

Vertex Deletion into Bipartite Permutation Graphs

Łukasz Bożyk 

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
l.bozyk@uw.edu.pl

Jan Derbisz

Theoretical Computer Science Department, Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland
jan.derbisz@doctoral.uj.edu.pl

Tomasz Krawczyk 

Theoretical Computer Science Department, Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland
krawczyk@tcs.uj.edu.pl

Jana Novotná 

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
janca@kam.mff.cuni.cz

Karolina Okrasa 

Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
k.okrasa@mini.pw.edu.pl

Abstract

A permutation graph can be defined as an intersection graph of segments whose endpoints lie on two parallel lines ℓ_1 and ℓ_2 , one on each. A bipartite permutation graph is a permutation graph which is bipartite. In this paper we study the parameterized complexity of the bipartite permutation vertex deletion problem, which asks, for a given n -vertex graph, whether we can remove at most k vertices to obtain a bipartite permutation graph. This problem is NP-complete by the classical result of Lewis and Yannakakis [17].

We analyze the structure of the so-called almost bipartite permutation graphs which may contain holes (large induced cycles) in contrast to bipartite permutation graphs. We exploit the structural properties of the shortest hole in a such graph. We use it to obtain an algorithm for the bipartite permutation vertex deletion problem with running time $f(k)n^{O(1)}$, and also give a polynomial-time 9-approximation algorithm.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases permutation graphs, comparability graphs, partially ordered set, graph modification problems

Digital Object Identifier 10.4230/LIPIcs.IPEC.2020.5

Related Version A full version of this paper is available at <https://arxiv.org/abs/2010.11440>.

Funding *Łukasz Bożyk, Jana Novotná, Karolina Okrasa:* This research is supported by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement 714704. It was partially carried out during the Parameterized Algorithms Retreat of the University of Warsaw, PARUW 2020, held in Krynica-Zdrój in February 2020.

Tomasz Krawczyk: This research is supported by National Science Center of Poland, grant UMO-2015/17/B/ST6/01873.

Jana Novotná: Supported by the student grant SVV-2020-260578.

Acknowledgements The authors would like to thank Bartosz Walczak for valuable comments and help with merging two groups of researchers working on similar projects into one. They also thank to anonymous reviewers for helpful comments.



© Łukasz Bożyk, Jan Derbisz, Tomasz Krawczyk, Jana Novotná, and Karolina Okrasa;
licensed under Creative Commons License CC-BY

15th International Symposium on Parameterized and Exact Computation (IPEC 2020).

Editors: Yixin Cao and Marcin Pilipczuk; Article No. 5; pp. 5:1–5:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

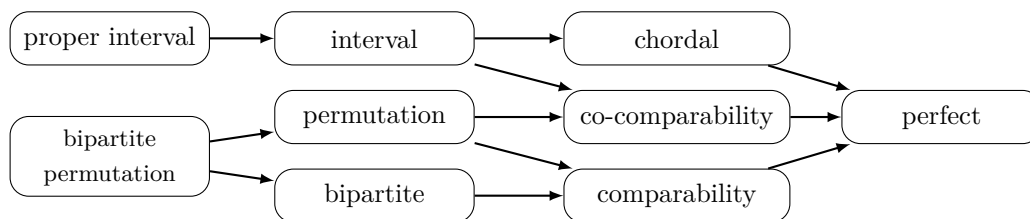


1 Introduction

Many standard computational problems, including maximum clique, maximum independent set, or minimum coloring, which are NP-hard in general, have polynomial-time exact or approximation algorithms in restricted classes of graphs. Due to the practical and theoretical applications, some of such graph classes are particularly intensively studied. Among them are:

- *interval graphs*: intersection graphs of intervals on a real line,
- *unit interval graphs*: intersection graphs of intervals none of which is contained in another,
- *chordal graphs*: intersection graphs of subtrees of a tree,
- *function and permutation graphs*: intersection graphs of continuous and linear functions, respectively, defined on the interval $[0, 1]$,
- *comparability graphs*: graphs whose edges correspond to the pairs of vertices comparable in some fixed partial order $<$ on the vertex set (such an order is called a *transitive orientation* of the graph),
- *co-comparability graphs*: the complements of comparability graphs.

It is well known that the class of function graphs corresponds to the class of co-comparability graphs [11], and the class of permutation graphs corresponds to the intersection of comparability and co-comparability graphs [21] (see Figure 1 for the hierarchy of inclusions). All these classes of graphs are *hereditary*, which means that they are closed under vertex deletion.



■ **Figure 1** An hierarchy of inclusion of the hereditary graph classes considered in the introduction. An arrow from graph class A to graph class B indicates that $A \subset B$.

Being hereditary is a very useful property in algorithmic design as every hereditary class of graphs can also be uniquely characterized in terms of minimal forbidden induced subgraphs: a graph belongs to a class \mathcal{G} if and only if it does not contain any graph from some family \mathcal{F} as an induced subgraph. For every graph class introduced above, a characterization by forbidden subgraphs is known, see [7] for perfect graphs, [16] for interval graphs, [10] for comparability and permutation graphs. However, for all of them, the family of forbidden subgraphs is infinite and it may also be quite complex. Moreover, every graph G from any class introduced above is *perfect*. Grötschel, Lovász, and Schrijver [12] showed that in the class of perfect graphs the maximum clique, the maximum independent set, and the minimum coloring problems can be solved in polynomial time.

Polynomial-time algorithms devised for the above-mentioned graph classes can sometimes be adjusted to also work on graphs that are “close” to graphs from these classes. Usually, the “closeness” of a graph G to a graph class \mathcal{G} is measured by the number of operations required to transform G into a graph from the class \mathcal{G} , where a single operation consists either on removing a vertex from G or on adding or removing an edge from G . Such an approach leads us to the following generic problem.

- Problem:** Graph modification problem into a class of graphs \mathcal{G}
- Input:** A graph G (typically not from \mathcal{G}) and a number k
- Question:** Can the graph G be transformed into a graph of the class \mathcal{G} by performing at most k modifications of an appropriate kind?

Depending on the kind of modifications allowed, we obtain four variants of this problem: vertex deletion problem, edge deletion problem, edge completion problem, and edge edition problem (the latter allowing both deletions and additions of edges). For the class of graphs defined above, all four variants of the modification problem are NP-hard – see [19] for references to NP-hardness proofs. In particular, Lewis and Yannakakis [17] showed that the vertex deletion problem into any non-trivial hereditary class of graphs is NP-hard. This is not surprising, as many classical hard problems can be formulated as vertex deletion problems into particular classes of graphs, for example, VERTEX COVER as vertex deletion to edgeless graphs, FEEDBACK VERTEX SET as vertex deletion to forests, and ODD CYCLE TRANSVERSAL as vertex deletion to bipartite graphs.

Graph modification problems are a popular research direction in the study of the *parameterized complexity* of NP-complete problems. In general, for a problem Π , an input of a parameterized problem consists of an instance I of Π and a *parameter* $k \in \mathbb{N}$. Then we say that Π is *fixed parameter tractable* (FPT) if there exists an algorithm deciding whether I is a yes-instance of Π in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where f is some computable function. For a graph modification problem, we often choose the parameter k as a number of allowed modifications, so the instance of such a problem is still a pair (G, k) .

It turns out that characterizations by forbidden structures are sometimes useful to design FPT algorithms for graph modification problems. For example, Cai [3] proposed an FPT algorithm for modification problems into classes of graphs characterized by a finite family of forbidden induced subgraphs \mathcal{F} . His algorithm identifies a forbidden structure in the input graph (which can be done in polynomial time when \mathcal{F} is finite) and branches over all possible ways of modifying that structure. Since the families of forbidden structures are infinite for graph classes introduced above, modification algorithms for these classes have to be much more sophisticated. For several of them modification problems have satisfactory solutions:

- chordal graphs: all four versions of the modification problem are FPT [6, 20];
- interval graphs: edge completion and edge deletion are FPT [25, 4], vertex deletion is FPT [6], edge edition remains open;
- proper interval graphs: all four versions of the modification problem are FPT [5].

On the other hand, it is known that the vertex deletion to perfect graphs is W[2]-hard [13]. It is worth mentioning that for a long time, it was unknown whether there are classes of graphs recognizable in polynomial time for which modification problems are hard. The first such example was given by Lokshantov [18], who proved that the vertex deletion is W[2]-hard for graphs avoiding all *wheels* (i.e., cycles with an additional vertex adjacent to all other vertices). It is unknown whether comparability graphs, co-comparability graphs, and permutation graphs have FPT modification algorithms. The class of co-comparability graphs, which constitutes the superclass of interval graphs and an important subclass of perfect graphs, seems to be particularly interesting from the parameterized point of view.

Our focus. Like the class of interval graphs, the class of permutation graphs admits polynomial-time algorithms for rich family problems which are NP-complete in general. Apart from the already mentioned classical hard problems which are polynomial-time solvable for perfect graphs, there also exist polynomial algorithms solving e.g., HAMILTONIAN CYCLE, FEEDBACK VERTEX SET or DOMINATING SET in the class of permutation graphs [2, 8].

In light of the above considerations, since all the modification problems into the class of permutation graphs – and the related classes of comparability and co-comparability graphs – remain open, restricting our attention to the class of *bipartite permutation graphs* appears to be a natural research direction. Bipartite permutation graphs form an interesting graph class themselves, first investigated by Spinrad, Brandstädt, and Stewart [22], who characterized them by means of appropriately chosen linear orderings of its bipartition classes.

One of the most interesting results concerning the bipartite permutation graphs is by Heggernes et al. [14], who showed that the NP-complete problem of computing the *cutwidth* of a graph (i.e., finding a linear order of the vertices of a graph that minimizes the maximum number of edges intersected by any line inserted between two consecutive vertices) is polynomial for bipartite permutation graphs.

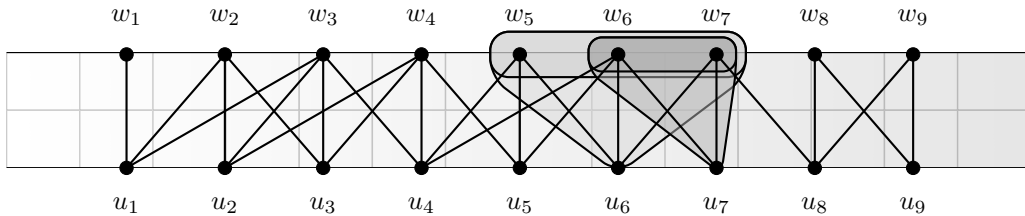
Our algorithm exploits the absence of some forbidden structures in bipartite permutation graphs. Since these structures cannot, in particular, occur in permutation graphs, we believe that besides being a complete result itself, our research is a step towards understanding the parameterized complexity of modification problems into permutation graphs.

Our results. We focus mainly on the modification by vertex deletion.

► **Theorem 1.** *There is an $\mathcal{O}(9^k \cdot |V(G)|^9)$ -time algorithm for instances (G, k) of the vertex deletion into bipartite permutation graphs problem.*

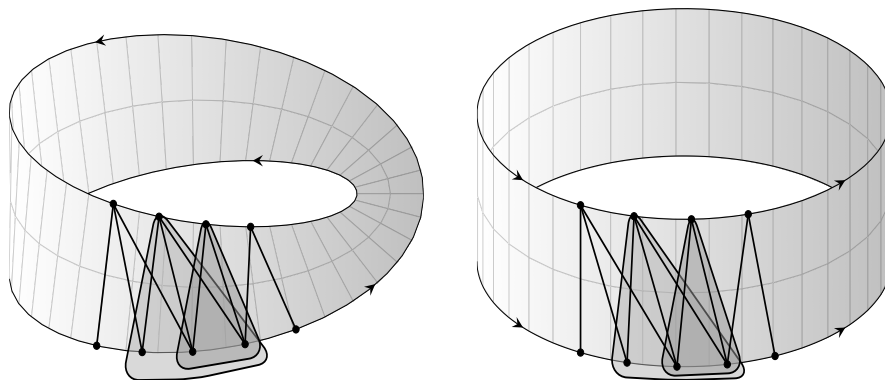
We prove Theorem 1 in Section 4. Our algorithm is based on the characterization of bipartite permutation graphs by forbidden subgraphs. Using the characterization, at first, we get rid of constant-size forbidden subgraphs by branching, which is a standard technique in modification problems on hereditary graph classes [24, 25]. We call graphs without these forbidden subgraphs *almost bipartite permutation graphs*.

Our main contribution is in the structural analysis of almost bipartite permutation graphs which may contain holes (on more than ten vertices) in contrast to bipartite permutation graphs. This approach is partially inspired by the ideas of van 't Hof and Villanger [24] who used similar tools in their work on proper interval vertex deletion problem. We use the result of Spinrad, Brandstädt, and Stewart [22], who showed that the vertices of every connected bipartite permutation graph $G = (U, W, E)$ can be embedded into a strip in such a way that the vertices from U are on the bottom edge of the strip, the vertices from W are on the top edge of the strip, the neighbors $N(u)$ of u occur consecutively on the top edge of the strip for every $u \in U$ (adjacency property), the vertices from $N(u) - N(u')$ occur consecutively on the top edge of the strip for every $u, u' \in U$ (enclosure property), and the analogous properties are satisfied by the vertices in W (see Figure 2).



■ **Figure 2** Embedding of a bipartite permutation graph (U, W, E) into a strip satisfying the adjacency and the enclosure properties.

Our structural result asserts that, depending on the parity of the length of the shortest hole, a connected almost bipartite permutation graph may be naturally embedded in either a cylinder, or a Möbius strip, locally satisfying adjacency and enclosure properties (see Fig. 3).



■ **Figure 3** Embedding of an almost bipartite permutation graph in a cylinder or Möbius strip.

Once we obtain such structure, we show that every minimal vertex cut that destroys all holes lies nearby a few consecutive vertices from the shortest hole. This allows us to check all the possibilities where we can find a minimum cut. Finally, we use a polynomial algorithm for finding maximum flow (and thus a minimum cut).

The approach used to prove Theorem 1 can be slightly modified to obtain a 9-approximation algorithm for the bipartite permutation vertex deletion problem. We show the following.

► **Theorem 2.** *There exists a polynomial-time 9-approximation algorithm for vertex deletion into bipartite permutation graphs problem.*

2 Preliminaries

Unless stated otherwise, all graphs considered in this work are simple, i.e., undirected, with no loops and parallel edges. Let $G = (V, E)$ be a graph. For a subset $S \subseteq V$, the subgraph of G induced by S is the graph $G[S] = (S, \{uv \mid uv \in E, u, v \in S\})$. The *neighborhood* of a vertex $u \in V$ is the set $N(u) = \{v \in V \mid uv \in E\}$. Similarly, we write $N(U) = \bigcup_{u \in U} N(u) \setminus U$ for a set $U \subseteq V$. Let $u, v \in V$. We say that u and v are at *distance k* (in G) if k is the length of a shortest path between u and v in G . We denote a complete graph and a cycle on n vertices by K_n and C_n , respectively. By *hole* we mean an induced cycle on at least five vertices. We say that a hole is *even* (or *odd*) if it contains even (odd) number of vertices, respectively.

For a graph $G = (V, E)$, a pair $(V, <)$ is a *transitive orientation* of G if $<$ is a transitive and irreflexive relation on V that satisfies either $u < v$ or $v < u$ iff $uv \in E$ for every $u, v \in V$.

A *partially ordered set* (shortly *partial order* or *poset*) is a pair $P = (X, \leq_P)$ that consists of a set X and a reflexive, transitive, and antisymmetric relation \leq_P on X . For a poset (X, \leq_P) , let the *strict partial order* $<_P$ be a binary relation defined on X such that $x <_P y$ if and only if $x \leq_P y$ and $x \neq y$. Equivalently, $(X, <_P)$ is a strict partial order if $<_P$ is irreflexive and transitive. Two elements $x, y \in X$ are *comparable* in P if $x \leq_P y$ or $y \leq_P x$; otherwise, x, y are *incomparable* in P . A *linear order* $L = (X, \leq_L)$ is a partial order in which for every $x, y \in X$ we have $x \leq_L y$ or $y \leq_L x$. A *strict linear order* $(X, <_L)$ is a binary relation defined in a way that $x <_L y$ if and only if $x \leq_L y$ and $x \neq y$.

Let $P = (X, \leq_P)$ be a poset. A linear order $L = (X, \leq_L)$ is called a *linear extension* of P if $\leq_P \subseteq \leq_L$. Given a family of posets $\mathcal{P} = \{P_i = (X, \leq_{P_i}) : i \in I\}$, we say that P is the *intersection* of \mathcal{P} if for every $x, y \in X$ we have $x \leq_P y$ if and only if $x \leq_{P_i} y$ for every $i \in I$. The *dimension* of a poset P is the minimal number of linear extensions of P that intersect to P . We say that P is *two-dimensional* if it is the intersection of two linear extensions of P .

A *comparability graph* (*incomparability graph*) of a poset $P = (X, \leq_P)$ has X as the set of its vertices and the set including every two vertices comparable (incomparable, respectively) in P as the set of its edges. Note the following: if (X, \leq_P) is a poset, then $(X, <_P)$ is a transitive orientation of the comparability graph of P . A graph $G = (V, E)$ is a *comparability graph* (*co-comparability graph*) if G is a comparability (incomparability, respectively) graph of some poset defined on V . So, G is a comparability graph if and only if G admits a transitive orientation. A graph G is a *permutation graph* if and only if G and the complement of G are comparability graphs [21] (or equivalently, G and the complement of G admit transitive orientations). Baker, Fishburn, and Roberts [1] proved that G is a permutation graph if and only if G is the incomparability graph of a two-dimensional poset.

We say that two sets X and Y are *comparable* if X and Y are comparable with respect to \subseteq -relation (that is, $X \subseteq Y$ or $Y \subseteq X$ holds). We use the convenient notation $[m] := \{0, 1, \dots, m\}$, for every $m \in \mathbb{N}$. For every $i, j \in \mathbb{Z}$ such that $i \leq j$ by $[i, j]$ we mean the set $\{i, i + 1, \dots, j\}$.

3 The structure of (almost) bipartite permutation graphs

The characterization of bipartite permutation graphs presented below was proposed by Spinrad, Brandstädt, and Stewart [22].

Suppose $G = (U, W, E)$ is a connected bipartite graph. A linear order $(W, <_W)$ satisfies *adjacency property* if for each vertex $u \in U$ the set $N(u)$ consists of vertices that are consecutive in $(W, <_W)$. A linear order $(W, <_W)$ satisfies *enclosure property* if for every pair of vertices $u, u' \in U$ such that $N(u)$ is a subset of $N(u')$, vertices in $N(u') - N(u)$ occur consecutively in $(W, <_W)$. A *strong ordering* of the vertices of $U \cup W$ consists of linear orders $(U, <_U)$ and $(W, <_W)$ such that for every $(u, w'), (u', w)$ in E , where u, u' are in U and w, w' are in W , $u <_U u'$ and $w <_W w'$ imply $(u, w) \in E$ and $(u', w') \in E$. Note that, whenever $(U, <_U)$ and $(W, <_W)$ form a strong ordering of $U \cup W$, then $(U, <_U)$ and $(W, <_W)$ satisfy the adjacency and enclosure properties.

► **Theorem 3** (Spinrad, Brandstädt, Stewart [22]). *The following three statements are equivalent for a connected bipartite graph $G = (U, W, E)$:*

1. (U, W, E) is a bipartite permutation graph.
2. There exists a strong ordering of $U \cup W$.
3. There exists a linear order $(W, <_W)$ of W satisfying adjacency and enclosure properties.

An example of a bipartite permutation graph $G = (U, W, E)$ with linear order $w_1 <_W w_2 <_W \dots <_W w_8 <_W w_9$ of the vertices of W which satisfies the adjacency and the enclosure properties is shown in Figure 2.

Another characterization of bipartite permutation graphs can be obtained by listing all minimal forbidden induced subgraphs for this class of graphs. Such a list can be compiled by taking all odd cycles of length ≥ 3 (forbidden structures for bipartite graphs) and all bipartite graphs from the list of forbidden structures for permutation graphs obtained by Gallai [10]. The whole list is shown in Figure 4.

3.1 Almost bipartite permutation graphs

The goal of this section is to characterize graphs which do not contain small forbidden subgraphs for the class of bipartite permutation graphs. Following terminology of van 't Hof and Villanger [24] we call such graphs almost bipartite permutation graphs.

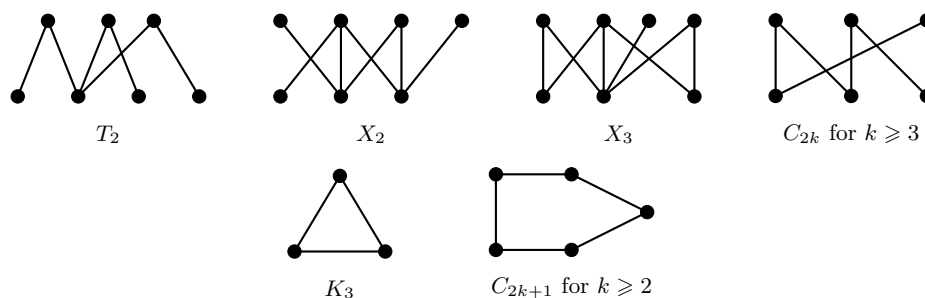


Figure 4 Forbidden structures for bipartite permutation graphs.

► **Definition 4.** A graph $G = (V, E)$ is almost bipartite permutation graph if G does not contain T_2, X_2, X_3, K_3, C_k for $k \in [5, 9]$ as induced subgraphs.

Suppose $G = (V, E)$ is a connected almost bipartite permutation graph.

► **Proposition 5.** Every hole in G is a dominating set.

Proof. Let $C = \{c_0, c_1, \dots, c_\ell\}$ be a hole in G . Hence, $\ell \geq 10$. Suppose, for contradiction, that there exists a vertex in the set $V \setminus (C \cup N(C))$. As G is connected, there must exist $v \in V$ in distance two from C . Let $w \in N(v) \cap N(C)$ and let c_j be a neighbor of w in C . We now look at the neighborhood of w . As G contains no triangle, wc_{j-1} and wc_{j+1} are non-edges. Moreover, as G contains no copy of T_2 , vertex w is adjacent to at least one of c_{j-2} and c_{j+2} , say c_{j-2} . Thus, w is nonadjacent to c_{j-3} . Therefore, the set $\{c_{j-3}, c_{j-2}, c_{j-1}, c_j, c_{j+1}, w, v\}$ induces a copy of X_2 , which leads to a contradiction. ◀

Let C be a shortest hole in G , m be the size of C , and c_0, c_1, \dots, c_{m-1} be the consecutive vertices of C , $m \geq 10$. In the remaining part of the paper we use the following notation with respect to C . For any integral number i by c_i we denote the unique vertex $c_{i \bmod m}$ from the cycle C . For any two different vertices c_i, c_j in C , by the set of all vertices between c_i and c_j from C we mean the set $\{c_i, c_{i+1}, \dots, c_{i+k}\}$, where k is the smallest natural number such that $c_{i+k} = c_j$. Note that this notion is not symmetric, i.e., the set of all vertices between c_j and c_i from C contains c_i, c_j and all the vertices from C that are not between c_i and c_j .

► **Proposition 6.** For every vertex $v \in V$ either:

- (1) $N(v) \cap C = \{c_i\}$ for some $i \in [m - 1]$, or
- (2) $N(v) \cap C = \{c_i, c_{i+2}\}$ for some $i \in [m - 1]$.

Proof. Since C is a cycle, (2) clearly holds for the vertices from C , so let v be a vertex in $V \setminus C$. As C is a dominating set, by Proposition 5, vertex v has at least one neighbor in C . If v has exactly one neighbor in C , then (1) holds and we are done. So assume that it has more than one neighbor. We now distinguish two cases. First, suppose that there exist two vertices $c_j, c_\ell \in N(v) \cap C$ at distance at least three in C such that v has no neighbor in the set of vertices between c_j and c_ℓ , except c_j and c_ℓ . Then, $\{c_j, c_{j+1}, \dots, c_\ell, v\}$ induces a cycle C' on at least five vertices in G . As c_j and c_ℓ are at distance at least three in C , C' is shorter than C . In particular, C' contradicts either G containing no copy of C_ℓ , for $\ell \in \{5, \dots, 9\}$, or C being a shortest hole in G . Therefore, this case never occurs.

Hence, v has either (i) exactly two neighbors in C and those are at distance two as there is no triangle in G , so (2) holds, or (ii) C has an even number of vertices and v is adjacent to every second vertex of C . It remains to show that the latter never occurs. Indeed, if it does, then without loss of generality $c_0 \in N(v)$. But observe that since C has at least ten vertices, the set $\{c_0, c_1, c_2, c_3, c_4, c_6, v\}$ induces a copy of X_3 . This concludes the proof. ◀

Given Prop. 6, for every $i \in [m-1]$ we can set $A_i = \{v \in V : N(v) \cap C = \{c_{i-1}, c_{i+1}\}\}$ and $B_i = \{v \in V : N(v) \cap C = \{c_i\}\}$. Note that $A_0, B_0, \dots, A_{m-1}, B_{m-1}$ is a partition of V and $c_i \in A_i$. Following our notation, for any integer i by A_i and B_i we denote the sets $A_{i \bmod m}$ and $B_{i \bmod m}$, respectively. Furthermore, for every $i \leq j$ we set:

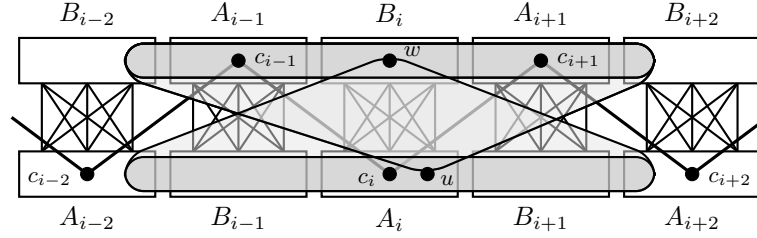
$$A_G[i, j] = \begin{cases} A_i \cup B_{i+1} \cup A_{i+2} \cup B_{i+3} \cup \dots \cup A_{j-1} \cup B_j & \text{if } j-i \text{ is odd,} \\ A_i \cup B_{i+1} \cup A_{i+2} \cup B_{i+3} \cup \dots \cup B_{j-1} \cup A_j & \text{if } j-i \text{ is even,} \end{cases}$$

$$B_G[i, j] = \begin{cases} B_i \cup A_{i+1} \cup B_{i+2} \cup A_{i+3} \cup \dots \cup B_{j-1} \cup A_j & \text{if } j-i \text{ is odd,} \\ B_i \cup A_{i+1} \cup B_{i+2} \cup A_{i+3} \cup \dots \cup A_{j-1} \cup B_j & \text{if } j-i \text{ is even,} \end{cases}$$

and $V_G[i, j] = A_G[i, j] \cup B_G[i, j]$.

We write just $A[i, j]$, $B[i, j]$, and $V[i, j]$, respectively, instead of $A_G[i, j]$, $B_G[i, j]$, and $V_G[i, j]$, when there is no confusion.

We now characterize the neighborhoods of the vertices in sets A_i and B_i , see also Figure 5.



■ **Figure 5** A possible neighborhood of u in A_i and w in B_i .

► **Proposition 7.** Let $i \in [m-1]$. Then:

- (1) A_i and B_i are independent sets.
- (2) For every $u \in A_i$ and every $w \in B_i$ we have $uw \in E$.
- (3) For every $u \in A_i$ we have $B_i \subseteq N(u) \subseteq B[i-2, i+2]$.
- (4) For every $w \in B_i$ we have $A_i \subseteq N(w) \subseteq A[i-2, i+2]$.

The proof of Proposition 7 can be found in the full version or it can also be easily seen by examining the forbidden subgraphs and the fact that C is a shortest hole containing at least 10 vertices. Proposition 7 asserts that all the neighbors of the vertices from A_i and from B_i are contained in the set $B[i-2, i+2]$ and $A[i-2, i+2]$, respectively. The next proposition describes the relations that hold between the neighborhoods of the vertices from $B[i-2, i+2]$ restricted to the set A_i and between the neighborhoods of the vertices from $A[i-2, i+2]$ restricted to the set B_i .

► **Proposition 8.** Let $i \in [m-1]$. For $(i \pm 2, i \pm 1) \in \{(i-2, i-1), (i+2, i+1)\}$, the following hold:

- (1) For every $w, w' \in B_{i \pm 2} \cup A_{i \pm 1}$ the sets $N(w) \cap A_i$ and $N(w') \cap A_i$ are comparable. Moreover, if $w \in B_{i \pm 2}$ and $w' \in A_{i \pm 1}$, then $N(w) \cap A_i \subseteq N(w') \cap A_i$.
- (2) For every $u, u' \in A_{i \pm 2} \cup B_{i \pm 1}$ the sets $N(u) \cap B_i$ and $N(u') \cap B_i$ are comparable. Moreover, if $u \in A_{i \pm 2}$ and $u' \in B_{i \pm 1}$, then $N(u) \cap B_i \subseteq N(u') \cap B_i$.

Proof. To prove (1), we consider the case $(i \pm 2, i \pm 1) = (i-2, i-1)$, as the other one follows by symmetry. Suppose that $w, w' \in B_{i-2} \cup A_{i-1}$ are such that neither $N(w) \cap A_i \subseteq N(w') \cap A_i$ nor $N(w') \cap A_i \subseteq N(w) \cap A_i$ holds. It means that there are $u, u' \in A_i$ such that $wu \in E$, $w'u' \in E$, $wu' \notin E$, and $w'u \notin E$. Since $w, w' \in B_{i-2} \cup A_{i-1}$, we

have $c_{i-2}w, c_{i-2}w' \in E$ and $c_{i-4}w, c_{i-3}w, c_{i-4}w', c_{i-3}w' \notin E$. Furthermore, $ww' \notin E$ and $uu' \notin E$ as G contains no triangle. Consequently, the set $\{c_{i-3}, w, w', c_{i-4}, c_{i-2}, u, u'\}$ induces a copy of T_2 in G , which cannot be the case. Moreover, if $w \in B_{i-2}, w' \in A_{i-1}$, then since $c_i \in (N(w') \cap A_i) \setminus (N(w) \cap A_i)$, the latter statement holds.

To show (2), we again only consider the case $(i \pm 2, i \pm 1) = (i - 2, i - 1)$. Suppose that $u, u' \in A_{i-2} \cup B_{i-1}$ are such that neither $N(u') \cap B_i \subseteq N(u) \cap B_i$ nor $N(u) \cap B_i \subseteq N(u') \cap B_i$ holds. It means that there are $w, w' \in B_i$ such that $uw, u'w' \in E$ and $u'w, uw' \notin E$. Since $u, u' \in A_{i-2} \cup B_{i-1}$, we have $uc_{i-1}, u'c_{i-1} \in E$ and $uc_{i+1}, u'c_{i+1} \notin E$. Furthermore, $uu' \notin E$ and $ww' \notin E$ as G contains no triangle. Hence, the set $\{c_{i-1}, w, w', c_{i+1}, u, c_i, u'\}$ induces a copy of X_3 in G , which cannot be the case.

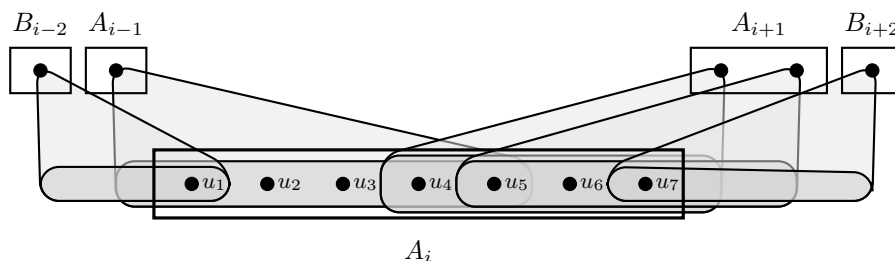
To see the second part of the statement, assume that $N(u) \cap B_i \not\subseteq N(u') \cap B_i$ for some $u \in A_{i-2}, u' \in B_{i-1}$. That is, there is $w \in B_i$ such that $uw \in E$ and $u'w \notin E$. In particular, it means that $u \neq c_{i-2}$. Note that $uc_{i-1}, u'c_{i-1} \in E$. Consequently, the set $\{c_{i-3}, c_{i-2}, c_{i-1}, c_i, u, u', w\}$ induces a copy of X_3 in G , which is a contradiction. ◀

Proposition 8 allows us to order vertices of A_i based on two properties. We now define relation $<_{A_i}$ which combines them and we show that $<_{A_i}$ is a partial order (see Figure 6 for an illustration). We define for every $u, u' \in A_i$:

$$u <_{A_i} u' \text{ iff } \begin{cases} \text{there is } w \in B_{i-2} \cup A_{i-1} \text{ such that } u \in N(w) \text{ and } u' \notin N(w), \text{ or} \\ \text{there is } w \in A_{i+1} \cup B_{i+2} \text{ such that } u' \in N(w) \text{ and } u \notin N(w), \end{cases}$$

Similarly, we define a relation $<_{B_i}$ for every $w, w' \in B_i$:

$$w <_{B_i} w' \text{ iff } \begin{cases} \text{there is } u \in A_{i-2} \cup B_{i-1} \text{ such that } w \in N(u) \text{ and } w' \notin N(u), \text{ or} \\ \text{there is } u \in B_{i+1} \cup A_{i+2} \text{ such that } w' \in N(u) \text{ and } w \notin N(u). \end{cases}$$



■ **Figure 6** The neighborhoods of the vertices from $B_{i-2} \cup A_{i-1} \cup A_{i+1} \cup B_{i+2}$ restricted to A_i . We have $u_1 <_{A_i} \{u_2, u_3\} <_{A_i} u_4 <_{A_i} u_5 <_{A_i} u_6 <_{A_i} u_7$.

► **Proposition 9.** *The following statements hold for every $i \in [m - 1]$:*

1. $(A_i, <_{A_i})$ is a strict partial order. Moreover, $u, u' \in A_i$ are incomparable in $(A_i, <_{A_i})$ if and only if $N(u) = N(u')$.
2. $(B_i, <_{B_i})$ is a strict partial order. Moreover, $w, w' \in B_i$ are incomparable in $(B_i, <_{B_i})$ if and only if $N(w) = N(w')$.

A formal proof of Proposition 9 can be found in the full version. Here, we only observe that we obtain two orders of vertices of A_i (resp. B_i) from Proposition 8 and they are reflected in the definition of $<_{A_i}$ (resp. $<_{B_i}$). However, the fact that our graph does not contain small forbidden induced subgraphs, combined with the previous proposition, implies that we cannot have $u <_{A_i} u'$ and $u' <_{A_i} u$ simultaneously, for any $u, u' \in A_i$ (an analogous

5:10 Vertex Deletion into Bipartite Permutation Graphs

situation holds for \prec_{B_i}). For instance, if it did not hold for some $u, u' \in A_i$ because of the existence of some $w \in B_{i-2} \cup A_{i-1}$ such that $u \in N(w)$ and $u' \notin N(w)$, and some $w' \in A_{i+1} \cup B_{i+2}$ such that $u \in N(w')$ and $u' \notin N(w')$, an induced copy of X_2 would be found on vertices $\{w, w', c_{i+1}, c_{i+2}, c_{i+3}, u, u'\}$. These observations are a key ingredient of proving the transitivity of our relation, the proof itself requires more technicalities.

Finally, for every $i \in [m-1]$ we order arbitrarily the elements inside every antichain of (A_i, \prec_{A_i}) and of (B_i, \prec_{B_i}) , obtaining strict linear orders (A_i, \prec_{A_i}) and (B_i, \prec_{B_i}) . We introduce a binary relation \prec defined on the set V , such that $v \prec v'$ for $v, v' \in V$ if one of the following conditions holds for some $i \in [m-1]$:

- $v, v' \in A_i$, $v \prec_{A_i} v'$, and v, v' are consecutive in (A_i, \prec_{A_i}) ,
- $v, v' \in B_i$, $v \prec_{B_i} v'$, and v, v' are consecutive in (B_i, \prec_{B_i}) ,
- v is the maximum of (A_i, \prec_{A_i}) and v' is the minimum of $(B_{i+1}, \prec_{B_{i+1}})$,
- v is the maximum of (B_i, \prec_{B_i}) and v' is the minimum of $(A_{i+1}, \prec_{A_{i+1}})$.

Informally, to get an embedding of G into a cylinder (the shortest hole is even) or into a Möbius strip (the shortest hole is odd) which locally satisfies the adjacency and the enclosure properties, we place the vertices v, v' satisfying $v \prec v'$ next to each other, v before v' assuming that the border of the cylinder or the Möbius strip are oriented as shown in Figure 3. In what follows we extend \prec relation as follows:

- For every $V' \subsetneq V$ by $\prec_{V'}$ we denote the transitive closure of \prec restricted to V' ,
- For $v, v' \in V$ we set $v \prec_{cl} v'$ if $v, v' \in A[i-2, i+2]$ and $v \prec_{A[i-2, i+2]} v'$ for some $i \in [m-1]$ or $v, v' \in B[i-2, i+2]$ and $v \prec_{B[i-2, i+2]} v'$ for some $i \in [m-1]$.

Finally, the following lemma characterizes the global structure of an almost bipartite permutation graph. The proof is omitted and can be found in the full version.

► **Lemma 10.** *Let i, j be such that $i \leq j$, $|j - i| = m - 3$. Let $U = A[i, j]$ and $W = B[i, j]$. Then $G[U \cup W]$ is a bipartite permutation graph with bipartition classes U and W .*

Moreover, (U, \prec_U) and (W, \prec_W) are strict linear orders that satisfy the adjacency and enclosure properties in $G[U \cup W]$.

Lemma 10 provides an interesting view on classification of almost bipartite permutation graphs. Specifically, if m is even, then the graph may be drawn on a cylinder, whose boundary consists of two closed curves, one of which traverses vertices of $A[0, m-1]$, and the second one – the vertices of $B[0, m-1]$. If in turn m is odd, then the graph can be represented on a Möbius strip, whose boundary traverses consecutive vertices of $A[0, m-1]$ and then $B[0, m-1]$ (recall Figure 3).

The following definitions are taken from [24]. A *hole cut* of G is a vertex set $X \subseteq V$ such that $G - X$ is a bipartite permutation graph. Lemma 10 asserts that for every $i \in [m-1]$ the set $V[i, i+1]$ is a hole cut in G . A hole cut X of G is *minimum* if G does not have a hole cut whose size is strictly smaller than the size of X . A hole cut X of G is *minimal* if any proper subset of X is not a hole cut in G . Note, in particular, that for every $i \in [m-1]$ the set $V[i, i+1]$ is a hole cut.

The next proposition describes the structure of every hole in G .

► **Proposition 11.** *Suppose C' is a hole of size k in G for some $k \geq m$. Then, the consecutive vertices of C' can be labeled by $c'_0, c'_1, \dots, c'_{k-1}$ so as the following conditions hold (the indices are taken modulo k):*

- $c'_i c'_{i+1} \in E$ for every $i \in [k-1]$,
- $c'_i \prec_{cl} c'_{i+2}$ for every $i \in [k-1]$,
- $C' \cap \{c'' : c'_i \prec_{cl} c'' \prec_{cl} c'_{i+2}\} = \emptyset$ for every $i \in [k-1]$.

Proof. Let $c'_0 = v_i, c'_1, c'_2, \dots, c'_{n-1}$ be consecutive vertices of C' denoted in such a way that $c'_0 <_{cl} c'_2$. We note that we can suppose it as $c'_0, c'_2 \in N(c'_1)$, thus, by Proposition 7(3) and (4), both c'_0, c'_2 belong to $A[\ell - 2, \ell + 2]$ or both belong to $B[\ell - 2, \ell + 2]$ for some $\ell \in [m - 1]$.

Now, we show that if there exists $j \in [m - 1]$ such that $c'_j <_{cl} c'_{j+2}$, then $c'_{j+1} <_{cl} c'_{j+3}$. Suppose, for contradiction that $c'_j <_{cl} c'_{j+2}$ and $c'_{j+1} \not<_{cl} c'_{j+3}$. Let $i \in [m - 1]$ be such that $c'_{j+2} \in (A_i \cup B_i)$. Similarly, as $c'_{j+1}, c'_{j+3} \in N(c'_{j+2})$, either $c'_{j+1}, c'_{j+3} \in B[i - 2, i + 2]$ if $c'_{j+2} \in A_i$ or $c'_{j+1}, c'_{j+3} \in A[i - 2, i + 2]$ if $c'_{j+2} \in B_i$. In both cases Lemma 10 implies that $<_{cl}$ restricted to $V[i - 4, i + 2]$ is a strong ordering of $G[V[i - 4, i + 2]]$. Moreover, c'_{j+1}, c'_{j+3} are comparable in $<_{cl}$, by Proposition 7(3) and (4), thus, $c'_{j+3} <_{cl} c'_{j+1}$. Therefore Theorem 3(2) yields $c'_j c'_{j+3} \in E$, so a chord in C' – contradiction. Therefore $c'_j <_{cl} c'_{j+2}$ implies $c'_{j+1} <_{cl} c'_{j+3}$ for every integer j . Applying above observation repeatedly for $j = 0, 1, 2, \dots$, we get that $c'_0 <_{cl} c'_2 <_{cl} c'_4 <_{cl} \dots$ and $c'_1 <_{cl} c'_3 <_{cl} c'_5 <_{cl} \dots$.

Suppose for the sake of contradiction that there exists $j \notin \{i, i + 2\}$ such that $c'_i <_{cl} c'_j <_{cl} c'_{i+2}$. Then by Lemma 10, $c'_j c'_{i+1} \in E$ due to the adjacency property. But the edge $c'_j c'_{i+1}$ is a chord in C' . This completes the proof. ◀

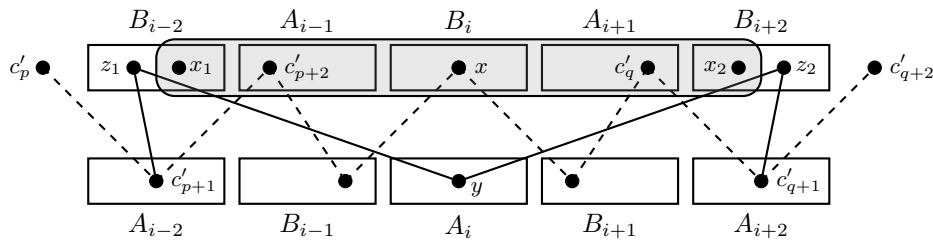
The structure of holes described above asserts that for every $i \in [m - 1]$ the sets $A[i - 2, i + 2]$ and $B[i - 2, i + 2]$ are hole cuts. We use this observation to prove the following statement about minimal hole cuts in G .

▶ **Proposition 12.** *Every minimal hole cut X in G is fully contained in the set $V[i - 2, i + 2]$ for some $i \in [m - 1]$.*

Proof. First, note that we can choose elements z_1, x_1, x_2, z_2 in V and an index $i \in [m - 1]$ such that the following conditions hold:

- we have $z_1 \prec x_1 \leq_{cl} x_2 \prec z_2$, the set $X' = \{x : x_1 \leq_{cl} x \leq_{cl} x_2\}$ is non-empty and is contained in X , and the elements z_1, z_2 are not in X ,
- the set $X' \cup \{z_1, z_2\}$ is contained in either $B[i - 2, i + 2]$ or in $A[i - 2, i + 2]$ and we have $z_1 \in B[i - 2, i]$ and $z_2 \in B[i, i + 2]$ if $X' \cup \{z_1, z_2\} \in B[i - 2, i + 2]$, and similarly for the other case.

Note that such a choice of z_1, x_1, x_2, z_2 and i is possible as the sets $A[j, j + 3]$ and $B[j, j + 3]$ are hole cuts for every $j \in [m - 1]$, by combining Proposition 11 and Proposition 7(3),(4). For the rest of the proof we assume $X' \subset B[i - 2, i + 2]$, $z_1 \in B[i - 2, i]$ and $z_2 \in B[i, i + 2]$ (see Figure 7 for an illustration).



■ **Figure 7** Illustration of the proof: the cycle C' is marked with a dashed line. The set X' is shaded.

Suppose Y' is the set consisting of all the neighbors of z_1 and z_2 ; that is, $Y' = N(z_1) \cap N(z_2)$. Clearly, we have $Y' \subset A[i - 2, i + 2]$. To complete the proof of the proposition we show that:

5:12 Vertex Deletion into Bipartite Permutation Graphs

- every element of Y' is a member of X ,
- $X' \cup Y'$ is a hole cut in G .

Then we have $X' \cup Y' = X$ by minimality of X and consequently $X \subset V[i-2, i+2]$. So, it remains to prove the claims about the set Y' .

Suppose we have $y \in Y'$ such that $y \notin X$. Since X is a minimal hole cut, $X \setminus \{x\}$ is not a hole cut, where x is some fixed element from X' . That is, there is a hole C' in $G - (X \setminus \{x\})$. Note that C' must contain x . Suppose $c'_0, \dots, c'_{\ell-1}$ for some $\ell \geq 9$ are consecutive vertices in C' chosen such that $c'_j <_{cl} c'_{j+2}$ for every $j \in [\ell-1]$ (indices are taken modulo ℓ). Now we pick $p, q \in [\ell-1]$ such that $c'_p <_{cl} z_1 \leq_{cl} c'_{p+2}$ and $c'_q \leq_{cl} z_2 <_{cl} c'_{q+2}$. Since $x \in C'$, we have $c'_{p+2} \leq_{cl} x$ and $c'_q \leq_{cl} x$. Note that c'_{p+1} is adjacent to z_1 and c'_{q+1} is adjacent to z_2 . Next we replace in C' all the vertices between c'_{p+2} and c'_q (this set includes x) with the vertices z_1, y, z_2 and we obtain a cycle C'' containing no elements from X . Clearly, we can easily find a hole among the elements from C'' that avoids all the elements from X . This yields a contradiction as X is a hole cut.

To prove the second claim, suppose there is a hole C' in $G - (X' \cup Y')$. By Proposition 11 there are $c_1, c_2, c_3 \in C'$ such that $c_1 <_{cl} X' <_{cl} c_3$ and $c_1, c_3 \in N(c_2)$. However, this yields $c_2 \in Y'$, which is a contradiction. ◀

4 Proof of Theorem 1

The aim of this section is to provide a complete proof of Theorem 1 using structural results from the previous section. Let us start by showing that the BIPARTITE PERMUTATION VERTEX DELETION problem can be decided in polynomial time on almost bipartite permutation graphs.

► **Lemma 13.** *Let (G, k) be an instance of BIPARTITE PERMUTATION VERTEX DELETION where G is an n -vertex almost bipartite permutation graph. Then BIPARTITE PERMUTATION VERTEX DELETION can be decided in time $\mathcal{O}(n^6)$.*

Proof. If G is a bipartite permutation graph, (G, k) is a **yes**-instance, thus, we are done in this case. If G is not connected, we can consider each connected component independently and, at the end, we compare k with the total number of deleted vertices over all components. Let G' be a connected r -vertex component of G such that G' is not a bipartite permutation graph (otherwise, clearly, no vertex needs to be deleted). Let $C = \{c_0, \dots, c_{m-1}\}$ be a shortest hole in G' (it exists as G' is not a bipartite permutation graph). It can be found in time $\mathcal{O}(r^6)$ as follows. We iterate over all possible four-element subsets $S = \{v_1, v_2, v_3, v_4\}$ of $V(G')$. For these S for which $G'[S]$ is an induced P_4 , with consecutive vertices v_1, v_2, v_3, v_4 , we construct a graph \widetilde{G}' by removing the vertices from $(N(v_2) \cup N(v_3)) \setminus \{v_1, v_4\}$ (note that v_2 and v_3 also get removed). Then we find a shortest v_1 - v_4 -path in \widetilde{G}' in time $\mathcal{O}(r^2)$.

By Proposition 12, every minimal hole cut X in G' is contained in the set $V' = V_{G'}[i-2, i+2]$ for some $i \in [m-1]$. Therefore, we may check all the possibilities where a minimal cut is contained. For every i , we run an algorithm for finding a maximum flow in the following digraph H_i .

Digraph H_i has the vertex set $V' \times \{\text{in}, \text{out}\} \cup \{s, t\}$ and arc set consisting of:

- all arcs of the form $(u, \text{out})(v, \text{in})$, where uv is an edge of $G'[V']$,
- $s(v, \text{in})$ if there exists $u \in V_{G'}[i-4, i-3]$ such that uv is an edge of G' ,
- $(u, \text{out})t$ if there exists $v \in V_{G'}[i+3, i+4]$ such that uv is an edge of G' ,
- $(u, \text{in})(u, \text{out})$ for all $u \in V'$.

Set capacities of arcs of the form $(u, \text{in})(u, \text{out})$ to 1 and capacities of all the remaining arcs to ∞ (practically $|V_{G'}|$). It is readily seen that minimum (s, t) -cut in the defined network H_i corresponds to minimum hole cut in $G'[V']$ (arc of unit capacity $(u, \text{in})(u, \text{out})$ naturally corresponds to the vertex u of G').

Therefore it remains to apply classical max-flow algorithm to each H_i for $i \in [m - 1]$ and remember the smallest size $k_{G'}$ of minimal (s, t) -cuts. This can be performed in time $\mathcal{O}(m \cdot (|V'| + 2) \cdot (|E_{G'[V']}| + 2|V'|)^2) = \mathcal{O}(r^6)$ [9]. Finally, (G, k) is a **yes**-instance if and only if the sum of remembered sizes $k_{G'}$ over the all considered connected components G' is at most k . Clearly, the total running time is $\mathcal{O}(n^6)$. ◀

We now propose the algorithm. Given an n -vertex graph $G = (V, E)$ and number k , we want to answer the BIPARTITE PERMUTATION VERTEX DELETION problem. We say that (G, k) is the *initial* instance. We split our algorithm into two parts. The first part consists of a branching algorithm for deletion to almost bipartite permutation graphs. The output of the first part is a set of instances (G', k') where G' is an almost bipartite permutation graph and $0 \leq k' \leq k$ (or **no**-answer is no such instance exists) such that the initial instance (G, k) is a **yes**-instance if and only if at least one of these instances is a **yes**-instance. We show that the overall time of the first phase is $\mathcal{O}(n^9 \cdot 9^k)$. In the second part, the algorithm runs an $\mathcal{O}(s^6)$ -time algorithm for BIPARTITE PERMUTATION VERTEX DELETION where the initial graph is already an almost bipartite permutation graph.

Let us start with the first part. We say that $X \subseteq V$ is a *forbidden set* if $G[X]$ is isomorphic to one of the graphs: $K_3, T_2, X_2, X_3, C_5, C_6, C_7, C_8, C_9$. We define the following rule.

Rule: Given an instance (G, k) , $k \geq 1$, and a minimal forbidden set X , branch into $|X|$ instances, $(G - v, k - 1)$ for each $v \in X$.

Starting with the initial instance, the algorithm applies the rule exhaustively. In other words, the algorithm is a branching tree with leaves corresponding to instances (G', k') where $k' = 0$ or G' is an almost bipartite permutation graph. Clearly, as at least one vertex from each forbidden set must be removed from G , the initial instance is a **yes**-instance if and only if at least one of the leaves is a **yes**-instance.

The algorithm continues to the second part only with such leaves (G', k') that G' is an almost bipartite permutation graph (as otherwise, the leaf is **no**-instance). It runs the algorithm described in Lemma 13 to find if G' can be transformed into a bipartite permutation graph by using at most k' vertex deletions. It either finds a **yes**-instance or concludes after checking all the instances that there is no solution; that is, the initial instance is a **no**-instance.

We note that such a branching into a bounded number of smaller instances is a standard technique, see e.g., [24] for more details.

We now analyze the running time of the whole algorithm. In the first part, observe that the branching tree has depth at most k and has at most 9^k leaves, as k decreases by one whenever the algorithm branches and each of the listed forbidden subgraphs has at most nine vertices. Therefore the total number of nodes in the branching tree is $\mathcal{O}(9^k)$. Moreover, in each node (G'', k'') , the algorithm works in time $\mathcal{O}(n^9)$ as it checks if G'' contains a forbidden set. In the second part, the algorithm does a work $\mathcal{O}(n^6)$ in each leaf, by Lemma 13. We conclude that the total running time of our algorithm for BIPARTITE PERMUTATION VERTEX DELETION is $\mathcal{O}(9^k \cdot n^9)$.

5 Proof of Theorem 2

In this section, we provide a proof of Theorem 2. The idea of the algorithm is very similar to the FPT algorithm described in Section 4.

Let $G = (V, E)$ be a graph and let $Y \subseteq V$ be a subset of vertices of G such that $G - Y$ is a bipartite permutation graph. We want to construct a set $Z \subseteq V$ in polynomial time such that $G - Z$ is a bipartite permutation graph and $|Z| \leq 9|Y|$. We construct Z as follows. We start with $Z = \emptyset$. Then, as long as $G - Z$ contains a set X isomorphic to one of $K_3, T_2, X_2, X_3, C_5, C_6, C_7, C_8, C_9$ we add all vertices of X to Z . Observe that $Y \cap X \neq \emptyset$ and $|X| \leq 9$.

After this step $G - Z$ is an almost bipartite permutation graph. Note that $|Z| \leq 9|Z \cap Y|$. We find a shortest hole $C = \{c_0, \dots, c_{m-1}\}$ in $G - Z$ and find a minimum hole cut X as described in Section 4. Since $(Y - Z)$ is a hole cut in $G - Z$ we have $|X| \leq |Y - Z|$. We add X to Z . Observe that $G - Z$ is a bipartite permutation graph.

Since $K_3, T_2, X_2, X_3, C_5, C_6, C_7, C_8, C_9$ have at most 9 vertices, we have that $|Z| \leq 9|Y|$. This implies that the above algorithm is a 9-approximation algorithm. It runs in polynomial time because finding small forbidden subgraphs can be done in polynomial time and finding minimum hole cut in an almost bipartite permutation graph can be done in polynomial time.

6 Conclusion

In this paper we investigate for the first time the modification problems in graph classes related to partial orders. Our main result says that the bipartite permutation vertex deletion problem is fixed parameter tractable. We leave open the following two questions that inspired our research.

► **Problem 1.** *What is the parameterized status of the vertex deletion problems to the class of permutation graphs and to the class of co-comparability graphs?*

We recall that, due to the result of Lewis and Yannakakis [17], both of these problems are NP-complete. One of the most important result of our work is the description of the structure of almost bipartite permutation graphs, which are defined as graphs which do not induce small graphs from the list of forbidden structures for bipartite permutation graphs. In a similar fashion we can define the class of *almost permutation* and *almost co-comparability graphs*. The next two questions seem very natural in order to solve Problem 1.

► **Problem 2.** *What is the structure of almost permutation and almost co-comparability graphs?*

We are aware that the two problems mentioned above can be quite difficult. Therefore, it is worth considering intermediate problems that may be easier to attack. One of the proposed simplifications relies on the transition from the world of graphs to the world of posets. The following *vertex deletion into two-dimensional posets* problem seems very natural in the context of our research: we are given in the input a poset P and a number k and we ask whether we can delete at most k points from P so that the remaining points induce a two-dimensional poset in P .

► **Problem 3.** *What is the parameterized status of the vertex deletion into two-dimensional poset problem?*

Since permutation graphs are co-comparability graphs of two-dimensional posets and since permutation graphs are both comparability and co-comparability graphs, the vertex deletion

into two-dimensional poset problem is equivalent to the vertex deletion into co-comparability graph (or into permutation graph) problem if we assume that only comparability graphs can be given in the input. The class of two-dimensional posets is very well understood; in particular, the list of minimal forbidden structures for this class of posets, which is still infinite, is known (obtained independently by Trotter and Moore [23] and by Kelly [15]). Of course, it is natural to ask the following question:

► **Problem 4.** *What is the structure of almost two-dimensional posets?*

Since the comparability graphs of posets do not contain odd holes of size ≥ 5 , we know the structure of almost two-dimensional posets that are bipartite. Indeed, these are the posets whose comparability graphs are almost bipartite permutation graphs embeddable into cylinder stripes. The last problem we want to ask is as follows:

► **Problem 5.** *Is there a polynomial kernel for the bipartite permutation vertex deletion problem?*

A positive answer to this question obtained by indicating so-called *irrelevant vertices* may give some hope to solve Problem 1 with the use of irrelevant vertex technique.

References

- 1 Kirby A Baker, Peter C Fishburn, and Fred S Roberts. Partial orders of dimension 2. *Networks*, 2(1):11–28, 1972.
- 2 Andreas Brandstädt and Dieter Kratsch. On the restriction of some *NP*-complete graph problems to permutation graphs. In *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9–13, 1985*, volume 199 of *Lecture Notes in Computer Science*, pages 53–62. Springer, 1985. doi:10.1007/BFb0028791.
- 3 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
- 4 Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10–12, 2016*, pages 1096–1115. SIAM, 2016. doi:10.1137/1.9781611974331.ch77.
- 5 Yixin Cao. Unit interval editing is fixed-parameter tractable. *Information and Computation*, 253:109–126, 2017.
- 6 Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016.
- 7 Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, pages 51–229, 2006.
- 8 Jitender S. Deogun and George Steiner. Polynomial algorithms for hamiltonian cycle in cocomparability graphs. *SIAM J. Comput.*, 23(3):520–552, 1994. doi:10.1137/S0097539791200375.
- 9 Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.
- 10 Tibor Gallai. Transitiv orientierbare graphen. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(1-2):25–66, 1967.
- 11 Martin Charles Golumbic, Doron Rotem, and Jorge Urrutia. Comparability graphs and intersection graphs. *Discret. Math.*, 43(1):37–46, 1983. doi:10.1016/0012-365X(83)90019-5.
- 12 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988. doi:10.1007/978-3-642-97881-4.
- 13 Pinar Heggenes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. *Theor. Comput. Sci.*, 511:172–180, 2013. doi:10.1016/j.tcs.2012.03.013.

5:16 Vertex Deletion into Bipartite Permutation Graphs

- 14 Pinar Hegghernes, Pim van 't Hof, Daniel Lokshtanov, and Jesper Nederlof. Computing the cutwidth of bipartite permutation graphs in linear time. *SIAM J. Discret. Math.*, 26(3):1008–1021, 2012. doi:10.1137/110830514.
- 15 David Kelly. The 3-irreducible partially ordered sets. *Can. J. of Math.*, 29:367–383, 1977.
- 16 C Lekkekerker and J Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51(1):45–64, 1962.
- 17 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.
- 18 Daniel Lokshtanov. Wheel-free deletion is $W[2]$ -hard. In Martin Grohe and Rolf Niedermeier, editors, *Parameterized and Exact Computation, Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings*, volume 5018 of *Lecture Notes in Computer Science*, pages 141–147. Springer, 2008. doi:10.1007/978-3-540-79723-4_14.
- 19 Federico Mancini. Graph modification problems related to graph classes. *PhD degree dissertation, University of Bergen Norway*, 2, 2008.
- 20 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. doi:10.1007/s00453-008-9233-8.
- 21 Amir Pnueli, Abraham Lempel, and Shimon Even. Transitive orientation of graphs and identification of permutation graphs. *Canadian Journal of Mathematics*, 23(1):160–175, 1971.
- 22 Jeremy P. Spinrad, Andreas Brandstädt, and Lorna Stewart. Bipartite permutation graphs. *Discret. Appl. Math.*, 18(3):279–292, 1987. doi:10.1016/S0166-218X(87)80003-3.
- 23 William T. Trotter and John I. Moore Jr. Characterization problems for graphs, partially ordered sets, lattices, and families of sets. *Discret. Math.*, 16(4):361–381, 1976. doi:10.1016/S0012-365X(76)80011-8.
- 24 Pim van 't Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013. doi:10.1007/s00453-012-9661-3.
- 25 Yngve Villanger, Pinar Hegghernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM J. Comput.*, 38(5):2007–2020, 2009. doi:10.1137/070710913.