Discrete Mathematics, Algorithms and Applications
Vol. 2, No. 4 (2010) 607–623

DOI: 10.1142/S1793830910000917

DISCOVERING PAIRWISE COMPATIBILITY GRAPHS

MUHAMMAD NUR YANHAONA*, MD. SHAMSUZZOHA BAYZID† and MD. SAIDUR RAHMAN‡

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh
**nur.yanhaona@gmail.com*
†shams.bayzid@gmail.com
‡saidurrahman@cse.buet.ac.bd

Accepted 9 August 2010

Let T be an edge weighted tree, let $d_T(u, v)$ be the sum of the weights of the edges on the path from u to v in T, and let $d_{\min}$ and $d_{\max}$ be two non-negative real numbers such that $d_{\min} \leq d_{\max}$. Then a pairwise compatibility graph of T for $d_{\min}$ and $d_{\max}$ is a graph $G = (V, E)$, where each vertex $u' \in V$ corresponds to a leaf u of T and there is an edge $(u', v') \in E$ if and only if $d_{\min} \leq d_T(u, v) \leq d_{\max}$. A graph G 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 G is a pairwise compatibility graph of T for $d_{\min}$ and $d_{\max}$. Kearney *et al.* conjectured that every graph is a PCG [3]. In this paper, we refute the conjecture by showing that not all graphs are PCGs. Moreover, we recognize several classes of graphs as pairwise compatibility graphs. We identify two restricted classes of bipartite graphs as PCG. We also show that the well known tree power graphs and some of their extensions are PCGs.

Keywords: Pairwise compatibility graphs; phylogenetic tree; tree power; tree root.

Mathematics Subject Classification 2000: 05C78, 05C90, 68R05, 68R10

1. Introduction

Let T be an edge weighted tree and let $d_{\min}$ and $d_{\max}$ be two non-negative real numbers such that $d_{\min} \leq d_{\max}$. A *pairwise compatibility graph of* T *for* $d_{\min}$ *and* $d_{\max}$ is a graph $G = (V, E)$, where each vertex $u' \in V$ represents a leaf u of T and there is an edge $(u', v') \in E$ if and only if the distance between u and v in T lies within the range from $d_{\min}$ to $d_{\max}$. T is called the *pairwise compatibility tree* of G. We denote a pairwise compatibility graph of T for $d_{\min}$ and $d_{\max}$ by $PCG(T, d_{\min}, d_{\max})$. A graph G is 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 $G = PCG(T, d_{\min}, d_{\max})$. Figure 1(a) depicts an edge weighted tree T and Fig. 1(b) depicts a pairwise compatibility graph G of T for $d_{\min} = 4$ and $d_{\max} = 7$;

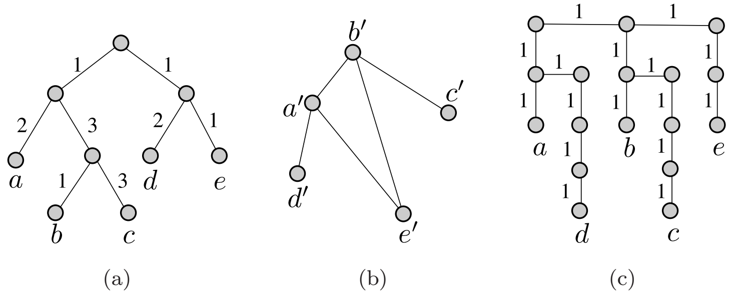

Fig. 1. (a) An edge weighted tree T_1, (b) a pairwise compatibility graph G, and (c) an edge weighted tree T_2.

there is an edge between a' and b' in G since in T the distance between a and b is six, but G does not contain the edge (a', c') since the distance between a and c in T is eight, which is larger than seven. It is quite apparent that a single edge weighted tree may have many pairwise compatibility graphs for different values of $d_{\min}$ and $d_{\max}$. Likewise, a single pairwise compatibility graph may have many trees of different topologies as its pairwise compatibility trees. For example, the graph in Fig. 1(b) is a *PCG* of the tree in Fig. 1(a) for $d_{\min} = 4$ and $d_{\max} = 7$, and it is also a *PCG* of the tree in Fig. 1(c) for $d_{\min} = 5$ and $d_{\max} = 6$.

In the realm of pairwise compatibility graphs, two fundamental problems are the tree construction problem and the pairwise compatibility graph recognition problem. Given a *PCG* G, the tree construction problem asks to construct an edge weighted tree T, such that G is a pairwise compatibility graph of T for suitable $d_{\min}$ and $d_{\max}$. The pairwise compatibility graph recognition problem seeks the answer whether or not a given graph is a *PCG*.

Pairwise compatibility graphs have their origin in *Phylogenetics*, which is a branch of computational biology that concerns with reconstructing evolutionary relationships among organisms [2, 4]. Phylogenetic relationships are commonly represented as trees known as the phylogenetic trees. From a problem of collecting leaf samples from large phylogenetic trees, Kearney *et al.* introduced the concept of pairwise compatibility graphs [3]. As their origin suggests, these graphs can be used in reconstruction of evolutionary relationships. However, their most intriguing potential lies in solving the "Clique Problem." A *clique* in a graph G is a set of pairwise adjacent vertices. The *clique problem* asks to determine whether a graph contains a clique of at least a given size k. It is a well known NP-complete problem. The corresponding optimization problem, the *maximum clique problem*, asks to find the largest clique in a graph [1]. Kearney *et al.* have shown that for a pairwise compatibility graph G, the clique problem is equivalent to a "leaf sampling problem" — which is solvable in polynomial time in any pairwise compatibility tree T of G [3].

Since their inception, pairwise compatibility graphs have raised several interesting problems, and hitherto most of these problems have remained unsolved.

Among the others, identifying different graph classes as pairwise compatibility graphs is an important concern. Although overlapping of pairwise compatibility graphs with many well known graph classes like chordal graphs and complete graphs is quite apparent; slight progresses have been made on establishing concrete relationships between pairwise compatibility graphs and other known graph classes. Phillips has shown that every graph of five vertices or less is a *PCG* [6] and Yanhaona *et al.* have shown that all cycles, cycles with a single chord, and cactus graphs are *PCG*s [9]. Seeing the exponentially increasing number of possible tree topologies for large graphs, the proponents of *PCG*s conceived that all undirected graphs are *PCG*s [3]. In this paper, we refute the conjecture by showing that not all graphs are *PCG*s. While proving that not all graphs are *PCG*s, we also prove that not even all bipartite graphs are *PCG*s. In this connection, we recognize two restricted classes of bipartite graphs as pairwise compatibility graphs.

Pairwise compatibility graphs have striking similarity, in their underlying concept, with the well studied graph roots and powers. A graph $G' = (V', E')$ is a *k-root* of a graph $G = (V, E)$ if $V' = V$ and there is an edge $(u, v) \in E$ if and only if the length of the shortest path from u to v in G' is at most k. G is called the *k-power* of G' [5]. A special case of graph power is the tree power, which requires G' to be a tree. Tree power graphs and their extensions (*Steiner k-power* graphs, *phylogenetic k-power* graphs, etc.) are by definition similar to pairwise compatibility graphs. However, the exact relationship of these graph classes with pairwise compatibility graphs was unknown. In this paper, we investigate the possibility of the existence of such a relationship, and show that tree power graphs and some of their extensions are in fact pairwise compatibility graphs. Such a relationship may serve the purpose of not only unifying related graph classes but also utilizing the method of tree constructions for one graph class in another.

The rest of the paper is organized as follows. Section 2 describes some of the definitions we have used in our paper, Sec. 3 shows that not all graphs are pairwise compatibility graphs. In Sec. 4 we establish two restricted classes of bipartite graphs as *PCG*s. Section 5 establishes a relationship of tree power graphs and their extensions with pairwise compatibility graphs. Finally, Sec. 5 concludes our paper with discussions. A primary version of this paper has been accepted for presentation at [8].

2. Preliminaries

In this section we define some terminologies that we have used in this paper.

Let $G = (V, E)$ be a simple graph with vertex set V and edge set E. The sets of vertices and edges of G are denoted by $V(G)$ and $E(G)$, respectively. An edge between two vertices u and v of G is denoted by (u, v). Two vertices u and v are *adjacent* and called *neighbors* if $(u, v) \in E$; the edge (u, v) is then said to be *incident* to vertices u and v. The *degree* of a vertex v in G is the number of edges incident to it. A *subgraph* of a graph $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and

$E' \subseteq E$; we then write $G' \subseteq G$. If G' contains all the edges of G that join two vertices in V' then G' is said to be the *subgraph induced by* V'. A *path* $P_{uv} = w_0, w_1, \ldots, w_n$ is a sequence of distinct vertices in V such that $u = w_0, v = w_n$ and $(w_{i-1}, w_i) \in E$ for every $1 \leq i \leq n$. A *subpath* of P_{uv} is a subsequence $P_{w_j w_k} = w_j, w_{j+1}, \ldots, w_k$ for some $0 \leq j < k \leq n$. A vertex x on P_{uv} is called an *internal node* of P_{uv} if $x \neq u, v$. G is *connected* if each pair of vertices of G belongs to a path, otherwise G is *disconnected*. A set S of vertices in G is called an *independent set* of G if the vertices in S are pairwise non-adjacent. A graph $G = (V, E)$ is a *bipartite graph* if V can be expressed as the union of two independent sets; each independent set is called a *partite set*. A *complete bipartite graph* is a bipartite graph where two vertices are adjacent if and only if they are in different partite sets. A *cycle* of G is a sequence of distinct vertices starting and ending at the same vertex such that two vertices are adjacent if they appear consecutively in the list.

A *tree* T is a connected graph with no cycle. Vertices of degree one in T are called *leaves*, and the rests are called *internal nodes*. A tree T is *weighted* if each edge is assigned a number as the weight of the edge. A *subtree induced by a set of leaves* of T is the minimal subtree of T which contains those leaves. Figure 2 illustrates a tree T with six leaves u, v, w, x, y and z, where the edges of the subtree of T induced by u, v and w are drawn by thick lines. We denote by T_{uvw} the subtree of a tree induced by three leaves u, v and w. One can observe that T_{uvw} has exactly one vertex of degree 3. We call the vertex of degree 3 in T_{uvw} the *core* of T_{uvw}. The vertex o is the core of T_{uvw} in Fig. 2. The *distance between two vertices* u *and* v in T, denoted by $d_T(u, v)$, is the sum of the weights of the edges on P_{uv}. In this paper we have considered only weighted trees. We use the convention that if an edge of a tree has no number assigned to it then its default weight is one. A *star* is a tree with exactly one internal node, and we call the internal node of a star the *base* of the star.

A graph $G = (V, E)$ is called a *phylogenetic k-power graph* if there exists a tree T such that each leaf of T corresponds to a vertex of G and an edge $(u, v) \in E$ if and only if $d_T(u, v) \leq k$, where k is a given proximity threshold. *Steiner k-power graphs* extend the notion of phylogenetic k-power graphs. For a Steiner k-power graph the corresponding tree may have some internal nodes as well as the leaves that correspond to the vertices of the graph. Both Steiner k-power graphs and

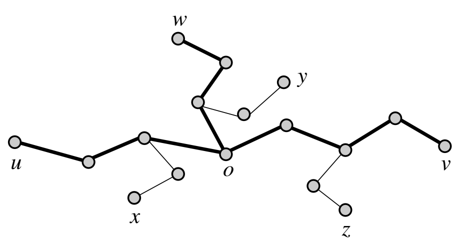

Fig. 2. Illustration for a leaf induced subtree.

phylogenetic k-power graphs belong to the widely known family of graph powers. Another special case of graph powers is the tree power graph. A graph $G = (V, E)$ is said to have a tree power for a certain proximity threshold k if a tree T can be constructed on V such that $(u, v) \in E$ if and only if $d_T(u, v) \leq k$.

3. Not all Graphs are PCGs

In this section, we show that not all graphs are pairwise compatibility graphs, as in the following theorem.

Theorem 3.1. *Not all graphs are pairwise compatibility graphs.*

To prove the claim of Theorem 3.1 we need the following lemmas.

Lemma 3.2. *Let T be an edge weighted tree, and u, v and w be three leaves of T such that P_{uv} is the largest 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 $d_T(w, x) \leq d_T(v, x)$.*

Proof. Let o be the core of T_{uvw}. Then each of the paths P_{uv}, P_{uw} and P_{wv} is composed of two of the three subpaths P_{uo}, P_{ow} and P_{ov}. Since $d_T(u, v)$ is the largest path in T_{uvw}, $d_T(u, v) \geq d_T(u, w)$. This implies that $d_T(u, o) + d_T(o, v) \geq d_T(u, o) + d_T(o, w)$. Hence $d_T(o, v) \geq d_T(o, w)$. Similarly, $d_T(u, o) \geq d_T(o, w)$ since $d_T(u, v) \geq d_T(w, v)$. Since T is a tree, there is a path from x to o. Let o_x be the first vertex in $V(T_{uvw}) \cap V(P_{xo})$ along the path P_{xo} from x. Then clearly o_x is on P_{uo}, P_{vo} or P_{wo}. We first assume that o_x is on P_{uo}, as illustrated in Fig. 3(a).

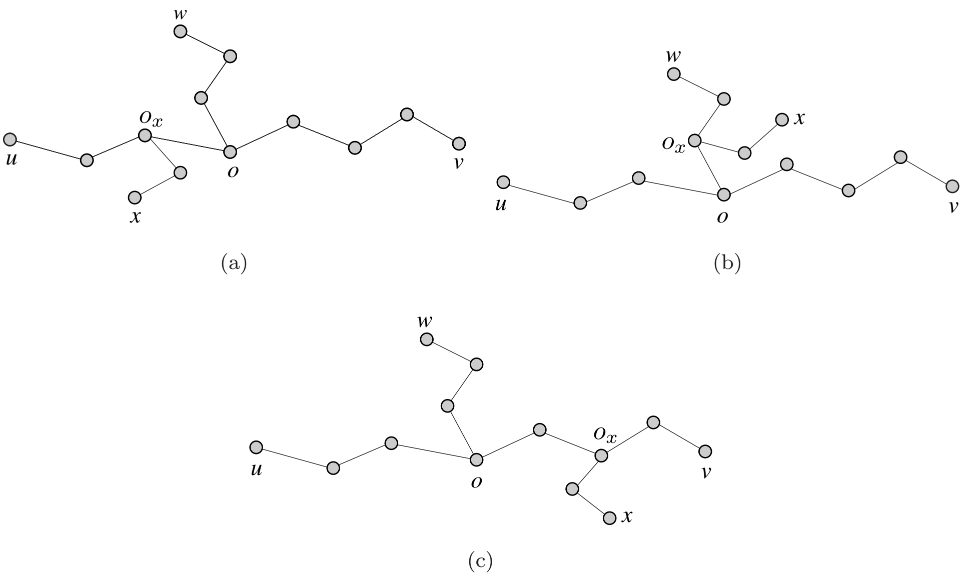

Fig. 3. Different positions of x.

Then $d_T(v,x) \geq d_T(w,x)$ since $d_T(w,x) = d_T(x,o)+d_T(w,o)$, $d_T(v,x) = d_T(x,o)+d_T(v,o)$ and $d_T(v,o) \geq d_T(w,o)$. We now assume that o_x is on P_{vo}, as illustrated in Fig. 3(c). Then $d_T(u,x) \geq d_T(w,x)$ since $d_T(w,x) = d_T(x,o)+d_T(w,o)$, $d_T(u,x) = d_T(x,o) + d_T(o,u)$ and $d_T(u,o) \geq d_T(w,o)$. We finally assume that o_x is on P_{wo}, as illustrated in Fig. 3(b). Then $d_T(u,x) = d_T(u,o) + d_T(o,o_x) + d_T(o_x,x)$ and $d_T(w,x) = d_T(w,o_x)+d_T(o_x,x)$. As $d_T(w,o_x) \leq d_T(w,o)$ and $d_T(u,o) \geq d_T(w,o)$, $d_T(u,x) \geq d_T(w,x)$. Likewise, $d_T(v,x) \geq d_T(w,x)$. Thus, in each case, at least one of u and v is at a distance from x that is either larger than or equals to the distance between w and x. □

Lemma 3.3. *Let $G = (V,E)$ be a $PCG(T, d_{\min}, d_{\max})$. Let a,b,c,d and e be five leaves of T and let a',b',c',d' and e' be five vertices of G corresponding to the five leaves a,b,c,d and e of T, respectively. Let P_{ae} be the largest path in the subtree of T induced by the leaves a,b,c,d and e, and P_{bd} be the largest path in T_{bcd}. Then G has no vertex x' such that x' is adjacent to a',c' and e' but not adjacent to b' and d'.*

Proof. Assume for a contradiction that G has a vertex x' such that x' is a neighbor of a',c' and e' but not of b' and d'. Let x be the leaves of T corresponding to the vertex x' of G. Since P_{ae} is the largest path in T among all the paths that connect a pair of leaves from the set $\{a,b,c,d,e\}$, $\max_{y\in\{a,e\}} d_T(x,y) \geq \max_{z\in\{b,c,d\}} d_T(x,z)$ by Lemma 3.2. Since both a and e are adjacent to x in G, $\max_{y\in\{a,e\}} d_T(x,y) \leq d_{\max}$. This implies that $d_T(x,y) \leq d_{\max}$, $y \in \{a,b,c,d,e\}$. Since P_{bd} is the largest path in T_{bcd}, $\max_{y\in\{b,d\}} d_T(x,y) \geq d_T(x,c)$ by Lemma 3.2. Without loss of generality assume that $d_T(x,b) \geq d_T(x,c)$. Since b' and x' are not adjacent in G and $d_T(x,b) \leq d_{\max}$, $d_T(x,b) < d_{\min}$. Then $d_T(x,c) < d_{\min}$ since $d_T(x,b) \geq d_T(x,c)$. Since $d_T(x,c) < d_{\min}$, c' cannot be adjacent to x' in G, a contradiction. □

Using Lemma 3.3 we now present a graph which is not a *PCG* as in the following lemma.

Lemma 3.4. *Let $G = (V,E)$ be a graph of 15 vertices, and let $\{V_1,V_2\}$ be a partition of the set V such that $|V_1| = 5$ and $|V_2| = 10$. Assume that each vertex in V_2 has exactly three neighbors in V_1 and no two vertices in V_2 has the same three neighbors in V_1. Then G is not a pairwise compatibility graph.*

Proof. Assume for a contradiction that G is a pairwise compatibility graph, i.e., $G = PCG(T, d_{\min}, d_{\max})$ for some T, $d_{\min}$ and $d_{\max}$. Let P_{uv} be the longest path in the subtree of T induced by the leaves of T representing the vertices in V_1. Clearly u and v are leaves of T. Let u' and v' be the vertices in V_1 corresponding to the leaves u and v of T, respectively. Let P_{wx} be the longest path in the subtree of T induced by the leaves of T corresponding to the vertices in $V_1 - \{u',v'\}$. Clearly w and x are also the leaves of T, and let w' and x' be the vertices in V_1 corresponding to w and x of T. Since $|V_1| = 5$, T has a leaf y corresponding to the vertex $y' \in V_1$ such that

$y' \notin \{u', v', w', x'\}$. Since G is a *PCG* of T, G cannot have a vertex adjacent to u', v' and y' but not adjacent to w' and x' by Lemma 3.3. However, for every combination of three vertices in V_1, V_2 has a vertex which is adjacent to only those three vertices of the combination. Thus there is indeed a vertex in V_2 which is adjacent to u', v' and y' but not to w' and x'. Hence G cannot be a pairwise compatibility graph of T by Lemma 3.3, a contradiction. □

Lemma 3.4 immediately proves Theorem 3.1. Figure 4 shows an example of a bipartite graph which is not a *PCG*. Quite interestingly, however, every complete bipartite graph is a *PCG*. It can be shown as follows. Let $K_{m,n}$ be a complete bipartite graph with two partite sets $X = \{x_1, x_2, x_3, \ldots, x_m\}$, and $Y = \{y_1, y_2, y_3, \ldots, y_n\}$. We construct a star for each partite set such that each leaf corresponds to a vertex of the respective partite set. Then we connect the bases of the stars through an edge as illustrated in Fig. 5. Finally, we assign one as the weight of each edge. Let T be the resulting tree. Now one can easily verify that $K_{m,n} = PCG(T, 3, 3)$.

Taking the graph described in Lemma 3.4 as a subgraph of a larger graph, we can show a larger class of graphs which is not *PCG*, as described in the following lemma.

Lemma 3.5. *Let $G = (V, E)$ be a graph, and let V_1 and V_2 be two disjoint subsets of vertices such that $|V_1| = 5$ and $|V_2| = 10$. Assume that each vertex in V_2 has*

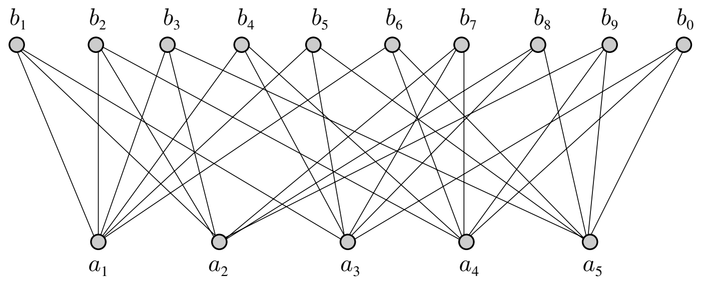

Fig. 4. Example of a graph which is not a *PCG*.

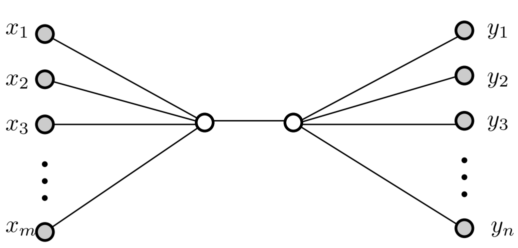

Fig. 5. Pairwise compatibility tree T of a complete bipartite graph $K_{m,n}$.

exactly three neighbors in V_1 *and no two vertices in* V_2 *has the same three neighbors in* V_1*. Then* G *is not a pairwise compatibility graph.*

Proof. Assume for a contradiction that G is PCG, i.e., $G = PCG(T, d_{\min}, d_{\max})$ for some T, $d_{\min}$ and $d_{\max}$. Let H be the subgraph of G induced by $V_1 \cup V_2$. Now, let T_H be the subtree of T induced by the leaves representing the vertices in $V_1 \cup V_2$. According to the definition of leaf induced subtree, for any pair of leaves u, v in T_H, $d_{T_H}(u, v) = d_T(u, v)$. Then $H = PCG(T_H, d_{\min}, d_{\max})$ since $G = PCG(T, d_{\min}, d_{\max})$. However, H is not a PCG by Lemma 3.4, a contradiction. □

4. Bipartite Graphs and *PCG*s

From Theorem 3.1, it is evident that not all bipartite graphs are PCGs (see Fig. 4). So it would be interesting to find some restricted classes of bipartite graphs that are PCGs. We have already seen in Sec. 3 that every complete bipartite graph is a PCG. We now show that two other subclasses of bipartite graphs are also PCGs as in Theorems 4.1 and 4.2.

Theorem 4.1. *Let* $G = (V, E)$ *be a bipartite graph with two partite sets* P *and* Q*. Let* $X \subset P$ *and* $Y \subset Q$ *such that there is no edge in* G *having one end in* X *and the other end in* Y*. Assume that the subgraph of* G *induced by* $(P - X) \cup Q$ *is a complete bipartite graph with partite sets* $P - X$ *and* Q*, and the subgraph of* G *induced by* $(Q - Y) \cup P$ *is a complete bipartite graph with partite sets* $Q - Y$ *and* P*. Then* G *is a* PCG*.*

Proof. We give a constructive proof. Let G be a bipartite graph as specified in Theorem 4.1 with $|P| = p$, $|Q| = q$, $|X| = r$, and $|Y| = s$. Let $p_1, p_2, \ldots, p_p$ be the p vertices in P and let $q_1, q_2, \ldots, q_q$ be the q vertices in Q. Without loss of generality we can assume that $p \geq q$. We first construct two caterpillars C_p and C_q corresponding to two partite sets P and Q such that each leaf of the C_p and C_q corresponds to a vertex of P and Q, respectively as follows. We make the path $u_1, u_2, \ldots, u_p$ as the spine of C_p and $u'_1, u'_2, \ldots, u'_p$ as the leaves of C_p such that u_i is adjacent to u'_i and $u'_p, u'_{p-1}, \ldots, u'_{p-r+1}$ are the r vertices in X. Similarly we make the path $v_1, v_2, \ldots, v_q$ as the spine of C_q and $v'_1, v'_2, \ldots, v'_q$ as the leaves of C_q such that v_i is adjacent to v'_i and $v'_q, u'_{q-1}, \ldots, v'_{q-s+1}$ are the s vertices in Y. Here u'_i and v'_i correspond to p_i and q_i, respectively. C_p and C_q are depicted in Fig. 6(a).

We next construct a single tree T by connecting C_p and C_q through an edge $u_p v_q$ as illustrated in Fig. 6(b). We finally assign the weight of the edges of T as follows. Let l_w be the weight of the edge $u_p v_q$. We assign $l_w = 2l$ where $l = \max\{p, q\}$. We assign weight one to each edge connecting a leaf of the caterpillar to its spine except for the edges incident to the leaves corresponding to the vertices in X and Y, i.e., weight of each edge $u_i u'_i$ and $v_j v'_j$ is one where $1 \leq i \leq p - r$ and $1 \leq j \leq q - s$. We assign $l + 1, l, l - 1, \ldots, l - r + 2$

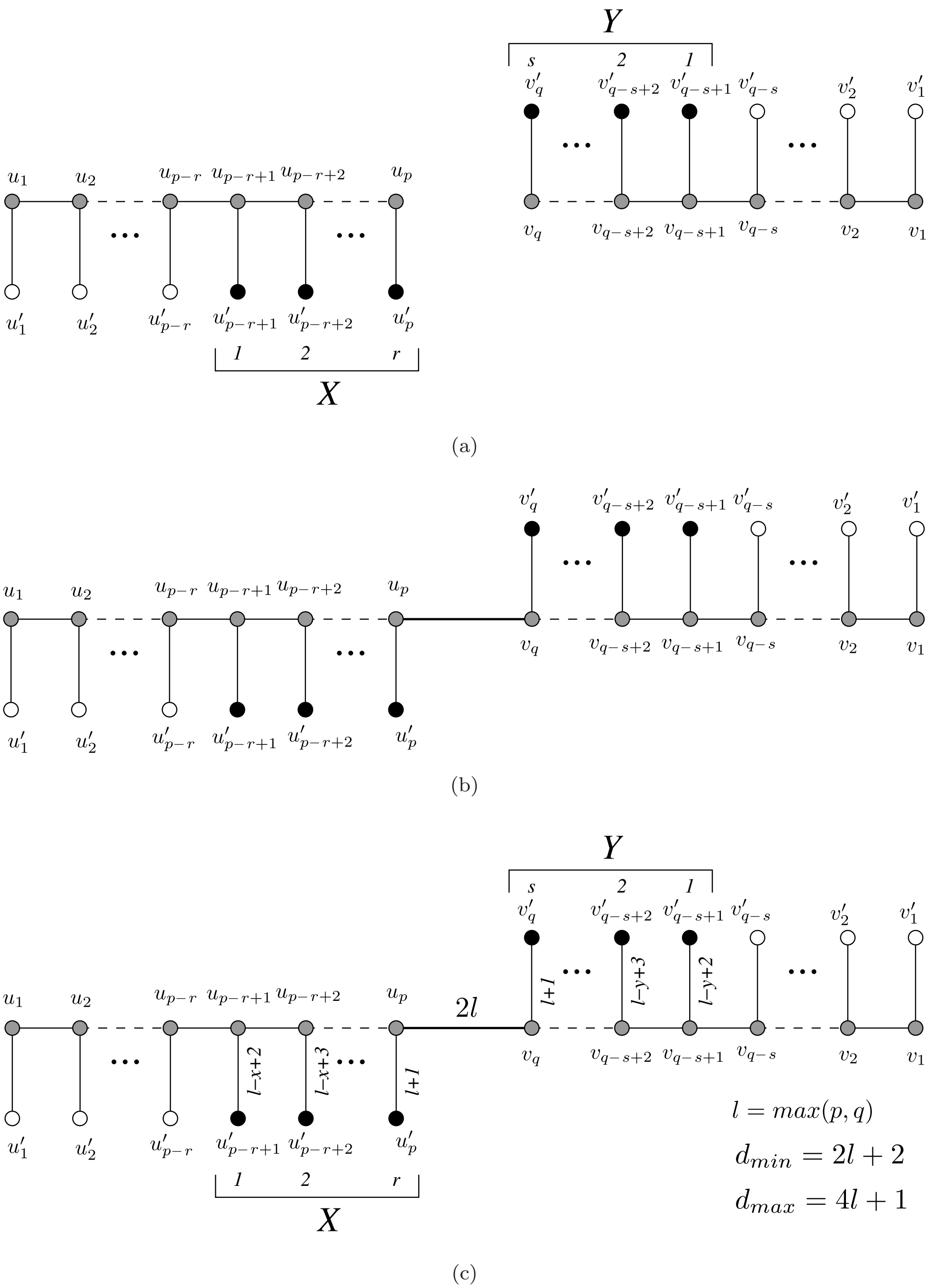

Fig. 6. (a) Two caterpillars C_p and C_q, (b) connecting C_p and C_q through the edge $u_p v_q$, and (c) the weight assignment.

as the weights of the edges $u_p u'_p, u_{p-1}u'_{p-1}, u_{p-2}u'_{p-2}, \ldots, u_{p-r+1}u'_{p-r+1}$ respectively. Similarly we assign $l+1, l, l-1, \ldots, l-s+2$ as the weights of the edges $v_q v'_q, v_{q-1}v'_{q-1}, v_{q-2}v'_{q-2}, \ldots, v_{q-s+1}v'_{q-s+1}$, respectively. This weight assignment is illustrated in Fig. 6(c).

We now show that T is a pairwise compatibility tree of G for $d_{\min} = 2l+2$ and $d_{\max} = 4l+1$. The distance between u'_1 and u'_p, and v'_1 and v'_q are $p+l+1$ and $q+l+1$, respectively (see Fig. 6(c)). Since $l = \max\{p,q\}$, the maximum possible distance between two leaves of the same caterpillar is $2l+1$. The distance between any two leaves of the same caterpillar should be out of the range defined by $d_{\min}$ and $d_{\max}$, and here we can see that $(2l+1) < d_{\min}$. Again the distance between any two leaves u and v, where $u \in X$ and $v \in Y$ is $(l+1)+(l+1)+2l = 4l+2$, which is greater than $d_{\max}$. The maximum possible distance between two leaves corresponding to two vertices that are adjacent in G is $l+2l+(l+1) = 4l+1$ (distance between u'_1 and v'_q where, in G, $p_1 \notin X$ and $q_q \in Y$), which is within the specified range from $d_{\min}$ to $d_{\max}$. Again the minimum possible distance between two leaves corresponding to two vertices that are adjacent in G is $1+2l+1 = 2l+2$ (distance between u'_p and v'_q while X and Y are empty), which is also within the specified range. Thus T is a pairwise compatibility tree of G for $d_{\min} = 2l+2$ and $d_{\max} = 4l+1$ and hence G is a *PCG*. □

Figure 7(a) illustrates an example of the bipartite graph as specified in Theorem 4.1, where $p=5$, $q=4$, $X=\{p_1,p_2,p_3\}$ and $Y=\{q_3,q_4\}$. Figure 7(b) illustrates the pairwise compatibility tree of the graph in Fig. 7(a) obtained by the construction in the proof of Theorem 4.1.

Theorem 4.2. *Let $G=(V,E)$ be a bipartite graph with two partite sets P and Q where $|P|=p$, $|Q|=q$ and $p \geq q$. Let $d_m = \max_{v \in P} deg(v)$. Then G is a PCG if the vertices in P can be labeled by $p_1, p_2, \ldots, p_p$ and the vertices in Q can be labeled by $q_1, q_2, \ldots, q_q$ such that the following condition holds for any vertex $u \in P$.*

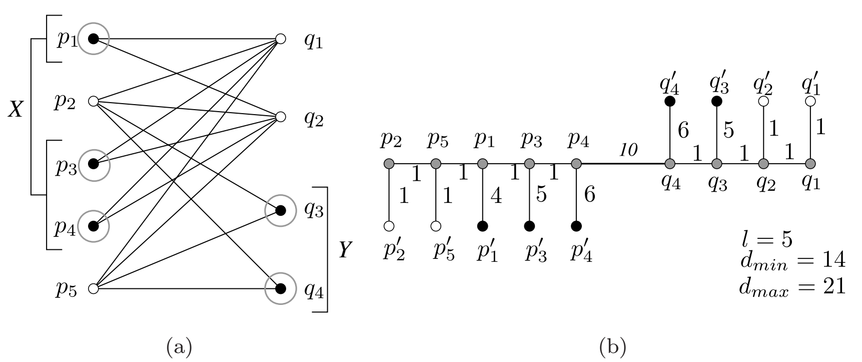

Fig. 7. (a) A bipartite graph G satisfying the conditions of Theorem 4.1, and (b) a pairwise compatibility tree of G.

(*con*) *If* $deg(u) = d_m$, *then the neighbors of* u *are* $q_i, q_{i+1}, \ldots, q_{i+deg(u)-1}$ *where* $1 \leq i \leq (q - deg(u) + 1)$, *otherwise, the neighbors of* u *are* $q_1, q_2, \ldots, q_{deg(u)}$ *or* $q_q, q_{q-1}, \ldots, q_{q-(deg(u)-1)}$.

Proof. We give a constructive proof. Let $G = (V, E)$ be a bipartite graph satisfying the conditions of Theorem 4.2. We now construct two caterpillars C_p and C_q corresponding to two partite sets P and Q such that each leaf of the C_p and C_q corresponds to a vertex of P and Q, respectively as follows. We make the path $u_1, u_2, \ldots, u_p$ as the spine of C_p and $u'_1, u'_2, \ldots, u'_p$ as the leaves of C_p such that u_i is adjacent to u'_i. Similarly we make the path $v_1, v_2, \ldots, v_q$ as the spine of C_q and $v'_1, v'_2, \ldots, v'_q$ as the leaves of C_q such that v_i is adjacent to v'_i. C_p and C_q are depicted in Fig. 8(a). Here u'_i and v'_i correspond to p_i and q_i, respectively. We now construct

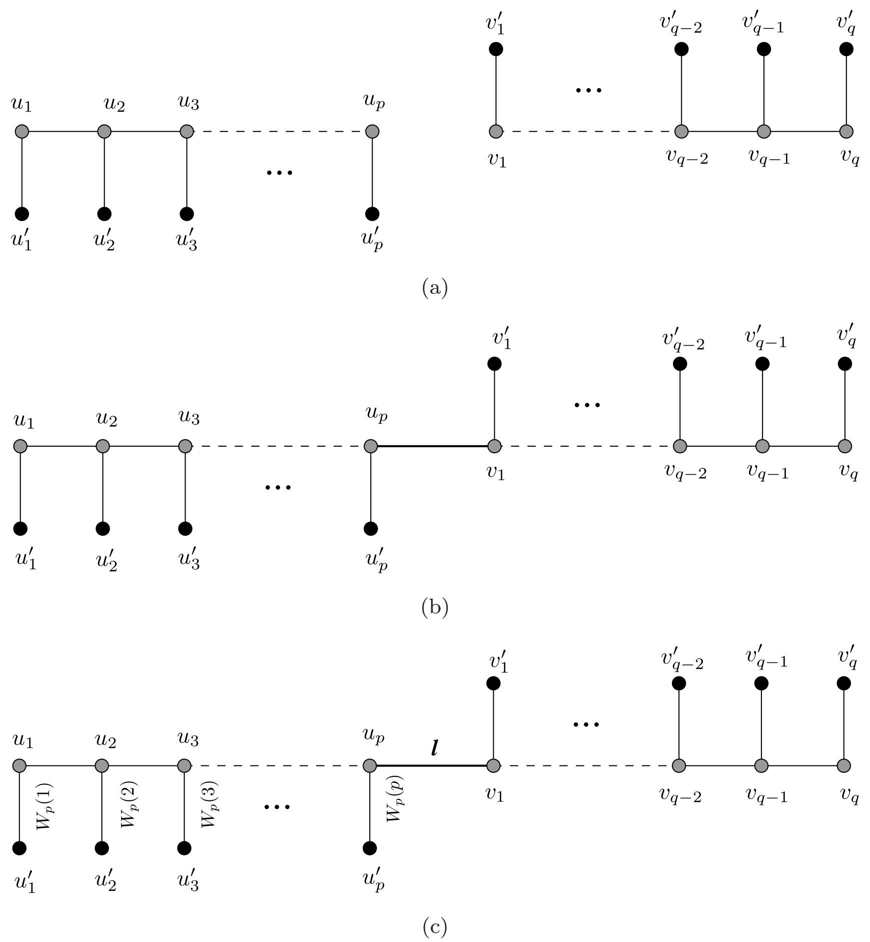

Fig. 8. (a) Two caterpillars C_p and C_q, (b) connecting C_p and C_q through the edge u_pv_1, and (c) the weight assignment.

a single tree T by connecting C_p and C_q through an edge u_pv_1 as illustrated in Fig. 8(b). We assign weight one to each edge of C_q. Let l, $W_p(i)$, be the weight of the edge u_pv_1, and the weight of the edge $u_iu'_i$, respectively. Let the neighbors of p_i in G be $q_j, q_{j+1}, \ldots, q_{j+deg(p_i)-1}$; then we define $N_{\text{skip}}(i)$ as $j-1$.

Let $d'_{\max} = p+q$, $d'_{\min} = d'_{\max} - (d_m - 1)$ and $l = 2(p+q-1) - d'_{\min} + 1$. We now take $d_{\min}$ as $d'_{\min} + l$, $d_{\max}$ as $d'_{\max} + l$, and take $W_p(i)$ as follows.

$$W_p(i) = \begin{cases} d'_{\max} - (p-i) - (N_{\text{skip}}(i) + deg(p_i)) & \text{if } deg(p_i) = d_m, \text{ or } deg(p_i) < d_m \\ & \quad \text{and its neighbors are } q_1, q_2, \ldots, \\ & \quad q_{deg(p_i)}; \\ d'_{\min} - (q - deg(p_i) + 1) - (p-i) & \text{if } deg(p_i) < d_m \text{ and its neighbors} \\ & \quad \text{are } q_q, q_{q-1}, \ldots, q_{q-(deg(p_i)-1)}. \end{cases} \tag{4.1}$$

We have defined $W_p(i)$ in Eq. (4.1) in such a way that if $deg(p_i) = d_m$ and its neighbors are $q_j, q_{j+1}, \ldots, q_{j+deg(p_i)-1}$ then the distances from u'_i to v'_j, and $v'_{j+deg(p_i)-1}$ are equal to $d_{\min}$ and $d_{\max}$, respectively. This implies that the distance from u'_i to v'_t, where $p_iq_t \notin E$ is either less than $d_{\min}$ or greater than $d_{\max}$. Next, if $deg(p_i) < d_m$ and its neighbors are $q_1, q_2, \ldots, q_{deg(p_i)}$ then the distance from u'_i to $v'_{deg(p_i)}$ is $d_{\max}$. In this case, the distance from u'_i to v'_t, where $p_iq_t \notin E$ is greater than $d_{\max}$. Finally, if $deg(p_i) < d_m$ and its neighbors are $q_q, q_{q-1}, \ldots, q_{q-(deg(p_i)-1)}$, then the distance from u_i to $v'_{q-(deg(p_i)-1)}$ is $d_{\min}$. Thus the distance from u'_i to v'_t where $p_iq_t \notin E$ is less than $d_{\min}$. Therefore, the distance between two leaves is within the range defined by $d_{\min}$ and $d_{\max}$ if and only if their corresponding vertices in G are adjacent. Again, $d_{\min}$ must be greater than the maximum possible distance between the two leaves of C_p; and the weight of l should be chosen accordingly so that the distance between the two leaves corresponding to two adjacent vertices of G is greater than the maximum possible distance between the two leaves of C_p. The maximum possible distance between the two leaves of C_p is the distance between leaves u_1 and u_p when $W_p(1)$ and $W_p(p)$ get their maximum possible weight. Again, they get their maximum weight when p_1 and p_p are connected only to the first vertex q_1 in the Q-partite set. In this situation $W_p(1) = p + q - (1-1) - (p-1+1) = q$ and $W_p(p) = p + q - (1-1) - (p-p+1) = p+q-1$ using the formula as stated in Eq. (4.1). Now the maximum possible distance between two vertices of the same caterpillar is equal to the maximum distance between u_1 and u_p = (maximum weight of leaf u_1) + (maximum weight of leaf u_p) + $(p-1) = q + (p+q-1) + (p-1) = 2(p+q-1)$. Then $d_{\min}$ must be greater than $2(p+q-1)$ and we assign $d_{\min} = 2(p+q-1)+1$. Now $l = d_{\min} - d'_{\min} = 2(p+q-1) - d'_{\min} + 1$. Therefore, T is a pairwise compatibility tree of G. □

Figure 9(a) illustrates an example of the bipartite graph as specified in Theorem 4.2. In this example, $p = 5$, $q = 4$ and $d_m = 3$. Note that $deg(p_1) = deg(p_3) = d_m = 3$ and each of the other vertices in set P has degree less than three.

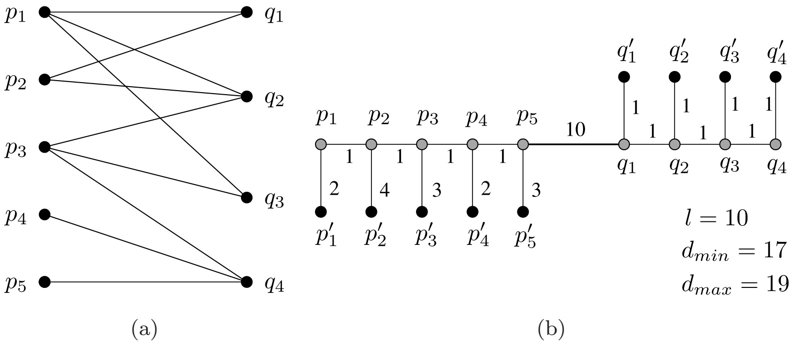

Fig. 9. (a) A bipartite graph G satisfying the conditions of Theorem 4.2, and (b) a pairwise compatibility tree of G.

Here p_1 and p_3 are connected to three vertices of set Q, and their neighbors are $\{q_1, q_2, q_3\}$ and $\{q_2, q_3, q_4\}$, respectively. On the other hand, $deg(p_2) = 2$ which is less than d_m and its neighbors are q_1 and q_2. Vertices p_4 and p_5 have degree one and their neighbor is q_4. Here $N_{\text{skip}}(1) = 0$, $N_{\text{skip}}(3) = 1$ and $N_{\text{skip}}(4) = 3$. Figure 9(b) illustrates the pairwise compatibility tree T of the graph given in Fig. 9(a) obtained by the construction in the proof of Theorem 4.2. One can easily verify that T is the pairwise compatibility tree of G.

The recognition of this graph class does not look so trivial. However, it can be recognized whether or not a graph belongs to this graph class by brute force method. In this approach, we need to consider every possible labeling of the given bipartite graph G, and there are $p!q!$ such labelings. For each labeling the recognition can be done in $O(p)$ amount of time.

5. Variants of Tree Power Graphs and *PCG*s

In this section we will show that tree power graphs and two of their extensions are *PCG*s.

Tree power graphs and their extensions (Steiner k-power and phylogenetic k-power graphs) have striking resemblance, in their underlying concept, with *PCG*s. But does this similarity signify any real relationship? It does indeed: we find that tree power graphs and these two extensions are essentially *PCG*s. To establish this relationship of aforementioned three graph classes with pairwise compatibility graphs, we introduce a generalized graph class which we call “tree compatible graphs.” A graph $G = (V, E)$ is a *tree compatible graph* if there exists a tree T such that all leaves and a subset of internal nodes of T correspond to the vertex set V of G, and for any two vertices $u, v \in V$; $(u, v) \in E$ if and only if $k_{\min} \le d_T(u, v) \le k_{\max}$. Here $k_{\min}$ and $k_{\max}$ are real numbers. We call G the *tree compatible graph* of T for $k_{\min}$ and $k_{\max}$. It is quite evident from this definition that tree compatible graph comprises tree power graphs, Steiner k-power graphs, and phylogenetic k-power graphs. We now have the following theorem.

Theorem 5.1. *Every tree compatible graph is a pairwise compatibility graph.*

Proof. Let G be a tree compatible graph of a tree T for non-negative real numbers $k_{\min}$ and $k_{\max}$. Then to prove the claim, it is sufficient to construct a tree T' and find two non-negative real numbers $d_{\min}$ and $d_{\max}$ such that $G = PCG(T', d_{\min}, d_{\max})$.

Clearly $G = PCG(T', d_{\min}, d_{\max})$ for $T' = T$, $d_{\min} = k_{\min}$ and $d_{\max} = k_{\max}$ if every vertex in V corresponds to a leaf in T. We thus assume that V contains a vertex which corresponds to an internal node of T. In this case we construct a tree T' from T as follows. For every internal node u of T that corresponds to a vertex in V, we introduce a surrogate internal node u'. In addition, we transform u into a leaf node by connecting u through an edge of weight λ with u'. Figure 10 illustrates this transformation. Here, in addition to the leaves of T, two internal nodes d and e correspond to the vertices in V. T' is the modified tree after transforming d and e into leaf nodes by replacing them by d' and e', respectively.

The aforementioned transformation transmutes the subset of internal nodes of T that participates in V into a subset of leaves in T'. Let u and v be two arbitrary nodes in T. If u and v are both leaves in T then $d_{T'}(u, v) = d_T(u, v)$. If both u and v are internal nodes of T that are contributing to V then in T' they are two leaf nodes, and $d_{T'}(u, v) = d_T(u, v) + 2\lambda$. Finally, if only one of u and v is transformed to leaf then $d_{T'}(u, v) = d_T(u, v) + \lambda$. We next define $d_{\min} = k_{\min}$ and $d_{\max} = k_{\max} + 2\lambda$. Since every vertex $u \in V$ is represented as a leaf in T', T' may be a pairwise compatibility tree of G. We will prove that T' is indeed a pairwise compatibility tree by showing that $G = PCG(T', d_{\min}, d_{\max})$ for an appropriate value of λ. Note that we cannot simply assign $\lambda = 0$ because, in the context of root finding as well as phylogenetics, an edge of zero weight is not meaningful. For example, if an evolutionary tree contains zero weighted edges then we may find a path of length zero between two different organisms, which is clearly unacceptable. Therefore, we have to choose a value for λ more intelligently.

According to the definition of tree compatible graphs, for every pair of vertices $u, v \in V$, $(u, v) \in E$ if and only if $k_{\min} \leq d_T(u, v) \leq k_{\max}$. Meanwhile, we have

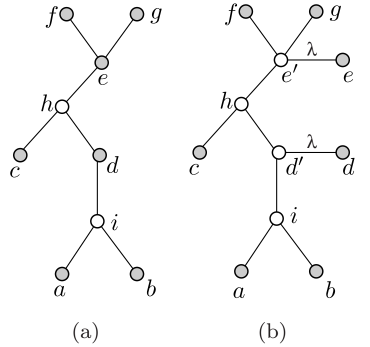

Fig. 10. (a) T and (b) T'.

derived T' from T in such a way that either the distance between u and v in T' remains the same as in T, or increased by at most 2λ. Therefore, if we can prove that $d_{\min} \leq d_{T'}(u,v) \leq d_{\max}$ if and only if $k_{\min} \leq d_T(u,v) \leq k_{\max}$ then it will imply that $G = PCG(T', d_{\min}, d_{\max})$. Depending on the nature of the change in the distance between u and v from T to T', we have to consider three different cases.

Case 1: $d_{T'}(u,v) = d_T(u,v)$.

In this case, three possible relationships can exist among $d_T(u,v)$, $k_{\min}$ and $k_{\max}$. First, if $d_T(u,v) < k_{\min}$ then $d_{T'}(u,v) < d_{\min}$ since $d_{\min} = k_{\min}$. Next, if $k_{\min} \leq d_T(u,v) \leq k_{\max}$ then $k_{\min} \leq d_T(u,v) \leq k_{\max} + 2\lambda$. That implies, $d_{\min} \leq d_T(u,v) \leq d_{\max}$. Finally, let $d_T(u,v) > k_{\max}$. Suppose p is the minimum difference between $k_{\max}$ and the length of a path in T that is longer than $k_{\max}$, that is $p = \min_{u,v \in V} \{d_T(u,v) - k_{\max}\}$. Then $d_T(u,v) - k_{\max} \geq p$. By subtracting 2λ from both side of the inequality we get, $d_T(u,v) - k_{\max} - 2\lambda \geq p - 2\lambda$. Which implies $d_{T'}(u,v) - d_{\max} \geq p - 2\lambda$. Therefore, if we can ensure that $p > 2\lambda$ then $d_{T'}(u,v)$ will be larger than $d_{\max}$.

Case 2: $d_{T'}(u,v) = d_T(u,v) + 2\lambda$.

In this case, we have to consider three scenarios as we have in case 1. First, if $k_{\min} \leq d_T(u,v) \leq k_{\max}$ then $k_{\min} \leq d_T(u,v) + 2\lambda \leq k_{\max} + 2\lambda$. Which implies $d_{\min} \leq d_T(u,v) + 2\lambda \leq d_{\max}$. Hence $d_{\min} \leq d_{T'}(u,v) \leq d_{\max}$. Next, if $d_T(u,v) > k_{\max}$ then adding 2λ in both sides we get $d_T(u,v) + 2\lambda > k_{\max} + 2\lambda$. That implies $d_{T'}(u,v) > d_{\max}$. Finally, let assume that $d_T(u,v) < k_{\min}$. Suppose q is the minimum difference between $k_{\min}$ and the length of a path in T that is smaller than $k_{\min}$; that is $q = \min_{u,v \in V} \{k_{\min} - d_T(u,v)\}$. Then $k_{\min} - d_T(u,v) \geq q$. Subtracting 2λ from both sides of the inequality we get $k_{\min} - d_T(u,v) - 2\lambda \geq q - 2\lambda$. Which implies $d_{\min} - d_{T'}(u,v) \geq q - 2\lambda$. Therefore, if we can ensure that $q > 2\lambda$ then $d_{T'}(u,v) < d_{\min}$.

Case 3: $d_{T'}(u,v) = d_T(u,v) + \lambda$.

This case is similar to case 2. By following the same reasoning as in case 2, we can show that $d_{\min} \leq d_{T'}(u,v) \leq d_{\max}$ if and only if $k_{\min} \leq d_T(u,v) \leq k_{\max}$, provided $q \geq \lambda$. If we can satisfy the inequality derived from case 2 ($q > 2\lambda$) then the inequality $q > \lambda$ will be immediately satisfied.

From our analysis of the three cases above, it is evident that if we can satisfy the two inequalities $p > 2\lambda$ and $q > 2\lambda$ simultaneously then $G = PCG(T', d_{\min}, d_{\max})$. We can do this by assigning λ any value smaller than $min(p,q)/2$. Thus T' is a pairwise compatibility tree of G, and hence G is a PCG. □

Figure 11(a) illustrates an example of a tree compatible graph $G = (V, E)$ and the corresponding tree T is depicted in Fig. 11(b). Here $k_{\min} = 2$, $k_{\max} = 4$, and the weight of every edge is one . Two internal nodes d and e along with the leaves of T correspond to the vertices in V of G. We now transfer T into T' according to the procedure described in Theorem 5.1. Figure 11(c) illustrates this transformation. Here, $p = q = 1$ and hence we can chose any positive value less than 0.5 for λ.

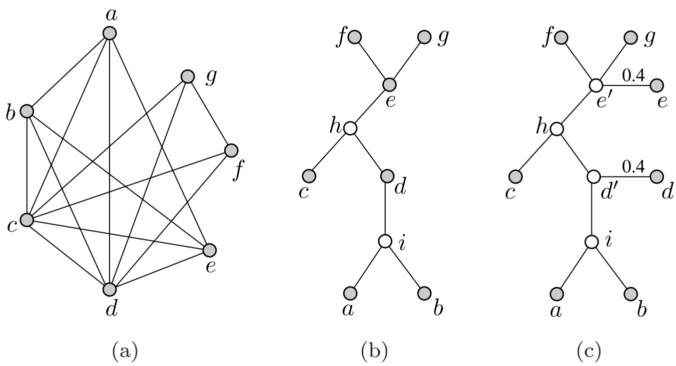

Fig. 11. (a) A tree compatible graph, (b) the corresponding tree T, and (c) the corresponding pairwise compatibility tree T'.

Let $\lambda = 0.4$ and then, $d_{\min} = k_{\min} = 2$ and $d_{\max} = k_{\max} + 2\lambda = 4.8$. One can now easily verify that $G = PCG(T', 2, 4.8)$.

6. Conclusion

In this paper, we have proved that not all graphs are *PCG*s. Additionally, we have proved that tree power graphs and two of their extensions are *PCG*s. We have also identified two restricted classes of bipartite graphs as *PCG*s. Our first proof establishes a necessary condition over the adjacency relationships that a graph must satisfy to be a *PCG*. However, a complete characterization of *PCG*s is not known. We left it as a future work. It would be quite challenging and significant to develop efficient algorithms for solving pairwise tree construction problem for other classes of graphs. Such algorithms may come handy in both clique finding and evolutionary relationships modeling contexts.

Acknowledgement

This work is done in Graph Drawing & Information Visualization Laboratory of the Department of CSE, BUET established under the project "Facility Upgradation for Sustainable Research on Graph Drawing & Information Visualization" supported by the Ministry of Science and Information & Communication Technology, Government of Bangladesh.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms* (MIT Press, Cambridge, 2001).
[2] N. C. Jones and P. A. Pevzner, *An Introduction to Bioinformatics Algorithms* (MIT Press, Cambridge, 2004).
[3] P. Kearney, J. I. Munro and D. Phillips, Efficient generation of uniform samples from phylogenetic trees, in *Proc. of WABI 2003*, Vol. 2812, LNBI (Springer, 2003), pp. 177–189.

[4] A. M. Lesk, *Introduction to Bioinformatics* (Oxford University Press, Great Clarendon Street, Oxford, 2002).

[5] G. H. Lin, P. E. Kearney and T. Jiang, Phylogenetic k-root and Steiner k-root, in *Proc. of International Conference on Algorithms and Computation (ISAAC 2000)*, Vol. 1969, LNCS (Springer, 2000), pp. 539–551.

[6] D. Phillips, Uniform sampling from phylogenetic trees, Master's thesis, University of Waterloo, August (2002).

[7] M. Pardalos and J. Xue, The maximum clique problem, *J. Global Optim.* **4** (1994) 301–328.

[8] M. N. Yanhaona, M. S. Bayzid and M. S. Rahman, Discovering Pairwise compatibility graphs, in *Proc. 16th International Computing and Combinatorics Conference (COCOON 2010)*, Vol. 6196, Lecture Notes in Computer Science, (2010), pp. 399–408.

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