

# Polar permutation graphs are polynomial-time recognisable\*

Tınaz Ekim<sup>†</sup>Pinar Heggernes<sup>‡</sup>Daniel Meister<sup>‡</sup>

## Abstract

Polar graphs generalise bipartite graphs, cobipartite graphs, and split graphs, and they constitute a special type of matrix partitions. A graph is polar if its vertex set can be partitioned into two, such that one part induces a complete multipartite graph and the other part induces a disjoint union of complete graphs. Deciding whether a given arbitrary graph is polar, is an NP-complete problem. Here, we show that for permutation graphs this problem can be solved in polynomial time. The result is surprising, as related problems like achromatic number and cochromatic number are NP-complete on permutation graphs. We give a polynomial-time algorithm for recognising graphs that are both permutation and polar. Prior to our result, polarity has been resolved only for chordal graphs and cographs.

## 1 Introduction

Many graph problems can be formulated as finding a partition of the vertices such that various parts satisfy certain properties internally, and at the same time certain other properties are satisfied regarding the interaction between these parts. Examples of such problems are the broad variety of colouring and homomorphism problems, and the matrix partition problem; the latter was posed by Feder et al. [15]. The matrix partition problem asks for a partition of the vertex set of a graph into subsets  $A_1, \dots, A_k$  such that each subset is either a clique or an independent set, and pairs of subsets are completely adjacent or completely non-adjacent, depending on a given pattern. If the pattern says that we partition into only cliques and independent sets, and two partition sets  $A_i, A_j$  should be completely adjacent if  $A_i, A_j$  are independent sets, completely non-adjacent if  $A_i, A_j$  are cliques, and there is no restriction for the two other cases, then we get exactly the polar graphs.

Polar graphs were defined in 1985 by Tyshkevich and Chernyak [26]. A graph is *polar* if its vertex set can be partitioned into  $A$  and  $B$  such that  $A$  induces a complete multipartite graph and  $B$  induces a cluster graph, i.e., a disjoint union of complete graphs. Such a partition is called *polar*. Since complements of cluster graphs are exactly complete multipartite graphs (and vice versa), the class of polar graphs is closed under taking complements. Furthermore, it contains the well-known classes of split graphs, bipartite graphs, and cobipartite graphs. If  $A$  is simply an independent set, then the graph (and the partition) is called *monopolar*. In addition to fitting into the matrix partition problem [15] described above, polar partitions can be seen as generalised colourings [5].

---

\*This work was supported by the Research Council of Norway, and the first author was supported by the B. U. Research Fund, grant 09A302P. A preliminary version of the paper was presented at IWOCA 2009 [10].

<sup>†</sup>Industrial Engineering Department, Boğaziçi University, Istanbul, Turkey. Email: [tinaz.ekim@boun.edu.tr](mailto:tinaz.ekim@boun.edu.tr)

<sup>‡</sup>Department of Informatics, University of Bergen, Norway. Emails: [pinar.heggernes@ii.uib.no](mailto:pinar.heggernes@ii.uib.no), [daniel.meister@ii.uib.no](mailto:daniel.meister@ii.uib.no)

$B \setminus A$	indep. set	clique	cluster gr.	compl. multip.
indep. set	P	P	<b>P</b>	?
clique		P	?	<b>P</b>
cluster gr.			?	<b>P</b>
compl. multip.				?

Figure 1: The table shows known and new computational complexity results for  $(A, B)$ -partition problems on permutation graphs, when  $A$  or  $B$  (symmetrically) is an independent set, is a clique, induces a cluster graph, or induces a complete multipartite graph. The boldface entries are the results of this paper. Empty cells are due to the symmetry of the table.

The recognition problems for polar and monopolar graphs are NP-complete [6, 14]. Notice, however, that “admitting a polar partition” can be expressed in monadic second-order logic without using edge-set quantification, and hence, polar graphs of bounded treewidth or bounded clique-width can be recognised in polynomial time, by the results of [1, 8] and [9]. Consequently, it is of interest to find out where the boundary goes between subclasses of polar graphs that are recognisable in polynomial time and those whose recognition is intractable. When it comes to graph classes of unbounded treewidth and clique-width whose intersection with polar graphs can be recognised in polynomial time, so far we know only of chordal graphs [11, 17].

In this paper, we give polynomial-time algorithms for the two problems of deciding whether a given permutation graph is polar or monopolar. Permutation graphs are a well-studied graph class with a large number of theoretical applications [4, 18], and they can be recognised in linear time [24]. Permutation graphs have unbounded treewidth and clique-width [19]. Although many NP-complete problems become tractable on permutation graphs, well-known colouring problems, like cochromatic number [16, 28] and achromatic number [2], remain NP-complete on this graph class. The class of monopolar graphs generalises bipartite graphs and split graphs, and thus the class of monopolar permutation graphs is a generalisation of the class of bipartite permutation graphs. Since permutation graphs are closed under taking complements, our recognition algorithm for monopolar permutation graphs can also be used to recognise polar permutation graphs whose  $B$ -set induces a complete graph. Thus, combined with the well-known recognition results for bipartite graphs, split graphs and cobipartite graphs, our results show that a range of partition problems, in the sense of the introductory matrix partition problem, are polynomial-time solvable on permutation graphs. We summarise these known and new results in Figure 1.

We show separately that monopolar permutation graph recognition and polar permutation graph recognition is polynomial-time solvable. The algorithm for the monopolar case partitions the input graph into small split graphs that satisfy certain conditions. The running time of this algorithm is  $\mathcal{O}(nm)$ , which is a significant improvement from its preliminary version [10]. For the case of polar permutation graphs, the main idea is to delete a suitable set of vertices to reduce the problem to a generalised monopolar recognition problem. As a result, we obtain an  $\mathcal{O}(nm^2)$ -time algorithm for recognising polar permutation graphs, which improves the running time of the preliminary version in [10]. We consider the monopolar permutation graph recognition problem in Section 3, and the polar permutation graph recognition problem in Section 4.

Other results on polynomial-time recognisable subclasses of polar graphs include [25] which studies polar partitions where the size of each independent set and clique is bounded, [11, 17] which give forbidden subgraph characterisations and a recognition algorithm for polar chordal graphs, and [13] which gives similar results for polar cographs. In addition, [22] and [12] give respectively a forbidden subgraph characterisation and a polynomial-time recognition algorithm for bipartite graphs whose line graphs are polar. Finally, [7] gives a polynomial-time recognition algorithm for monopolar claw-free graphs. Another research direction is to study which NP-complete problems become tractable on polar graphs. For example, [23] gives polynomial-time algorithms for finding a minimum maximal independent set in some subclasses of polar graphs. This problem remains NP-hard in polar graphs admitting a polar partition where the size of every independent set is at most one and the size of every clique is at most two.

## 2 Definitions and notation, polar partitions, and permutation graphs

Our input graphs are simple and undirected. Only in Section 3, we use directed graphs (*digraphs*) as auxiliary tools.

Let  $G$  be a simple and undirected graph. We denote its vertex set by  $V(G)$  and its edge set by  $E(G)$ . An edge between vertices  $u$  and  $v$  is denoted by  $uv$ . If  $uv$  is an edge of  $G$  then  $u$  and  $v$  are *adjacent* in  $G$ . For a vertex  $x$  of  $G$ , the *neighbourhood* of  $x$ , denoted as  $N_G(x)$ , is the set of vertices that are adjacent to  $x$ , and  $N_G[x] = N_G(x) \cup \{x\}$ . The *degree* of a vertex  $x$  is  $d_G(x) = |N_G(x)|$ . Let  $X$  be a set of vertices of  $G$ . The subgraph of  $G$  *induced* by  $X$  is denoted as  $G[X]$  and defined as the graph on vertex set  $X$  and edge set the set of edges of  $G$  that join only vertices in  $X$ . By  $G \setminus X$ , we denote the graph  $G[V(G) \setminus X]$ . A graph is called *complete* if every pair of vertices is adjacent. A set  $X \subseteq V(G)$  is called a *clique* if  $G[X]$  is complete, and it is called an *independent set* if  $G[X]$  has no edges. A graph is *connected* if there is a path between every pair of vertices; otherwise, the graph is called *disconnected*. The *connected components* of a graph are the maximal connected (induced) subgraphs.

The *disjoint union* of two graphs  $G$  and  $H$  is the graph on vertex set  $V(G) \cup V(H)$  and edge set  $E(G) \cup E(H)$ ; the disjoint union of more than two graphs is defined analogously. The *complement* of  $G$ , denoted as  $\overline{G}$ , is the graph on vertex set  $V(G)$  and edge set  $\{uv \notin E(G) \mid u, v \in V(G) \text{ and } u \neq v\}$ . A *complete multipartite graph* is the complement of the disjoint union of complete graphs. Equivalently, the vertex set of a complete multipartite graph admits a unique partition into maximal independent sets.

For a given graph  $G$ , a partition  $(A, B)$  of  $V(G)$ , where  $A$  or  $B$  can also be empty, is called *polar* if  $G[A]$  is a complete multipartite graph and  $G[B]$  is a disjoint union of complete graphs, i.e., a cluster graph. Equivalently,  $(A, B)$  is a polar partition for  $G$  if both  $\overline{G}[A]$  and  $G[B]$  are cluster graphs. Note that  $(A, B)$  is a polar partition for  $G$  if and only if  $(B, A)$  is a polar partition for  $\overline{G}$ . A polar partition  $(A, B)$  for  $G$  is called *monopolar* if  $A$  is an independent set in  $G$ . We say that a polar partition  $(A, B)$  for  $G$  is *B-maximal* if there is no polar partition  $(A', B')$  for  $G$  with  $B \subset B'$ . Note that, for a polar partition  $(A, B)$ , if there is a vertex  $u$  in  $A$  without a neighbour in  $B$  then  $(A \setminus \{u\}, B \cup \{u\})$  is also a polar partition for  $G$ . Hence the following result is immediate.

**Lemma 2.1** *Let  $(A, B)$  be a B-maximal polar partition for a graph  $G$ . Every vertex in  $A$  has a neighbour in  $B$ .*

A graph  $G$  is called *split graph* if  $V(G)$  admits a partition  $(A, B)$  such that  $A$  is an independent set of  $G$  and  $B$  is a clique of  $G$ . Such a partition is called *split partition*. It holds that split graphs are special monopolar graphs and split partitions are special monopolar partitions.

Let  $n \geq 1$  and  $\pi$  be a permutation over  $\{1, \dots, n\}$ , i.e., a bijection between  $\{1, \dots, n\}$  and  $\{1, \dots, n\}$ . We will denote  $\pi$  equivalently as a *permutation sequence*  $(\pi(1), \dots, \pi(n))$ . The position of an integer  $x$  in  $\pi$  is  $\pi^{-1}(x)$ . By  $\pi^{-1}(X)$  for  $X \subseteq \{1, \dots, n\}$ , we mean  $\{\pi^{-1}(x) \mid x \in X\}$ . The *inversion graph* of  $\pi$  has vertex set  $\{1, \dots, n\}$  and two vertices  $u, v$  are adjacent if and only if  $(u - v)(\pi^{-1}(u) - \pi^{-1}(v)) < 0$ . A graph is a *permutation graph* if it is isomorphic to the inversion graph of a permutation sequence [4, 18]. Permutation graphs can be recognised in linear time [24]. Permutation graphs also have a geometric intersection model: for two horizontal lines, mark  $n$  points on each line, assign to each point on the upper line a point on the lower line, and connect the two points by a line segment. The corresponding graph has a vertex for every line segment and two vertices are adjacent if the corresponding line segments cross. This representation is called a *permutation diagram*. A graph is a permutation graph if and only if it has a permutation diagram. Given a permutation graph, a permutation diagram for it can be computed in linear time [24]. It is important to note that every induced subgraph of a permutation graph is a permutation graph. For our purposes, we assume that a permutation graph is given as a permutation sequence and equal to the defined inversion graph. Every permutation graph with permutation sequence  $\pi$  has a permutation diagram  $\mathcal{D}$  in which the endpoints of the line segments on the lower line appear in the same order as they appear in  $\pi$ . For such pairs  $(\pi, \mathcal{D})$ , we say that  $\mathcal{D}$  *corresponds to*  $\pi$ . For convenience reasons, sometimes we will not distinguish between vertices of the graph and line segments in the permutation diagram; however, the meaning will always be clear. Line segments, and thus vertices, have an upper and a lower endpoint.

### 3 Recognising monopolar permutation graphs

A polar graph  $G$  is monopolar if it has a polar partition  $(A, B)$  such that  $A$  is an independent set of  $G$ . Monopolar graphs are thus a generalisation of bipartite graphs and split graphs. We show in this section that monopolarity is polynomial-time decidable for permutation graphs. Our algorithm has running time  $\mathcal{O}(nm)$  and is even able to output a monopolar partition, if such exists.

Since connected bipartite graphs admit unique partitions and since partitions for split graphs are very restricted, an interesting question is whether connected monopolar graphs have a similar property regarding monopolar partitions. A simple example shows that this is not the case: a simple path is a monopolar graph, since it is bipartite, but it has exponentially many monopolar partitions. The algorithm that we present in this section is able to answer an even more general question: given a connected permutation graph  $G$  and a set  $F \subseteq V(G)$ , does  $G$  have a monopolar partition  $(A, B)$  such that  $A \subseteq F$ . By choosing  $F = V(G)$ , we get exactly the monopolar permutation graph recognition problem.

We partition this section into two parts. In the first part, we present the main algorithmic idea. It is based on a partition of the graph into small split graphs. In the second part, we consider implementation aspects. As important auxiliary results, we will closely consider split partitions.

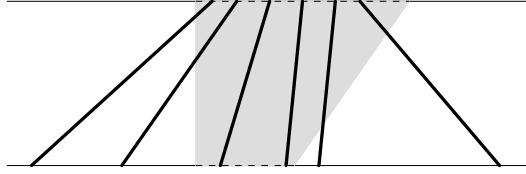


Figure 2: The figure shows a trapezoid (in grey) and some of the vertices in the intersection of the trapezoid, as it may appear in a permutation diagram. The vertices form an independent set.

### 3.1 Algorithm and correctness

Let  $G$  be a permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . A *trapezoid* in  $\mathcal{D}$  is a pair  $(I_1, I_2)$  of intervals of integers with  $I_1 = \{i_1, \dots, i'_1\}$  and  $I_2 = \{i_2, \dots, i'_2\}$  for  $1 \leq i_1 \leq i'_1 \leq |V(G)|$  and  $1 \leq i_2 \leq i'_2 \leq |V(G)|$ . Trapezoids are the main substructures in permutation diagrams that we consider in this paper. We define four sets of vertices for trapezoids. Let  $\mathbf{T} = (I_1, I_2)$  be a trapezoid in  $\mathcal{D}$  with  $I_1 = \{i_1, \dots, i'_1\}$  and  $I_2 = \{i_2, \dots, i'_2\}$ . We define the *left side*, the *right side*, the *containment* and the *intersection* of  $\mathbf{T}$ :

- left side:  $L(\mathbf{T}) =_{\text{def}} \{x \in V(G) \mid x < i_1 \text{ and } \pi^{-1}(x) < i_2\}$
- right side:  $R(\mathbf{T}) =_{\text{def}} \{x \in V(G) \mid i'_1 < x \text{ and } i'_2 < \pi^{-1}(x)\}$
- containment:  $\text{con}(\mathbf{T}) =_{\text{def}} \{x \in V(G) \mid i_1 \leq x \leq i'_1 \text{ and } i_2 \leq \pi^{-1}(x) \leq i'_2\}$
- intersection:  $\text{int}(\mathbf{T}) =_{\text{def}} V(G) \setminus (L(\mathbf{T}) \cup R(\mathbf{T}))$ .

Note that  $\text{con}(\mathbf{T}) \subseteq \text{int}(\mathbf{T})$ . For  $X \subseteq V(G)$ , the  $X$ -trapezoid in  $\mathcal{D}$  is the trapezoid  $\mathbf{U} = (J_1, J_2)$  with  $J_1 = \{\min X, \dots, \max X\}$  and  $J_2 = \{\min \pi^{-1}(X), \dots, \max \pi^{-1}(X)\}$ . Furthermore,  $X \subseteq \text{con}(\mathbf{U})$ . Informally, the  $X$ -trapezoid is the smallest trapezoid that contains  $X$ . Every trapezoid is not necessarily an  $X$ -trapezoid for some set  $X$ . A schematic example of a trapezoid in a permutation diagram, with some vertices in its containment and intersection, is shown in Figure 2. It follows from the properties of cliques in permutation diagrams that  $\min \pi^{-1}(X) = \pi^{-1}(\max X)$  and  $\max \pi^{-1}(X) = \pi^{-1}(\min X)$  when  $X$  is a clique.

Our solution of the problem in this section is based on the following idea: Let  $(A, B)$  be a monopolar partition. Then every clique in  $G[B]$  can be connected to a trapezoid, and every vertex in  $A$  intersects with some trapezoid. For  $X \subseteq V(G)$  with  $1 \leq |X| \leq 2$ , we call the  $X$ -trapezoid  $\mathbf{T}$  *good* (with respect to  $F$ ) if  $G[\text{int}(\mathbf{T})]$  has a split partition  $(C, D)$  so that  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T}) \subseteq C \subseteq F$  and  $X \subseteq D \subseteq \text{con}(\mathbf{T})$ . We call this split partition  $(C, D)$  *good* for  $\mathbf{T}$  (with respect to  $F$ ). For our algorithm, we construct an auxiliary digraph whose vertices correspond to the good trapezoids, and our problem is solved by deciding the existence of a path.

In a digraph edges are ordered pairs, and they are called *arcs*. An arc from vertex  $u$  to vertex  $v$  is denoted by  $(u, v)$ . A path in a digraph follows the directions of the arcs; hence there is a path  $(x_1, x_2, \dots, x_s)$  if there are arcs  $(x_1, x_2), (x_2, x_3), \dots, (x_{s-1}, x_s)$ . Such a path from  $x_1$  to  $x_s$  is called an  $x_1, x_s$ -path.

Let  $G$  be a permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ , and let  $F \subseteq V(G)$ . By  $\text{aux}(\mathcal{D}, F)$ , we denote the auxiliary digraph with the following vertices and arcs:

- $\text{aux}(\mathcal{D}, F)$  has a vertex for every good  $X$ -trapezoid where  $X \subseteq V(G)$  and  $1 \leq |X| \leq 2$ ; denote by  $\mathbf{T}_x$  the trapezoid that corresponds to vertex  $x$  of  $\text{aux}(\mathcal{D}, F)$ ;
- for two vertices  $u$  and  $v$  of  $\text{aux}(\mathcal{D}, F)$ ,  $(u, v)$  is an arc in  $\text{aux}(\mathcal{D}, F)$  if  $\text{con}(\mathbf{T}_v) \subseteq \text{R}(\mathbf{T}_u)$  and  $\text{R}(\mathbf{T}_u) \cap \text{L}(\mathbf{T}_v) = \emptyset$  and  $(\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)) \cup (\text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v))$  is an independent set of  $G$ .

To complete the definition of  $\text{aux}(\mathcal{D}, F)$ , there are two more vertices,  $\mathbf{0}$  and  $\mathbf{1}$ , and there is an arc  $(\mathbf{0}, x)$  for every vertex  $x$  in  $\text{aux}(\mathcal{D}, F)$  with  $\text{L}(\mathbf{T}_x) = \emptyset$ , and there is an arc  $(x, \mathbf{1})$  for every vertex  $x$  in  $\text{aux}(\mathcal{D}, F)$  with  $\text{R}(\mathbf{T}_x) = \emptyset$ .

**Lemma 3.1** *Let  $G$  be a connected permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . Let  $F \subseteq V(G)$ . The auxiliary digraph  $\text{aux}(\mathcal{D}, F)$  has a  $\mathbf{0}, \mathbf{1}$ -path if and only if  $G$  has a monopolar partition  $(A, B)$  with  $A \subseteq F$ .*

**Proof.** Let  $P = (x_0, \dots, x_s)$  be a  $\mathbf{0}, \mathbf{1}$ -path in  $\text{aux}(\mathcal{D}, F)$ . Note that  $s \geq 2$ , since  $(\mathbf{0}, \mathbf{1})$  is not an arc of  $\text{aux}(\mathcal{D}, F)$  due to the definition of  $\text{aux}(\mathcal{D}, F)$ . Let  $\mathbf{T}_1, \dots, \mathbf{T}_{s-1}$  be the trapezoids corresponding to  $x_1, \dots, x_{s-1}$ , respectively. For  $1 \leq i \leq s-1$ , let  $(C_i, D_i)$  be a good split partition for  $\mathbf{T}_i$ . We show that  $(C_1 \cup \dots \cup C_{s-1}, D_1 \cup \dots \cup D_{s-1})$  is a monopolar partition for  $G$ . First, observe that  $(C_1 \cup \dots \cup C_{s-1}) \cap (D_1 \cup \dots \cup D_{s-1}) = \emptyset$ . Otherwise, there are  $x \in V(G)$  and  $1 \leq i, j \leq s-1$  with  $i \neq j$  such that  $x \in C_i \cap D_j$ . By definition of good split partition, this particularly means that  $x \in \text{con}(\mathbf{T}_j)$ . Since  $\text{con}(\mathbf{T}_i) \cap \text{con}(\mathbf{T}_j) = \emptyset$  due to the definition of the arcs of  $\text{aux}(\mathcal{D}, F)$ , it holds that  $x \in \text{int}(\mathbf{T}_i) \setminus \text{con}(\mathbf{T}_i)$ . Then, however,  $\text{con}(\mathbf{T}_j) \cap \text{int}(\mathbf{T}_i) \neq \emptyset$ , which is a contradiction to the properties of permutation diagrams and the definition of the arcs of  $\text{aux}(\mathcal{D}, F)$ . Second, we show that  $(C_1 \cup \dots \cup C_{s-1}) \cup (D_1 \cup \dots \cup D_{s-1}) = V(G)$ . Suppose there is  $x \in V(G)$  such that there is no  $1 \leq i \leq s-1$  with  $x \in \text{int}(\mathbf{T}_i)$ . Since  $\text{L}(\mathbf{T}_1) = \emptyset$  and  $\text{R}(\mathbf{T}_{s-1}) = \emptyset$ , there is  $1 \leq i < s-1$  such that  $x \in \text{R}(\mathbf{T}_i) \cap \text{L}(\mathbf{T}_{i+1})$ . This, however, contradicts the definition of the arcs of  $\text{aux}(\mathcal{D}, F)$ . We conclude that  $(C_1 \cup \dots \cup C_{s-1}, D_1 \cup \dots \cup D_{s-1})$  defines a partition for  $G$ .

Since  $D_i \subseteq \text{con}(\mathbf{T}_i)$  for every  $1 \leq i \leq s-1$  and  $\text{con}(\mathbf{T}_i) \cap \text{con}(\mathbf{T}_j) = \emptyset$  for every  $1 \leq i < j \leq s-1$ ,  $G[D_1 \cup \dots \cup D_{s-1}]$  is the disjoint union of complete graphs. It remains to show that  $C_1 \cup \dots \cup C_{s-1}$  is an independent set of  $G$ . Suppose the contrary, i.e., there are  $u, v \in V(G)$  and  $1 \leq i \leq j \leq s-1$  with  $u \in C_i$  and  $v \in C_j$  and  $uv \in E(G)$ . Note that  $i \neq j$  since  $(C_i, D_i)$  is a split partition for  $G[\text{int}(\mathbf{T}_i)]$  and therefore  $C_i$  is an independent set of  $G$ . If  $j - i \geq 2$  then  $u \in C_{i+1}$  or  $v \in C_{j-1}$ , so that we can assume without loss of generality that  $j - i = 1$ . Since  $u \notin \text{con}(\mathbf{T}_i)$  and  $v \notin \text{con}(\mathbf{T}_j)$ ,  $u$  and  $v$  are contained in  $(\text{int}(\mathbf{T}_i) \setminus \text{con}(\mathbf{T}_i)) \cup (\text{int}(\mathbf{T}_j) \setminus \text{con}(\mathbf{T}_j))$ , which is an independent set of  $G$  due to the definition of the arcs of  $\text{aux}(\mathcal{D}, F)$ . We obtain our contradiction, and the defined partition of  $V(G)$  is a monopolar partition for  $G$ . And with the condition  $C_i \subseteq F$  for every  $1 \leq i \leq s-1$ , we conclude that  $(C_1 \cup \dots \cup C_{s-1}, D_1 \cup \dots \cup D_{s-1})$  is a monopolar partition for  $G$  of the desired form.

For the converse, let  $(A, B)$  be a monopolar partition for  $G$  with  $A \subseteq F$ . Let  $D_1, \dots, D_r$  be the sets of vertices that induce the connected components of  $G[B]$ . Without loss of generality, we can assume that  $\min D_1 < \dots < \min D_r$ . With the properties of permutation diagrams, it follows that  $\min D_1 \leq \max D_1 < \dots < \min D_r \leq \max D_r$ . By definition of monopolar partition,  $D_1, \dots, D_r$  are cliques of  $G$ . For every  $1 \leq i \leq r$ , let  $X_i =_{\text{def}} \{\min D_i, \max D_i\}$  and let  $\mathbf{T}_i$  be the  $X_i$ -trapezoid in  $\mathcal{D}$ . By the properties of permutation diagrams,  $D_i \subseteq \text{con}(\mathbf{T}_i)$ . And since  $G[B]$  is the disjoint union of cliques, it follows with our assumption that  $\text{con}(\mathbf{T}_{i+1}) \subseteq \text{R}(\mathbf{T}_i)$  for every  $1 \leq i \leq r-1$ . First, we show that for every  $1 \leq i \leq r$ ,  $\mathbf{T}_i$  is good with respect to  $F$ . It

holds that  $1 \leq |X_i| \leq 2$ . Since  $B \cap \text{con}(\mathbf{T}_i) = B \cap \text{int}(\mathbf{T}_i) = D_i$ , it holds that  $\text{int}(\mathbf{T}_i) \setminus D_i \subseteq A$ . Hence,  $(\text{int}(\mathbf{T}_i) \setminus D_i, D_i)$  is a split partition for  $G[\text{int}(\mathbf{T}_i)]$  that is good. It follows that there is a vertex of  $\text{aux}(\mathcal{D}, F)$  that corresponds to  $\mathbf{T}_i$ . Let  $x_1, \dots, x_r$  be the vertices of  $\text{aux}(\mathcal{D}, F)$  that correspond to respectively  $\mathbf{T}_1, \dots, \mathbf{T}_r$ .

Second, we show that  $(x_i, x_{i+1})$  is an arc of  $\text{aux}(\mathcal{D}, F)$  for every  $1 \leq i \leq r-1$ . We have already seen that  $\text{con}(\mathbf{T}_{i+1}) \subseteq \text{R}(\mathbf{T}_i)$ . Since  $\text{int}(\mathbf{T}_i) \setminus \text{con}(\mathbf{T}_i) \subseteq A$  and  $\text{int}(\mathbf{T}_{i+1}) \setminus \text{con}(\mathbf{T}_{i+1}) \subseteq A$  due to the definition of  $X_i$  and  $X_{i+1}$ , it holds that  $(\text{int}(\mathbf{T}_i) \setminus \text{con}(\mathbf{T}_i)) \cup (\text{int}(\mathbf{T}_{i+1}) \setminus \text{con}(\mathbf{T}_{i+1}))$  is an independent set of  $G$ . Finally, suppose that there is  $x \in V(G)$  with  $x \in \text{R}(\mathbf{T}_i) \cap \text{L}(\mathbf{T}_{i+1})$ . This means that  $x \notin B$ , so that  $x \in A$ . Furthermore,  $x$  is not adjacent to any vertex in  $B$ , thus,  $x$  is an isolated vertex. This yields a contradiction to  $G$  being connected. We conclude that  $(x_i, x_{i+1})$  is an arc of  $\text{aux}(\mathcal{D}, F)$ .

Third, we show that  $(\mathbf{0}, x_1)$  and  $(\mathbf{1}, x_r)$  are arcs of  $\text{aux}(\mathcal{D}, F)$ . Similar to the previous arguments, if  $\text{L}(\mathbf{T}_1) \neq \emptyset$  or  $\text{R}(\mathbf{T}_r) \neq \emptyset$  then there is a vertex in  $A$  without a neighbour in  $B$ , so that  $G$  cannot be connected. We conclude that  $\text{aux}(\mathcal{D}, F)$  has a  $\mathbf{0}, \mathbf{1}$ -path. ■

Note that a  $\mathbf{0}, \mathbf{1}$ -path in  $\text{aux}(\mathcal{D}, F)$  does not correspond to a specific monopolar partition for  $G$  but can represent many partitions. The main reason is that a good split partition for a trapezoid is not unique, thus a vertex can belong to a clique and to the independent set in different monopolar partitions.

### 3.2 Running time

The main computational task is to construct the auxiliary digraph. We partition this task into listing the vertices and listing the arcs. For listing the vertices, the algorithm mainly needs to decide for a given trapezoid whether it is good. This is decided by checking the existence of a split partition that respects constraints. We want to decide this question in  $\mathcal{O}(n)$  time per trapezoid, which requires a careful analysis of the structure of possible split partitions.

Let  $G$  be a graph with at least one edge. The *split partition number* of  $G$  is defined as the smallest number  $k$  such that there are vertices  $u, v$  of  $G$  where  $d_G(u) = k$  and  $d_G(v) \leq k$  and  $uv \in E(G)$ . If  $G$  has a vertex of degree smaller than  $k$  then the largest degree  $k'$  with  $k' < k$  is called the *second split partition number* of  $G$ . Note that  $k'$  does not always exist. A split partition  $(C, D)$  for  $G$  is *clique-maximal* if there is no split partition  $(C', D')$  for  $G$  with  $D \subset D'$ .

**Lemma 3.2** *Let  $G$  be a split graph with at least one edge. Let  $k$  and  $k'$  be respectively the split partition number and the second split partition number of  $G$  (if the latter exists). Let  $(C, D)$  be a clique-maximal split partition. Then, one of the following holds:*

- $C = \{x \mid d_G(x) < k\}$  and  $D = \{x \mid d_G(x) \geq k\}$
- if  $k'$  exists:  $N_G(x) = N_G(x')$  and  $xx' \notin E(G)$  for all vertex pairs  $x, x'$  of  $G$  with  $d_G(x) = d_G(x') = k'$ , and  $D = N_G[y]$  for some vertex  $y$  with  $d_G(y) = k'$ .

**Proof.** We first show that all vertices of  $G$  with degree more than  $k$  belong to  $D$ . Let  $u, v$  be a vertex pair of  $G$  to witness the split partition number, i.e.,  $d_G(u) = k$  and  $d_G(v) \leq k$  and  $uv \in E(G)$ . Suppose that there is a vertex  $x$  of  $G$  with  $d_G(x) > k$  and  $x \in C$ . Since all neighbours of  $x$  are in  $D$ , it follows that  $|D| > k$  and all vertices in  $D$  with a neighbour in  $C$  have degree larger than  $k$ . Since  $u$  and  $v$  are adjacent, at most one of the two vertices can be in  $C$ . However, if one of the two vertices is in  $C$  then the degrees of  $u$  and  $v$  imply  $|D| \leq k$ , in contradiction to the above lower bound on  $|D|$ . Thus,  $u, v \in D$ . This also implies

$|D| = k + 1$  due to  $|D| \leq d_G(u) + 1$ . Since  $d_G(x) \geq k + 1$  and  $N_G(x) \subseteq D$ , it follows that  $x$  is a neighbour of  $u$ . This implies  $d_G(u) > k$ , which is a contradiction. Hence,  $x \in D$ , and therefore,  $\{x \mid d_G(x) > k\} \subseteq D$ .

Next, we consider vertices of degree smaller than  $k$ . Assume that there is a vertex  $y$  with  $d_G(y) < k$  and  $y \in D$ . Suppose that  $y$  has a neighbour  $z$  in  $C$ . Then,  $y$  has at most  $k - 2$  neighbours in  $D$ , which implies  $|D| \leq k - 1$ , and thus  $d_G(z) \leq k - 1$ . This, however, contradicts the definition of the split partition number of  $G$ . Hence, all neighbours of  $y$  are in  $D$ , and since  $D$  is a clique of  $G$ ,  $N_G[y] = D$ . Note that all vertices in  $C$  have degree at most  $d_G(y)$ , since none of these vertices is adjacent to  $y$ . Suppose there is another vertex  $y'$  in  $D$  without a neighbour in  $C$ . Then,  $d_G(y) = d_G(y')$  and  $yy' \in E(G)$ , contradicting the definition of the split partition number. Thus, all vertices in  $D \setminus \{y\}$  have degree more than  $d_G(y)$ . If there is a vertex  $z$  of  $G$  with  $d_G(z) = d_G(y)$  then  $z \in C$  by the above argument, and  $N_G(z) = N_G(y)$  due to the size of  $D$ . This particularly means that all vertices of degree  $d_G(y)$  form an independent set of  $G$ . Furthermore,  $d_G(y) = k'$ . This shows that  $(C, D)$  is a split partition for  $G$  of the second type.

As the complementary case, we assume that all vertices of degree smaller than  $k$  are contained in  $C$ . With the result of the first paragraph, we thus have  $\{x \mid d_G(x) < k\} \subseteq C$  and  $\{x \mid d_G(x) > k\} \subseteq D$ . We consider the vertices of degree exactly  $k$ . Let  $E =_{\text{def}} \{x \mid d_G(x) = k\}$ . If  $E \subseteq D$  then  $(C, D)$  is a split partition for  $G$  of the first type. Otherwise let  $E \not\subseteq D$ . We show that this leads to a contradiction. By  $E \not\subseteq D$ , there is a vertex  $y \in E \setminus D$ . And since some vertex in  $E$  is adjacent to a vertex of degree at most  $k$  due to the definition of  $k$ ,  $C \cup E$  is not an independent set of  $G$ , thus there is  $x \in E \cap D$ . Similar to the previous paragraph, if  $x$  has no neighbour in  $C$  then  $|D| = k + 1$ . Because of  $y \in C$ , all vertices in  $D \setminus \{x\}$  have a neighbour in  $C$  and thus degree at least  $k + 1$ . In particular, no vertex in  $E$  is adjacent to a vertex of degree at most  $k$ , which is a contradiction to the definition of the split partition number. Thus,  $x$  has a neighbour in  $C$ . Then,  $|D| \leq k$ , and  $|D| \geq k$  due to  $d_G(y) = k$ . This implies  $|D| = k$  and that  $x$  has  $y$  as the only neighbour in  $C$ , in particular,  $C$  contains only  $y$  as a vertex from  $E$ . It follows that  $(C \setminus \{y\}, D \cup \{y\})$  is a split partition for  $G$  with  $D \subset D \cup \{y\}$ , contradicting the assumption about  $(C, D)$  as being a clique-maximal split partition for  $G$ . This completes the proof. ■

The result of Lemma 3.2 implies an efficient algorithm for checking whether a trapezoid is good with respect to some set of vertices.

**Lemma 3.3** *There is an  $\mathcal{O}(n)$ -time algorithm that, given a permutation graph  $G$  with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$  and sets  $F, F' \subseteq V(G)$  and each vertex labelled with its degree in  $G$  and the split partition number  $k$  of  $G$  (if it exists), decides whether  $G$  has a split partition  $(A, B)$  where  $F' \subseteq A \subseteq F$ .*

**Proof.** Let  $G, F, F'$  and  $k$  be the input according to the lemma. If  $F'$  is not an independent set of  $G$  or if  $F' \not\subseteq F$  then the algorithm immediately rejects. So, let  $F'$  be an independent set and  $F' \subseteq F$ . We want to apply Lemma 3.2. Observe that if  $G$  has a split partition  $(A, B)$  with  $F' \subseteq A \subseteq F$  then  $G$  has a clique-maximal split partition  $(A', B')$  with  $A' \subseteq F$  and  $|B' \cap F'| \leq 1$ . Hence, for deciding the question it mainly suffices to consider clique-maximal split partitions. If  $G$  is edgeless then a desired split partition for  $G$  exists if and only if  $|V(G) \setminus F| \leq 1$ . If  $G$  is a complete graph then a desired split partition for  $G$  exists if and only if  $|F'| \leq 1$ . Using the degrees of the vertices, the two cases can be recognised and decided in  $\mathcal{O}(n)$  time. Henceforth, let  $G$  be neither edgeless nor complete. Let  $k'$  be the second split partition number of  $G$  (if it exists). We check the following conditions:



- 1)  $G$  is a split graph
- 2)  $G$  has a clique-maximal split partition  $(A, B)$  with  $A \subseteq F$ .

If  $G$  is a split graph then  $G$  has a split partition thus a clique-maximal split partition, and thus, one of the following vertex partitions is a split partition for  $G$ :

- $(\{x \mid d_G(x) < k\}, \{x \mid d_G(x) \geq k\})$
- if  $k'$  exists:  $(V(G) \setminus N_G[y], N_G[y])$  for some vertex  $y$  with  $d_G(y) = k'$ .

As vertex  $y$  in the second case, we choose  $y$  with  $y \notin F'$  if possible. The correctness of this check follows from the result of Lemma 3.2. In a second step, we check whether one of the two partitions yields a desired split partition. If  $k'$  does not exist then we have to consider only the first partition. Otherwise, if  $k'$  exists, and if  $y' \in F'$  for all vertices  $y'$  with  $d_G(y') = k'$  then the second partition does not yield a desired partition. We consider the first partition; denote it by  $(A_1, B_1)$ . Clearly, if  $A_1 \not\subseteq F$  then  $(A_1, B_1)$  does not yield a desired partition. If  $F' \subseteq A_1 \subseteq F$  then  $(A_1, B_1)$  is a desired partition. Finally, assume that  $B_1 \cap F' = \{x\}$ . If  $x$  has at least two neighbours in  $A_1$  then  $(A_1, B_1)$  does not yield a desired partition. If  $x$  has no neighbour in  $A_1$  then  $(A_1 \cup \{x\}, B_1 \setminus \{x\})$  is a desired partition. So, let  $x$  have exactly one neighbour in  $A_1$ , say  $z$ . Since  $d_G(x) \geq k$  and all vertices in  $A_1$  have degree smaller than  $k$ ,  $z$  is non-adjacent to some vertex in  $B_1$ , and thus exchanging  $x$  and  $z$  does not yield a split partition. We consider the second partition, denoted as  $(A_2, B_2)$ . This case can be rejected if  $y$  cannot be chosen with  $y \notin F'$ . Similar to the previous case, if  $B_2 \cap F' = \{x\}$  then we can accept if and only if  $x$  has at most one neighbour in  $A_2$  and the neighbour is adjacent to all vertices in  $B_2$ .

For the running time of the algorithm, we see that, using the degree every vertex is labelled with, the vertex partitions can be computed straightforward in  $\mathcal{O}(n)$  time. Also membership in  $F$  and  $F'$  is a simple table look-up for each vertex. We check whether a given set of vertices forms an independent set or a clique of  $G$ . Let  $S \subseteq V(G)$ . In  $\mathcal{O}(n)$  time, the vertices in  $S$  can be ordered by increasing upper endpoint in  $\mathcal{D}$ . This ordering arranges the vertices by increasing lower endpoints if and only if  $S$  forms an independent set, and the ordering arranges the vertices by decreasing endpoints if and only if  $S$  forms a clique (for an illustration of the independent set case, see also Figure 2). This follows from the properties of permutation diagrams. The two properties can be verified in  $\mathcal{O}(n)$  time. Scanning the degree labels,  $k'$  can be computed in  $\mathcal{O}(n)$  time from  $k$ . All other checks can be executed in  $\mathcal{O}(n)$  time. This completes the algorithm. ■

For constructing the auxiliary digraph, we split the task into two subtasks, one of which is listing the vertices. To decide whether a trapezoid represents a vertex we have to decide whether the trapezoid is good, and we can apply the result of Lemma 3.3 to solve this problem. To obtain a complete algorithm, it remains to compute the split partition number. This can easily be done in linear time for arbitrary graphs. But we want to be faster. A partial algorithmic solution is given in the next lemma.

**Lemma 3.4** *There is an  $\mathcal{O}(n)$  time algorithm that given a split graph  $G$  whose vertices are labelled with their degrees in  $G$ , computes the split partition number of  $G$ .*

**Proof.** If  $G$  is edgeless then all vertices have degree 0 and the split partition number of  $G$  is undefined. If  $G$  is complete and has at least two vertices then all vertices have degree  $|V(G)| - 1$  and the split partition number of  $G$  is  $|V(G)| - 1$ . This check requires  $\mathcal{O}(n)$  time. Henceforth,

assume that  $G$  is not edgeless and not complete. Using the degree sequence of  $G$ , a split partition  $(I, C)$  for  $G$  can be computed in  $\mathcal{O}(n)$  time (see, for instance, [18] and [20]). We observe, similar to the proof of Lemma 3.2, that all vertices in  $I$  have degree at most  $|C|$ , and all vertices in  $C$  with at least one neighbour in  $I$  have degree at least  $|C|$ . Let  $x$  be a vertex in  $C$  of smallest degree. Thus, if all vertices in  $C$  have a neighbour in  $I$  then the split partition number of  $G$  is equal to  $d_G(x)$ . Now, assume that  $x$  has no neighbour in  $I$ . It holds that  $d_G(x) = |C| - 1$  and all vertices in  $I$  have degree at most  $|C| - 1$ . Then, the split partition number of  $G$  is equal to  $\min d_G(C \setminus \{x\})$ . Note that  $C \setminus \{x\}$  is not empty, since otherwise  $G$  would be edgeless. This algorithm takes  $\mathcal{O}(n)$  time. ■

For permutation graphs, the split partition number can be computed in  $\mathcal{O}(n \log n)$  time: order the vertices by increasing degree, which defines a vertex ordering  $\sigma$ , and determine the leftmost vertex with a neighbour to its left. To decide the existence of a left neighbour, we assign to every vertex the number of vertices to its left in  $\sigma$  that have smaller upper endpoint in the permutation diagram and the number of vertices to its left in  $\sigma$  that have smaller lower endpoint in the permutation diagram. The two numbers are different for a vertex if and only if it has a neighbour to its left in  $\sigma$ . We leave it as an open problem whether the split partition number of a permutation graph with degree-labelled vertices can be computed in  $\mathcal{O}(n)$  time.

Now, we are ready for giving the main result and completing the main algorithm of this section.

**Theorem 3.5** *There is an  $\mathcal{O}(nm)$ -time algorithm that given a connected permutation graph  $G$  and a set  $F \subseteq V(G)$ , decides whether there is a monopolar partition  $(A, B)$  for  $G$  with  $A \subseteq F$ .*

**Proof.** The algorithm is as follows: on input  $G$  a connected permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$  and  $F \subseteq V(G)$ , construct the auxiliary digraph  $\text{aux}(\mathcal{D}, F)$  and check whether  $\text{aux}(\mathcal{D}, F)$  has a  $\mathbf{0}, \mathbf{1}$ -path; accept if a  $\mathbf{0}, \mathbf{1}$ -path exists, otherwise reject. Due to Lemma 3.1, the algorithm accepts if and only if a desired monopolar partition for  $G$  exists. It remains to consider the running time of the algorithm, which is determined by the two tasks: constructing  $\text{aux}(\mathcal{D}, F)$  and finding a  $\mathbf{0}, \mathbf{1}$ -path.

We begin with the construction of  $\text{aux}(\mathcal{D}, F)$ . We first list the vertices of  $\text{aux}(\mathcal{D}, F)$  and then we list the arcs. Let  $G$  have  $n$  vertices and  $m$  edges. In linear time, we can label every vertex of  $G$  with its degree in  $G$ . By definition,  $\text{aux}(\mathcal{D}, F)$  has at most  $n + m$  vertices. Let  $X \subseteq V(G)$  with  $|X| = 1$  and let  $\mathbf{T}$  be the  $X$ -trapezoid. Note that  $X = \text{con}(\mathbf{T})$ . It can be tested in  $\mathcal{O}(n)$  time whether  $\text{int}(\mathbf{T}) \setminus X$  is an independent set of  $G$ , and thus, it can be checked in  $\mathcal{O}(n)$  time whether  $\mathbf{T}$  is good. Now, let  $X \subseteq V(G)$  with  $|X| = 2$ , and let  $\mathbf{T}$  be the  $X$ -trapezoid. To determine whether  $\mathbf{T}$  is good, we have to decide for the vertices in  $\text{con}(\mathbf{T})$  which ones belong to the independent set and which ones to the clique. We compute the subgraph of  $G$  that is induced by  $\text{int}(\mathbf{T})$  and assign to every vertex its degree in  $G[\text{int}(\mathbf{T})]$ . For the vertices in  $\text{con}(\mathbf{T})$ , the degree in  $G[\text{int}(\mathbf{T})]$  is equal to the degree in  $G$ . It remains to determine the degrees of the vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ ; these are the vertices with neighbours in  $G$  that are not contained in  $\text{int}(\mathbf{T})$ . If  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  is not an independent set in  $G$ , then  $\mathbf{T}$  cannot be a good trapezoid due to the definition of good trapezoid. So, let  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  be an independent set. By the properties of permutation diagrams, the upper endpoint ordering is equal to the lower endpoint ordering of these vertices; let this ordering be  $\sigma$ . Let  $x$  be a vertex in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ . It holds that  $d_{G[\text{int}(\mathbf{T})]}(x) = d_G(x) - |\text{L}(\mathbf{T}) \cap N_G(x)| - |\text{R}(\mathbf{T}) \cap N_G(x)|$ . We show that  $|\text{L}(\mathbf{T}) \cap N_G(x)|$  and  $|\text{R}(\mathbf{T}) \cap N_G(x)|$  can be computed in amortised constant time per vertex. If  $x < \min X$  then

the upper endpoint of  $x$  is to the left of  $\mathbf{T}$ , and  $|\mathbf{L}(\mathbf{T}) \cap N_G(x)|$  is equal to  $|x - \min X| - 1$  minus the number of vertices in  $\sigma$  to the right of  $x$  and with upper endpoint to the left of  $\mathbf{T}$ . If  $\pi^{-1}(x) < \pi^{-1}(\max X)$  then the lower endpoint of  $x$  is to the left of  $\mathbf{T}$ , and  $|\mathbf{L}(\mathbf{T}) \cap N_G(x)|$  is equal to  $|\pi^{-1}(x) - \pi^{-1}(\max X)| - 1$  minus the number of vertices in  $\sigma$  to the right of  $x$  and with lower endpoint to the left of  $\mathbf{T}$ . If none of the two cases holds then  $|\mathbf{L}(\mathbf{T}) \cap N_G(x)| = 0$ . For  $|\mathbf{R}(\mathbf{T}) \cap N_G(x)|$ , analogue equivalences hold. Since these numbers can be computed in overall  $\mathcal{O}(n)$  time for all vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ , we conclude that we can compute a permutation diagram for  $G[\text{int}(\mathbf{T})]$  and assign the degrees to the vertices in  $\mathcal{O}(n)$  time. As described in the proof of Lemma 3.4, we can check in  $\mathcal{O}(n)$  time whether  $G[\text{int}(\mathbf{T})]$  is a split graph. If no then  $\mathbf{T}$  is not good. If yes then we compute the split partition number of  $G[\text{int}(\mathbf{T})]$  using the algorithm of Lemma 3.4 and then we check whether  $G[\text{int}(\mathbf{T})]$  has a split partition  $(C, D)$  with  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T}) \subseteq C \subseteq F \cap \text{int}(\mathbf{T})$  by applying the algorithm of Lemma 3.3. Hence, in  $\mathcal{O}(n)$  time, we can decide whether  $\mathbf{T}$  is a good trapezoid with respect to  $F$ . Summarising, the set of vertices of  $\text{aux}(\mathcal{D}, F)$  and the corresponding good trapezoids can be determined in time  $\mathcal{O}(nm)$ .

Now, we determine the arcs of  $\text{aux}(\mathcal{D}, F)$ . We first give an algorithm for deciding whether there is an arc between two given vertices. Let  $u$  and  $v$  be vertices of  $\text{aux}(\mathcal{D}, F)$  and let  $\mathbf{T}_u$  and  $\mathbf{T}_v$  be the corresponding good  $X_u$ - and  $X_v$ -trapezoids. We decide whether  $(u, v)$  is an arc of  $\text{aux}(\mathcal{D}, F)$ . According to the definition, we have to check three conditions. The condition  $\text{con}(\mathbf{T}_v) \subseteq \mathbf{R}(\mathbf{T}_u)$  can be checked in constant time by verifying that  $\max X_u < \min X_v$  and  $\pi^{-1}(\min X_u) < \pi^{-1}(\max X_v)$ . For the second condition, we check whether there is a vertex in  $\mathbf{R}(\mathbf{T}_u) \cap \mathbf{L}(\mathbf{T}_v)$ . Let  $Y_u$  be the set of vertices in  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)$  that have an endpoint to the right of  $\mathbf{T}_u$ , and let  $Y_v$  be the set of vertices in  $\text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v)$  that have an endpoint to the left of  $\mathbf{T}_v$ . Note that both sets contain only vertices from the independent set of the split partitions for  $\text{int}(\mathbf{T}_u)$  and  $\text{int}(\mathbf{T}_v)$ . We apply the following equivalence:

$$2|\mathbf{R}(\mathbf{T}_u) \cap \mathbf{L}(\mathbf{T}_v)| = |\max X_u - \min X_v| + |\pi^{-1}(\min X_u) - \pi^{-1}(\max X_v)| - |Y_u \Delta Y_v| - 2.$$

The main argument for correctness of the equivalence is the fact that vertices with exactly one endpoint between  $\mathbf{T}_u$  and  $\mathbf{T}_v$  have to intersect exactly one of the two trapezoids. The running time for evaluating the equation's right hand side is mainly determined by computing the cardinality of the symmetric difference  $Y_u \Delta Y_v$ . Without pre-processing, the computation of this value requires  $\mathcal{O}(n)$  time. To make this step faster, we have to apply properties of independent sets. It holds that  $|Y_u \Delta Y_v| = |Y_u| + |Y_v| - 2|Y_u \cap Y_v|$ . The values  $|Y_u|$  and  $|Y_v|$  are constants for each trapezoid, whereas  $|Y_u \cap Y_v|$  is not. Exactly one of the two cases holds:

- all vertices in  $Y_u$  have their upper endpoint to the right of  $\mathbf{T}_u$  in  $\mathcal{D}$
- all vertices in  $Y_u$  have their lower endpoint to the right of  $\mathbf{T}_u$  in  $\mathcal{D}$ .

For a contradiction, if a vertex has its upper endpoint to the right of  $\mathbf{T}_u$  and another vertex has its lower endpoint to the right of  $\mathbf{T}_u$ , the two vertices are adjacent by the properties of permutation diagrams and the fact that both are intersected by  $\mathbf{T}_u$ . Applying the above fact, it follows that  $|Y_u \cap Y_v|$  is equal to the number of vertices in  $Y_u$  whose upper endpoint is larger than  $\min X_v$  or whose lower endpoint is larger than  $\pi^{-1}(\max X_v)$ . To achieve efficient checking of the second condition, we run a pre-processing for every good trapezoid  $\mathbf{T}$ , i.e., for every vertex of  $\text{aux}(\mathcal{D}, F)$ . Firstly, we compute the set of vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  with an endpoint to the left of  $\mathbf{T}$  and the similar set of vertices with an endpoint to the right of  $\mathbf{T}$  and the cardinalities of the two sets. Secondly, we construct an array with the information about how many vertices

in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  have their upper/lower endpoint to the left or right of a given value. This array can be computed in a single sweep through the two computed subsets of  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ , that can be assumed ordered by their upper or lower endpoints. The described pre-processing requires  $\mathcal{O}(n)$  time per trapezoid, which makes  $\mathcal{O}(nm)$  time in total. Then, the second condition can be checked in constant time.

Finally, for determining whether  $(u, v)$  is an arc of  $\text{aux}(\mathcal{D}, F)$ , we have to check the third condition. This condition can be checked in several ways. If  $Y_u \cap Y_v$  is non-empty then  $(\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)) \cup (\text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v))$  is an independent set by the properties of permutation diagrams. Suppose that  $Y_u \cap Y_v$  is empty. Since  $G$  is connected,  $Y_u \cup Y_v$  is not an independent set or  $\text{R}(\mathbf{T}_u) \cap \text{L}(\mathbf{T}_v)$  is not empty (to establish a path from a vertex in  $Y_u$  to a vertex in  $Y_v$ ). Hence, if  $(u, v)$  is an arc of  $\text{aux}(\mathcal{D}, F)$  then  $Y_u \cap Y_v$  is non-empty. And non-emptiness can be checked by using the information computed during the pre-processing, or we store the largest vertex in  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)$  and compare its endpoints with  $\min X_v$  and  $\pi^{-1}(\max X_v)$ .

Another possibility for checking the third condition is to check that the largest vertex in  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)$  is non-adjacent to the smallest vertex in  $\text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v)$ . Sufficiency of this check follows from the properties of independent sets in permutation diagrams. The smallest and largest vertex can be computed in overall  $\mathcal{O}(nm)$  time for all good trapezoids. We conclude that, with an  $\mathcal{O}(nm)$ -time pre-proceeding, it takes constant time for checking for a specific arc between two given vertices of  $\text{aux}(\mathcal{D}, F)$ .

For completing the construction algorithm for  $\text{aux}(\mathcal{D}, F)$ , we show that there are at most  $\mathcal{O}(nm)$  arcs in  $\text{aux}(\mathcal{D}, F)$  and that the candidate pairs for being connected by an arc can be listed in  $\mathcal{O}(nm)$  time. Let  $(u, v)$  be an arc of  $\text{aux}(\mathcal{D}, F)$  for  $u \neq \mathbf{0}$ . Let  $\mathbf{T}_u$  and  $X_u$  be defined as above. If  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u) = \emptyset$  then  $V(G) = \text{con}(\mathbf{T}_u)$  and  $(u, \mathbf{1})$  is an arc of  $\text{aux}(\mathcal{D}, F)$ . Now, let  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u) \neq \emptyset$ . Let  $y$  be the largest vertex in  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)$ . Due to the properties of permutation diagrams,  $y$  is the vertex in  $\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)$  that is adjacent to the largest set of vertices in  $\text{R}(\mathbf{T}_u)$ . By definition of polar partitions, all neighbours of  $y$  are clearly elements of the disjoint union of cliques. If  $y$  has no neighbour in  $\text{R}(\mathbf{T}_u)$  then  $\text{R}(\mathbf{T}_u) = \emptyset$  due to the connectedness of  $G$  and  $v = \mathbf{1}$ . Let  $y$  have neighbours in  $\text{R}(\mathbf{T}_u)$ . Let  $z$  and  $z'$  be the neighbours of  $y$  in  $\text{R}(\mathbf{T}_u)$  with smallest respectively upper and lower endpoint. Note that  $z = z'$  or  $zz' \in E(G)$ . Let  $\mathbf{T}_v$  and  $X_v$  be defined as above; in particular, they exist due to the existence of  $z$  and  $z'$ . We show that  $z \in X_v$  or  $z' \in X_v$ . We have to consider two cases according to whether the upper endpoint of  $y$  is larger than  $\max X_u$  or the lower endpoint of  $y$  is larger than  $\pi^{-1}(\min X_u)$ . By a symmetry argument, it suffices to argue for the latter case. As a first observation note that  $z, z' \in \text{int}(\mathbf{T}_v)$ . This is the case, since  $(u, v)$  being an arc of  $\text{aux}(\mathcal{D}, F)$  requires  $\text{R}(\mathbf{T}_u) \cap \text{L}(\mathbf{T}_v)$  to be empty, and since  $X_v \subseteq \text{con}(\mathbf{T}_v)$  is non-empty,  $z \in \text{R}(\mathbf{T}_v)$  or  $z' \in \text{R}(\mathbf{T}_v)$  yields a contradiction to the choice of  $z$  and  $z'$ . As a second observation note that  $z, z' \notin \text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v)$ , since  $(u, v)$  being an arc of  $\text{aux}(\mathcal{D}, F)$  requires  $(\text{int}(\mathbf{T}_u) \setminus \text{con}(\mathbf{T}_u)) \cup (\text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v))$  to be an independent set of  $G$ . Remember that  $y \in \text{int}(\mathbf{T}_v) \setminus \text{con}(\mathbf{T}_v)$ . Thus,  $z, z' \in \text{con}(\mathbf{T}_v)$ . Let  $(C, D)$  be an arbitrary good split partition for  $\text{int}(\mathbf{T}_v)$ . Clearly,  $z, z' \in D$ . Let  $x$  be an arbitrary vertex in  $D$ . Since  $D$  is a clique of  $G$ ,  $x$  is adjacent to  $z$  and  $z'$ . If  $\pi^{-1}(x) < \pi^{-1}(z')$  then  $x$  is a neighbour of  $y$  and contradicts the choice of  $z'$ . Thus,  $\pi^{-1}(z') < \pi^{-1}(x)$ . Since  $z' \in \text{con}(\mathbf{T}_v)$ , it follows that  $z' \in X_v$ . And since  $X_v$  contains at most two vertices, there are at most  $n$  possibilities for a second vertex in  $X_v$ . We conclude that  $u$  can have at most  $n + 1$  out-neighbours, and the candidates (encoded by the sets  $X_v$ ) can be listed in  $\mathcal{O}(n)$  time, since  $y, z, z'$  for  $\mathbf{T}_u$  can be determined in  $\mathcal{O}(n)$  time.

We sum up the running time. In  $\mathcal{O}(nm)$  time, the vertices of  $\text{aux}(\mathcal{D}, F)$  can be listed and

corresponding good split partitions can be computed. In  $\mathcal{O}(nm)$  time, an arc candidate list can be obtained, and for determining the arc set of  $\text{aux}(\mathcal{D}, F)$  it remains to check for every arc candidate whether it forms an arc of  $\text{aux}(\mathcal{D}, F)$ . This can be done in constant time per arc candidate, after the  $\mathcal{O}(n)$  time pre-processing per vertex of  $\text{aux}(\mathcal{D}, F)$ . Thus,  $\text{aux}(\mathcal{D}, F)$  can be constructed in  $\mathcal{O}(nm)$  time. The arcs of  $\text{aux}(\mathcal{D}, F)$  of the form  $(\mathbf{0}, v)$  can be determined straightforward. Checking the existence of a  $\mathbf{0}, \mathbf{1}$ -path takes linear time in  $\text{aux}(\mathcal{D}, F)$ . Since  $\text{aux}(\mathcal{D}, F)$  contains  $\mathcal{O}(m)$  vertices and  $\mathcal{O}(nm)$  edges, such a path can be found in  $\mathcal{O}(nm)$  time. This concludes the proof. ■

Note that it is easy to also obtain a monopolar partition for  $G$  from the discovered path in the auxiliary digraph. This is described in the proof of Lemma 3.1.

For disconnected input graphs, we run the algorithm of Theorem 3.5 on every connected component and combine the partial solutions to a global solution. Computing the connected components of a graph can be done in linear time.

## 4 Recognising polar permutation graphs

The idea of the algorithm of this section is: delete a set of vertices that induces a monopolar graph in the complement of the input graph and obtain a monopolar graph. The existence of such a set of vertices is easy to decide from a given polar partition. However, finding such a set of vertices is not easy in general. The major part of this section is dedicated to identifying possible candidates and showing that there are only few of them and that they can be listed efficiently.

For identifying candidate sets of vertices, we use trapezoids. Recall the definitions from the beginning of Subsection 3.1. Let  $G$  be a permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . Let  $G$  have a polar partition  $(A, B)$  and let  $Y \subseteq V(G)$ . A trapezoid  $\mathbf{T}$  in  $\mathcal{D}$  is called *centre trapezoid for  $(A, B)$  in  $\mathcal{D}$  (with  $Y$  in-cliqued)* if the following conditions are satisfied:

- 1)  $G[\text{int}(\mathbf{T})]$  has a polar partition  $(A', C)$  with
  - $C$  is a clique of  $G$  (and  $Y \subseteq C$ ) and  $C \subseteq \text{con}(\mathbf{T})$
  - $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T}) \subseteq A' \subseteq A$
- 2) one of the following two cases holds:
  - $A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T}))$  is an independent set of  $G$
  - $A \cap \text{L}(\mathbf{T}) \neq \emptyset$  and  $A \cap \text{R}(\mathbf{T}) = \emptyset$  and there is a vertex  $v \in \text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  with  $N_G(v) \cap \text{L}(\mathbf{T}) \subseteq A$  such that  $\{v\} \cup (A \cap \text{L}(\mathbf{T})) \setminus N_G(v)$  is an independent set of  $G$ .

Note that condition 1 is similar to *good trapezoid* as defined in Subsection 3.1. We show that such centre trapezoids indeed exist. A graph that is polar but not monopolar is called *multipolar*. In particular, vertices in the  $A$ -set of a polar partition can be adjacent.

**Lemma 4.1** *Let  $G$  be a permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . Let  $G$  be multipolar and  $(A, B)$  a  $B$ -maximal polar partition for  $G$ . There are a trapezoid  $\mathbf{T}$  in  $\mathcal{D}$  and a clique  $X$  of  $G$  with  $X \subseteq B$  and  $1 \leq |X| \leq 2$  such that  $\mathbf{T}$  is the  $X$ -trapezoid in  $\mathcal{D}$  and  $\mathbf{T}$  is a centre trapezoid for  $(A, B)$  in  $\mathcal{D}$  with  $X$  in-cliqued.*

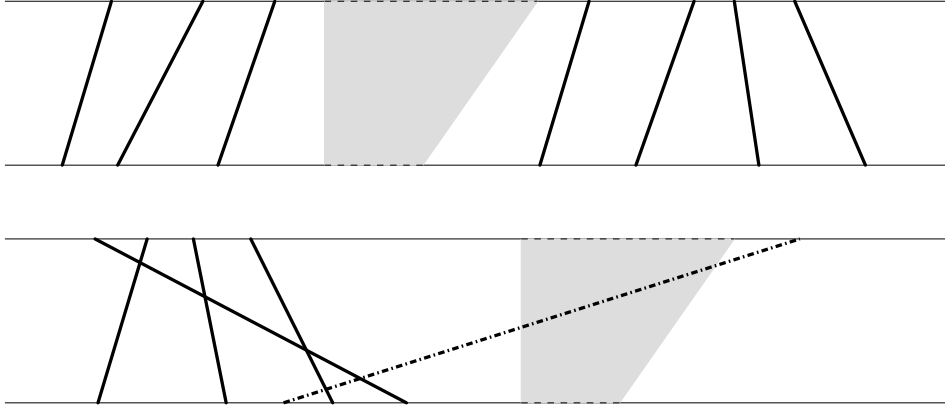


Figure 3: The figure illustrates the two cases in the second condition of the definition of a centre trapezoid. The trapezoid itself is represented by the grey area. The thick full line segments represent vertices from set  $A$ . In the above situation, the vertices from  $A$  on the left and right side form an independent set, which corresponds to the first case. The below situation illustrates aspects of the second case. Vertex  $v$ , depicted as the broken line segment, is adjacent to some vertices from  $A$  on the left side of the trapezoid and forms an independent set with the non-adjacent vertices.

**Proof.** By Lemma 2.1, every vertex in  $A$  has a neighbour in  $B$ . Let the connected components of  $G[B]$  be induced by the sets  $C_1, \dots, C_r$ ; without loss of generality we can assume that  $\min C_1 < \dots < \min C_r$ . For every  $x \in A$ , denote by  $\alpha(x)$  and  $\omega(x)$  the respectively smallest and largest index  $i$  with  $C_i$  containing a neighbour of  $x$ .

As the first case, assume that there are adjacent vertices  $u, v \in A$  with  $\omega(u) < \alpha(v)$ . Note that  $uv \in E(G)$  implies  $\alpha(v) = \omega(u) + 1$ . Let  $C =_{\text{def}} C_{\alpha(v)}$  and let  $\mathbf{T}$  be the  $C$ -trapezoid in  $\mathcal{D}$ . Note that  $u \in L(\mathbf{T})$  and therefore  $v \in \text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  by the properties of permutation diagrams. Furthermore,  $L(\mathbf{T}) \cap N_G(v) \subseteq A$  by the definition of  $\alpha(v)$  and  $C \subseteq \text{con}(\mathbf{T})$ . For the first subcase, assume that  $A \subseteq L(\mathbf{T}) \cup \text{int}(\mathbf{T})$ , i.e.,  $A \cap R(\mathbf{T}) = \emptyset$ . A vertex in  $A \cap L(\mathbf{T})$  is either adjacent or non-adjacent to  $v$ . By the definition of complete multipartite graphs, the vertices in  $A \cap L(\mathbf{T})$  that are non-adjacent to  $v$  are in the same maximal independent set as  $v$  in  $G[A]$ . Hence, they are pairwise non-adjacent, which means that  $\{v\} \cup (A \cap L(\mathbf{T})) \setminus N_G(v)$  is an independent set in  $G$ . For the second subcase, let  $A \cap R(\mathbf{T})$  be non-empty. This means that both  $A \cap L(\mathbf{T})$  and  $A \cap R(\mathbf{T})$  are non-empty. Since vertices from  $L(\mathbf{T})$  and  $R(\mathbf{T})$  are pairwise non-adjacent by the properties of permutation diagrams, the definition of complete multipartite graphs implies that  $A \cap (L(\mathbf{T}) \cup R(\mathbf{T}))$  is an independent set in  $G$ . Hence, in both subcases,  $\mathbf{T}$  satisfies the second condition of the definition of centre trapezoid.

As the second case, assume that for all pairs  $u, v \in A$  of adjacent vertices,  $\alpha(u) \leq \alpha(v) \leq \omega(u)$  or  $\alpha(v) \leq \alpha(u) \leq \omega(v)$ . The following construction and argumentation is more difficult than it would be expected, since the condition of the second case does not compare all pairs of vertices from  $A$  but only adjacent vertices. Let  $u, v \in A$  be a pair of adjacent vertices such that the intersection  $\{\alpha(u), \dots, \omega(u)\} \cap \{\alpha(v), \dots, \omega(v)\}$  is of smallest size. Without loss of generality, we can assume that  $\alpha(u) \leq \alpha(v)$ . Let  $C =_{\text{def}} C_{\alpha(v)}$  and let  $\mathbf{T}$  be the  $C$ -trapezoid in  $\mathcal{D}$ . For the first subcase, let  $\omega(v) \leq \omega(u)$ . Then,  $\alpha(u) \leq \alpha(v) \leq \omega(v) \leq \omega(u)$  with our assumptions, which implies  $\alpha(x) \leq \alpha(v) \leq \omega(v) \leq \omega(x)$  for all vertices  $x \in A \cap N_G(v)$ . This follows from the choice of the pair  $u, v$  as of smallest intersection size. Hence,  $A \cap N_G(v) \subseteq \text{int}(\mathbf{T})$ , and

by the properties of complete multipartite graphs,  $A \cap (\mathbf{L}(\mathbf{T}) \cup \mathbf{R}(\mathbf{T}))$  is an independent set of  $G$ . For the second subcase, let  $\omega(u) < \omega(v)$ . For every vertex  $x \in A \cap \mathbf{L}(\mathbf{T})$ ,  $\omega(x) < \alpha(v)$ , which means by the assumptions of the case that  $(A \cap \mathbf{L}(\mathbf{T})) \cap N_G(v) = \emptyset$ . Hence, due to the properties of complete multipartite graphs,  $\{v\} \cup (A \cap \mathbf{L}(\mathbf{T}))$  is an independent set of  $G$ . In the following, we distinguish between the cases  $A \cap \mathbf{L}(\mathbf{T})$  non-empty and empty. First, let  $A \cap \mathbf{L}(\mathbf{T})$  be non-empty. By the properties of permutation diagrams, the vertices in  $A \cap \mathbf{L}(\mathbf{T})$  and  $A \cap \mathbf{R}(\mathbf{T})$  are pairwise non-adjacent. Then, the properties of complete multipartite graphs imply that  $\{v\} \cup (A \cap \mathbf{L}(\mathbf{T})) \cup (A \cap \mathbf{R}(\mathbf{T}))$  is an independent set of  $G$ . Second, let  $A \cap \mathbf{L}(\mathbf{T})$  be empty. Let  $A \cap \mathbf{R}(\mathbf{T})$  be non-empty and let  $w \in A \cap \mathbf{R}(\mathbf{T})$ . Since  $u \in \text{int}(\mathbf{T})$ ,  $w \neq u$ . Suppose that  $uw \in E(G)$ . Then,  $\alpha(v) < \alpha(w)$  and  $\alpha(v) \leq \omega(u) < \omega(v)$  yield a contradiction to the intersection size of  $u, v$ . Hence,  $u$  and  $w$  are non-adjacent, which implies that  $\{u\} \cup (A \cap \mathbf{R}(\mathbf{T})) = \{u\} \cup (A \cap \mathbf{L}(\mathbf{T})) \cup (A \cap \mathbf{R}(\mathbf{T}))$  is an independent set of  $G$ . Hence, in both subcases,  $A \cap (\mathbf{L}(\mathbf{T}) \cup \mathbf{R}(\mathbf{T}))$  is an independent set of  $G$ , which shows that  $\mathbf{T}$  satisfies the second condition of the definition of centre trapezoids.

It remains to check whether the chosen trapezoids satisfy the first condition of the definition of centre trapezoids. We can consider the two cases above simultaneously. Let  $C$  and  $\mathbf{T}$  be as defined above. First, we show that  $(\text{int}(\mathbf{T}) \setminus C, C)$  is a polar partition for  $G[\text{int}(\mathbf{T})]$ . Since no edge of  $G$  joins vertices from different cliques among  $C_1, \dots, C_r$ ,  $B \cap \text{int}(\mathbf{T}) = C$ . Hence,  $\text{int}(\mathbf{T}) \setminus B = \text{int}(\mathbf{T}) \setminus C \subseteq A$ . By the definition of  $C$ -trapezoid,  $C \subseteq \text{con}(\mathbf{T})$ , which also implies that  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T}) \subseteq \text{int}(\mathbf{T}) \setminus C \subseteq A$ . Hence,  $\mathbf{T}$  satisfies the first condition of the definition of centre trapezoids. To complete the proof of the lemma, let  $x =_{\text{def}} \min C$  and  $y =_{\text{def}} \max C$ . Note that  $x = y$  in case  $|C| = 1$ . Then, with the properties of permutation diagrams and the representation of cliques, it holds that  $\mathbf{T}$  is the  $\{x, y\}$ -trapezoid and a centre trapezoid for  $(A, B)$  in  $\mathcal{D}$  with  $\{x, y\}$  in-cliqued. ■

Lemma 4.1 is the main tool of our algorithm. Informally, the algorithm removes a trapezoid from the graph and checks whether the remaining subgraph is monopolar. The main problem then is to combine the independent set of the monopolar partition with the complete multipartite graph of the polar partition of the removed trapezoid. Not every monopolar partition is suitable. The next lemma will be useful for choosing a suitable monopolar partition. Let  $G$  be a permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . Let  $\mathbf{T} = (I_1, I_2)$  be a trapezoid in  $\mathcal{D}$  with  $I_1 = \{i_1, \dots, i'_1\}$  and  $I_2 = \{i_2, \dots, i'_2\}$  and let  $S \subseteq \text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ . A vertex  $x \in S$  is *left-endpoint close to  $\mathbf{T}$  among the vertices in  $S$*  if  $x < i_1$  and  $x \geq y$  for all vertices  $y \in S$  or if  $\pi^{-1}(x) < i_2$  and  $\pi^{-1}(x) \geq \pi^{-1}(y)$  for all vertices  $y \in S$ . *Right-endpoint close vertex* is defined symmetrically. Note that there are at most two left-endpoint close and at most two right-endpoint close vertices for every trapezoid and chosen set.

**Lemma 4.2** *Let  $G$  be a permutation graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . Let  $G$  be multipolar and  $(A, B)$  a  $B$ -maximal polar partition for  $G$ . Let  $\mathbf{T}$  be a centre trapezoid for  $(A, B)$  in  $\mathcal{D}$ .*

- 1) *Let  $A \cap (\mathbf{L}(\mathbf{T}) \cup \mathbf{R}(\mathbf{T}))$  be an independent set of  $G$  and let  $A \cap \text{con}(\mathbf{T})$  be non-empty. Let  $x$  be an arbitrary vertex in  $A \cap \text{con}(\mathbf{T})$ . Then,  $\{x\} \cup (A \cap (\mathbf{L}(\mathbf{T}) \cup \mathbf{R}(\mathbf{T})))$  is an independent set of  $G$ .*
- 2) *Let  $A \cap (\mathbf{L}(\mathbf{T}) \cup \mathbf{R}(\mathbf{T}))$  be an independent set of  $G$  and let  $A \cap \text{con}(\mathbf{T})$  be empty. If there is  $x \in A \cap \text{int}(\mathbf{T})$  such that  $\{x\} \cup (A \cap (\mathbf{L}(\mathbf{T}) \cup \mathbf{R}(\mathbf{T})))$  is an independent set in  $G$*

then  $x$  can be chosen as left-endpoint close or right-endpoint close to  $\mathbf{T}$  among the vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ .

3) Let  $A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T}))$  not be an independent set of  $G$ .

Let  $A \cap \text{L}(\mathbf{T}) \neq \emptyset$  and  $A \cap \text{R}(\mathbf{T}) = \emptyset$  and let there be a vertex  $x \in \text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  with  $N_G(x) \cap \text{L}(\mathbf{T}) \subseteq A$  such that  $(A \cap \text{L}(\mathbf{T})) \setminus N_G(x)$  is an independent set of  $G$ . Then,  $x$  can be chosen as left-endpoint close to  $\mathbf{T}$  among the vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ .

**Proof.** We consider the three cases separately. For the first two cases, let  $A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T}))$  be an independent set of  $G$ . If  $A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T}))$  is empty then the two cases trivially hold. So, let  $A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T}))$  be non-empty. For the first case, let there be a vertex  $x \in A \cap \text{con}(\mathbf{T})$ . Then,  $x$  is not adjacent to any vertex in  $\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T})$  and thus  $\{x\} \cup (A \setminus \text{int}(\mathbf{T}))$  is an independent set of  $G$  due to the properties of complete multipartite graphs.

For the second case, let  $A \cap \text{con}(\mathbf{T}) = \emptyset$ . Assume that there is  $x \in A \cap \text{int}(\mathbf{T})$  such that  $\{x\} \cup (A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T})))$  is an independent set of  $G$ . By a symmetry argument for permutation diagrams, we can assume that  $A \cap \text{L}(\mathbf{T}) \neq \emptyset$  and  $x$  is smaller than the vertices in  $\text{con}(\mathbf{T})$ . The three other cases are obtained from flipping the permutation diagram vertically or horizontally. Informally spoken, the upper endpoint of  $x$  in  $\mathcal{D}$  is to the left of  $\mathbf{T}$ . By assumption, all vertices in  $A \cap \text{L}(\mathbf{T})$  are non-adjacent to  $x$ . Let  $y$  be the left-endpoint close vertex for  $\mathbf{T}$  among the vertices in  $\text{int}(\mathbf{T})$  with the upper endpoint of  $y$  to the left of  $\mathbf{T}$  in  $\mathcal{D}$ . Assume that  $x \neq y$ . Note that  $x < y$ . No vertex from  $A \cap \text{L}(\mathbf{T})$  is adjacent to  $y$ , since otherwise such a vertex would be adjacent also to  $x$  due to the properties of permutation diagrams. Thus,  $\{y\} \cup (A \cap \text{L}(\mathbf{T}))$  is an independent set of  $G$ . And with the properties of complete multipartite graphs,  $\{y\} \cup (A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T})))$  is an independent set of  $G$ .

For the third case, let  $A \cap (\text{L}(\mathbf{T}) \cup \text{R}(\mathbf{T}))$  not be an independent set of  $G$ . By assumption about  $\mathbf{T}$ ,  $A \cap \text{L}(\mathbf{T}) \neq \emptyset$  and  $A \cap \text{R}(\mathbf{T}) = \emptyset$  and there is a vertex  $x$  that satisfies the assumptions of the case. By a symmetry argument for permutation diagrams, we can assume that  $x$  is smaller than the vertices in  $\text{con}(\mathbf{T})$ , i.e., the upper endpoint of  $x$  is to the left of  $\mathbf{T}$  in  $\mathcal{D}$ . Let  $y$  be the left-endpoint close vertex for  $\mathbf{T}$  among the vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$  with the upper endpoint of  $y$  to the left of  $\mathbf{T}$  in  $\mathcal{D}$ . Assume that  $y \neq x$ . With the properties of permutation diagrams and the assumptions about  $x$ ,  $N_G(y) \cap \text{L}(\mathbf{T}) \subseteq N_G(x) \cap \text{L}(\mathbf{T}) \subseteq A$ . We show that  $(A \cap \text{L}(\mathbf{T})) \setminus N_G(y)$  is an independent set of  $G$ . If there is a pair  $u, v \in (A \cap \text{L}(\mathbf{T})) \setminus N_G(y)$  of adjacent vertices then the properties of complete multipartite graphs imply that at least one of them is adjacent to  $y$ , which is a contradiction. ■

We are ready to give the algorithm for recognising connected polar permutation graphs. The algorithm is called **Polar-Permutation-Graphs-Recognition**, and it is given in Figure 4. If the input graph is polar, the algorithm outputs a polar partition, thus provides a certificate.

**Theorem 4.3** *Algorithm Polar-Permutation-Graphs-Recognition recognises connected polar permutation graphs in  $\mathcal{O}(nm^2)$  time.*

**Proof.** For the correctness of the algorithm, let  $G$  be the input graph with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ . We first show that every ‘yes’ answer (in lines 2, 14, 20, 28, 34) is correct and the output partition is a proper polar partition for  $G$ . So, let the answer of the algorithm on input  $G, \pi, \mathcal{D}$  be ‘yes’. It is a simple check that the output vertex partition is indeed a partition of  $V(G)$ . If the answer is output in line 2 then  $G$  is monopolar, thus polar, and the output partition is a polar partition for  $G$ . We consider the four



**Algorithm Polar-Permutation-Graphs-Recognition****Input** connected permutation graph  $G$ with permutation sequence  $\pi$  and corresponding permutation diagram  $\mathcal{D}$ **Output** answer ‘yes’ or ‘no’, and if ‘yes’ then a polar partition for  $G$ 

```

begin
1  if  $G$  is monopolar then
2    compute a monopolar partition  $(A, B)$  for  $G$ ; return ‘yes’ and  $(A, B)$ 
3  end if;
4  for  $X$  a clique of  $G$  of size 1 or 2 do
5    let  $\mathbf{T}$  be the  $X$ -trapezoid in  $\mathcal{D}$ ;
6    if  $G[\text{int}(\mathbf{T})]$  is polar and has polar partition  $(A', C)$  such that  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T}) \subseteq A'$  and
7     $C$  is a clique of  $G$  and maximal with  $X \subseteq C \subseteq \text{con}(\mathbf{T})$  then
8    let  $(A', C)$  be the computed polar partition for  $G[\text{int}(\mathbf{T})]$ ;
9    if  $A' \cap \text{con}(\mathbf{T}) \neq \emptyset$  then
10   let  $x \in A' \cap \text{con}(\mathbf{T})$ ;
11   if  $G \setminus \text{int}(\mathbf{T})$  has monopolar partition  $(A, B)$  with every vertex in  $A$  is adjacent to
12   every vertex in  $A' \cap N_G(x)$  and non-adjacent to every vertex in  $A' \setminus N_G(x)$  then
13   let  $(A, B)$  be the computed monopolar partition for  $G \setminus \text{int}(\mathbf{T})$ ;
14   return ‘yes’ and  $(A \cup A', B \cup C)$ 
15   end if
16   else
17   if  $G \setminus \text{int}(\mathbf{T})$  has monopolar partition  $(A, B)$  with
18   every vertex in  $A$  is adjacent to every vertex in  $A'$  then
19   let  $(A, B)$  be the computed monopolar partition for  $G \setminus \text{int}(\mathbf{T})$ ;
20   return ‘yes’ and  $(A \cup A', B \cup C)$ 
21   end if;
22   let  $L$  and  $R$  be the sets of respectively left-endpoint close and right-endpoint close vertices
23   for  $\mathbf{T}$  among the vertices in  $\text{int}(\mathbf{T}) \setminus \text{con}(\mathbf{T})$ ;
24   if there are vertex  $x \in L \cup R$  and monopolar partition  $(A, B)$  for  $G \setminus \text{int}(\mathbf{T})$  with
25   every vertex in  $A$  is adjacent to every vertex in  $A' \cap N_G(x)$  and non-adjacent to
26   every vertex in  $A' \setminus N_G(x)$  then
27   let  $(A, B)$  be the computed monopolar partition for  $G \setminus \text{int}(\mathbf{T})$ ;
28   return ‘yes’ and  $(A \cup A', B \cup C)$ 
29   end if;
30   if there are vertex  $x \in L$  and monopolar partition  $(A, B)$  for  $G \setminus (\text{int}(\mathbf{T}) \cup (N_G(x) \cap L(\mathbf{T})))$ 
31   with  $A \cap R(\mathbf{T}) = \emptyset$  and  $\{x\} \cup A$  an independent set of  $G$  and
32    $G[A \cup A' \cup (N_G(x) \cap L(\mathbf{T}))]$  complete multipartite then
33   let  $(A, B)$  be the computed monopolar partition for  $G \setminus \text{int}(\mathbf{T})$ ;
34   return ‘yes’ and  $(A \cup A' \cup (N_G(x) \cap L(\mathbf{T})), B \cup C)$ 
35   end if
36   end if
37   end if
38 end for;
39 return ‘no’
end.

```

Figure 4: The polar permutation graph recognition algorithm.

other cases. We consider the **for** loop during its last execution. Let  $\mathbf{T}$  be the trapezoid defined in line 5 and let  $(A', C)$  be the polar partition for  $G[\text{int}(\mathbf{T})]$  chosen in line 8. Note that  $(A', C)$

has the properties of lines 6–7. In particular, no vertex from  $C$  has a neighbour in  $G \setminus \text{int}(\mathbf{T})$ , which follows directly from  $C \subseteq \text{con}(\mathbf{T})$  and the properties of permutation diagrams. Hence, since  $C$  is a clique of  $G$  and  $G[B]$  is an induced subgraph of  $G \setminus \text{int}(\mathbf{T})$  and the disjoint union of complete graphs, for each of the four cases for  $B$ ,  $G[B \cup C]$  is the disjoint union of complete graphs. It remains to show for each of the four cases that the first component set of the output vertex partition induces a complete multipartite graph in  $G$ . The case is clear for the output in line 34 by the condition in line 32. Assume that the answer is output in line 14. Let  $(A, B)$  be the monopolar partition for  $G \setminus \text{int}(\mathbf{T})$  in line 13 and let  $x$  be the vertex chosen in line 10. Since  $A$  and  $A' \setminus N_G(x)$  are independent sets of  $G$ ,  $A \cup (A' \setminus N_G(x))$  is an independent set of  $G$  due to the conditions in line 11–12. And by the properties of complete multipartite graphs and the adjacency condition in lines 11–12, every vertex in  $A \cup (A' \setminus N_G(x))$  is adjacent to every vertex in  $A' \cap N_G(x)$ . Hence,  $G[A \cup A']$  is complete multipartite. The cases for the output in lines 20 and 28 follow similarly. We conclude that the output partition is indeed a polar partition for  $G$  and  $G$  is polar.

For the converse, let  $G$  be polar. We show that **Polar-Permutation-Graphs-Recognition** answers ‘yes’. If  $G$  is monopolar then **Polar-Permutation-Graphs-Recognition** returns answer ‘yes’ in line 2. Let  $G$  not be monopolar, and let  $(P, Q)$  be a  $Q$ -maximal polar partition for  $G$ . According to Lemma 4.1, there are a trapezoid  $\mathbf{T}$  in  $\mathcal{D}$  and a clique  $X$  of  $G$  with  $X \subseteq Q$  and  $1 \leq |X| \leq 2$  such that  $\mathbf{T}$  is the  $X$ -trapezoid in  $\mathcal{D}$  and  $\mathbf{T}$  is a centre trapezoid for  $(P, Q)$  in  $\mathcal{D}$  with  $X$  in-cliqued. Note that  $X$  and thus  $\mathbf{T}$  can be chosen by **Polar-Permutation-Graphs-Recognition** in lines 4 and 5. Let  $(A', C)$  be a polar partition for  $G[\text{int}(\mathbf{T})]$  as defined in condition 1 of the definition of centre trapezoids; such a partition exists by assumption. Without loss of generality, we can assume that every vertex in  $A' \cap \text{con}(\mathbf{T})$  is non-adjacent to some vertex in  $C$ . This partition satisfies the conditions in lines 6–7, and **Polar-Permutation-Graphs-Recognition** continues execution in line 8. Observe for the following arguments that  $P \cap \text{int}(\mathbf{T}) = A'$ : if there is  $u \in (P \cap \text{int}(\mathbf{T})) \setminus A'$  then  $u \in C$ . By the definition of  $(A', C)$  according to condition 1 of the definition of centre trapezoids, it holds that  $u \in \text{con}(\mathbf{T})$ , which means that  $u$  is not adjacent to any vertex in  $Q \setminus \text{int}(\mathbf{T})$ . Then,  $(P \setminus \{u\}, Q \cup \{u\})$  is a polar partition for  $G$ , which contradicts the choice of  $(P, Q)$  as  $Q$ -maximal. Then,  $P \cap (L(\mathbf{T}) \cup R(\mathbf{T})) = P \setminus A'$ . We distinguish between the two cases in condition 2 of the definition of centre trapezoids. As the first main case, let  $P \setminus A'$  be an independent set of  $G$ . Then,  $G \setminus \text{int}(\mathbf{T})$  is monopolar with monopolar partition  $(P \setminus A', Q \setminus C)$ . As a first subcase, let  $P \cap \text{con}(\mathbf{T}) = A' \cap \text{con}(\mathbf{T})$  be non-empty (line 9). For any vertex  $x \in A' \cap \text{con}(\mathbf{T})$ ,  $\{x\} \cup (P \setminus A')$  is an independent set of  $G$  and particularly in  $G[P]$ . With the properties of complete multipartite graphs, all vertices in  $P \setminus A'$  are adjacent to all vertices in  $A' \cap N_G(x)$  and non-adjacent to all vertices in  $A' \setminus N_G(x)$ . Then, the conditions in lines 11–12 are satisfied by partition  $(P \setminus A', Q \setminus C)$ , and the algorithm accepts. As a second subcase, let  $A' \cap \text{con}(\mathbf{T})$  be empty. If all vertices in  $P \setminus A'$  are adjacent to all vertices in  $A'$  then **Polar-Permutation-Graphs-Recognition** accepts in line 20. Let there be a vertex  $y \in P \setminus A'$  that is non-adjacent to some vertex  $x \in A'$ . Then,  $x$  and  $y$  are in the same maximal independent set of  $G[P]$ , which implies that  $\{x\} \cup (P \setminus A')$  is an independent set of  $G$ . Due to Lemma 4.2, there is a vertex  $z$  in  $L \cup R$  of lines 22–23 such that  $\{z\} \cup (P \setminus A')$  is an independent set of  $G$ . Analogous to the first subcase, **Polar-Permutation-Graphs-Recognition** accepts in line 28. As the second main case, let  $P \setminus A'$  not be an independent set of  $G$ . According to condition 2 of the definition of centre trapezoids,  $P \cap L(\mathbf{T}) \neq \emptyset$  and  $P \cap R(\mathbf{T}) = \emptyset$  and there is a vertex  $v \in A' \setminus \text{con}(\mathbf{T})$  with  $N_G(v) \cap L(\mathbf{T}) \subseteq P$  and  $\{v\} \cup (P \cap L(\mathbf{T})) \setminus N_G(v)$  is an independent set of  $G$ . Then,  $((P \setminus A') \setminus N_G(v), Q \setminus C)$  is a monopolar partition for  $G \setminus (\text{int}(\mathbf{T}) \cup (N_G(v) \cap L(\mathbf{T})))$ .

Due to Lemma 4.2, we can choose  $v$  from  $L$ . Then, **Polar-Permutation-Graphs-Recognition** accepts in line 34. This completes the correctness proof.

For the running time, observe that the **for** loop is executed at most  $n + m$  times. First, we show that each **for** loop execution takes time  $\mathcal{O}(n + m)$  plus the time for deciding whether a permutation graph has a monopolar partition  $(A, B)$  with  $A \subseteq F$  for  $F$  a given set of vertices. Trapezoid  $\mathbf{T}$  in line 5 can be computed in constant time from the given set  $X$ . The sets  $\text{int}(\mathbf{T})$ ,  $\text{con}(\mathbf{T})$ ,  $L(\mathbf{T})$ ,  $R(\mathbf{T})$  and  $L$  and  $R$  (in lines 22–23) can be computed in linear time straightforward by checking the endpoints of every vertex against the intervals of  $\mathbf{T}$ . We consider the conditional in lines 9–36. Assume that partition  $(A', C)$  for  $\text{int}(\mathbf{T})$  is given. The test  $A' \cap \text{con}(\mathbf{T}) \neq \emptyset$  (line 9) can be done in linear time. Vertex  $x$  in line 10 is chosen arbitrarily. For the conditional in lines 11–12, we need to compute an appropriate set  $F$ . We define  $F$  as the set of vertices in  $L(\mathbf{T}) \cup R(\mathbf{T})$  that are adjacent to every vertex in  $A' \cap N_G(x)$  and non-adjacent to every vertex in  $A' \setminus N_G(x)$ . This set can be computed in linear time. For the conditionals in lines 17–18 and 24–26, the corresponding sets  $F$  can be computed similarly. For the conditional in lines 30–32,  $F$  contains only vertices from  $L(\mathbf{T})$  due to the condition  $A \cap R(\mathbf{T}) = \emptyset$  and no vertices from  $N_G(x)$ . With the requirements on the computed polar partition,  $F$  can be computed in linear time analogous to the previous cases. The conditionals in lines 24–26 and 30–32 are executed several times with different choices for  $x$ . Since  $|L \cup R| \leq 4$ , the number of executions is constant. Hence, besides the running time for obtaining partitions  $(A', C)$  and  $(A, B)$ , a **for** loop execution takes linear time. The existence of partition  $(A, B)$  with the restriction  $F$  can be decided in  $\mathcal{O}(nm)$  time due to Theorem 3.5. By the remark at the end of Section 3, a partition  $(A, B)$  can be computed in the same time if it exists.

It remains to give an algorithm for deciding the conditional in lines 6–7. Since the vertices in  $X$  are adjacent to all vertices in  $\text{con}(\mathbf{T})$ , it suffices to compute a desired polar partition for  $G[\text{int}(\mathbf{T})] \setminus X$  and we only need to decide for vertices in  $\text{con}(\mathbf{T}) \setminus X$  whether they belong to the clique or to the complete multipartite graph. A polar partition for  $G[\text{int}(\mathbf{T})]$  of the type of lines 6–7 can be computed by applying the algorithm of Theorem 3.5 to the complement of  $G[\text{int}] \setminus X$  with set  $F$  chosen as  $\text{con}(\mathbf{T})$ . Let the output polar partition be  $(A', C)$ . If a vertex in  $A'$  is adjacent to all vertices in  $C$  then this vertex is moved to set  $C$ . Hence, in  $\mathcal{O}(nm)$  time, the conditional in lines 6–7 can be decided and a corresponding polar partition can be computed in the positive case. In total, we obtain  $\mathcal{O}(nm^2)$  running time. ■

An arbitrary graph is polar if and only if each of its connected components is polar. Hence, the result of Theorem 4.3 gives an  $\mathcal{O}(nm^2)$ -time algorithm for polar permutation graph recognition.

## 5 Concluding remarks and open problems

The running time of our monopolar permutation graph recognition algorithm is determined by the construction of the auxiliary digraph. The running time of the construction algorithm is equal to the theoretical upper bound on the number of arcs of the digraph. Is this bound tight? Is there an algorithm with running time dependent on the number of vertices and arcs of the auxiliary digraph? This is a possible approach to reduce the running time of recognising monopolar permutation graphs and in consequence also polar permutation graphs.

Permutation graphs are both comparability and cocomparability graphs. An interesting question is whether polar comparability graphs, or equivalently polar cocomparability graphs, can be recognised in polynomial time.

As we have mentioned in the introduction, there are well-studied problems that are NP-complete on permutation graphs. What are their computational complexities on polar permutation graphs? Extending the table in Figure 1, we can for instance ask for the complexity of partitioning a permutation graph into two cographs. In other words, is there a polynomial-time algorithm for the problem, given a permutation graph  $G$ , is there a partition  $(A, B)$  of  $V(G)$  with  $A$  and  $B$  induce graphs with no induced path on four vertices? Such graphs are particularly interesting for problems that are hard on the whole class of permutation graphs or for problems whose complexity has not yet been determined completely. Another question is whether a maximum induced polar subgraph in non-polar permutation graphs can be computed in polynomial time.

## Acknowledgement

We would like to thank Daniel Lokshantov for useful comments. We would also like to thank anonymous referees for their careful reviews of our paper and their helpful comments.

## References

- [1] S. Arnborg, J. Lagergren, D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.
- [2] H. Bodlaender. Chromatic number is NP-complete for cographs and interval graphs. *Information Processing Letters*, 31:135–138, 1989.
- [3] H. L. Bodlaender, T. Kloks, D. Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM Journal on Discrete Mathematics*, 8:606–616, 1995.
- [4] A. Brandstädt, V. B. Le, J. P. Spinrad. *Graph Classes: A Survey*. SIAM, 1999.
- [5] J. I. Brown and D. G. Corneil. On generalized graph colorings. *Journal of Graph Theory*, 11:87–99, 1987.
- [6] Z. A. Chernyak and A. A. Chernyak. About recognizing  $(\alpha, \beta)$ -classes of polar graphs. *Discrete Mathematics*, 62:133–138, 1986.
- [7] R. Churchley and J. Huang. The polarity and monopolarity of claw-free graphs, manuscript.
- [8] B. Courcelle. The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues. *Theoretical Informatics and Applications*, 26:257–286, 1992.
- [9] B. Courcelle, J. A. Makowsky, U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33:125–150, 2000.
- [10] T. Ekim, P. Heggernes, D. Meister. Polar permutation graphs. *Proceedings of IWOCOA 2009*, Springer LNCS, 5874:218–229, 2009.
- [11] T. Ekim, P. Hell, J. Stacho, D. de Werra. Polar chordal graphs. *Discrete Applied Mathematics*, 156:2469–2479, 2008.

- [12] T. Ekim and J. Huang. Recognizing line-polar bipartite graphs in time  $O(n)$ . *Discrete Applied Mathematics*, 158:1593–1598, 2010.
- [13] T. Ekim, N. V. R. Mahadev, D. de Werra. Polar cographs. *Discrete Applied Mathematics*, 156:1652–1660, 2008.
- [14] A. Farrugia. Vertex-partitioning into fixed additive induced-hereditary properties is NP-hard. *Electronic Journal of Combinatorics* 11, Vol. 1, 2004.
- [15] T. Feder, P. Hell, S. Klein, R. Motwani. Complexity of graph partition problems. *Proceedings of STOC 1999*, pp. 464–472, 1999.
- [16] F. V. Fomin, D. Kratsch, J.-C. Novelli. Approximating minimum cocolorings. *Information Processing Letters*, 84:285–290, 2002.
- [17] A. V. Gagarin. Chordal  $(1, \beta)$ -polar graphs. *Vestsi Nats. Akad. Navuk Belarusi Ser. Fiz.-Mat. Navuk*, 143:115–118, 1999.
- [18] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Annals of Discrete Mathematics, Vol 57, North-Holland, 2004.
- [19] M. C. Golumbic and U. Rotics. On the Clique-Width of Some Perfect Graph Classes. *International Journal of Foundations of Computer Science*, 11:423–443, 2000.
- [20] P. L. Hammer and B. Simeone. The splittance of a graph. *Combinatorica*, 1:275–284, 1981.
- [21] P. Hell, S. Klein, L. T. Nogueira, F. Protti. Partitioning chordal graphs into independent sets and cliques. *Discrete Applied Mathematics*, 141:185–194, 2004.
- [22] J. Huang and B. Xu. A forbidden subgraph characterization of line-polar bipartite graphs. *Discrete Applied Mathematics*, 158:666–680, 2010.
- [23] V. V. Lozin and R. Mosca. Polar graphs and maximal independent sets. *Discrete Mathematics*, 306:2901–2908, 2006.
- [24] R. M. McConnell and J. P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201:189–241, 1999.
- [25] O. I. Melnikov and P. P. Kozhich. Algorithms for recognizing the polarity of a graph with bounded parameters. *Vestsi Akad. Navuk BSSR Ser. Fiz.-Mat. Navuk*, 6:50–54, 1985.
- [26] R. I. Tyshkevich and A. A. Chernyak. Decompositions of Graphs. *Cybernetics and System Analysis*, 21:231–242, 1985.
- [27] R. I. Tyshkevich and A. A. Chernyak. Algorithms for the canonical decomposition of a graph and recognizing polarity. *Izvestia Akad. Nauk BSSR, ser. Fiz. Mat. Nauk*, 6:16–23, 1985, (in Russian).
- [28] K. Wagner. Monotonic coverings of finite sets. *Elektronische Informationsverarbeitung und Kybernetik*, 20:633–639, 1984.