On Graphs that are not PCGs☆

Stephane Durocher[1]

Department of Computer Science, University of Manitoba, Canada

Debajyoti Mondal[2]

Department of Computer Science, University of Manitoba, Canada

Md. Saidur Rahman[3]

Graph Drawing & Information Visualization Laboratory,
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

Abstract

Let T be an edge-weighted tree and let d_{min}, d_{max} be two nonnegative real numbers. The pairwise compatibility graph (PCG) of T is a graph G such that each vertex of G corresponds to a distinct leaf of T and two vertices are adjacent in G if and only if the weighted distance between their corresponding leaves in T is in the interval $[d_{min}, d_{max}]$. Similarly, a given graph G is a PCG if there exist suitable T, d_{min}, d_{max}, such that G is a PCG of T. Yanhaona, Bayzid and Rahman proved that there exists a graph with 15 vertices that is not a PCG. On the other hand, Calamoneri, Frascaria and Sinaimeri proved that every graph with at most seven vertices is a PCG. In this paper we

☆A preliminary version of these results appeared in the Proceedings of the 7th International Workshop on Algorithms and Computation (WALCOM 2013) [9].

Email addresses: `durocher@cs.umanitoba.ca` (Stephane Durocher), `jyoti@cs.umanitoba.ca` (Debajyoti Mondal), `saidurrahman@cse.buet.ac.bd` (Md. Saidur Rahman)

[1]Work of the author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

[2]Work of the author is supported in part by a University of Manitoba Graduate Fellowship.

[3]Work of the author is supported in part by the Ministry of Science and Information & Communication Technology, Government of Bangladesh.

construct a graph of eight vertices that is not a PCG, which strengthens the result of Yanhaona, Bayzid and Rahman, and implies optimality of the result of Calamoneri, Frascaria and Sinaimeri. We then construct a planar graph with sixteen vertices that is not a PCG. Finally, we prove a variant of the PCG recognition problem to be NP-complete.

Keywords: pairwise compatibility graph, leaf powers, tree-based distance metric

---

1. Introduction

Let T be an edge-weighted tree and let d_{min}, d_{max} be two nonnegative real numbers. The *pairwise compatibility graph (PCG)* of T is a graph G such that each vertex of G corresponds to a distinct leaf of T and two vertices are adjacent in G if and only if the weighted distance between their corresponding leaves in T is in the interval $[d_{min}, d_{max}]$. Similarly, a given graph G is a PCG if there exist suitable T, d_{min}, d_{max}, such that G is a PCG of T. Figure 1(a) illustrates an edge-weighted tree T, and Figure 1(b) shows the corresponding PCG G, where $d_{min} = 2$ and $d_{max} = 3.5$. Figure 1(c) shows another edge-weighted tree T' such that G is a PCG of T' when $d_{min} = 1.5$ and $d_{max} = 2$.

In 2003, Kearney et al. [11] introduced the concept of PCG and showed how to use it to model evolutionary relationships among a set of organisms. Moreover, they proved that the problem of finding a maximal clique can be solved in polynomial time for pairwise compatibility graphs if one can find their corresponding edge-weighted trees in polynomial time. They hoped to show that every graph is a PCG, but later, Yanhaona et al. [17] constructed a 15-vertex graph that is not a PCG.

Several researchers have attempted to characterize pairwise compatibility graphs. Yanhaona et al. [18] proved that graphs having cycles as its maximal biconnected components are PCGs. Salma and Rahman [15] proved that every triangle-free maximum-degree-three outerplanar graph is a PCG. Calamoneri et al. [8] gave some sufficient conditions for a split matrogenic graph to be a PCG, and examined the graph classes that arise from using the intervals $[0, d_{max}]$ (LPG, also known as leaf powers) and $[d_{min}, \infty]$ (mLPG) [1, 3, 13]. They proved that the intersection of these classes is not empty, and neither of them is contained in the other. Several variants of these graph classes, e.g., exact k-leaf powers, (k, l)-leaf powers and so on, have been extensively studied in the literature [1, 2, 3, 10, 12].

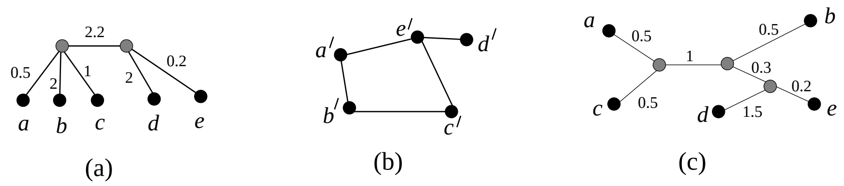

Figure 1: (a) An edge-weighted tree T. (b) A PCG G of T, where $d_{min} = 2, d_{max} = 3.5$. (c) Another edge weighted tree T' such that G is a PCG of T' when $d_{min} = 1.5, d_{max} = 2$.

Finding a pairwise compatibility tree of a given graph appeared to be difficult, even for graphs with few vertices. In 2003, Kearney et al. [11] showed that every graph with at most five vertices is a PCG. Prior to the results of this paper, the smallest graph known not to be a PCG was a 15-vertex graph constructed by Yanhaona et al. [17]. This graph consists of a bipartite graph with partite sets A and B, where $|A| = 5$ and $|B| = 10$, and each subset of three vertices of A is adjacent to a distinct vertex of B. Phillips [14] proved that every graph with at most five vertices is a PCG, and later Calamoneri et al. [4] showed that every graph with at most seven vertices is also a PCG. Some recent research examined the pairwise compatibility of graphs with bounded Dilworth number [5, 6, 7].

In this paper we construct a graph of eight vertices that is not a PCG, which strengthens the result of Yanhaona et al. [17], and implies optimality of the result of Calamoneri et al. [4]. We then construct a planar graph with twenty vertices that is not a PCG; this is the first planar graph known not to be a PCG. Finally, we examine a generalized PCG recognition problem that given a graph G and a subset S of edges of its complement graph, asks to determine a PCG $G' = (T, d_{min}, d_{max})$ that contains G as a subgraph, but does not contain any edge of S. Observe that if S contains all the edges of the complement graph, then it is the problem of deciding whether G is a PCG. Thus the PCG recognition problem is a special case of the generalized PCG recognition problem. We prove that the generalized PCG recognition problem is NP-hard if we require maximum number of edges of S to have weighted tree distance greater than d_{max} between their corresponding leaves. We hope that this is a step towards understanding the complexity of the PCG recognition problem, and conjecture both the PCG recognition problem and its generalized version to be NP-hard.

The rest of the paper is organized as follows. In Section 2 we discuss some technical background. In Section 3 we construct a graph G_1 with nine vertices that is not a PCG. The construction of G_1 motivated us to study the structural properties of G_1 for obtaining a graph G_2 of eight vertices by deleting a vertex from G_1 so that G_2 is not a PCG. This would give a tight result since every graph with at most seven vertices is a PCG. In Section 4 we thus analyze and compile the structural properties of G_1, and prove that the graph G_2 obtained by deleting a vertex of degree three from G_1 is not a PCG. In Section 5 we use the building blocks and ideas of Sections 3 and 4 to construct a planar graph that is not a PCG. In Section 6 we prove the NP-hardness result. Finally, Section 7 concludes the paper.

2. Preliminaries

In this section we introduce some definitions and review relevant results. Let $G = (V, E)$ be a graph with vertex set V and edge set E. The complement graph $\overline{G}$ of G is the graph with vertex set V and edge set $\overline{E}$, where $\overline{E}$ consists of the edges that are determined by the non-adjacent pairs of vertices of G. For a vertex v (respectively, a set of vertices S) in G, we use the notation $G \setminus v$ (respectively, $G \setminus S$) to denote the subgraph of G induced by the vertices $V \setminus \{v\}$ (respectively, $V \setminus S$). Let T be an edge-weighted tree. Let u and v be two leaves of T. By P_{uv} we denote the unique path between u and v in T. By $d_T(u, v)$ we denote the *weighted distance* between u and v, i.e., the sum of the weights of the edges on P_{uv}. Let d_{min}, d_{max} be two nonnegative real numbers. Then by $PCG(T, d_{min}, d_{max})$ we denote the PCG of T that respects the interval $[d_{min}, d_{max}]$. By $T_{x_1x_2\ldots x_t}$ we denote the subgraph of T induced by the paths $P_{x_ix_j}$, where $1 \leq i, j \leq t$. Figures 2(a)–(b) illustrate an example of such a subgraph.

Lemma 1 (Yanhaona et al. [17]). *Let T be an edge-weighted tree, and let u, v and w be three leaves of T such that P_{uv} is the longest path in T_{uvw}. Let x be a leaf of T other than u, v and w. Then $d_T(w, x) \leq d_T(u, x)$, or $d_T(w, x) \leq d_T(v, x)$.*

Let $G = PCG(T, d_{min}, d_{max})$. Then by u' we denote the vertex of G that corresponds to the leaf u of T. The following lemma illustrates a relationship between a PCG and its corresponding edge-weighted tree.

Lemma 2 (Yanhaona et al. [17]). *Let $G = PCG(T, d_{min}, d_{max})$. Let a, b, c, d, e be five leaves of T and a', b', c', d', e' be the corresponding vertices of G,*

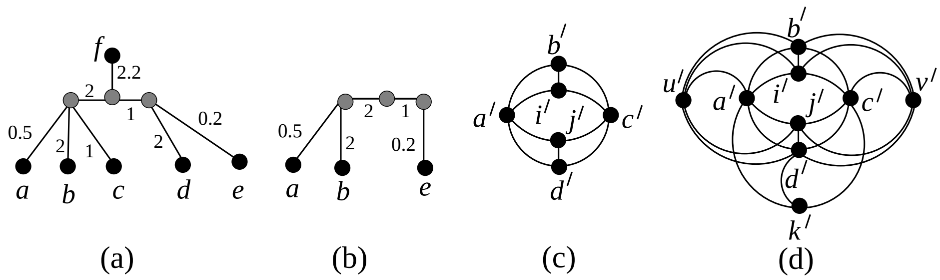

Figure 2: (a) An edge-weighted tree T. (b) T_{abe}. Illustration for (c) H and (d) G_1.

respectively. Let P_{ae} and P_{bd} be the longest path in T_{abcde} and T_{bcd}, respectively. Then any vertex x' in G that is adjacent to a', c', e' must be adjacent to at least one vertex in $\{b', d'\}$.

3. Not all 9-Vertex Graphs are PCGs

In this section we construct a graph G_1 of nine vertices that is not a PCG. Here we describe an outline of the construction.

We use three lemmas to construct G_1. In Lemma 3 we prove that for a cycle a', b', c', d' of four vertices, $d_T(a, c)$ and $d_T(b, d)$ cannot be both greater than d_{max}. We then construct a graph H with six vertices a', b', c', d', i', j' such that each pair of vertices in H are adjacent except the pairs (a', c'), (b', d'), (i', d'), (j', b'), (i', j'), as shown in Figure 2(c). Using Lemma 3 we prove in Lemma 4 that at least one of $d_T(a, c), d_T(b, d), d_T(i, d), d_T(j, b), d_T(i, j)$ must be greater than d_{max}. In Lemma 5 we prove that any PCG that contains H as an induced subgraph must satisfy the inequality $d_T(a, c) < d_{min}$, where a' and c' are the only vertices of degree four in H. We add three vertices k', u', v' to H to construct G_1, as shown in Figure 2(d). In Theorem 1 we show that for every non-adjacent pair (x', y') in H, the graph G_1 contains an induced subgraph isomorphic to H that contains x' and y' as its degree four vertices. By Lemma 5, $d_T(x, y) < d_{min}$. Observe that this contradicts Lemma 4. Consequently, G cannot be a PCG.

In the following lemma we prove that for a cycle a', b', c', d' of four vertices, $d_T(a, c)$ and $d_T(b, d)$ cannot be both greater than d_{max}.

Lemma 3. *Let C be the cycle a', b', c', d' of four vertices. If C=$PCG(T, d_{min}, d_{max})$ for some tree T and values d_{min} and d_{max}, where the leaves a, b, c and d*

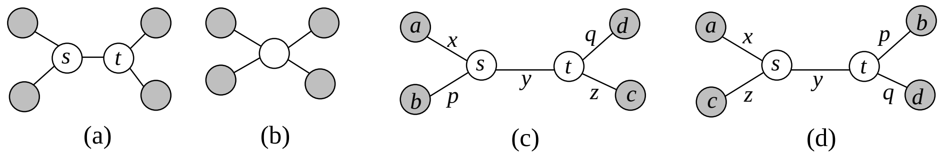

Figure 3: (a)–(b) Possible topologies for T. Illustration for (c) Case 1, and (d) Case 2, where each edge is labeled with its corresponding weight.

of T correspond to the vertices a', b', c' and d' of G, respectively, then $d_T(a,c)$ and $d_T(b,d)$ cannot be both greater than d_{max}.

Proof. Without loss of generality we assume that T does not have any vertices of degree two. Otherwise, it is straightforward to replace each vertex of degree two and its adjacent edges with a single edge, where the weight of the new edge is the sum of the weights of the two corresponding deleted edges. Consequently, T can have one of the two topologies as depicted in Figures 3(a)–(b). Since the topology of Figure 3(b) can be derived from the topology of Figure 3(a) by setting the weight of edge (s,t) to zero, we only examine the topology of Figure 3(a). Without loss of generality, we now need to consider two cases depending on the position of the leaves in this tree topology.

Case* 1*: Assume that T takes the form of Figure 3(c).

Suppose for a contradiction that both $d_T(a,c)$ and $d_T(b,d)$ are greater than d_{max}. Then $d_T(a,c) + d_T(b,d) = x + z + p + q + 2y > 2d_{max}$. Since a' and d' are adjacent, $d_{min} \leq x + y + q \leq d_{max}$. Again, since b' and c' are adjacent, $d_{min} \leq p+y+z \leq d_{max}$. Consequently, $x+z+p+q+2y \leq 2d_{max}$, which contradicts that $d_T(a,c) + d_T(b,d) > 2d_{max}$.

Case* 2*: Assume that T takes the form of Figure 3(d).

Suppose for a contradiction that both $d_T(a,c)$ and $d_T(b,d)$ are greater than d_{max}, i.e., $d_T(a,c) = x + z > d_{max}$ and $d_T(b,d) = p + q > d_{max}$. Then either $x > d_{max}/2$ or $z > d_{max}/2$. Similarly, either $p > d_{max}/2$ or $q > d_{max}/2$. Since $d_T(a,b) \geq x + p, d_T(b,c) \geq z + p, d_T(c,d) \geq z + q$ and $d_T(a,d) \geq x + q$, one of the four pairs among $(a',b'), (b',c'), (c',d'), (a',d')$ must be non-adjacent in G. This contradicts that a', b', c', d' is a cycle. □

We now construct a graph H with six vertices a', b', c', d', i', j' such that each pair of vertices in H are adjacent except the pairs $(a',c'), (b',d'), (i',d')$, $(j',b'), (i',j')$, as shown in Figure 2(c). The following lemma proves that at least one of $d_T(a,c), d_T(b,d), d_T(i,d), d_T(j,b), d_T(i,j)$ must be greater than d_{max}.

Lemma 4. *Let $H = PCG(T, d_{min}, d_{max})$. Let a, b, c, d, i, j be the leaves of T that correspond to the vertices a', b', c', d', i', j' of H. Then at least one of $d_T(a,c)$, $d_T(b,d), d_T(i,d), d_T(j,b), d_T(i,j)$ must be greater than d_{max}.*

PROOF. For each pair $(x', y') \in \{(a', c'), (b', d'), (i', d'), (j', b'), (i', j')\}$, x' and y' are non-adjacent in H. Therefore, either $d_T(x, y) < d_{min}$ or $d_T(x, y) > d_{max}$.

If one of $d_T(a,c), d_T(b,d), d_T(i,d), d_T(j,b)$ is greater than d_{max}, then the lemma holds irrespective of whether $d_T(i,j) < d_{min}$ or $d_T(i,j) > d_{max}$. We thus assume that each of $d_T(a,c), d_T(b,d), d_T(i,d), d_T(j,b)$ is less than d_{min}, and then prove that $d_T(i,j)$ must be greater than d_{max}.

Suppose for a contradiction that $d_T(i,j) < d_{min}$. Recall that we assumed $d_T(j,b) < d_{min}$. Consequently, since i' and b' are adjacent in H, the path P_{ib} must be the longest path T_{ijb}. By Lemma 1, $d_T(j,d) \leq d_T(i,d)$ or $d_T(j,d) \leq d_T(b,d)$. Since we assumed that $d_T(i,d) < d_{min}$ and $d_T(b,d) < d_{min}$, the inequality $d_T(j,d) < d_{min}$ holds. But this contradicts that j', d' are adjacent in G. Therefore, $d_T(i,j)$ must be greater than d_{max}. □

In the following lemma we prove that any PCG that contains H as an induced subgraph must satisfy the inequality $d_T(a,c) < d_{min}$, where a' and c' are the only vertices of degree four in H.

Lemma 5. *Let $G = PCG(T, d_{min}, d_{max})$ be a graph that contains an induced subgraph G' isomorphic to H. Let a, b, c, d, i, j be the leaves of T that correspond to the vertices a', b', c', d', i', j' of G'. Let a' and c' be the vertices of degree four in G'. Then $d_T(a,c)$ must be less than d_{min}.*

PROOF. Since a', c' are non-adjacent in G', either $d_T(a,c) < d_{min}$ or $d_T(a,c) > d_{max}$. Suppose for a contradiction that $d_T(a,c) > d_{max}$.

Since the subgraph induced by a', b', c', d' is a cycle, by Lemma 3, $d_T(b', d') < d_{min}$. Again, since the subgraph induced by a', i', c', d' is a cycle, by Lemma 3, $d_T(i', d') < d_{min}$. Consequently, P_{bi} is the longest path in T_{ibd}. Observe that we assumed $d_T(a,c) > d_{max}$. On the other hand, for each pair $(x', y') \in \{(a', b'), (a', d'), (a', i'), (b', d'), (b', c'), (b', i'), (c', d'), (c', i'), (d', i')\}$, $d_T(x, y) \leq d_{max}$. Therefore, P_{ac} is the longest path in T_{abcdi}.

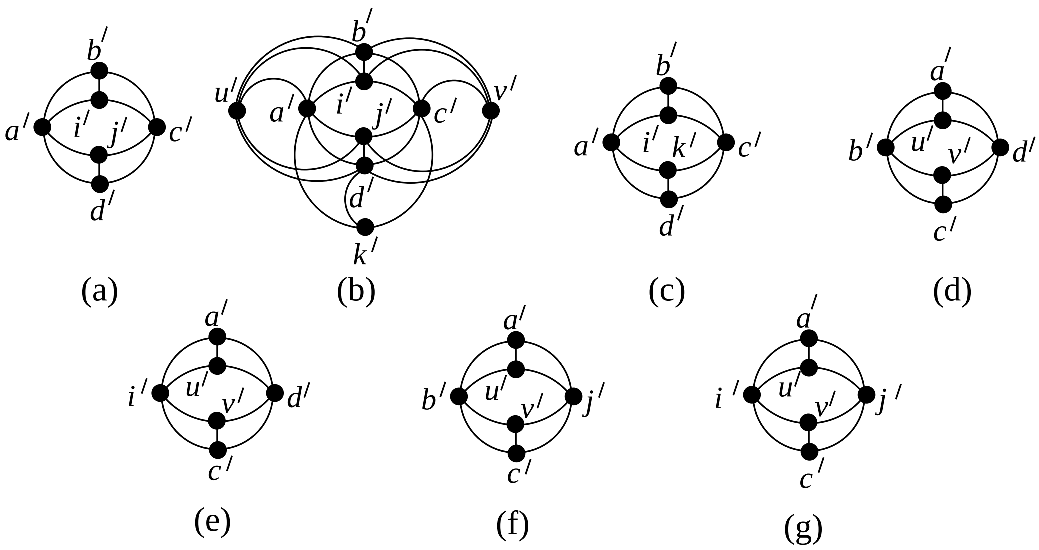

Figure 4: (a) H. (b) G_1. (c)–(g) Five induced subgraphs of G, when (c) $d_T(a,c) > d_{max}$, (d) $d_T(b,d) > d_{max}$, (e) $d_T(i,d) > d_{max}$, (f) $d_T(j,b) > d_{max}$, (g) $d_T(i,j) > d_{max}$.

By Lemma 2, any vertex j' in G' that is adjacent to a', c', d' must be adjacent to i' or b'. Although j' is adjacent to a', c', d' in G, neither i' nor b' is adjacent to j', a contradiction. □

We now add three vertices k', u', v' to H to construct G_1, as shown in Figures 4(a)–(b). In the following theorem we show that G_1 is not a PCG.

Theorem 1. *G_1 is not a PCG.*

Proof. For every non-adjacent pair (x', y') in H, the graph G_1 contains an induced subgraph isomorphic to H that contains x' and y' as its degree four vertices, as shown in Figures 4(c)–(g). By Lemma 5, $d_T(x,y) < d_{min}$. This contradicts Lemma 4 that says there exists a non-adjacent pair (x', y') in H such that $d_T(x,y) > d_{max}$. Consequently, G cannot be a PCG. □

4. Not all 8-Vertex Graphs are PCGs

In this section we analyze the structure of the graph G_1, and modify it to obtain a graph of eight vertices that is not a PCG.

We refer the reader to Figure 4. Observe that G_1 has only one vertex of degree three, i.e., vertex k'. The proof of Theorem 1 refers to vertex k' only in the case when $d_T(a, c) > d_{max}$, as shown in Figure 4(c). This observation inspired us to examine whether the graph $G_1 \setminus k'$ is a PCG or not. In this section we denote the graph $G_1 \setminus k'$, shown in Figure 5(a), by G_2 and prove that G_2 is not a PCG. The following lemma will be useful to prove the main result.

Lemma 6. *Let G be a graph of four vertices a', b', c', d' and two edges (a', b') and $(c'd')$. If $G = PCG(T, d_{min}, d_{max})$ for some tree T and values d_{min} and d_{max}, where the leaves a, b, c and d of T correspond to the vertices a', b', c' and d' of G, respectively, then at least one of $d_T(a, d), d_T(b, d)$, $d_T(b, c), d_T(a, c)$ must be greater than d_{max}.*

Proof. Since every pair of vertices among $(a', d'), (b', d'), (b', c'), (a', c')$ are non-adjacent in G, each of $d_T(a, d), d_T(b, d), d_T(b, c), d_T(a, c)$ is either greater than d_{max} or less than d_{min}. Suppose for a contradiction that $d_T(a, d), d_T(b, d)$, $d_T(b, c)$, $d_T(a, c)$ are less than d_{min}.

Since a' and b' are adjacent and $d_T(a, c), d_T(b, c)$ are less than d_{min}, P_{ab} must be the longest path in T_{abc}. By Lemma 1, $d_T(c, d) \leq d_T(a, d)$ or $d_T(c, d) \leq d_T(b, d)$. By assumption, both $d_T(a, d)$ and $d_T(b, d)$ are less than d_{min}. Therefore, $d_T(c, d) < d_{min}$, which contradicts that c' and d' are adjacent in G. □

We now use Lemma 6 to obtain the following lemma.

Lemma 7. *Let $G_2 = PCG(T, d_{min}, d_{max})$ and let a, b, c, d, i, j, u, v be the leaves of T that correspond to the vertices $a', b', c', d', i', j', u', v'$ of G_2. Then (a) at least one of $d_T(u, v), d_T(a, v), d_T(a, c), d_T(u, c)$ must be greater than d_{max}, and (b) at least one of $d_T(b, j)$, $d_T(b, d), d_T(i, d), d_T(i, j)$ must be greater than d_{max}.*

Proof. We only prove claim (a), i.e., one of $d_T(u, v), d_T(a, v), d_T(a, c)$ must be greater than d_{max}, since the proof for claim (b) is similar.

Since every pair of vertices among $(u', v'), (a', v'), (a', c')$ are non-adjacent in G_2, each of $d_T(u, v), d_T(a, v), d_T(a, c)$ is either greater than d_{max} or less than d_{min}. Suppose for a contradiction that $d_T(u, v), d_T(a, v), d_T(a, c)$ are less than d_{min}

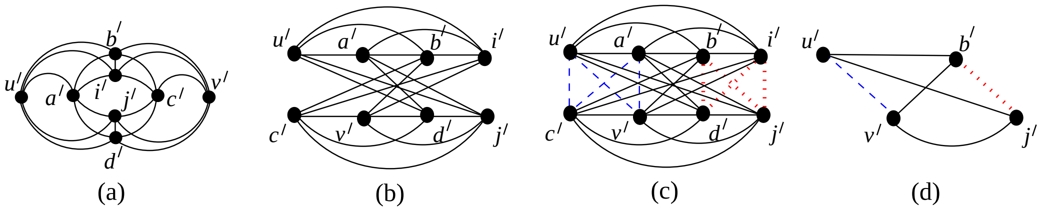

Figure 5: (a) G_2. (b) Another drawing of G_2. (c) Illustration for $((w', x'), (y', z'))$, where (w', x') and (y', z') are shown in dashed lines and dotted lines, respectively. (d) $((w', x'), (y', z')) = ((u', v'), (b', j'))$.

Since u' and a' are adjacent and $d_T(u, c), d_T(a, c)$ are less than d_{min}, P_{au} must be the longest path in T_{acu}. By Lemma 1, $d_T(c, v) \leq d_T(a, v)$ or $d_T(c, v) \leq d_T(u, v)$. Recall that according to our assumption, $d_T(u, v), d_T(a, v)$ are less than d_{min}. Therefore, $d_T(c, v) < d_{min}$, which contradicts that c' and v' are adjacent in G_2. $\square$

Theorem 2. *G_2 is not a PCG.*

Proof. Suppose for a contradiction that $G_2 = PCG(T, d_{min}, d_{max})$, where a, b, c, d, i, j, u, v are the leaves of T that correspond to the vertices $a', b', c', d', i', j', u', v'$ of G_2. Observe that for any $((w', x'), (y', z'))$, where $(w', x') \in \{(u', v'), (a', v'), (a', c'), (u', c')\}$ and $(y', z') \in \{(b', j'), (b', d'), (i', d'), (i', j')\}$, the vertices $\{w', x', y', z'\}$ induce a cycle C such that w', x' and y', z' are non-adjacent in C. Figures 5(b)–(d) illustrate this scenario. By Lemma 7, for some $((w', x'), (y', z'))$, both $d_T(w, x)$ and $d_T(y, z)$ are greater than d_{max}. This contradicts Lemma 3 since the vertices $\{w', x', y', z'\}$ induce a cycle. $\square$

5. Not all Planar Graphs are PCGs

In this section we prove that the planar graph G_p with twenty vertices, shown in Figure 6(a), is not a PCG.

Theorem 3. *G_p is not a PCG.*

Proof. Suppose for a contradiction that $G_p = PCG(T, d_{min}, d_{max})$, where $a, b, \ldots, s, t$ are the leaves of T that correspond to the vertices $a', b', \ldots, s', t'$ of G_p.

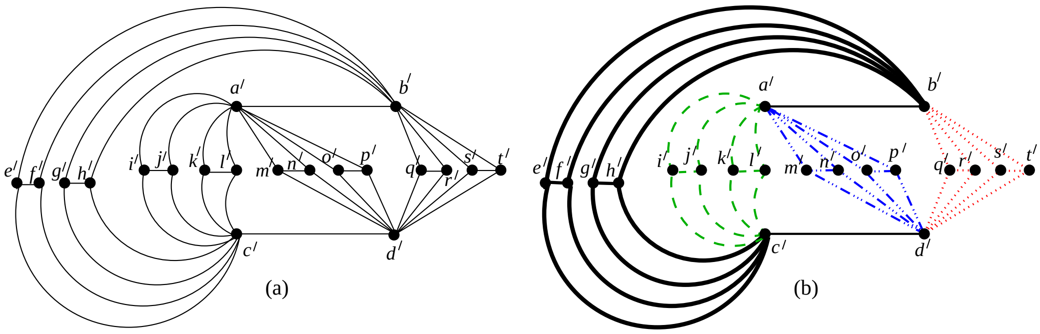

Figure 6: (a) G_p. (b) Illustration for the proof of Theorem 3. The graphs isomorphic to H are shown in bold lines ($d_T(b,c) > d_{max}$), regular dashed lines ($d_T(a,c) > d_{max}$), regular dotted lines ($d_T(b,d) > d_{max}$) and irregular dashed lines ($d_T(a,d) > d_{max}$).

Since a', b', c', d' induce a graph with two edges (a', b') and (c', d'), by Lemma 6, one of $d_T(a,d)$, $d_T(b,d), d_T(b,c), d_T(a,c)$ must be greater than d_{max}. For any pair $(x', y') \in \{(a', d'), (b', d'), (b', c'), (a', c')\}$, there exists an induced subgraph in G_p that is isomorphic to H (i.e., the graph of Figure 4(c)) that contains x' and y' as its degree four vertices. By Lemma 5, $d_T(x,y) < d_{min}$, which contradicts that at least one of $d_T(a,d), d_T(b,d), d_T(b,c), d_T(a,c)$ must be greater than d_{max}. Consequently, G_p cannot be a PCG. □

Observe that G_p has twenty vertices. However, the proof of Theorem 3 holds even for the planar graph obtained from G_p by merging the pair of vertices $(e', t'), (h', i'), (l', m'), (p', q')$ and then removing the resulting multi-edges. Therefore, there exists a planar graph with sixteen vertices that is not a PCG.

6. NP-hardness

In this section we examine a generalized PCG recognition problem that given a graph $G = (V, E)$ and a set $S \subseteq \overline{E}$, asks to determine a PCG $G' = (T, d_{min}, d_{max})$ that contains G as a subgraph but does not contain any edge of S, where $\overline{E}$ is the set of edges in the complement graph of G. Observe that if $S = \overline{E}$, then the problem asks to decide whether G is a PCG. We prove that the generalized PCG recognition problem is NP-hard if we require maximum number of edges of S to have weighted tree distance greater than d_{max} between their corresponding leaves. A decision version of the problem is as follows.

Problem : MAX-GENERALIZED-PCG-RECOGNITION

Instance : A graph G, a subset S of the edges of its complement graph, and a positive integer k.

Question : Is there a PCG $G' = PCG(T, d_{min}, d_{max})$ such that G' contains G as a subgraph (not necessarily an induced subgraph), but does not contain any edge of S; and at least k edges of S have distance greater than d_{max} between their corresponding leaves in T?

We prove the NP-hardness of MAX-GENERALIZED-PCG-RECOGNITION by reduction form MONOTONE-ONE-IN-THREE-3-SAT [16].

Problem : MONOTONE-ONE-IN-THREE-3-SAT

Instance : A set U of variables and a collection C of clauses over U such that each clause consists of exactly three non-negated literals.

Question : Is there a satisfying truth assignment for U such that each clause in C contains exactly one true literal?

Given an instance $I(U, C)$ of MONOTONE-ONE-IN-THREE-3-SAT, we construct an instance $I(G, S, k)$ of MAX-GENERALIZED-PCG-RECOGNITION such that $I(U, C)$ has an affirmative answer if and only if $I(G, S, k)$ has an affirmative answer. The idea of the reduction is as follows. Given an edge-weighted tree T with n leaves, $d_{min} = 0$ and $d_{max} = +\infty$, the corresponding PCG is a complete graph K_n of n vertices. Observe that as the interval $[d_{min}, d_{max}]$ begins to shrink, more and more edges of K_n disappear. Some edges disappear due to the increase of d_{min} and some other edges disappear due to the decrease of d_{max}. We use these two events to set the truth values of the literals.

Let G_{not} be the graph of Figure 7(a). The following lemma shows how to use this graph as a NOT gate.

Lemma 8. *Assume that $G_{not} = PCG(T, d_{min}, d_{max})$, where $a, b, \ldots, q$ are the leaves of T that correspond to the vertices $a', b', \ldots, q'$ of G_{not}. Then $d_T(a, b) < d_{min}$ if and only if $d_T(c, d) > d_{max}$.*

PROOF. By Lemma 6, one of $d_T(e,g), d_T(e,h), d_T(f,g), d_T(f,h)$ must be greater than d_{max}. Observe that for any pair $(x,y) \in \{(e',g'),(e',h'),(f',g'),(f',h')\}$, the vertices b',x',d',y' form an induced cycle. Therefore, by Lemma 3, $d_T(b,d) < d_{min}$. Similarly, we can prove that $d_T(a,q) < d_{min}$ and $d_T(c,q) < d_{min}$. Since a',c',b',q',d' induce a cycle of five vertices, one of $d_T(a,b), d_T(c,d)$, $d_T(a,q)$, $d_T(c,q)$, $d_T(b,d)$ is greater than d_{max} [8, Lemma 2]. Since $d_T(a,q)$, $d_T(c,q), d_T(b,d)$ are less than d_{min}, one of or both $d_T(a,b)$ and $d_T(c,d)$ are greater than d_{max}.

Without loss of generality assume that $d_T(a,b) > d_{max}$. Then by Lemma 1, $d_T(c,d) \leq d_T(a,d)$ or $d_T(c,d) \leq d_T(b,d)$. Since $d_T(a,d) \leq d_{max}$ and $d_T(b,d) < d_{min} \leq d_{max}$, $d_T(c,d)$ must be less than d_{min}. Similarly, we can prove that if $d_T(c,d) > d_{max}$, then $d_T(a,b) < d_{min}$. □

6.1. Properties of G_{not}

The vertices a,b and c,d play the role of the input and output of a NOT gate, respectively. Figure 7(b) illustrates a pairwise compatibility tree T, where $G_{not} = PCG(T,7,11)$ and $d_T(a,b) > d_{max}$. Observe that once we construct the tree T_{abqcd}, it becomes straightforward to add the trees T_{efgh}, T_{ijkl} and T_{mnop}. Therefore, in the rest of this section we only consider the simplified representation for T, as shown in Figure 7(c). We can cascade several NOT gates to duplicate or invert the input, as illustrated below.

Cascading of NOT gates. We can cascade NOT gates to duplicate or invert the input. Figure 7(d) illustrates the cascading of NOT gates. Figure 7(e) shows the simplified tree representations for three gates of Figure 7(d) assuming that the literal corresponding to a',b' is true. Observe that if any input pair (respectively, output pair) x,y of the NOT gate is true (respectively, false), then the corresponding unique path in the tree has the weight sequence $(4,2,2,4)$ (respectively, $(2,2,2)$). Each time we cascade a new gate, we maintain this invariant as follows. If the new tree T', i.e., the tree corresponding to the new gate, and the existing tree T contain a common subgraph T'', then we add to T the edges and vertices of T' that does not belong to T''. We denote this operation as a *tree merging operation*, i.e., merging of T' into T. Figure 7(f) illustrates the tree that corresponds to the cascading of the NOT gates of Figure 7(d). The PCG G' of the final tree T contains all the edges that belong to the constituent G_{not} graphs, and also many redundant edges (e.g., the edges $(a',c'_3),(c'_2,c'_3)$ and so on). However,

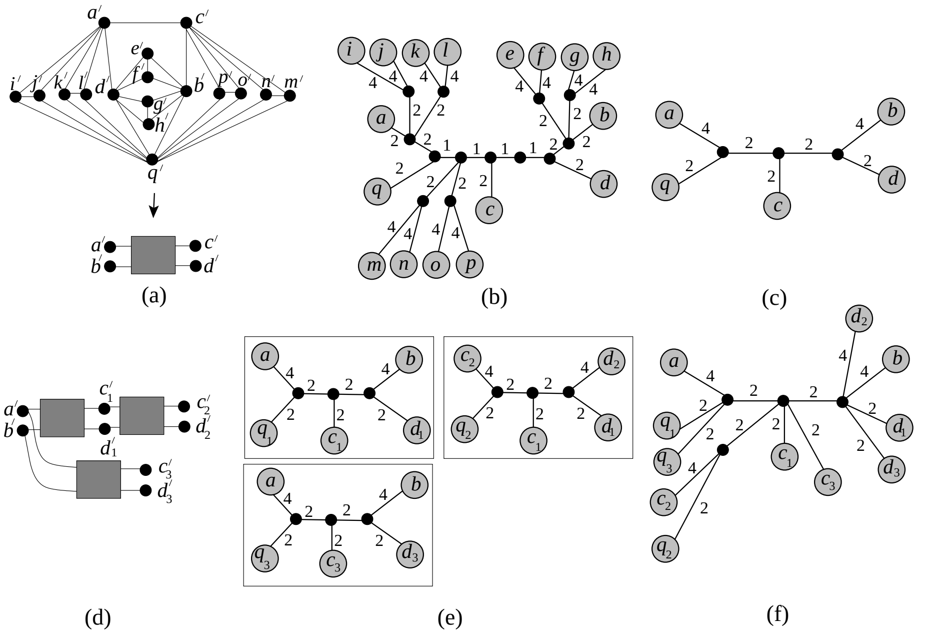

Figure 7: (a) G_{not}, and its hypothetical representation. (b) $G_{not} = PCG(T, 7, 11)$. (c) Simplified representation of T. (d)–(f) Illustration for the cascading of NOT gates.

since the PCG of every tree that we used to construct T is a G_{not}, none of these redundant edges can belong to a single G_{not}.

In the reduction, all the edges of $\overline{G_{not}}$ will belong to S. Every G_{not} has 101 non-adjacent pairs, and by construction, in any pairwise compatibility tree T' of G_{not}, all the distances $d_{T'}(a, q), d_{T'}(c, q), d_{T'}(b, d)$ and one of $d_{T'}(a, b), d_{T'}(c, d)$ must be less than d_{min}. Therefore, at most 97 edges of $\overline{G_{not}}$ can have distance greater than d_{max} between their corresponding leaves in T'. Since the tree T, shown in Figure 7(b), determines 97 such edges, it maximizes the number of edges of $\overline{G_{not}}$ that have distance greater than d_{max} between their corresponding leaves.

6.2. Literal and Clause Gadgets

Each literal gadget consists of a pair of non-adjacent vertices. Every edge determined by these two vertices, belongs to S. We say that *a literal (or, any non-adjacent pair of vertices)* (a', b') *is true* if and only if $d_T(a, b) > d_{max}$; otherwise, it is *false*.

Every clause gadget G_{clause}, as shown in Figure 8(a), corresponds to a logic circuit L that is consistent if and only if at most one of its three inputs

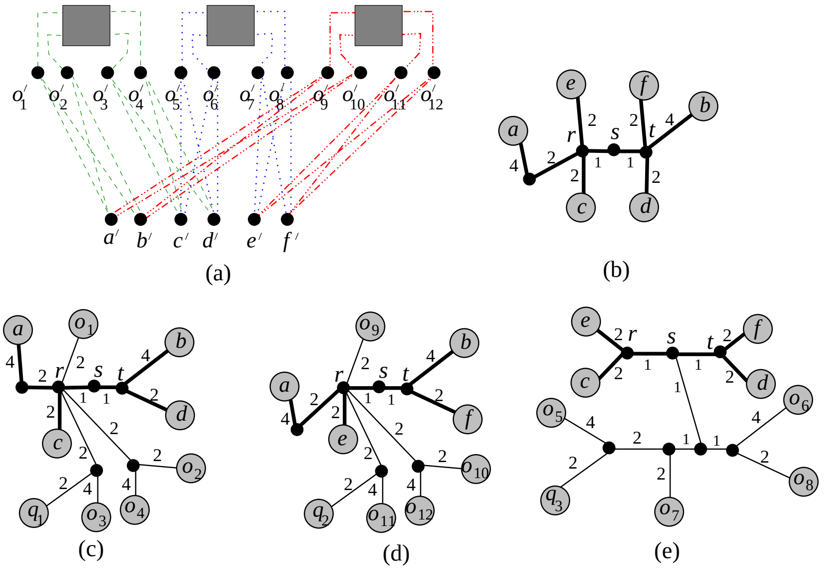

Figure 8: (a) A clause gadget G_{clause}. (b) Simplified representation of a pairwise compatibility tree T that determines the truth values of its literals. Here, $(a', b'), (c', d')$ and (e', f') correspond to the values true, false and false, respectively. (c)–(e) Subtrees of T that correspond to a G_{not} and its associated literal gadgets.

is true. The three pairs of vertices $(a', b'), (c', d')$, and (e', f') of G_{clause} play the role of the inputs. For each pair of inputs, e.g., $((a', b'), (c', d'))$, G_{clause} contains a G_{not} such that the ports o'_1, o'_2 of G_{not} form a cycle with a', b', and the ports o'_3, o'_4 of G_{not} form a cycle with c', d'. In the following we show that L is consistent if and only if at most one input is true.

Suppose for a contradiction that at least two of the three inputs, without loss of generality (a', b') and (c', d'), are true. Since (a', b') is true, by Lemma 3, (o'_1, o'_2) must be false. Consequently, (o'_3, o'_4) must be true. Since c', o'_3, d', o'_4 induce a cycle, by Lemma 3, (c', d') must be false, a contradiction.

Assume now that at most one of the three inputs is true. In this case, we show how to construct a pairwise compatibility tree such that the corresponding PCG G'_{clause} contains G_{clause} as a subgraph. Without loss of generality assume that (a', b') is true. (The construction when all the inputs are false is similar.) Construct an edge-weighted tree T as illustrated in Figure 8(b). Observe that $d_T(c, d) < d_{min}, d_T(e, f) < d_{min}$ and $d_T(a, b) > d_{max}$, which implies that $(c', d'), (e', f')$ are false and (a', b') is true. We call r, s, t the *medial path* of T. Figures 8(c)–(e) illustrate how to add the subtrees (shown in thin lines) that correspond to the instances of G_{not} to T. These trees not only realize the instances of G_{not}, but also determine the cycles that are incident to the inputs of the clause gadget.

6.3. Proof of Reduction

The following theorem uses the literal gadgets and clause gadgets to prove the NP-hardness result.

Theorem 4. MAX-GENERALIZED-PCG-RECOGNITION *is NP-hard.*

PROOF. Let $I(U, C)$ be an instance of MONOTONE-ONE-IN-THREE-3-SAT. We construct a corresponding instance $I(G, S, k)$ of MAX-GENERALIZED-PCG-RECOGNITION as follows. Assume that U consists of the literals $x_i, \ldots, x_t$ and C consists of the clauses $c_1, c_2, \ldots, c_{t'}$. For each clause c_j, $1 \leq j \leq t'$, we construct a copy of G_{clause}. If the same literal appears in more than two clauses, then we create a copy of that literal by cascading of NOT gates as illustrated in Figure 9(a). The resulting graph is the required graph G, which is straightforward to construct in polynomial time. The set S consists of the edges of $\overline{G_{not}}$s and the edges that are determined by the literal gadgets. Let N be the number of instances of G_{not} in G. Since each $\overline{G_{not}}$ has 101 non-adjacent pairs, S has $101N + t$ edges. We set $k = 97N + t'$. In the following we prove that $I(U, C)$ has an affirmative answer if and only if $I(G, S, k)$ has an affirmative answer.

We first assume that $I(U, C)$ has an affirmative answer, and then construct a PCG $G' = (T, d_{min}, d_{max})$ such that G' contains G as a subgraph, does not contain any edge of S, and at least k edges of S have distance greater than d_{max} between their corresponding leaves in T. For each clause c_j, we construct an edge-weighted tree $T(j)$ as described in the construction of clause gadget. Then for each index j from 1 to $t' - 1$, we merge the medial path $r_{j+1}, s_{j+1}, t_{j+1}$ of $T(j+1)$ with the medial path r, s, t of $T(j)$, as illustrated in Figures 9(b)–(d). Specifically, we merge the vertices $r_{j+1}, s_{j+1}, t_{j+1}$ with the vertices r, s, t, respectively, and then remove any resulting multi-edges or duplicate vertices. Finally, we complete the subtrees corresponding to the instances of G_{not} that we used for duplicating the input values, as illustrated in Figures 9(e)–(g).

Let the resulting tree be T. We prove that its corresponding PCG is the required PCG G'. Since we start with the edge-weighted trees for the basic gadgets, and the merging operations do not destroy any adjacency relationship, G' contains G as a subgraph. On the other hand, every redundant

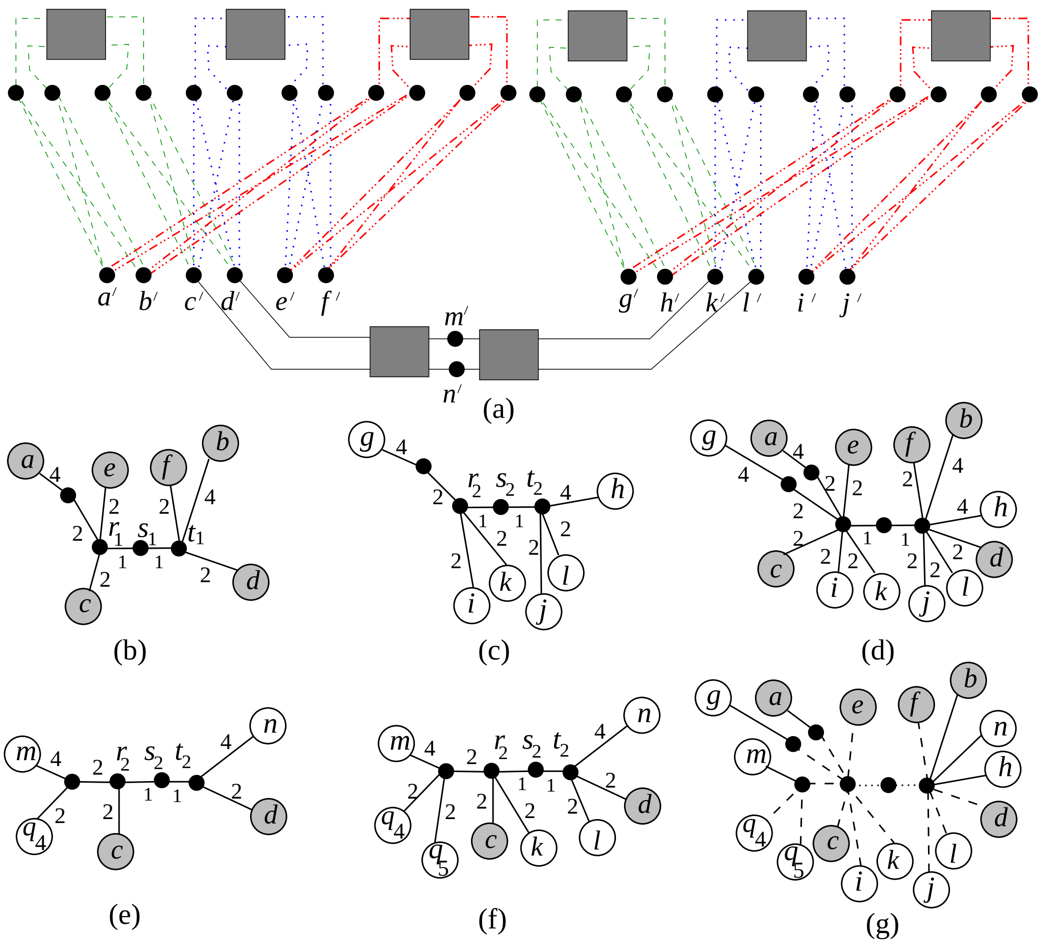

Figure 9: (a) The graph G that correspond to the instance $I(U, C) = (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_2 \vee x_5)$, where x_1, x_2, x_3, x_4, x_5 correspond to $(a', b'), (c', d'), (e', f'), (g', h'), (i', j')$, respectively. (b)–(c) Compatibility trees for the clauses, where the literals except x_1 and x_2 are false. (d) Merging the medial paths. (e)–(f) Compatibility trees for the instances of G_{not} that propagate the truth value from (c', d') to (k', l'). (g) A compatibility tree of G'. The edges with weights $1, 2$ and 4 are shown in dotted, dashed and solid lines, respectively.

edge in G' creates an adjacency between two different instances of G_{not}, or between two different literal gadgets, or between a G_{not} and a literal gadget. Since every edge in S is contained either in a single G_{not} or in a single literal gadget, G' does not contain any edge from S. We now need to verify that at least k edges of S have distance greater than d_{max} between their corresponding leaves in T. Recall that each $\overline{G_{not}}$ has exactly 97 such edges. Furthermore, every clause has exactly one true literal. Therefore, S has exactly $k = 97N + t'$ edges that have distance greater than d_{max} between their corresponding leaves in T.

We now assume that $I(U, C)$ does not have any affirmative answer, and then prove that in any PCG G' that contains G as a subgraph, must have less than $k = 97N + t'$ edges of S that have distance greater than d_{max} between their corresponding leaves in T. Since each $\overline{G_{not}}$ can have at most 97 such edges, at least t' edges that contribute to k must come from the literal gadgets. Since no two literal gadget that lie in the same clause can simultaneously have distance greater than d_{max} between their corresponding leaves in T, each clause must have at least one true literal. Therefore, we can construct satisfying truth assignment for U such that each clause in C contains exactly one true literal, which contradicts that $I(U, C)$ does not have any affirmative answer. □

7. Conclusion

We have constructed a nonplanar graph with eight vertices that is not a PCG. The graph we construct is not split matrogenic, leaving open the question of Calamoneri et al. [8] of whether every split matrogenic is a PCG. See [8] for the definition of a split matrogenic graph.

We also construct a planar graph that is not a PCG, but the graph is not outerplanar. Since every triangle-free outerplanar graph with degree at most three is a PCG [15], an interesting question is whether there exists any outerplanar graph that is not a PCG. Another important open problem that remains is to determine the complexity of the (original, or generalized) PCG recognition problem.

References

[1] Brandstädt, A., Hundt, C., Mancini, F., Wagner, P., 2010. Rooted directed path graphs are leaf powers. Discrete Mathematics 310 (4), 897–910.

[2] Brandstädt, A., Le, V. B., Rautenbach, D., 2010. Exact leaf powers. Theoretical Computer Science 411 (31-33), 2968–2977.

[3] Brandstädt, A., Wagner, P., 2010. Characterising (k, l)-leaf powers. Discrete Applied Mathematics 158 (2), 110–122.

[4] Calamoneri, T., Frascaria, D., Sinaimeri, B., 2013. All graphs with at most seven vertices are pairwise compatibility graphs. The Computer Journal 56 (7), 882–886, http://arxiv.org/abs/1202.4631.

[5] Calamoneri, T., Petreschi, R., 2014. On dilworth k graphs and their pairwise compatibility. In: Proceedings of the 8th International Workshop on Algorithms and Computation (WALCOM). Vol. 8344 of LNCS. Springer, pp. 213–224.

[6] Calamoneri, T., Petreschi, R., 2014. On pairwise compatibility graphs having dilworth number k. Theoretical Computer Science 547, 82–89.

[7] Calamoneri, T., Petreschi, R., 2014. On pairwise compatibility graphs having dilworth number two. Theoretical Compututer Science 524, 34–40.

[8] Calamoneri, T., Petreschi, R., Sinaimeri, B., 2012. On relaxing the constraints in pairwise compatibility graphs. In: Proceedings of the 6th International Workshop on Algorithms and Computation (WALCOM), Bangladesh. Vol. 7157 of LNCS. Springer, pp. 124–135.

[9] Durocher, S., Mondal, D., Rahman, M. S., 2013. On graphs that are not PCGs. In: Proceedings of the 7th International Workshop on Algorithms and Computation (WALCOM). Vol. 7748 of LNCS. Springer, pp. 310–321.

[10] Fellows, M. R., Meister, D., Rosamond, F. A., Sritharan, R., Telle, J. A., 2008. Leaf powers and their properties: Using the trees. In: Proceedings of the 19th International Symposium, on Algorithms and Computation (ISAAC). Vol. 5369 of LNCS. Springer, pp. 402–413.

[11] Kearney, P. E., Munro, J. I., Phillips, D., 2003. Efficient generation of uniform samples from phylogenetic trees. In: Proceedings of the 3rd International Workshop on Algorithms in Bioinformatics (WABI), Budapest, Hungary. Vol. 2812 of LNCS. Springer, pp. 177–189.

[12] Kennedy, W. S., Lin, G., Yan, G., 2006. Strictly chordal graphs are leaf powers. J. Discrete Algorithms 4 (4), 511–525.

[13] Nishimura, N., Ragde, P., Thilikos, D. M., 2002. On graph powers for leaf-labeled trees. Journal of Algorithms 42 (1), 69–108.

[14] Phillips, D., 2002. Uniform sampling from phylogenetic trees. Master's thesis, University of Waterloo, Canada.

[15] Salma, S. A., Rahman, M. S., Hossain, M. I., 2013. Triangle-free outerplanar 3-graphs are pairwise compatibility graphs. Journal of Graph Algorithms and Applications 17 (2), 81–102.

[16] Schaefer, T. J., 1978. The complexity of satisfiability problems. In: Proc. of Symposium on Theory of Computing (STOC 1978). pp. 216–226.

[17] Yanhaona, M. N., Bayzid, M. S., Rahman, M. S., 2010. Discovering pairwise compatibility graphs. Discrete Mathematics, Algorithms and Applications 2 (4), 607–624.

[18] Yanhaona, M. N., Hossain, K. S. M. T., Rahman, M. S., 2009. Pairwise compatibility graphs. Journal of Applied Mathematics and Computing (30), 479–503.