Università di Pisa

Dipartimento di Informatica

Technical Report: TR-13-09

Pairwise Compatibility Graphs of Caterpillars

Tiziana Calamoneri
Dipartimento di Informatica, Sapienza Università di Roma
calamo@di.uniroma1.it

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
frangio@di.unipi.it

Blerina Sinaimeri
Dipartimento di Informatica, Sapienza Università di Roma
sinaimeri@di.uniroma1.it

May 19, 2013

ADDRESS: Largo B. Pontecorvo 3, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

Pairwise Compatibility Graphs of Caterpillars

Tiziana Calamoneri
Dipartimento di Informatica, Sapienza Università di Roma
calamo@di.uniroma1.it

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
frangio@di.unipi.it

Blerina Sinaimeri
Dipartimento di Informatica, Sapienza Università di Roma
sinaimeri@di.uniroma1.it

May 19, 2013

Abstract

A graph $G = (V, E)$ is called a pairwise compatibility graph (PCG) if there exists an edge-weighted tree T and two non-negative real numbers d_{min} and 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 focus our attention on PCGs for which the witness tree is a caterpillar. We first give some properties of graphs that are PCGs of a caterpillar. We formulate this problem as an integer linear programming problem and we exploit this formulation to show that for the wheels on n vertices W_n, $n = 7, \ldots, 11$, the witness tree cannot be a caterpillar. Related to this result, we conjecture that no wheel is PCG of a caterpillar. Finally, we state a more general result proving that any pairwise compatibility graph admits a full binary tree as witness tree T.

Keywords: *Pairwise Comparability Graphs, Caterpillar, Centipede, Wheel.*

1 Introduction

A graph $G = (V, E)$ is a *pairwise compatibility graph* (PCG) if there exists a tree T, an edge-weight function w that assigns positive values to the edges of T and two non-negative real numbers d_{min} and d_{max}, with $d_{min} \leq d_{max}$, such that there is a bijection between the vertices of G and the leaves of T (so 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, $G = PCG(T, w, d_{min}, d_{max})$.

It is clear that 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 a tree T, an edge-weight function w and 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*.

The concept of pairwise compatibility was introduced in [1] in a computational biology context and the weight function w has positive values, as it represents a not null distance. Moreover, this problem has many other applications as it is part of a class of problems that are motivated by issues in graph powers, intersection graphs, tree representations of similarity relations and evolutionary processes [2].

There are several known specific graph classes that have been proved to be pairwise compatibility graphs, e.g., all the graphs with at most 7 vertices [3, 4], cliques and disjoint union of cliques [5], chordless cycles and single chord cycles [6], some particular subclasses of bipartite graphs [7], some particular subclasses of split matrogenic graphs [8], triangle-free outerplanar 3-graphs [9], Dilworth 2 graphs [10]. From the other side, it is known that not all graphs are PCGs [7, 11]. Furthermore a lot of work has been done concerning some particular subclasses of PCGs as leaf power graphs [5], exact leaf power graphs [2] and min-leaf power graphs [8].

A *caterpillar* Γ_n is an n-leaf tree for which any leaf is at a distance exactly one from a central path called *spine*. A *centipede* Π_n is an n-leaf caterpillar whose edges incident to the leaves induce a perfect matching. As an example, Π_7 is depicted in Fig. 1.

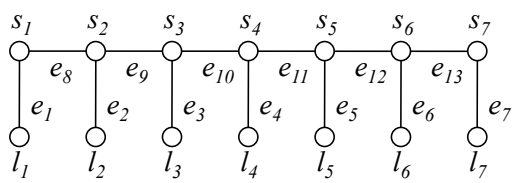


Figure 1: A 7-leaf centipede Π_7.

Caterpillars are interesting trees in the context of PCGs, as in most of the cases, the pairwise compatibility tree construction problem admits as solution a tree that is in fact a caterpillar. For this reason, we focus on this special kind of tree, providing the following results:

- We study the properties of the graphs that are PCGs of a caterpillar. First we consider the special case in which the edge-weight function assigns weight 1 to each edge of the caterpillar, and we provide a characterization of these graphs. Observe that this restriction is natural as in many papers (e.g. see [5, 1]) the tree is not weighted and the distance is defined as the number of edges on the (unique) path connecting two leaves.

  Then we consider the general case and exploit sufficient conditions concerning d_{max} and d_{min} to guarantee that the PCGs are triangle free.

- We formulate the pairwise compatibility tree construction problem as an integer linear programming problem (ILP). We exploit this formulation to show that the witness tree of the

wheels W_n, $n = 7, \ldots, 11$ cannot be a caterpillar, where W_n is the graph formed by connecting a single vertex to all vertices of an $(n-1)$-cycle. Related to this result, we conjecture that no wheel is PCG of a caterpillar. We recall that it is anyway known that W_7 is PCG [4], while it is not known whether W_n, $n \geq 8$ is a PCG or not.

As a consequence of this latter result, caterpillars cannot generate all the PCGs, so we focus on a more general tree structure, namely full binary trees. A binary tree is said to be *full* if all its internal vertices have two children; in other words, all its internal vertices, except for its root r, have degree exactly 3; we denote a full binary tree by Λ.

Concerning PCGs, we prove that it is sufficient to focus only on full binary trees, i.e. if G is a PCG then one of its possible witness trees must be a full binary tree.

2 Preliminaries

In this section we recall some graph theory definitions and some theorems dealing with PCGs, that will be useful in the rest of the paper.

The *k-th power of a graph G*, denoted by G^k, is a graph with the same set of vertices as G and an edge between two vertices if and only if in G there is a path of length at most k between them. The n vertex simple path is denoted by P_n.

Given two graphs defined on the same vertex set $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ such that $E_2 \subseteq E_1$, their *difference* $G_1 - G_2$ is the graph defined on the node set V, whose edges are all the edges in G_1 that are not in G_2.

A *unit interval graph* is the intersection graph of a set of unit length intervals on the real line. It has one vertex for each interval in the set, and an edge between every pair of vertices corresponding to intervals that intersect. We denote by $\mathcal{I}_n$ the set of n vertex unit interval graphs.

Given a centipede Π_n, consider its plane representation in which the path representing its spine lies on a horizontal line, and the leaves lie on a parallel horizontal line, ordered in a way that does not introduce crosses (as an example, see Fig. 1). We name all vertices and edges of Π_n as follows:

- let $l_1, l_2, \ldots, l_{n-1}, l_n$ be the leaves, considered from left to right;
- let e_i, $i = 1, \ldots, n$ be the (unique) edge incident to the leaf l_i;
- let $s_1, \ldots, s_n$ be the vertices on the spine, considered from left to right; thus s_i is the parent of l_i, $i = 1, \ldots, n$;
- finally, let e_{n+i} be the edge on the spine connecting vertices s_i and s_{i+1}, $i = 1, \ldots, n-1$.

In view of the following result, it is possible to get rid of other kinds of caterpillar structures and restrict our attention to consider only centipedes.

Theorem 1 *[4] Let G be an n vertex graph, Γ_n and Π_n be an n leaf caterpillar and centipede, respectively.*

Let $G = PCG(\Gamma_n, w, d_{min}, d_{max})$. Then it is possible to choose w' and d'_{max} such that it also holds that $G = PCG(\Pi_n, w', d_{min}, d'_{max})$.

We conclude this section with some useful general properties concerning the edge-weight function of a pairwise compatibility tree.

Theorem 2 *[12] Let $G = PCG(T, w, d_{min}, d_{max})$, where d_{min}, d_{max} and the weight $w(e)$ of each edge e of T are nonnegative real numbers. Then it is possible to choose natural numbers $\hat{d}_{min}, \hat{d}_{max}$ and an edge wight function $\hat{w}$ such that for any e, $\hat{w}(e)$ is a natural number and $G = PCG(T, \hat{w}, \hat{d}_{min}, \hat{d}_{max})$.*

Theorem 3 *[4] Let $G = PCG(T, w, d_{min}, d_{max})$. It is possible to choose natural numbers $\hat{d}_{min}, \hat{d}_{max}$ and for any e in $E(T)$, $\hat{w}(e)$, such that $\min_{e \in E(T)} \hat{w}(e) = 1$ and $G = PCG(T, \hat{w}, \hat{d}_{min}, \hat{d}_{max})$.*

Due to the last results, in the rest of the paper we will assume that the weights and d_{min}, d_{max} are integers and that the smallest weight is 1.

The next section is devoted to the study of some properties of PCGs for which the witness tree is a centipede.

3 Properties of PCGs of Centipedes

There are a number of papers dealing with the attempt of characterizing the classes of PCGs derived by special trees or by special values of d_{min} and d_{max}; for instance, PCGs of a star $K_{1,n}$ are characterized in [12], PCGs of any tree with $d_{min} = d_{max} = 3, 4, 5$ are studied in [2] and PCGs of caterpillars for which $d_{min} = 0$ are considered in [13].

In this section we try to derive some properties of the PCGs of centipedes. As they seem to be very general graphs, we first consider a simplified model, i.e. we assume that $w(e) = 1$ for each edge of the tree. Observe that this restriction is natural as in many papers (e.g. see [5, 1]) the tree is not weighted and the distance is defined as the number of edges on the (unique) path connecting two leaves. Then, we slightly extend the class of weight functions we consider, and finally we give some properties when w is arbitrary.

3.1 Unit edge-weight

Observe that the problem to characterize PCGs of caterpillars has been considered in [13] in the special case in which $d_{min} = 0$ (in such a case PCGs coincide with the class of *Leaf Power Graphs* [14]), providing the following result:

Theorem 4 *[13] Let G be an n vertex connected graph and Γ_n be an n leaf caterpillar and let the edge-weight function $u(e) = 1$ for each edge e of Γ_n.*

Then the following statements are equivalent:

1. *$G = PCG(\Gamma_n, u, 0, d_{max})$;*
2. *G is a unit interval graph.*

We now characterize the class of graphs that are PCGs of a centipede with unit edge weights.

Theorem 5 *Let G be an n vertex connected graph, Π be an n leaf centipede and let the edge-weight function $u(e) = 1$ for each edge e of Π_n.*

Then the following statements are equivalent:

1. $G = PCG(\Pi_n, u, d_{min}, d_{max})$;
2. $G = P_n^{d_{max}-2} - P_n^{d_{min}-3}$ *if* $d_{min} > 3$ *and* $G = P_n^{d_{max}-2}$ *otherwise.*

Proof. Let $G = PCG(\Pi_n, u, d_{min}, d_{max})$. Observe that due to the unitary weights, the weighted distance between any two leaves in Π_n coincides with the length of the shortest (unique) path between them. Thus, two vertices are adjacent in G if and only if their corresponding leaves are connected in Π_n by a path of a length belonging to the interval $[d_{min}, d_{max}]$.

So, if we consider the vertices $v_1, \ldots, v_n$ of G lying on a line, each of them is connected to the vertices at distance $d_{min} - 2, \ldots, d_{max} - 2$ on the line. These edges can be obtained by considering the n vertex path P_n and computing its $(d_{max} - 2)$-th power; but this graph contains even edges that are not in G, and these edges are exactly those present in $P_n^{d_{min}-3}$. On the other hand, based on the same argument, it is easy to verify that if $G = P_n^{d_{max}-2} - P_n^{d_{min}-3}$ then it is a PCG of the centipede with unitary weights.

To conclude the proof, observe that values of d_{min} too small imply that this constraint has no effect because every pair of leaves is at distance greater than or equal to d_{min}. ■

Let $\mathcal{P}_n$ be the class of n vertex graphs that are PCGs of a unit edge-weight centipede. It is easy to see that $\mathcal{P}_n \cap \mathcal{I}_n \neq \emptyset$ (indeed P_n belongs to this set); moreover $\mathcal{P}_n - \mathcal{I}_n$ and $\mathcal{I}_n - \mathcal{P}_n$ are both non empty. Indeed, observe that any $PCG(\Pi_n, u, d_{min}, d_{max})$ such that $d_{max} - d_{min} \geq 3$ and $d_{min} \geq 2$ contains a $K_{1,3}$ as an induced subgraph. Hence such graphs belong to $\mathcal{P}_n - \mathcal{I}_n$ (indeed unit interval graphs are $K_{1,3}$-free). Furthermore, it is not difficult to see that the unit interval graph constituted by two copies of $K_{n/2}$ joined by an edge cannot be expressed in the form $P_n^{d_{max}-2} - P_n^{d_{min}-3}$ and thus it is in $\mathcal{I}_n - \mathcal{P}_n$.

This is not a contradiction, as Theorem 1 does not apply to caterpillars with unit edge-weight, and so the result in Theorem 4 is not a particular case of the result in Theorem 5; instead, this latter result constitutes a further puzzle-piece toward the comprehension of the PCG properties.

Exploiting the same technique used in the proof of Theorem 5, we can state the following results:

Corollary 1 *Let G be an n vertex connected graph and Π_n an n leaf centipede. Furthermore, let be given an integer value k and define the edge-weight function u_k on Π_n as follows: $u_k(e_i) = k$ and $u_k(e_{n+i}) = 1$ for each $i = 1, \ldots n$.*

Then the following statements are equivalent:

1. $G = PCG(\Pi_n, u_k, d_{min}, d_{max})$;
2. $G = P_n^{d_{max}-2k} - P_n^{d_{min}-2k-1}$ *if* $d_{min} > 2k + 1$ *and* $G = P_n^{d_{max}-2k}$ *otherwise.*

Corollary 2 *Let G be an n vertex connected graph and Π_n an n leaf centipede. Furthermore, let be given an integer value k and define the an edge-weight function $\hat{u}_k$ on Π_n as follows: $\hat{u}_k(e_i) = 1$ and $\hat{u}_k(e_{n+i}) = k$ for each $i = 1, \ldots n$.*

Then the following statements are equivalent:

1. $G = PCG(\Pi_n, \hat{u}_k, d_{min}, d_{max})$;

2. $G = P_n^{\lfloor \frac{d_{max}-2}{k} \rfloor} - P_n^{\lceil \frac{d_{min}-2}{k} \rceil - 1}$ *if* $d_{min} > 2k+1$ *and* $G = P_n^{\lfloor \frac{d_{max}-2}{k} \rfloor}$ *otherwise.*

Corollary 3 *Let G be an n vertex connected graph and Π_n an n leaf centipede. Furthermore, let be given an integer value k and let the edge-weight function $\hat{u}_k = k$ for each edge e of Π_n.*

Then the following statements are equivalent:

1. $G = PCG(\Pi_n, \hat{u}_k, d_{min}, d_{max})$;

2. $G = P_n^{\lfloor \frac{d_{max}-2k}{k} \rfloor} - P_n^{\lceil \frac{d_{min}-2k}{k} \rceil - 1}$ *if* $d_{min} > 2k+1$ *and* $G = P_n^{\lfloor \frac{d_{max}-2k}{k} \rfloor}$ *otherwise.*

3.2 General edge-weight

Let us now consider the more general case in which $G = PCG(\Pi_n, w, d_{min}, d_{max})$, for any edge-weight function w, whose values are integer numbers, and their minimum value is 1, according to Theorems 2 and 3. We exploit some conditions on w, d_{min} and d_{max} under which G presents some interesting properties.

Theorem 6 *Let $G = PCG(\Pi_n, w, d_{min}, d_{max})$ and $\max_{e \in \Pi_n} w(e) = p$. If $d_{max} < 2d_{min} - 2p$ then G is triangle free. On the contrary, if $d_{min} = d_{max} = d$, for each pair of edges (v_i, v_j) and (v_j, v_k) that are in G, if $w(e_j) = d/2$, then the edge (v_i, v_k) is in G, too.*

Proof. Let G be a graph satisfying the condition of the theorem. Suppose on the contrary that there are three vertices v_i, v_j and v_k in G that form a triangle, i.e. such that (v_i, v_j), (v_j, v_k) and (v_i, v_k) are edges. Consider their corresponding leaves l_i, l_j and l_k; without loss of generality, assume $i < j < k$. The existence of edges (v_i, v_j) and (v_j, v_k) implies that $d_{min} \leq d_{\Pi_n,w}(l_i, l_j) \leq d_{max}$ and $d_{min} \leq d_{\Pi_n,w}(l_j, l_k) \leq d_{max}$. Consider the edge (v_i, v_k). We have:

$$\begin{aligned} d_{\Pi_n,w}(l_i, l_k) &= d_{\Pi_n,w}(l_i, l_j) + d_{\Pi_n,w}(l_j, l_k) - 2w(e_j) \\ &\geq 2d_{min} - 2w(e_j). \end{aligned}$$

In order to prove the first claim, observe that this latter term is greater than or equal to $2d_{min} - 2p$. The hypothesis $2d_{min} - 2p > d_{max}$ implies a contradiction.

To prove the second claim, notice that $2d_{min} - 2w(e_j) \geq d_{min}$ if and only if $2w(e_j) \leq d_{min}$.

On the other hand, $d_{\Pi_n,w}(l_i, l_k) = w(e_i) + w(e_k) + x + y \leq 2d_{max} - 2w(e_j)$, and this latter term is upper bounded by d_{max} if and only if $2w(e_j) \geq d_{max}$. Joining together the two obtained inequalities, we have that (v_i, v_k) is surely an edge of G if $d_{max} \leq 2w(e_j) \leq d_{min}$. Since, by definition of PCG, $d_{min} \leq d_{max}$, the claim follows. ■

Given a graph G, let $v_1, \ldots, v_n$ be any ordering on the line of its vertices. We define the *fan with respect to nodes v_i, v_j and v_k*, and denote it by $F_{i,j,k}$, the set of edges (v_i, v_k) for each $i \leq k \leq j$.

Theorem 7 *Let $G = PCG(\Pi_n, w, d_{min}, d_{max})$ and let $w(e_i) \leq w(e_{i+1}) + w(e_{n+i})$, for each $i = 1, \ldots, n-1$. In other words, if $i < j < k$, $d_{\Pi_n,w}(l_i, l_j) \leq d_{\Pi_n,w}(l_i, l_k)$, for each $k \geq j$. If (v_i, v_j) and (v_i, v_k), $i < k$, are edges of G, then the whole fan $F_{i,j,k}$ belongs to G.*

Proof. First observe that, if (v_i, v_j) and (v_i, v_k) are edges of G, then $d_{min} \leq d_{\Pi_n,w}(l_i, l_j) \leq d_{max}$ and $d_{min} \leq d_{\Pi_n,w}(l_i, l_k) \leq d_{max}$. Let us now consider a node v_m such that $j < m < k$. To prove the claim we have to show that $d_{min} \leq d_{\Pi_n,w}(l_i, l_m) \leq d_{max}$. It is easy to convince oneself that:

$$\begin{aligned} d_{\Pi_n,w}(l_i, l_m) &= d_{\Pi_n,w}(l_i, l_{m-1}) - w(e_{m-1}) + w(e_{n+m-1}) + w(e_m) \\ &= d_{\Pi_n,w}(l_i, l_{m+1}) - w(e_{m+1}) - w(e_{n+m}) + w(e_m) \end{aligned}$$

From the hypothesis $w(e_m) \leq w(e_{m+1}) + w(e_{n+m})$, for each $m = 1, \ldots, n-1$ and from the previous equalities, it follows that $d_{\Pi_n,w}(l_i, l_{m-1}) \leq d_{\Pi_n,w}(l_i, l_m) \leq d_{\Pi_n,w}(l_i, l_{m+1})$. Iterating these inequalities, we have $d_{min} \leq d_{\Pi_n,w}(l_i, l_j) \leq d_{\Pi_n,w}(l_i, l_m) \leq d_{\Pi_n,w}(l_i, l_k) \leq d_{max}$, so the claim follows. ■

4 The ILP Model

In this section we propose an Integer Linear Programming (ILP) model for the pairwise compatibility tree construction problem, when the shape of the tree is given. That is, given an n vertex graph $G = (V, E)$ and an n leaf tree T, we want to determine whether there exists an assignment (bijective mapping $\sigma : V \to F$) between the vertex set V and the set F of the leaves of T, integer weights $w(a)$ for each edge $a \in A$ of T, and two integers $d_{min} \leq d_{max}$ such that $G = PCG(T, w, d_{min}, d_{max})$.

In the following, we denote by $\hat{E} = \{(i, j) \in V \times V \ : \ i < j\}$ the set of all possible edges in G and by $\tilde{F} = \{(u, v) \in F \times F \ : \ u < v\}$ the set of all pairs of leaves in T; since the shape of T is fixed, for each $(u, v) \in \tilde{F}$ we know the subset $A(u, v) \subseteq A$ defining the unique path between the leaf u and the leaf v in T. With this notation, we want to determine whether it is possible to satisfy the condition

$$(i, j) \in E \iff d_{min} \leq \textstyle\sum_{a \in A(\sigma(i),\sigma(j))} w(a) \leq d_{max}. \tag{1}$$

We will show that we can (reasonably easily, for small n) solve this problem by formulating it as an ILP and using available tools. To do that, we first introduce the classical (binary) assignment variables

$$x_{iu} = \begin{cases} 1 & \text{if } \sigma(i) = u \\ 0 & \text{otherwise} \end{cases}$$

for all n^2 pairs $(i, u) \in V \times F$, together with the $2n$ assignment constraints

$$\textstyle\sum_{i \in V} x_{iu} = 1, u \in F \qquad \text{and} \qquad \sum_{u \in F} x_{iu} = 1, i \in V. \tag{2}$$

For each $(u, v) \in \tilde{F}$ we then introduce binary variables

$$y_{uv} = \begin{cases} 1 & \text{if } (\sigma^{-1}(u), \sigma^{-1}(v)) \in E \\ 0 & \text{otherwise.} \end{cases}$$

In order to guarantee the intended semantic, for each $(u, v) \in \tilde{F}$ and $(i, j) \in \hat{E}$ we add

$$\text{the constraint } y_{uv} \geq x_{iu} + x_{jv} - 1 \qquad \text{if } (i, j) \in E \text{ and} \tag{3}$$

$$\text{the constraint } y_{uv} \leq 2 - x_{iu} - x_{jv} \qquad \text{if } (i, j) \notin E. \tag{4}$$

These do the intended job. Indeed, consider two leaves of T, $u, v \in F$ and two vertices of G, $u, v \in V$. If $i \neq \sigma^{-1}(u)$ or $j \neq \sigma^{-1}(v)$, then at least one among x_{iu} and x_{jv} is 0. If $(i, j) \in E$ then the right-hand-side of (3) is ≤ 0, while if $(i, j) \notin E$ then the right-hand-side of (4) is ≥ 1; in either case the constraint is redundant since $y_{uv} \in \{0, 1\}$. Thus, the constraint only becomes "active" for these quadruples $((u, v), (i, j))$ such that $x_{iu} = x_{jv} = 1$, i.e., $i = \sigma^{-1}(u)$ and $j = \sigma^{-1}(v)$; there, if $(i, j) \in E$ then constraint (3) forces $y_{uv} = 1$, while if $(i, j) \notin E$ then constraint (4) forces $y_{uv} = 0$.

Given these constraints, we can model the "if" part of (1). To do that we first introduce (positive) integer variables: d_{min}, d_{max}, and $w(a)$ for each $a \in A$, with obvious meaning. We must now represent by linear constraints, for each $(u, v) \in \tilde{F}$, the logical condition "if $y_{uv} = 1$, then $d_{min} \leq \sum_{a \in A(u,v)} w(a) \leq d_{max}$". The standard approach for representing this within an ILP would ask for a-priori knowledge of a "sufficiently large" value M, i.e., such that $M \geq \sum_{a \in A(u,v)} w(a)$ for all possible $(u, v) \in \tilde{F}$ and each possible feasible value of w (if any). If we had such M at our disposal, we could write the two classical "big-M constraints"

$$\begin{array}{l} \sum_{a \in A(u,v)} w(a) \leq d_{max} + M(1 - y_{uv}) \\ \sum_{a \in A(u,v)} w(a) \geq d_{min} - M(1 - y_{uv}) \quad . \end{array} \tag{5}$$

When $y_{uv} = 0$, both constraints are clearly redundant; conversely, when $y_{uv} = 1$ we precisely obtain the condition that the weight of the path $A(u, v)$ lies between d_{min} and d_{max}. Unfortunately, there is not any obvious way to find such an M a priori.

However, modern ILP solvers like the one we used, `Cplex 12.3`, allow to add to the formulation the so-called *indicator constraints*. These have the generic form

$$\text{binary variable} \;=\; \text{value} \quad \rightarrow \quad \text{linear constraint}$$

and their semantic is that the "linear constraint" must be satisfied by any feasible solution of the ILP where the "binary variable" has the prescribed "value" (either 0 or 1), while the solutions where the binary variable does not have that value can violate the constraint. Therefore, the two indicator constraints

$$\begin{array}{lll} y_{uv} = 1 & \rightarrow & \sum_{a \in A(u,v)} w(a) \leq d_{max} \\ y_{uv} = 1 & \rightarrow & \sum_{a \in A(u,v)} w(a) \geq d_{min} \end{array} \tag{6}$$

have precisely the same semantic of (5), while not requiring knowledge of M.

To enforce the "only if" part we need to introduce two further binary variables for each $(u, v) \in \tilde{F}$:

$$y^+_{uv} = \begin{cases} 1 & \text{if } (\sigma^{-1}(u), \sigma^{-1}(v)) \notin E \text{ and } \sum_{a \in A(u,v)} w(a) \geq d_{max} + 1 \\ 0 & \text{otherwise} \end{cases}$$

$$y^-_{uv} = \begin{cases} 1 & \text{if } (\sigma^{-1}(u), \sigma^{-1}(v)) \notin E \text{ and } \sum_{a \in A(u,v)} w(a) \leq d_{min} - 1 \\ 0 & \text{otherwise.} \end{cases}$$

These need be linked to the y_{uv} by the constraint

$$1 - y_{uv} = y^+_{uv} + y^-_{uv} \tag{7}$$

which guarantees that if $y_{uv} = 1$ (and therefore $d_{min} \leq \sum_{a \in A(u,v)} w(a) \leq d_{max}$ because of (6)) then $y^+_{uv} + y^-_{uv} = 0$, while if $y_{uv} = 0$ then exactly one among y^+_{uv} and y^-_{uv} is equal to one. We can then

finish up with the two further indicator constraints

$$\begin{array}{lcl} y^+_{uv} = 1 & \rightarrow & \sum_{a\in A(u,v)} w(a) \geq d_{max} + 1 \\ y^-_{uv} = 1 & \rightarrow & \sum_{a\in A(u,v)} w(a) \leq d_{min} - 1 \end{array} \tag{8}$$

which enforce that whenever one (and only one) among y^+_{uv} and y^-_{uv} is equal to one, then $\sum_{a\in A(u,v)} w(a)$ lies outside the interval $[d_{min}, d_{max}]$ (in either of the two possible directions).

Collating all the above constraints provides a valid ILP formulation (with indicator constraints) of our problem, that can therefore be solved by standard ILP tools. Note that we have not specified any objective function; indeed, since we are only interested in determining the feasibility of the integer system, we can leave the cost coefficients of all variables to zero. Alternatively we may add a single variable v, constraints

$$v \geq w(a) \qquad a \in A$$

and minimize the objective function v: this results in an optimization problem whereby we seek for the labels (if they exist) with minimum maximal value.

The model has $O(n^2)$ binary variables, $|A| + 2$ general integer variables, and $O(n^4)$ constraints, primarily due to (3) and (4) (all other constraints are $O(n^2)$). Even for relatively small n, the corresponding ILP may be rather large, and therefore difficult to solve. If the permutation σ were fixed, then the problem would considerably simplify: we would not need the x_{iu} variables and the corresponding constraints, and for each $(u, v) \in \tilde{F}$ we could determine a-priori whether or not $(\sigma^{-1}(u), \sigma^{-1}(v)) \in E$, that is, the value of the y_{uv} variables. Let us denote by $\hat{F}$ the subset of $\tilde{F}$ such that $(\sigma^{-1}(u), \sigma^{-1}(v)) \in E$, i.e., $y_{uv} = 1$. We must of course keep the general integer variables w, d_{min} and d_{max} in the model, but for each $(u, v) \in \hat{F}$ we can substitute the corresponding indicator constraints (6) with the simple

$$d_{min} \leq \sum_{a\in A(u,v)} w(a) \leq d_{max}.$$

For $(u, v) \notin \hat{F}$ ($y_{uv} = 0$), we must rather constrain $\sum_{a\in A(u,v)} w(a)$ to lie outside the interval; however, we must still choose the "side" of the interval, and therefore we still need indicator constraints like (8). Yet, because we know that $y_{uv} = 0$, (7) gives $1 = y^+_{uv} + y^-_{uv}$, i.e., $1 - y^-_{uv} = y^+_{uv}$; that is, the two variables $y^\pm_{uv}$ can be eliminated, and (8) can be simplified to

$$\begin{array}{lcl} z_{uv} = 1 & \rightarrow & \sum_{a\in A(u,v)} w(a) \geq d_{max} + 1 \\ z_{uv} = 0 & \rightarrow & \sum_{a\in A(u,v)} w(a) \leq d_{min} - 1 \end{array} \tag{9}$$

for binary variables z_{uv} only defined for $(u, v) \notin \hat{F}$. Therefore, denoting by $k = |\tilde{F}| - |\hat{F}| \leq |\tilde{F}| = n(n-1)/2$ the cardinality of the complement of $\hat{F}$, the model for fixed σ only has k binary variables, $|A| + 2$ general integer variables, k indicator constraints and $|\hat{F}| = n(n-1)/2 - k$ linear constraints. This model is significantly easier to solve: for $n = 7$, it only takes a small fraction of a second using a state-of-the-art ILP solver like `Cplex 12.3` on an ordinary laptop computer. Of course, the drawback is that we need, in principle, to solve $n!$ of such models, one for each of the possible permutations σ, in order to solve the overall problem. However, when we apply this model to the case in which G is the 7 vertex wheel W_7 and T is the centipede Π_7, we can exploit

the symmetry of the tree and of the graph to reduce the number of permutations to be considered. We can extend these arguments easily by induction to reduce the number of the permutations that need to be considered for $n \leq 11$. We have therefore written a small `C++` program that automatically constructs all these models, each one corresponding to one fixed permutation σ, and solves them with `Cplex 12.3`. None of these models turned out to have any feasible solution. The code and the accompanying files (encoding the necessary permutations) are freely available at `http://www.di.unipi.it/optimize/Software/PCGCat.html`.

Calling $\mathcal{M}$ the class of graphs that are PCGs of a caterpillar, in view of Theorem 1, this proves the following result:

Theorem 8 *The wheels W_n, $n = 7, \ldots, 11$ do not belong to $\mathcal{M}$.*

We recall that each graph G with at most 7 vertices is a PCG, and the witness tree is a centipede, except in the case of W_7, whose witness is a more general tree [4]. So, the previous theorem concludes the study of the graphs with at most 7 vertices.

5 PCGs of general trees

In the general case, the ILP model can be used to check whether a given graph is PCG or not implementing it by choosing the possible witness tree T among all the n leaf trees. Of course, this is impracticable. For this reason, in this section we study PCGs of general trees with the aim of understanding if there exists a unifying tree structure allowing one to check only it instead of all possible n leaf trees, so taking a role that is analogous to the centipede for all the caterpillars.

Theorem 9 *Let G be a graph, and T a tree. If $G = PCG(T, w, d_{min}, d_{max})$, then there always exists a full binary tree Λ, a new edge-weight function w', and a new value d'_{max} such that $G = PCG(\Lambda, w', d_{min}, d'_{max})$.*

Proof. Given T with a positive edge-weight function w, we will first construct Λ, with a non negative edge-weight function w'' and then we deduce a positive edge-weight function w' for Λ, modifying the value of d_{max} accordingly.

We perform a breadth first search on T; each time we examine a vertex v and its children on T, we construct a portion of Λ inserting both v and its children, guaranteeing that the new structure is a full binary tree. Namely, let us call $ch(v)$ the number of children of v and $N^+(v)$ the subtree of T induced by v and by all its children $c_1(v), \ldots c_{ch(v)}(v)$. If $ch(v) > 3$, then we substitute $N^+(v)$ with a $ch(v)$ leaf complete binary tree (that is, all levels, except possibly the deepest one are fully filled, and, if the last level is not complete, the nodes of that level are filled from left to right) whose root corresponds to v, and whose leaves correspond to the children of v in the same order from left to right. On this portion of Λ we define the weight function w'': calling $p(u)$ the parent vertex of a vertex u, for each edge $(c_i(v), p(c_i(v)))$, $1 \leq i \leq deg(v)$, define $w''((c_i(v), p(c_i(v)))) = w(c_i(v), v)$; the weights of all the other edges of the complete binary tree are set to 0. This portion of Λ must be merged with the previous currently constructed part, by overlapping the two copies of v, the one generated when v is considered as child of its father and the one just generated. Once that all the vertices of T have been examined, Γ is completely constructed. An example of execution of this

procedure is depicted in Figure 2.

It is easy to check that Λ is a full binary tree and that $PCG(T, w, d_{min}, d_{max})$ and $PCG(\Lambda, w'', d_{min}, d_{max})$ are in fact the same graph G.

It remains to modify the non negative edge-weight function w'' into a positive function w', varying the value of d_{max} accordingly. Let us define:

$$L = \min_{(u,v)\notin E(G)} \left\{ \left| d_{min} - d_{\Lambda,w''}(l_u, l_v) \right|, \left| d_{max} - d_{\Lambda,w''}(l_u, l_v) \right| \right\}$$
$$N = \left| \{ e : e \in E(\Lambda), w(e) = 0 \} \right|.$$

L is the smallest distance between the quantities d_{min}, d_{max} and the weighted distances on the tree of the paths corresponding to non-edges of G; N is the number of edges of Λ of weight 0.

Observe that, unless G coincides with the clique K_n (which trivially is PCG of a full binary tree), there always exists a pair of leaves such that their distance on Λ falls out of the interval $[d_{min}, d_{max}]$ and hence $L > 0$. Furthermore, as any edge incident to a leaf in Λ is strictly greater than 0, it is not difficult to see that in a full binary tree $N \leq 2n$. So, the value $\epsilon = \frac{L}{N+1}$ is well defined.

Now define a new weight function w' on Λ by assigning the weight ϵ to any edge of weight 0. More formally, $w'(e) = w''(e)$ if $w''(e) \neq 0$ and $w'(e) = \epsilon$ otherwise. In this way the distance between any two leaves in Λ can result increased by a value upper bounded by $\epsilon N < L$. Set the new value $d'_{max} = d_{max} + \epsilon N$.

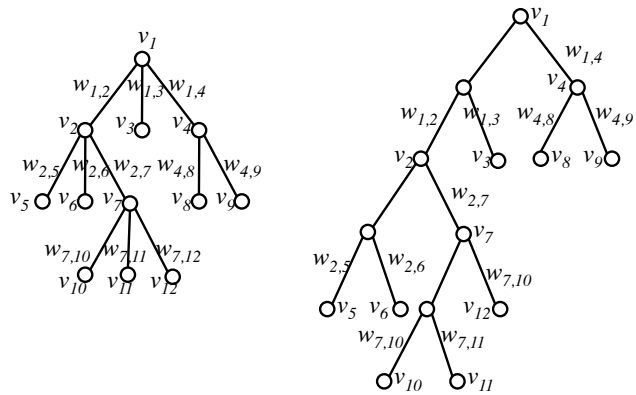
The following three observations conclude the proof:

- any distance between leaves in Λ that was strictly smaller than d_{min} with respect to the weight function w'' remains so after this transformation as $\epsilon N < L$;
- any distance that was strictly greater than d_{max} with respect to the weight function w'' is strictly greater than d'_{max} due to the definition of L;
- any distance that was in the interval $[d_{min}, d_{max}]$ with respect to the weight function w'' is now in the interval $[d_{min}, d'_{max}]$.

■ Unfortunately, the previous theorem does not guarantee to have a unique tree, but it is anyway an important improvement in the complexity of the pairwise compatibility tree construction problem, as it leads to consider only a particular subclass of all the n leaf trees.

6 Conclusions and Open Problems

In this paper we consider the pairwise tree construction problem with particular attention to the cases when the pairwise compatibility tree is a caterpillar. This was first motivated by the fact that in the literature, the pairwise compatibility tree construction problem of many graphs has as a solution a tree that is a caterpillar. Moreover, due to the simple and symmetric structure of this class of trees it is also one of the first non trivial cases to be considered when trying to identifying the class of PCGs generated by a specific tree structure.



a. b.

Figure 2: a) An example of tree T. b) T transformed into a full tree; edges without a weight are intended to have weight 0.

It is known that every graph that is PCG of a caterpillar is PCG of a centipede (for opportune values of weight function, d_{min} and d_{max}). In view of this, we first characterize the class $\mathcal{P}_n$ of graphs that are PCGs of a unit edge-weight centipede, and then we put it in relation with the class $\mathcal{I}_n$ of unit interval graphs, that are all the PCGs of a unit edge-weight caterpillar in the special case when $d_{min} = 0$.

For what concerns arbitrary edge-weighted centipedes, we give some conditions on w and d_{max} so that $PCG(\Pi_n, w, d_{min}, d_{max})$ is triangle free or has a certain fan as a subgraph.

Then, we propose an ILP model when the structure of the tree is given. We apply it to the special case when the graph is the 7 vertex wheel W_7 and the tree a centipede, so proving that W_7, (that is known to be PCG) cannot be PCG of a caterpillar. As a consequence, caterpillars cannot generate all the PCGs, so we focus on a more general tree structure, with the aim of understanding if there exists a unifying structure allowing one to check only it instead of all possible n leaf trees, so taking for all trees a role that is analogous to the centipede for all the caterpillars. We individuate this general structure in the full binary tree Γ_n. Unfortunately for our purposes, this structure is not as good as the centipede as, for a fixed n, there is a unique n leaf centipede but many n leaf full binary trees. Nevertheless, they are much less than all the n leaf trees.

Clearly this work gives rise to many open problems.

First, reminding that $\mathcal{M}$ denotes the class of PCGs generated by a caterpillar, it would be interesting to solve the following:

Problem 1: Give a complete characterization of the class $\mathcal{M}$.

Moreover, we have shown that W_7 is the smallest graph which is not a PCG of a caterpillar. As the same holds for $W_8, \ldots W_{11}$, it is natural to ask if this results extends to the whole class of wheels, i.e., if every wheel graph W_n with $n \geq 9$ is not a PCG of a caterpillar. In this context, we state the following conjecture:

Conjecture 1 *Let n be an integer such that $n \geq 7$, then $W_n \notin \mathcal{M}$.*

In fact, it is not even known if wheels on at least eight vertices are PCGs. We do not propose a conjecture here, although there is some evidence that these graphs are not PCGs: for example, the

tree presented in [4] for W_7 cannot be generalized and it seems it is an ad hoc construction for this particular case.

Finally, it would be clearly interesting to generalize the ILP model in order to enlarge the class of problems it can solve. For instance, it could be possible (although not straightforward) to extend it to determine whether a fixed graph G is the PGC of any tree (with fixed leaf set F) that is a subgraph of another given graph G'. Even modeling the problem of proving whether or not G is a PCG of any tree T, without any assumption on the shape of T, appears more difficult, but it would obviously be very helpful for the study of this problem.

References

[1] Kearney, P. E., Munro, J. I., Phillips, D. (2003) Efficient generation of uniform samples from phylogenetic trees. Proceedings of WABI '03. Budapest, Hungary, 15-20 September, Lecture Notes in Computer Science 2812, 177–189, Springer-Verlag, Berlin.

[2] Brandstädt, A., Le, V.B., Rautenbach, D. (2010) Exact leaf powers, *Theor. Comput. Sci.* 411, 2968–2977.

[3] Phillips, D. (2002) Uniform sampling from phylogenetics trees. Masters Thesis, University of Waterloo.

[4] Calamoneri,T., Frascaria, D., Sinaimeri, B. (2012) All graphs on at most seven vertices are PCG, *the Computer Journal* first published online July 18, doi:10.1093/comjnl/bxs087

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

[6] Yanhaona, M. N., Hossain, K. S. M. T., Rahman, M. S. (2009) Pairwise compatibility graphs. *J. Appl. Math. Comput.* 30, 479–503.

[7] Yanhaona, M. N., Bayzid, Md. S., Rahman, Md. S. (2010) Discovering Pairwise compatibility graphs. *Discrete Mathematics, Algorithms and Applications* 2(4), 607–623.

[8] Calamoneri,T., Petreschi, R., Sinaimeri, B. (2012) On relaxing the constraints in pairwise compatibility graphs, Proceedings of WALCOM 2012, Dhaka, Bangladesh, February 15-17, Lecture Notes in Computer Science 7157, 124–135, Springer-Verlag, Berlin.

[9] Salma, S. A., Rahman, Md. S. (2012) Triangle-Free Outer Planar 3-Graphs are Pairwise Compatibility Graphs, Proceedings of WALCOM 2012, Dhaka, Bangladesh, February 15-17, Lecture Notes in Computer Science 7157, 124–135, Springer-Verlag, Berlin.

[10] Calamoneri, T., Petreschi, R. (2013) Graphs with Dilworth Two are Pairwise Compatibility Graphs. Proceedings of LAGOS 2013, Playa del Carmen, Mexico, April 22-26, Electronic Notes in Theoretical Computer Science, Elsevier.

[11] Durocher, S., Debajyoti, M., Rahman, Md. S. (2013) On Graphs that are not PCGs, Proceedings of WALCOM 2013, Kharagpur, India, February 14-16, Lecture Notes in Computer Science 7748, 310–321.

[12] Calamoneri, T., Montefusco, E., Petreschi, R., Sinaimeri, B. (2013) Exploring Pairwise Compatibility Graphs, Theoretical Computer Science 468: 23–36.

[13] Brandstädt, A., Hundt, C. (2008) Ptolemaic Graphs and Interval Graphs Are Leaf Powers. Proceedings of LATIN 2008, Búzios, Brazil, April 7-11, Lecture Notes in Computer Science 4957, pp. 479–491, Springer-Verlag, Berlin.

[14] Nishimura, N., Ragde, P., Thilikos, D. M. (2002) On graph powers for leaf-labeled trees. *J. Algorithms* 42, 69–108.