Multi-interval Pairwise Compatibility Graphs
(Extended Abstract)

Shareef Ahmed and Md. Saidur Rahman

Graph Drawing and Visualization Laboratory,
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology
`shareef.tamal@gmail.com,saidurrahman@cse.buet.ac.bd`

Abstract. Let T be an edge weighted tree and let d_{min}, d_{max} be two non-negative real numbers where $d_{min} \leq d_{max}$. A pairwise compatibility graph (PCG) of T for d_{min}, d_{max} 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 lies within the interval $[d_{min}, d_{max}]$. A graph G is a PCG if there exist an edge weighted tree T and suitable d_{min},d_{max} such that G is a PCG of T. Knowing that all graphs are not PCGs, in this paper we introduce a variant of pairwise compatibility graphs which we call multi-interval PCGs. A graph G is a multi-interval PCG if there exist an edge weighted tree T and some mutually exclusive intervals of nonnegative real numbers such that there is an edge between two vertices in G if and only if the distance between their corresponding leaves in T lies within any such intervals. If the number of intervals is k, then we call the graph a k-interval PCG. We show that every graph is a k-interval pairwise compatibility graph for some k. We also prove that wheel graphs and a restricted subclass of series-parallel graphs are 2-interval PCGs.

Keywords: Pairwise compatibility graphs, Phylogenetic trees, Series-parallel graphs

1 Introduction

Let T be an edge weighted tree and let d_{min}, d_{max} be two non-negative real numbers where $d_{min} \leq d_{max}$. A *pairwise compatibility graph* (PCG) of T for d_{min} and d_{max} is a graph $G = (V, E)$ where 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 lies within the interval $[d_{min}, d_{max}]$. The tree T is called a *pairwise compatibility tree* (PCT) of G. We denote a pairwise compatibility graph T for d_{min}, d_{max} by PCG (T,d_{min},d_{max}). A given graph is a PCG if there exist suitable T, d_{min}, d_{max} such that G is a PCG of T. Figure 1(b) illustrates a pairwise compatibility graph G of the edge weighted tree in Fig. 1(a) T for d_{min} =3 and d_{max} =5. For a pairwise compatibility graph G, pairwise compatibility tree T may not be unique. For example, Fig. 1(c) shows another pairwise compatibility tree of the graph G in Fig. 1(b) for the same d_{min} and d_{max}.

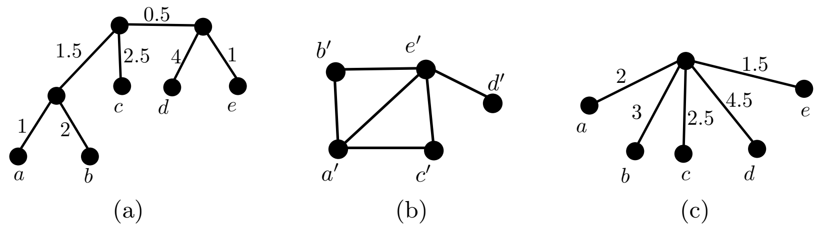


Fig. 1. (a) An edge weighted tree T, (b) a pairwise compatibility graph G of T for $d_{min} = 3$ and $d_{max} = 5$ and (c) another pairwise compaitibility tree of G.

PCGs have their application in modeling evolutionary relationship among set of organisms from biological data which is also called phylogeny. Phylogenetic relationships are normally represented as a tree called phylogenetic tree. While dealing with a sampling problem from large phylogenetic tree, Kearney *et al.* [9] introduced the concept of PCGs. They also showed that "the clique problem" can be solved in polynomial time for a PCG if a pairwise compatibility tree can be constructed in polynomial time.

Kearney *et al.* [9] conjectured that all graphs are PCGs, but later Yanhaona *et al.* [12] refuted the conjecture showing a bipartite graph with fifteen vertices is not a PCG. Later Calamoneri *et al.* proved that every graph with at most seven vertices is a PCG [4]. It is also known that the graphs having cycles as their maximum biconnected components, tree power graphs, Steiner k-power graphs, phylogenetic k-power graphs, some restricted subclasses of bipartite graphs, triangle-free maximum-degree-three outer planar graphs and some superclass of threshold graphs are PCGs [13], [12], [11], [6]. Calamoneri *et al.* gave some sufficient conditions for split matrogenic graph to be a PCG [5]. Recently a graph with eight vertices and a planar graph with sixteen vertices is proved not to be PCGs [7]. Iqbal *et al.* showed a necessary condition and a sufficent condition for a graph to be PCG [8]. Howerver, the complete characterization of PCGs is not known yet.

As not all graphs are PCGs, some researchers has tried to relax constraint on PCGs and thus some variants of PCGs are introduced [3], [5]. One such variant of PCG is improper PCG which allows multiple leaves corresponding to a vertex of a graph [3]. In this paper we introduce a new variant of PCGs which we call k-interval PCGs. The idea behind a k-interval PCG is to allow k mutually exclusive intervals of nonnegative real numbers instead of one. A graph G is a *k-interval PCG* of an edge weighted tree T for mutually exclusive intervals $I_1, I_2, \cdots, I_k$ of nonnegative real numbers where each vertex in G corresponds to a leaf in T and there is an edge between two vertices in G if the distance between their corresponding leaves lies in $I_1 \cup I_2 \cup \cdots I_k$. Figure 2(a) illustrates an edge weighted tree T and Fig. 2(b) shows the corresponding 2-interval PCG where $I_1 = [1, 3]$ and $I_2 = [5, 6]$.

In this paper we show that all graphs are k-interval PCGs for some k. We also show that wheel graphs W_n, which are not yet proved to be PCGs for $n \geq 8$

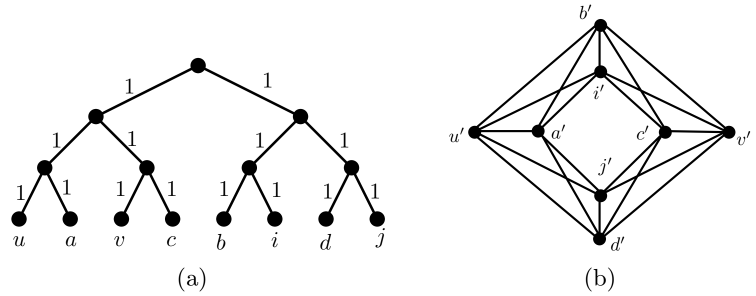


Fig. 2. (a) An edge weighted tree T, (b) a 2-interval PCG G of T where $I_1 = [1, 3]$,$I_2 = [5, 6]$.

are 2-interval PCGs. Moreover, we proved that a restricted subclass of series-parallel graphs are 2-interval PCGs and provide an algorithm for constructing 2-interval pairwise compatibility tree for graphs of this subclass.

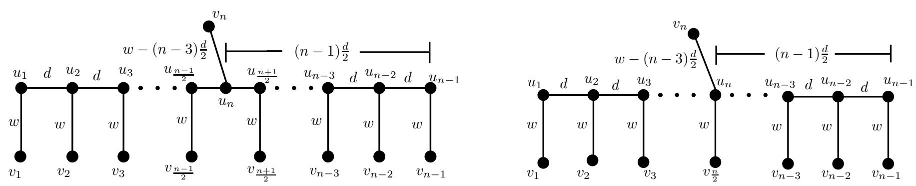
The remainder of the paper is organized as follows. Section 2 gives some necessary definitions, previous results and preliminary results on k-interval PCGs. In Sect. 3 we give our results on 2-interval PCGs. Finally we conclude in Sect. 4.

2 Preliminaries

In this section we define some terms which will be used throughout this paper and present some preliminery results.

Let, $G = (V, E)$ be a simple, undirected graph with vertex set V and edge set E. An edge between two vertices u and v is denoted by (u, v). If $(u, v) \in E$, then u and v are *adjacent* and the edge (u, v) is incident to u and v. The *degree* of a vertex is the number of edges incident to it. A *path* P_{uv} in G is a sequence of distinct vertices $w_1, w_2, w_3, \cdots, w_n$ in V such that $u = w_1$ and $v = w_n$ and $(w_i, w_{i+1}) \in E$ for $1 \leq i < n$. The vertices u and v are called *end-vertices* of path P_{uv}. If the end-vertices are same then the path is called a *cycle*. A *tree* T is a graph with no cycle. A vertex with degree one in a tree is called *leaf* of the tree. All the vertices other than leaves are called *internal nodes*. An *weighted tree* is a tree where each edge is assigned a number as the weight of the edge. The weight of an edge (u, v) is denoted as $w(u, v)$. The distance between two nodes u, v in T is the sum of the weights of the edges on path P_{uv} and denoted by $d_T(u, v)$. A star graph S_n is a tree on n nodes with one node having degree $n - 1$ and all other nodes having degree 1. A *caterpillar* is a tree for which deletion of leaves together with their incident edges produces a path. The *spine* of a caterpillar is the longest path to which all other vertices of the caterpillar are adjacent. A *wheel graph* with n vertices, denoted by W_n, is obtained from a cycle graph C_{n-1} with $n - 1$ vertices by adding a new vertex p and joining an edge from p to each vertex of C_{n-1}. The vertex p is called *hub*. A graph $G = (V, E)$ is called a *series-parallel* (SP) graph with source s and sink t if either G consists of a pair of vertices connected by a single edge or there exists two series-parallel graphs $G_i(V_i, E_i)$ with source s_i and sink t_i for $i = 1, 2$ such that $V = V_1 \cup V_2$, $E = E_1 \cup E_2$ and either $s = s_1$, $t_1 = s_2$ and $t = t_2$ or $s = s_1 = s_2$ and $t = t_1 = t_2$ [10].

We now review a previous result on cycles [13], [11] and show a construction process of a pairwise compatibility tree of a cycle which will be used later in this paper. Let C_n be a cycle with n vertices $v'_1, v'_2, v'_3 \cdots, v'_n$ where (v'_i, v'_{i+1}) are adjacent for $1 \leq i < n$ and (v'_1, v'_n) are also adjacent. We construct an edge weighted caterpillar T as follows. Let $v_1, v_2, v_3 \cdots, v_{n-1}$ be the leaves of T and $u_1, u_2, u_3, \cdots, u_{n-1}$ be the vertices on the spine of T such that u_i is adjacent to v_i for $1 \leq i < n$. We assign weight d to edge (u_i, u_{i+1}) for $1 \leq i < n-1$ and weight w to the edges incident to a leaf where $w > (n+1)\frac{d}{2}$. If n is odd then we put a vertex u_n in the middle of the path $P_{u_1 u_{n-1}}$ as illustrated in Fig. 3(a). If n is even then we use $u_{\frac{n}{2}}$ as u_n which is shown in Fig. 3(b). Then we place the last vertex v_n as a leaf adjacent to u_n. We assign weight $w - (n-3)\frac{d}{2}$ to the edge (u_n, v_n). This concludes the construction of T and we call this construction process **Algorithm ConstructCyclePCT**. The leaf v_i of T corresponds to the vertex v'_i of C_n. The constructed tree in this way is a PCT of C_n for $d_{min} = 2w + d$ and $d_{max} = 2w + d$. It is easy to observe that $max\{d_T(v_i, v_j)\} = 2w + (n-1)d$.



(a) (b)

Fig. 3. (a) A pairwise compatibility tree of a cycle with odd number of vertices and (b) a pairwise compatibility tree of a cycle with even number of vertices.

We now introduce a new concept called k-interval PCG. Let T be an edge weighted tree and $I_1, I_2, I_3, \cdots, I_k$ be k non-negative intervals such that $I_i \cap I_j = \emptyset$ for $i \neq j$. A *k-interval PCG* of T for $I_1, I_2, I_2, \cdots, I_k$ is a graph $G = (V, E)$ where each vertex $u' \in V$ represent a leaf u in T and there is an edge $(u', v') \in E$ if and only if $d_T(u, v) \in I_1 \cup I_2 \cup I_3 \cup \cdots \cup I_k$. Obviously, a PCG is a k-interval PCG for $k = 1$, but a k-interval PCG may not be a PCG. The graph shown in Fig. 2 is not a PCG [7] but a 2-interval PCG.

The following theorem describes a preliminary result on k-interval PCGs.

Theorem 1. *Every graph is an $|E|$-interval PCG.*

Outline of the Proof: We give a constructive proof. Let $G = (V, E)$ be a graph with n vertices $v'_1, v'_2, v'_3, \cdots, v'_n$. We construct a star T with n leaves v_1, v_2, v_3,

$\cdots, v_n$ where v_i corresponds to v'_i of G as illustrated in Fig. 4. Let $w(i)$ be the weight of the edge incident to v_i in T. We take $w(i)$ as follows.

$$w(i) = \begin{cases} 1 & \text{if } i = 1 \\ 2 & \text{if } i = 2 \\ w(i-1) + w(i-2) & \text{if } i > 2 \end{cases}$$

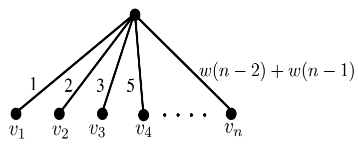


Fig. 4. An $|E|$-interval pairwise compatibility tree for any graph with n vertices.

For each edge (v_i, v_j) in E we take an interval $I_{ij} = [d_T(v_i, v_j), d_T(v_i, v_j)]$. Thus we have total $|E|$ number of intervals. Then for every edge $(v_i, v_j) \in E$, $d_T(v_i, v_j) \in I_{ij}$. Similarly, if $(v_i, v_j) \notin E$, then there is no such interval I_{ij} such that $d_T(v_i, v_j) \in I_{ij}$. Thus T is an $|E|$-interval PCT of G. □

3 2-interval PCGs

In this section we give some results on 2-interval PCGs.

3.1 Wheel graphs

In this section we prove that wheel graphs are 2-interval PCGs as in the following theorem.

Theorem 2. *Every wheel graph is a 2-interval PCG.*

Proof. Let W_{n+1} be a wheel graph with $n+1$ vertices $v'_1, v'_2, v'_3 \cdots, v'_n, p'$ where p' is the hub and $v'_1, v'_2, v'_3 \cdots, v'_n$ forms the outer cycle C. We first construct a pairwise compatibility tree T for C by **Algorithm ConstructCyclePCT**. Note that the maximum distance between any pair of leaves in T is $2w + (n-1)d$. We then place a vertex p representing the vertex p' in W_{n+1} such that it is adjacent to u_n in T and assign weight w_p to the edge (p, u_n) as illustrated in Fig. 5. We choose w_p such that $w_p > 2w + (n-1)d$.

Clearly $d_T(p, v_i) > 2w + (n-1)d = max\{v_i, v_j\}$ for $i, j \leq n$. Then T is a 2-interval pairwise compatibility tree of W_n for $I_1 = [2w + d, 2w + d]$ and $I_2 = (2w + (n-1)d, \infty)$. □

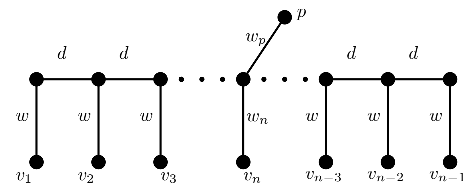


Fig. 5. A 2-interval pairwise compatibility tree of W_n.

3.2 Series-parallel graphs

In this section we define a restricted subclass of series-parallel graphs which we call SQQ series-parallel graphs and show that this class of graphs are 2-interval PCGs.

Let $G = (V, E)$ be a series-parallel graph with source s and sink t. A pair of vertices $\{u, v\}$ of a connected graph is a *split pair* if there exist two subgraphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ satisfying following two conditions: 1. $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{u, v\}$; and 2. $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$, $|E_1| \geq 1$, $|E_2| \geq 1$. The *SPQ-tree* $\mathcal{T}$ of a series-parallel graph G with respect to a reference edge (u, v) describes a recursive decomposition of G induced by its split pairs [2, 1]. Figure 6(a) illustrates a series-parallel graph G and Fig. 6(b) shows the SPQ-tree of G with respect to s, t. $\mathcal{T}$ is a rooted ordered tree and it contains three types of nodes: S, P and Q. Subtrees rooted at each node x of $\mathcal{T}$ corresponds to a subgraph of G called its *pertinent graph* $G(x)$. In this paper we use a modified definition of $G(x)$: $G(x)$ contains the leftmost and rightmost children of x in $\mathcal{T}$ in order from source to sink if x is a P-node or Q-node; if x is an S-node $G(x)$ does not contain the leftmost and rightmost children. Figure 6(c) illustrates the pertinent graph of the P-node at height 2 in $\mathcal{T}$. Let x be an S-node in $\mathcal{T}$ other than the root and let $y_1, y_2, y_3, \cdots, y_n$ be the children of x in order from source to sink. If both y_1 and y_n are Q-nodes then we call G an *SQQ* series-parallel graph. We now give the following theorem.

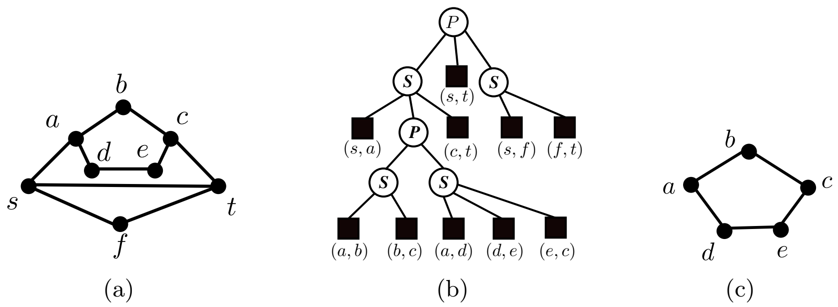


Fig. 6. (a) A series-parallel graph G, (b) An SPQ-tree of G with respect to s and t and (c) pertinent graph of the non-root P-node.

Theorem 3. *Every SQQ series-parallel graph is a 2-interval PCG.*

Proof. We give a constructive proof. Let $G = (V, E)$ be an SQQ series-parallel graph with source s' and sink t' and $\mathcal{T}$ be an SPQ-tree of G with respect to s' and t'. Note that if $\mathcal{T}$ consists of a single Q-node then G is trivially a 2-interval PCG. We thus assume that $\mathcal{T}$ has at least one S-node or P-node. We construct a 2-interval pairwise compatibility tree of G using a bottom up computation on $\mathcal{T}$. For each internal node x of $\mathcal{T}$ we first compute 2-interval PCT for each of it's child node and then we add additional component and combine them to get a 2-interval PCT T_x of G(x). Let s'_x and t'_x be the source and sink of G(x) and s_x, t_x be the leaves of T_x representing s'_x and t'_x respectively. Depending on the type of the current node we have to consider two cases.

Case 1: *The current node x is an S-node.* Let $y_1, y_2, y_3, \cdots, y_n$ be the children of x in order from s'_x to t'_x. This is illustrated in Fig. 8(a). According to the property of an SQQ series-parallel graph y_1 and y_n are Q-nodes. If $n = 2$, then we have only one node between s'_x and t'_x in G. In this case we construct a tree T_x with two leaves and one edge between them. One of the two leaves of T_x represents the only node between s'_x and t'_x. We assign weight $w + \frac{d}{2}$ to that edge. This is illustrated in Fig. 7(a),(b). If x is root node then we also place two leaves representing s'_x, t'_x and make them adjacent to a leaf in T_x. We then assign weight $w + \frac{d}{2}$ to the newly added edges.

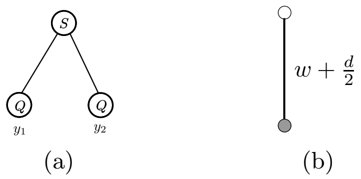


Fig. 7. (a) An S-node x with 2 children and (b) constructed tree T_x for x.

We now consider the case where $n > 2$. In this case we have two subcases.

Case 1(a): *y_i be a Q-node.* At first we consider y_i for $i \neq 1, n$. In this case we construct a caterpillar Γ_{y_i} with two leaves s_{y_i}, t_{y_i} and two internal nodes u_{y_i}, v_{y_i} where u_{y_i}, v_{y_i} are adjacent to s_{y_i}, t_{y_i} respectively. Here s_{y_i}, t_{y_i} represent s'_{y_i}, t'_{y_i} of G respectively. Let g_x be an indicator variable which is 1 if depth of x modulo 4 is equal to 0 or 1 in $\mathcal{T}$ and -1 otherwise. We now assign weight $w - g_x l$ to each edge incident to a leaf and weight $d + 2g_x l$ to the edge (u_{y_i}, v_{y_i}) where $l << d$ at least as small as $\frac{d}{100|V|}$ as is illustrated in Fig. 8(b). Then for $i = 1, n$ we also construct trees in the way mentioned above if x is the root node of $\mathcal{T}$, otherwise trees will be constructed for y_1 and y_n while processing the parent P-node of x.

Case 1(b): *y_i be a P-node.* In this case we have a caterpillar Γ_{y_i} induced by two leaves s_{y_i} and t_{y_i} of T_{y_i} according to the construction process described in case 2 as shown in Fig. 8(c). Let u_{y_i}, v_{y_i} be the vertices on spine of Γ_{y_i} that are adjacent to the leaves s_{y_i} and t_{y_i}.

We thus have a caterpillar Γ_{y_i} for each $i \neq 1, n$. We next merge all this caterpillars such that t_{y_i} and v_{y_i} lie on $s_{y_{i+1}}$ and $u_{y_{i+1}}$ and get a single caterpillar Γ_x with $n-1$ leaves induced by $s_2, s_3, \cdots, s_n$ as illustrated in Fig. 8(d).

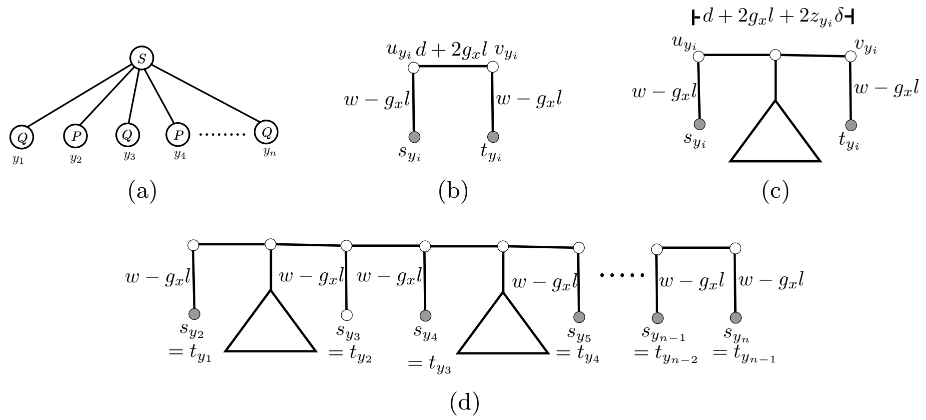


Fig. 8. (a) An S-node with more than 2 children, (b) constructed tree Γ_{y_i} for a child Q-node y_i, (c) constructed tree T_{y_i} for a child P-node y_i and (d) merged tree T_x for S node x.

Case 2: *The current node x is a P-node.* In this case x can have at most one Q-node as its child and if it has one then it represents an (s'_x, t'_x) edge. We first construct a caterpillar Γ_x with two leaves s_x,t_x representing s'_x and t'_x, and two internal nodes u_x, v_x where u_x is adjacent to s_x and v_x is adjacent to t_x. We now assign weight $w + g_{y_i}l$ to each edge incident to a leaf in Γ_x where g_{y_i} is the indicator variable of any child S-node y_i of x in $\mathcal{T}$. If x has a child Q-node in $\mathcal{T}$ we assign weight $d-2g_{y_i}l$ to the edge (u_x, v_x), otherwise we assign $d-2g_{y_i}l+2\delta$ where $\delta << l$. We now replace the edge (u_x, v_x) by a path u_x, a_x, b_x, v_x where a_x and b_x are two degree 2 vertices. We call a_x, b_x the *port nodes* of Γ_x. Then we reassign weight such that $w(u_x, a_x) = \frac{1}{2}d_{T_x}(u_x, v_x)$ and $w(a_x, b_x) = \delta$. Let z_x be an indicator variable which is 1 if there is a child Q-node of x and 0 otherwise. Then $w(u_x, a_x) = \frac{d}{2} - g_{y_i}l + z_x\delta$, $w(a_x, b_x) = \delta$ and $w(b_x, v_x) = \frac{d}{2} - g_{y_i}l + (z_x - 1)\delta$. See Fig. 9(a).

Let $y_1, y_2, y_3, \cdots, y_n$ be the children of x where y_i is an S-nodes for $1 \leq i \leq n$. At first we construct 2-interval PCT T_{y_i} of $G(y_i)$ for $1 \leq i \leq n$ according to case 1. Let y_i be an S-node with n_i children where $n_i > 2$. Then we have a caterpillar Γ_{y_i} with $n_i - 1$ leaves induced by the sources and sinks of some children of y_i in T_{y_i}, which is merged while processing the S-node according to case 1. Let $u_{i1}, u_{i2}, u_{i3}, \cdots, u_{i(n_i-1)}$ be the leaves of Γ_{y_i} and let $u'_{i1}, u'_{i2}, u'_{i3} \cdots, u'_{i(n_i-1)}$ be the vertices on spine where u_{ij} is adjacent to u'_{ij} for $1 \leq j \leq n$. Note that any edge (u_{ij}, u'_{ij}) has weight w and $(u'_{ij}, u'_{i(j+1)})$ has weight $d+2g_{y_i}l$ or $d+2g_{y_i}l+2\delta$. Let m_i be the number of edges of weight $d+2g_{y_i}l+2\delta$ on the spine where $m_i \le (n_i-2)$. Thus the spine has length of $(n_i-2)(d+2g_{y_i}l)+2m_i\delta$. We now put a vertex v_i on the spine such that $d_{\Gamma_i}(u'_{i1}, v_i) = \frac{(n_i-2)(d+2g_{y_i}l)+2m_i\delta}{2} - \delta$ and we add an edge between v_i and port node b_x. We assign weight $c-(n_i-1)\frac{d}{2} - (n_i-3)g_{y_i}l - (m_i+z_x)\delta$ to the edge (v_i, b) as illustrated in Fig. 9(b). We choose a very large value for c such that $c > 2(d+2\delta+2l)|V|$ where $|V|$ is the number of vertices in G.

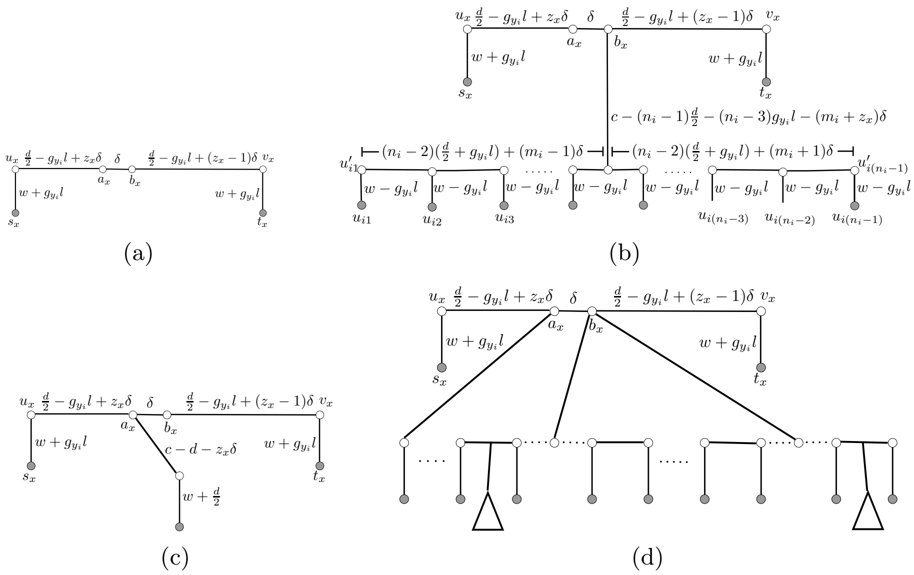


Fig. 9. (a) Constructed tree Γ_x with souce and sink for P-node, (b) merged tree with 2-interval PCT of a children S-node having than 2 children, (c) merged tree with 2-interval PCT of a children S-node having 2 children and (d) final 2-interval PCT T_x of G(x) where x is a P-node.

Let y_j be an S-node with exactly 2 children. Then we have 2-interval PCT T_{y_j} consists of two nodes and the edge between them has weight $w+\frac{d}{2}$. In this case we add an edge between port node a_x and one of the leaves. We assign weight $c-d-z_x\delta$ to the newly added edge as illustrated in Fig. 9(c). We call any edge joining Γ_x with T_{y_i} for $i \le n$ a *caterpillar-connecting edge.* An example of the construction process is illustrated in Fig. 10.

We now proof that the tree T constructed by above algorithm is a 2-interval PCT of G for intervals $I_1 = [2w+d, 2w+d]$ and $I_2 = [c+2w, c+2w]$. We prove this by an induction on the height $h(\mathcal{T})$ of the SPQ-tree $\mathcal{T}$ of G. Let x be the root of $\mathcal{T}$ having n children $y_1, y_2, \cdots, y_n$ and n_i be the number of children of y_i.

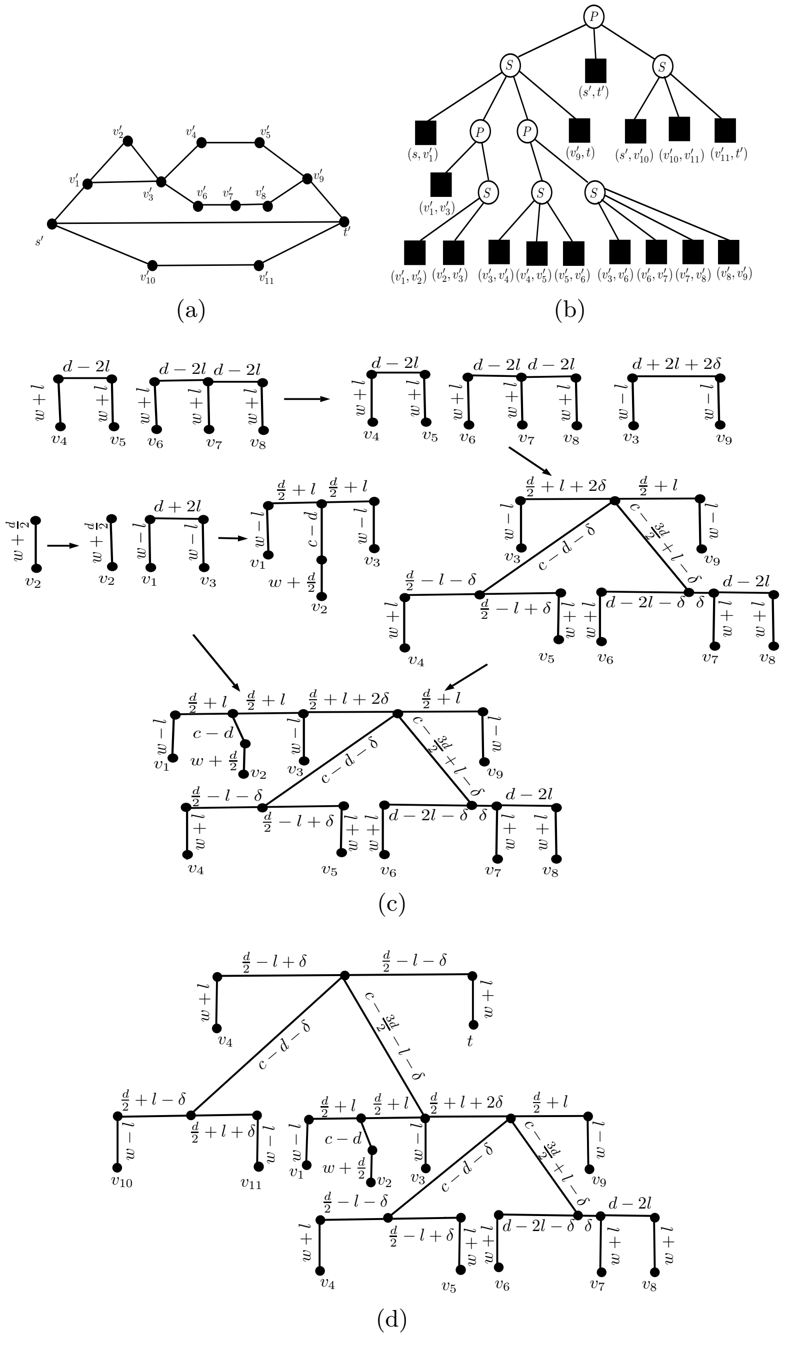


Fig. 10. (a) An SQQ series-parallel graph G, (b) an SPQ-tree of G (c) construction of 2-interval PCT of pertinant graph of the leftmost child of the root which is an S-node and (d) constructed 2-interval PCT of G.

Assume that G is an SQQ series-parallel graph with $h(\mathcal{T})$ =1. Then $\mathcal{T}$ consists of an S-node x as its root and all the children of the root are Q-nodes. In this case the algorithm produces a caterpillar with n leaves where each edge incident to a leaf has weight $w - g_x l$ and each edge on the spine has weight $d + 2g_x l$. Thus if (u, v) is a Q-node in $\mathcal{T}$ then $d_T(u, v) = 2w + d$ and otherwise $2w + d < d_T(u, v) < c + 2w$ because of our choice of c being very large. Thus the basis is true.

Assume that $h(\mathcal{T})$ >1 and the claim is true for every SQQ series-parallel graph with $h(\mathcal{T}) < h$. Let G be an SQQ series-parallel graph with $h(\mathcal{T}) = h$ and let x be the root of $\mathcal{T}$. Let $y_1, y_2, y_3, \cdots, y_n$ be the children of x and pertinent graphs of $y_1, y_2, \cdots, y_n$ are 2-interval PCGs for I_1 and I_2 by the induction hypothesis. Let $T_{y_1}, T_{y_2}, \cdots, T_{y_n}$ be the 2-interval PCTs constructed by the algorithm for $y_1, y_2, \cdots, y_n$.

We first consider the case where x is a P-node. Then according to case 2 we have $d_T(s_x, t_x) = 2w + d$, if there is an edge (s_x, t_x). Otherwise, we have $2w + d < d_T(s_x, t_x) = 2w + d + 2\delta < c + 2w$. Let y_i be an S-node. If y_i has 2 children, then there is only one node u'_{i1} between s_x and t_x in G(y_i) and u_{i1} is its corresponding leaf in T_{y_i}. In this case $d_T(s_x, u_{i1}) = d_T(t_x, u_{i2}) = 2w + c$ which lies in interval I_2. If y_i has more than 2 children then the distance $d_T(s_x, u_{i1})$ is computed as follows.

$$\begin{aligned} d_T(s_x, u_{i1}) &= d_{\Gamma_x}(s_x, b) + w(b, v_i) + d_{\Gamma_{y_i}}(v_i, u_{i1}) \\ &= w + g_{y_i} l + \frac{d}{2} - g_{y_i} l + z_x\delta + \delta + c - (n_i - 1)\frac{d}{2} - (n_i - 3) g_{y_i} l \\ &\quad -(m_i + z_x)\delta + w - g_{y_i} l + (n_i - 2)(\frac{d}{2} + g_{y_i} l) + m_i\delta - \delta \\ &= 2w + c \end{aligned}$$

Similarly the distance $d_T(s_x, u_{n_i})$ is computed as follows.

$$\begin{aligned} d_T(s_x, u_{in_i}) &= d_{\Gamma_x}(s_x, b) + W(b, v_i) + d_{\Gamma_i}(v_i, u_{in}) \\ &= w + g_{y_i} l + \frac{d}{2} - g_{y_i} l + z_x\delta + \delta + c - (n_i - 1)\frac{d}{2} - (n_i - 3) g_{y_i} l \\ &\quad -(m_i + z_x)\delta + w - g_{y_i} l + (n_i - 2)(\frac{d}{2} + g_{y_i} l) + m_i\delta + \delta \\ &= 2w + c + 2\delta \end{aligned}$$

Thus $d_T(s_x, u_{in_i}) > c + 2w$. Now clearly $d_T(s_x, u_{ij}) < 2w + c$ for $j \neq 1, n_i$; as they are at least $d + 2g_{y_i} l$ less than $2w + c + 2\delta$. Again $d_T(s_x, u_{ij}) > 2w + d$ because we choose $c > 2(d + 2\delta + 2l)|V|$. Doing similar calculation for t_x we get, $d_T(t_x, u_{in_i}) = 2w + c$, $d_T(t_x, u_{i1}) = 2w + c - 2\delta$ and $2w + d < d_T(t_x, u_{ij}) < 2w + c$ for $j \neq 1, n_i$. Now the path from u_{ij} and u_{kl} where $i \neq k$ consists of 2 caterpillar-connecting edge, 2 edge from leaf to spine for each leaf and some additional edges on the spines. Thus we get,

$$\begin{aligned}
d_T(u_{ij}, u_{kl}) &\geq 2(w - g_{y_i} l) + c - (n_i - 1)\frac{d}{2} - (n_i - 3)g_{y_i} l - (m_i + z_x)\delta \\
&\quad + c - (n_k - 1)\frac{d}{2} - (n_k - 3)g_{y_i} l - (m_k + z_x)\delta \\
&\geq 2w + 2c - (n_i + n_k - 2)\frac{d}{2} - (m_i + m_k + 2z_x)\delta \\
&\quad - (n_i + n_k - 4)g_{y_i} l \\
&> 2w + 2c - c \\
&= 2w + c
\end{aligned}$$

The above calculation implies that for any two leaves (u, v) who have more than two caterpillar- connecting edges on path P_{uv} we get $d_T(u, v) > 2w + c$. Thus if x is a P-node then only the distance between s_x,u_{i1} and t_x,u_{in_i} are equal to $2w + c$, distance between s_x,u_{ij} and t_x,u_{ij} are less than $2w + c$ but greater than $2w + d$, any distance between two leaves having two or more caterpillar-connecting edge between them is greater than $2w + c$.

On the other hand if x is an S-node then Γ_x is a caterpillar with $n - 1$ leaves $s_{y_2} = t_{y_1}, s_{y_3} = t_{y_2}, \cdots, s_{y_n} = t_{y_{n-1}}$. If y_i is a child Q-node of x then $d_T(s_{y_i}, t_{y_i}) = 2w + d$ for $i \neq 1, n$. Also $2w + d < d_T(s_{y_i}, s_{y_j})$, $d_T(t_{y_i}, t_{y_j})$, $d_T(s_{y_i}, t_{y_j}) < 2w + c$ for $i \neq j$ as the path between any of the mentioned pair of leaves contains at least two edge with weight $d + 2gl$ or larger and $c > 2(d + 2\delta + 2l)|V|$.

Let y_i be a child P-node of x and r_j be any child S-node of y_i in $\mathcal{T}$. Clearly Γ_x and Γ_{r_j} is connected by a caterpillar connecting edge. Let r_j has n_{r_j} children which implies Γ_{r_j} has $n_{r_j} - 1$ leaves. Let u_1, u_2 be two leaves in Γ_{r_j} where $d_{\Gamma_{r_j}}(u_1, u_2) = max\{d_{\Gamma_{r_j}}(u_i, u_j)\}$. From the proof of processing at P-node we know $d_T(s_{y_i}, u_1) = 2w + c$ and $d_T(t_{y_i}, u_2) = 2w + c$. Let v be a leaf in Γ_{r_j} where $d_{\Gamma_{r_j}}(u_1, v) < d_{\Gamma_{r_j}}(u_2, v)$ and the path $P_{u_1 v}$ contains e_{r_j} edges on the spine. We also assume that f_{r_j} edges among those e_{r_j} edges are of weight $d + 2g_{r_j} + 2\delta$. Thus $d_T(v, s_{y_i}) = 2w + c - e_{r_j}(d + 2g_{r_j} l) - 2f_{r_j}\delta$. Let s_{y_k} be a leaf in Γ_x where $d_{\Gamma_x}(s_{y_k}, s_{y_i}) < d_{\Gamma_x}(s_{y_k}, t_{y_i})$. We also assume that the path $P_{s_{y_i} s_{y_k}}$ contains e_x edges on the spine of Γ_x and f_x edges among them are of weight $d + 2g_x + 2\delta$. Then $d_T(v, s_{y_k}) = 2w + c - e_{r_j}(d + 2g_{r_j} l) - 2f_{r_j}\delta + e_x(d + 2g_x l) + 2f_x\delta = 2w + c + (e_x - e_{r_j})d + 2(e_x g_x - e_{r_j} g_{r_j})l + 2(f_x - f_{r_j})\delta$. Now as r_j is a grandchild of x we get $g_x = -g_{r_j}$. So, $d_T(v, s_{y_k}) > c + 2w$ if $e_x > w_{r_j}$, $2w + d < d_T(v, s_{y_k}) < c + 2w$ if $e_x < w_{r_j}$. On the other hand if $e_x = e_{r_j}$ then $d_T(v, s_{y_k}) > c + 2w$ if $g_x = 1$ and $2w + d < d_T(v, s_{y_k}) < c + 2w$ if $g_x = -1$. Similarly if $d_{\Gamma_x}(s_{y_k}, s_{y_i}) > d_{\Gamma_x}(s_{y_k}, t_{y_i})$, we get $d_T(v, s_{y_k}) = 2w + c + (e_x - e_{r_j})d + 2(e_x g_x - e_{r_j} g_{r_j})l + 2(f_x - f_{r_j} - 2)\delta$. This also implies that $d_T(c, s_{y_k}) \notin I_2$. By doing similar calculation it can be shown that $d_T(v, s_{y_k}) \notin I_2$ if $d_{\Gamma_{r_j}}(u_1, v) \geq d_{\Gamma_{r_j}}(u_2, v)$. Also the distance between any pair of leaves that have more than two caterpillar-connecting edge in the path between them is greater than $2w + c$. Thus T is a 2-interval PCT of G for $I_1 = [2w + d, 2w + d]$ and $I_2 = [c + 2w, c + 2w]$. □

4 Conclusion

In this paper, we have introduced a new notion named k-interval pairwise compatibility graphs. We have proved that every graph is a k-interval PCGs for some k. We have also showed that wheel graphs and a restricted subclass of series-parallel graphs are 2-interval PCGs. Inception of k-interval PCGs brings in some interesting open problems. It is not known whether some constant number of intervals are sufficient for every graph to be a k-interval PCG. Whether all series-parallel graphs are 2-interval PCGs or not is also unknown.

Acknowledgments

We thank Kazuo Iwama of Kyoto University who pointed out this variant of the problem when the second author discussed the PCG problem with him in 2014.

References

1. G. D. Battista and R. Tamassia. On-line maintenance of triconnected components with spqr-trees. *Algorithmica*, 15(4):302–318, 1996.
2. G. D. Battista, R. Tamassia, and L. Vismara. Output-sensitive reporting of disjoint paths. *Algorithmica*, 23(4):302–340, 1999.
3. M. S. Bayzid. On pairwise compatibility graphs. Master's thesis, Bangladesh University of Engeneering and Technology, June 2010.
4. T. Calamoneri, D. Frascaria, and B. Sinaimeri. All graphs with at most seven vertices are pairwise compatibility graphs. *The Computer Journal*, 57:882–886, 2013.
5. T. Calamoneri, R. Petreschi, and B. Sinaimeri. On relaxing the constraints in pairwise compatibility graphs. In *Proceedings of 6th the International Workshop on Algorithm and Computation, WALCOM*, volume 7157 of Lecture Notes in Computer Science, pages 124–135, 2012.
6. T. Calamoneri, R. Petreschi, and B. Sinaimeri. On the pairwise compatibility property of some superclasses of threshold graphs. *Discrete Mathematics, Algorithms and Applications*, 5(2), 2013.
7. S. Durocher, D. Mondal, and M. S. Rahman. On graphs that are not pcgs. *Theoretical Computer Science*, 571:78–87, 2015.
8. M. I. Hossain, S. A. Salma, and M. S. Rahman. A necessary condtion and a sufficient condition for pairwise compatibility graphs. In *Proceedings of 10th the International Workshop on Algorithm and Computation, WALCOM*, volume 9627 of Lecture Notes in Computer Science, pages 105–113, 2016.
9. P. E. Kearney, J. I. Murno, and D. Phillips. Efficient generation of uniform samples from phylogenetic trees. In *Proceedings of Algorithms and Bioinformatics: 3rd International Workshop (WABI)*, volume 2812 of Lecture Notes in Bioinformatics, pages 177–189. Springer, September 2003.
10. M. S. Rahman, N. Egi, and T. Nishizeki. No-bend orthogonal drawings of series-parallel graphs. In *Proceedings of the 13th international conference on Graph Drawing (GD05), Lecture Notes on Computer Science*, pages 409–420. Springer-Verlag, 2005.

11. S. A. Salma, M. S. Rahman, and M. I. Hossain. Triangle-free outerplanar 3-graphs are pairwise compatibility graphs. *Journal of Graph Algorithms and Applications*, 17(2):81–102, 2013.
12. M. N. Yanhaona, M. S. Bayzid, and M. S. Rahman. Discovering pairwise compatibility graphs. *Discrete Mathematics, Algorithms and Applications*, 2:607–623, 2010.
13. M. N. Yanhaona, K. S. M. T. Hossain, and M. S. Rahman. Pairwise compatibility graphs. *Journal of Applied Mathematics and Computing*, 30:479–503, 2009.