

Exploring pairwise compatibility graphs

T. Calamoneri^{a,1}, E. Montefusco^c, R. Petreschi^{a,2}, B. Sinimeri^{b,3},

^a*Department of Computer Science, Sapienza” University of Rome, Italy.*

^b*INRIA and Université de Lyon; Université Lyon 1, LBBE, CNRS UMR558, France.*

^c*Department of Mathematics, Sapienza University of Rome, Italy.*

Abstract

A graph $G = (V, E)$ is called a pairwise compatibility graph (PCG) if there exists a tree T , a positive edge weight function w on T , and two non-negative real numbers $d_{min} \leq d_{max}$, such that each leaf l_u of T corresponds to a vertex $u \in V$ and there is an edge $(u, v) \in E$ if and only if $d_{min} \leq d_{T,w}(l_u, l_v) \leq d_{max}$ where $d_{T,w}(l_u, l_v)$ is the sum of the weights of the edges on the unique path from l_u to l_v in T .

In this paper we analyze the class of PCGs in relation to two particular subclasses resulting from the cases where the constraints on the distance between the pairs of leaves concern only d_{max} (LPG) or only d_{min} (mLPG). In particular, we show that the union of LPG and mLPG classes does not coincide with the whole class of PCGs, their intersection is not empty, and that neither of the classes LPG and mLPG is contained in the other.

Finally, we study the closure properties of the classes PCG, mLPG and LPG, under some common graph operations. In particular, we consider the following operations: adding an isolated or universal vertex, adding a pendant vertex, adding a false or a true twin, taking the complement of a graph and taking the disjoint union of two graphs.

Keywords: PCG, leaf power graphs, threshold graphs, graph complement, disjoint union of graphs, twins.

Email addresses: calamoneri@di.uniroma1.it (T. Calamoneri),
montefusco@mat.uniroma1.it (E. Montefusco), petreschi@di.uniroma1.it (R. Petreschi),
blerina.sinimeri@inria.fr (B. Sinimeri)

¹Partially supported by the Italian Ministry of Education and University PRIN project COGENT (Computational and GamE-theoretic aspects of uncoordinated NeTworks).

²Partially supported by the Italian Ministry of Education and University PRIN project Algo-DEEP (Algorithmic Challenges for Data-intensive Processing on Emerging Computing Platforms).

³This work was done when the author was a postdoctoral member of the Department of Computer Science, Sapienza University of Rome.

1. Introduction

A graph $G = (V, E)$ is a *pairwise compatibility graph* (PCG) if there exists a tree T , a positive edge weight function w on T and two non-negative real numbers d_{min} and d_{max} , $d_{min} \leq d_{max}$, such that each vertex $u \in V$ is uniquely associated to a leaf l_u of T and there is an edge $(u, v) \in E$ if and only if $d_{min} \leq d_{T,w}(l_u, l_v) \leq d_{max}$ where $d_{T,w}(l_u, l_v)$ is the sum of the weights of the edges on the unique path from l_u to l_v in T . In such a case, we say that G is a PCG of T for d_{min} and d_{max} ; in symbols, $G = PCG(T, w, d_{min}, d_{max})$. In Fig.1 the corresponding pairwise compatibility graph for a weighted tree and two values d_{min} and d_{max} is depicted.

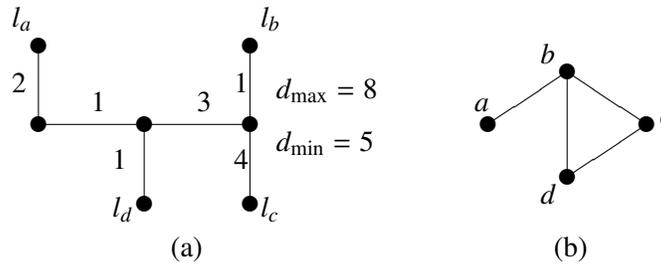


Figure 1: (a) A pairwise compatibility tree. (b) the corresponding pairwise compatibility graph.

Clearly, if a tree T , an edge weight function w and two values d_{min} and d_{max} are given, the construction of a PCG is a trivial problem. We focus on the reverse of this problem, i.e., given a graph G we have to find out a tree T , an edge weight function w and two suitable values, d_{min} and d_{max} , such that $G = PCG(T, w, d_{min}, d_{max})$. Such a problem is called the *pairwise compatibility tree construction problem*. Meanwhile, the *pairwise compatibility graph recognition problem* asks to determine whether a given graph is a pairwise compatibility graph or not (without necessarily exhibiting the corresponding pairwise compatibility tree T).

The concept of pairwise compatibility was introduced in [10] in a computational biology context. A fundamental problem in computational biology is the reconstruction of the phylogeny, or the evolutionary history, of a set of species, populations, individuals or genes (generally referred as taxa) usually represented as a phylogenetic tree. In a phylogenetic tree each leaf represents a distinct known taxon and the edges indicate possible ancestors that might have led to this set of taxa. Thus, the filogenetic tree reconstruction problem is the following: Given a set of taxa find a fully labeled phylogenetic tree that "best" explains the given data. Due to the difficulty in determining the criteria for an "optimal" tree, the performance of the reconstruction algorithms is evaluated through experimental methods.

However, as the tree reconstruction problem is proved to be NP-hard, these heuristics are usually slow and as real phylogenetic trees are of a large dimension, testing these algorithms on real data is difficult. Thus, it is interesting to find efficient ways to sample subsets of taxa from a large phylogenetic tree, subject to some biologically-motivated constraints, in order to test the reconstruction algorithms on the subtree induced by the sample. In particular, given a phylogenetic tree, we want to select a subset of leaves, uniformly at random, according to certain constraints on the subtree induced by the sample. An interesting constraint is the one dealing with the pairwise distance between any two taxa in the induced subtree. This is important because, as observed in [17], very close or very distant taxa can create problems for phylogeny reconstruction algorithms. Given a phylogenetic tree T and two integers d_{min}, d_{max} , the aim is to select a sample from the leaves of T such that the pairwise distance of any two leaves in the sample is at least d_{min} and at most d_{max} . By the definition of PCG, a subset of leaves of a tree T such that the distance of any two leaves belongs to the interval $[d_{min}, d_{max}]$, corresponds to a subset of vertices in the pairwise compatibility graph G of T , in which any two of them are connected by an edge (for example the set of vertices $\{b, c, d\}$ in Fig.1). Thus, it is clear that this sampling problem reduces to selecting a clique uniformly at random from the graph $PCG(T, w, d_{min}, d_{max})$. In view of this, it is interesting to determine which graphs are PCGs as it would help to solve the clique problem [9] in polynomial time for a restricted class of graphs. Moreover, since the space and the time required by the polynomial algorithm for the sampling problem with pairwise leaf distance constraints is not feasible for large phylogenetic trees, the investigation of properties of PCGs may give new insights for finding more efficient algorithms for this problem.

Initially, the authors of [10] conjectured that every graph is a PCG, but this conjecture has been confuted in [15], where a particular bipartite graph is proved not to be a PCG. However, there are several known specific graph classes of pairwise compatibility graphs, e.g., graphs with five vertices or less [13], graphs with at most seven vertices [6], cliques and disjoint union of cliques [1], chordless cycles and single chord cycles [16], ladder graphs [14] and some particular subclasses of bipartite graphs [15].

As we have seen, the pairwise compatibility concept is defined with respect to two bounds concerning d_{min} and d_{max} . If we relax these conditions, requiring only that the distance between some pairs of leaves is smaller than or equal to d_{max} (i.e. we set $d_{min} = 0$) then we are considering a particular subclass of PCGs, namely the leaf power graphs (LPGs). More formally, $G(V, E)$ is a *leaf power graph* if there exists a tree T , a positive edge weight function w on T and a nonnegative number d_{max} such that there is an edge (u, v) in E if and only if for their corresponding leaves in T , l_u, l_v , we have $d_{T,w}(l_u, l_v) \leq d_{max}$, in symbols $G = LPG(T, w, d_{max})$.

Although there has been a lot of work on this class of graphs (see e.g. [1, 2, 4, 5, 8, 11]), a complete description of leaf power graphs is still unknown.

In [7] the subclass of PCGs when the constraint concerns only the minimum distance has been introduced. In this case d_{max} is set to the maximum weighted distance between any pair of leaves, i.e. there is an edge in E if and only if the corresponding leaves are at a distance greater than d_{min} in the tree. This subclass of PCGs is called *mLPG (minimum Leaf Power Graphs)* for highlighting the similitude with LPGs, even if these graphs are not power of trees. More formally, $G = (V, E)$ is an mLPG if there exists a tree T , a positive edge weight function w on T and an integer d_{min} such that there is an edge (u, v) in E if and only if for their corresponding leaves in T l_u, l_v we have $d_{T,w}(l_u, l_v) \geq d_{min}$; in symbols, $G = mLPG(T, w, d_{min})$. The relations between the classes of PCGs, LPGs and mLPGs are studied in Section 4.

Furthermore, it is known that many graph classes can be built by means of recursive applications of some graph operations. For this reason we focus on this issue, trying to understand whether the PCG, LPG and mLPG classes remain close under some well-known graph operations. More specifically, when a graph operation is performed on one or more PCGs (or LPGs or mLPGs), we investigate whether the resulting graph still belongs to the class of PCGs (or LPGs or mLPGs).

This paper is organized as follows: in Sections 2 and 3, we present information useful for the forthcoming work. Namely, in Section 2 we introduce some terminology and recall some known concepts, while in Section 3 we prove some technical results. As we said, in Section 4 we study the relations between the PCG, LPG and mLPG classes. In particular, we show that: the union of LPG and mLPG does not coincide with the whole PCG class, neither of the classes LPG and mLPG is contained in the other and their intersection is not empty, as threshold graphs belong to both the classes mLPG and LPG and in each of the cases the associated trees can be stars. In Section 5, we investigate whether and in which cases the PCG class and its subclasses LPG and mLPG are closed under several graph operations. To this purpose we study the operations of adding an isolated vertex, an universal vertex and a pendant vertex. Moreover, we analyze the operations of adding a false or a true twin and finally we study the graph complement and the disjoint union of two given graphs. All the claims presented, except for the results regarding the complement of a graph, are proved in a constructive way, so we not only solve the PCG recognition problem but we also exhibit the corresponding tree.

In the last section of this paper we summarize the achieved results and propose some open questions arisen by this work.

2. Basic definitions

In this section we introduce some terminology and recall some definitions that will be used in the rest of the paper. The reader is referred to [3] for undefined terms and notation.

In this paper we consider simple graphs $G = (V, E)$ with vertex set V and edge set E . For a vertex v of a graph G , $N(v) = \{u | (u, v) \in E\}$ denotes the *open neighborhood* while $N[v] = N(v) \cup \{v\}$ is the *closed neighborhood*. For a graph $G = (V, E)$ and a subset $V' \subseteq V$ we denote by $N_{V'}(v)$ ($N_{V'}[v]$) the open (closed) neighborhood of v in the graph induced by V' .

Two vertices in a graph are called *true* (respectively *false*) *twins* if they have the same closed (respectively open) neighborhood.

A vertex of a graph is *universal* (*isolated*) if it is adjacent to all (none of) the other vertices of the graph.

A vertex v of a graph G having degree one is called *pendant vertex*.

Given a graph $G = (V, E)$, its *complement* G^C has the same vertex set V of G and two vertices are adjacent if and only if they are not adjacent in G . Extending the concept of the complement from single graphs to graph classes, we have that, two classes of graphs A and B , are complements of each other if for each graph G belonging to A , it holds that G^C belongs to B , and viceversa.

The *disjoint union* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph whose vertex and edge sets are the disjoint unions of the vertex and edge sets of G_1 and G_2 , respectively.

A *caterpillar* is a tree in which all the vertices are within distance one of a central path which is called the *spine*.

A *cactus* is a connected graph in which any two simple cycles have at most one vertex in common. Equivalently, every edge in such a graph may belong to at most one cycle. We will denote by C the class of cacti with at least a cycle of length $n \geq 5$.

A graph $G = (K, S, E)$ is said to be *split* if there is a vertex partition $V = K \cup S$ such that the subgraphs induced by K and S are complete and stable, respectively.

A set M of edges is a *perfect matching* of dimension m of A onto B if and only if A and B are disjoint subsets of vertices both of cardinality m and each vertex in A is adjacent to exactly one vertex in B and no two edges share a point. We say that the split graph $G = (K, S, E)$ is a *split matching* if the subset of edges in E not belonging to the clique forms a perfect matching. We denote by SM the class of split matching graphs.

An *antimatching* of dimension m of A onto B is a set of edges such that the non-edges between A and B form a perfect matching. We say that the split graph $G = (K, S, E)$ is a *split antimatching* if the subset of edges in E not belonging to

the clique forms an antimatching. We denote by \mathcal{SA} the class of split antimatching graphs.

Given a split matching (respectively, split antimatching) $G = (K, S, E)$, with $|K| = |S| = n$, in the following we will call the vertices $k_1, \dots, k_n \in K$ and $s_1, \dots, s_n \in S$ and we will implicitly assume that they are ordered so that k_i and s_i are connected (respectively, not connected).

Given two split graphs $G_1 = (K_1, S_1, E_1)$ and $G_2 = (K_2, S_2, E_2)$ their *composition* $G_1 \circ G_2$ is formed by taking the disjoint union of G_1 and G_2 and adding all the edges $\{u, v\}$ such that $u \in K_1$ and $v \in (K_2 \cup S_2)$. Observe that $G_1 \circ G_2$ is again a split graph.

Given a sequence of t split graphs $G_i = (K_i, S_i, E_i)$ with $i = 1, \dots, t$, we say the graph $H = G_1 \circ \dots \circ G_t$ is a *split matching (antimatching) sequence* if each of the graphs G_i is either a split matching (antimatching), or a stable graph or a clique graph.

Given a connected graph G whose distinct vertex degrees are $\delta_1 > \dots > \delta_r$, we define $B_i = \{v \in V(G) : \deg(v) = \delta_i\}$, for any $i = 1, \dots, r$. The sets B_i are usually referred as *boxes* and the sequence B_1, \dots, B_r is called the *degree partition of G into boxes*. Given a graph G with degree partition B_1, \dots, B_r , G is a *threshold graph* if and only if for all $u \in B_i, v \in B_j, u \neq v$, we have $(u, v) \in E(G)$ if and only if $i + j \leq r + 1$. We will denote by \mathcal{T} the class of threshold graphs.

In this paper we will consider some well-known operations on graphs in relation to the property of being PCG, mLPG or LPG. More specifically we will consider the following operations:

1. adding an isolated vertex;
2. adding an universal vertex;
3. adding a pendant vertex;
4. adding a vertex that is a false twin for the old vertex v ;
5. adding a vertex that is a true twin for the old vertex v ;
6. graph complement;
7. disjoint union of two graphs

It is known that many graph classes can be obtained by recursively applying one or more graph operations among the previous list (see for e.g. [3]). In particular we will deal with the following graph classes:

Threshold graphs can be recursively built from the one vertex graph by adding either an universal or an isolated vertex.

Quasi-threshold graphs can be recursively built from the one vertex graph by adding a universal vertex or by the disjoint union of two quasi-threshold graphs.

Distance-hereditary graphs can be recursively built from the one vertex graph by adding either an isolated vertex or a new pendant vertex or a new vertex that is a false or true twin for an old vertex v . Notice that if the constructed distance hereditary graph is bipartite, the operation concerning the addition of a true twin is never used.

3. Technical results

In this section we prove some technical results that will be used in the rest of the paper.

Throughout the paper we will assume w.l.o.g. that any non trivial tree T with at least three leaves does not contain vertices of degree 2. Indeed, we could merge the two incident edges of a vertex of degree 2 into a unique edge whose weight is the sum of the weights of the original edges. In both cases the distance between the leaves does not change.

Proposition 1. *Let G be a graph that does not belong to some class L from the set $\{PCG, LPG, mLPG\}$; then every graph H that contains G as an induced subgraph does not belong to L either.*

Given two vertices u, v in a tree T , P_{uv} denotes the unique path in T connecting the vertices u and v . A *subtree induced by a set of leaves* of T is the minimal subtree of T which contains those leaves. T_{uvx} denotes the subtree of a tree induced by three leaves u, v and x .

Lemma 1. [15] *Let T be a tree, w be a positive edge weight function on T , and u, v, x be three leaves of T such that P_{uv} is the largest path in T_{uvx} (i.e. $d_{T,w}(u, v) \geq d_{T,w}(u, x)$ and $d_{T,w}(u, v) \geq d_{T,w}(v, x)$). Let y be a leaf of T other than u, v and x . Then, either $d_{T,w}(x, y) \leq d_{T,w}(u, y)$ or $d_{T,w}(x, y) \leq d_{T,w}(v, y)$.*

Observe that here we always assume d_{min}, d_{max} and the weight of the edges of the tree of a PCG all positive real numbers. In the original problem concerning the LPGs, these quantities are required to be natural numbers. This is not a loss of generality as in [5] it is proved that it is equivalent to consider positive real numbers instead of naturals for LPGs. Here we give a simpler prove of this result while extending it to the general case of PCGs.

Lemma 2. *Let $G = PCG(T, w, d_{min}, d_{max})$, where d_{min}, d_{max} are non-negative real numbers and the weight $w(e)$ of each edge e of T is a positive real number. Then it is possible to choose natural numbers $\hat{w}, \hat{d}_{min}, \hat{d}_{max}$ such that $G = PCG(T, \hat{w}, \hat{d}_{min}, \hat{d}_{max})$.*

Proof. We will assume that $d_{min}, d_{max} > 0$. This, is not restrictive as if d_{min} or d_{max} is equal to 0 we can increase by 1 all the weights of the edges incident to a leaf in the tree. Then, increasing d_{min} and d_{max} by 2 the modified tree defines the same pairwise compatibility graph.

Now, the result will be proved in two steps. First we show that it is possible to approximate all the real numbers with rational numbers, and then we will define $\hat{d}_{min}, \hat{d}_{max} \in \mathbb{N}$ and \hat{w} such that $\hat{w}(e)$ is a natural number for each edge e of T .

Since \mathbb{Q} is dense in \mathbb{R} , for any $\epsilon > 0$ and any $r \in \mathbb{R}$ there exists $q_r \in \mathbb{Q}$ such that

$$0 < r - q_r < \epsilon. \quad (1)$$

Note that q_r is a rational approximation of r such that $q_r < r$.

Now let us introduce the following two numbers:

$$L = \min_{(u,v) \in E} \{|d_{min} - d_{T,w}(l_u, l_v)|, |d_{max} - d_{T,w}(l_u, l_v)|\}, \quad N = \binom{n}{2},$$

L is the smallest distance between the weighted distances of the paths on the tree and the quantities d_{min}, d_{max} and N is the number of the paths joining two leaves of T . Suppose first $L > 0$.

Define $\epsilon_0 = L/3$ and $\epsilon_1 = L/3N$.

By (1), we can fix some rational numbers such that it holds:

$$\begin{aligned} 0 < d_{min} - q_{d_{min}} < \epsilon_0, \quad 0 < d_{max} - q_{d_{max}} < \epsilon_0, \\ 0 < w(e) - q_{w(e)} < \epsilon_1 \text{ for each edge } e \text{ of } T. \end{aligned}$$

If we denote by \tilde{w} the weight function such that $\tilde{w}(e) = q_{w(e)}$, it is not difficult to check that $G = PCG(T, \tilde{w}, q_{d_{min}}, q_{d_{max}})$, since all the distances of the paths decrease for a positive quantity (less than $L/3$, by the choice of ϵ_1) and, similarly, the bounds $q_{d_{min}}$ and $q_{d_{max}}$ (recall that $\epsilon_0 < L/3$), so that all the new quantities preserve the original ordering.

Consider now $q_{d_{min}}, q_{d_{max}}$ and $\tilde{w}(e)$ for each e of T ; they are all numbers in \mathbb{Q} , so let K be the least common multiple of their denominators. Define:

$$\hat{w}(e) = Kq_{w(e)} \text{ for each edge } e \text{ of } T, \quad \hat{d}_{min} = Kq_{d_{min}}, \quad \hat{d}_{max} = Kq_{d_{max}} \in \mathbb{N}.$$

Since $K > 0$ and the multiplication by K preserves the ordering, the claim follows.

Finally, observe that it is not restrictive to assume $L > 0$. Indeed, if $L = 0$, i.e. there are some pair of leaves whose distance is exactly d_{max} or d_{min} then we define

$$r = \min_{(u,v)} \{d_{T,w}(l_u, l_v) : d_{T,w}(l_u, l_v) > d'_{max}\}$$

and set $d'_{max} = \frac{1}{2}(r - d'_{max})$. Similarly, we can define d'_{min} . Observe that $G = PCG(T, w, d'_{min}, d'_{max})$ and there are no pair of leaves whose distance equals d'_{max} or d'_{min} . Thus, $L > 0$ and thus we return to the previous case. \square

Lemma 3. *Let $G = PCG(T, w, d_{min}, d_{max})$. If $d_{min} = d_{max} = d$ then there exist $\epsilon > 0$ such that $G = PCG(T, \hat{w}, d - \epsilon, d + \epsilon)$*

Proof. First, observe that if there is no pair of leaves whose distance is exactly d then the graph is null and it is easy to find an $\epsilon > 0$ sufficiently small such that the claim holds. Otherwise, there is some pair of leaves at a distance d ; then we define

$$r_{max} = \min_{(u,v)} \{d_{T,w}(l_u, l_v) : d_{T,w}(l_u, l_v) > d\},$$

$$r_{min} = \min_{(u,v)} \{d_{T,w}(l_u, l_v) : d_{T,w}(l_u, l_v) < d\}.$$

Define $2\epsilon = \min\{d - r_{min}, r_{max} - d\}$. It is easy to check that $G = PCG(T, w, d - \epsilon, d + \epsilon)$. \square

Lemma 4. *Let $G = PCG(T, w, d_{min}, d_{max})$. For any two positive numbers $m < M$ such that*

$$m > \frac{M(d_{min} - 2 \min w(e))}{(d_{max} - 2 \min w(e))}, \quad (2)$$

there is an edge weight \tilde{w} on T such that $G = PCG(T, \tilde{w}, m, M)$.

Proof. Observe that from Lemma 3 we can assume $d_{min} < d_{max}$. Given the tree T with edge weight function w , we define a new edge weight function \tilde{w} as follows: for each edge $e = (u, v) \in T$ non incident to any leaf of T , define $\tilde{w}(e) = a \cdot w(e)$ for an opportune positive constant a that we will define later. Otherwise, if u is a leaf in T , define the weight of its unique incident edge $e = (u, v)$ as $\tilde{w}(e) = a \cdot w(e) + b/2$ for an opportune real constant b that we will define later.

Clearly, for any two leaves u and v of T such that $d_{T,w}(u, v) = x$ it turns out that $d_{T,\tilde{w}}(u, v) = ax + b$.

Hence, we have to choose a, b in order to satisfy

$$\begin{cases} ad_{min} + b = m \\ ad_{max} + b = M, \end{cases}$$

it is easy to see that we have to fix $a > 0$ and $b \in \mathbb{R}$ such that

$$\begin{cases} a = \frac{M - m}{d_{max} - d_{min}} \\ b = m - ad_{min}. \end{cases} \quad (3)$$

These relations are an admissible choice for the weight function \tilde{w} provided that

$$\min_{e = (u, v) \text{ with } u \text{ leaf of } T} a \cdot w(e) + \frac{b}{2} > 0,$$

since it is not possible that \tilde{w} possesses a nonpositive component, this means that

$$a > 0 \text{ and } b > -2 \min_{e = (u, v) \text{ with } u \text{ leaf of } T} a \cdot w(e).$$

Note that the positivity of a is equivalent to require that $m < M$ (since it holds $d_{max} > d_{min}$) and the hypothesis on b becomes

$$m = ad_{min} + b > ad_{min} - 2 \min_{e = (u, v) \text{ with } u \text{ leaf of } T} a \cdot w(e) > 0,$$

putting these relations into (3) it is possible to verify that are equivalent to (2), and this concludes the proof. \square

The following proposition shows that the LPG and mLPG classes are complements of each other.

Proposition 2. *The complement of every graph in LPG is in mLPG and conversely, the complement of every graph in mLPG is in LPG.*

Proof. Observe that the proof trivially holds for complete graphs. So, consider first a graph $G \neq K_n$ such that $G = LPG(T, w, d_{max})$. This means that two vertices u and v are adjacent in G if and only if $d_{T,w}(l_u, l_v) \leq d_{max}$, where l_u, l_v are the corresponding leaves in T . Recall that by definition G^C does not contain (u, v) . Let $r = \min_{\{u,v\} \notin E(G)} d_{T,w}(l_u, l_v)$. It is enough to define $d_{min} = r > d_{max}$. It is trivial to verify that $G^C = mLPG(T, w, d_{min})$.

The same argument can be used to prove that the complement of a mLPG is a LPG. \square

4. Relationships between PCG, LPG and mLPG classes

In this section we study the relationships between the classes of PCGs, LPGs and mLPGs⁴. First, in Subsection 4.1 we show that the union of mLPG and LPG classes does not contain the whole class of PCGs. Next, in Subsection 4.2 we show that their intersection $LPG \cap mLPG$ is not empty. Finally, in Subsection 4.3 we show that neither of the classes LPG and mLPG is contained in the other one by providing for each of these classes a particular graph which is proper to it. These relations are graphically shown in Figure 2.

⁴In [7] a preliminary version of the results presented in this Section are reported.

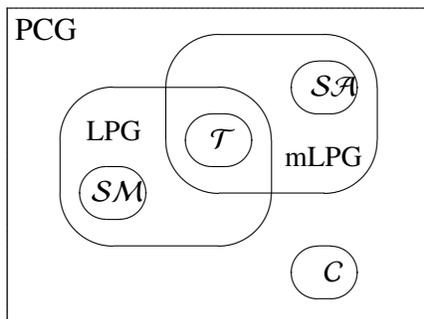


Figure 2: Relationships between PCG, LPG and mLPG.

4.1. $PCG \supset (LPG \cup mLPG)$

Let us prove that the PCG class does not coincide with the union of LPGs and mLPGs. To this aim, observe that it is known that any cycle is a PCG [16] and that LPGs are a subclass of strongly chordal graphs (see, for example, [1]); so cycles of length $n \geq 5$ are not LPGs as they are not strongly chordal graphs. The following lemma states that cycles do not belong to mLPGs, deducing that $(LPG \cup mLPG) \subset PCG$.

Lemma 5. *Let C_n be a cycle of length $n \geq 5$, then $C_n \notin mLPG$.*

Proof. Let v_1, \dots, v_n be the ordered vertices of a cycle C_n with $n \geq 5$. Suppose by contradiction that $C_n = mLPG(T, w, d_{min})$ and let l_i be the leaf in T corresponding to the vertex v_i , for any $i \leq n$. Let v_1, v_2, v_3 be the first three consecutive vertices in C_n and consider the largest path in $T_{l_1 l_2 l_3}$. As $(v_1, v_3) \notin E$ (as $n \geq 5$) then $d_{T,w}(l_1, l_3) < d_{min}$. Hence, the largest path must be one from $P_{l_1 l_2}$ and $P_{l_2 l_3}$.

Suppose first the largest path is $P_{l_1 l_2}$. Using Lemma 1 with $x = l_4$ we have that either $d_{min} \leq d_{T,w}(l_4, l_3) \leq d_{T,w}(l_4, l_2)$ or $d_{min} \leq d_{T,w}(l_4, l_3) \leq d_{T,w}(l_4, l_1)$, deducing that at least one from the edges (v_4, v_2) and (v_4, v_1) must be in C_n , a contradiction.

If $P_{l_2 l_3}$ is the largest path, we arrive at the same result by taking this time $x = l_n$. This concludes the proof. \square

From the previous lemma and from Proposition 1 it easily descends that the class C of cacti with at least one cycle of length $n \geq 5$ belongs neither to LPG nor to mLPG.

4.2. $LPG \cap mLPG \neq \emptyset$

We prove that the intersection of LPG and mLPG classes is not empty by showing that threshold graphs belong to $LPG \cap mLPG$. In our proof we use the definition of threshold graphs in terms of boxes.

Theorem 1. *Let G be a threshold graph, then $G \in \text{LPG} \cap \text{mLPG}$. In both of the cases a tree T , an edge weight function w and a value d_{\min} or d_{\max} associated to G can be found in polynomial time.*

Proof. Let G be a threshold graph on n vertices and let B_1, \dots, B_r be the degree partition into boxes of G . As the pairwise compatibility tree T we consider an n -leaf star with center a vertex c .

To prove that $G \in \text{LPG}$, we define the edge weight function w that, for each vertex v of G , assigns weight i to the edge (l_v, c) in T if $v \in B_i$. Define $d_{\max} = r + 1$. As for each $u \in B_i, v \in B_j, u \neq v$, we have $(u, v) \in E(G)$ if and only if $i + j \leq r + 1$; it is straightforward that $G = \text{LPG}(T, w, d_{\max})$.

On the other hand, to prove $G \in \text{mLPG}$ for any $v \in V(G)$, define the edge weight function w' that assigns $r + 1 - i$ to the edge (l_v, c) in T if $v \in B_i$. Note that, as $i \leq r$ we assign nonnegative weights to the edges of the star. Define $d_{\min} = r + 1$. For any two vertices $v \in B_i$ and $u \in B_j$, we have that if $i + j \leq r + 1$ (meaning that $(u, v) \in E(G)$) then $d_{T,w}(l_u, l_v) = 2(r + 1) - (i + j) \geq r + 1 = d_{\min}$. Otherwise, if $i + j > r + 1$ (meaning that $(u, v) \notin E(G)$) then $d_{T,w}(l_u, l_v) = 2(r + 1) - (i + j) < r + 1 = d_{\min}$. Easily, $G = \text{mLPG}(T, w', d_{\min})$ and this concludes the proof. \square

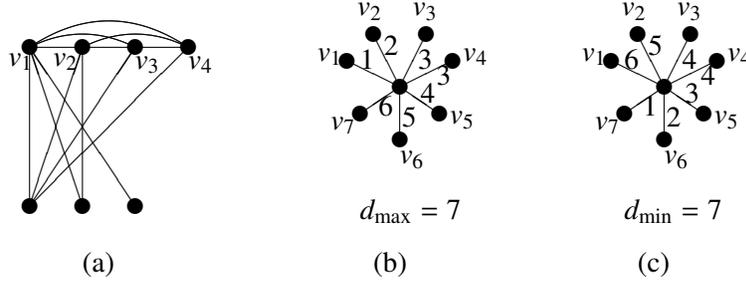


Figure 3: (a) A threshold graph (b) The pairwise compatibility tree witnessing that G is a LPG (c) The pairwise compatibility tree witnessing that G is an mLPG.

4.3. $\text{LPG} \setminus \text{mLPG} \neq \emptyset$ and $\text{mLPG} \setminus \text{LPG} \neq \emptyset$

In conclusion of this section we show that neither of the classes LPG and mLPG is contained in the other one by providing, for each of these classes, a particular graph which is proper to it.

Theorem 2. *Let G be a split matching graph, then $G \notin \text{mLPG}$ and $G \in \text{LPG}$. A tree T , an edge weight function w and a value d_{\max} associated to G can be found in polynomial time.*

The proof will follow immediately by the next two lemmas.

Lemma 6. *Let G be a split matching graph. Then $G \notin mLPG$.*

Proof. Given a split matching graph $G = (K, S, E)$ with $|K| = |S| = n$, we assume by contradiction $G = mLPG(T, d_{min})$. Then let a_1, a_2, a_3 be three leaves of T corresponding to three vertices of K , k_1, k_2, k_3 . Without loss of generality let $P_{a_1 a_2}$ be the largest path in the subtree $T_{a_1 a_2 a_3}$. Consider the vertex s_3 in S associated to the leaf b_3 in T , with $(k_3, s_3) \in E$. From Lemma 1 we deduce that either $d_T(b_3, a_3) \leq d_T(b_3, a_2)$ or $d_T(b_3, a_3) \leq d_T(b_3, a_1)$. The existence of the edge (k_3, s_3) in G implies $d_T(b_3, a_3) \geq d_{min}$, therefore one from (k_1, s_3) and (k_2, s_3) must be an edge in G , a contradiction. \square

Lemma 7. *Let G be a split matching graph, then $G \in LPG$. A tree T , an edge weight function w and a value d_{max} associated to G can be found in polynomial time.*

Proof. Given a split matching graph $G = (K, S, E)$ with $|K| = |S| = n$ (see Figure 4(a)), we associate a caterpillar tree T as in Figure 4(b). The leaves a_i , corresponding to the vertices k_i of K , are connected to the spine with edges of weight 1 and the leaves b_i , corresponding to vertices $s_i \in S$, with edges of weight n . It is clear that $G = LPG(T, n+1)$. Indeed, for any two a_i, a_j it holds that $3 \leq d_T(a_i, a_j) \leq n+1$, for any two b_i, b_j we have $d_T(b_i, b_j) \geq 2n+1$ and for any a_i, b_i we have $d_T(a_i, b_i) = n+1$ (hence the edge $(k_i, s_i) \in E$) and for any a_i, b_j with $i \neq j$ we have $d_T(a_i, b_j) \geq n+2$ (hence the edge $(k_i, s_j) \notin E$). \square

Note that this representation is not unique. Indeed, one can easily check that the binary tree T in Figure 4(c) also is a pairwise compatibility tree of a split matching graph when $d_{max} = 4$.

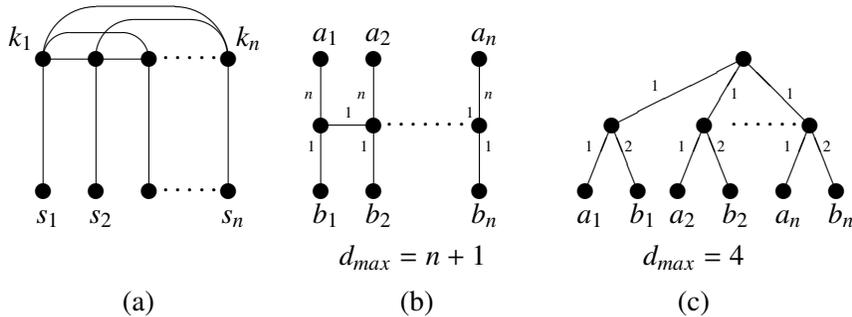


Figure 4: (a) A split matching graph (b) A pairwise compatibility caterpillar tree for a split matching graph. (c) A pairwise compatibility tree for a split matching graph.

It is worth to note that the previous construction can be generalized to split matching sequences as shown by the next theorem:

Theorem 3. *Let H be a split matching sequence, then $H \in LPG$. A tree T , an edge weight function w and a value d_{max} associated to H can be found in polynomial time.*

Proof. Let $H = G_1 \circ \dots \circ G_t$ be a split matching sequence. For each graph G_i we define a tree T_i as shown in Figure 5(a) (where the leaves a_i (b_i) may not possibly appear if G_i is a stable (clique) graph). It is clear that $G_i = LPG(T_i, d_{max})$ where d_{max} is a value to be defined later, but surely greater than or equal to $2(i+1)$. Indeed, let a_1, \dots, a_n be the leaves of T_i corresponding to vertices of K_i and let b_1, \dots, b_n be those corresponding to vertices of S_i . For any two leaves a_r, a_s it holds that $d_{T_i}(a_r, a_s) = 2 + 2i \leq d_{max}$ and for any two b_s, b_r we have $d_{T_i}(b_r, b_s) = 2d_{max} - 2i \geq d_{max} + 2i + 2 - 2i > d_{max}$. Finally, for any two leaves a_s, b_s that correspond to an edge of the matching their distance is $d_{max} - 2i + 1 \leq d_{max}$ and for any two leaves corresponding to a non edge a_r, b_s their distance is $d_{max} + 1$.

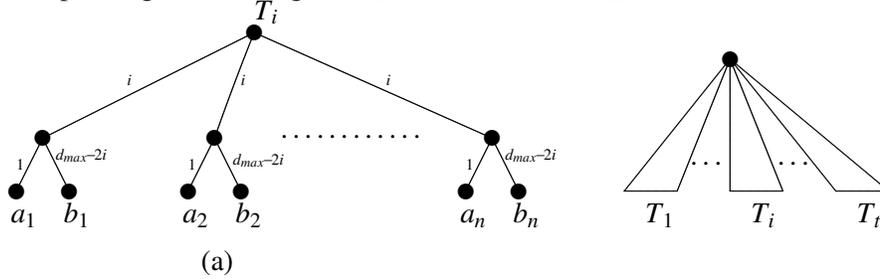


Figure 5: (a) The pairwise compatibility tree for the split matching graph G_i . (b) The pairwise compatibility tree for the split matching sequence H .

In order to prove that $H \in LPG$, we define a new tree T starting from the trees T_1, \dots, T_t , simply by contracting all their roots to a single vertex as shown in Figure 5(b). We claim that $H = LPG(T, d_{max})$ where we set $d_{max} = 2(t+1)$. In order to prove it, consider two graphs G_i and G_j with $i < j$. Let a, a', b and b' be four distinct leaves corresponding to vertices in K_i, K_j, S_i and S_j respectively. Observe that the vertices in K_i are connected to all the other vertices in $K_j \cup S_j$ as the distances in T are $d_T(a, a') = 1 + i + j + 1 \leq 2(j+1) \leq d_{max}$ and $d_T(a, b') = 1 + i + j + d_{max} - 2j = d_{max} + (i - j + 1) \leq d_{max}$ (as $j \geq i + 1$). Finally, any vertex in S_i is not connected to any vertex K_j and to any vertex S_j as in these cases the distances are $d_T(b, a') = d_{max} - 2i + i + j + 1 > d_{max}$ (as $j \geq i + 1$) and $d_T(b, b') = d_{max} - 2i + i + j + d_{max} - 2j \geq 2d_{max} - 2j > d_{max}$. \square

Analogously, we can show that the set $mLPG \setminus LPG$ is not empty.

Theorem 4. Let G be a split antimatching graph, then $G \notin \text{LPG}$ and $G \in \text{mLPG}$. A tree T , an edge weight function w and a value d_{\min} associated to G can be found in polynomial time.

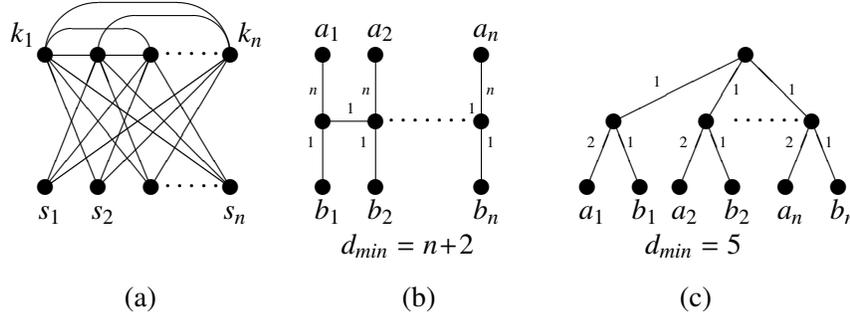


Figure 6: (a) A split antimatching graph. (b) A pairwise compatibility caterpillar tree for a split antimatching graph. (c) A pairwise compatibility tree for a split antimatching graph.

We omit the proof of this theorem, as it is almost identical to the proof we supply for Theorem 2. The edge weighted tree T associated to a split antimatching graph is one from the ones depicted in Figure 6.

Again, observe that using arguments similar to the proof of Theorem 3 we can generalize the last result to split antimatching sequences. The edge weighted tree T associated to a split antimatching sequence is the one depicted in Fig. 7).

Theorem 5. Let H be a split antimatching sequence, then $H \in \text{mPCG}$. A tree T , an edge weight function w and a value d_{\min} associated to H can be found in polynomial time.

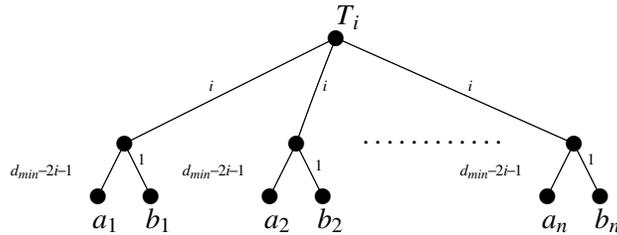


Figure 7: The pairwise compatibility tree for the split antimatching graph G_i .

operation o	PCG	LPG	mLPG
adding an isolated vertex	PCG (Thm. 6)	LPG (Thm. 7)	mLPG (Thm. 9)
adding an universal vertex	?	LPG (Thm. 8)	mLPG (Thm. 9)
adding a pendant vertex	PCG (Thm. 10)	LPG (Thm. 10)	PCG (Thm. 10, Prop. 3)
adding a false twin	PCG (Thm. 11)	PCG (Thm. 11)	mLPG (Thm. 11)
adding a true twin	?	LPG (Thm. 13)	?
graph complement	?	mLPG (Prop. 2)	LPG (Prop. 2)
disjoint union	PCG (Thm. 16)	LPG (Thm. 16)	PCG (Thm. 16)

Figure 8: Summary of the results obtained applying an operation o to a class $D \in \{\text{PCG}, \text{LPG}, \text{mLPG}\}$.

5. Graph Operations

In this section we study the effect of some classical graph operations on the PCG property. In particular, when a graph operation is performed on one or more PCGs (or LPGs or mLPGs), we investigate whether the resulting graph still belongs to the class of PCGs (or LPGs or mLPGs). We study the operations listed in Section 2, i.e. adding an isolated vertex or an universal vertex, adding a pendant vertex, adding a false or a true twin, the graph complement and the disjoint union. These closure results are summarized in the table in Figure 8. It is worth to note that some classes are not closed under some specific graph operations. In this case, we are interested to identify the class containing the result. However, as shown in the table, four cases remain unsolved.

5.1. Adding an universal or an isolated vertex

Here we consider the operations of adding either an universal or an isolated vertex. We prove that the classes of LPGs and mLPGs are closed under both operations, whereas for the PCG class we are able to prove the closure only under the operation of adding an isolated vertex.

Theorem 6. *Given a graph $G = \text{PCG}(T, w, d_{\min}, d_{\max})$, any graph G^{+i} obtained from G by adding an isolated vertex is still a PCG. It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and two values \tilde{d}_{\min} and \tilde{d}_{\max} such that $G^{+i} = \text{PCG}(\tilde{T}, \tilde{w}, \tilde{d}_{\min}, \tilde{d}_{\max})$.*

Proof. Given $G = \text{PCG}(T, w, d_{\min}, d_{\max})$ let u be the isolated vertex to be added. Consider an internal vertex p of T . We obtain \tilde{T} by adding to T a new leaf

l_u , corresponding to u and the edge (l_u, p) . The weight function \tilde{w} is defined on \tilde{T} as w , and $\tilde{w}((l_u, p))$ is set to d_{max} . Clearly, $G^{+i} = PCG(\tilde{T}, \tilde{w}, d_{min}, d_{max})$. \square

Following to the previous constructive proof it is easy to convince oneself that the next theorem holds, too.

Theorem 7. *Given a graph $G = LPG(T, w, d_{max})$, any graph G^{+i} obtained from G by adding a new isolated vertex is still a LPG. It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and a value \tilde{d}_{max} such that $G^{+i} = LPG(\tilde{T}, \tilde{w}, \tilde{d}_{max})$.*

Concerning the addition of an universal vertex we prove the following results:

Theorem 8. *Given a graph $G = LPG(T, w, d_{max})$, any graph G^{+u} obtained from G by adding a new universal vertex is still a LPG. It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and a value \tilde{d}_{max} such that $G^{+u} = LPG(\tilde{T}, \tilde{w}, \tilde{d}_{max})$.*

Proof. Consider $G = LPG(T, w, d_{max})$ and let u be the universal vertex to be added. Choose any internal vertex p in T . The new tree \tilde{T} is built from T by adding a new leaf l_u corresponding to u and a new edge (p, l_u) . Consider the weighted paths from p to any other leaf of T . Let l_v be a leaf for which $d_{T,w}(l_v, p)$ attains maximum value. We distinguish two cases according to the fact that $d_{T,w}(l_v, p) \geq d_{max}$ or not.

If $d_{T,w}(l_v, p) < d_{max}$, then it is enough to define the edge weight function \tilde{w} as identical to w on the edges of T and $\tilde{w}((p, l_u)) = d_{max} - d_{T,w}(l_v, p)$; the value of \tilde{d}_{max} is set equal to d_{max} .

Otherwise, let $d_{T,w}(l_v, p) > d_{max}$. Then, it is possible to modify the weight function w adding a value h to the weights of the edges incident to all the leaves; in order to keep the feasibility, d_{max} must be increased by $2h$. So we can define w and d_{max} according to the previous modifications. It is easy to see that there exists an opportune value of h , such that $d_{T,w}(l_v, p) < d_{max}$; in this way, we fall in the previous case, and the statement follows. \square

Now, observe that adding an isolated vertex to a graph G is equivalent to complementing the graph obtained by adding an universal vertex to the complement of G . Using this simple property we are able to prove the following results:

Theorem 9. *Given a graph $G = mLPG(T, w, d_{min})$, any graph G^{+u} (G^{+i}) obtained from G by adding a new universal (isolated) vertex is still a mLPG. It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and a value \tilde{d}_{min} such that $G^{+u} = mLPG(\tilde{T}, \tilde{w}, \tilde{d}_{min})$ ($G^{+i} = mLPG(\tilde{T}, \tilde{w}, \tilde{d}_{min})$).*

Proof. Let G be an mLPG. According to Proposition 2, its complement G^C is a LPG. We can follow the constructive proof of the Theorem 7 and add to G^C an isolated vertex v , so that $G^C \cup \{v\}$ is still a LPG. The thesis follows observing that

the complement of $G^C \cup \{v\}$ is G^{+u} , obtained from G to which the universal vertex v is added. The same argument can be used in proving the closure of mLPG under the addition of an isolated vertex. \square

It is to notice that the results of this subsection show that threshold graphs are in both mLPG and LPG. Finally, we want to remark that the construction provided in the proof of Theorem 8 does not necessarily work on PCGs, because there is no way to handle at the same time d_{max} and d_{min} . It remains an open problem to determine whether the PCG class is closed under the operation of adding an universal vertex.

5.2. Adding a pendant vertex

Here we prove that if we add a new pendant vertex of a given PCG, the new graph is still a PCG and we provide the corresponding tree. Thus, the PCG class is closed under this operation.

Theorem 10. *Given $G = PCG(T, w, d_{min}, d_{max})$ ($LPG(T, w, d_{max})$), any graph G^{+p} obtained by G adding a pendant vertex is still a PCG (LPG). It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and two values \tilde{d}_{min} and \tilde{d}_{max} such that $G^{+p} = PCG(\tilde{T}, \tilde{w}, \tilde{d}_{min}, \tilde{d}_{max})$ ($G^{+p} = LPG(\tilde{T}, \tilde{w}, \tilde{d}_{max})$).*

Proof. Suppose we are adding a new vertex v to the neighborhood of an old vertex u belonging to $G = PCG(T, w, d_{min}, d_{max})$. Consider the tree T and, in particular its leaf l_u associated to the vertex u . Let p be the only vertex adjacent to l_u in T and call h the weight $w((p, l_u))$ associated to the edge (p, l_u) .

Consider any $0 < \epsilon < \min\{h/2, d_{max}\}$. We can set $\tilde{d}_{min} = d_{min}$ and $\tilde{d}_{max} = d_{max}$ and easily construct \tilde{T} as follows: eliminate from T the edge (p, l_u) , add vertices l_v and p' and edges (p, p') , (p', l_u) and (p', l_v) . Assign to these edges the following weights: $\tilde{w}((p', l_u)) = \epsilon$, $\tilde{w}((p', l_v)) = d_{max} - \epsilon$ and $\tilde{w}((p, p')) = h - \epsilon$, while on all the other edges \tilde{w} is equal to w . It is easy to see that G^{+p} is PCG for this new tree. This proof holds even if $d_{min} = 0$ and hence G is a LPG, proving that G^{+p} is a LPG, too. \square

The constructive method exploited in the previous proof cannot be applied to the mLPG class. Indeed, in this case, we can prove that the mLPG class is not closed under the operation of adding a pendant vertex. To this purpose consider the graph G in Figure 9(a). It is not difficult to see that $G = mLPG(T, w, 7)$, where T and w are defined in Figure 9(b). Now let G^{+p} be the graph obtained by adding a new vertex s_3 and connecting it with the vertex k_3 in G (see Figure 9(c)). From Lemma 6 the resulting graph cannot belong to the mLPG class. Hence the next proposition holds.

Proposition 3. *The mLPG class is not closed under the operation of adding a new pendant vertex.*

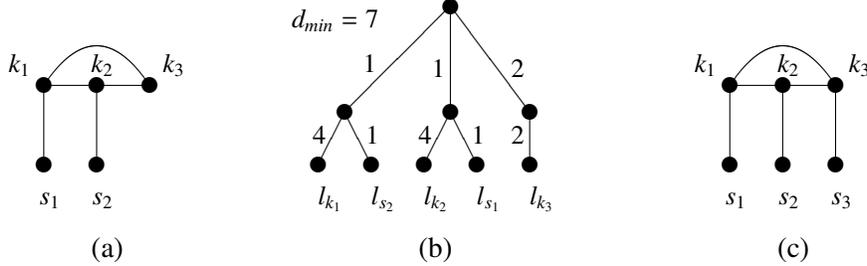


Figure 9: (a) An mLPG G and (b) its corresponding pairwise compatibility tree T . (c) The graph G^{+p} .

5.3. Adding a (false or true) twin

In this subsection we prove that the PCG class is closed under the addition of a false twin. For what concerns adding a true twin, we are able to provide a tree for the increased graph in the case in which G is either a LPG or an mLPG, but not when G is a PCG, and this remains an interesting open problem.

Theorem 11. *Given $G = PCG(T, w, d_{min}, d_{max})(mLPG(T, w, d_{min}))$, any graph G^{+ft} obtained from G by adding a false twin to any of its vertices is still a PCG (mLPG). It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and two values \tilde{d}_{min} and \tilde{d}_{max} such that $G^{+ft} = PCG(\tilde{T}, \tilde{w}, \tilde{d}_{min}, \tilde{d}_{max})$ ($G^{+ft} = mLPG(\tilde{T}, \tilde{w}, \tilde{d}_{min})$).*

Proof. Let us consider G , its associated tree T and the corresponding edge-weighted function w . Let v be any of the vertices in G to which we want to add a false twin v' . So G^{+ft} has vertex set $V \cup \{v'\}$ and edge set $E \cup \{(u, v') | u \in N(v)\}$. Consider now the leaf l_v of T corresponding to the vertex v ; call p the father of l_v in T and let h be the value of $w((p, l_v))$. The values of \tilde{d}_{min} and \tilde{d}_{max} are set equal to d_{min} and d_{max} , respectively.

We distinguish two cases according to the value h .

- If $h < d_{min}/2$ or $h > d_{max}/2$ then we construct the tree \tilde{T} by simply adding a new leaf $l_{v'}$ to p . The weights on the edges of \tilde{T} are set equal to the weights on the edges of T , so we only need to define $\tilde{w}((p, l_{v'}))$ equal to h in order to guarantee that edge (v, v') is not present in G^{+ft} while all the edges in $(u, v') | u \in N(v)$ are in G^{+ft} and this concludes the proof.
- If $d_{min}/2 \leq h \leq d_{max}/2$ we eliminate the edge (p, l_v) from T and add vertices p' and $l_{v'}$ and edges (p, p') , (p', l_v) and $(p', l_{v'})$. We define the weight

function \tilde{w} on these edges in the following way: $\tilde{w}((p', l_v)) = \tilde{w}((p', l_{v'})) = d_{\min}/2 - \epsilon$ and $\tilde{w}((p, p')) = h - (d_{\min}/2 - \epsilon)$ for some positive constant ϵ guaranteeing that all these weights are positive. On all the other edges \tilde{w} is defined equal to w . This assignment guarantees that the distances in \tilde{T} between $l_v, l_{v'}$ and the leaves associated to v 's neighbors remain equal to the distances in T between v and the leaves associated to its neighbors. Nevertheless, the distance between l_v and $l_{v'}$ in \tilde{T} is $d_{\min} - 2\epsilon$ and hence v and v' are not connected in G^{+ft} .

Observe that the previous proof can be exploited to show also that the mLPG class is closed under the operation of adding a false twin. \square

However, the same argument cannot be applied to the LPG class as in this case, when $h \leq d_{\max}/2$, it is not possible to locally transform the tree so that v and v' are not connected in G^{+ft} due to the bound concerning only d_{\min} . Nevertheless, as LPG is in PCG, by adding a false twin to a LPG, due to Theorem 11, we clearly obtain a PCG.

Theorem 12. *Given $G = \text{LPG}(T, w, d_{\max})$, the graph G^{+tt} obtained from G by adding a true twin to any of its vertices is still a LPG. It is polynomial to find a tree \tilde{T} , an edge weight function \tilde{w} and a value \tilde{d}_{\max} such that $G^{+tt} = \text{LPG}(\tilde{T}, \tilde{w}, \tilde{d}_{\max})$.*

Proof. This proof is very similar to the previous one. The only difference is that we have to modify T so that the newly added vertex v' is now connected even to its twin v .

Let l_v be the leaf of T associated to vertex v , let p be its father, and h be $w((l_v, p))$. If $h \leq d_{\max}/2$, then it is enough to add a new child $l_{v'}$ to p and the weight of this new edge is h . If, on the contrary, $h > d_{\max}/2$, then we remove from T the edge (p, l_v) and add vertices p' and $l_{v'}$ and edges (p, p') , (p', l_v) and $(p', l_{v'})$. To these edges we assign the following weights: $\tilde{w}((p', l_v)) = \tilde{w}((p', l_{v'})) = d_{\max}/2$ and $\tilde{w}((p, p')) = h - d_{\max}/2$. It is easy to see that these values guarantee that G^{+tt} is leaf power of the transformed tree for $\tilde{d}_{\max} = d_{\max}$. \square

Observe that it is not possible to state a similar result for the more general PCG class. The reason is that, if $h < d_{\min}/2$, there is no way to locally transform the tree so that v and v' are connected in the graph. It remains an open problem to understand if adding a true twin in a PCG results in a PCG; if this is possible, it would probably require to completely modify the tree.

We conclude observing that the results of this subsection together with the results of Section 5.1 and Section 5.2, respectively, imply that distance-hereditary bipartite graphs are PCGs.

5.4. The complement

We have already highlighted in Proposition 2 that the complement of a LPG is an mLPG and vice-versa, moreover in Section 4 it is proved that these classes do not coincide. So we may affirm that LPGs and mLPGs are not closed under the complement operation, but they are transformed one into the other one. We now study the more general class of PCGs. To this purpose, let $G = (V, E) = PCG(T, w, d_{min}, d_{max})$ and let Λ be the set of the edges of T , $\Lambda = \{e_1, \dots, e_{|\Lambda|}\}$. Without loss of generality we introduce any ordering in the set of $N = \binom{n}{2}$ paths connecting any pair of leaves and let $\Gamma = \{\gamma_1, \dots, \gamma_N\}$ be this ordered set of paths. Let $\underline{w} = (w_1, \dots, w_{|\Lambda|})$ and $\underline{d} = (d_1, \dots, d_N)$ be two vectors, where w_i is the weight associated to edge e_i and d_i is the distance between the leaves connected by path γ_i . The *edge-path incidence matrix* is an $N \times |\Lambda|$ matrix M such that M_{ij} is equal to 1 if $e_j \in \gamma_i$ and to 0 otherwise.

As an example, consider the tree shown in Fig. 10.a., which is the weighted tree T realizing a 5-cycle as a $PCG(T, w, d_{min}, d_{max})$, with $\underline{w} = (2, 12, 1, 12, 2, 1, 1)$, $d_{min} = 6$ and $d_{max} = 14$ (see [16] Theorem 3). The path set results: $\Gamma = \{\gamma_1 = P_{ab}, \gamma_2 = P_{ac}, \gamma_3 = P_{ad}, \gamma_4 = P_{ae}, \gamma_5 = P_{bc}, \gamma_6 = P_{bd}, \gamma_7 = P_{be}, \gamma_8 = P_{cd}, \gamma_9 = P_{ce}, \gamma_{10} = P_{de}\}$. Hence, the distance vector is $\underline{d} = (14, 4, 16, 6, 14, 26, 14, 4, 14)$. In Fig. 10.b. the corresponding matrix is represented.

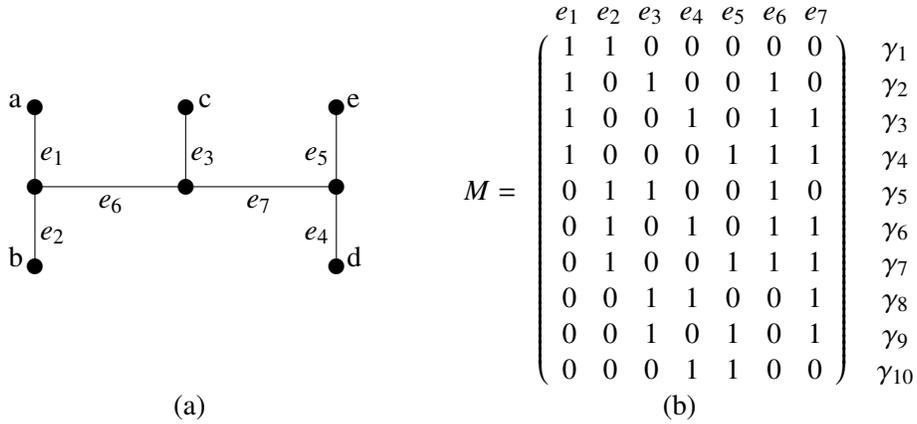


Figure 10: (a) the tree T ; (b) its corresponding edge-path incidence matrix.

The introduction of matrix M leads us to describe the connection between weights and distances by the formula: $M\underline{w}^t = \underline{d}^t$. Observe that the vertices of V corresponding to the leaves of T connected by path γ_i are adjacent in G if and only if

$$d_{min} \leq d_i = (M\underline{w}^t)_i \leq d_{max}.$$

Now we wonder under which conditions the complement of a PCG is still a PCG with the same tree, possibly varying the weights, in other words: if $G = PCG(T, w, d_{min}, d_{max})$, under which conditions is it possible to choose a new weight function \tilde{w} such that $G^C = PCG(T, \tilde{w}, \tilde{d}_{min}, \tilde{d}_{max})$?

This problem is equivalent to finding a positive solution \underline{w}' of the linear system:

$$M\underline{w}' = \underline{\tilde{d}} \quad (4)$$

subject to the following requirements on the positive distances vector $\underline{\tilde{d}}$:

$$\exists \tilde{I} = [\tilde{d}_{min}, \tilde{d}_{max}] \subseteq (0, +\infty) \quad \text{such that} \quad \tilde{d}_i \in \tilde{I} \Leftrightarrow d_i \notin I = [d_{min}, d_{max}] \dots \quad (5)$$

This means that the question is equivalent to looking for a distance vector $\underline{\tilde{d}} \in \mathbb{R}^N$ which is a linear combination (with positive coefficients) of the columns of matrix M and which verifies the relations in (5). These arguments prove the following result:

Theorem 13. *Given $G = PCG(T, w, d_{min}, d_{max})$ then $G^C = PCG(T, \tilde{w}, \tilde{d}_{min}, \tilde{d}_{max})$ for some $\tilde{d}_{min} \leq \tilde{d}_{max}$ if and only if it exists a solution of (4)-(5).*

As an example, we apply this theorem to the construction given in [16] where it is proven that cycles are PCGs. Solving the system generated by the edge-path incidence matrix corresponding to an odd cycle C_{2n+1} , it is possible to show that its complement C_{2n+1}^C is in PCG with respect to the same tree T , $\tilde{d}_{min} = 3.5$, $\tilde{d}_{max} = n + 4.5$ and weight function $\tilde{w} = (\tilde{w}_1, \dots, \tilde{w}_{|A|})$ defined as follows:

$$\tilde{w}_1 = \tilde{w}_{2n+1} = 2, \quad \tilde{w}_i = 1 \text{ for any } i \neq 1, 2n + 1,$$

Similarly, the complement G_{2n+2}^C of any even cycle is a PCG with respect to the same tree T and bounds $\tilde{d}_{min} = 3.5$, $\tilde{d}_{max} = 2 + 3n/2$ and weight function \tilde{w} such that

$$\tilde{w}_1 = \tilde{w}_{2n+1} = 2, \quad \tilde{w}_{2n+2} = n, \quad \tilde{w}_i = 1 \text{ for any } i \neq 1, 2n + 1, 2n + 2,$$

We have hence proven the following:

Proposition 4. *The complement of any cycle C is a PCG.*

The pairwise compatibility trees of the complement of an odd and even cycle are depicted in Fig. 11.

We conclude by observing that the previous result does not imply that if a solution of (4)-(5) does not exist, then G^C is not a PCG. Thus, it remains an open problem to establish the graphs in the PCG class for which their complement is still in PCG.

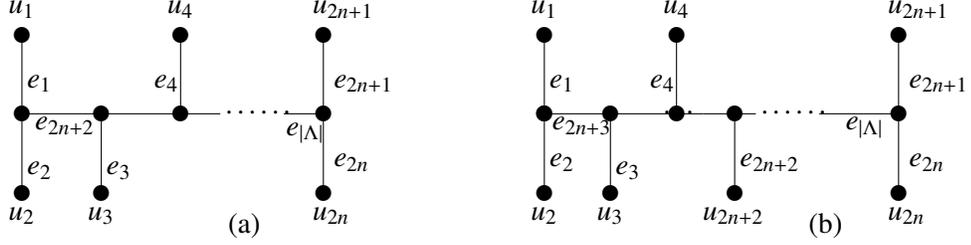


Figure 11: The pairwise compatibility tree of (a) an odd cycle; (b) even cycle.

5.5. The disjoint union

In this last subsection we prove that the disjoint union of two PCGs G_1 and G_2 is still a PCG. Notice that adding an isolated vertex is a particular case of disjoint union of two graphs when one of the graphs consists of a single vertex so here we will generalize the techniques used in Theorem 6 and Theorem 7.

Theorem 14. *Given two graphs $G_1 = PCG(T_1, w_1, d_{1min}, d_{1max})$ and $G_2 = PCG(T_2, w_2, d_{2min}, d_{2max})$, their disjoint union $G = G_1 \cup G_2$ is still a PCG. It is polynomial to determine a tree \tilde{T} , an edge weight function \tilde{w} and two values \tilde{d}_{min} and \tilde{d}_{max} such that $G = PCG(\tilde{T}, \tilde{w}, \tilde{d}_{min}, \tilde{d}_{max})$.*

Proof. Considering Lemma 4, we can modify the weight functions such that for some integer $m \geq \max\{d_{1min}, d_{2min}\}$ and two integers M_1, M_2 it holds $G_1 = PCG(T_1, \hat{w}_1, m, M_1)$ and $G_2 = (T_2, \hat{w}_2, m, M_2)$. Moreover, Lemma 2 guarantees us that it is not restrictive to assume $m \neq M_1$ and $m \neq M_2$. Finally, let $M = \min\{M_1, M_2\}$. Observe that $m < M$, then using again Lemma 4 we can find two edge weight functions such that $G_1 = PCG(T_1, w'_1, m, M)$ and $G_2 = (T_2, w'_2, m, M)$. Let $G = G_1 \cup G_2$. Define a new tree \tilde{T} with $V(\tilde{T}) = V(T_1) \cup V(T_2)$ and $E(\tilde{T}) = E(T_1) \cup E(T_2) \cup \{v_1, v_2\}$ where v_1 and v_2 are non-leaf vertices in T_1 and T_2 , respectively. Define the weight of the edge (v_1, v_2) equal to M , while keeping the weights of all the other edges unchanged. It is clear that $G = PCG(\tilde{T}, \tilde{w}, m, M)$. \square

Using the same argument it is easy to prove that a similar result holds for LPG graphs (hence LPG is closed under this operation), while there is no weight to assign to edge (v_1, v_2) able to guarantee that no edge between vertices of G_1 and G_2 is created in $G_1 \cup G_2$ when G_1 and G_2 are mLPGs.

We conclude this section observing that these results imply that quasi-threshold graphs are LPGs.

6. Conclusions and Open Problems

In this paper, we focus on the general pairwise compatibility tree construction problem by approaching a number of subproblems.

Namely, we first analyze the relations between the three classes of PCGs, LPGs and mLPGs. In particular, we show that the union of LPG and mLPG classes does not coincide with the whole class of PCGs, that neither of the classes of LPGs and mLPGs is contained in the other and that their intersection is not empty as threshold graphs belong to both the classes of mLPGs and LPGs.

Finally, we investigate whether the graph resulting when a graph operation is performed on one or more PCGs (or LPGs or mLPGs) still belongs to the class of PCGs (or LPGs or mLPGs). The considered graph operations consist in adding a new special vertex (either isolated or universal vertex, pendant vertex, either true or false twin), in complementing the old graph or in performing the disjoint union of two graphs. As a side effect, we prove that bipartite distance-hereditary graphs are PCGs and quasi-threshold graphs are LPGs.

A number of open problems arise from this work.

First of all, since the pairwise compatibility problem has been investigated only for a few classes of graphs, it remains an open problem to analyze other graph classes for which this property holds. This is even more interesting because it is known [10] that the clique problem can be solved in polynomial time for the class of compatibility graphs if we are able to construct in polynomial time the pairwise compatibility tree.

Another issue consists in completing the table in Figure 8. Considering the results presented in the table, it is to underline that only in the case involving the operation of adding a new pendant vertex, we were able to prove that the mLPG class is not closed. Thus, it remains an open problem even to determine those operations for which a class is closed.

We have proved that the complement of a PCG is a PCG, too, only if some precise conditions are verified. We conjecture that it is not true in general that the complement of a PCG is always a PCG. Thus, it would be interesting to prove (or disprove) this conjecture.

Finally, we are interested in determining the structure of graphs that are PCGs of a star, in the light of the fact that both LCG-tree and mLPG-tree associated to a threshold graph are stars.

References

- [1] Brandstädt, A.: On Leaf Powers. Technical report, University of Rostock, (2010).

- [2] Brandstädt, A., Le, V. B.: Structure and linear time recognition of 3-leaf powers. *Inform. Process. Lett.* 98 (2006) 133–138.
- [3] Brandstädt, A., Le, V. B., Spinrad, J.: Graph classes: a survey, *SIAM Monographs on discrete mathematics and applications*, (1999).
- [4] Brandstädt, A., Le, V. B., Sritharan, R.: Structure and linear time recognition of 4-leaf powers. *ACM Trans. Algorithms* 5(1) (2008) 11:1–11:22.
- [5] Brandstädt, A., Hundt, C., Mancini, F., Wagner, P.: Rooted directed path graphs are leaf powers. *Discrete Math.* 310(4) (2010) 897–910.
- [6] Calamoneri, T., Frascaria, D., Sinaimeri, B.: All graphs with at most seven vertices are Pairwise Compatibility Graphs. *Computer J.*, 2012, in press. doi:10.1093/comjnl/bxs087.
- [7] Calamoneri, T., Petreschi, R., Sinaimeri, B.: On relaxing the constraints in pairwise compatibility graphs, In: Md. S. Rahman and S.-i. Nakano (Eds.), WALCOM 2012, LNCS vol. 7157, 124–135, Springer, Berlin (2012).
- [8] Fellows, M., Meister, D., Rosamond, F., Sritharan, R., Telle, J.A.: Leaf Powers and Their Properties: Using the Trees. In: Proceedings of ISAAC 2008 LNCS 5369, Springer-Verlag (2008) 402–413.
- [9] Garey, M. R., Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York 1979.
- [10] Kearney, P. E., Munro, J. I., Phillips, D.: Efficient generation of uniform samples from phylogenetic trees. In: Benson, G., Page, R.D.M. (eds.) WABI. LNCS 2812, Springer, Berlin (2003) 177–189.
- [11] Kennedy, W., Lin, G., Yan, G.: Strictly chordal graphs are leaf powers. *J. of Discrete Algorithms* 4 (2006) 511–525.
- [12] Nishimura, N., Ragde, P., Thilikos, D. M.: On graph powers for leaf-labeled trees. *J. Algorithms* 42 (2002) 69–108.
- [13] Phillips, D.: Uniform sampling from phylogenetics trees. Masters Thesis, University of Waterloo (2002).
- [14] Salma, S. A., Rahman, Md. S.: Ladder Graphs Are Pairwise Compatibility Graphs. Abstract at Asian Association for Algorithms and Computation (AAAC) 2011 Hsinchu, Taiwan.
- [15] Yanhaona, M. N., Bayzid, Md. S., Rahman, Md. S.: Discovering Pairwise compatibility graphs. *Discrete Mathematics, Algorithms and Applications (DMAA)* 2(4) (2010) 607–623.

- [16] Yanhaona, M. N., Hossain, K. S. M. T., Rahman, M. S.: Pairwise compatibility graphs. *J. Appl. Math. Comput.* 30 (2009) 479–503.
- [17] J. Felsenstein: Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology* 27, (1978) 401-410.