



ELSEVIER

Contents lists available at ScienceDirect

## Theoretical Computer Science

[www.elsevier.com/locate/tcs](http://www.elsevier.com/locate/tcs)

## Recognition and drawing of stick graphs ☆

Felice De Luca<sup>a</sup>, Md Iqbal Hossain<sup>a</sup>, Stephen Kobourov<sup>a</sup>, Anna Lubiw<sup>b</sup>,  
Debajyoti Mondal<sup>c,\*</sup><sup>a</sup> Department of Computer Science, University of Arizona, USA<sup>b</sup> Cheriton School of Computer Science, University of Waterloo, Canada<sup>c</sup> Department of Computer Science, University of Saskatchewan, Canada

## ARTICLE INFO

## Article history:

Received 25 March 2019

Received in revised form 31 July 2019

Accepted 12 August 2019

Available online xxxx

Communicated by D.-Z. Du

## Keywords:

Graph Drawing

Bipartite Graphs

Intersection Graphs

Stick Graphs

## ABSTRACT

A *Stick graph* is an intersection graph of axis-aligned segments such that the left end-points of the horizontal segments and the bottom end-points of the vertical segments lie on a “ground line,” a line with slope  $-1$ . It is an open question to decide in polynomial time whether a given bipartite graph  $G$  with bipartition  $A \cup B$  has a Stick representation where the vertices in  $A$  and  $B$  correspond to horizontal and vertical segments, respectively. We prove that  $G$  has a Stick representation if and only if there are orderings of  $A$  and  $B$  such that  $G$ 's bipartite adjacency matrix with rows  $A$  and columns  $B$  excludes three small ‘forbidden’ submatrices. This is similar to characterizations for other classes of bipartite intersection graphs.

We present an algorithm to test whether given orderings of  $A$  and  $B$  permit a Stick representation respecting those orderings, and to find such a representation if it exists. The algorithm runs in time linear in the size of the adjacency matrix. For the case when only the ordering of  $A$  is given, or neither ordering is given, we present some partial results about graphs that are, or are not, Stick representable.

© 2019 Elsevier B.V. All rights reserved.

## 1. Introduction

Let  $\mathcal{O}$  be a set of geometric objects in the Euclidean plane. The *intersection graph* of  $\mathcal{O}$  is a graph where each vertex corresponds to a distinct object in  $\mathcal{O}$ , and two vertices are adjacent if and only if the corresponding objects intersect. Recognition of intersection graphs that arise from different types of geometric objects such as segments, rectangles, discs, intervals, etc., is a classic problem in combinatorial geometry. Some of these classes, such as interval graphs [2], can be recognized in polynomial-time, whereas many others are NP-hard [4,26,28]. There are many beautiful results that characterize intersection classes in terms of a vertex ordering without certain forbidden patterns, and recently, Hell *et al.* [21] unified many previous results by giving a general polynomial time recognition algorithm for all cases of small forbidden patterns.

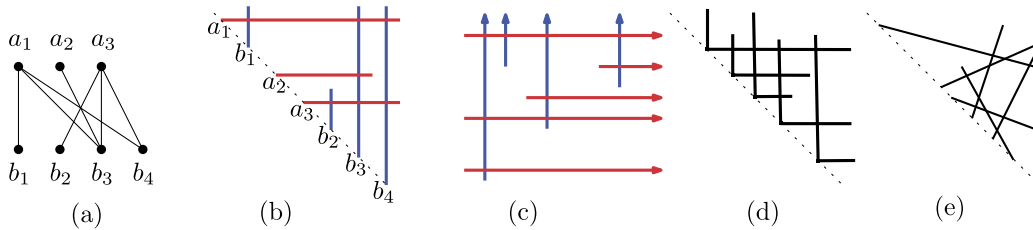
☆ A preliminary version of this work appeared in the proceedings of the 26th International Symposium on Graph Drawing and Network Visualization (GD2018), vol. 11282 of Lecture Notes in Computer Science, pp. 303–316, 2018.

\* Corresponding author.

E-mail addresses: [felicedeluca@email.arizona.edu](mailto:felicedeluca@email.arizona.edu) (F. De Luca), [hossain@email.arizona.edu](mailto:hossain@email.arizona.edu) (M.I. Hossain), [kobourov@cs.arizona.edu](mailto:kobourov@cs.arizona.edu) (S. Kobourov), [alubiw@uwaterloo.ca](mailto:alubiw@uwaterloo.ca) (A. Lubiw), [dmondal@cs.usask.ca](mailto:dmondal@cs.usask.ca) (D. Mondal).

<https://doi.org/10.1016/j.tcs.2019.08.018>

0304-3975/© 2019 Elsevier B.V. All rights reserved.



**Fig. 1.** (a) A bipartite graph  $G = (A \cup B, E)$ . (b) A Stick representation of  $G$ . (c)–(e) Illustration for different types of intersection representations. (c) A 2DOR representation. (d) A Hook representation. (e) A grounded segment representation.

In this paper we study a class of bipartite intersection graphs called *Stick graphs*. A *Stick graph* is an intersection graph of axis-aligned segments with the property that the left end-points of horizontal segments and the bottom end-points of vertical segments all lie on a *ground line*,  $\ell$ , which we take, without loss of generality, to be a line of slope  $-1$ . See Figs. 1(a)–(b). It is an open problem to recognize Stick graphs in polynomial time [9].

Stick graphs lie between two well-studied classes of bipartite intersection graphs. First of all, they are a subset of the *grid intersection graphs* (GIG) [20]—intersection graphs of horizontal and vertical segments in the plane—which are NP-complete to recognize [26]. When all the horizontal segments extend rightward to infinity and the vertical segments extend upward to infinity, we obtain the subclass of *2-directional orthogonal ray* (2DOR) graphs (e.g., see Fig. 1(c)), which can be recognized in polynomial time [31]. It is easy to show that every 2DOR graph is a Stick graph—truncate each ray at a ground line placed above and to the right of every intersection point (and then flip the picture upside-down). Thus the class of Stick graphs lies strictly between these two classes.

What the two classes (GIG and 2DOR) have in common is a nice characterization in terms of vertex orderings. A bipartite graph  $G$  with vertex bipartition  $A \cup B$  can be represented as a *bipartite adjacency matrix*,  $M(G)$  with rows and columns corresponding to  $A$  and  $B$ , respectively, and a 1 in row  $i$ , column  $j$ , if  $(i, j)$  is an edge. Both GIG graphs and 2DOR graphs can be characterized as graphs  $G$  for which  $M(G)$  has a row and column ordering without certain ‘forbidden’ submatrices. (Details below.) Many other bipartite intersection graphs can be similarly characterized in terms of forbidden submatrices, see [25].

One of our main results is a similar characterization of Stick graphs. Specifically, we will prove that a bipartite graph  $G$  with vertex bipartition  $A \cup B$  has a Stick representation with vertices of  $A$  corresponding to horizontal segments and vertices of  $B$  corresponding to the vertical segments if and only if there is an ordering of  $A$  and an ordering of  $B$  such that  $M(G)$  has no submatrix of the following form, where  $*$  stands for either 0 or 1:

$$\begin{bmatrix} * & 1 & * \\ * & 0 & 1 \\ 1 & * & * \end{bmatrix} \quad \begin{bmatrix} 1 & * \\ 0 & 1 \\ 1 & * \end{bmatrix} \quad \begin{bmatrix} * & 1 & * \\ 1 & 0 & 1 \end{bmatrix}$$

Although this characterization does not (yet) give us a polynomial time algorithm to recognize Stick graphs, it allows us to make some progress. Given a bipartite graph  $G$  with vertex bipartition  $A \cup B$ , we want to know if  $G$  has a Stick representation with  $A$  and  $B$  corresponding to horizontal and vertical segments, respectively. It is easy to show that a solution to this problem is completely determined by a total ordering  $\sigma$  of the vertices of  $G$  corresponding to the order (from left to right) in which the segments touch the ground line. A natural way to tackle the recognition of Stick graphs is as a hierarchy of problems, each (possibly) more difficult than the next:

- (i) **Fixed As and Bs:** In this case an ordering,  $\sigma_A$ , of the vertices in  $A$  and an ordering,  $\sigma_B$ , of the vertices in  $B$  are given, and the output ordering  $\sigma$  must respect these given orderings. Because of our forbidden submatrix characterization, this problem can be solved in polynomial time.
- (ii) **Fixed As:** In this case only the ordering  $\sigma_A$  is given.
- (iii) **General Stick graphs:** In this case, neither  $\sigma_A$  nor  $\sigma_B$  is given, i.e., there is no restriction on the ordering of the vertices.

**Our Results:** We give an algorithm with run-time  $O(|A||B|)$  for problem (i). This is faster than naively looking for the forbidden submatrices. (And in fact, we use our algorithm to prove the forbidden submatrix characterization). Furthermore, the algorithm will find a Stick representation when one exists.

For problems (ii) and (iii), recognizing Stick graphs with fixed As and in general, we give some conditions that ensure a graph is a Stick graph, and some conditions that ensure a graph is not a Stick graph.

**Related Work:** We now review the research related to the recognition of intersection graphs, in particular those that are bipartite.

Interval graphs, i.e., intersection graph of horizontal intervals on the real line, can be recognized in linear time [2,14]. Bipartite interval graphs with a fixed bipartition are known as *interval bigraphs* (IBG) [15,29], and can be recognized in polynomial time [29]. In contrast to the interval graphs, no linear-time recognition algorithm is known for IBG.

$$\gamma_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \gamma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \gamma_3 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \gamma_4 = \begin{bmatrix} 1 & 0 & 1 \\ * & 1 & * \end{bmatrix} \quad \gamma_5 = \begin{bmatrix} * & 1 & * \\ 1 & 0 & 1 \\ * & 1 & * \end{bmatrix}$$

Fig. 2. Forbidden submatrices, where \* stands for either 0 or 1.

Many bipartite graph classes have been characterized in terms of forbidden submatrices of the graph’s bipartite adjacency matrix, and a rich body of research examines when the rows and columns of a matrix can be permuted to avoid forbidden submatrices [25]. For example, a graph  $G$  is chordal bipartite if and only if  $M(G)$  can be permuted to avoid the matrix  $\gamma_1$  in Fig. 2 [25], which led to a polynomial-time algorithm [27].  $G$  is a bipartite permutation graph if and only if  $M(G)$  can be permuted to avoid  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  [12].

A graph is a *two-directional orthogonal ray* (2DOR) graph if it admits an intersection representation of upward and rightward rays [31,32]. A graph is a 2DOR graph if and only if its incidence matrix admits a permutation of its rows and columns that avoids  $\gamma_1$  and  $\gamma_2$  [31].

There is a linear-time algorithm to recognize 2DOR graphs [13,31]. If there are 3 or 4 allowed directions for the rays, then the graphs are called 3DOR or 4DOR graphs, respectively. Felsner *et al.* [17] showed that if the direction (right, left, up, or down) for each vertex is given, then the existence of a 4DOR representation respecting the given directions can be decided in polynomial time. If the horizontal elements are segments and the vertical elements are rays, then the corresponding intersection graphs are called *SegRay* graphs [7–9,24]. A graph  $G$  is a SegRay graph if and only if  $M(G)$  can be permuted to avoid  $\gamma_4$  [10]. The time-complexity questions for 3DOR, 4DOR and SegRay are all open.

The class of *segment graphs* contains the graphs that can be represented as intersections of segments (with arbitrary slopes and intersection angles). Every planar graph has a segment intersection representation [6]. Restricting to axis-aligned segments gives rise to *grid intersection graphs* (GIG) [26]. A bipartite graph is a GIG graph if and only if its incidence matrix admits a permutation of its rows and columns that avoids  $\gamma_5$  [20]. If all the segments must have the same length, then the graphs are known as *unit grid intersection graphs* (UGIG) [28]. The recognition problem is NP-complete for both GIG [26] and UGIG [28]. We note that 4DOR is a subset of UGIG but Stick is not [9].

Researchers have examined further restrictions on GIG. For example, the graphs that admit a GIG representation with the additional constraint that all the segments must intersect (or be “stabbed by”) a ground line form the *stabable grid intersection* (StabGIG) graph class [9].

Another class of intersection graphs that restricts the objects on a ground line is defined in terms of *hooks*. A *hook* consists of a center point on the ground line together with an incident vertical segment and horizontal segment above the ground line. *Hook* graphs are intersection graphs of hooks [5,22,33], e.g., see Fig. 1(d). Hook graphs are also known as *max point-tolerance graphs* [5] and *heterozygosity graphs* [19]. The bipartite graphs that admit a Hook representation are called BipHook [9]. The complexities of recognizing the classes StabGIG, BipHook, and Stick are all open [9]. Chaplick *et al.* [9] examined the containment relations of these graph classes.

*Grounded segment representations* are a generalization of Stick representations, where the segments can have arbitrary slopes, e.g., see Fig. 1(e). Note that the segments are still restricted to lie on the same side of the ground line. Cardinal *et al.* [4] showed that the problem of deciding whether a graph admits a grounded segment representation is  $\exists\mathbb{R}$ -complete. We refer to [3,4] for other related classes such as outersegment and outerstring graphs, and for the study of their containment relations.

The shorter version of this work [16] was presented at the 26<sup>th</sup> International Symposium on Graph Drawing and Network Visualization.

The following table summarizes the time complexities of recognizing different classes of bipartite intersection graphs, where  $n$  and  $m$  are the sizes of the two vertex sets of the bipartition.

## 2. Fixed As and Bs

In this section we study Stick representations of graphs with a fixed bipartition of the vertices and fixed vertex orderings for each vertex set. We call this problem  $\text{Stick}_{AB}$ , defined formally as follows.

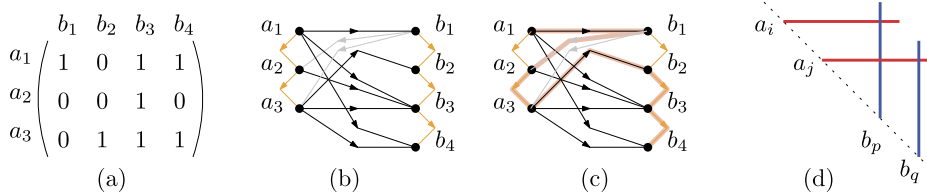
**Problem:** STICK REPRESENTATION WITH FIXED AS AND BS ( $\text{STICK}_{AB}$ )

**Input:** A bipartite graph  $G = (A \cup B, E)$ , an ordering  $\sigma_A$  of the vertices in  $A$ , and an ordering  $\sigma_B$  of the vertices in  $B$ .

**Question:** Does  $G$  admit a Stick representation such that the  $i$ th horizontal segment on the ground line  $\ell$  corresponds to the  $i$ th vertex of  $\sigma_A$  and the  $j$ th vertical segment on  $\ell$  corresponds to the  $j$ th vertex of  $\sigma_B$ ?

Graph class	Time complexity	Ref
Chordal Bipartite Graphs	$O((n+m)^2)$ , or $ E \log(n+m)$	[27,34]
Bipartite Permutation Graphs	$O(nm)$ -time	[35]
2-Directional Ray Graphs (2DOR)	$O(nm)$ -time	[13,31]
3- or 4-Directional Ray Graphs (3DOR, 4DOR)	Open	[13,31]
4-DOR with given directions for vertices	$f(n,m)$ -time <sup>a</sup>	[17]
3-DOR with a given bipartition $(A \cup B)$ , and an ordering for $A$ s, i.e., vertical rays	$O((n+m)^2)$ -time	[17]
Grid Intersection Graphs (GIG)	NP-complete	[26]
Unit Grid Intersection Graphs (UGIG)	NP-complete	[28]
Grounded Segment Intersection Graphs	$\exists\mathbb{R}$ -complete	[4]
StabGIG, SegRay, Hook, BipHook and Stick Graphs	Open	[9,23]

<sup>a</sup> Multiplication time for two  $(n+m) \times (n+m)$  matrices.



**Fig. 3.** (a) The incidence matrix  $M$  for the graph of Fig. 1(a). (b) The directed graph  $H$ . (c) A total order  $(a_1, b_1, a_2, a_3, b_2, b_3, b_4)$  of the vertices in  $H$ . The corresponding Stick drawing is in Fig. 1(b). (d) Illustration of a forbidden ordering if  $m_{j,p} = 0$ .

We first present an  $O(|A||B|)$ -time algorithm for  $Stick_{AB}$ . A Stick representation is totally determined by the order  $\sigma$  of the segments' intersection with the ground line (details in the proof of Lemma 1). Thus the idea of the algorithm is to impose some ordering constraints between the vertices of  $A$  and  $B$  based on some submatrices of the adjacency matrix of  $G$ . We show that the required Stick representation exists if and only if there exists a total order  $\sigma$  of  $(A \cup B)$  that satisfies the constraints and preserves the given orderings  $\sigma_A$  and  $\sigma_B$ . We now describe the details.

Assume that  $\sigma_A = (a_1, \dots, a_n)$  and  $\sigma_B = (b_1, \dots, b_m)$ . Let  $M$  be the ordered bipartite adjacency matrix of  $A$  and  $B$ , i.e.,  $M$  has rows  $a_1, \dots, a_n$  and columns  $b_1, \dots, b_m$ , where the entry  $m_{i,p}$ , i.e., the entry at the  $i$ th row and  $p$ th column, is 1 or 0 depending on whether  $a_i$  and  $b_p$  are adjacent or not, as illustrated in Fig. 3(a).

We start with the constraints  $a_{i-1} < a_i$ , where  $2 \leq i \leq n$ , and  $b_{p-1} < b_p$ , where  $2 \leq p \leq m$  to enforce the given orderings  $\sigma_A$  and  $\sigma_B$ . We now add some more constraints, as follows.

**C<sub>1</sub>:** If an entry  $m_{i,p}$  is 1, then add the constraint  $a_i < b_p$ , e.g., see the black edges in Fig. 3(b).

**C<sub>2</sub>:** If  $M$  contains an ordered submatrix  $\begin{matrix} & b_p & b_q \\ a_i & \begin{bmatrix} 1 & * \\ 0 & 1 \end{bmatrix} \\ a_j & \end{matrix}$ , then add the constraint  $b_p < a_j$ . For example, see the gray edges in Fig. 3(b).

We now test whether the set of constraints is consistent. Consider a directed graph  $H$  with vertex set  $(A \cup B)$ , where each constraint corresponds to a directed edge (Fig. 3(b)). Then the set of constraints is consistent if and only if  $H$  is acyclic, and the following lemma claims that this occurs if and only if the graph admits a Stick representation.

**Lemma 1.**  $G$  admits a Stick representation respecting  $\sigma_A$  and  $\sigma_B$  if and only if  $H$  is acyclic, i.e., the constraints are consistent.

**Proof.** We first show that the constraints are necessary. Every constraint between two vertices of the same set is implied by  $\sigma_A$  or  $\sigma_B$ . For Condition  $C_1$ , observe that a horizontal segment  $a_i$  can intersect a vertical segment  $b_p$  only if  $a_i$  precedes  $b_p$ , i.e., we must have  $a_i < b_p$ . For Condition  $C_2$ , we already have  $a_i < a_j, b_p < b_q, a_i < b_p, a_j < b_q$ . If we assume that  $a_j < b_p$ , then we have  $a_i < a_j < b_p < b_q$ , and to reach the vertical segment  $b_q, a_j$  would intersect  $b_p$ . Since  $m_{j,p} = 0$ , this intersection is forbidden. Fig. 3(d) illustrates this scenario. Therefore, we must have the constraint  $b_p < a_j$ . Since all the constraints are necessary, if  $G$  admits an intersection representation, then the set of constraints is consistent.

We now prove the converse. Suppose the set of constraints is consistent. Take a total order of  $A \cup B$  which is consistent with all the constraints, e.g., see Fig. 1(c). This is a “topological order” of  $H$ . Initiate the drawing of the corresponding orthogonal segments in this order on the ground line  $\ell$ . This determines the  $y$ -coordinate of every  $a \in A$  and the  $x$ -coordinate of every  $b \in B$ . For each vertex  $a \in A$ , let  $\max_B(a)$  be the neighbor of  $a$  in  $G$  with the largest index. We extend the horizontal segment corresponding to  $a$  to the right until the  $x$ -coordinate of  $\max_B(a)$ . Similarly, for each vertex  $b \in B$ , let  $\min_A(b)$  be the neighbor of  $b$  in  $G$  with the minimum index. We extend the vertical segment corresponding to  $b$  upward until the  $y$ -coordinate of  $\min_A(b)$ .

We must show that the resulting drawing does not contain any forbidden intersection. Suppose by contradiction that the segments of  $a_j$  and  $b_p$  intersect, but they are not adjacent in  $G$ , i.e.,  $m_{j,p} = 0$ . We now have  $a_j < b_p$ , and the entries

$b_q = \max_B(a_j)$  and  $a_i = \min_A(b_p)$  give the submatrix described in Condition  $C_2$ , thus the constraint  $b_p < a_j$  applies, a contradiction.

An algorithm to solve  $Stick_{AB}$  follows immediately, and can be implemented in linear time in the size of the adjacency matrix  $M$ .

**Theorem 1.** *There is an  $O(|A||B|)$ -time algorithm to decide the  $Stick_{AB}$  problem, and construct a Stick representation if one exists.*

**Proof.** The algorithm was given above: We construct the directed graph  $H$  from the 0-1 matrix  $M$  and test if  $H$  is acyclic. This correctly decides  $Stick_{AB}$  by Lemma 1. Furthermore, if  $H$  is acyclic, then we can construct a Stick representation as specified in the proof of Lemma 1. Pseudocode of this algorithm is given in Algorithm 1.

**Algorithm 1** Algorithm for Fixed- $A$ -Fixed- $B$ -Stick Recognition.

```

1: Input: A bipartite graph  $G = (A \cup B, E)$ , an ordering  $\sigma_A$  and an ordering  $\sigma_B$ 
2:  $M \leftarrow$  A matrix representation of  $G$  whose rows follow the ordering  $\sigma_A$ , and columns follow the ordering  $\sigma_B$ 
3:  $H \leftarrow$  A graph with vertex set  $(A \cup B)$  but without any edge.
4: for each consecutive vertices  $a_{i-1}, a_i$  in  $\sigma_A$  do add the edge  $(a_{i-1}, a_i)$  to  $H$ 
5: for each consecutive vertices  $b_{j-1}, b_j$  in  $\sigma_B$  do add the edge  $(b_{j-1}, b_j)$  to  $H$ 
6: for each entry  $m_{i,j} = 1$  in  $M$  do add the edge  $(a_i, b_j)$  to  $H$ 
7: for each  $a_i, a_j, b_p, b_q$  (respecting  $\sigma_A$  and  $\sigma_B$ ) of the form  $\begin{matrix} a_i & & b_p & b_q \\ a_j & \begin{bmatrix} 1 & * \\ 0 & 1 \end{bmatrix} \end{matrix}$  do add the edge  $(b_p, a_j)$  to  $H$ 
8: if  $H$  contains a cycle then return false //No solution exists
9:  $\sigma \leftarrow$  A topological ordering of the vertices of  $H$ 
10: return  $\sigma$ 
    
```

The matrix  $M$  has size  $O(nm)$  where  $n = |A|$  and  $m = |B|$ , and the graph  $H$  has  $n + m$  vertices and  $O(nm)$  edges. We can test acyclicity of a graph and find a topological ordering in linear time. Also, the construction of the Stick representation is clearly doable in linear time.

Thus we only need to give details on constructing  $H$  in time  $O(nm)$ . We can construct the edges of  $H$  that correspond to  $\sigma_A$  and  $\sigma_B$  in time  $O(n + m)$ . The edges arising from constraints  $C_1$  correspond to the 1's in the matrix  $M$ , so we can construct them in  $O(nm)$  time. The edges arising from constraints  $C_2$  correspond to some of the 0's in the matrix  $M$ . Specifically, a 0 in position  $m_{j,p}$  gives a  $C_2$  constraint  $b_p < a_j$  if and only if there is a 1 in row  $j$  to the right of the 0 and a 1 in column  $p$  above the 0. We can flag the 0's that have a 1 to their right by scanning each row of  $M$  from right to left. Similarly, we can flag the 0's that have a 1 above them by scanning each column of  $M$  from bottom to top. These scans take time  $O(nm)$ . Finally, if a 0 in  $M$  has both flags, then we add the corresponding edge to  $H$ . The total time is  $O(nm)$ .

Lemma 1 also yields a forbidden submatrix characterization for  $Stick_{AB}$ .

**Theorem 2.** *An instance of  $Stick_{AB}$  with graph  $G = (A \cup B, E)$  has a solution if and only if  $G$ 's ordered adjacency matrix  $M$  has no ordered submatrix of the following form:*

$$P_1 = \begin{matrix} & b_p & b_q & b_r \\ a_i & \begin{bmatrix} * & 1 & * \\ * & 0 & 1 \\ 1 & * & * \end{bmatrix} \\ a_j & & & \\ a_k & & & \end{matrix}, P_2 = \begin{matrix} & b_p & b_q \\ a_i & \begin{bmatrix} 1 & * \\ 0 & 1 \\ 1 & * \end{bmatrix} \\ a_j & & \\ a_k & & \end{matrix}, P_3 = \begin{matrix} & b_p & b_q & b_r \\ a_i & \begin{bmatrix} * & 1 & * \\ 1 & 0 & 1 \end{bmatrix} \\ a_j & & & \end{matrix}.$$

Observe that  $P_2$  and  $P_3$  are special cases of  $P_1$  with  $p=q$  and  $j=k$ , respectively.

**Proof.** We will use the graph  $H$  that we constructed above and used in Lemma 1. By Lemma 1, the theorem statement is equivalent to the statement that  $M$  has a submatrix  $P_1, P_2$  or  $P_3$  if and only if  $H$  has a directed cycle.

We first show that if the matrix  $M$  has one of the ordered submatrices  $P_1, P_2, P_3$  then  $H$  has a directed cycle. For  $P_1$ , the cycle in  $H$  is  $a_k < b_p$  (by  $C_1$ ),  $b_p < b_q$  (by  $\sigma_B$ ),  $b_q < a_j$  (by  $C_2$ ),  $a_j < a_k$  (by  $\sigma_A$ ). For  $P_2$ , the cycle is  $b_p < a_j$  (by  $C_2$ ),  $a_j < a_k$  (by  $\sigma_A$ ),  $a_k < b_p$  (by  $C_1$ ). For  $P_3$ , the cycle is  $b_q < a_j$  (by  $C_2$ ),  $a_j < b_p$  (by  $C_1$ ),  $b_p < b_q$  (by  $\sigma_B$ ).

To prove the other direction, suppose that  $H$  has a directed cycle  $O$ . We will show that  $M$  has one of the submatrices  $P_1, P_2, P_3$ . Let  $b_q$  be the rightmost vertex of  $O$  in  $\sigma_B$ , and let  $(b_q, z)$  be the outgoing edge of  $b_q$  in  $O$ . Since  $b_q$  is the rightmost vertex of  $O$  in  $\sigma_B$ ,  $z$  must be a vertex  $a_j$  of  $A$ . The constraint  $b_q < a_j$  can only be added by  $C_2$ . Therefore, we

must have the configuration  $\begin{matrix} & b_q & b_r \\ a_i & \begin{bmatrix} 1 & * \\ 0 & 1 \end{bmatrix} \\ a_j & & \end{matrix}$ . The path can now continue from  $a_j$  following zero or more  $A$  vertices, but

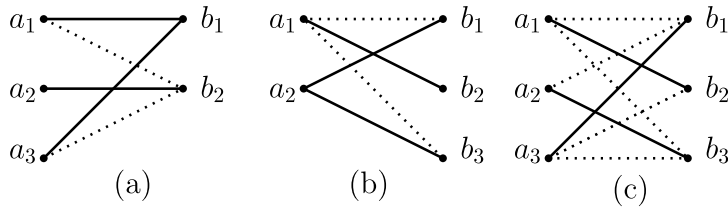


Fig. 4. The forbidden subgraphs for Theorem 3. Dotted edges are optional.

to complete the cycle, it eventually needs to reach a vertex  $b_p$  of  $B$ . Since  $b_q$  is the rightmost in  $\sigma_B$ ,  $b_p$  must appear either before  $b_q$  or coincide with  $b_q$ . First suppose that  $b_p \neq b_q$ . If the outgoing edge of  $a_j$  is  $(a_j, b_p)$ , then we obtain the configuration  $P_3$ . Otherwise, the path visits several vertices of  $A$  and then visits  $b_p$ , and we thus obtain the configuration  $P_1$ .

Suppose now that  $b_p = b_q$ . In this case the outgoing edge of  $a_j$  cannot be  $(a_j, b_p)$ , because such an edge can only be added by  $C_1$ , which would imply  $m_{j,p} = m_{j,q} = 1$ , violating the configuration above. If the path visits several vertices of  $A$  and then visits  $b_p (= b_q)$ , then there must be a 1 in the  $q$ th column below the  $j$ th row. We thus obtain the configuration  $P_2$ .

2.1. Bipartite graphs representable for all orderings:

The above forbidden submatrix characterization allows us to characterize the bipartite graphs  $G = (A \cup B, E)$  that have a Stick representation for every possible ordering of  $A$  and  $B$ . Observe that the forbidden submatrices  $P_2, P_3, P_1$  correspond, respectively, to the bipartite graphs shown in Figs. 4(a)–(c). We can construct  $2^2 = 4$  graphs from Fig. 4(a) based on whether each of the dotted edges is present or not. Similarly, we can construct  $2^2 = 4$  graphs from Fig. 4(b), and  $2^5 = 32$  graphs from Fig. 4(c). Let  $\mathcal{H}$  be the set that consists of these 40 graphs. From Theorem 2 we immediately obtain:

**Theorem 3.** A bipartite graph  $G = (A \cup B, E)$  admits a Stick representation for every possible ordering of  $A$  and  $B$  if and only if  $G$  does not contain any graph of  $\mathcal{H}$ .

3. Fixed As

In this section we study the Stick representation problem when the ordering of only the vertices in  $A$  is given. A formal description of the problem, which we call  $Stick_A$ , is as follows.

**Problem:** STICK REPRESENTATION WITH FIXED AS ( $Stick_A$ )  
**Input:** A bipartite graph  $G = (A \cup B, E)$ , and a vertex-ordering  $\sigma_A$  of  $A$ .  
**Question:** Does  $G$  admit a Stick representation such that the  $i$ th horizontal segment on the ground line corresponds to the  $i$ th vertex of  $\sigma_A$ ?

We give a non-trivial necessary condition for a graph to have a Stick representation, but we do not know whether this condition is sufficient.<sup>1</sup> The idea is to use the forbidden submatrix characterization for  $Stick_{AB}$  (Theorem 2). We need an ordering of the  $B$  vertices that, together with the given ordering  $\sigma_A$ , avoids the forbidden submatrices  $P_1, P_2, P_3$ . We will express the conditions for the ordering of the  $B$  vertices as a 2-SAT formula, i.e., a CNF (conjunctive normal form) formula where each clause contains at most two literals. A 2-SAT can be solved in polynomial time [1].

For each pair of vertices  $v, w$  of  $G$ , we create variables  $p_{v < w}$  and  $p_{w < v}$  (representing the ordering of segments  $v$  and  $w$  on the ground line). We will enforce  $p_{v < w} = \neg p_{w < v}$  by adding clauses  $(\neg p_{v < w} \vee \neg p_{w < v}) \wedge (p_{v < w} \vee p_{w < v})$ . (One variable would suffice, but it is notationally easier to have both.) We first set the truth values of all the variables involving two vertices of  $A$  based on  $\sigma_A$ . We then add a few other clauses based on  $P_1, P_2, P_3$ , as follows.

For every  $b_p, b_q, b_r$  giving rise to  $P_1$ , we add the clauses  $(\neg p_{b_q < b_r} \vee p_{b_q < b_p})$  and  $(\neg p_{b_p < b_q} \vee p_{b_r < b_q})$ . The first clause means that if  $b_q < b_r$ , then to avoid  $P_1$ , we must have  $b_q < b_p$ . Similarly, the second clause means if  $b_p < b_q$ , then to avoid  $P_1$ , we must have  $b_r < b_q$ . These clauses ensure that if the SAT formula has a solution, then no configuration of the form  $P_1$  can arise.

For every  $b_p, b_q$  giving rise to  $P_2$ , we set  $p_{b_q < b_p}$  to true. This would avoid any forbidden configuration of the form  $P_2$  in a solution of the 2-SAT formula.

<sup>1</sup> The conference version [16] of the paper claimed the necessary condition to be sufficient, but we found a gap in the proof for sufficiency, which was also pointed out by Michael A. Bekos.

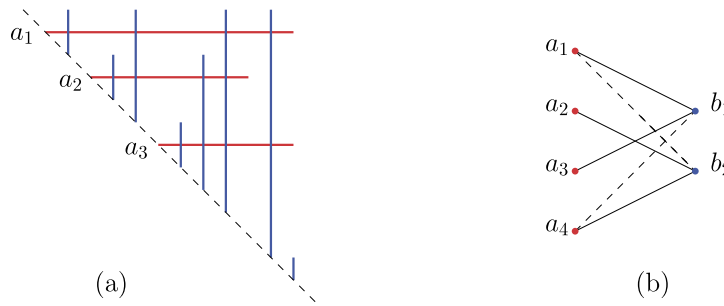


Fig. 5. (a) Illustration for Remark 1. (b) Illustration for Remark 2.

Finally, for every  $b_p, b_q, b_r$  giving rise to  $P_3$ , we add the clauses  $(\neg p_{b_q < b_r} \vee p_{b_q < b_p})$  and  $(\neg p_{b_p < b_q} \vee p_{b_r < b_q})$ . Note that these clauses can be interpreted in the same way as for  $P_1$ , i.e., if the 2-SAT formula has a solution, then no configuration of the form  $P_3$  can arise.

Let  $F$  be the resulting 2-SAT formula. If  $F$  does not have a solution, then there does not exist any ordering of the  $B$ s that avoids the forbidden patterns. Thus  $G$  does not admit the required Stick representation. We thus have the following theorem.

**Theorem 4.** *If  $Stick_A$  has a solution, then the corresponding 2-SAT formula  $F$  must be satisfiable.*

We do not know whether this condition of Theorem 4 is sufficient. The satisfiability of  $F$  may not guarantee a Stick representation. However, we are not aware of any graph which does not have a Stick representation but the corresponding 2-SAT formula is satisfiable.

Recently, Chaplick et al. [11] have developed an  $O(|A||B|)$ -time algorithm to solve  $Stick_A$ . Their algorithm does not use the forbidden matrix characterization, rather it relies on a data structure that carefully maintains all possible ‘realizable permutations’.

**Bipartite Graphs Representable for All A Orderings:** We also considered the class of bipartite graphs  $G = (A \cup B, E)$  such that for every ordering of the vertices of  $A$  there exists a Stick representation. We will call this the  $Stick_{\forall A}$  class. Although we do not have a characterization of the  $Stick_{\forall A}$  class, we describe some positive and negative instances below in Remark 1 and Remark 2.

**Remark 1.** Any bipartite graph  $G = (A \cup B, E)$  with at most three vertices in  $A$  belongs to the  $Stick_{\forall A}$  class.

**Proof.** The proof is by construction. For any ordering  $\sigma_A$ , we can categorize the columns into at most eight categories (assuming  $A$  has exactly three vertices, the other cases are straightforward). We then organize those categories in the

following order: 
$$\begin{matrix} & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$
. It is now straightforward to see that the resulting ordering corresponds to the required Stick representation, e.g., see Fig. 5(a).

**Remark 2.** A graph does not belong to  $Stick_{\forall A}$  if its bipartite adjacency matrix contains the submatrix 
$$\begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{matrix} \begin{bmatrix} 1 & * \\ 0 & 1 \\ 1 & 0 \\ * & 1 \end{bmatrix}$$
. (Here the columns are unordered.)

**Proof.** We refer the reader to Fig. 5(b). Any  $\sigma_A$  consistent with the given matrix would serve as a negative instance, as

follows. The adjacency matrix 
$$\begin{matrix} & b_1 & b_2 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} & \begin{bmatrix} 1 & * \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix}$$
 would require  $b_2$  to come after  $b_1$ , while the matrix 
$$\begin{matrix} & b_1 & b_2 \\ \begin{matrix} a_2 \\ a_3 \\ a_4 \end{matrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ * & 1 \end{bmatrix} \end{matrix}$$
 would require  $b_1$  to come after  $b_2$ . Thus there is no consistent ordering for the vertices of  $B$ , and hence  $G$  does not have a Stick representation respecting  $\sigma_A$ .

4. Stick graphs

In this section we examine general Stick representations, i.e., we do not impose any constraints on the ordering of the vertices.

**Problem:** STICK REPRESENTATION  
**Input:** A bipartite graph  $G = (A \cup B, E)$ .  
**Question:** Does  $G$  admit a Stick representation such that the vertices in  $A$  and  $B$  correspond to horizontal and vertical segments, respectively?

It is an open question to find a polynomial time algorithm for the above problem of recognizing Stick graphs. In this section we give some partial results (Remarks 4–5).

We begin with some definitions. A matrix has the *simultaneous consecutive ones* property if the rows and columns can be permuted so that the 1’s in each row and each column appear consecutively [30]. A *one-sided drawing* of a planar bipartite graph  $G = (A \cup B, E)$  is a planar straight-line drawing of  $G$ , where all vertices in  $A$  lie on the  $x$ -axis, and the vertices of  $B$  lie strictly above the  $x$ -axis [18].

**Remark 3.** Let  $G = (A \cup B, E)$  be a bipartite graph and let  $M$  be its adjacency matrix, where the rows and columns correspond to  $A$ s and  $B$ s, respectively. If  $M$  has the simultaneous consecutive ones property, then  $G$  admits a Stick representation, which can be computed in  $O(|A||B|)$  time.

**Proof.** One can determine whether  $M$  has the simultaneous consecutive ones property in  $O(|A||B|)$  time [30], and if so, then one can construct such a matrix  $M'$  within the same time complexity. We now show how to construct the Stick representation from  $M'$ . For each row (resp., column), we draw a horizontal (resp., vertical) segment starting from the rightmost (resp., topmost) 1 entry. We extend the horizontal segments to the left and vertical segments downward such that they touch a ground line  $\ell$ , e.g., see Figs. 6(a)–(b).

Let the resulting drawing be  $D$ , which may contain many unnecessary crossings. However, for each unnecessary crossing, we can follow the segments involved in the crossings upward and rightward to find two distinct 1 entries. Since the matrix has the simultaneous ones property, the violated entries in each row (column) must lie consecutively at the left end of the row (bottom end of the column). Therefore, one can find a  $(+x, -y)$ -monotone path  $P$  that separates the violated entries from the rest of the matrix, e.g., see the shaded region in Fig. 6(b).

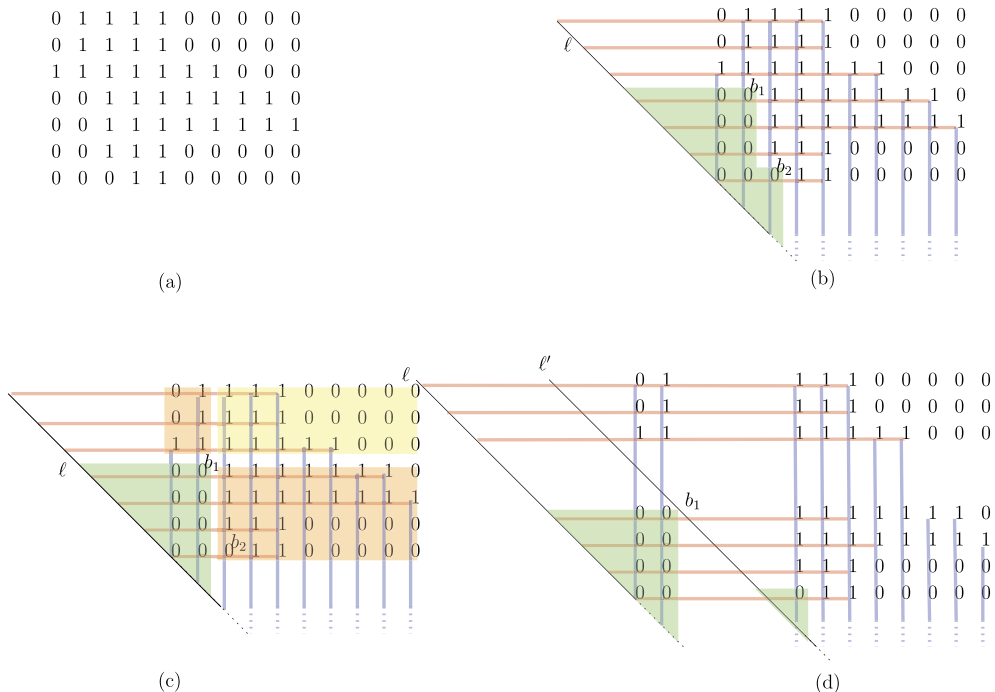


Fig. 6. Illustration for Remark 3.



Let  $b_1, b_2, \dots, b_k$  be the bend points creating  $90^\circ$  angles towards  $\ell$ . To compute the required Stick representation, we remove these bends one after another, as follows. Consider the topmost bend point  $b_i$ , e.g., see  $b_1$  in Fig. 6(c). Imagine a Cartesian coordinate system with origin at  $b_i$ . Move the rows above  $b_i$  and columns to the right of  $b_i$  towards the upward and rightward directions, respectively, as illustrated in Fig. 6(d). It is straightforward to observe that one now can construct a ground line  $\ell'$  through  $b_i$  such that the violated entries lie in the region below the path determined by  $b_{i+1}, \dots, b_k$ .

**Remark 4.** Let  $G = (A \cup B, E)$  be an  $n$ -vertex bipartite graph that admits a one-sided planar drawing. Then  $G$  is a Stick graph, and its Stick representation can be computed in  $O(n^2)$  time.

**Proof.** Given an  $n$ -vertex bipartite graph  $G = (A \cup B, E)$ , one can decide whether  $G$  admits a one-sided planar drawing in  $O(n^2)$  time [18], and if so, then one can construct such a drawing within the same time complexity.

It now suffices to show how to construct a Stick representation from the one-sided drawing. Let  $\Gamma$  be a one-sided drawing of  $G$ . Imagine that we connected the  $A$ -vertices in a path from left to right in order they appear on the  $x$ -axis such that all the edges of the path are on the outerface. We define the *upper set* of  $\Gamma$  to be the  $B$  vertices that lie on the outerface, including their neighbors.

We start by drawing the upper set. It is straightforward to draw a Stick representation by first drawing the  $A$ -vertices in the order they appear on the upper set from top to bottom on the ground line, and then the  $B$ -vertices to complete the Stick representation of the set. Fig. 7(a) illustrates a one-sided drawing, and Fig. 7(c) illustrates a drawing of its upper set.

Let  $\Gamma'$  be the one-sided drawing by deleting the  $B$ -vertices of degree one from the upper set of  $\Gamma$ . Note that every  $B$ -vertex  $w$  of degree two or more on the upper set, ‘covers’ smaller one-sided drawings below the edges that connect  $w$  to a pair of consecutive  $A$ -neighbors, e.g., see the shaded regions of Fig. 7(b). We continue drawing each of these smaller one-sided drawings recursively between their corresponding leftmost and rightmost  $A$ -vertices, which have already been drawn in the previous level. Fig. 7(d) illustrates the drawing in the second recursion level, and Figs. 7(e)–(f) illustrates the third recursion level.

**Remark 5.** Let  $H$  be the graph obtained by deleting a perfect matching from a complete bipartite graph  $K_{4,4}$ . Any graph  $G = (A \cup B, E)$  containing  $H$  as an induced subgraph does not admit a Stick representation. Since  $H$  is a planar graph, not all planar bipartite graphs are Stick graphs.

**Proof.** It suffices to show that  $H$  (Fig. 8(a)) does not admit a Stick representation. It is straightforward to permute the

rows and columns of the matrix representation of  $H$  such that all zeros lie on the main diagonal, e.g.,  $M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$ .

Therefore, for every permutation of rows and columns, we can pick the two columns that have zeros in the middle two

rows to obtain the configuration  $\begin{bmatrix} 1 & * \\ 0 & 1 \\ 1 & 0 \\ * & 1 \end{bmatrix}$ . As we discussed in Sec. 3, there cannot be any Stick representation with this

row ordering. Consequently, the graph  $H$ , as well as no graph containing  $H$  as an induced subgraph, can have a Stick representation.

Since  $H$  is a planar graph (see Fig. 8(b)), not all planar bipartite graphs are Stick graphs.

### 5. Open problems

We conclude the paper with some open problems related to our research.

**Open Problem 1.** What is the complexity of recognizing Stick graphs? Is the problem NP-complete? By Theorem 2 the problem is equivalent to ordering the rows and columns of a 0-1 matrix to exclude the 3 forbidden submatrices given in the Theorem statement. Note that these forbidden submatrices involve 5 or 6 rows and columns (vertices of the graph) so the results of Hell et al. [21], which apply to patterns of at most 4 vertices in a bipartite graph, do not provide a polynomial time algorithm.

One possible approach using 3-SAT is as follows. Given a bipartite graph  $G = (A \cup B, E)$ , one can create a 3-SAT formula  $\Phi$  such that  $\Phi$  is satisfiable if and only if  $G$  admits a Stick representation, as follows. For each pair of vertices  $v, w$  of  $G$ , create variables  $p_{v < w}$  and  $p_{w < v}$  (representing the ordering of  $v$  and  $w$  on the ground line), and add clauses  $(\neg p_{v < w} \vee \neg p_{w < v}) \wedge (p_{v < w} \vee p_{w < v})$  to enforce  $p_{v < w} = \neg p_{w < v}$ . Now express the conditions C1 and C2 from Section 2 as 3-SAT clauses.

$\Phi_1$ : (Condition C1.) If  $m_{i,p} = 1$ , then set  $p_{a_i < b_p} = 1$ .

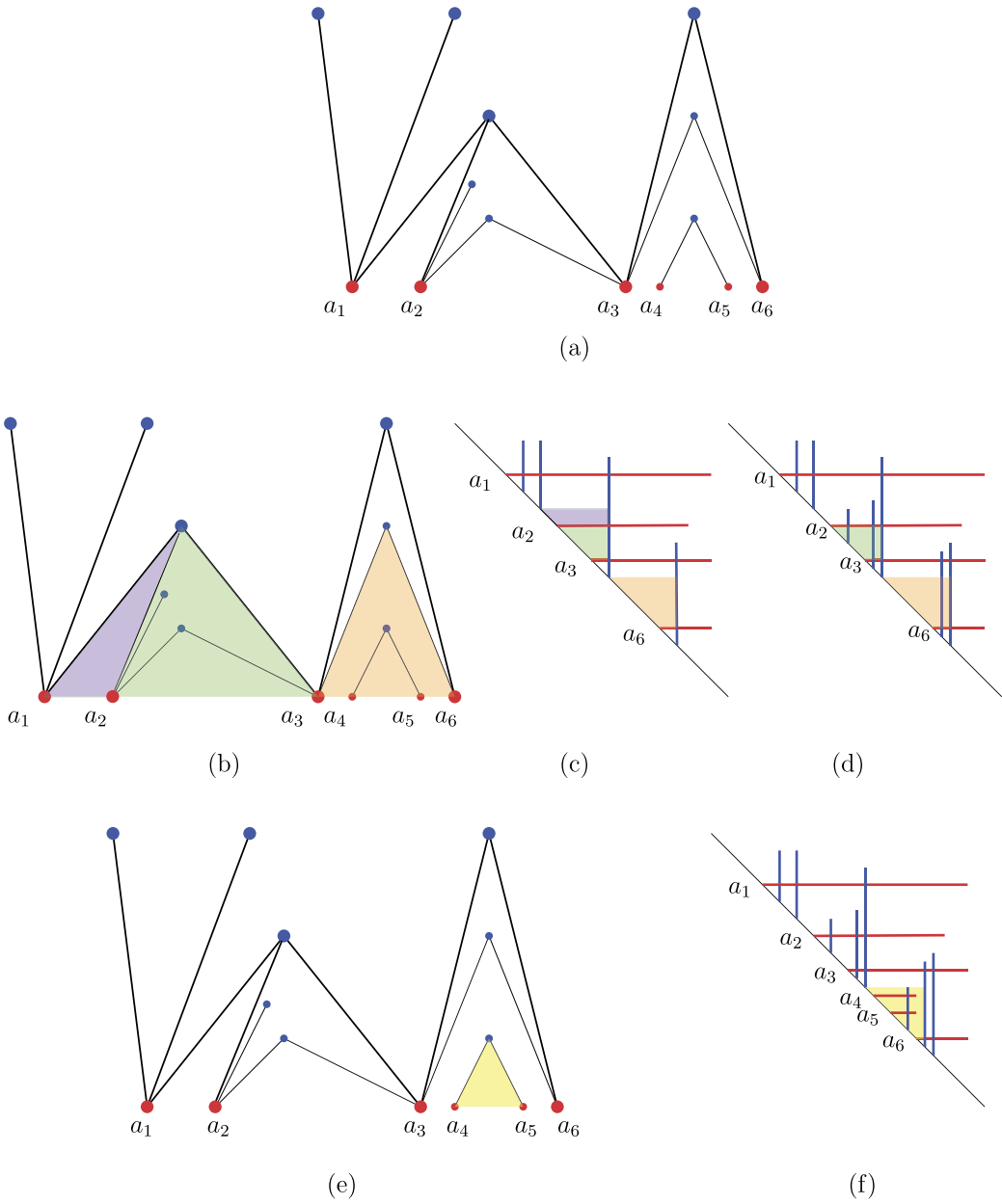


Fig. 7. Illustration for Remark 4.

$\Phi_2$ : (Condition C2.) We must express the condition that if the ordered submatrix  $\begin{matrix} & b_p & b_q \\ a_i & \begin{bmatrix} 1 & * \\ 0 & 1 \end{bmatrix} \end{matrix}$  exists, then  $p_{b_p < a_j} = 1$ .  
 Thus, if  $m_{i,p} = 1, m_{j,q} = 1$  and  $m_{j,p} = 0$ , then we add the clause  $(\neg p_{a_i < a_j} \vee \neg p_{b_p < b_q} \vee \neg p_{a_j < b_p})$ .  
 $\Phi_3$ : For each triple  $u, v, w$  of vertices, add the clause  $(\neg p_{u < v} \vee \neg p_{v < w} \vee p_{u < w})$ . Intuitively, these are transitivity constraints, which would ensure a total ordering on the ground line.

It is not difficult to show that the 3-SAT  $\Phi$  is satisfiable if and only if  $G$  admits the required intersection representation. However, since  $\Phi$  contains  $O(n^2)$  variables, using known SAT-solvers would not be faster than a naive algorithm that simply guesses the order of the segments along the ground line. Therefore, an interesting direction for future research would be to find a 3-SAT formulation with a linear number of variables.

**Open Problem 2.** Can we improve the time complexity of the recognition algorithm for graphs with fixed  $As$ ?

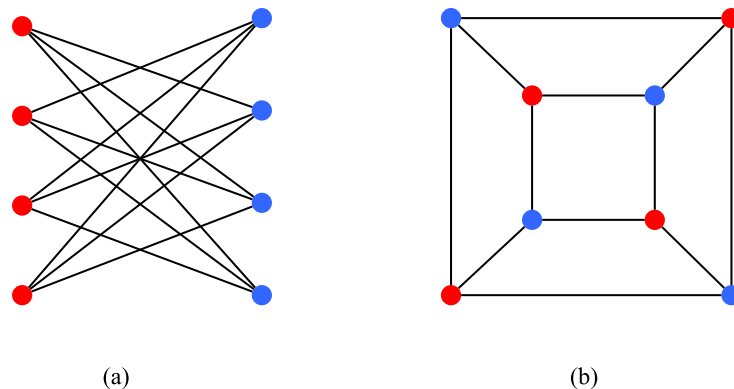


Fig. 8. Graph that does not admit a Stick representation. Its matrix representation can have all zeros placed on a diagonal.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

The research of A. Lubiw and D. Mondal is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). Also, this work is supported in part by NSF grants CCF-1740858, CCF-1712119 and DMS-1839274, DMS-1839307.

### References

- [1] B. Aspvall, M.F. Plass, R.E. Tarjan, A linear-time algorithm for testing the truth of certain quantified Boolean formulas, *Inf. Process. Lett.* 8 (3) (1979) 121–123.
- [2] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, *J. Comput. Syst. Sci.* 13 (3) (1976) 335–379.
- [3] S. Cabello, M. Jejcic, Refining the hierarchies of classes of geometric intersection graphs, *Electron. J. Comb.* 24 (1) (2017) 33.
- [4] J. Cardinal, S. Felsner, T. Miltzow, C. Tompkins, B. Vogtenhuber, Intersection graphs of rays and grounded segments, in: H.L. Bodlaender, G.J. Woeginger (Eds.), *Proceedings of the 43rd International Workshop on Graph-Theoretic Concepts in Computer Science*, in: LNCS, vol. 10520, Springer, 2017, pp. 153–166.
- [5] D. Catanzaro, S. Chaplick, S. Felsner, B.V. Halldórsson, M.M. Halldórsson, T. Hixon, J. Stacho, Max point-tolerance graphs, *Discrete Appl. Math.* 216 (2017) 84–97.
- [6] J. Chalopin, D. Gonçalves, Every planar graph is the intersection graph of segments in the plane, in: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, ACM, 2009, pp. 631–638.
- [7] T.M. Chan, E. Grant, Exact algorithms and APX-hardness results for geometric packing and covering problems, *Comput. Geom.* 47 (2, Part A) (2014) 112–124.
- [8] S. Chaplick, E. Cohen, G. Morgenstern, Stabbing polygonal chains with rays is hard to approximate, in: *Canadian Conference on Computational Geometry (CCCG)*, 2013.
- [9] S. Chaplick, S. Felsner, U. Hoffmann, V. Wiechert, Grid intersection graphs and order dimension, *Order* 35 (2) (2018) 363–391.
- [10] S. Chaplick, P. Hell, Y. Otachi, T. Saitoh, R. Uehara, Intersection dimension of bipartite graphs, in: T.V. Gopal, M. Agrawal, A. Li, S.B. Cooper (Eds.), *Theory and Applications of Models of Computation*, in: LNCS, vol. 8402, Springer, 2014, pp. 323–340.
- [11] S. Chaplick, P. Kindermann, A. Löffler, F. Thiele, A. Wolff, A. Zaft, J. Zink, Stick graphs with length constraints, arXiv preprint, arXiv:1907.05257, 2019.
- [12] L. Chen, Y. Yesha, Efficient parallel algorithms for bipartite permutation graphs, *Networks* 23 (1) (1993) 29–39.
- [13] O. Cogis, On the Ferrers dimension of a digraph, *Discrete Math.* 38 (1) (1982) 47–52.
- [14] D.G. Corneil, S. Olariu, L. Stewart, The LBFS structure and recognition of interval graphs, *SIAM J. Discrete Math.* 23 (4) (2009) 1905–1953.
- [15] S. Das, M. Sen, A.B. Roy, D.B. West, Interval digraphs: an analogue of interval graphs, *J. Graph Theory* 13 (2) (1989) 189–202.
- [16] F. De Luca, M.I. Hossain, S. Kobourov, A. Lubiw, D. Mondal, Recognition and drawing of stick graphs, in: T. Biedl, A. Kerren (Eds.), *Graph Drawing and Network Visualization*, Springer International Publishing, Cham, 2018, pp. 303–316.
- [17] S. Felsner, G.B. Mertzios, I. Mustafa, On the recognition of four-directional orthogonal ray graphs, in: K. Chatterjee, J. Sgall (Eds.), *Mathematical Foundations of Computer Science 2013*, Springer, Berlin, Heidelberg, 2013, pp. 373–384, Berlin Heidelberg.
- [18] U. Fößmeier, M. Kaufmann, Nice drawings for planar bipartite graphs, in: G.C. Bongiovanni, D.P. Bovet, G.D. Battista (Eds.), *Proceedings of the 3rd Italian Conference on Algorithms and Complexity*, in: LNCS, vol. 1203, Springer, 1997, pp. 122–134.
- [19] B.V. Halldórsson, D. Aguiar, R. Tarpine, S. Istrail, The Clark phase-able sample size problem: long-range phasing and loss of heterozygosity in gwas, in: B. Berger (Ed.), *Research in Computational Molecular Biology*, Springer, Berlin Heidelberg, 2010, pp. 158–173.
- [20] I.B.-A. Hartman, I. Newman, R. Ziv, On grid intersection graphs, *Discrete Math.* 87 (1) (1991) 41–52.
- [21] P. Hell, B. Mohar, A. Rafiey, Ordering without forbidden patterns, in: A.S. Schulz, D. Wagner (Eds.), *Algorithms - ESA 2014*, Springer, Berlin Heidelberg, 2014, pp. 554–565.
- [22] T. Hixon, *Hook Graphs and More: Some Contributions to Geometric Graph Theory*, Master's thesis, Technische Universität Berlin, 2013.
- [23] U. Hoffmann, *Intersection Graphs and Geometric Objects in the Plane*, Master's thesis, Technische Universität Berlin, 2016.
- [24] M.J. Katz, J.S. Mitchell, Y. Nir, Orthogonal segment stabbing, *Comput. Geom.* 30 (2) (2005) 197–205.

- [25] B. Klinz, R. Rudolf, G.J. Woeginger, Permuting matrices to avoid forbidden submatrices, *Discrete Appl. Math.* 1 (60) (1995) 223–248.
- [26] J. Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, *Discrete Appl. Math.* 52 (3) (1994) 233–252.
- [27] A. Lubiw, Doubly lexical orderings of matrices, *SIAM J. Comput.* 16 (5) (1987) 854–879.
- [28] I. Mustață, M. Pergel, Unit grid intersection graphs: recognition and properties, arXiv preprint, arXiv:1306.1855, 2013.
- [29] H. Müller, Recognizing interval digraphs and interval bigraphs in polynomial time, *Discrete Appl. Math.* 78 (1) (1997) 189–205.
- [30] M. Oswald, G. Reinelt, The simultaneous consecutive ones problem, *Theor. Comput. Sci.* 410 (21–23) (2009) 1986–1992.
- [31] A.M.S. Shrestha, S. Tayu, S. Ueno, On orthogonal ray graphs, *Discrete Appl. Math.* 158 (15) (2010) 1650–1659.
- [32] J.A. Soto, C. Telha, Jump number of two-directional orthogonal ray graphs, in: O. Günlük, G.J. Woeginger (Eds.), *Integer Programming and Combinatorial Optimization*, Springer, Berlin Heidelberg, 2011, pp. 389–403.
- [33] M. Soto, C.T. Caro, P-box: a new graph model, *Discret. Math. Theor. Comput. Sci.* 17 (1) (2015) 169.
- [34] J.P. Spinrad, Doubly lexical ordering of dense 0-1 matrices, *Inf. Process. Lett.* 45 (5) (1993) 229–235.
- [35] J.P. Spinrad, A. Brandstädt, L. Stewart, Bipartite permutation graphs, *Discrete Appl. Math.* 18 (3) (1987) 279–292.