore, the P_4-components $\mathcal{C}_k$, $\mathcal{C}_j$, and $\mathcal{C}_{\hat{\jmath}}$ have a common midpoint. Then, Lemma 2.12 applies again, for the P_4-components $\mathcal{C}_k$, $\mathcal{C}_{\hat{j}}$, and $\mathcal{C}_{\hat{j}}$ this time ($\mathcal{C}_k$ is of type B with respect to $\mathcal{C}_{\hat{j}}$, $\mathcal{C}_j$ is of type B with respect to $\mathcal{C}_{\hat{j}}$, and $|V(\mathcal{C}_k)| \geq |V(\mathcal{C}_{\hat{j}})|$ since $k > \hat{k} > \hat{j}$), and implies that the component $\mathcal{C}_k$ is of type B with respect to $\mathcal{C}_{\hat{j}}$, which contradicts the minimality of $\hat{k}$, since $\hat{j} < \hat{k}$.

(ii) *the P_4-component $\mathcal{C}_\ell$ is of type B with respect to a component $\mathcal{C}_i$, where $1 \leq i < \ell$:* Let $\hat{\ell} = \min\{i \,|\, i < \ell$ and $\mathcal{C}_\ell$ is of type B with respect to $\mathcal{C}_i\}$. If $\hat{j} < \hat{\ell}$, then we reach a contradiction as in case (i); note that $\hat{j} < \hat{\ell}$ implies that $\ell > \hat{j}$, and recall that the P_4-component $\mathcal{C}_\ell$ cannot be of type B with respect to $\mathcal{C}_{\hat{j}}$. If $\hat{j} = \hat{\ell}$, then the triangle with vertices u, v, and w cannot form a directed cycle; the edges uw and uv, which belong to $\mathcal{C}_j$ and $\mathcal{C}_\ell$ respectively, get oriented both towards u if $u \in V(\mathcal{C}_{\hat{j}})$, or both away from u if $u \notin V(\mathcal{C}_{\hat{j}})$, according to the orientation convention in the statement of the lemma. Suppose now that $\hat{\ell} < \hat{j}$. If the P_4-components $\mathcal{C}_j$, $\mathcal{C}_\ell$, and $\mathcal{C}_{\hat{\ell}}$ have a common midpoint, then Lemma 2.12 applies: note that $\mathcal{C}_j$ is of type B with respect to $\mathcal{C}_\ell$, $\mathcal{C}_\ell$ is of type B with respect to $\mathcal{C}_{\hat{\ell}}$, and $|V(\mathcal{C}_j)| \geq |V(\mathcal{C}_{\hat{\ell}})|$ since $j > \hat{j} > \hat{\ell}$. Lemma 2.12 implies that the component $\mathcal{C}_j$ is of type B with respect to $\mathcal{C}_{\hat{\ell}}$, which contradicts the minimality of $\hat{j}$, since $\hat{\ell} < \hat{j}$. If the P_4-components $\mathcal{C}_j$, $\mathcal{C}_\ell$, and $\mathcal{C}_{\hat{\ell}}$ do not have a common midpoint, then, as in case (i), the P_4-components $\mathcal{C}_k$, $\mathcal{C}_\ell$, and $\mathcal{C}_{\hat{\ell}}$ do. Then, again Lemma 2.12 applies, implying that the component $\mathcal{C}_k$ is of type B with respect to $\mathcal{C}_{\hat{\ell}}$, which contradicts the minimality of $\hat{k}$, since $\hat{\ell} < \hat{j} < \hat{k}$.

In either case, we reached a contradiction, which proves that, if the orientation convention described in the statement of the lemma is followed, then no directed cycle exists in the directed subgraph of G spanned by the edges of the non-trivial P_4-components of G. ∎

Theorem 4.1. *When applied to a P_4-comparability graph, our orientation algorithm produces an acyclic P_4-transitive orientation.*

Proof: The application of the recognition algorithm in Step 1 of the orientation algorithm and the fact that thereafter the inversion of the orientation of an edge causes the inversion of the orientation of all the edges in the same P_4-component imply that the resulting orientation is P_4-transitive. The proof of the theorem will be complete if we show that it is also acyclic. Since the edges of the trivial P_4-components do not introduce cycles given that they are oriented according to a topological sorting of the vertices of the graph, it suffices to show that the directed subgraph of G spanned by the edges of the non-trivial P_4-components of G, which results after the last execution of Step 3, is acyclic. This follows directly from Lemmata 4.1 and 4.2. ∎

Time and Space Complexity. As described in the previous section, Step 1 of the algorithm can be completed in $O(n + m^2)$ time. Step 2 takes $O(m \log m)$ time, since there are $O(m)$ non-trivial P_4-components. Since the degree of a vertex of the graph does not exceed $n - 1$, the total number of edges processed while processing the P_4-component $\mathcal{C}_i$ in Step 3 is $O(n\,|V(\mathcal{C}_i)|)$, where $|V(\mathcal{C}_i)|$ is the cardinality of the vertex set of $\mathcal{C}_i$. This upper bound is $O(n\,(|E(\mathcal{C}_i)| + 1)) = O(n\,|E(\mathcal{C}_i)|)$, because the component $\mathcal{C}_i$ is connected (Lemma 2.1, statement (ii)) and hence $|V(\mathcal{C}_i)| \leq |E(\mathcal{C}_i)| + 1$. The time to process each such edge is $O(1)$, thus implying a total of $O(n\,|E(\mathcal{C}_i)|)$ time for the execution of Step 3 for the component $\mathcal{C}_i$; since an edge of the graph belongs to one P_4-component and a component is processed only once, the overall time for all the executions of Step 3 is $O(nm)$. Finally, Step 4 takes $O(n + m)$ time.

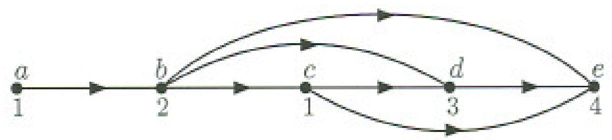

Figure 6

Summarizing, the time complexity of the orientation algorithm is $O(n + m^2)$. It is interesting to note that the time complexity is dominated by the time to execute Step 1; the remaining steps take a total of $O(nm)$ time. Therefore, an $o(n + m^2)$-time algorithm to recognize a P_4-comparability graph and to compute its P_4-components will imply an $o(n + m^2)$-time algorithm for the acyclic P_4-transitive orientation of a P_4-comparability graph. The space complexity is linear in the size of the given graph G.

From the above discussion, we obtain the following theorem.

Theorem 4.2. *Let G be a P_4-comparability graph on n vertices and m edges. Then, an acyclic P_4-transitive orientation of G can be computed in $O(n + m^2)$ time and $O(n + m)$ space.*

Note that the input to our orientation algorithm does not need to be a P_4-comparability graph. If it is not, this will be detected in Step 1, and the algorithm will stop and will report it; otherwise, it will proceed, eventually computing the desired acyclic P_4-transitive orientation.

5. Optimal Coloring and Maximum Clique

In [5], Chvátal proved that if a perfect order of a perfectly orderable graph G is given then an optimal coloring of G can be found by the greedy (first-fit) algorithm in linear time. Moreover, he showed that, for a perfectly orderable graph with chromatic number k, if H is a clique consisting of vertices with colors $c, c+1, \ldots, k$, then there exists a vertex with color $c - 1$ that is adjacent to all the vertices of H. This result can be used in an algorithm to compute the maximum clique of a perfectly orderable graph [5]. As mentioned in [17], it is easy to see that this algorithm can be made to run in $O(n^2)$ time. Below, we show how the above result can be used to yield an $O(n + m)$-time algorithm which, given a perfect order on the vertices of a perfectly orderable graph G, computes a maximum clique of G. Clearly, the algorithm can be applied to the class of P_4-comparability graphs, since a P_4-comparability graph is also a perfectly orderable graph.

It is interesting to note that, unlike the comparability graphs where in each clique the order of vertices by color matches the perfect order of the vertices, in a P_4-comparability graph the vertex with color $c-1$ is not necessarily a predecessor (with respect to the perfect order of the graph) of all the vertices in the clique H. Consider, for example the graph of Figure 6: the indicated orientations of the edges define an acyclic P_4-transitive orientation of the graph, which thus is a P_4-comparability graph; the maximum clique consists of the vertices b, c, d, and e with respective colors 2, 1, 3, and 4.

Maximum Clique Algorithm.

1. We compute an optimal coloring of the graph by applying the first-fit algorithm along the given perfect order; let k be the maximum color assigned.

2. We partition the set of vertices of the graph into the color sets $V_1, V_2, \ldots, V_k$; V_i denotes the set of vertices colored with the color i.

3. We use an auxiliary array with one entry per vertex of the graph, which stores information about whether the corresponding vertex is marked or unmarked; initially, all vertices are unmarked.

4. We mark an arbitrary vertex of the set V_k.

5. `for` $i = k-1, \ldots, 1$
   `for` each vertex v in V_i
   `if` the number of marked vertices adjacent to v equals $k - i$
   we mark the vertex v;
   we exit the inner loop and continue with the next iteration of the outer loop;
   `end-if`

6. The clique consists of the marked vertices.

Note that Chvátal's result implies that, for every iteration of the outer loop in step 5, the inner loop will always produce a vertex adjacent to all the currently marked vertices.

Time and Space Complexity. As mentioned earlier, an optimal coloring of a perfectly orderable graph can be computed by the first-fit algorithm in time linear in the size of the graph, if a perfect order on the graph's vertices is given; thus, step 1 is completed in linear time. Steps 2, 3 and 5 take linear time as well, while step 4 takes constant time. Thus, the time complexity of the maximum clique algorithm is linear in the size of the given graph. The space complexity is also linear in the size of the graph. Therefore, we have:

Theorem 5.1. *Let G be a perfectly orderable graph on n vertices and m edges. If a perfect order on the vertices of G is given, then an optimal coloring and a maximum clique of G can be found in $O(n+m)$ time and space.*

6. Concluding Remarks

In this paper, we presented an $O(n + m^2)$-time and linear space algorithm to recognize whether a graph of n vertices and m edges is a P_4-comparability graph. We also described an algorithm to compute an acyclic P_4-transitive orientation of a P_4-comparability graph which runs in $O(n+m^2)$ time and linear space as well. Both algorithms exhibit the currently best time and space complexities to the best of our knowledge, are simple enough to be easily used in practice, are non-recursive, and admit efficient parallelization. Finally, we also showed how the maximum clique of a perfectly orderable graph can be computed in linear time given a perfect order on the vertices of the graph.

The obvious open question is whether the P_4-comparability graphs can be recognized and oriented in $o(n + m^2)$ time. Note that a better time complexity for the recognition problem — assuming that the recognition process determines the P_4-components as well — will imply a better time complexity for our orientation algorithm.

7. References

[1] S.R. Arikati and U.N. Peled, A polynomial algorithm for the parity path problem on perfectly orderable graphs, *Discrete Appl. Math.* **65** (1996), 5–20.

[2] L. Babel and S. Olariu, A new characterization of P_4-connected graphs, *Proc. 22nd International Workshop on Graph-theoretic concepts in Computer Science (WG '96)* (F. d' Amore, P.G. Franciosa, and A. Marchetti-Spaccamela, eds.), LNCS 1197, 1996, 17–30.

[3] A. Brandstädt, Classes of bipartite graphs related to chordal graphs, *Discrete Appl. Math.* **32** (1991), 51–60.

[4] A. Brandstädt and D. Kratsch, On domination problems for permutation and other graphs, *Theoret. Comput. Sci.* **54** (1987), 181–198.

[5] V. Chvátal, Perfectly ordered graphs, *Annals of Discrete Math.* **21** (1984), 63–65.

[6] D.G. Corneil, H. Lerches, and L. Burlingham, Complement reducible graphs, *Discrete Appl. Math.* **3** (1981), 163–174.

[7] D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. on Comput.* **14** (1985), 926–934.

[8] C.M.H. de Figueiredo, J. Gimbel, C.P. Mello, and J.L. Szwarcfiter, Even and odd pairs in comparability and in P_4-comparability graphs, *Discrete Appl. Math.* **91** (1999), 293–297.

[9] S. Even, A. Pnueli, and A. Lempel, Permutation graphs and transitive graphs, *J. ACM* **19** (1972), 400–410.

[10] P.C. Gilmore and A.J. Hoffman, A characterization of comparability graphs and of interval graphs, *Canad. J. Math.* **16** (1964), 539–548.

[11] M.C. Golumbic, The complexity of comparability graph recognition and coloring, *Computing* **18** (1977), 199–208.

[12] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, Inc., New York, 1980.

[13] M.C. Golumbic, D. Rotem, and J. Urrutia, Comparability graphs and intersection graphs, *Discrete Math.* **43** (1983), 37–46.

[14] M. Habib, R.M. McConnell, C. Paul, and L. Viennot, Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing, *Theoret. Comput. Sci.* **234** (2000), 59–84.

[15] C.T. Hoàng, Efficient algorithms for minimum weighted colouring of some classes of perfect graphs, *Discrete Appl. Math.* **55** (1994), 133–143.

[16] C.T. Hoàng and B.A. Reed, Some classes of perfectly orderable graphs, *J. Graph Theory* **13** (1989), 445–463.

[17] C.T. Hoàng and B.A. Reed, P_4-comparability graphs, *Discrete Math.* **74** (1989), 173–200.

[18] W.L. Hsu and T.H. Ma, Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs, *SIAM J. on Comput.* **28** (1999), 1004–1020.

[19] B. Jamison and S. Olariu, Recognizing P_4-sparse graphs in linear time, *SIAM J. on Comput.* **21** (1992), 381–406.

[20] R.M. McConnell and J. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, *Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms* (1994), 536–545.

[21] R.M. McConnell and J. Spinrad, Linear-time transitive orientation, *Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms* (1997), 19–25.

[22] M. Middendorf and F. Pfeiffer, On the complexity of recognizing perfectly orderable graphs, *Discrete Math.* **80** (1990), 327–333.

[23] R.H. Möhring, Algorithmic aspects of comparability graphs and interval graphs, *Graphs and Order* (I. Rival, ed.), Reidel, Dordrecht, 1985, 41–101.

[24] S.D. Nikolopoulos and C. Papadopoulos, On the performance of the first-fit coloring algorithm on permutation graphs, *Inform. Process. Lett.* **75** (2000), 265–273.

[25] S. Olariu, All variations on perfectly orderable graphs, *J. Combin. Theory* Ser. B **45** (1988), 150–159.

[26] S. Olariu, On sources in comparability graphs, with applications, *Discrete Math.* **110** (1992), 289–292.

[27] A. Pnueli, A. Lempel, and S. Even, Transitive orientation of graphs and identification of permutation graphs, *Canad. J. Math.* **23** (1971), 160–175.

[28] T. Raschle and K. Simon, On the P_4-components of graphs, *Discrete Appl. Math.* **100** (2000), 215–235.

[29] J.P. Spinrad, On comparability and permutation graphs, *SIAM J. on Comput.* **14** (1985), 658–670.

[30] J.P. Spinrad, A. Brandstädt, and L.K. Stewart, Bipartite permutation graphs, *Discrete Appl. Math.* **18** (1987), 279–292.