

# On edge intersection graphs of paths with 2 bends

Martin Pergel<sup>1\*</sup> and Paweł Rzążewski<sup>2,3\*\*</sup>

<sup>1</sup> Department of Software and Computer Science Education, Charles University,  
Praha, Czech Republic. E-mail: perm@kam.mff.cuni.cz

<sup>2</sup> Institute of Computer Science and Control, Hungarian Academy of Sciences  
(MTA SZTAKI), Budapest, Hungary

<sup>3</sup> Warsaw University of Technology, Faculty of Mathematics and Information Science,  
Warszawa, Poland. E-mail: p.rzazewski@mini.pw.edu.pl

**Abstract.** An EPG-representation of a graph  $G$  is a collection of paths in a grid, each corresponding to a single vertex of  $G$ , so that two vertices are adjacent if and only if their corresponding paths share infinitely many points. In this paper we focus on graphs admitting EPG-representations by paths with at most 2 bends. We show hardness of the recognition problem for this class of graphs, along with some subclasses.

We also initiate the study of graphs representable by unaligned polylines, and by polylines, whose every segment is parallel to one of prescribed slopes. We show hardness of recognition and explore the trade-off between the number of bends and the number of slopes.

## 1 Introduction

The concept of *edge intersection graphs of paths in a grid (EPG-graphs)* was introduced by Golumbic *et al.* [7]. By an EPG-representation of a graph  $G$  we mean a mapping from vertices of  $G$  to paths in a grid, such that two vertices are adjacent if and only if their corresponding paths share a grid edge. As each graph can be represented in this way [7], it makes sense to consider representations with some restricted set of shapes. A usual parameterization is by bounding the number  $k$  of times each path is allowed to change the direction. Graphs with such a representation are called *k-bend graphs*. So far, the case of 1-bend graphs received most attention [7, 4].

Since 0-bend graphs are just *interval graphs*, they can be recognized in polynomial time [1]. The recognition of 1-bend graphs is NP-complete [8], even if the representation is restricted to any prescribed set of 1-bend objects [4]. However, the problem becomes trivially solvable when  $k$  is at least the maximum degree of the input graph [8]. Thus it is unclear whether  $k$ -bend graphs are hard to recognize for all  $k \geq 2$ .

It is worth mentioning the closely related notion of  *$B_k$ -VPG-graphs*. These graphs are defined as intersection graphs of axis-aligned paths with at most  $k$  bends. So, unlike in the EPG-representation, paths that share a finite number

---

\* Partially supported by a Czech research grant GAČR GA14-10799S.

\*\* Supported by ERC Starting Grant PARAMTIGHT (No. 280152).

of points define adjacent vertices. Chaplick *et al.* [5] showed it is NP-complete to recognize  $B_k$ -VPG-graphs, for all  $k \geq 0$ .

In this paper we explore the problem of recognition of subclasses of EPG-graphs. Namely, we show that it is NP-complete to recognize 2-bend graphs. We also consider some restrictions, where we permit just some types of the curves in an EPG-representation (similarly to [4]). One of these restrictions, i.e., *monotonic EPG-representations*, where each path ascends in rows and columns, was already considered by Golubic *et al.* [7]. Our hardness proof even shows that between monotonic 2-bend graphs and 2-bend graphs, no polynomially recognizable class can be found.

The class of 2-bend graphs can be perceived as a generalization of quite well-studied class of 1-bend graphs. We also consider some generalizations of the concept of EPG-representations. We do not require individual segments to be axis-aligned, but we permit them to use any slope. We call such graphs *unaligned EPG-graphs* and study the number of bends in this setting. After this generalization, we may ask about particular restrictions. These restrictions are represented by restricting number of slopes that segments may use or even by using just prescribed shapes (in a flavor similar to [4]).

For unaligned EPG-graphs, we show that it is NP-hard to determine whether a graph is an unaligned 2-bend graph (hardness of the recognition for 1-bend graphs follows from [4]).

Having introduced unaligned EPG-graphs, we observe that there is a trade-off between the number of bends and the number of slopes used in a representation. We also show that representing an unaligned 2-bend graph on  $n$  vertices, may require using  $\Omega(\sqrt{n})$  slopes. This result follows from our hardness reduction.

## 2 Preliminaries

For an EPG-representation of a graph  $G$ , by  $P_v$  we shall denote the path representing a vertex  $v$ . Often we shall identify the vertex  $v$  with  $P_v$ . For example, if we say that two paths are adjacent, we mean that they share infinitely many points (note that if two paths *intersect*, one common point is enough).

A central notion in the study of EPG-graphs is the *bend number*. The bend number of a graph  $G$ , denoted by  $b(G)$ , is the minimum  $k$ , such that  $G$  has an EPG-representation, in which every path changes its direction at most  $k$  times. W.l.o.g. we can assume that every path in a  $k$ -bend EPG-representation bends exactly  $k$  times [4].

Each 2-bend path will be classified as *vertical* or *horizontal*, if its middle segment is resp. vertical or horizontal. This middle segment will be called the *body* of the path, while the remaining two segments will be referenced as its *legs*.

For a set  $X$  of shapes of polylines (i.e., piecewise-linear curves), by  $X$ -graphs we shall denote the class of graphs admitting an EPG-representation, in which the shape of every path is in  $X$  (similar notation was used in [4]). So for example monotonic 2-bend graphs are exactly  $\{\_ \_ , \_ \_ \}$ -graphs.

Golubic *et al.* [7] analyzed the structure of cliques in 1-bend graphs and proved that in 1-bend graphs each clique  $C$  is either an *edge clique* or a *claw-clique*. A maximal edge clique consists of vertices whose representing paths share

a common grid edge. A *claw* is a set of three distinct grid edges sharing a single endpoint and a maximal claw-clique consists of all paths containing two out of three edges of a given claw. Since we can safely assume that each 1-bend representation of a graph with  $n$  vertices can be embedded in a  $2n \times 2n$  grid, we obtain that the number of maximal cliques in a 1-bend graph is at most  $O(n^2)$ , i.e., is polynomial in  $n$ . This is no longer the case with 2-bend graphs.

Let  $n$  be an integer and let  $K_{2n}^-$  be the *cocktail-party graph*, i.e., a complete graph on  $2n$  vertices with a perfect matching removed. It is clear that  $K_{2n}^-$  has  $2^n = 2^{|V(K_{2n}^-)|/2}$  maximal cliques. Fig. 1 (left) shows that  $K_{2n}^-$  is a 2-bend graph.

**Proposition 1.** *2-bend graphs can have exponentially many maximal cliques.*

The restricted structure of cliques in 1-bend graphs follows from the fact that the 1-bend paths representing pairwise adjacent vertices must all share at least one grid point. It is easy to observe that cliques in 2-bend graphs do not have such a simple structure. One could be inclined by Fig. 1 (left) that every maximal clique is contained in the union of two edge-cliques or claw-cliques (a similar situation appears in unit disk graphs and is the main ingredient of a polynomial algorithm for CLIQUE in these graphs [3]). However, Fig. 1 (right) shows it is not true.

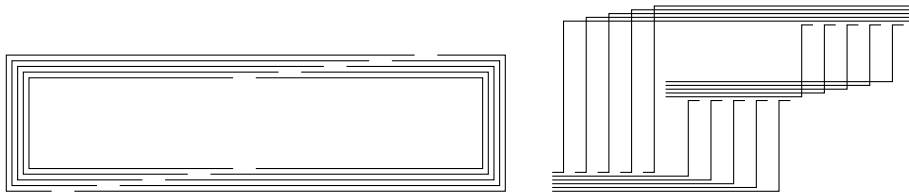


Fig. 1: **Left:**  $K_{10}^-$  as 2-bend graph. **Right:** A clique is not contained in two edge-cliques.

### 3 Aligned 2-bend graphs

The main results of this section is the following complexity result.

**Theorem 1.** *It is NP-complete to decide if a given graph is a 2-bend graph.*

*Proof.* The NP-membership is obvious. As a polynomial certificate we use a list of coordinates denoting start- and end-points of straight-line segments. Such a representation has polynomial size w.r.t. the given graph.

For the NP-hardness we use a polynomial reduction from PURE-NAE-3-SAT. The instance of this problem is a set of clauses, each containing three variables. We ask for the existence of a truth assignment, such that each clause contains at least one true variable and at least one false variable (we say that such a clause is *satisfied*). The problem is NP-complete and equivalent to 2-coloring of 3-uniform hypergraphs [9].

For a given formula  $\varphi$ , we shall construct a graph  $G$ , which is a 2-bend graph iff the formula is satisfiable. We start by replicating  $\varphi$  21 times (each time over

a distinct copy of the set of variables), obtaining an equivalent formula  $\varphi'$ . The reason of this operation will be made clear in a while.

We start the construction of  $G$  with two special vertices  $a$  and  $b$ . Then for each variable  $i$  of  $\varphi$ , we add a vertex  $v_i$  adjacent to both  $a$  and  $b$ . For each occurrence of  $i$  in a clause  $z$  of  $\varphi'$ , we add another vertex  $o_{i,z}$ , adjacent to  $a$ ,  $b$ , and  $v_i$ . Finally, for each clause  $z = (i, j, k)$  we add mutually non-adjacent vertices  $c_z$ ,  $d_z$ ,  $e_z$ , and  $f_z$ , with the following neighbors:  $N(c_z) = \{o_{i,z}, o_{j,z}, o_{k,z}\}$ ;  $N(d_z) = \{o_{i,z}, o_{j,z}\}$ ;  $N(e_z) = \{o_{i,z}, o_{k,z}\}$ ; and  $N(f_z) = \{o_{j,z}, o_{k,z}\}$  (see Fig. 2 (left)).

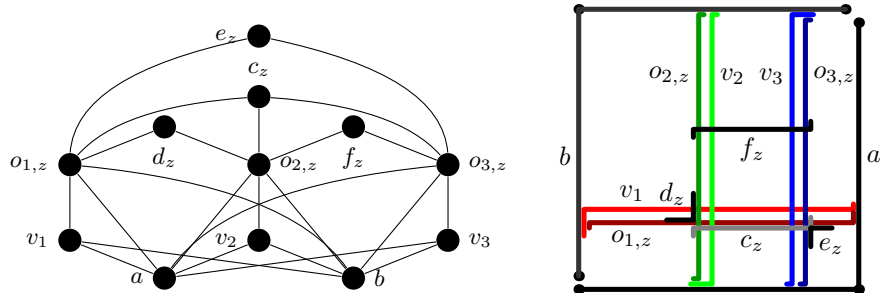


Fig. 2: **Left:** The graph obtained from a formula consisting of a single clause  $z = (1, 2, 3)$ . For clarity we did not replicate the formula. **Right:** An EPG-representation of the graph on the left. The variable 1 is false, while 2 and 3 are true.

Now let us explain the main ideas behind the reduction. The purpose of vertices  $a$  and  $b$  is to cover the legs of each  $P_{v_i}$  and  $P_{o_{i,z}}$ , keeping just their bodies exposed for possible intersections with clause-vertices. This assumption may fail, as some  $P_{v_i}$  or  $P_{o_{i,z}}$  can be positioned over an end of a segment of  $P_a$  or  $P_b$ , or on an intersection point of  $P_a$  and  $P_b$ . However, each end can be used at most once and each intersection point at most twice (see Fig. 3). As  $P_a$  and  $P_b$

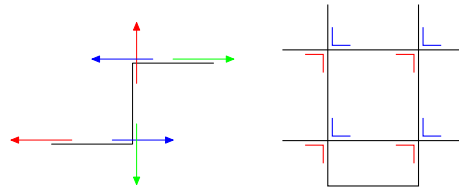


Fig. 3: **Left:** 6 pairwise non-adjacent segments may exit a 2-bend path without having to bend inside it. **Right:** At most 8 pairwise non-adjacent 2-bend paths may be adjacent to both  $P_a$  and  $P_b$  and contain their intersection point.

have (together) 12 ends of segments and at most 4 intersection points, we have at most 20 special situations. But since we replicated  $\varphi$  21 times, we are sure that for at least one copy of  $\varphi$  our assumption holds (this type of trick we call the “quantitative trick” and we use it to cope with some obstructions which may appear only a constant number of times). Let us focus on this “clean” copy of  $\varphi$  in  $\varphi'$ . One leg of each  $P_{o_{i,j}}$  is adjacent to  $a$  and the other one is adjacent to  $b$ . Also, at least one of them has to be adjacent to  $P_{v_i}$ , since otherwise clause-vertices

would be adjacent to  $v_i$ . Thus the body of  $P_{o_{i,j}}$  is exposed for representing clause-related vertices. Moreover, the orientation of the body (and this of whole path) is the same as the orientation of  $P_{v_i}$ , so all variable-occurrences are “synchronized”. The orientation of the paths will decide on truth assignment (horizontal means false, vertical means true).

First we show irrepresentability of the graph for an unsatisfiable formula. Let  $z = (i, j, k)$  be an unsatisfied clause. We will show that it cannot be represented. Observe that it is impossible to have a 2-bend path adjacent to three parallel, pairwise non-collinear segments, while it is possible for two parallel and one perpendicular segments (see Fig. 4 left).

The situation with three parallel segments corresponds to all-true or all-false clause. So, if no pair of middle segments of  $P_{o_{i,z}}, P_{o_{j,z}}, P_{o_{k,z}}$  (and thus  $P_{v_i}, P_{v_j}, P_{v_k}$ ) is collinear, we cannot represent  $c_z$ .

However, it might still happen that the bodies of, say,  $P_{o_{i,z}}$  and  $P_{o_{j,z}}$  are lying on the same line. But this pair of segments cannot be adjacent to more than one 2-bend path (see Fig. 4 (middle)). So if we represent  $c_z$ , then we cannot represent  $d_z, e_z$ , or  $f_z$ . This shows irrepresentability of an unsatisfied clause.

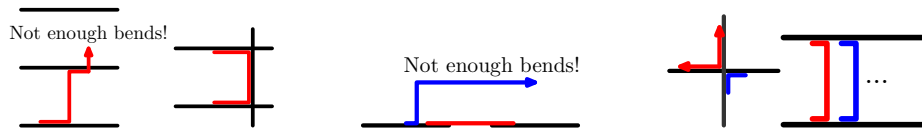


Fig. 4: **Left:** It is impossible to intersect three parallel, pairwise non-collinear segments with a 2-bend path, while two parallel and one perpendicular segments can be intersected. **Middle:** Two collinear segments cannot be adjacent to two mutually non-adjacent 2-bend paths. **Right:** This is possible for two mutually intersecting segments or two non-collinear parallel segments.

For a representable formula, we build a canonical representation shown in Fig. 2 (right) (for a clause with one false and two true literals, we rotate everything except of  $a$  and  $b$  by 90 degrees). Figure 2 shows one clause in one of (21) replicated copies and one occurrence of each variable. The full construction with all 21 copies would consist of 21 copies of all items present in the picture, except for  $P_a$  and  $P_b$ . Note that there are no edges between vertices belonging to different copies of  $\varphi$ . Further occurrences, e.g., of  $v_2$  in the same formula can be represented next to  $o_{2,z}$  intersecting  $v_2$  in the bottom (or top) horizontal leg (where it simultaneously intersects  $a$  (or  $b$ , respectively and it has to avoid legs of other possible occurrences). Anyway, their truth assignments are “synchronized” in all possible cases as they have to intersect  $a$  or  $b$  together with the vertex representative  $v_2$ . Considering two (and more) clauses in the representation, each clause has its own occurrences, so the representation of one clause does not influence representations of other clauses (as representatives of distinct occurrences are not mutually adjacent, i.e., they are disjoint up to finitely many points). In this representation, the body of each  $P_{o_{i,z}}$  intersects the body of each  $P_{o_{j,z}}$ , for all  $i$  evaluated to true and  $j$  evaluated to false. Thus it is possible to represent all clause-vertices, just as depicted.  $\square$

### 3.1 Subclasses of aligned 2-bend graphs

Here we focus on the recognition of particular subclasses of 2-bend graphs. Note that as there are many classes (whose recognition is often NP-hard), it is important to ask whether even some polynomially recognizable class can exist “in between”. This concept is called *sandwiching*. Formally, having two classes of graphs  $\mathcal{A} \subseteq \mathcal{B}$ , a class  $\mathcal{C}$  is *sandwiched between  $\mathcal{A}$  and  $\mathcal{B}$*  if  $\mathcal{A} \subseteq \mathcal{C} \subseteq \mathcal{B}$ . For optimization problems, it holds that if an algorithm works for class  $\mathcal{B}$ , it works also for the class  $\mathcal{A}$ . Also a hardness result for  $\mathcal{A}$  carries over to  $\mathcal{B}$ . However, the recognition problem behaves in a different way. As a trivial example we may pick a class  $\mathcal{A}$  containing only complete graphs (this class is polynomially recognizable), for class  $\mathcal{B}$  we may take class of all graphs (which is also polynomially recognizable) and between them we can find, e.g., classes of 2-bend graphs, whose recognition is NP-complete, as shown in Theorem 1. Similarly, between two NP-hard classes, a polynomially-recognizable class can be sandwiched (consider e.g. 3-colorable planar graphs, planar graphs, and 4-colorable graphs).

In this section we do not only show the recognition hardness of individual classes, but we are trying to find the smallest class  $\mathcal{A}$  and the largest class  $\mathcal{B}$ , such that no polynomially-recognizable class can be sandwiched between them.

We start with first two subclasses where our reduction for 2-bend graphs can be applied directly. One of them is a class of monotonic 2-bend graphs (i.e.,  $\{\lrcorner, \ulcorner\}$ -graphs) and the other is the class of  $\{\lrcorner, \urcorner\}$ -graphs.

We observe that in the proof of Theorem 1 we produce a monotonic 2-bend graph from each satisfiable formula. As a non-satisfiable formula cannot be represented by any 2-bend graph, if there was a polynomially-recognizable class between monotonic 2-bend graphs and 2-bend graphs, we would be able to distinguish satisfiable formulae from non-satisfiable ones, showing  $P=NP$ .

It is very simple to redraw the representation used in the proof of Theorem 1, using only  $\lrcorner$  and  $\urcorner$ -shapes.

**Corollary 1.** *It is NP-complete to recognize monotonic 2-bend graphs and  $\{\lrcorner, \urcorner\}$ -graphs. Moreover, between 2-bend graphs and any of these classes, or even their intersection, no polynomially recognizable class can be sandwiched (unless  $P=NP$ ).*

Now we shall modify the construction a bit to show a cascade of further results. Note that there are four possible patterns of horizontal paths ( $\sqcap, \sqcup, \lrcorner, \urcorner$ ) and another four for vertical paths ( $\sqcup, \sqcap, \ulcorner, \llcorner$ ) and another four for vertical paths. As we want to show that it is NP-complete to recognize graphs of any class  $X \in \{\sqcap, \sqcup, \lrcorner, \urcorner\} \times \{\sqcup, \sqcap, \ulcorner, \llcorner\}$ , we need to start with exploring the symmetries, to classify possible classes  $X$ .

So consider a pair of shapes, one of which is horizontal and the other one is vertical. If both legs of each shape bend in the same direction, we obtain the class  $\{\sqcap, \sqcup\}$ , which is equivalent to each  $\{\sqcup, \sqcup\}$ ,  $\{\sqcap, \sqcap\}$ , and  $\{\sqcup, \sqcap\}$  (consider a rotation of flipping of an EPG-representation). If both legs of one shape bend in the same direction, and the legs of the other shape bend in opposite directions, we get the class  $\{\sqcap, \ulcorner\}$  (again, up to symmetry). Finally, if the legs of both shapes bend in opposite directions, we get two possibilities, i.e.,  $\{\lrcorner, \ulcorner\}$  (monotonic 2-bend graphs) and  $\{\lrcorner, \llcorner\}$ . Although for the latter two classes we have already shown

NP-hardness, now we show yet one construction that works for all four cases. Such a general construction is important from the point of view of sandwiching.

The new construction, in fact, is just a simplified version of the one in the proof of Theorem 1. Again, for a formula  $\varphi$ , we replicate it to obtain  $\varphi'$  (using “quantitative trick”) and introduce variable-vertices  $v_i$  and occurrence-vertices  $o_{i,z}$ . The difference is that now each clause  $z = (i, j, k)$  is represented by just one vertex  $c_z$ , adjacent to  $o_{i,z}$ ,  $o_{j,z}$ , and  $o_{k,z}$  (so we omit vertices  $d_z, e_z$ , and  $f_z$ ). For a formula  $\varphi$ , let us call such constructed graph  $G(\varphi')$ .

Using this construction we can show that it is NP-complete to recognize  $X$ -graphs for each of the pairs  $X$  of permitted shapes, one of which is vertical and the other horizontal.

**Lemma 1.** *It is NP-complete to recognize  $X$ -graphs, for any  $X \in \{\sqcap, \sqcup, \sqsubset, \sqsupset\} \times \{\sqcap, \sqcup, \sqsubset, \sqsupset\}$ .*

Note that the lemma above shows that, both, an intersection and a union of the mentioned subclasses (as well as anything sandwiched between them) is NP-hard to get recognized. Also, note that it does not show that all classes representable by a given subset of 2-bend shapes (which includes at least one vertical and at least one horizontal shape) are NP-complete to get recognized. It still may happen that there exists such a set  $X$  of patterns, that  $X$ -graphs can be polynomially recognized. However, we know that if such a class exists, it must not contain even the intersection of  $\{\sqsubset, \sqsupset\}$ -graphs and  $\{\sqsubset, \sqsupset\}$ -graphs.

Finally, let us try to explore limits of the original hardness reduction for 2-bend graphs (Theorem 1). We know that it works for 2-bend graphs, for  $\{\sqsubset, \sqsupset\}$ -graphs, and for  $\{\sqsubset, \sqsupset\}$ -graphs (and where the inclusion-relation applies, then also for everything in between). However, we may show that the reduction works also for all triples of 2-bend shapes, in which at least one shape is vertical, at least one is horizontal, and they are not symmetric to the triple  $\{\sqcup, \sqsubset, \sqsupset\}$ , i.e., w.l.o.g., two vertical shapes, one having its legs in the same direction, the other having legs in mutually opposite directions, and the legs of the horizontal one go in the same direction and yet in the direction “towards the common angle” of the other two gadgets. It is easy to observe that the “simplified” construction can be represented, so we need to show, for a particular satisfied clause  $z = (i, j, k)$ , how to represent vertices  $d_z, e_z$ , and  $f_z$ . Suppose w.l.o.g.  $i, j$  are evaluated true and  $k$  is evaluated false. The path  $P_{c_z}$  passes through the intersection point of  $P_{o_{i,z}}$  and  $P_{o_{k,z}}$ , and through the intersection point of  $P_{o_{j,z}}$  and  $P_{o_{k,z}}$ . In order to represent  $d_z$  (adjacent to  $o_{i,z}$  and  $o_{j,z}$ ) we need to use the same intersection-point, i.e., we need the angle obtained from  $c_z$  rotated by 180 degrees. The case analysis shows that this is possible.

As a corollary of the previous statement, the reduction works for all such 4-tuples of 2-bend shapes, where at least one shape is vertical and at least one horizontal (non-trivial situation arises only when extending  $\{\sqcup, \sqsubset, \sqsupset\}$ ). Note also that the reduction works for any  $k$ -tuple of 2-bend shapes for  $k \geq 5$  (as there are just 4 vertical and 4 horizontal shapes, we are sure that at least one will be horizontal and at least one will be vertical).

Summing up the results from this section, we obtain the following.

**Theorem 2.** *It is NP-complete to recognize  $X$ -graphs, where  $X$  is:*

- (i) any of  $\{\square, \sqcup, \sqcap, \sqsubset\} \times \{\square, \sqsupset, \sqsubseteq, \sqsupseteq\}$ ,
- (ii) any triple of 2-bend shapes containing at least one vertical and one horizontal shape, and is not symmetric to  $\{\square, \sqsubseteq, \sqsupseteq\}$ .
- (iii) any 4-tuple of 2-bend shapes, containing at least one horizontal and one vertical shape.
- (iv) any  $k$ -tuple of 2-bend shapes for  $k \geq 5$ .

Moreover, one cannot sandwich any polynomially recognizable class between:

- (a) the intersection of  $\{\square, \sqcup, \sqcap, \sqsubset\} \times \{\square, \sqsupset, \sqsubseteq, \sqsupseteq\}$
  - (b) intersection of classes given in (ii),
- and the class of 2-bend graphs.

## 4 More slopes

In this section we relax the definition of an EPG-representation. By an *unaligned EPG-representation* of a graph  $G$  we mean a mapping from vertices of  $G$  to a set of polylines (piecewise linear curves), such that two vertices are adjacent iff their corresponding polylines share infinitely many points. Again, we are interested in keeping the number of bends (or equivalently, segments in a polyline) small.

Here we show hardness of the recognition of unaligned 2-bend graphs and conclude the section with discussion of a trade-off between the number of slopes used and the number of bends.

**Theorem 3.** *It is NP-hard to recognize unaligned 2-bend graphs.*

*Proof.* This time we reduce from 3-COLORING. For a graph  $G$  we shall construct a graph  $H$ , which is an unaligned 2-bend graph iff  $G$  is 3-colorable.

The reduction uses ideas similar to the reduction for aligned 2-bend graphs. This time we use 12 service vertices and again we want our gadgets to avoid being represented over the ends of segments of these service vertices, and over their mutual intersection points. So we use the “quantitative trick” again. This time we may have no more than 1 260 special places ( $12 \cdot 2 \cdot 3$  ends of segments,  $\binom{12}{2} \cdot 9$  possible intersection points, each of which can be used at most twice). Thus we take 1 261 disjoint copies of the graph  $G$ , obtaining the graph  $G'$  (clearly  $G'$  is 3-colorable iff  $G$  is 3-colorable).

The main idea of the reduction is that one service vertex of  $H$ , named  $a$ , simulates the 3-coloring of  $G'$ . The individual segments of  $P_a$  correspond to three color classes. Each vertex  $v$  of  $G'$  will be represented by several vertices of  $H$ . One of them, called  $v_2$ , will have the property that one of the legs of  $P_{v_2}$  lies on a segment of  $P_a$  (thus defining the color of  $v$  in a 3-coloring of  $G'$ ), and the remaining two segments of  $P_{v_2}$  will be fully covered by some other paths, non-adjacent to edge-representatives. An edge  $uv$  of  $G'$  will be represented by a pair of mutually non-adjacent vertices of  $H$ . Both of them will be made adjacent to  $a$  and the representatives of both  $u$  and  $v$ . The main idea is that we cannot construct edge-representatives, if  $v_2$  and  $u_2$  are adjacent to the same segment of  $a$  (and thus  $v$  and  $u$  get the same color). This part of  $H$  is illustrated in Fig. 5.



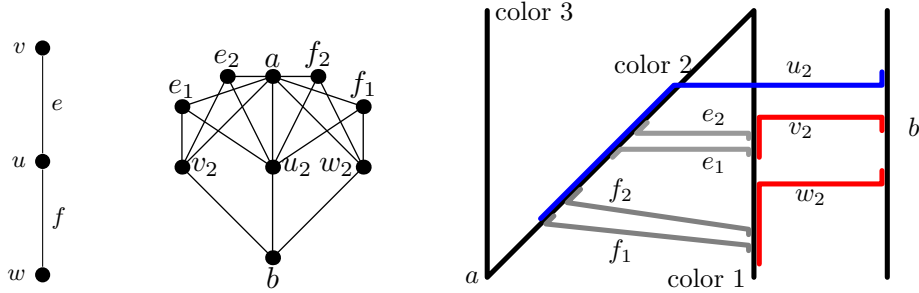


Fig. 5: **Left:** The graph  $G$ . **Middle:** The main part of  $H$ . For clarity, just the main vertex-representants are depicted. Also the replication (“quantitative trick”) was not performed. **Right:** An unaligned 2-bend representation of  $H$ . Note that we are unable to represent the edge  $vw$  (having fixed representations of  $v_2$  and  $u_2$ ).

Formally, the graph  $H$  has 12 service vertices  $a_0, a_{0.5}, a_1, a_{1.5}, a_2, a_{2.5}, a_3, a_{3.5}, a, b, a_B$ , and  $b_B$ . For each vertex  $v$  of  $G'$ , we add to  $H$  vertices  $v_1, v_{1.5}, v_2, v_{2.5}, v_3$ , and  $v_b$  (we will call them  $v$ -vertices). The vertex  $v_b$  is adjacent to all other  $v$ -vertices. Furthermore,  $v_{1.5}$  is adjacent to  $v_1, v_2$ , and  $v_{2.5}$  is adjacent to  $v_2, v_3$ . Finally, each  $v$ -vertex is adjacent to two service vertices:  $v_1$  to  $a_0, a_1$ ,  $v_{1.5}$  to  $a_{0.5}, a_{1.5}$ ,  $v_2$  to  $a, b$ ,  $v_{2.5}$  to  $a_{2.5}, a_{3.5}$ ,  $v_3$  to  $a_2, a_3$ . For each edge  $e = uv$  we add a pair of mutually non-adjacent vertices  $e_1, e_2$ , both adjacent to  $a, u_2$ , and  $v_2$ .

Suppose we have an unaligned 2-bend representation of  $H$ . First, by the “quantitative trick”, we know that at least for one copy of  $G$ , for any vertex  $v$ , all vertices  $v_i$  ( $i \in \{1, 1.5, 2, 2.5, 3, b\}$ ) are represented by 2-bend paths having both legs covered by the segments of the appropriate pair of service vertices. Let us focus on this copy of  $G$ .

We observe that the body of  $P_{v_2}$  (for any  $v$ ) is covered by (at least)  $P_{v_b}$ . This follows from the fact that  $P_{v_b}$  can intersect the other  $v$ -vertices only by its body (as one leg lies on  $P_{a_B}$ , and the second on  $P_{b_B}$ ). Thus the bodies of  $P_{v_1}, P_{v_{1.5}}, \dots, P_{v_3}, P_{v_b}$  must form an interval representation of  $H[\{v_1, v_{1.5}, \dots, v_3, v_b\}]$  and in no such representation the body of  $P_{v_2}$  can exceed the body of  $P_{v_b}$ . Therefore the body of  $P_{v_2}$  is fully covered by (at least) the body of  $P_{v_b}$ .

Now, we are in a desired situation. Consider an edge  $e = uv$ . For each  $P_{u_2}$  and  $P_{v_2}$ , only the leg lying on  $P_a$ , can be made adjacent to both  $P_{e_1}$  and  $P_{e_2}$ , as using any other segment would cause some unwanted adjacency. If these legs are on distinct segments of  $P_a$ , obviously we can represent both  $e_1$  and  $e_2$ . Conversely, if they are on the same segment of  $P_a$ , we can represent at most one of them (similarly to Fig. 4 (left)). This shows irrepresentability for a non-3-colorable  $G$ .

On the other hand, if  $G$  has a 3-coloring, we use it for distributing segments of  $P_{v_2}$  of each vertex  $v$  over the segments of  $P_a$ . Note that we may create a representation, where the bodies of  $P_{v_2}$ , for all  $v$ , are parallel. Then other  $v$ -vertices may be represented in the way shown in Fig. 6. For any edge  $e$ , paths  $P_{e_1}$  and  $P_{e_2}$  connect two non-collinear segments, which can be easily done.  $\square$

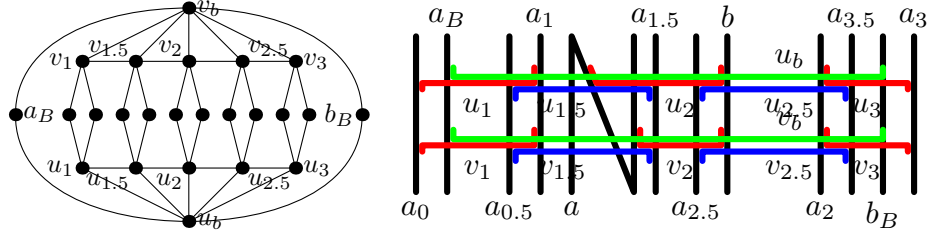


Fig. 6: **Left:** A graph  $H$  for  $G$  being an edge  $uv$  (replication is omitted). Unlabeled vertices between  $a_B$  and  $b_B$  are, respectively:  $a_0, a_1, a_{0.5}, a_{1.5}, a, b, a_{2.5}, a_{3.5}, a_2, a_3$ . **Right:** Unaligned 2-bend representation of  $H$ .

#### 4.1 Slopes and bends

Defining unaligned bend graphs permits us to introduce a new measure of complexity of a representation, i.e., the number of slopes used. There is an obvious trade-off between the number of bends and the number of slopes. Before we explore this relation a little more, let us try to minimize the number of different slopes used by the unaligned 2-bend representation.

**Proposition 2.** *In order to represent all unaligned 2-bend graphs on  $n$  vertices, we need  $\Omega(\sqrt{n})$  slopes.*

*Proof.* The proof follows from the construction in the proof of Theorem 3. Let  $G \sim K_{m,m,1}$  be a complete bipartite graph with bipartition classes  $X, Y$ , both of size  $m$ , and one extra vertex  $z$  adjacent to all other ones.

We replicate  $G$  1261 times, obtaining  $G'$ , and construct  $H$  in the way described in the proof of Theorem 3. Since  $G$  has  $2m+1$  vertices and  $\Theta(m^2)$  edges,  $H$  has  $n = \Theta(m^2)$  vertices.

As  $G$  is 3-colorable,  $H$  has an unaligned 2-bend representation. As always, we will focus on the “clean” copy of  $G$ . Consider the path  $P_a$ , and let  $p, q, r$  denote its three segments. By the properties of  $H$ , w.l.o.g. one leg of every  $P_{x_2}$  for  $x \in X$  lies on  $p$ , while one leg of every  $P_{y_2}$  for  $y \in Y$  lies on  $r$ .

Now consider the paths  $P_{e_1}$  (for  $e = xy$ ,  $x \in X$ ,  $y \in Y$ ). There are  $m^2$  such paths. We observe that every slope  $\ell$  can be used by the bodies of at most  $2m$  paths  $P_{e_1}$ . To see this, we use a sweeping line, parallel to  $\ell$ . As each path  $P_{e_1}$  connects a pair of segments of a different pair  $(P_{x_2}, P_{y_2})$ , the sweeping line must leave at least one of the segments before meeting a new one. As there are in total  $2m$  segments of  $P_{x_2}$  or  $P_{y_2}$  on  $P_a$ , at most  $2m$  paths  $P_{e_1}$  can have their bodies parallel to  $\ell$ . Thus we need at least  $\lceil \frac{m^2}{2m} \rceil = \Theta(m) = \Theta(\sqrt{n})$  different slopes to represent the bodies of paths  $P_{e_1}$ .  $\square$

To see a trade-off between the number of bends and the number of slopes, observe that the for  $G \sim K_{m,m,1}$ , the graph  $H$  can be easily represented by 3-bend paths, using only 2 slopes ( $P_a$  is represented by a  $\square$ -shape with segments of  $P_{v_2}$  on three different segments of it).

## 4.2 $d$ -bend number

Let us conclude the section with some generalization of the bend number. Fix a set  $D$  of  $d$  pairwise non-parallel lines (slopes) containing the origin point. We say that an unaligned EPG-representation is a  $EPG(D)$ -representation if every segment of each polyline is parallel to some line in  $D$ .

The  $d$ -bend number  $b_d(G)$  of a graph  $G$  is the minimum  $k$  for which there exists a set  $D$  of  $d$  slopes, such that  $G$  has an  $EPG(D)$ -representation in which every path bends at most  $k$  times. We also define  $b_\infty(G) := \min_{d \in \mathbb{N}} b_d(G)$ , which corresponds to unaligned EPG-representations.

Observe that the 2-bend number is just the classical bend number. It is also straightforward to observe that if  $d_1 < d_2$ , then  $b_{d_1}(G) \geq b_{d_2}(G)$  for all graphs  $G$ . Moreover, if there exists  $d \in \mathbb{N}$  such that  $b_d(G) = 0$ , then  $b_{d'}(G) = 0$  for all  $d' \in \mathbb{N}$  (as this means that  $G$  is an interval graph).

As we have seen in Proposition 2, introducing more slopes may help us reduce the number of bends needed to represent a given graph. Here we show two more examples of this. Consider a *wheel graph*  $W_n$  on  $n+1$  vertices ( $n \geq 3$ ). It follows from the work of Golumbic *et al.* [7] that  $W_n$  is not a 1-bend graph (using 2 slopes only) and one can easily find a representation using 2 bends. On the other hand, for  $d \geq 3$ , we can represent  $W_n$  using 1-bend paths (see Fig. 7 (left)). Thus  $b_2(W_n) = 2$  and  $b_d(W_n) = 1$  for all  $d \geq 3$ .

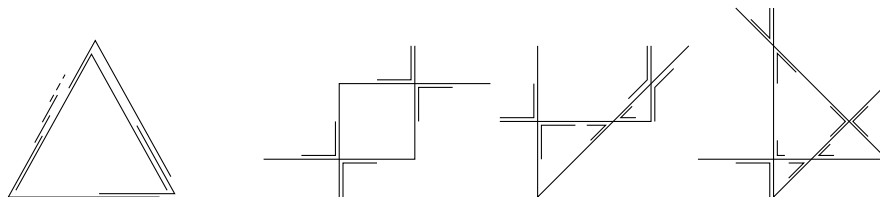


Fig. 7: **Left:** Representation of a wheel using 1-bend paths. **Right:** Representations of  $K_{2,s}$  with 1-bend paths.

Another examples of graphs with bend number depending on the number of slopes are complete bipartite graphs. Consider e.g. a graph  $K_{2,s}$ . When only 2 slopes are available, then  $K_{2,s}$  has a 1-bend representation only for  $s \leq 4$ . Introducing a third slope allows us to represent  $K_{2,5}$  and  $K_{2,6}$ . Fourth slope allows representing  $K_{2,7}$  and  $K_{2,8}$ . By analyzing the possible intersection points of two 1-bend paths, we observe that  $K_{2,s}$  for any  $s \geq 9$  does not have a 1-bend representation for any number of slopes. On the other hand, every  $K_{2,s}$  is a 2-bend graph on 2 slopes (see Fig. 7 (right) and Fig. 4 (right)).

## 5 Conclusions and open problems

Although all non-trivial classes of EPG-graphs are considered hard for recognition, not much is known. It is an open problem whether the recognition problem remains NP-hard for  $k$ -bend graphs (for  $k \geq 3$ ).

*Problem 1. Is the recognition of  $k$ -bend graphs NP-complete for every fixed  $k \geq 1$ ?*

For unaligned bend graphs and aligned bend graphs, using more than 2 slopes, naturally arises the question on inclusions between different classes. Also the complexity of the recognition problem is unknown (for more than 1 bend, when we restrict the number of slopes). Note that none of our reductions can be easily used. The unaligned version increases the number of slopes, while in the aligned version a new slope introduces a new “truth value”, but in a way that does not seem to be suitable for a reduction from any form of coloring.

As mentioned before, the CLIQUE problem is polynomially solvable in 1-bend graphs. On the other hand, the problem is shown to be NP-complete in 2-interval graphs [6]. Since every 2-interval graph is a 3-bend graph and also a 2-bend graph with 3 slopes, we know that the problem is NP-complete in these classes as well. The complexity for 2-bend graphs remains open.

*Problem 2. What is the complexity of the CLIQUE problem in 2-bend graphs?*

It is not hard to observe that for any two sets  $D, D'$  with  $|D| = |D'| = 3$ , one can transform an  $EPG(D)$ -representation of any graph  $G$  to its  $EPG(D')$ -representation. However, it is not clear if the same holds for sets with at least 4 directions of slopes. It is worth mentioning that there are infinitely many classes of intersection graphs of segments, each of which is parallel to one of 4 slopes [2].

*Problem 3. Is the minimum number of bends (per path) in an  $EPG(D)$ -representation of a graph  $G$  always equal to  $b_d(G)$ , for any set  $D$  of  $d > 3$  slopes?*

Our generalization raises yet further questions. Especially, we may put individual vertices into points with integral coordinates. Now, we may ask, how large grid is necessary and sufficient to represent any graph with  $n$  vertices and prescribed number of permitted slopes, or even, with prescribed slopes.

## References

1. K. Booth, G. Lueker, Testing for the consecutive ones property, interval graphs, and planarity using  $PQ$ -tree algorithms. *J. Comput. System Sci.* 13, 1976, pp. 335–389.
2. J. Černý, D. Král, H. Nyklová, O. Pangrác, On Intersection Graphs of Segments with Prescribed Slopes. *GD 2001 Proc.*, LNCS 2265, 2002, pp. 261–271.
3. B. Clark, C. Colbourn, D. Johnson, Unit Disk Graphs. *Disc. Math.* 86, 1990, pp. 165–177.
4. K. Cameron, S. Chaplick, C. T. Hoang, Edge intersection graphs of  $L$ -shaped paths in grids. *Disc. Appl. Math.* (in press, available online), 2015.
5. S. Chaplick, V. Jelinek, J. Kratochvíl, T. Vyskočil, Bend-Bounded Path Intersection Graphs: Sausages, Noodles, and Waffles on a Grill. *WG 2012 Proc.*, LNCS 7551, 2012, pp. 274–285.
6. M. Francis, D. Gonçalves, P. Ochem, The Maximum Clique Problem in Multiple Interval Graphs. *Algoritmica* 71, 2015, pp. 812–836.
7. M. Golumbic, M. Lipshteyn, M. Stern, Edge intersection graphs of single bend paths on a grid. *Networks* 54, 2009, pp. 130 – 138.
8. D. Heldt, K. Knauer, T. Ueckerdt, Edge-intersection graphs of grid paths: The bend-number. *Disc. Applied Math.* 167, 2014, pp. 144 – 162.
9. L. Lovász, Coverings and coloring of hypergraphs. *4th SEICCGTC Proc.*, 1973, pp. 3–12.