Edge-Intersection Graphs of Boundary-Generated Paths in a Grid

Martin Charles Golumbic[1], Gila Morgenstern[2], and Deepak Rajendraprasad[3]

[1] Caesarea Rothschild Institute, University of Haifa, Mount Carmel, Haifa 31905, Israel
[2] HIT - Holon Institute of Technology, Israel
[3] Department of Computer Science and Engineering, Indian Institute of Technology Palakkad, India.

October 3, 2017

Abstract

Edge-intersection graphs of paths on a grid (or EPG graphs) are graphs whose vertices can be represented as simple paths on a rectangular grid such that two vertices are adjacent in the graph if and only if the corresponding paths share at least one edge of the grid. For two boundary points p and q on two adjacent boundaries of a rectangular grid $\mathcal{G}$, we call the unique single-bend path connecting p and q in $\mathcal{G}$ using no other boundary point of $\mathcal{G}$ as the path *generated* by (p, q). A path in $\mathcal{G}$ is called *boundary-generated*, if it is generated by some pair of points on two adjacent boundaries of $\mathcal{G}$. In this article, we study the edge-intersection graphs of boundary-generated paths on a grid or ∂EPG graphs. The motivation for studying these graphs comes from problems in the context of circuit layout.

We show that ∂EPG graphs can be covered by two collections of vertex-disjoint co-bipartite chain graphs. This leads us to a linear-time testable characterization of ∂EPG trees and also an almost tight upper bound on the equivalence covering number of general ∂EPG graphs. We also study the cases of two-sided ∂EPG and three-sided ∂EPG graphs, which are respectively, the subclasses of ∂EPG graphs obtained when all the boundary-vertex pairs which generate the paths are restricted to lie on at most two or three boundaries of the grid. For the former case, we give a complete characterization.

1 Introduction

Edge intersection graphs of paths on a grid (or for short EPG graphs) were first introduced by Golumbic, Lipshteyn and Stern in [13]. This is the class of graphs whose vertices can be represented as simple paths on a rectangular grid so that two vertices are adjacent if and only if the corresponding paths share at least one edge of the grid.

EPG graphs have a practical use, e.g., in the context of circuit layout setting, which may be modeled as paths (wires) on a grid. In the knock-knee layout model, two wires may either cross or bend (turn) at a common grid point, but are not allowed to share a grid-edge; that is, overlap of wires is not allowed.

1.1 B_k-EPG graphs

In [13], the authors show that every graph is an EPG graph. That is, for every graph $G = (V, E)$ there exists an EPG representation $\langle \mathcal{G}, \mathcal{P} \rangle$ where $\mathcal{P} = \{P_v : v \in V\}$ is a collection of paths on a grid $\mathcal{G}$, corresponding to the vertices of V and satisfying: paths $P_v, P_u \in \mathcal{P}$ share a grid-edge of $\mathcal{G}$ if and only if $(v, u) \in E$. Moreover, they show that if G has n vertices and m edges, then there exists an EPG representation $\langle \mathcal{G}, \mathcal{P} \rangle$ of G in which $\mathcal{G}$ is a grid of size $n \times (n+m)$ and the paths in $\mathcal{P}$ are monotonic. As such, much of the current research today focuses on subclasses of EPG graphs, and, in particular, limiting the type of paths allowed.

A turn of a path at a grid point is called a *bend* and a graph is called a *k-bend EPG graph* (denoted B_k-EPG) if it has an EPG representation in which each path has at most k bends. It is both interesting mathematically, and justified by the circuit layout application described above, to consider subclasses of graphs, e.g., by bounding the number of bends allowed in each path.

In [4], the authors show that for any k, only a small fraction of all labeled graphs on n vertices are B_k-EPG, and that for any fixed degree $d \geq 4$, a grid size of $\Theta(n^2)$ is needed to give an EPG representation of every graph with n vertices and maximum degree d, for sufficiently large n. For example, a representation of the complete bipartite graph $K_{n/2,n/2}$ needs at least $n^2/4$ grid-edges, and [13] showed that $3n^2$ grid-edges is sufficient to represent any graph.

The class of B_0-EPG graphs is easily seen to be equivalent to the well known family of interval graphs (see [11]). B_1-EPG graphs are the single bend EPG graphs, studied further in [3, 6, 8, 13, 14]. Improving a result of [5], it was shown in [17] that every planar graph is a B_4-EPG graph. It is still open whether $k = 4$ is best possible. So far it is only known that there are planar graphs that are B_3-EPG graphs and not B_2-EPG graphs. Some subclasses of planar graphs have showen to be B_2-EPG graphs, e.g., Halin graphs [10] and outerplanar graphs [17] (thus proving a conjecture of [5]). Also, [1] have shown that circular-arc graphs are B_3-EPG graphs, and that this is best possible.

For the case of B_1-EPG graphs, Golumbic, Lipshteyn and Stern [13] showed that every tree is a B_1-EPG graph, and in [14] they showed that single bend paths on a grid have strong Helly number 4. Asinowski and Ries [3] proved that every B_1-EPG graph on n vertices contains either a clique or a stable set of size at least $n^{1/3}$. In [3], the authors also give a characterization of the B_1-EPG graphs among some subclasses of chordal graphs, namely, chordal bull-free graphs, chordal claw-free graphs, chordal diamond-free graphs, and special cases of split graphs. In [8], a characterization of the sub-family of cographs that are B_1-EPG graphs is given by a complete family of minimal forbidden induced subgraphs.

No characterization is known for B_k-EPG graphs (for any $k \geq 1$) and the recognition problems are NP-complete for $k = 1$ [16] and $k = 2$ [22]. For $k = 1$, the recognition problem remains NP-complete even if just one of the four single bend shapes is allowed, the so called L-*shaped* B_1-EPG graphs [6].

1.2 Boundary generated EPG graphs

In this paper, we consider a further restriction on B_1-EPG graphs, namely that, the endpoints of every path lie on the boundary of the host rectangular grid; see Figure 1 for an illustration. This restriction is motivated by applications in circuit design, where it is easier to take out connections from the edge of the chip or board. This notion was first proposed for investigation in [12]. Formally,

Definition 1.1. For two boundary points p and q on two adjacent boundaries of a rectangular grid $\mathcal{G}$, we call the unique single-bend path connecting p and q in $\mathcal{G}$ using no other boundary point of $\mathcal{G}$ the path *generated* by (p, q). A path in $\mathcal{G}$ is called *boundary-generated*, if it is generated by two points on adjacent boundaries of $\mathcal{G}$. A graph G is called an *edge-intersection graph of boundary-generated paths in a grid*, ∂EPG graphs for short, if there exists a rectangular grid $\mathcal{G}$ and a representation ψ which assigns to every vertex in G, a boundary-generated path in $\mathcal{G}$ such that two vertices $u, v \in V(G)$ are adjacent in G, if and only if the corresponding paths $\psi(u)$ and $\psi(v)$ share a common grid-edge of $\mathcal{G}$. In this case, we call $\langle \mathcal{G}, \mathcal{P} \rangle$ a ∂EPG representation of G, where $\mathcal{P}$ is the multiset $\{\psi(v) \ : \ v \in V(G)\}$.

Figure 1: A 4-side boundary generated representation for $K_{2,4}$.

2 Preliminaries

All graphs considered are finite and undirected. The complement of a graph G is denoted by $\overline{G}$. Two adjacent (non-adjacent) vertices with the same neighborhood are called *true twins* (*false twins*). The *reduced graph* of a graph G is the graph obtained from G by deleting all but one vertex from each set of false twins. The *line graph* $L(G)$ of a graph G is the intersection graph of the edge-set of G.

An *equivalence graph* is a vertex disjoint union of cliques, or equivalently, the graph where the adjacency relation is an equivalence relation. The *equivalence covering number* $\mathrm{eq}(G)$ of a graph G is the minimum number of equivalence graphs whose union is G [2]. For triangle-free graphs, equivalence covering number is the same as edge-chromatic number.

The product dimension or Prague dimension of a graph is a parameter which is closely related to the equivalence covering number. A *product k-encoding* of a graph G is obtained by associating to each vertex v a unique vector $f(v) = (v_1, \ldots, v_k)$ over the natural numbers so that for $xy \in E(G)$ the vectors $f(x)$ and $f(y)$ differ in all coordinates and for $xy \notin E(G)$ the vectors $f(x)$ and $f(y)$ agree in at least one coordinate. The *product dimension* or *Prague dimension* of a graph G, $\mathrm{pdim}(G)$, is the smallest number k such that G has a product k-encoding. It is an easy observation (cf. [21]) that

$$\mathrm{eq}(G) \leq \mathrm{pdim}(\overline{G}) \leq \mathrm{eq}(G) + 1.$$

The difference of 1 occurs because a product k-encoding needs to associate a *unique* vector to each vertex. For instance, the product dimension of the empty graph G on two vertices is 2 whereas $\overline{G}$ can be covered by one clique. But if G has no true twins (i.e., $\overline{G}$ has no false twins), then $\text{eq}(G) = \text{pdim}(\overline{G})$.

The equivalence covering number was first studied by Duchet in 1979 [9]. The concept of product dimension of a graph was first used by Nešetřil and Rödl to prove the Galvin-Ramsey property of the class of all finite graphs [21]. Lovász, Nešetřil and Pultr [19] and Alon [2] describe a structure in a graph that forces its product dimension (equivalence covering number) to be high. Both the proofs employ exterior algebra techniques. We use the same structure to show the existence of ∂EPG graphs with arbitrarily large equivalence covering number. It is established in [16] that the bend number of a graph G, which is the smallest k for which G is B_k-EPG, is at most $\text{eq}(G) - 1$. (There, equivalence number of a graph is referred to as its global clique covering number).

A bipartite graph is called a *chain graph* if, for each color class, the neighborhoods of the vertices in that color class can be linearly ordered with respect to inclusion. Equivalently, it is a bipartite graph which is $2K_2$-free. A *co-bipartite chain graph* is the complement of a bipartite chain graph. Note that the linear orderings of neighborhoods of vertices of each part is still preserved (there is a reversing).

Remark. It is easy to verify that a co-bipartite chain graph can be represented as the intersection graph of open intervals of the form $(-\infty, x)$ or $(x, +\infty)$ where $x \in \mathbb{R}$.

We use the shorthand $[n]$ for the set $\{1, \ldots, n\}$ and lg to denote logarithm to the base 2.

3 Equivalence cover for ∂EPG graphs

Let us befriend this new class of graphs by taking a close look at the subclass of two-sided ∂EPG graphs. In a two-sided ∂EPG graph, the set of all the boundary points which generate the paths are restricted to lie on two adjacent boundaries of the grid. Without loss of generality, we may assume that all the paths are restricted to start from the top boundary and bend towards the right boundary. The following characterization reaffirms the feeling that we have met two-sided ∂EPG graphs in many guises before.

Theorem 3.1. *The following conditions are equivalent for a graph G:*

(i) *G is a two-sided ∂EPG graph.*

(ii) *G has equivalence covering number at most* 2.

(iii) *The reduced graph of $\overline{G}$ has product dimension at most* 2.

(iv) *G is the line graph of a bipartite multigraph.*

(v) *G contains no claw, no gem, no 4-wheel, and no odd hole.*

(vi) *The clique graph of G, i.e., the intersection graph of maximal cliques in G, is bipartite.*

Proof. We show that (i) $\implies$ (ii) $\implies$ (iii) $\implies$ (iv) $\implies$ (i). The equivalence (iv) $\iff$ (v) is Theorem 4 in [20]. It is easy to verify that (iv) $\implies$ (vi) $\implies$ (ii).

(i) $\implies$ (ii): Let G be a two-sided ∂EPG graph. If two boundary-generated paths share a horizontal edge, then they have the same generating end point on the right boundary. Hence they share the horizontal edge incident on that right boundary-point. Similarly, if two boundary-generated paths share a vertical edge, then they share the vertical edge incident on their top boundary-point. Now if we consider two subgraphs G_h and G_v of G where two vertices are adjacent in G_h (resp., G_v) if and only if the two corresponding boundary-generated paths share a horizontal edge (resp., vertical edge). It is easy to verify that $G = G_h \cup G_v$, and the above discussion shows that both G_h and G_v are equivalences.

(ii) $\implies$ (iii): Suppose that a graph G can be covered by two spanning equivalence graphs G_h and G_v. Let G_h (resp., G_v) be the disjoint union of cliques $H_1, \ldots, H_k$ (resp., $V_1, \ldots, V_l$). To every vertex x in G, we assign an ordered pair $\phi(x) = (i, j)$, where V_i is the (unique) clique in G_v which contains x and H_j is the (unique) clique in G_h which contains x. If G has no true-twins, then ϕ is an injective labeling. Two vertices x and y are adjacent in G if and only if $\phi(x)$ and $\phi(y)$ agree in at least one coordinate. Hence ϕ is a product 2-encoding of $\overline{G}$ and thus $\mathrm{pdim}(\overline{G}) \leq 2$.

(iii) $\implies$ (iv): Let H be a reduced graph with a product 2-encoding $\phi : V(H) \to [k] \times [l]$. Consider the bipartite graph X on parts $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_l\}$ with $a_i \in A$ adjacent to $b_j \in B$ if and only if there exists a vertex $v \in V(H)$ with $\phi(v) = (i, j)$. Two edges of X are adjacent only if the corresponding ordered pairs agree in at least one coordinate. Hence the line graph of X is isomorphic to $\overline{H}$. If H had false twins, we can do the same construction of X for the reduced graph of H and then represent the false twins of H with multi-edges in X.

(iv) $\implies$ (i): Let X be a bipartite multi-graph with parts $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$. Consider the $(k+1) \times (l+1)$ grid $\mathcal{G}$. Assign to every edge $a_i b_j$ in X, the boundary-generated single-bend path starting from the i-th boundary-point on the top boundary of $\mathcal{G}$ and ending at the j-th boundary-point on the right boundary of $\mathcal{G}$. It is easy to verify that we have a ∂EPG representation of $L(X)$, the line graph of X. □

Notice that two-sided ∂EPG graphs is a subclass of the L-*shaped* B_1-EPG graphs. The recognition problem is NP-complete for the latter class [6], while it is easily seen, for instance by characterization (iv) of Theorem 3.1 above, to be poly-time for our new friend, the class of two-sided ∂EPG graphs. Once we discovered that two-sided ∂EPG graphs have equivalence covering number at most 2, it was natural to investigate the equivalence covering number of three-sided and general (i.e., four-sided) ∂EPG graphs. In Section 4, we establish that, unlike the case with two-sided ∂EPG graphs, the equivalence covering number of three-sided and four-sided ∂EPG graphs can be unbounded. In particular, we prove that the maximum possible equivalence covering number of an n-vertex ∂EPG graph is $\Theta(\lg n)$. Our main tool to study ∂EPG graphs is the following covering theorem, where co-bipartite chain graphs steal the limelight from cliques.

Theorem 3.2. *If G is a ∂EPG graph, then G can be covered by two graphs G_h and G_v, where both G_h and G_v are vertex-disjoint unions of co-bipartite chain graphs. Moreover, if G is three-sided ∂EPG, then G_v can be further restricted to be an equivalence graph.*

Proof. Let $\langle \mathcal{P}, \mathcal{G} \rangle$ be a ∂EPG representation of G. Consider two subgraphs G_h and G_v of G on $V(G)$ defined as follows:

(i) $E(G_h)$ consists of all the pairs of paths in $\mathcal{P}$ which share some horizontal edge of the grid $\mathcal{G}$.

(ii) $E(G_v)$ consists of all the pairs of paths in $\mathcal{P}$ which share some vertical edge of the grid $\mathcal{G}$.

It is easy to see that the two graphs are subgraphs of G such that $G = G_h \cup G_v$. It can also be checked that the paths whose horizontal parts are at the same height in the grid form a co-bipartite chain subgraph and that G_h is a vertex disjoint union of such co-bipartite chain graphs, one co-bipartite chain graph for each horizontal level of the grid used. Similarly G_v is also a disjoint union of co-bipartite chain graphs.

If $\mathcal{P}$ does not contain any paths which use the top boundary of $\mathcal{G}$, then it is clear that all the paths which share a vertical edge share a vertical edge incident on the bottom boundary and hence form a clique. Hence G_v will be a disjoint union of cliques. □

We explore two consequences of Theorem 3.2. In Section 4, we use it to give a good upper bound on the equivalence covering number of ∂EPG graphs. In Section 5, we use it as the first step towards giving a complete characterization of ∂EPG trees.

4 Bounding the equivalence covering number of ∂EPG graphs

We start by studying the equivalence covering number of co-bipartite chain graphs, or equivalently, the product dimension of bipartite chain graphs.

Consider the following two encoding schemes for numbers in $\mathbb{Z}_n = \{0, \ldots, n-1\}$. Let $k = \lceil \lg n \rceil + 1$. Then the encodings $f : \mathbb{Z}_n \to \mathbb{Z}_n^k$ and $g : \mathbb{Z}_n \to [n]^k$ are defined as follows.

$$
\begin{aligned}
f(i) &= \left(\lfloor \tfrac{i}{2^0} \rfloor, \lfloor \tfrac{i}{2^1} \rfloor, \ldots, \lfloor \tfrac{i}{2^{k-1}} \rfloor\right), \text{ and} \\
g(i) &= f(i) + (1, \ldots, 1).
\end{aligned}
$$

For example, if $n = 8$, the f and g are as below.

i	$f(i)$	$g(i)$
0	$(0, 0, 0, 0)$	$(1, 1, 1, 1)$
1	$(1, 0, 0, 0)$	$(2, 1, 1, 1)$
2	$(2, 1, 0, 0)$	$(3, 2, 1, 1)$
3	$(3, 1, 0, 0)$	$(4, 2, 1, 1)$
4	$(4, 2, 1, 0)$	$(5, 3, 2, 1)$
5	$(5, 2, 1, 0)$	$(6, 3, 2, 1)$
6	$(6, 3, 1, 0)$	$(7, 4, 2, 1)$
7	$(7, 3, 1, 0)$	$(8, 4, 2, 1)$

The important property of this encoding is that $f(i)$ and $g(i')$ agree in at least one position if and only if $i > i'$. Indeed, if $i \leq i'$ every coordinate of $g(i')$ is strictly greater than the corresponding coordinate of $f(i)$; and if $i > i'$, one can verify that $\lfloor i/2^l \rfloor = \lfloor i'/2^l \rfloor + 1$ when $l = \lfloor \lg(i - i') \rfloor$. Also notice that $f(i)$ and $f(i')$ agree on the last coordinate for every i and i'. Similarly, $g(i)$ and $g(i')$ agree on the last coordinate for every i and i'. We use this encoding to prove an upper bound on the product dimension of bipartite chain graphs. We will use the following Proposition from [19] to show that this upper bound is tight up to additive constants.

Proposition 4.1. *([19]; Proposition 5.3) Let $x_1, \ldots, x_k$ be distinct vertices of a graph G such that for some $y_1, \ldots, y_k \in V(G)$ we have $x_i y_i \in E(G), \forall i \in [k]$ and $x_i y_j \notin E(G), \forall i < j$. Then $\text{pdim}(G) \geq \lg k$.*

Theorem 4.2. *If H is a bipartite chain graph on n vertices, then the product dimension of H is at most $\lceil \lg n \rceil + 1$. Moreover, there exists a bipartite chain graph on n vertices with product dimension at least $\lceil \lg n \rceil - 1$.*

Proof. Let G be a bipartite chain graph on n vertices. Recall that the co-bipartite chain graph $\overline{G}$ can be represented as the intersection graph of open intervals on the real line which are infinite in exactly one direction. Without loss of generality, let us assume that the finite end-point of each interval is a unique integer between 0 and $n - 1$. Now we describe an encoding for each interval using k coordinates ($k = \lceil \lg n \rceil + 1$) such that the encodings of two intervals agree on at least one coordinate if and only if the intervals intersect. This gives a product encoding of G and hence will prove the upper bound.

Let $\mathcal{L}$ denote the collection of intervals in the representation of $\overline{G}$ which are infinite to the left and $\mathcal{R}$ denote the collection of intervals which are infinite to the right. For an interval $I \in \mathcal{L}$ with right end-point i where $i \in \mathbb{Z}_n$, the encoding for I is $f(i)$. For an interval $I' \in \mathcal{R}$ with left end-point i' where $i' \in \mathbb{Z}_n$, the encoding for I' is $g(i')$. It can be checked that the encodings for two intervals in $\mathcal{L}$ agree on the last coordinate. So do two intervals in $\mathcal{R}$. For two intervals $I = (-\infty, i) \in \mathcal{L}$ and $I' = (i', +\infty) \in \mathcal{R}$, their encodings agree in at least one coordinate if and only if $i > i'$, i.e., if and only if $I \cap I' \neq \emptyset$.

Now we show that this upper bound is nearly tight. Let G be a bipartite graph on $n = 2k$ vertices with parts $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_k\}$ such that $x_i y_i \in E(G)$ if and only if $i \geq j$. By Proposition 4.1, $\text{pdim}(G) \geq \lg k = \lg n - 1$. It is easy to check that G does not contain an induced $2K_2$ and hence G is a bipartite chain graph. □

Hence, a co-bipartite chain graph on n vertices can be covered by at most $\lceil \lg n \rceil + 1$ equivalence graphs.. It is also useful to note the following easy observation.

Observation 4.3. *If G is the union of two graphs G_1 and G_2 on the same vertex set, then $\mathrm{eq}(G) \leq \mathrm{eq}(G_1) + \mathrm{eq}(G_2)$. Further, if G is a disjoint union of two non-trivial graphs G_1 and G_2 (on two different vertex sets), then the $\mathrm{eq}(G) = \max\{\mathrm{eq}(G_1), \mathrm{eq}(G_2)\}$.*

The main result of this section now follows from Theorem 3.2, the above observation, and Theorem 4.2. Also note that every co-bipartite chain graph can be represented as a ∂EPG graph using boundary generated paths, all of which lie in the same row of the grid. Thus, they are three-sided ∂EPG.

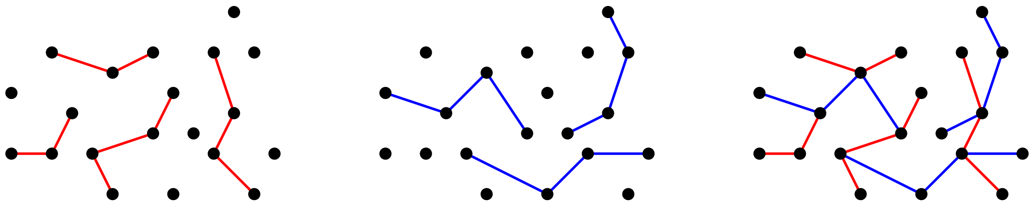
Theorem 4.4. *If G is a ∂EPG graph on n vertices, then $\mathrm{eq}(G) \leq 2\lceil \lg n \rceil + 2$. Further, if G has a three-sided ∂EPG representation then $\mathrm{eq}(G) \leq \lceil \lg n \rceil + 2$. Moreover, there exist n-vertex three-sided ∂EPG graphs G_n with equivalence covering number at least $\lceil \lg n \rceil - 2$.*

5 Characterizing and recognizing ∂EPG trees

We know that every tree is a B_1-EPG graph [13]. The restriction to use only boundary-generated paths disqualifies a large portion of trees. One can verify using the geometry of a grid that the maximum degree of a ∂EPG tree is at most 4. In this section, we characterize trees which have a ∂EPG representation.

In the following, we refer to edges and paths in a tree just as "edges" and "paths", while we refer to edges of a grid and to paths in an EPG representation as "grid-edges" and "grid-paths".

A *linear forest* is a forest in which every connected component is a path. A *linear k-forest* is a linear forest in which every path has at most k edges (i.e., at most P_{k+1}, a path on $k+1$ vertices). The *linear k-arboricity* $\mathrm{la}_k(G)$ of a graph G is the minimum number of linear k-forests whose union is G; see Figure 2 for an illustration. Notice that the linear 1-arboricity of a graph is its chromatic index. This parameter was introduced by Habib and Peroche in 1982 [15].



(a) (b) (c)

Figure 2: Figures (a) and (b) illustrates two edge-disjoint 3-linear forests over the same vertex-set, whose union in (c) is a tree.

Note that a triangle-free clique is a single edge, a triangle-free co-bipartite chain graph is a subgraph of P_4, a path on 4 vertices, and that a triangle-free graph has the same equivalence covering number and chromatic index. Hence, the following is an immediate corollary of Theorems 3.2 and 3.1 for triangle-free ∂EPG graphs.

Corollary 5.1. *If G is a triangle-free ∂EPG graph, then G can be covered by two linear 3-forests. Moreover, if G has a three-sided ∂EPG representation, then one of the forests can be restricted to be a matching of edges (1-forest). Finally, if G has a two-sided ∂EPG representation, then G is a disjoint union of two matchings, that is, G is a disjoint union of paths and even cycles.*

The next result shows that the converse of Corollary 5.1 is true for trees.

Theorem 5.2. *A tree T is a ∂EPG graph if and only if T can be covered by two linear 3-forests. Moreover, T has a three-sided (resp. two-sided) ∂EPG representation if and only if we can restrict one (resp. both) of the above forests to be a disjoint collection of edges.*

One direction of the above theorem follows from Corollary 5.1. For the other direction, we need to introduce a new partial-order extension question.

Let $\mathcal{A} = \{A_1, \ldots, A_k\}$ and $\mathcal{B} = \{B_1, \ldots, B_{k'}\}$ be two edge-disjoint linear 3-forests whose union is a tree T. We will realize each A_i as the grid-edge-intersection graph of a collection of grid-paths whose vertical part is at the same column c_i in the grid. Similarly, we will realize each B_j as the grid-edge-intersection graph of a collection of grid-paths whose horizontal part is at the same row r_j in the grid. We have two kinds of choices and one constraint. Firstly, a P_4 has essentially two ways of being realized as the edge-intersection graph of a collection of horizontal (vertical) grid-paths with at least one end point on a vertical (horizontal) boundary of the grid. One representation is the left-to-right (top-to-bottom) mirrored version of the other. Secondly, we have the freedom to choose the relative order among c_i's and the relative order among r_j's. The constraint comes from the fact that $\mathcal{A}$ and $\mathcal{B}$ are graphs over the same vertex set. Hence, a representation we choose for each A_i imposes a partial order on some r_j's, namely on $\{r_j : V(A_i) \cap V(B_j) \neq \emptyset\}$. There is a similar restriction on c_i's arising from the representation of the B_j's. We formalize this constraint using the language of partial orders, and then show that the second freedom suffices to satisfy this constraint.

Definition 5.3. Let V be a set and let $\mathcal{S}$ be a partition of V. Further let $\prec_V$ and $\prec_\mathcal{S}$ be partial orders, respectively, on V and $\mathcal{S}$. We say that $\prec_\mathcal{S}$ is $\prec_V$*-preserving* if, for every pair of vertices $u, v \in V$ with $u \prec_V v$ we have $S_u \prec_\mathcal{S} S_v$ where S_u and S_v are, respectively, the (unique) members in $\mathcal{S}$ containing u and v.

Notice that, in general, preserving orders do not necessarily exist; for example, the partition $\{\{1,4\}, \{2,3\}\}$ of $\{1, \ldots, 4\}$ has no $<$-preserving order. Given a partition $\mathcal{S}$ of the ground-set V of a partial order $\prec_V$, let D be the directed multigraph (possibly with self-loops) on the vertex-set $\mathcal{S}$ which contains one directed edge from S_u to S_v for each comparable pair $u \prec_V v$ in V, where S_u and S_v are, respectively, the sets in $\mathcal{S}$ containing u and v. One can verify that there exists a $\prec_V$-preserving total order $\prec_\mathcal{S}$ on $\mathcal{S}$ if and only if D does not contain a directed cycle. In fact, if D is acyclic, then any topological sorting of D is a $\prec_V$- preserving total order. Next, we show that this is indeed the case that we will face.

Lemma 5.4. *Let T be a tree and let $\mathcal{A}$ and $\mathcal{B}$ be two edge-disjoint sub-forests of T. Further let $\prec_b$ be a partial order on $V(T)$, in which any two comparable vertices lie in the same connected component of $\mathcal{B}$. Then there exists a $\prec_b$-preserving total order $\prec_{\mathcal{A}}$ among the connected components of $\mathcal{A}$.*

Proof. Let $T_{\mathcal{A}}$ be the tree obtained from T by contracting all the edges in the forest $\mathcal{A}$. Since contracting an edge of a tree produces another tree (without any loops or multiple edges), $T_{\mathcal{A}}$ is a tree. Each vertex of $T_{\mathcal{A}}$ corresponds to a connected component in $\mathcal{A}$ and is labeled by it. Our goal is to define a $\prec_b$-preserving total order on the vertices of $T_{\mathcal{A}}$ and thereby among the connected components of $\mathcal{A}$.

Let $B \in \mathcal{B}$ be a subtree of T with vertices $v_1, \ldots, v_t$. For $1 \leq i \leq t$, the vertex v_i belongs to a distinct tree $A_i \in \mathcal{A}$, which corresponds to a distinct vertex $A_i \in T_{\mathcal{A}}$ (as otherwise T contains a cycle). Since B is connected, the vertices $A_1, \ldots, A_t$ form a connected subgraph (i.e, a subtree) of $T_{\mathcal{A}}$, which we denote by $T_{\mathcal{A}}[B]$. Removing the edges of $T_{\mathcal{A}}[B]$ from $T_{\mathcal{A}}$ results in a forest F_B with t subtrees $T^1_{\mathcal{A}}, \ldots, T^t_{\mathcal{A}}$, where $A_i \in T^i_{\mathcal{A}}$.

Since any other tree $B' \in \mathcal{B}$ is edge-disjoint from B, for every $B' \in \mathcal{B} \setminus \{B\}$, the tree $T_{\mathcal{A}}[B']$ is a subtree of F_B. Hence, vertices from any two different subtrees $T^i_{\mathcal{A}}, T^j_{\mathcal{A}}$, $i \neq j$, are incomparable by $\prec'_b$, where $\prec'_b$ is the partial order obtained from $\prec_b$ by ignoring all the relations between vertices in B. Inductively, the vertices of each of these subtrees, can be ordered to preserve $\prec'_b$. We concatenate the t resulting orders into a single ordering $\prec_{\mathcal{A}}$ according to the order of $v_1, v_2, \ldots, v_t$ determined by $\prec_b$. It is easy to verify that $\prec_{\mathcal{A}}$ is $\prec_b$-preserving. □

Proof of Theorem 5.2. One direction of the theorem follows from Corollary 5.1. We only need to show the converse. Let $\mathcal{A}$ and $\mathcal{B}$ be two edge-disjoint spanning 3-forests which cover a tree T.

We define an ordering $\prec_b$ on $V(T)$ as follows: For each path $B_i = (v_1, v_2, v_3, v_4) \in \mathcal{B}$, define $v_1 \prec_b v_3 \prec_b v_2 \prec_b v_4$, where $v_{l+2}, \ldots, v_4$ will be absent if the path is of length $l \in \{0, 1, 2\}$. Notice firstly that if we place the vertices of B on the real line according to $\prec_b$ and consider the intervals $(-\infty, v_1)$, $(-\infty, v_2)$, $(v_3, +\infty)$ and $(v_4, +\infty)$, the corresponding interval graph will be a representation of the path B.

Notice secondly that two vertices in different connected components of $\mathcal{B}$ are incomparable. Hence by Lemma 5.4, there exists a $\prec_b$-preserving total order $\prec_{\mathcal{A}}$ among the connected components of $\mathcal{A}$. Similarly, we define a partial order $\prec_a$ on $V(T)$ comparing only those pairs of vertices in the same connected component of $\mathcal{A}$ and a $\prec_a$-preserving total order $\prec_{\mathcal{B}}$ among the connected components of $\mathcal{B}$. In the remainder of the proof we assume that the elements of $\mathcal{A} = \{A_1, \ldots, A_k\}$ and $\mathcal{B} = \{B_1, \ldots, B_{k'}\}$ are indexed according to $\prec_{\mathcal{A}}$ and $\prec_{\mathcal{B}}$ respectively.

We now depict a four-sided ∂EPG representation of T. Let $v \in V(T)$ and suppose $v \in A_i \cap B_j$. Associate with v a grid-path P_v bending at the grid-point (i, j). Notice that the grid-paths that correspond to vertices in A_i (resp. B_j) all bend at column i (resp. row j). Moreover, their order along the columns (resp. rows) preserves their order in $\prec_b$ (resp. $\prec_a$). For each column i of the grid, direct the $|A_i|$ grid-paths with bend points on that column upwards or downwards so that their intersection graph is the path A_i. Similarly, for each row j of the grid, direct the B_j grid-paths with bend points on that row leftwards or rightwards so that their intersection graph is the path B_j.

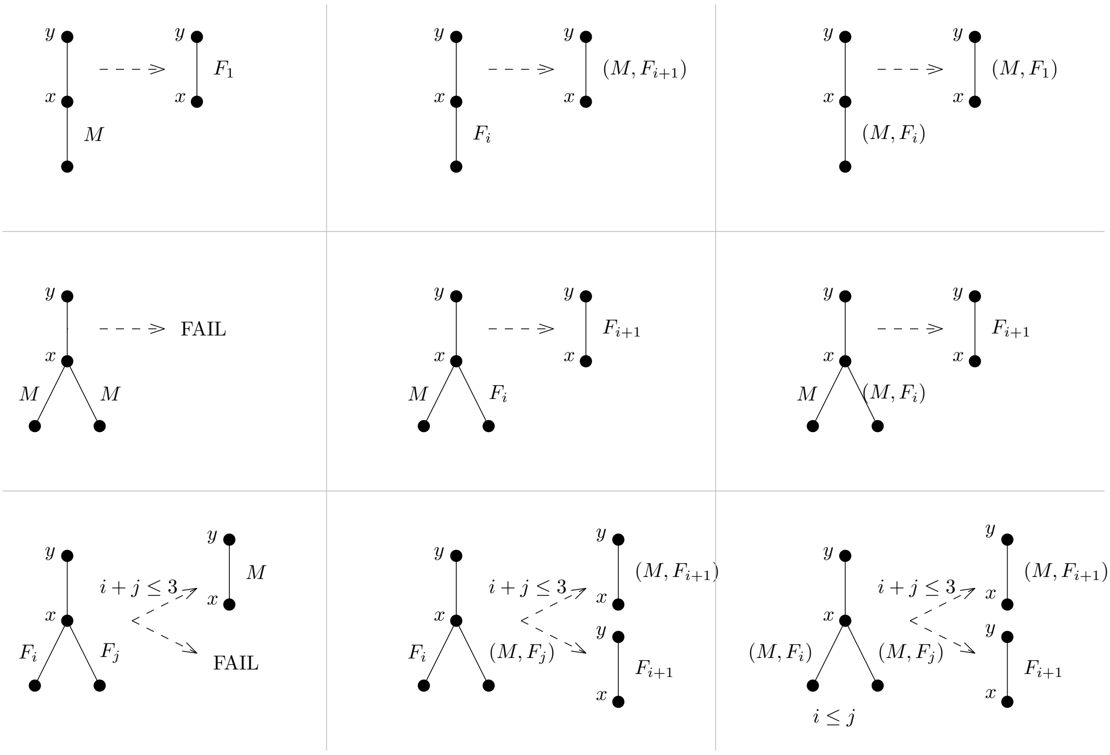


Figure 3: The label propagation rules for Algorithm 5.5

By construction, grid-paths P_v and P_u share a vertical (resp. horizontal) grid-edge iff the corresponding vertices of T are adjacent, and the edge connecting them is covered by $\mathcal{A}$ (resp. $\mathcal{B}$), which concludes the proof.

The proof for the cases of three-sided and two-sided representations is similar by directing all grid paths to the same direction. Assume $\mathcal{A}$ is a disjoint collection of edges, then at most two grid-paths bend at the same column, and can be directed downwards. Similarly, if both $\mathcal{A}$ and $\mathcal{B}$ are disjoint collections of edges, then every grid-paths can be directed leftward and downwards, resulting with a $\neg$-shape grid-path. □

Theorem 5.2 tells us that the problem of recognizing ∂EPG trees is the same as deciding whether the linear 3-arboricity of a tree is at most 2. In [7], Chang et al., give a linear time algorithm to find the linear k-arboricity of a tree. We can use the same algorithm to recognize ∂EPG trees. The case with three-sided ∂EPG trees gets a bit more tricky because of the asymmetry in the cover. Still, extending the idea in [7], we give a linear-time algorithm for recognizing three-sided ∂EPG trees.

Algorithm 5.5. The input to the algorithm is a tree T. The algorithm decides whether T can be covered by a matching M and a linear 3-forest F.

Step 1 : If the maximum degree of T is more than 3, reject. Otherwise, root the tree at a leaf vertex so that every vertex in the tree has at most 2 children.

The algorithm tries to label each edge e of T, starting with the leaf-edges and going up, such that the label encodes the possible places for e in the cover: (i) the label M denotes that e can belong to M but not to F. (ii) the label F_i denotes that e cannot belong to M but e can be the i-th edge of a P_4 in F, counting towards the root. (That is, if we consider a P_4 as a subtree T' of T rooted at the vertex in the P_4 nearest to the root of T, then all the edges incident to a leaf of T' will be labeled F_1, the edges incident to a leaf of T' with all the F_1-labeled edges removed will be labeled F_2 and so on.) (iii) the label (M, F_i) denotes that e can either belong to M or it can be the i-th edge of a P_4 in F, counting towards the root, and (iv) the label FAIL denotes that the edge e can be neither in M nor in F and hence T is not a three-sided ∂EPG.

Step 2 : Label every leaf-edge with (M, F_1).

A vertex of T is called *penultimate* if it is not a leaf and all its children are leaves.

Step 3 : Let x be a penultimate vertex in T farthest from the root. Accept T if x is the root. Otherwise, delete all the leafs of x and label the parent edge yx of x according to the rules depicted in Fig. 3, with the following two additional sanity rules: (i) if yx gets a label F_4, it is relabeled FAIL, and (ii) if yx gets a label (M, F_4), it is relabeled M.

Step 4 : Reject T if the label on yx is FAIL. Otherwise go back to Step 3.

Since the algorithm visits each edge at most twice, once to label it, and once to label its parent edge, it is clear that the algorithm runs in linear-time.

Correctness of Algorithm 5.5. Verifying the correctness of the algorithm is essentially verifying that, for each of the label propagation rules depicted in Fig. 3, the initial tree can be covered with M and F, respecting the constraints indicated by the labels on the leaf-edge, if and only if the same holds for the resulting tree.

For example, consider the first rule. If the leaf-edge at x has to be in M, then the edge yx has to be the first edge of a P_4 in F. Also, if yx is the first edge of a P_4 in F, then the leaf-edge at x can safely be in M. Hence, the first tree can be covered with a matching and a linear 3-forest such that the leaf-edge at x belongs to the matching if and only if the second tree can be covered by a matching and a linear 3-forest such that the edge yx is the first edge of a component in the linear 3-forest.

As a second example, consider the final rule. When $i + j \leq 3$, we have two options: (i) we can cover the two leaf edges with a P_{i+j} and the edge yx with M, or (ii) cover the right leaf-edge with M and cover the left leaf-edge and x with F. (Covering the left leaf-edge with M is at most as good as covering the right leaf-edge with M.) These two options give rise to the options indicated by the label on the edge xy in the resulting tree. When $i+j > 3$, the first option above is no longer available, and hence we have fewer options on that branch. The analysis for all the other rules is similar.

6 Conclusion and Open Questions

We introduce in this paper, the boundary generated EPG graphs, which more accurately model circuits where each wire (i.e., path) must be anchored at a terminal on the boundary of a rectangular grid. Several open questions remain.

We do not know yet whether one can efficiently recognize ∂EPG graphs. Though the problem is linear-time solvable on trees, we suspect that it might be NP-hard in general.

References

[1] L. Alcón, F. Bonomo, G. Durán, M. Gutierrez, P. Mazzoleni, B. Ries, and M. Valencia-Pabon, On the bend number of circular-arc graphs as edge intersection graphs of paths on a grid. Proc. LAGOS 2015, *Electronic Notes in Discrete Mathematics* 50 (2015), 249–254.

[2] N. Alon, Covering graphs with the minimum number of equivalence relations. *Combinatorica* 6 (1986), 201–206.

[3] A. Asinowski and B. Ries, Some properties of edge intersection graphs of single bend paths on a grid. *Discrete Mathematics* 312 (2012), 427–440.

[4] A. Asinowski and A. Suk, Edge intersection graphs of systems of grid paths with bounded number of bends. *Discrete Applied Mathematics* 157 (2009), 3174–3180.

[5] T. Biedl and M. Stern, On edge intersection graphs of k-bend paths in grids. *Discrete Mathematics and Theoretical Computer Science* 12 (2010), 1–12.

[6] K. Cameron, S. Chaplick, and C.T. Hoàng, Edge intersection graphs of L-shaped paths in grids. Proc. LAGOS 2013, *Electronic Notes in Discrete Mathematics* 44 (2013), 363–369. Journal version: *Discrete Applied Mathematics* 210 (2016), 185–194.

[7] G.J. Chang, B.L. Chen, H.L. Fu, and K.C. Huang, Linear k-arboricities on trees. *Discrete Applied Mathematics* 103 (2000), 281–287.

[8] E. Cohen, M.C. Golumbic, and B. Ries, Characterizations of cographs as intersection graphs of paths on a grid. *Discrete Applied Mathematics* 178 (2014), 46–57.

[9] P. Duchet, Représentations, noyaux en théorie des graphes et hypergraphes. *Thèse dâĂŹÉtat, Université Paris VI*, 1979.

[10] M.C. Francis and A. Lahiri, *VPG and EPG bend-numbers of Halin graphs.* *Discrete Applied Mathematics* 215 (2016), 95–105.

[11] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980. Second edition: *Annals of Discrete Mathematics* 57, Elsevier, Amsterdam, 2004.

[12] M.C. Golumbic, The recognition problem of B_1-EPG and B_1-VPG graphs, in: *Exploiting Graph Structure to Cope with Hard Problems.* Andreas BrandstÃďdt, Martin Charles Golumbic, Pinar Heggernes, Ross M. McConnell, eds., (Dagstuhl Seminar 11182). *Dagstuhl Reports* 1(5), 29–46 (2011), Problem 4.8.

[13] M.C. Golumbic, M. Lipshteyn and M. Stern, Edge intersection graphs of single bend paths on a grid. *Networks* 54 (2009), 130–138.

[14] M.C. Golumbic, M. Lipshteyn and M. Stern, Single bend paths on a grid have strong Helly number 4. *Networks* 62 (2013), 161–163.

[15] M. Habib and P. Peroche, Some problems about linear arboricity. *Discrete Mathematics* 41 (1982), 219–220.

[16] D. Heldt, K. Knauer and T. Ueckerdt, Edge-intersection graphs of grid paths: the bend-number. *Discrete Applied Mathematics* 167 (2014), 144–162.

[17] D. Heldt, K. Knauer and T. Ueckerdt, On the bend-number of planar and outerplanar graphs. *Discrete Applied Mathematics* 179 (2014), 109–119.

[18] P. Hell and J. Nešetřil, *Graphs and Homomorphisms*. Oxford University Press, 2004.

[19] L. Lovász, J. Nešetřil and A. Pultr, On the product dimension of graphs. *J. Combin. Theory Ser.* B 29 (1980), 47–67.

[20] F. Maffray, and B.A. Reed, A description of claw-free perfect graphs. *Journal of Combinatorial Theory, Series B* 75.1 (1999), 134–156.

[21] J. Nešetřil and V. Rödl, A simple proof of the Galvin-Ramsey property of graphs and a dimension of a graph. *Discrete Mathematics* 23 (1978), 49–55.

[22] M. Pergel and P. Rzążewski, *On edge intersection graphs of paths with 2 bends. Discrete Applied Mathematics* 226 (2017), 106–116.