Characterizing Star-PCGs

Mingyu Xiao1 · Hiroshi Nagamochi2

Received: 20 January 2019 / Accepted: 15 April 2020 / Published online: 16 May 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
A graph \( G \) is called a pairwise compatibility graph (PCG, for short) if it admits a tuple \((T, w, d_{\text{min}}, d_{\text{max}})\) of a tree \( T \) whose leaf set is equal to the vertex set of \( G \), a non-negative edge weight \( w \), and two non-negative reals \( d_{\text{min}} \leq d_{\text{max}} \) such that \( G \) has an edge between two vertices \( u, v \in V \) if and only if the distance between the two leaves \( u \) and \( v \) in the weighted tree \((T, w)\) is in the interval \([d_{\text{min}}, d_{\text{max}}]\). The tree \( T \) is also called a witness tree of the PCG \( G \). How to recognize PCGs is a wide-open problem in the literature. This paper gives a complete characterization for a graph to be a star-PCG (a PCG that admits a star as its witness tree), which provides us the first polynomial-time algorithm for recognizing star-PCGs.

Keywords Pairwise compatibility graph · Polynomial-time algorithm · Graph algorithm · Graph theory

1 Introduction

The pairwise compatibility graph is a graph class originally motivated from computational biology. In biology, the evolutionary history of a set of organisms is represented by a phylogenetic tree, which is a tree with leaves representing known taxa and internal nodes representing ancestors that might have led to these taxa through evolution. Moreover, the edges in the phylogenetic tree may be assigned weights to represent the evolutionary distance among species. Given a set of taxa and some relations among the taxa, we may want to construct a phylogenetic tree of the taxa. The set of taxa may be a subset of taxa from a large phylogenetic tree, subject to some biologically-motivated constraints. Kearney et al. [13] considered the following constraint on sampling based on the observation in [11]: the pairwise distance

\[ \text{Mingyu Xiao} \\
myxiao@gmail.com \\
1 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China \\
2 Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, Japan \\

© Springer
between any two leaves in the sample phylogenetic tree is between two given integers $d_{\text{min}}$ and $d_{\text{max}}$. This motivates the introduction of pairwise compatibility graphs (PCGs). Given a phylogenetic tree $T$ with an edge weight $w$ and two real numbers $d_{\text{min}}$ and $d_{\text{max}}$, we can construct a graph $G$ whose each vertex is corresponding to a leaf of $T$ so that there is an edge between two vertices in $G$ if and only if the corresponding two leaves of $T$ are at a distance within the interval $[d_{\text{min}}, d_{\text{max}}]$ in $T$. The graph $G$ is called the PCG of the tuple $(T, w, d_{\text{min}}, d_{\text{max}})$.

It is straightforward to construct a PCG from a given tuple $(T, w, d_{\text{min}}, d_{\text{max}})$. However, the inverse direction seems to be a considerably hard task. Few methods have been known for constructing a corresponding tuple $(T, w, d_{\text{min}}, d_{\text{max}})$ from a given graph $G$. The inverse problem attracts certain interests in graph algorithms, which may also have potential applications in computational biology. PCG has been extensively studied from many aspects after its introduction [3, 7, 8, 10, 19, 20].

A natural question was whether all graphs are PCGs. This was proposed as a conjecture in [13], and was confuted in [19] by giving a counterexample of a bipartite graph with 15 vertices. Later, a counterexample with eight vertices and a counterexample of a planar graph with 20 vertices were found [10]. It has been checked that all graphs with at most seven vertices are PCGs [3] and all bipartite graphs with at most eight vertices are PCGs [15]. In fact, it is even not easy to check whether a graph with a small constant number of vertices is a PCG or not. Whether recognizing PCGs is NP-hard or not is currently open. Some references conjecture the NP-hardness of the problem [8, 10]. A generalized version of PCG recognition is shown to be NP-hard [10].

Several graph classes contained in PCG have been studied. PCG contains the well-studied graph class of leaf power graphs (LPGs) as a subset of instances such that $d_{\text{min}} = 0$, which was introduced in the context of constructing phylogenies from species similarity data [9, 14, 16]. Another natural relaxation of PCG is to set $d_{\text{max}} = \infty$. This graph class is known as min leaf power graph (mLPG) [7], which is the complement of LPG. Several other known graph classes have been shown to be subclasses of PCG, e.g., disjoint union of cliques [2], forests [12], cacti [18], chordless cycles and single chord cycles [20], complete $k$-partite graphs [18], tree power graphs [19], threshold graphs [7], triangle-free outerplanar 3-graphs [17], some particular subclasses of split matrojenic graphs [7], Dilworth $k$ graphs [5, 6], the complement of a forest [12] and so on. It is also known that a PCG with a witness tree being a caterpillar also allows a witness tree being a centipede [4]. A method for constructing PCGs is derived in [18], where it is shown that a graph $G$ consisting of two graphs $G_1$ and $G_2$ that share a vertex as a cut-vertex in $G$ is a PCG if and only if both $G_1$ and $G_2$ are PCGs.

How to recognize PCGs or construct a corresponding phylogenetic tree for a PCG becomes a wide-open problem in this area. To make a step toward this open problem, we consider PCGs with a witness tree being a star in this paper, which we call star-PCGs. It mentioned in [8] that most of the witness trees of PCGs in the literature have simple graph structures, such as stars and caterpillars. It is fundamental to consider the problem of characterizing subclasses of PCGs derived from a specific topology of
trees. Although stars are trees with a rather simple topology, star-PCG recognition is not easy at all. It is known that threshold graphs are star-PCGs (even in star-LPG and star-mLPG) and the class of star-PCGs is nearly the class of three-threshold graphs, a graph class extended from the threshold graphs [7]. However, no complete characterization of star-PCGs and no polynomial-time recognition of star-PCGs are known. In this paper, we give a complete characterization for a graph to be a star-PCG, which provides us the first polynomial-time algorithm for recognizing star-PCGs. One of the most important results in the paper is the following theorem.

**Theorem 1** Whether a graph with \( n \) vertices and \( m \) edges is a star-PCG or not can be tested in \( O(n^3m) \) time.

The main idea of our algorithm is as follows. Without loss of generality, we always rank the leaves of the witness star \( T_V \) (and the corresponding vertices in the star-PCG \( G \)) according to the weight of the edges incident on it. When such an ordering of the vertices in a star-PCG \( G \) is given, we get that all the neighbors of each vertex in \( G \) must appear consecutively in the ordering. This motivates us to define such an ordering to be “consecutive ordering.” To check if a graph is a star-PCG, we can first check if the graph can have a consecutive ordering of vertices. Consecutive orderings can be computed in polynomial time by reducing to the problem of recognizing interval graphs. However, this is not enough to test star-PCGs. A graph may not be a star-PCG even if it has a consecutive ordering of vertices. We further study the structural properties of star-PCGs on a fixed consecutive ordering of vertices. We find that three cases of non-adjacent vertex pairs called gaps, can be used to characterize star-PCGs. A graph is a star-PCG if and only if it admits a consecutive ordering of vertices that is gap-free (Theorem 2). Finally, to show that whether a given graph is gap-free or not can be tested in polynomial time (Theorem 5), we also use a notion of “contiguous orderings.” All these together contribute to a polynomial-time algorithm for our problem.

The paper is organized as follows. Section 2 introduces some basic notions and notations necessary to this paper. Section 3 characterizes the class of star-PCGs \( G = (V, E) \) in terms of an ordering \( \sigma \) of the vertex set \( V \), called a “gap-free” ordering, and shows that given a gap-free ordering of \( V \), a tuple \((T, w, d_{\min}, d_{\max})\) that represents \( G \) can be computed in polynomial time. The results in Sect. 3 are not enough to lead a polynomial-time recognition of star-PCGs. We still need to design a polynomial-time algorithm to find “gap-free” orderings of the vertices, which are handled in Sects. 4 and 5. Section 4 discusses how to test whether a given family \( S \) of subsets of an element set \( V \) admits a special ordering on \( V \), called “consecutive” or “contiguous” orderings and proves the uniqueness of such orderings under some conditions on \( S \). Section 5 first derives structural properties on a graph that admits a “gap-free” ordering, and then presents a method for testing whether a given graph is a star-PCG or not in polynomial time by using the result on contiguous orderings to a family of sets. Finally, Sect. 6 makes some concluding remarks.
2 Preliminaries

For two integers $a$ and $b$, let $[a, b]$ denote the set of integers $i$ with $a \leq i \leq b$. For a sequence $\sigma$ of elements, let $\overline{\sigma}$ denote the reversal of $\sigma$. A sequence obtained by concatenating two sequences $\sigma_1$ and $\sigma_2$ in this order is denoted by $(\sigma_1, \sigma_2)$.

Families of Sets Let $V$ be a set of $n \geq 1$ elements. We call a subset $S \subseteq V$ trivial in $V$ if $|S| \leq 1$ or $S = V$. We say that a set $X$ has a common element with a set $Y$ if $X \cap Y \neq \emptyset$. We say that two subsets $X, Y \subseteq V$ intersect (or $X$ intersects $Y$) if three sets $X \cap Y$, $X \setminus Y$, and $Y \setminus X$ are all non-empty sets. A partition $\{V_1, V_2, \ldots, V_k\}$ of $V$ is defined to be a collection of disjoint non-empty subsets $V_i$ of $V$ such that their union is $V$, where possibly $k = 1$.

Let $S \subseteq 2^V$ be a family of $m$ subsets of $V$. A total ordering $u_1, u_2, \ldots, u_n$ of elements in $V$ is called consecutive to $S$ if each non-empty set $S \subseteq V$ consists of elements with consecutive indices, i.e., $S$ is equal to $\{u_i, u_{i+1}, \ldots, u_{i+|S|-1}\}$ for some $i \in [1, n - |S| - 1]$. A consecutive ordering $u_1, u_2, \ldots, u_n$ of elements in $V$ to $S$ is called contiguous if any two sets $S, S' \subseteq S$ with $S' \subseteq S$ start from or end with the same element along the ordering, i.e., $S' = \{u_j, u_{j+1}, \ldots, u_{j+|S'|-1}\}$ and $S = \{u_i, u_{i+1}, \ldots, u_{i+|S|-1}\}$ satisfy $j = i$ or $j + |S'| = i + |S|$.

Graphs Let a graph stand for a simple undirected graph. A graph (resp., bipartite graph) with a vertex set $V$ and an edge set $E$ (resp., an edge set $E$ between two vertex sets $V_1$ and $V_2 = V \setminus V_1$) is denoted by $G = (V, E)$ (resp., $(V_1, V_2, E)$). Let $G$ be a graph, where $V(G)$ and $E(G)$ denote the sets of vertices and edges in a graph $G$, respectively. For a vertex $v$ in $G$, we denote by $N_G(v)$ the set of neighbors of a vertex $v$ in $G$, and define degree $\text{deg}_G(v)$ to be the $|N_G(v)|$. A pair of vertices $u$ and $v$ in $G$ are called twins if $N_G(v) \setminus \{u\} = N_G(u) \setminus \{v\}$, where $u$ and $v$ are not necessarily adjacent. Let $X$ be a subset of $V(G)$. Define $N_G(X)$ to be the set of neighbors of $X$, i.e., $N_G(X) = \{u \in N_G(v) \mid v \in X\}$. Let $G - X$ denote the graph obtained from $G$ by removing vertices in $X$ together with all edges incident to vertices in $X$, where $G - \{v\}$ for a vertex $v$ may be written as $G - v$. Let $G[X]$ denote the graph induced by $X$, i.e., $G[X] = G - (V(G) \setminus X)$.

Let $T$ be a tree. A vertex $v$ in $T$ is called an inner vertex if $\text{deg}_T(v) \geq 2$ and is called a leaf otherwise. Let $L(T)$ denote the set of leaves in $T$. An edge incident to a leaf in $T$ is called a leaf edge of $T$. A tree $T$ is called a star if it has at most one inner vertex.

Weighted Graphs An edge-weighted graph $(G, w)$ is defined to be a pair of a graph $G$ and a non-negative weight function $w : E(G) \to \mathbb{R}_+$. For a subgraph $G'$ of $G$, let $w(G')$ denote the sum $\sum_{e \in E(G')} w(e)$ of edge weights in $G'$.

Let $(T, w)$ be an edge-weighted tree. For two vertices $u, v \in V(T)$, let $d_{T,w}(u, v)$ denote the sum of weights of edges in the unique path of $T$ between $u$ and $v$.

PCGs For a tuple $(T, w, d_{\text{min}}, d_{\text{max}})$ of an edge-weighted tree $(T, w)$ and two non-negative reals $d_{\text{min}}$ and $d_{\text{max}}$, define $G(T, w, d_{\text{min}}, d_{\text{max}})$ to be the simple graph
(L(T), E) such that, for any two distinct vertices u, v ∈ L(T), uv ∈ E if and only if 

\[ d_{\text{min}} \leq d_{T,w}(u, v) \leq d_{\text{max}} \]

Note that G(T, w, d_{\text{min}}, d_{\text{max}}) is not necessarily connected.

A graph G is called a pairwise compatibility graph (PCG, for short) if there exists a tuple (T, w, d_{\text{min}}, d_{\text{max}}) such that G is isomorphic to the graph G(T, w, d_{\text{min}}, d_{\text{max}}), where we call such a tuple a pairwise compatibility representation (PCR, for short) of G, and call a tree T in a PCR of G a pairwise compatibility tree (PCT, for short) of G. The tree T is called a witness tree of G. We call a PCG G a star-PCG if it admits a PCR (T, w, d_{\text{min}}, d_{\text{max}}) such that T is a star. Figure 1 illustrates examples of star-PCGs and PCRs of them. Although phylogenetic trees may not have edges with weight 0 or degree-2 vertices by some biological motivations [4], our PCTs do not have these constraints. This relaxation will be helpful for us to analyze the structural properties of PCGs from graph theory. Furthermore, it is easy to get rid of edges with weight 0 or degree-2 vertices in a tree by contracting an edge.

**Lemma 1** [18] Every PCG admits a PCR (T, w, d_{\text{min}}, d_{\text{max}}) such that \( 0 < d_{\text{min}} < d_{\text{max}} \) and \( w(e) > 0 \) for all edges \( e \in E(T) \).

A construction proof for this lemma can be found in [18]. For Star-PCGs, all edges in the PCT are leaf edges. There is a simple method to prove that every PCG admits a PCR (T, w, d_{\text{min}}, d_{\text{max}}) such that \( 0 < d_{\text{min}} < d_{\text{max}} \) and \( w(e) > 0 \) for all leaf edges L(T) in the tree T. The proof is as follows. We only argue the non-trivial cases where each connected component of the graph G contains at least three vertices. Let (T, w', d'_{\text{min}}, d'_{\text{max}}) be a PCR of G, where each path between two leaves in T contains exactly two leaf edges since each connected component of the graph

![Illustration of examples of star-PCG](image-url)

**Fig. 1** Illustration of examples of star-PCG. a A connected and bipartite star-PCG \( G_1 = (V_1, V_2, E) \). b The same graph \( G_1 \) in a with a different ordering of vertices (the two different orderings of vertices in a, b will be used to illustrate the concept of “gap-free” later). c A PCR (T, w, d_{\text{min}} = 8, d_{\text{max}} = 9) of \( G_1 \). d A connected and non-bipartite star-PCG \( G_2 \). e A PCR (T, w, d_{\text{min}} = 4, d_{\text{max}} = 8) of \( G_2 \) in d
$G$ contains at least three vertices. Let $\delta > \max\{|d'_{\min}|, |\min_{e \in L(T)} w'(e)|\}$ be a constant. We increase by $\delta$ the values of $d'_{\min}, d'_{\max}$ and edge weights for all leaf edges in $L(T)$, i.e., let $w(e) = w'(e) + \delta$ for each leaf edge $e \in L(T)$, $w(e) = w'(e)$ for all non-leaf edges $e \in E(T) \setminus L(T)$, $d'_{\min} := d'_{\min} + 2\delta$, and $d'_{\max} := d'_{\max} + 2\delta$. We observe that $(T, w, d'_{\min}, d'_{\max})$ is a PCR of $G$ satisfying the requirement because each path between two leaves in $T$ contains exactly two leaf edges.

### 3 Star-PCGs with Fixed Orderings of Vertices

Let $G = (V, E)$ be a graph with $n \geq 2$ vertices, not necessarily connected. In this section, we assume that the graph $G$ does not contain any isolated vertex (degree-0 vertex). In fact, degree-0 vertices can be easily handled by letting them corresponding to leave edges with the weight $> d_{\max}$ in the PCT. Let $T_V$ be a star with a center $v^*$ and $L(T) = V$. An ordering of $V$ is defined to be a bijection $\sigma : V \to \{1, 2, \ldots, n\}$, and we simply write a vertex $v$ with $\sigma(v) = i$ as $v_i$. For an edge weight $w$ in $T_V$, we simply denote $w(v_i, v_j)$ by $w_{ij}$.

When $G$ is a star-PCG of a tuple $(T_V, w, d_{\min}, d_{\max})$, there is an ordering $\sigma$ of $V$ such that $w_1 \leq w_2 \leq \cdots \leq w_n$. Conversely, this section derives a necessary and sufficient condition for a pair $(G, \sigma)$ of a graph $G$ and an ordering $\sigma$ of $V$ to admit a PCR $(T_V, w, d_{\min}, d_{\max})$ of $G$ such that $w_1 \leq w_2 \leq \cdots \leq w_n$.

For an ordering $\sigma$ of $V$, a non-adjacent vertex pair $\{v_i, v_j\}$ with $i < j$ in $G$ is called a gap (with respect to edges $e_1, e_2 \in E$) if there are edges $e_1, e_2 \in E$ that satisfy one of the following:

- **(g1)** $e_1 = v_i v_j$ and $e_2 = v_j v_{i'}$ such that $j' < j < j''$ (or $e_1 = v_j v_i$ and $e_2 = v_{i'} v_j$ such that $i' < i < i''$), as illustrated in Fig. 2a;
- **(g2)** $e_1 = v_i v_{i'}$ and $e_2 = v_{i'} v_j$ such that $j' < i$ and $j < i'$, as illustrated in Fig. 2b;
- **(g3)** $e_1 = v_i v_{i'}$ and $e_2 = v_j v_j$ such that $i' < j$ and $i < j'$, as illustrated in Fig. 2c.

We call an ordering $\sigma$ of $V$ gap-free in $G$ if it has no gap. Clearly, the reversal of a gap-free ordering of $V$ is also gap-free. For the same graph, some orderings of the vertices can be gap-free and some may not. Figure 1a, b illustrate the same graph $G_1$ with

---

**Fig. 2** Illustration of a gap $\{v_i, v_j\}$ with respect to edges $e_1$ and $e_2$ in an ordered graph $G = (V = \{v_1, v_2, \ldots, v_n\}, E)$, where edges are denoted by solid lines and anti-edges are denoted by dashed lines: a $e_1 = v_i v_j$ and $e_2 = v_j v_{i'}$ such that $j' < j < j''$, b $e_1 = v_j v_i$ and $e_2 = v_{i'} v_j$ such that $j < i$ and $j < i'$, c $e_1 = v_{i'} v_i$ and $e_2 = v_j v_j$ such that $i' < j$ and $i < j'$, where possibly $j' \leq i'$ or $i' < j'$.
different orderings $\sigma_a = v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8$ and $\sigma_b = v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8$. For the ordering $\sigma_a$, we can see that $\{v_1, v_5\}$ is a gap with respect to edges $v_1v_8$ and $v_4v_5$ [the condition of (g3)]. So $\sigma_a$ is not gap-free while $\sigma_b$ is gap-free.

We can test if a given ordering of the vertices of a graph is gap-free or not in polynomial time by checking the three conditions (g1), (g2) and (g3) for each non-adjacent vertex pair $\{v_i, v_j\}$ in the graph. We give a simple algorithm in the following lemma.

**Lemma 2**  Whether an ordering $\sigma = v_1, v_2, \ldots, v_n$ of the vertices of a graph $G = (V, E)$ is gap-free or not can be checked in $O(n^2)$ time.

**Proof**  For each vertex $v_i \in V$, we check whether or not the neighbor set $N_G(v_i)$ is given by a set $\{v_{a(i)}, v_{a(i)+1}, \ldots, v_{b(i)}\}\{v_i\}$ of vertices with consecutive indices. If no, there is a gap in the condition (g1). If yes, there is no gap in the condition (g1) and the set of indices $\{a(i), b(i) | i = 1, 2, \ldots, n\}$ can be computed in $O(n^2)$.

For any non-adjacent vertex pair $\{v_i, v_j\}$, it is not a gap in the condition (g2) if and only if $b(i) < j$ or $a(j) < i$; it is not a gap in the condition (g3) if and only if $a(i) > j$ or $b(j) < i$. For each non-adjacent vertex pair $\{v_i, v_j\}$, we use only a constant time to check whether it is a gap in the condition (g2) or (g3). There are $O(n^2)$ non-adjacent vertex pairs.

In total, we can check whether an ordering of vertices is gap-free or not in $O(n^2)$ time.

The main result in this section is the following theorem, which implies that a graph $G = (V, E)$ is a star-PCG if and only if it admits a gap-free ordering of $V$.

**Theorem 2**  For a graph $G = (V, E)$ having no isolated vertex, let $\sigma$ be an ordering of $V$. Then there is a PCR $(T_V, w, d_{\min}, d_{\max})$ of $G$ such that $w_1 \leq w_2 \leq \cdots \leq w_n$ if and only if $\sigma$ is gap-free.

The necessity of this theorem is relatively easy to prove, which is observed as follows.

**Lemma 3**  For a graph $G = (V, E)$, let $\sigma = v_1, v_2, \ldots, v_n$ be an ordering of $V$. If $\sigma$ is not gap-free, then there is no PCR $(T_V, w, d_{\min}, d_{\max})$ of $G$ such that $w_1 \leq w_2 \leq \cdots \leq w_n$.

**Proof**  Let $(G, \sigma)$ admit a PCR $(T_V, w, d_{\min}, d_{\max})$ with $w_1 \leq w_2 \leq \cdots \leq w_n$. To derive a contradiction, assume that $\sigma$ has a gap $\{v_i, v_j\}$ with $i < j$. If it satisfies the condition (g1) with respect to edges $e_1 = v_iv_j$ and $e_2 = v_jv_{i'}$ such that $j' < j$ (or $e_1 = v_jv_i$ and $e_2 = v_{i'}v_j$ such that $i'' < i$), then $d_{\min} \leq w_i + w_{j'} \leq w_i + w_j \leq w_i + w_{i} \leq d_{\max}$ (or $d_{\min} \leq w_{i'} + w_j \leq w_i + w_j \leq w_{i'} + w_j \leq d_{\max}$), which implies $v_iv_j \in E$, i.e., $v_i$ and $v_j$ must be adjacent in $G$, a contradiction. If the gap satisfies the condition (g2) with respect to edges $e_1 = v_iv_{j'}$ and $e_2 = v_{j'}v_j$ such that $j' < i$ and $j < i''$, then $d_{\min} \leq w_{j'} + w_j \leq w_i + w_j \leq w_{j'} + w_j \leq d_{\max}$, again implying that $v_i$ and $v_j$ must be
adjacent, a contradiction. Analogously with the case where the gap satisfies the condition \((g^3)\) with respect to edges \(e_1 = v_i v_j\) and \(e_2 = v_j v_i\) such that \(i' < j\) and \(i < j'\), where \(d_{\min} \leq w_i + w_j \leq w_i + w_j \leq w_{j'} + w_i \leq d_{\max}\) would imply that \(v_i\) and \(v_j\) are adjacent in \(G\).

In the rest of this section, we prove the sufficiency of Theorem 2 with the next lemma.

**Lemma 4** For a graph \(G = (V, E)\) with \(|V| = n\) vertices having no isolated vertex, let \(\sigma = v_1, v_2, \ldots, v_n\) be a gap-free ordering of \(V\). There is a PCR \((T, v, w, d_{\min}, d_{\max})\) of \(G\) such that \(w_1 \leq w_2 \leq \cdots \leq w_n\). Such a set \(\{w_1, w_2, \ldots, w_n, d_{\min}, d_{\max}\}\) of weights and bounds can be obtained in \(O(n^2)\) time.

An important technique used to prove this lemma is to color all edges and anti-edges in the graph (i.e., all edges in the complete graph) adjacent in \(G\), i.e., there is an edge or anti-edge in the PCT. Note that when two vertices \(u\) and \(v\) are not adjacent in a PCG \(G\), there are two reasons: one is that the distance between them in the PCR \((T, w, d_{\min}, d_{\max})\) is smaller than \(d_{\min}\), and the other is that the distance is larger than \(d_{\max}\). We will use two different colors to distinguish these two kinds of anti-edges. It will become easier to set weights to edges and bounds after coloring all the anti-edges.

A coloring of a graph \(G = (V, E)\) is a function \(c : E \cup \bar{E} \to \{r, g, b\}\) a red (resp., green and blue) edge. We use \(\text{red}\) (resp., \(\text{green}\) and \(\text{blue}\)) to denote the set of vertex pairs \((i, j)\) such that edges \(v_i v_j \in E \cup \bar{E}\) are red (resp., green and blue) edges. We denote by \(N_{\text{red}}(v)\) the set of neighbors of a vertex \(v\) via red edges. We define \(N_{\text{green}}(v)\) and \(N_{\text{blue}}(v)\) analogously.

For a graph \(G\) with an ordering \(\sigma = v_1, v_2, \ldots, v_n\) of the vertices, a coloring \(c\) is *proper* to \((G, \sigma)\) if it satisfies the following conditions:

- each \(v_i \in V\) admits integers \(a(i), b(i) \in [1, n]\) such that
  \[
  N_{\text{red}}(v_i) = \{v_j \mid 1 \leq j \leq a(i) - 1\} \setminus \{v_i\},
  \]
  \[
  N_{\text{blue}}(v_i) = \{v_j \mid b(i) + 1 \leq j \leq n\} \setminus \{v_i\},
  \]
  where \(a(i) = 1\) if \(N_{\text{red}}(v_i) = \emptyset\), and \(b(i) = n\) if \(N_{\text{blue}}(v_i) = \emptyset\).

  Note that when the graph has no isolated vertex, for each \(v_i \in V\), \(N_{\text{green}}(v_i) \neq \emptyset\) and then \(a(i) \leq b(i)\). There are cases where \(a(i)\) and \(b(i)\) may not be fixed indices. When \(N_{\text{red}}(v_i) = \{v_j \mid 1 \leq j \leq i - 1\}\), according to the above definition of \(a(i)\), we know that \(a(i)\) can be \(i - 1\) or \(i\). When \(N_{\text{blue}}(v_i) = \{v_j \mid i + 1 \leq j \leq n\}\), according to the definition of \(b(i)\), we know that \(b(i)\) can be \(i + 1\) or \(i\). For these cases, to avoid ambiguity, we may let \(a(i) = i - 1\) if \(w_i + w_j \geq d_{\min}\) and \(a(i) = i\) if \(w_i + w_j < d_{\min}\), let \(b(i) = i + 1\) if \(w_i + w_j \leq d_{\max}\) and \(b(i) = i\) if \(w_i + w_j > d_{\max}\).
Lemma 5 For a graph $G = (V, E)$ having no isolated vertex and a gap-free ordering $\sigma$ of $V$, there is a coloring $c$ of $G$ that is proper to $(G, \sigma)$, which can be found in $O(n^2)$ time.

Proof First, we assign color green to all edges in $E$. Since $(G, \sigma)$ has no gap in the condition (g1), the neighbor set $N_G(v_i)$ of each vertex $v_i \in V$ is given by a set $\{v_{a(i)}, v_{a(i)+1}, \ldots, v_{b(i)}\} \setminus \{v_i\}$ of vertices with consecutive indices. Note that any $v_i \in V$ is not an isolated vertex and then the set $\{v_{a(i)}, v_{a(i)+1}, \ldots, v_{b(i)}\} \setminus \{v_i\}$ is not an empty set. Next, for each vertex $v_i \in V$ we assign color red to all edges $v_jv_i \in \overline{E}$ with $j < a(i)$ and color blue to all edges $v_jv_k \in \overline{E}$ with $b(i) < k$. This is the whole algorithm, which runs in $O(n^2)$ time since each edge in $E$ will be colored by only one time and each edge in $\overline{E}$ will be colored by at most two times.

To prove the correctness of the algorithm, it suffices to show that no edge $v_iv_j \in \overline{E}$ with $i < j$ is assigned two colors at the same time, in such a way that either (1) color red from $v_i$ and color blue from $v_j$, or (2) color blue from $v_i$ and color red from $v_j$.

When (1) (resp., (2)) occurs, there are edges $v_i v_j \in E$ and $v_j v_i \in E$ such that $j' < i < j < i'$ (resp., $i' < j$ and $i' < j'$), which means that $\{v_i, v_j\}$ would be a gap in the condition (g2) (resp., (g3)), a contradiction. Hence the above process constructs a coloring $c$ proper to $(G, \sigma)$. \hfill $\square$

We next show that for a graph $G$ having no isolated vertex and an ordering $\sigma$ of the vertices, if there is a coloring $c$ proper to $(G, \sigma)$, then we can find weights $w_i$ and bounds $d_{\min}$ and $d_{\max}$ so that the following inequalities hold:

\begin{align*}
w_1 &\leq w_2 \leq \cdots \leq w_n; \\
d_{\min} &\leq w_i + w_j \leq d_{\max} \text{ for } (i, j) \in \text{green}; \\
w_i + w_j &< d_{\min} \text{ for } (i, j) \in \text{red}; \\
w_i + w_j &> d_{\max} \text{ for } (i, j) \in \text{blue}.
\end{align*}

With these weights and bounds, we can obtain a PCR of $G$ and then can prove Lemma 4.

Recall that $a(i)$ and $b(i)$ are defined in the above definition of proper coloring. We define integers $i_{\text{red}}$ and $i_{\text{blue}}$ as follows.

\begin{align*}i_{\text{red}} &= \begin{cases} 
\text{the largest index } i \text{ such that } i < a(i) & \text{if } \text{red} \neq \emptyset, \\
0 & \text{if } \text{red} = \emptyset,
\end{cases} \\
i_{\text{blue}} &= \begin{cases} 
\text{the smallest index } i \text{ such that } b(i) < i & \text{if } \text{blue} \neq \emptyset, \\
n + 1 & \text{if } \text{blue} = \emptyset.
\end{cases}
\end{align*}

The following properties hold for the set of indices $\{a(i), b(i) \mid i = 1, 2, \ldots, n\} \cup \{i_{\text{red}}, i_{\text{blue}}\}$.

Lemma 6 For a graph $G = (V, E)$ having no isolated vertex, an ordering $\sigma = v_1, v_2, \ldots, v_n$ of $V$, and a coloring $c$ of $G$ proper to $(G, \sigma)$, it holds that...
(1) when \( i_{\text{red}} \neq 0 \), the index \( i_{\text{red}} \) is the largest index \( i \) with \( (i, i + 1) \in \text{red} \);

(2) when \( i_{\text{blue}} \neq n + 1 \), the index \( i_{\text{blue}} \) is the smallest index \( i \) with \( (i - 1, i) \in \text{blue} \);

(3) \( i_{\text{red}} \leq n - 1 \) and \( i_{\text{blue}} \geq 2 \).

**Proof**

(1) When \( i_{\text{red}} \neq 0 \), the set \( \text{red} \) is not an empty set. If \( i_{\text{red}} \) is an index such that \( (i_{\text{red}}, i_{\text{red}} + 1) \in \text{red} \), then \( i_{\text{red}} \) is the largest \( i \) with \( (i, i + 1) \in \text{red} \) by the definition of \( i_{\text{red}} \). Next we assume that \( (i_{\text{red}}, i_{\text{red}} + 1) \notin \text{red} \), which means that \( a(i_{\text{red}}) > i_{\text{red}} + 1 \). However, by the definition of \( a(i_{\text{red}}) \), we know that for any index \( j < a(i_{\text{red}}) \), the pair \( (i_{\text{red}}, j) \) must be in \( \text{red} \), contradicting the assumption that \( (i_{\text{red}}, i_{\text{red}} + 1) \notin \text{red} \). So \( (i_{\text{red}}, i_{\text{red}} + 1) \) must be in \( \text{red} \) and then (1) holds.

(2) It can be proved analogously with (i).

(3) It clearly holds because either \( i_{\text{red}} = 0 \) or \( i_{\text{red}} < a(i_{\text{red}}) \leq n \), and either \( i_{\text{blue}} = n + 1 \) or \( i_{\text{blue}} > b(i_{\text{blue}}) \geq 1 \).

**Lemma 7** For a graph \( G = (V, E) \) having no isolated vertex, an ordering \( \sigma = v_1, v_2, \ldots, v_n \) of \( V \), and a coloring \( c \) of \( G \) proper to \( (G, \sigma) \), the following holds

(1) Every two indices \( i \) and \( j \) with \( 1 \leq i < j \leq n \) satisfy \( a(j) \leq a(i) \) and \( b(j) \leq b(i) \);

(2) It holds that \( i_{\text{red}} + 1 \leq i_{\text{blue}} - 1 \), \( (i, j) \in \text{red} \) for \( i < j \leq i_{\text{red}} \), \( (i, j) \in \text{blue} \) for \( i_{\text{blue}} \leq i < j \), and \( (i, j) \in \text{green} \) for \( i_{\text{red}} < i < j < i_{\text{blue}} \).

**Proof** (1) Let \( 1 \leq i < j \leq n \). Assume to the contrary that \( a(i) < a(j) \). By the definition of \( a(j) \) and \( a(i) < a(j) \), we know that \( (j, a(j)) \in \text{green} \) and \( (j, a(i)) \in \text{red} \). By the definition of \( a(i) \), we know that \( (i, a(i)) \in \text{green} \). We look at the index \( a(i) \). Since \( c \) is proper and \( (j, a(i)) \in \text{red} \), we know that for any \( j' < j \), it holds that \( (j', a(i)) \in \text{red} \), a contradiction to that \( (i, a(i)) \in \text{green} \) and \( i < j \).

Assume to the contrary that \( b(i) < b(j) \). By the definition of \( b(i) \) and \( b(i) < b(j) \), we know that \( (b(i), i) \in \text{green} \) and \( (b(j), i) \in \text{blue} \). By the definition of \( b(j) \), we know that \( (b(j), j) \in \text{green} \). We look at the index \( b(j) \). Since \( c \) is proper and \( (b(j), i) \in \text{blue} \), we know that for any \( i' > i \), it holds that \( (b(j), i') \in \text{blue} \), a contradiction to that \( (b(j), j) \in \text{green} \) and \( j > i \).

So it holds that \( a(j) \leq a(i) \) and \( b(j) \leq b(i) \).

(2) By Lemma 6(3), we know that that \( i_{\text{red}} \leq n - 1 \) and \( i_{\text{blue}} \geq 2 \). If \( i_{\text{red}} = 0 \) or \( i_{\text{blue}} = n + 1 \), then \( i_{\text{red}} + 2 \leq i_{\text{blue}} \) holds. Next we assume that \( \text{red} \neq \emptyset \) and \( \text{blue} \neq \emptyset \). Now the index \( i_{\text{red}} \) is the largest \( i \) with \( (i, i + 1) \in \text{red} \) by Lemma 6, which implies that \( (j, i_{\text{red}} + 1) \in \text{red} \) for all \( j \leq i_{\text{red}} \). Hence for any index \( i \) if \( (i - 1, i) \in \text{blue} \) then \( i_{\text{red}} + 1 \leq i - 1 \). Since the index \( i_{\text{blue}} \) is the smallest \( i \) with \( (i - 1, i) \in \text{blue} \) by Lemma 6, we know that \( i_{\text{red}} + 1 \leq i_{\text{blue}} - 1 \), as required.

For every \( i_{\text{red}} < i < j \), it holds that \( (i, j) \notin \text{red} \) by the definition of \( i_{\text{red}} \), and for every \( i < j < i_{\text{blue}} \), it holds that \( (i, j) \notin \text{blue} \) by the definition of \( i_{\text{blue}} \). We show that \( (i, j) \notin \text{green} \) for every \( i < j \leq i_{\text{red}} \). Assume to the contrary that \( (i, j) \in \text{green} \) for some \( i < j \leq i_{\text{red}} \). Since the coloring is proper and \( j < i_{\text{red}} \), we know that
Given a graph $G = (V, E)$ having no isolated vertex, a gap-free ordering $\sigma$ of $V$, and a coloring $c$ proper to $(G, \sigma)$, we can compute in $O(n^2)$ time the set \{a(i), b(i) \mid i = 1, 2, \ldots, n\} \cup \{i_{\text{red}}, i_{\text{blue}}\}$ of indices. To finish the proof of Lemma 4, next we design an $O(n)$-time algorithm that assigns the right values to weights $w_1, w_2, \ldots, w_n$ in $T_V$, where the values may be negative. Note that there is an easy way to convert weights of edges in a PCT into positive values if necessary without changing the tree in the PCR, see Lemma 1.

The algorithm contains three steps, each of which runs in $O(n)$ time. The first step is to set the values for \{\{w_{i_{\text{red}}}, w_{i_{\text{red}} + 1}\}, \ldots, w_{i_{\text{blue}} - 1}, d_{\text{min}}, d_{\text{max}}\};$ the second step is to set the values for \{\{w_1, w_2, \ldots, w_n\};$ the third step is to set the values for \{\{w_{i_{\text{blue}}}, w_{i_{\text{blue}} + 1}, \ldots, w_n\}.

In the algorithm, we will maintain two indices \(p, q \in [1, n]\) with $p \leq q - 2$ such that the weights $w_i$ with $p + 1 \leq i \leq q - 1$ and bounds $d_{\text{min}}$ and $d_{\text{max}}$ have been determined so that

(a) \(w_{p+1} \leq w_{p+2} \leq \cdots \leq w_{q-1};\)
(b) \(w_j < w_{j+1}\) for $j \in [p + 1, q - 2]$ such that $v_j$ and $v_{j+1}$ are not twins;
(c) For any different indices $i, j \in [p + 1, q - 1]$

\[
d_{\text{max}} > w_i + w_j > d_{\text{min}} \text{ if } (i, j) \in \text{green},
\]
\[
w_i + w_j < d_{\text{min}} \text{ if } (i, j) \in \text{red, and}
\]
\[
w_i + w_j > d_{\text{max}} \text{ if } (i, j) \in \text{blue.}
\]

Step 1 We start with setting weights $w_i$ with $p + 1 \leq i \leq q - 1$ and bounds $d_{\text{min}}$ and $d_{\text{max}}$ for $p = i_{\text{red}}$ and $q = i_{\text{blue}}$.

Lemma 8 For a graph $G = (V, E)$ having no isolated vertex and an ordering $\sigma = v_1, v_2, \ldots, v_n$ of $V$, let $c$ be a coloring of $G$ proper to $(G, \sigma)$. For $p = i_{\text{red}}$ and $q = i_{\text{blue}}$, we can compute in linear time a set \{\{w_{p+1}, w_{p+2}, \ldots, w_{q-1}, d_{\text{min}}, d_{\text{max}}\}$ of weights and bounds that satisfies the conditions (a)–(c).

Proof By Lemma 7(1), we know that \((i, j) \in \text{green for all } i_{\text{red}} < i < j < i_{\text{blue}}. We set the required weights $w_i$ for all $i$ with $i_{\text{red}} < i < i_{\text{blue}}$ so that $d_{\text{min}} < w_i + w_{p} < d_{\text{max}}$ for all $i, i'$ with $i_{\text{red}} < i < i' < i_{\text{blue}}$. We use the following program to set the weights $w_i$: Let $w_{i_{\text{red}} + 1} := 1$. For each $i = i_{\text{red}} + 1, i_{\text{red}} + 2, \ldots, i_{\text{blue}} - 2$, do that let $w_{i+1} := w_i$ if $v_i$ and $v_{i+1}$ are twins, and let $w_{i+1} := w_i + 1$ otherwise. Let $d_{\text{min}} := 2w_{i_{\text{red}} + 1} - 0.5 = 1.5$ and $d_{\text{max}} := 2w_{i_{\text{blue}} - 1} + 0.5$. It is easy to verify that this setting satisfies the conditions (a)–(c). \[\Box\]
In the next steps, without changing the determined values to \(w_{p+1}, w_{p+2}, \ldots, w_{q-1}\), we determine \(w_p\) or \(w_q\) so that the conditions (a)–(c) hold for \((p := p - 1, q)\) or \((p, q := q + 1)\).

Step 2 We determine values to weights \(w_{i_{\text{red}}}, w_{i_{\text{red}}-1}, \ldots, w_1\) in this order by using the following lemma.

**Lemma 9** For a graph \(G = (V, E)\) no isolated vertex and an ordering \(\sigma = v_1, v_2, \ldots, v_n\) of \(V\), let \(c\) be a coloring of \(G\) proper to \((G, \sigma)\). For \(p \leq i_{\text{red}}\) and \(q \geq i_{\text{blue}}\), assume that a set \(\{w_{p+1}, w_{p+2}, \ldots, w_{q-1}, d_{\min}, d_{\max}\}\) of weights and bounds satisfies the conditions (a)–(c). When \(p \geq 1\), let weight \(w_p\) be determined such that

\[
\begin{align*}
    w_p &= \begin{cases} 
        w_{p+1} & \text{if } v_p \text{ and } v_{p+1} \text{ are twins,} \\
        w_{p+1} - \alpha/2 & \text{if } v_p \text{ and } v_{p+1} \text{ are not twins, and } a(p + 1) = a(p) \leq q - 1, \\
        w_{p+1} - \beta - \delta/2 & \text{if } v_p \text{ and } v_{p+1} \text{ are not twins, and } a(p + 1) < a(p) \leq q - 1, \\
        \min\{w_{p+1}, d_{\min} - w_{q-1}\} - 1 & \text{otherwise, i.e., } v_p \text{ and } v_{p+1} \text{ are not twins and } q \leq a(p). 
    \end{cases}
\end{align*}
\]

Then the set \(\{w_p, w_{p+1}, w_{p+2}, \ldots, w_{q-1}, d_{\min}, d_{\max}\}\) of weights and bounds satisfies the conditions (a)–(c) for \(p := p - 1\) and \(q\).

**Proof** We distinguish the four cases.

1. \(v_p\) and \(v_{p+1}\) are twins: Since \(v_p\) and \(v_{p+1}\) are twins and \(w_p = w_{p+1}\), we see that for each \(i \in [p + 2, q]\) with \((p, i) \in \text{green}\) (resp., \((p, i) \in \text{red}\) and \((p, i) \in \text{blue}\)), \(d_{\min} < w_p + w_i = w_{p+1} + w_i < d_{\max}\) (resp., \(w_p + w_i = w_{p+1} + w_i < d_{\min}\) and \(w_p + w_i = w_{p+1} + w_i > d_{\max}\)). To show that the conditions (a)–(c) hold for \((p := p - 1, q)\), it suffices to show that \((p, p + 1) \in \text{red}\) and \(w_p + w_{p+1} < d_{\min}\). It is impossible that \(p = i_{\text{red}}\), otherwise \(i_{\text{red}}\) would be larger than \(p\) since \(v_p\) and \(v_{p+1}\) are twins. Hence \(p < i_{\text{red}}\) and then \((p, p + 1) \in \text{red}\) by Lemma 7(1). We consider \((p + 1, p + 2) \in \text{red}\). If \((p + 1, p + 2) \not\in \text{red}\), then \(p + 1 \in \text{red}\) since \(p \leq i_{\text{red}}\). However, we have proved that \(p \neq i_{\text{red}}\). Hence \((p + 1, p + 2) \in \text{red}\). Then \(w_{p+1}\) and \(w_{p+2}\) have been determined so that \(w_{p+1} + w_{p+2} < d_{\min}\) since the condition (c) holds. Therefore \(w_p + w_{p+1} \leq w_{p+1} + w_{p+2} < d_{\min}\) since the condition (a) holds.

2. \(v_p\) and \(v_{p+1}\) are not twins and \(a(p + 1) = a(p) \leq q - 1\): Since the condition (a) holds and \(a(p) \leq q - 1\), it holds that \(w_{p+1} \leq \cdots \leq w_{a(q)-1} \leq w_{a(q)} \leq w_{q-1}\). Since \(a(p + 1) = a(p)\), we see that \((p, a(p) - 1), (p + 1, a(p) - 1) \in \text{red}\) and \((p, a(a)), (p + 1, a(p)) \in \text{green}\). Since the condition (c) holds for \((p + 1, a(p)) \in \text{green}\), we see that \(\alpha = w_{p+1} + w_{a(p)} - d_{\min}\) is positive. To show that the conditions (a)–(c) hold for \((p := p - 1, q)\), it suffices to show that \(w_p < w_{p+1}\), \(w_p + w_{a(p)} > d_{\min}\), and \(w_p + w_{a(p)-1} < d_{\min}\). Since \(\alpha > 0\), we have \(w_p = w_{p+1} - \alpha/2 < w_{p+1}\). We next see that \(w_p + w_{a(p)} = (w_{p+1} - \alpha/2) + w_{a(p)} = d_{\min} + \alpha/2 > d_{\min}\), as required. Finally, we
observe that \( w_p + w_{a(p)} - 1 < w_{p+1} + w_{a(p)+1} < d_{\min} \) since the condition (c) holds for \((p + 1, a(p) - 1) \in \text{red}\). Figure 3a illustrates this case.

(3) \( v_p \) and \( v_{p+1} \) are not twins and \( a(p + 1) < a(p) \leq q - 1 \): As in (2), we see that \( w_{p+1} \leq \cdots \leq w_{a(q)-1} \leq w_{a(q)} \leq w_{q-1} \). Since \( a(p + 1) < a(p) \), we see that \((p, a(p) - 1) \in \text{red} \) and \((p + 1, a(p) - 1) \in \text{green}\). This means that \( v_{a(p)-1} \) and \( v_{a(p)} \) are not twins, where \( w_{a(p)-1} < w_{a(p)} \) by the condition (b) and then \( \delta = w_{a(p)} - w_{a(p)-1} \) is positive. Since the condition (c) holds for \((p + 1, a(p) - 1) \in \text{green}\), we know that \( \beta = w_{p+1} + w_{a(p)-1} - d_{\min} \) is positive. To show that the conditions (a)–(c) hold for \((p := p - 1, q)\), it suffices to show that \( w_p < w_{p+1}, w_p + w_{a(p)} > d_{\min} \), and \( w_p + w_{a(p)-1} < d_{\min} \). Since \( \beta, \delta > 0 \), we have \( w_p = w_{p+1} - \beta - \delta/2 < w_{p+1} \). We next see that \( w_p + w_{a(p)} = (w_{p+1} - \beta - \delta/2) + (w_{a(p)-1} + \delta) = d_{\min} + \delta/2 > d_{\min} \), as required. Finally, we observe that \( w_p + w_{a(p)-1} = (w_{p+1} - \beta - \delta/2) + w_{a(p)-1} = d_{\min} - \delta/2 < d_{\min} \), as required. Figure 3b illustrates this case.

(4) \( v_p \) and \( v_{p+1} \) are not twins and \( q \leq a(p) \): Since \( q \leq a(p) \), we know that \((p, i) \in \text{red} \) for all \( i \in [p + 1, q - 1] \). As in (2), we see that \( w_{p+1} \leq \cdots \leq w_{q-1} \). To show that the conditions (a)–(c) hold for \((p := p - 1, q)\), it suffices to show that \( w_p < w_{p+1} \), and \( w_p + w_{q-1} < d_{\min} \). Obviously \( w_p = \min\{w_{p+1}, d_{\min} - w_{q-1}\} - 1 \leq w_{p+1} - 1 < w_{p+1} \), as required. Finally, we observe that \( w_p + w_{q-1} \leq (d_{\min} - w_{q-1} - 1) + w_{q-1} = d_{\min} - 1 < d_{\min} \), as required.

\[\square\]

**Step 3** Analogously with the process of computing weights \( w_{i_{\text{red}}} - 1, \ldots, w_i \) by Lemma 9, we compute the weights \( w_{i_{\text{blue}}} - 1, \ldots, w_i \) in this order by using the following lemma.

**Lemma 10** For a graph \( G = (V, E) \) having no isolated vertex and an ordering \( \sigma = v_1, v_2, \ldots, v_n \) of \( V \), let \( c \) be a coloring of \( G \) proper to \((G, \sigma)\). For \( p \leq i_{\text{red}} \) and \( q \geq i_{\text{blue}} \), assume that a set \{\( w_{p+1}, w_{p+2}, \ldots, w_{q-1}, d_{\min}, d_{\max} \)\} of weights and bounds satisfies the conditions (a)–(c). When \( q \leq n, \) let weight \( w_q \) be determined such that

![Fig. 3 Illustration of a process of determining weight \( w_q \) in the PCT with a center \( v^* \): a Case (2) in Lemma 9, where \( v_p \) and \( v_{p+1} \) are not twins and \( a(p + 1) = a(p) \leq q - 1 \); b Case (3) Lemma 9, where \( v_p \) and \( v_{p+1} \) are not twins and \( a(p + 1) < a(p) \leq q - 1 \). In the graph, black lines mean edges in the PCT, green lines mean green edges in the proper coloring \( c \), and red lines mean red edges in the proper coloring \( c \). Edges incident on \( p \) and \( q \) are denoted by broken lines because the weights \( w_p \) and \( w_q \) are not determined yet (Color figure online).](image-url)
Then the set \( \{w_{p+1}, w_{p+2}, \ldots, w_{q-1}, w_q, d_{\min}, d_{\max}\} \) of weights and bounds satisfies the conditions (a)–(c) for \( p \) and \( q := q + 1 \).

This lemma can be proved analogously with the proof of Lemma 9.

We are ready to prove Lemma 4. The main steps of the algorithm are as follows. Given a graph \( G = (V, E) \) with a gap-free ordering \( \sigma = v_1, v_2, \ldots, v_n \) of \( V \), we first compute a coloring \( c \) of \( G \) proper to \( (G, \sigma) \) in \( O(n^2) \) time by Lemma 5. Under the coloring \( c \), we compute indices \( a(i) \) and \( b(i) \) for \( i = 1, 2, \ldots, n \), and \( i_{\text{red}} \) and \( i_{\text{blue}} \) in \( O(n^2) \) time. Based on these, we determine weights \( w_i \) with \( i_{\text{red}} < i < i_{\text{blue}} \) in \( O(n) \) time by the method in the proof of Lemma 8. Note that whether two vertices \( v_i \) and \( v_{i+1} \) are twins can be tested in a constant time by checking whether \( a(i) = a(i + 1) \) and \( b(i) = b(i + 1) \). Finally we determine weights \( w_i \) with \( 1 \leq i \leq i_{\text{red}} \) and \( i_{\text{blue}} \leq i \leq n \) in \( O(n) \) time by Lemmas 9 and 10, respectively. This gives a PCR \( (T_v, w, d_{\min}, d_{\max}) \) of \( G \) such that \( w_1 \leq w_2 \leq \cdots \leq w_n \) in \( O(n^2) \) time. This proves Lemma 4.

We have proved Theorem 2 in this section. However, Theorem 2 and Lemma 4 are not enough to recognize star-PCGs in polynomial time. We still need to test whether or not a graph \( G = (V, E) \) allows a gap-free ordering of the vertices, which can be done by generating all the \( n! \) orderings of \( V \). However, this method is not a polynomial-time algorithm. In the next two sections, we will show that whether a graph has a gap-free ordering of vertices can be tested in polynomial time. In Sect. 4, we first introduce some properties of certain families of elements, which will be used in Sect. 5 to obtain our main results.

# 4 Consecutive/Contiguous Orderings of Elements

Let \( S \subseteq 2^V \) be a family of \( m \) subsets of a set \( V \) of \( n \geq 1 \) elements in this section. Let \( V(S) \) denote the union of all subsets in \( S \).

## 4.1 Consecutive Orderings of Elements

Recall that a total ordering \( u_1, u_2, \ldots, u_n \) of elements in \( V \) is called consecutive to \( S \) if each non-empty set \( S \in S \) consists of elements with consecutive indices. Observe that when \( S \) admits a consecutive ordering of \( V(S) \), any subfamily \( S' \subseteq S \) admits a consecutive ordering of \( V(S') \). We call a non-trivial set \( C \subseteq V \) a cut to \( S \) if no set
$S \in S$ intersects $C$, i.e., each $S \in S$ satisfies one of $S \supseteq C$, $S \subseteq C$ and $S \cap C = \emptyset$. We call $S$ cut-free if $S$ has no cut.

**Theorem 3**  For a set $V$ of $n \geq 1$ elements and a family $S \subseteq 2^V$ of $m \geq 1$ sets, a consecutive ordering of $V$ to $S$ can be found in $O(nm^2)$ time if one exists. Moreover, if $S$ is cut-free, then a consecutive ordering of $V$ to $S$ is unique up to reversal.

**Proof** First, we prove the time complexity of the theorem. A graph $H$ with $n \geq 1$ vertices is called an interval graph if each vertex $v \in V(H)$ can represented by an ordered pair $(a_v, b_v)$ of reals $a_v \leq b_v$ so that two vertices $u, v \in V(H)$ are adjacent if and only if there is a real $c$ such that $a_u \leq c \leq b_u$ and $a_v \leq c \leq b_v$. It is known that testing whether a given graph $H$ is an interval graph and finding such a representation $(a_v, b_v)$, $v \in V(H)$, if one exists can be done in $O(|V(H)| + |E(H)|)$ time [1].

Given a family $S \subseteq 2^V$, an auxiliary graph $H_S$ for $S$ is defined to be the graph $(S, E_S)$ that joins two sets $S, S' \in S$ with an edge $SS' \in E_S$ if and only if $S \cap S' \neq \emptyset$. We see that $S$ admits a consecutive ordering if and only if the auxiliary graph $H_S$ for $S$ is an interval graph by the definition of $H_S$.

The time to construct $H_S$ from $S$ is $O(nm^2)$ since we can check in $O(n)$ time whether $S \cap S' \neq \emptyset$ for two sets $S, S' \in S$, i.e., whether vertices $S, S' \in V(H)$ are adjacent in $H$. Clearly $|V(H)| = |S| = m$ and $|E(H)| \leq |S|^2 = O(m^2)$. Then testing whether $H_S$ is an interval graph and finding a representation $(a_v, b_v)$, $v \in V(H)$ can be done in $O(n + m^2)$ time. In total, it takes $O(nm^2)$ time to find a consecutive ordering for $S$, if one exists.

Next, we prove that a consecutive ordering $\sigma$ of $V$ to a cut-free family $S$ is unique up to reversal. We assume that $S$ admits another consecutive ordering $\tau \notin \{\sigma, \bar{\sigma}\}$ to derive a contradiction that $\bar{S}$ would have a cut. Since $\tau \notin \{\sigma, \bar{\sigma}\}$, some two elements $a, b \in V$ appear consecutively in $\sigma$. But another element $c \in V \setminus \{a, b\}$ appears in the subsequence $\tau_{ab}$ of $\tau$ between $a$ and $b$. Hence $a = v_k$, $b = v_{k+1}$ and $c = v_p$ for $\sigma = v_1, v_2, \ldots, v_n$, where $k + 1 < p$ is assumed without loss of generality and $c = v_p$ is chosen from $\tau_{ab}$ so that the index $p$ is maximized. We claim that the set $C = \{v_{k+1}, v_{k+2}, \ldots, v_p\}$ is a cut to $S$. Assume to the contrary that $C$ intersects a set $S \in S$, where $S = \{v_s, v_{s+1}, \ldots, v_t\}$ for $s \leq k < k + 1 \leq t < p$ or $k + 1 < s \leq p < t$ by the consecutiveness of $S$ in $\sigma$. In the former, $S$ with $a, b \in S$ contains $\tau_{ab}$, implying $c = v_p \in S$, a contradiction. In the latter, $S$ is contained in $\tau_{ab}$, since $a, b \notin S$ and $c \in S$, implying that $v_i$ with $p < t$ belongs to $\tau_{ab}$, a contradiction to the choice of $c = v_p$. So we know that $C$ is a cut to $S$. The uniqueness property holds.

### 4.2 Contiguous Orderings of Elements

Recall that a consecutive ordering $u_1, u_2, \ldots, u_n$ of elements in $V$ to $S$ is called contiguous if any two sets $S, S' \in S$ with $S' \subseteq S$ start from or end with the same element along the ordering. We call two elements $u, v \in V$ equivalent in $S$ if no set $S \in S$ satisfies $|\{u, v\} \cap S| = 1$. We call $S$ simple if there is no pair of equivalent elements.
u, v ∈ V. Define \( \mathcal{X}_S \) to be the family of maximal sets \( X \subseteq V \) such that any two vertices in \( X \) are equivalent and \( X \) is maximal subject to this property.

A non-trivial set \( S \in \mathcal{S} \) is called a separator if no other set \( S' \in \mathcal{S} \) contains or intersects \( S \), i.e., each \( S' \in \mathcal{S} \) satisfies \( S' \subseteq S \) or \( S' \cap S = \emptyset \). We call \( S \) separator-free in \( \mathcal{S} \) if \( S \) has no separator.

**Theorem 4** For a set \( V \) of \( n \geq 1 \) elements and a family \( \mathcal{S} \subseteq 2^V \) of \( m \geq 1 \) sets,

1. a contiguous ordering of \( V \) to \( S \) can be found in \( O(nm^2) \) time if one exists;
2. all elements in each set \( X \in \mathcal{X}_S \) appear consecutively in any contiguous ordering of \( V \) to \( S \);
3. if \( S \) is separator-free, then a contiguous ordering of \( V \) to \( S \) is unique up to reversal of the entire ordering and arbitrariness of orderings of elements in each set \( X \in \mathcal{X}_S \).

To prove Theorem 4, we show that an instance \((V, S)\) of the problem of finding a contiguous ordering of \( V \) to \( S \) can be modified to an instance \((V, S')\) of the problem of finding a consecutive ordering of \( V \) to \( S' \) by introducing some new subsets of \( V \).

A set \( S \in \mathcal{S} \) is called minimal (resp., maximal) if \( S \) contains no proper subset (resp., superset) of \( S \). Define \( \mathcal{S}_{\text{min}} \) (resp., \( \mathcal{S}_{\text{max}} \)) to be the family of minimal (resp., maximal) sets in \( \mathcal{S} \).

Theorem 4 follows from (4) and (5) of the next lemma.

**Lemma 11** For a set \( V \) of \( n \geq 1 \) elements, let \( \mathcal{S} \subseteq 2^V \setminus \{\emptyset\} \) be a family with \( m \geq 1 \) sets.

1. If some set \( B \in \mathcal{S}_{\text{max}} \) contains at least three sets from \( \mathcal{S}_{\text{min}} \), then \( \mathcal{S} \) cannot have a contiguous ordering of \( V \);
2. Assume that \( \mathcal{S} \) is separator-free. Let \( X \) be a maximal set of elements any two of which are equivalent. Then the elements in \( X \) appear consecutively in any contiguous ordering of \( V \) to \( S \);
3. Assume that \( \mathcal{S} \) is simple and separator-free. Let \( \mathcal{S}' \) denote the family obtained from \( \mathcal{S} \) by adding a new set \( S_{A,B} = B \setminus A \) for each pair of sets \( A \in \mathcal{S}_{\text{min}} \) and \( B \in \mathcal{S}_{\text{max}} \) such that \( A \subseteq B \) and \( B \setminus A \not\in \mathcal{S} \). Then \( \mathcal{S}' \) is cut-free and any consecutive ordering of \( V \) to \( \mathcal{S}' \) is a contiguous ordering of \( V \) to \( S \);
4. Assume that \( \mathcal{S} \) is separator-free. Then a contiguous ordering of \( V \) to \( S \) can be found in \( O(nm^2) \) time, if one exists. Moreover all elements in each set \( X \in \mathcal{X}_S \) appear consecutively in any contiguous ordering of \( V \) to \( S \), and a contiguous ordering of \( V \) to \( S \) is unique up to reversal of the entire ordering and arbitrariness of orderings of elements in each set \( X \in \mathcal{X}_S \);
5. A contiguous ordering of \( V \) to \( \mathcal{S} \) can be found in \( O(nm^2) \) time, if one exists.

**Proof** (1) Let three sets \( A_i \in \mathcal{S}_{\text{min}}, i = 1, 2, 3 \) be contained in some set \( B \in \mathcal{S}_{\text{max}} \). Note that each \( A_i \) is a proper subset of \( B \), and no set \( A_i \) is contained in any other set.
$A_j$ with $j \neq i$. Hence in any contiguous ordering of $V$ to $S$, at least one of the three sets $A_i$, $i = 1, 2, 3$ cannot share the first element in $B$ or the last element in $B$. This means that $S$ cannot have a contiguous ordering of $V$.

(2) To derive a contradiction, assume that there is a contiguous ordering $u_1, u_2, \ldots, u_n$ of $V$ to $S$ wherein the indices of elements $X$ are not consecutive, i.e., there are elements $u_i, u_j, u_k \in V$ with $i < j < k$ and a set $S \subseteq S$ such that \( \{u_i, u_j, u_k\} \cap S = \{u_j\} \). Since $S$ is separator-free, there is a set $S' \in S$ that intersects $S$ or contains $S$. Since any two elements in $X$ are equivalent, it holds that either $X \subseteq S'$ or $X \cap S' = \emptyset$. If $S'$ intersects $S$, then $S' \cap S \neq \emptyset$ means that $S' \supseteq X$ and $S'$ would contain $S$ too since the indices of elements in $S'$ are consecutive in the ordering. Hence $S'$ always contains $S$, and it also contains $X$, where $S' \setminus S \supseteq X$. Now $S$ does not contain the first element or the last element of $S'$ in the ordering. This contradicts that the ordering is contiguous to $S$.

(3) For any two sets $A \in S_{\min}$ and $B \in S_{\max}$ such that $A \subseteq B$ and $B \setminus A \neq \emptyset$, the set $B \setminus A$ must consist of elements with consecutive indices in a contiguous ordering of $V$ to $S$. Hence after adding such a set $B \setminus A$ to $S$, the contiguous ordering of $V$ to $S$ is a consecutive ordering of $V$ to $S'$. We show that any consecutive ordering of $V$ to $S'$ is a contiguous ordering of $V$ to $S'$. Assume that, for a consecutive ordering $u_1, u_2, \ldots, u_n$ of $V$ to $S'$, there are sets $X, Y \in S$ such that $X \cup Y = \{u_i, u_{i+1}, \ldots, u_{i+1}|_{i-1}\}$ but $X$ does not contain any of $u_i$ and $u_{i+1}|_{i-1}$. Let $A_X \in S_{\min}$ be a set such that $A_X \subseteq X$ and let $B_Y \in S_{\max}$ be a set such that $Y \subseteq B_Y$. Note that $S' \supseteq B_Y \setminus A_X$, where $u_i, u_{i+1}|_{i-1} \in B_Y \setminus A_X$. However $B_Y \setminus A_X$ does not consist of elements with consecutive indices in a the consecutive ordering $u_1, u_2, \ldots, u_n$, a contradiction. Hence any consecutive ordering of $V$ to $S'$ is a contiguous ordering of $V$ to $S'$.

We next prove that $S'$ is cut-free. Let $C$ be a non-trivial subset of $V$, and assume that no set $S$ intersects $C$. Since $|C| \geq 2$ and $S$ is simple, there is a set $S \subseteq S$ such that $S \cap C \neq \emptyset \neq C \setminus S$. If $S \cap C = \emptyset$, then $S$ would intersect $C$. Hence $S \setminus C$. If each set $S' \subseteq S$ with $S' \cap (V \setminus C) \neq \emptyset$ is disjoint with $C$, then this would contradict that $S$ is separator-free. Hence there is a set $S' \subseteq S$ with $S' \cap (V \setminus C) \neq \emptyset$ and $S' \supseteq C$. After the above procedure, the resulting family $S'$ contains a set $S''$ such that $S \cap S'' \neq \emptyset$ and $S'' \subseteq S \setminus S$, where $C \cap S'' \supseteq C \setminus S \neq \emptyset$, $C \cap S'' \supseteq C \setminus S'' \neq \emptyset$, and $S'' \cap C \supseteq S' \cap (V \setminus C) \neq \emptyset$. Therefore $S''$ intersects $C$. This proves that $S'$ is cut-free.

(4) Let $S_0 \subseteq 2^V$ be a given separator-free family with $m \geq 1$ subsets. First, we compute the family $\mathcal{X}_{S_0}$ of all maximal subsets. This takes $O(nm^2)$ time. By (2) of this lemma, all elements in each $X \in \mathcal{X}_{S_0}$ appear consecutively in any contiguous ordering of $V$. Next, we construct a simple family $S$ from $S_0$ as follows. For each set $X \in \mathcal{X}_{S_0}$, we choose one element $\nu_X$ from $X$ and replace $X$ with $X \setminus \{\nu_X\}$ in the family. Note that the resulting family $S$ is simple and separator-free and $|S| \leq |S_0|$. If $S_0$ admits a contiguous ordering of $V$ then so does $S$. For the family $S$, we then compute the families $S_{\min}$ and $S_{\max}$ and construct a bipartite graph $B = (S_{\min}, S_{\max}, E^*)$ such that for each pair of sets $A \in S_{\min}$, $B \in S_{\max}$, $AB \in E^*$ if and only if $A \subseteq B$. This also can be done in $O(nm^2) \cdot \Delta$ time. Now testing whether there is a set $B \in S_{\max}$ satisfying the condition (1) of this lemma can be done in $O(m^2n)$ time, because a set $B \in S_{\max}$ contains three sets in $S_{\min}$ if and only if the degree of vertex $B \in S_{\max}$ in $B$ is at least 3. When there exists such a set $B \in S_{\max}$, we conclude that $S_0$ does not admits any contiguous ordering of $V$. Assume that no set $B \in S_{\max}$ satisfies the
condition (1) of this lemma, and constructs from $S$ the cut-free family $S'$ in (3) of this lemma, where we see that $|S'| \leq |S| + 2|S_{\text{max}}| \leq 3m$. By Theorem 3 applied to $S'$, we can find a consecutive ordering $\sigma$ of $V(S')$ to the cut-free family $S'$ in $O(m^2n)$ time, if one exists, where $\sigma$ is unique up to reversal. By (3) of this lemma, the ordering $\sigma$ is contiguous to $S$. All contiguous orderings of $V$ to the input separator-free family $S_0$ can be obtained by replacing the representative element $v_X$ in $\sigma$ for each set $X \in X_{S_0}$ with an arbitrary ordering of $X$. Therefore a contiguous ordering $\sigma^*$ of $V$ to $S_0$ can be found in $O(nm^2)$ time, if one exists, and $\sigma^*$ is unique up to reversal and arbitrariness of orderings of elements in each set $X \in X_{S_0}$.

(5) Given a family $S \subseteq 2^V$, we can test in $O(nm)$ time if a set $X \in S$ is a cut of $S$, i.e., $X$ does not intersect any other set $S \in S \setminus \{X\}$. Then the family $C_S$ of all cuts of $S$ can be found in $O(nm^2)$ time, where $|C_S| \leq 2n$ since $C_S$ is a laminar.

Note that any inclusion-wise maximal set in $C_S$ is a separator of $S$. For each cut $X \in C_S$, let $\mathcal{C}(X)$ denote the family of sets $Y \in C_S$ such that $Y \subseteq X$ and $Y$ is maximal, i.e., no other set $S \in C_S$ satisfies $Y \subseteq S \subseteq X$. For each set $X \in C_S$, let $S[X]$ denote the family of sets $S \in S$ with $S \subseteq X$, and $S\langle X \rangle$ denote the family obtained from $S[X]$ by contracting each set $Y \in \mathcal{C}(X)$ into a single element $v_Y$, ignoring all sets $S \in S$ with $S \subseteq Y$, where $|V(S\langle X \rangle)| = |X| - \sum_{y \in \mathcal{C}(X)}(|Y| - 1)$ and $|S\langle X \rangle| = |S[X]| - \sum_{T \in \mathcal{C}(X)}|T[Y]|$. We easily see that $\sum_{X \in C_S} |V(S\langle X \rangle)| \leq n + |C_S| \leq 3n$, and $\sum_{X \in C_S} |S\langle X \rangle| \leq m$. Observe that, for each set $X \in C_S$, the family $S\langle X \rangle$ is separator-free, and $S\langle X \rangle$ admits a contiguous ordering of $V$ if and only if $S\langle X \rangle$ admits a contiguous ordering of $V(S\langle X \rangle)$ and $S[Y]$ for each set $Y \in \mathcal{C}(X)$ admits a contiguous ordering of $V(S[Y])$. To construct a contiguous ordering of $V$ to $S$, we choose each set $X \in C_S$ in a non-decreasing order of size $|X|$, where contiguous orderings $\sigma_{[Y]}$ for families $S[Y]$ with sets $Y \in \mathcal{C}(X)$ are available by induction. We then find a contiguous ordering $\sigma_{[X]}$ for family $S\langle X \rangle$ in $O(|V(S\langle X \rangle)||S\langle X \rangle|^2)$ time, if one exists by (4) of this lemma, and construct a contiguous ordering $\sigma_{[Y]}$ for $S\langle X \rangle$ by replacing the element $v_Y$ in $\sigma_{[X]}$ with ordering $\sigma_{[Y]}$ for each set $Y \in \mathcal{C}(X)$ in $O(n)$ time. Since $\sum_{X \in C_S} |V(S\langle X \rangle)||S\langle X \rangle|^2 = O(nm^2)$, we can find a contiguous ordering of $V$ to $S$ in $O(nm^2)$ time, if one exists. 

\subsection{5 Recognizing Star-PCGS}

Based on the properties in the above section, we next show that whether a graph has a gap-free ordering of vertices can be tested in polynomial time.

\textbf{Theorem 5} \textit{Whether a graph $G = (V, E)$ with $n$ vertices and $m$ edges has a gap-free ordering of $V$ can be tested in $O(n^3m)$ time.} 

We first give some properties of gap-free orderings in Lemma 12, based on which we will show in Lemma 13 that a graph having a gap-free ordering of vertices has at most one non-bipartite component, and a gap-free ordering for a
Lemma 12 For a graph $G = (V, E)$ with a gap-free ordering $\sigma = v_1, v_2, \ldots, v_n$ of $V$ and a coloring $c$ proper to $(G, \sigma)$, let $V_1 = \{v_i \mid 1 \leq i \leq i_{\text{red}}\}$, $V_2 = \{v_i \mid i_{\text{blue}} \leq i \leq n\}$, and $V^* = \{v_i \mid i_{\text{red}} - 1 \leq i \leq i_{\text{blue}} + 1\}$. Then

1. If two edges $v_ivj$ and $v_i'v_j'$ with $i < j$ and $i' < j'$ cross (i.e., $i < i' < j < j'$ or $i' < i < j' < j$), then they belong to the same component of $G$;
2. $G - V^*$ is a bipartite graph between vertex sets $V_1$ and $V_2$;
3. Every two vertices $v_i, v_j \in V_1 \cap N_G(V^*)$ with $i < j$ satisfy $v_{i_{\text{blue}} - 1} \in N_G(v_i) \cap V^* \subseteq N_G(v_j) \cap V^* \subseteq V^* \setminus \{v_{i_{\text{red}} + 1}\}$; and
4. Every two vertices $v_i, v_j \in V_2 \cap N_G(V^*)$ with $i < j$ satisfy $v_{i_{\text{red}} + 1} \in N_G(v_i) \cap V^* \subseteq N_G(v_j) \cap V^* \subseteq V^* \setminus \{v_{i_{\text{blue}} - 1}\}$.

Proof

1. Let edges $e = v_ivj, e' = v_i'v_j' \in E$ cross, where $i < i' < j < j'$. If $e$ and $e'$ belong to different components of $G$, then $\{v_i', v_j\}$ would be a gap with respect to edges $e$ and $e'$.
2. It can be derived from Lemma 7(2) directly.
3. We show that every two vertices $v_i, v_j \in V_1 \cap N_G(V^*)$ with $i < j$ satisfy $v_{i_{\text{blue}} - 1} \in N_G(v_i) \cap V^* \subseteq N_G(v_j) \cap V^* \subseteq V^* \setminus \{v_{i_{\text{red}} + 1}\}$ (the other case can be treated symmetrically). For this, it suffices to show that, for any vertex $v_i \in V_1$ with $N_G(v_i) \cap V^* \neq \emptyset$,

   (a) it holds $v_{i_{\text{blue}} - 1} \in N_G(v_i)$, and
   (b) for any vertex $v \in N_G(v_i) \cap V^*$ with $v_j \in V_1$ and $i < j$, it holds that $v \in N_G(v_i)$.

Fig. 4 a A disconnected graph $G = (V, E)$, b a connected non-bipartite graph $G = (V, E)$, where the edges between two vertices in $V^* = \{u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}\}$ are not depicted
In (a), otherwise \( \{ v_i, v_{i+1} \} \) would be a gap with respect to edges \( v_i v \) and \( v_{i+1} v \) for any vertex \( v \in N_G(v_i) \cap V^v \). In (b), otherwise \( \{ v_j, v \} \) would be a gap with respect to edges \( v_j v \) and \( v_{j+1} v \).

We call the complete graph \( G[V^v] \) in Lemma 12(2) the core of \( G \). Based on the next lemma, we can treat each component of a disconnected graph \( G \) separately to test whether \( G \) is a star-PCG or not. Figure 4 illustrates a disconnected PCG and a connected non-bipartite PCG.

**Lemma 13** Let \( G = (V, E) \) be a graph with at least two components.

1. If \( G \) admits a gap-free ordering of \( V \), then each component of \( G \) admits a gap-free ordering of its vertex set, and there is at most one non-bipartite component in \( G \);
2. Let \( G' = (V'_1, V'_2, E') \) be a bipartite component of \( G \), and \( G'' = G - V(G') \). Assume that \( G' \) admits a gap-free ordering \( v'_1, v'_2, \ldots, v'_{p} \) of \( V'_1 \cup V'_2 \) and \( G'' \) admits a gap-free ordering \( v_1, v_2, \ldots, v_q \) of \( V_2 \). Then there is an index \( k \) such that \( \{ \{ v'_1, v'_2, \ldots, v'_{p} \}, \{ v'_k+1, v'_k+2, \ldots, v'_{p} \} \} = \{ V'_1, V'_2 \} \). Moreover, the ordering \( v'_1, v'_2, \ldots, v'_k, v_1, v_2, \ldots, v_q, v'_k+1, v'_k+2, \ldots, v'_{p} \) of \( V \) is gap-free to \( G \).

**Proof**

1. Let \( G \) admit a gap-free ordering of \( V \). Any induced subgraph \( G \) such as a component of \( G \) is a star-PCG, and a gap-free ordering of its vertex set by Theorem 2. By Lemma 12(1), at most one component \( H \) containing a complete graph with at least three vertices can be non-bipartite, and the remaining graph \( G - V(H) \) must be a collection of bipartite graphs.
2. It follows immediately from the definition of gap-free orderings.

We first consider the problem of testing if a given connected bipartite graph is a star-PCG or not. We reduce this to the problem of finding contiguous ordering to a family of sets. For a bipartite graph \( G = (V_1, V_2, E) \) and \( i \in \{ 1, 2 \} \), define \( S_i \) to be the family \( \{ N_G(v) \mid v \in V_j \} \) for the \( \{ i, j \} = \{ 1, 2 \} \), where even if there are distinct vertices \( u, v \in V_j \) with \( N_G(u) = N_G(v) \), \( S_i \) contains exactly one set \( S = N_G(u) = N_G(v) \).

For the example of a connected bipartite graph \( G_1 = (V_1, V_2, E) \) in Fig. 1a, we have \( S_1 = \{ \{ v_3, v_4 \}, \{ v_1, v_2, v_3, v_4 \} \} \), and \( S_2 = \{ \{ v_5, v_6 \}, \{ v_5, v_6, v_7, v_8 \} \} \).

**Lemma 14** Let \( G = (V_1, V_2, E) \) be a connected bipartite graph with \( |E| \geq 1 \). Then

1. For each \( i = 1, 2 \), the family \( S_i \) is separator-free;
2. the graph \( G \) has a gap-free ordering of \( V \) if and only if for each \( i = 1, 2 \), the family \( S_i \) admits a contiguous ordering \( \sigma_i \) of \( V_i \);
3. For any contiguous ordering \( \sigma_i \) of \( V_i \), \( i = 1, 2 \), one of orderings \( (\sigma_1, \sigma_2) \) and \( (\sigma_1, \sigma_2) \) of \( V \) is a gap-free ordering to \( G \).
Proof

(1) Since \( G \) is connected, we see that, for each \( i = 1, 2 \), the family \( S_i \) is separator-free.

(2) The “only if” part: Let \( v_1, v_2, \ldots, v_k \) be a gap-free ordering of \( V \) to \( G \), where \( V_1 = \{ v_1, v_2, \ldots, v_k \} \) and \( V_2 = \{ v_{k+1}, \ldots, v_n \} \). Since there is no gap, for each vertex \( v \in V_2 \), the neighbors in \( N_G(v) \) appear consecutively as a subsequence of \( v_1, v_2, \ldots, v_k \). Also for any two vertices \( u, v \in V_2 \) such that \( N_G(u) \) is a proper subset of \( N_G(v) \), the sequence \( v_i, v_{i+1}, \ldots, v_j \) for \( N_G(u) \) must contain the first vertex or the last vertex in the subsequence \( v_{h}, v_{h+1}, \ldots, v_{p} \) for \( N_G(v) \). This is because otherwise \( v_h, v_p \not\in N_G(u) \) would imply that \( \{u, v_h\} \) (or \( \{u, v_p\} \)) is a gap with respect to edges \( e_1 = uv_i \) and \( e_1 = vv_h \) (or \( e_1 = uv_j \) and \( e_1 = vv_p \)). Therefore \( v_1, v_2, \ldots, v_k \) is a contiguous ordering of \( V_1 \) to \( S_1 \). Analogously with \( V_2 \) and \( S_2 \).

The “if” part: Assume that for each \( i = 1, 2 \), the family \( S_i \) has a contiguous ordering \( \sigma_i \) of \( V_i \). By the definition of contiguous orderings, we can easily see that in any ordering of \( (\sigma_1, \sigma_2) \) and \( (\sigma_1, \overline{\sigma_2}) \), no pair of vertices satisfies the condition (g1) of gaps. Also by the definition of contiguous orderings, we can verify that it is impossible that both of \( (\sigma_1, \sigma_2) \) and \( (\sigma_1, \overline{\sigma_2}) \) have a gap satisfying the condition (g2) or (g3). Then we know that at least one of \( (\sigma_1, \sigma_2) \) and \( (\sigma_1, \overline{\sigma_2}) \) is gap-free.

(3) By (1) we know that for each \( i = 1, 2 \), the family \( S_i \) is separator-free and then \( S = S_1 \cup S_2 \) is separator-free. Together with Theorem 4(3), we know that the contiguous ordering of \( V \) to \( S \) is unique up to reversal of the entire ordering and arbitrariness of orderings of elements in each set \( X \in \mathcal{X}_S \). By (2) and its proof, we know that if \( V_i \) for \( i \in \{1, 2\} \) allows a contiguous ordering \( \sigma_i \), then one of orderings \( (\sigma_1, \sigma_2) \) and \( (\sigma_1, \overline{\sigma_2}) \) is gap-free.

Therefore, to see if \( G \) admits a gap-free ordering of \( V \), we only need to check whether at least one of \( (\sigma_1, \sigma_2) \) and \( (\sigma_1, \overline{\sigma_2}) \) is gap-free in \( G \). Computing contiguous ordering \( \sigma_i \) of \( V(S_i) \) for each \( i = 1, 2 \) can be computed in \( O(|V(S_i)||S_i|^2) = O(n^3) \) time by Theorem 4, since \( |S_1| + |S_2| + |V(S_1)| + |V(S_2)| = O(n) \). To check whether an ordering of vertices is gap-free or not can be executed in \( O(n^2) \) time by Lemma 2. So we have that

Lemma 15 For a connected bipartite graph \( G = (V, E) \) with \( n \) vertices, we can check whether \( G \) has a gap-free ordering of \( V \) and find one if it exists in \( O(n^3) \) time.

Figure 1a illustrates an ordering \( \sigma_d = v_1, v_2, v_3, v_4, v_8, v_7, v_6, v_5 \) of \( V(G_1) \) of a connected bipartite graph \( G_1 = (V_1, V_2, E) \), where \( \sigma_d \) consists of a contiguous ordering \( \sigma_1 = v_1, v_2, v_3, v_4 \) of \( V_1 \) and a contiguous ordering \( \sigma_2 = v_8, v_7, v_6, v_5 \) of \( V_2 \). Although \( \sigma_d \) is not gap-free in \( G \), the other ordering \( \sigma_h \) of \( V(G_1) \) that consists of \( \sigma_1 \) and the reversal of \( \sigma_2 \) is gap-free, as illustrated in Fig. 1b.

Finally, we consider the case where a given graph \( G \) is a connected and non-bipartite graph. Figure 1d illustrates a connected and non-bipartite star-PCG whose maximum clique is not unique.
In a graph $G = (V, E)$, let $E^z$ denote the union of edge sets of all cycles of length 3 in $G$, $V^z$ denote the set of end-vertices of edges in $E^z$, and $N^z_G(v)$ denote the set of neighbors $u \in N_G(v)$ of a vertex $v \in V$ such that $uv \in E^z$.

**Lemma 16** For a connected non-bipartite graph $G = (V, E)$ with at least 2 vertices, if $G$ has a gap-free ordering, then

1. the set $V^z$ is not empty;
2. there are two adjacent vertices $v^*_1, v^*_2 \in V^z$, a gap-free ordering $\sigma$ of $V$, and a proper coloring $c$ to $(G, \sigma)$ such that $v^*_1 = v_{i_{\text{red}}+1}$, $v^*_2 = v_{i_{\text{blue}}-1}$.

**Proof** Let $\sigma = v_1, v_2, \ldots, v_n$ be an arbitrary gap-free ordering of $V$ and $c$ be an arbitrary coloring of $G$ proper to $(G, \sigma)$.

1. By Lemma 7(2), we know that $i_{\text{red}} + 2 \leq i_{\text{blue}}$. We claim that if $i_{\text{red}} + 2 = i_{\text{blue}}$, then $G$ is a bipartite graph with two parts of vertices being $V_r = \{v_1, v_2, \ldots, v_{i_{\text{red}}+1}\}$ and $V_b = \{v_{\text{blue}-1}, v_{\text{blue}}, \ldots, v_n\}$. By Lemma 7(2), there is no edge between any two vertices in $V_r = \{v_1, v_2, \ldots, v_{i_{\text{red}}}\}$ (resp., $V_b = \{v_{\text{blue}}, \ldots, v_n\}$). By Lemma 6, there is no edge between $v_{i_{\text{red}}}$ and $v_{i_{\text{red}}+1}$ (resp., between $v_{\text{blue}-1}$ and $v_{\text{blue}}$). By Lemma 12(3), there is no edge between $v_{i_{\text{red}}}$ and a vertex in $V_r$ (resp., between $v_{\text{blue}-1}$ and a vertex in $V_b$). So $V_r$ and $V_b$ are two independent sets and then $G$ is a bipartite graph. Since $G$ is not a bipartite graph, we know that $i_{\text{red}} + 2 < i_{\text{blue}}$, which means there are at least three vertices $v_i$ with $i_{\text{red}} < i < i_{\text{blue}}$. By Lemma 12(2), any two vertices with the index in $i_{\text{red}}$ and $i_{\text{blue}}$ are adjacent. So there are at least one triangle in $G$ and then $V^z$ is not empty.

2. By Lemma 12(2) again, all the vertices $v_i$ with $i_{\text{red}} < i < i_{\text{blue}}$ will form a clique. We have proved that $i_{\text{red}} + 2 < i_{\text{blue}}$ in (1). So $v_{i_{\text{red}}+1}$ and $v_{i_{\text{blue}}-1}$ are adjacent and contained in some triangle. Thus, $v_{i_{\text{red}}+1}, v_{i_{\text{blue}}-1} \in V^z$.

**Lemma 17** For a connected non-bipartite graph $G = (V, E)$ with $V^z \neq \emptyset$, let $v^*_1$ and $v^*_2$ be two adjacent vertices in $V^z$. Let $V^* = \{v^*_1, v^*_2\} \cup (N_G(v^*_1) \cap N_G(v^*_2))$, $V_1 = N_G(v^*_1) \setminus V^*$, and $V_2 = N_G(v^*_2) \setminus V^*$. Assume that $G$ has a gap-free ordering $\sigma$ of $V$ and a proper coloring $c$ to $(G, \sigma)$ such that $v^*_1 = v_{i_{\text{red}}+1}$ and $v^*_2 = v_{i_{\text{blue}}-1}$. Then:

1. A maximal clique in $G$ that contains edge $v^*_1v^*_2$ is uniquely given as $G[V^*]$. The graph $G[V^*]$ is the core of the ordering $\sigma$, and $G - V^*$ is a bipartite graph $(V_1, V_2, E^*)$;
2. Let $S_i$ denote the family $\{N_G(v) \mid v \in V_j\}$ for $\{i, j\} = \{1, 2\}$, and $S = S_1 \cup S_2 \cup \{V^*\}$. Then $S$ is a separator-free family that admits a contiguous ordering $\sigma$ of $V$, and any contiguous ordering $\sigma$ of $V$ is a gap-free ordering to $G$.

 Springer
Proof

(1) By Lemma 12(3), we see that \{v_i \mid i_{\text{red}} + 1 < i < i_{\text{blue}} - 1\} \subseteq N_G(v_{i_{\text{red}}+1}) \cap N_G(v_{i_{\text{blue}}-1}). On the other hand, by Lemma 12(3) again, vertex \(v_{i_{\text{red}}+1}\) (resp., \(v_{i_{\text{blue}}-1}\)) is not adjacent to any vertex in \(V_2\) (resp., \(V_1\)) in \(G\). Hence \(v_{i_{\text{red}}+1}, v_{i_{\text{blue}}-1}\) \(\cup (N_G(v_{i_{\text{red}}+1}) \cap N_G(v_{i_{\text{blue}}-1}))\) induces uniquely a maximal clique that contains \(v_{i_{\text{red}}+1}\) and \(v_{i_{\text{blue}}-1}\). Hence the clique is the core of the gap-free ordering of \(V\). By Lemma 12(2), \(G - V^*\) is a bipartite graph \((V_1, V_2, E')\).

(2) Since \(G\) is connected, we see that \(S_i\) is separation-free. First, we prove that \(S\) admits a contiguous ordering of \(V\). Any set \(S \in \mathcal{S}\) with \(S \cap V^* \neq \emptyset\) satisfies one of the following:

- \(v_{i_{\text{red}}+1} \not\in S\) and \(v_{i_{\text{blue}}-1} \in S \supset S_2\),
- \(v_{i_{\text{blue}}-1} \not\in S\) and \(v_{i_{\text{red}}+1} \in S \supset S_1\), and
- \(v_{i_{\text{red}}+1}, v_{i_{\text{blue}}-1} \in S \equiv V^*\).

This means that any two sets \(S, S' \in \mathcal{S}\) with \(S \subseteq S'\) belong to the same family \(S_j\). Hence any gap-free ordering \(\sigma\) of \(V\) to \(G\) is a contiguous ordering of \(V\) to \(S\), as discussed in the proof of the “only if” part of Lemma 14.

Next we prove that any contiguous ordering \(\sigma\) of \(V\) to \(S\) is a gap-free ordering of \(V\) in \(G\). Since \(G\) is connected, each \(S_i\) is separator-free in \(V(S_i)\). We see that \(S = S_1 \cup S_2 \cup \{V^*\}\) is separator-free in \(V\) even if \(V(S_1) \cap V(S_2) \neq \emptyset\). Note that any set \(X \in \mathcal{X}_S\) is either contained in \(N_G(v)\) or disjoint with \(N_G(v)\) for each vertex \(v \in V_j, j \neq i\). By Theorem 4 applied to \(S\), the vertices in each maximal set \(X \in \mathcal{X}_S\) appear consecutively in any contiguous ordering of \(V(S)\). Also a contiguous ordering of \(V(S)\) is unique up to reversal and choice of an ordering of each set \(X \in \mathcal{X}_S\). This means that an ordering \(\sigma\) of \(V\) is gap-free if and only if any ordering obtained from \(\sigma\) by changing an ordering of vertices in each set \(X \in \mathcal{X}_S\). Therefore any contiguous ordering \(\sigma\) of \(V\) to \(S\) is a gap-free ordering of \(V\) in \(G\).

For example, when we choose vertices \(v^*_1 = u_7\) and \(v^*_2 = u_{14}\) in the connected non-bipartite graph \(G = (V, E)\) in Fig. 4b, we have \(V^* = \{u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}\}, S_1 = \{u_1, u_2\}, \{u_2, u_3, u_4\}, \{u_3, u_4, u_5\}, \{u_3, u_4, u_5, u_6, u_7, u_8\}, \{u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}\}\), and \(S_2 = \{u_{19}\}, \{u_{18}, u_{19}\}, \{u_{16}, u_{17}, u_{18}\}, \{u_{13}, u_{14}, u_{15}, u_{16}, u_{17}\}, \{u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_{16}\}\).

Lemma 18 For a connected non-bipartite graph \(G = (V, E)\) with \(n\) vertices and \(m\) edges, we can check whether \(G\) has a gap-free ordering of \(V\) and find one if it exists in \(O(n^3 m)\) time.

The correctness of this lemma is based on the following observation. We compute the set \(V^*\) in Lemma 17 in \(O(n^3)\) time. If \(V^*\) is empty, then the graph has no
gap-free ordering of vertices according to Lemma 16(1). Otherwise, for each pair of connected vertices in $V^c$, we do the following steps:

1. Compute $V^c$, $V_1$, $V_2$, $S_1$, $S_2$ and $S$ in Lemma 17 in $O(n^2)$ time;
2. Compute a contiguous ordering of $V$ (if one exists) in $O(|V(S)| |S|^2) = O(n^3)$ time by Theorem 4 since $|S| + |V(S)| = O(n)$ holds;
3. Test whether the contiguous ordering of vertices is gap-free or not in $O(n^2)$ time by Lemma 2.

If after checking all pairs of connected vertices in $V^c$ we do not find any gap-free ordering of vertices, then the graph has no gap-free ordering of vertices by Lemmas 16 and 17. The above procedure runs in $O(n^3m)$ time since there are $O(m)$ pairs of connected vertices in $V^c$.

Now we are ready to prove Theorem 5. The main steps of the algorithm are as follows. First, we test in linear time whether each connected component of the graph $G$ is bipartite or not. If at least two connected components are non-bipartite, then $G$ has no gap-free ordering of the vertices [by Lemma 13(1)]. Otherwise, we will test whether each connected component has a gap-free ordering of vertices. For each bipartite component, we find a gap-free ordering of vertices if one exists in $O(n^3)$ time by Lemma 15. For each non-bipartite component, we find a gap-free ordering of vertices if one exists in $O(n^3m)$ time by Lemma 18. If one connected component has no gap-free ordering of vertices, then the graph has no gap-free ordering of the vertices by Lemma 13(1). Otherwise, we can construct a gap-free ordering of the vertices for the whole graph in linear time by using the method in Lemma 13(2). Then we can find a gap-free ordering of a given graph (if one exists) in $O(n^3m)$ time, proving Theorem 5.

By Theorem 2, Lemma 4 and Theorem 5, we get Theorem 1.

6 Concluding Remarks

Pairwise compatibility graphs were initially introduced from the context of phylogenetics in computational biology and later became an interesting graph class in graph theory. PCG recognition is a hard task and we are still far from a complete characterization of PCG. Significant progress toward PCG recognition would be interesting from a graph theory perspective and also help design sampling algorithms for phylogenetic trees. In this paper, we give the first polynomial-time algorithm to recognize star-PCGs. Although stars are trees of a simple topology, it is not an easy task to recognize star-PCGs. To do so, we need to develop several structural properties of this problem. We first show that three cases of non-adjacent vertex pairs (called gaps) under a fixed ordering of vertices can be used to characterize star-PCGs and the gaps in a graph can be tested in polynomial time. Then we show that we only need to test a polynomial number of orderings of vertices and thus we can get a polynomial-time algorithm. For further study, it is an interesting topic to study the characterization of PCGs with witness trees of other particular topologies.
Acknowledgements The work was supported by the National Natural Science Foundation of China, under Grants 61972070 and 61772115.

References

1. Booth, S., Lueker, S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. J. Comput. Syst. Sci. 13, 335–379 (1976)
2. Brandstädt, A.: On leaf powers. Technical report, University of Rostock (2010)
3. Calamoneri, T., Frascaria, D., Sinaimeri, B.: All graphs with at most seven vertices are pairwise compatibility graphs. Comput. J. 56(7), 882–886 (2013)
4. Calamoneri, T., Frangioni, A., Sinaimeri, B.: Pairwise compatibility graphs of caterpillars. Comput. J. 57(11), 1616–1623 (2014)
5. Calamoneri, T., Petreschi, R.: On pairwise compatibility graphs having Dilworth number two. Theor. Comput. Sci. 524, 34–40 (2014)
6. Calamoneri, T., Petreschi, R.: On pairwise compatibility graphs having Dilworth number k. Theor. Comput. Sci. 547, 82–89 (2014)
7. Calamoneri, T., Petreschi, R., Sinaimeri, B.: On the pairwise compatibility property of some superclasses of threshold graphs. Discrete Math. Algorithms Appl. 5(2), 360002 (2013)
8. Calamoneri, T., Sinaimeri, B.: Pairwise compatibility graphs: a survey. SIAM Rev. 58(3), 445–460 (2016)
9. Chen, Z.-Z., Jiang, T., Lin, G.: Computing phylogenetic roots with bounded degrees and errors. SIAM J. Comput. 32, 864–879 (2003)
10. Durocher, S., Mondal, D., Rahman, MdS: On graphs that are not PCGs. Theor. Comput. Sci. 571, 78–87 (2015)
11. Felsenstein, J.: Cases in which parsimony or compatibility methods will be positively misleading. Syst. Zool. 27, 401–410 (1978)
12. Hossain, MdI, Salma, S.A., Rahman, MdS, Mondal, D.: A necessary condition and a sufficient condition for pairwise compatibility graphs. J. Graph Algorithms Appl. 21(3), 341–352 (2017)
13. Kearney, P.E., Munro, J.I., Phillips, D.: Efficient generation of uniform samples from phylogenetic trees. In: Benson, G., Page, R. (eds.) Algorithms in Bioinformatics, LNCS 2812, pp. 177–189. Springer, Berlin (2003)
14. Lin, G., Kearney, P.E., Jiang, T.: Phylogenetic k-root and Steiner k-root. In: Goos, G., Hartmanis, J., van Leeuwen, J., Lee, D.T., Teng, S.-H. (eds.) Algorithms and Computation, LNCS 1969, pp. 539–551. Springer, Berlin (2000)
15. Mehnaz, S., Rahman, M.S.: Pairwise compatibility graphs revisited. In: Proceedings of the 2013 International Conference on Informatics, Electronics Vision (ICIEV), pp. 1–6 (2013)
16. Nishimura, N., Ragde, P., Thilikos, D.M.: On graph powers for leaf-labeled trees. J. Algorithms 42, 69–108 (2002)
17. Salma, S.A., Rahman, MdS, Hossain, MdI: Triangle-free outerplanar 3-graphs are pairwise compatibility graphs. J. Graph Algorithms Appl. 17, 81–102 (2013)
18. Xiao, M., Nagamochi, H.: Some reduction operations to pairwise compatibility graphs. Inf. Process. Lett. 153, 105875 (2020)
19. Yanhaona, M.N., Bayzid, MdS, Rahman, MdS: Discovering pairwise compatibility graphs. Discrete Math. Algorithm Appl. 2(4), 607–624 (2010)
20. Yanhaona, M.N., Hossain, K.S.M.T., Rahman, MdS: Pairwise compatibility graphs. J. Appl. Math. Comput. 30, 479–503 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.