J Braz Comput Soc
DOI 10.1007/s13173-012-0064-8

SI: GRAPHCLIQUES

Gene clusters as intersections of powers of paths

Vítor Costa · Simone Dantas · David Sankoff · Ximing Xu

Received: 31 January 2012 / Accepted: 1 February 2012
© The Brazilian Computer Society 2012

Abstract There are various definitions of a gene cluster determined by two genomes and methods for finding these clusters. However, there is little work on characterizing configurations of genes that are eligible to be a cluster according to a given definition. For example, given a set of genes in a genome, is it always possible to find two genomes such that their intersection is exactly this cluster? In one version of this problem, we make use of the graph theory to reformulated it as follows: Given a graph G with n vertices, do there exist two θ-powers of paths $G_S = (V_S, E_S)$ and $G_T = (V_T, E_T)$ such that $G_S \cap G_T$ contains G as an induced subgraph? In this work, we divide the problem in two cases, depending on whether or not G is an induced subgraph of G_S or G_T. We show an $\mathcal{O}(n^2)$ time algorithm that generates the smallest θ-powers of paths G_S and G_T (with respect to and the number of vertices) that contains G as an induced subgraph. Finally, we discuss the problem when G is an induced subgraph neither of G_S nor of G_T and we present a method of finding the smallest power of a path when graph G is a cycle C_n.

V. Costa · S. Dantas (✉)
Instituto de Matemática e Estatística, Universidade Federal Fluminense, 24.020-140, Niterói, Brazil
e-mail: sdantas@im.uff.br

V. Costa
e-mail: vitorsilcost@mat.uff.br

D. Sankoff
Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada
e-mail: sankoff@uottawa.ca

X. Xu
Department of Statistics, University of Toronto, Toronto, Canada
e-mail: ximing@utstat.utoronto.ca

Published online: 23 February 2012

Keywords Power of a path · Unit interval graph · Genome · Gene clusters

1 Introduction

Due to recent research on genetic mapping, a large amount of information is available and stored in databases of various research centers in the world. Processing these data, in order to obtain relevant biological conclusions, is one of the challenges in biology. One way to structure these data is using comparison of genomes, i.e., the search for similarities and differences between two or more organisms. The central question of this paper proposes to deal with a problem in this area by asking: given a set of genes in a genome, called *cluster*, is it always possible to find two genomes such that their intersection is exactly this cluster? First, we show the modeling presented by Adam et al. [1] and Sankoff and Xu [8], which will be used in this paper.

A *marker* is a gene with a known location on a chromosome. Let V_X be the set of n markers in the genome X. These markers are partitioned among a number of total orders called *chromosomes*. For markers g and h in V_X on the same chromosome in X, let $gh \in E_X$ if the number of genes intervening between g and h in X is less than θ, where $\theta \geq 1$ is a fixed *neighborhood parameter*. We call $G_X = (V_X, E_X)$ a θ*-adjacency graph* if its edges are determined by a neighborhood parameter θ.

Consider the θ-adjacency graphs $G_S = (V_S, E_S)$ and $G_T = (V_T, E_T)$ with a non-null set of vertices in common $V_{ST} = V_S \cap V_T$. We say that a subset of $V \subseteq V_{ST}$ is a *generalized adjacency cluster* if it consists of vertices of a maximal connected subgraph of $G_{ST} = (V_{ST}, E_S \cap E_T)$. We call $G = G_{ST}[V]$ the subgraph induced by set V.

Let $G = (V(G), E(G))$ be a graph with vertex set $V(G)$ and edge set $E(G)$, such that $|V(G)| = n$. Let $v, \bar{v} \in V(G)$. The *distance* between vertices v and $\bar{v}$, denoted by $d_G(v, \bar{v})$, is the number of edges in a shortest path between v and $\bar{v}$ in G. A *path* between two vertices v_0 and v_t of graph G is a sequence of vertices $v_1, v_2, \dots, v_t$ such that $v_i v_{i+1}$ is an edge of G, $1 \le i \le t-1$. Let P_n be a graph that is a path with n vertices. A *θ-power of a path* P_{n_θ}, denoted by $P^{\theta}_{n_\theta}$, $\theta > 0$, is graph such that $V(P^{\theta}_{n_\theta}) = V(P_{n_\theta})$ and $E(P^{\theta}_{n_\theta}) = \{v\bar{v} : d_{P_{n_\theta}}(v, \bar{v}) \le \theta \text{ with } v, \bar{v} \in V(P^{\theta}_{n_\theta})\}$. For the benefit of the reader, we denote the power of a path $P^{\theta}_{n_\theta}$ by P^{θ}. The definition of a chromosome with n_θ markers in a θ-adjacency graph is similar to a power of a path $P^{\theta}_{n_\theta}$. Now, the central question of this work can be reformulated as follows:

Question 1 ([2, 5]) Given a connected graph G, do there exist G_S and G_T, two θ-powers of paths P_S and P_T, whose intersection contains G as an induced subgraph?

If the answer is yes, we are also interested in finding the minimum value of power θ and number vertices n_θ for these two θ-powers of paths.

In order to contribute to this challenging problem, we divide our study in two cases, depending on whether or not G is an induced subgraph of G_S or G_T. First, we give some definitions. We say that G is an *unit interval graph* if there exists a family I of intervals (a, b) on the real line such that each $v \in V(G)$ can be put in a one-to-one correspondence with $(a_v, b_v) \in I$; the intervals in I are of same length; and $v\bar{v}$ is a edge of $E(G)$ if, and only if, $(a_v, b_v) \cap (a_{\bar{v}}, b_{\bar{v}}) \ne \emptyset$. This family of intervals is called an *interval model* for G. Lin et al. [6] and Soulignac [9] present a proof that the class of proper interval graphs precisely the class of unit interval graphs. There exist linear-time recognition algorithms for unit interval graphs, for example Figueiredo et al. [4] and Corneil et al. [3].

Brandstädt et al. [2] and Lin et al. [5] proved independently the following structural property:

Theorem 1 ([2, 5]) *A graph G is an induced subgraph of a power of a path if, and only if, G is an unit interval graph.*

Thus, given an unit interval graph G with n vertices, there exists a θ-power of a path P_{n_θ} that contains G as an induced subgraph. But the proofs of the structural characterization given by Theorem 1 [2, 5] does not lead to an algorithm that constructs G_S and G_T for Question 1 with minimum value of power θ and number vertices n_θ.

In the paper [6], the authors show an $\mathcal{O}(n)$ time algorithm that includes new intervals into a proper interval model I of a connected graph G, constructing an extended model I' containing I. This extended model I' gives an implicit representation of a power of a path for all proper interval graph G, but the number of inserted intervals, or the size of the power θ, cannot be minimum. The authors also remark that any explicit representation would require $\mathcal{O}(n^2)$ steps.

We present in this work an $\mathcal{O}(n^2)$ time algorithm that generates, from a connected unit interval graph G, an explicit representation of the smallest θ-power of path, G_S (with respect to θ and to the number of vertices), that contains G as an induced subgraph. Next, we construct G_T, a θ-power of a path with the same number of vertices of G_S, such that the intersection $G_S \cap G_T$ contains G as an induced subgraph.

This paper is organized as follows. In Sects. 2 and 3, we present the algorithm and we prove its correctness and complexity. In Sect. 4, we discuss the problem when G is an induced subgraph neither of G_S nor of G_T and we present a method of finding the smallest power of a path when graph G is a cycle C_n.

2 The algorithm

Our result is based on the ordering of the vertex set of G, given by Algorithm *Recognize* [3], which satisfies the property proved by Roberts in [7]:

Property 2 *A graph G is an unit interval graph if and only if there is an order $<$ on vertices such that for all vertices v, the closed neighborhood of v is a set of consecutive vertices with respect to the order $<$.*

Since all powers of paths are unit interval graphs, we can insert the vertices of $V(G)$ in the vertex set of a power of a path $P^{\theta}_{n_\theta}$ until this power of a path contains G as an induced subgraph.

This construction is done by Algorithm *CPP* as follows. First, let $v_1 < v_2 < \cdots < v_n$ be an ordering of $V(G)$ given by Algorithm *Recognize* [3]. We consider θ_0 as the number of vertices of the maximal clique that contains v_1, minus one; and we insert the vertices of this clique in P^{θ_0}. The Algorithm *CPP* constructs a sequence of power of a paths $P^{\theta_0} \subset P^{\theta_1} \subset \cdots \subset P^{\theta_{l-1}} \subset P^{\theta_l}$ such that $\theta_i = \theta_{i-1} + 1$.

Let v be the first vertex non-adjacent to v_1 in the order on $V(G)$. If v is adjacent to v_2, Algorithm *CPP* must insert v in the vertex of P^{θ_0} that is at distance $\theta_0 + 1$ from vertex v_1 in P^{θ_0}. Similarly, if v is not adjacent to v_t, but is adjacent to v_{t+1}, Algorithm *CPP* must insert v in the vertex of P^{θ_0} that is at a distance $\theta_0 + 1$ from vertex v_t in P^{θ_0}. This is done by inserting $t - 1$ vertices between the vertex of largest index adjacent to v_1 and v in P^{θ_0}. Now, suppose that there exist at least two vertices $v, \bar{v}$ that are not adjacent to v_1 and adjacent to v_2. Let $\bar{v}$ be the second vertex of this set. In order to minimize the number of vertices of P^{θ_0}, vertex $\bar{v}$ must be a vertex of P^{θ_0} at distance $\theta_0 + 2$ of vertex v_1 in P^{θ_0}. Then Algorithm *CPP* must call Procedure *SHIFT* to increase θ_0 to $\theta_1 := \theta_0 + 1$ because of the edge $\bar{v}v_2$. On the other hand, this increase adds several edges in P^{θ_0} which are not in $E(G)$. Thus, Procedure *SHIFT* adjusts the power of a path P^{θ_0} for the new θ_1, by inserting vertices in P^{θ_0} in order to preserve the adjacencies and non-adjacencies between vertices of G and generates a new P^{θ_1}. Algorithm *CPP* proceeds until all vertices of $V(G)$ are included in $P^{\theta}_{n_\theta}$, a smallest power of a path with respect to θ and n_θ.

Before describing Algorithm *CPP*, we borrow some definitions from [3]. Given an ordering of $V(G)$ returned by Algorithm *Recognize* [3], then $\mathrm{order}_G(v)$ is the position of vertex v considering this ordering; $\xi_G(v) = \max\{\mathrm{order}_G(\overline{v}) : \bar{v} \in N_G[v]\}$ and $\eta_G(v) = \min\{\mathrm{order}_G(\bar{v}) : \bar{v} \in N_G[v]\}$, where $N_G[v] = \{w \in V(G) : vw \in E(G)\} \cup \{v\}$. Let $v \in V(G)$ and $u \in V(P^\theta)$. We refer to $\mathrm{order}_{P^\theta}(v)$ as the position of vertex v in the ordering of the vertex set of P^θ, i.e., $\mathrm{order}_{P^\theta}(v) = i$, if $u_i = v$ in P^θ. We denote $\xi_{P^\theta}(u) = \max\{\mathrm{order}_{P^\theta}(\bar{u}) : \bar{u} \in N_{P^\theta}[u]\}$ and $\eta_{P^\theta}(u) = \min\{\mathrm{order}_{P^\theta}(\bar{u}) : \bar{u} \in N_{P^\theta}[u]\}$.

Next, we present Algorithm *CPP* and Procedure *SHIFT*.

Algorithm *CONSTRUCTING_POWER_OF_PATH(CPP)*

- Input: a connected unit interval graph G and an ordering of $V(G)$, $v_1 < \cdots < v_n$, given by Algorithm *Recognize* [3].
- Output: a smallest power of a path, $P^{\theta}_{n_\theta}$, with respect to θ and to number of the vertices n_θ, that contains G as an induced subgraph.

1. $\theta := \xi_G(v_1) - 1$.
2. $P^\theta := (u_1, u_2, \ldots, u_{\theta(n-1)}, u_{\theta(n-1)+1})$ null-vector.
3. For $j := 1$ to $\xi_G(v_1)$ do
   $u_j := v_j$.
4. For $i := 1$ to $\eta_G(v_n) - 1$ do
   For $j := 1$ to $\xi_G(v_{i+1}) - \xi_G(v_i)$ do
   $u_{\mathrm{order}_{P^\theta}(v_i)+\theta+j} := v_{\xi_G(v_i)+j}$.
   If $|\mathrm{order}_{P^\theta}(v_{\xi_G(v_i)+j}) - \mathrm{order}_{P^\theta}(v_{i+1})| > \theta$ then
   $SHIFT(P^\theta[u_1, u_2, \ldots, u_{\mathrm{order}_{P^{\theta-1}}(v_i)+\theta+j}])$.
5. Return $P^\theta := (u_1, u_2, \ldots, u_{\mathrm{order}_{P^\theta}(v_n)})$.

Procedure *SHIFT* receives as input a smallest power of a path P^θ that contains $G[v_1, \ldots, v_{l-1}]$, $\xi_G(v_1) + 1 \le l \le n$ as an induced subgraph in P^θ. Power P^θ contains the last vertex v_l inserted by Algorithm *CPP*. Vertex v_l raises Procedure *SHIFT* because v_l is not adjacent to some vertex v_{l-t} in P^θ, but $v_{l-t}v_l \in E(G)$.

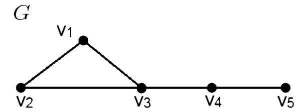

Fig. 1 Algorithm *CPP* returns the 2-power of path $P_6 = v_1, v_2, v_3, 0, v_4, v_5$ for unit interval graph G

Procedure *SHIFT*

- Input: a smallest power of a path P^θ that contains $G[v_1, \ldots, v_{l-1}]$ as an induced subgraph.
- Output: a smallest power of a path, $P^{\theta+1}$, that contains $G[v_1, \ldots, v_l]$ as an induced subgraph.

1. $\theta := \theta + 1$.
2. $P^\theta := (w_1, w_2, \ldots, w_{\theta(l-1)+1})$ null-vector.
3. $k := \max\{\mathrm{order}_{P^{\theta-1}}(v) : \mathrm{order}_{P^{\theta-1}}(v) < \eta_{P^{\theta-1}}(u_{n_{\theta-1}}) - 1,\ v \in V(G)\}$
   $s := \min\{t \ge 1 : t \equiv k \bmod \theta\}$.
4. For $j := 1$ to s do
   $w_j := u_j$.
5. For $j := s + 1$ to $k + 1$ do
   If $j \equiv (s + 1) \bmod \theta$
   then $w_{\mathrm{order}_{P^\theta}(u_{j-1})+2} := u_j$;
   else $w_{\mathrm{order}_{P^\theta}(u_{j-1})+1} := u_j$.
6. For $j := k + 2$ to $n_{\theta-1}$ do
   $w_{\mathrm{order}_{P^\theta}(u_{j-1})+1} := u_j$.
7. Return P^θ.

Algorithm *CPP* returns $P^\theta_{n_\theta}$, the smallest power of a path (with respect to θ and n_θ) that contains G as an unit interval graph. We construct two powers of paths, $G_T = (V_T, E_T)$ and $G_S = (V_S, E_S)$, from $P^\theta_{n_\theta}$ as follows. First, $V_T = V_S = V(P^\theta_{n_\theta})$. Then, vertices of V_T, which are not in V, receive different labels from vertices in $V(P^\theta_{n_\theta})$.

We show an example of an unit interval graph G in Fig. 1. For this graph G, Algorithm *CPP* returns G_S, the 2-power of path $P_S = v_1, v_2, v_3, 0, v_4, v_5$. Then, G_T is a 2-power of path $P_T = v_1, v_2, v_3, v_b, v_4, v_5$.

3 Proofs

In this section, we present the proofs of correctness of the Procedure *SHIFT* (Lemma 1) and Algorithm *CPP* (Theorem 4).

Lemma 1 *Let P^θ be a smallest power of a path that contains $G_{l-1} = G[v_1, \ldots, v_{l-1}]$ as an induced subgraph, with respect to the ordering $v_1 < \cdots < v_{l-1}$. Let $v_l \in V(G)$ be the next vertex inserted in P^θ and $v_{l-t-1}v_l \notin E(G)$, $v_{l-t}v_l \in E(G)$ and $d_{P_{n_\theta}}(v_{l-t}, v_l) = \theta + 1$. Then, the output of the Procedure* SHIFT, *the power of a path $P^{\theta+1}$, is a smallest power of a path that contains $G_l = G[v_1, \ldots, v_{l-1}, v_l]$ as an induced subgraph, with respect to the ordering $v_1 < \cdots < v_{l-1} < v_l$.*

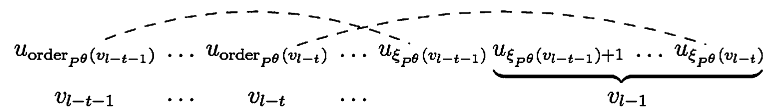

Fig. 2 Bracket indicates possible positions of v_l

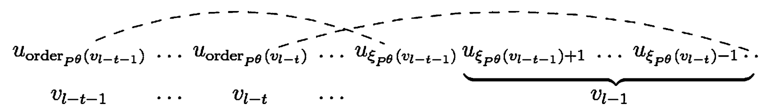

Fig. 3 Bracket indicates possible positions of v_{l-1}

Proof Since $v_{l-t-1}v_l \notin E(G)$, $v_{l-t}v_l \in E(G)$ and $\theta + 1 = d_{P_{n_\theta}}(v_{l-t}, v_l)$, the Procedure *SHIFT* must increase the power θ by one unit (Step 1). But the increase of θ to $\theta+1$ creates several adjacencies in P^θ between pairs of vertices of the set $\{v_1, \ldots, v_l\}$ that are non-adjacent in G. In order to preserve the adjacencies and non-adjacencies between vertices of G in P^θ, Procedure *SHIFT* is forced to insert one vertex between the vertex that received v_{l-t-1} in P^θ and its consecutive vertex in P^θ. Again, counting in descending order from vertex v_{l-t-1}, the adjacencies were violated in each "block" of θ vertices in P^θ. So, the procedure must insert one vertex to each $\theta+1$ vertices in descending order, from vertex v_{l-t-1} in P^θ. We observe that the set formed by the initial vertices of $V(P^\theta)$ has cardinality less than or equal to $\theta+1$, because dividing $\mathrm{order}_{P^\theta}(v_{l-t-1})$ by $\theta+1$ the remainder is greater than or equal to 1 and less than or equal to $\theta+1$.

In each step, the procedure inserts the smallest number of vertices necessary to guarantee that the power of a path $P^{\theta+1}$, created by Procedure *SHIFT*, contains $G_l[v_1, \ldots, v_l]$ as an induced subgraph. So, the power $\theta+1$ and the number of inserted vertices are minimum and, consequently, $P^{\theta+1}$ is a smallest power of a path that contains $G_l[v_1, \ldots, v_l]$ as an induced subgraph. □

First, we prove that Algorithm *CPP* correctly returns a smallest power of a path according to the ordering given by Algorithm *Recognize* [3].

Lemma 2 *Let G be a connected unit interval graph. Algorithm* CPP *generates the smallest power of a path $P^\theta_{n_\theta}$, with respect to θ and n_θ, that contains G as an induced subgraph according to the ordering $v_1 < \cdots < v_n$ given by the input of* CPP.

Proof Algorithm *CPP* constructs a sequence of powers of paths $P^{\theta_0} \subseteq P^{\theta_1} \subseteq \cdots \subseteq P^\theta$, where $\theta_i = \theta_{i-1} + 1$. This is done by successively adding, in each P^{θ_i}, vertices of G following the input ordering, preserving the adjacencies and non-adjacencies between vertices of G and minimizing θ and n_θ. Initially, the power of a path P^{θ_0} receives the maximal clique containing v_1, i.e., $V(P^{\theta_0}) = \{u_1, \ldots, u_{\xi_{P^{\theta_0}}(v_1)}\}$ and $\theta_0 = \xi_G(v_1) - 1$. This is the smallest power of a path that contains $G[v_1, \ldots, v_{\xi_G(v_1)}]$ as an induced subgraph.

Suppose that the $l-1$ first vertices, i.e., $\{v_1, \ldots, v_{l-1}\}$, were already been inserted by Algorithm *CPP* in the power of a path P^θ, i.e., P^θ is the smallest power of a path, with respect to θ and n_θ that contains $G[v_1, \ldots, v_{l-1}]$ as an induced subgraph. Let $v_l \in V(G)$ the next vertex to be inserted by Algorithm *CPP* in P^θ. Suppose that v_l is adjacent, in P^θ, to $\{v_{l-t}, \ldots, v_{l-1}\}$. Vertex v_l must be inserted in P^θ between positions $\xi_{P^\theta}(v_{l-t-1}) + 1$ and $\xi_{P^\theta}(v_{l-t})$ so that $G[v_1, \ldots, v_l]$ be an induced subgraph of P^θ. Then, $d_{P_{n_\theta}}(v_{l-t-1}, v_l) \geq \theta + 1$ and $d_{P_{n_\theta}}(v_{l-t}, v_l) \leq \theta$. We consider two cases with respect to the adjacencies of v_l in G. From now on, we refer to Fig. 2, and Fig. 3 and Fig. 4, where dashed lines represent adjacencies.

Case 1: If $t = \theta$, then after insertion of v_l, $\theta + 1 \leq d_{P_{n_\theta}}(v_{l-t-1}, v_l)$, because the set $\{v_{l-t}, \ldots, v_{l-1}\}$ has $t = \theta$ elements (see Fig. 2). In order to minimize θ and n_θ, Algorithm *CPP* must insert v_l in the consecutive vertex to v_{l-1} in the power of a path P^θ, and as a consequence $d_{P_{n_\theta}}(v_{l-t}, v_l) \leq \theta + 1$. In effect, since v_{l-1} is adjacent to v_{l-t} in P^θ, by hypothesis, v_{l-1} was inserted in P^θ such that $d_{P_{n_\theta}}(v_{l-t}, v_{l-1}) \leq \theta$ and v_l was inserted in the consecutive vertex to v_{l-1} in P^θ, then the claim is true. If $d_{P_{n_\theta}}(v_{l-t}, v_l) \leq \theta$, Algorithm *CPP* inserted v_l without changing θ, the number of vertices of P^θ became $n_\theta + 1$, and so this insertion was minimum. If $d_{P_{n_\theta}}(v_{l-t}, v_l) = \theta + 1$, Algorithm *CPP* called the Procedure *SHIFT* and, by Lemma 1, we conclude the proof.

Case 2: If $1 < t < \theta$, Algorithm *CPP* must insert v_l in P^θ such that $d_{P_{n_\theta}}(v_{l-t-1}, v_l) \geq \theta + 1$ so that v_{l-t-1} and v_l are not adjacent. We observe the position of v_{l-1} in P^θ. If v_{l-1} is not adjacent to v_{l-t-1} in P^θ (see Fig. 3), in order to minimize the number of vertices of P^θ, Algorithm *CPP* inserts v_l in the consecutive vertex to v_{l-1}. By hypothesis, vertex v_{l-1} was inserted in P^θ so that $d_{P_{n_\theta}}(v_{l-t}, v_{l-1}) \leq \theta$. Then, if $d_{P_{n_\theta}}(v_{l-t}, v_{l-1}) < \theta$, we have $d_{P_{n_\theta}}(v_{l-t}, v_l) \leq \theta$. Thus v_l was inserted in P^θ without changing θ, the number of vertices of P^θ became $n_\theta + 1$, and so this insertion was minimum. If $d_{P_{n_\theta}}(v_{l-t}, v_{l-1}) = \theta$, we have $d_{P_{n_\theta}}(v_{l-t}, v_l) = \theta + 1$, Procedure *SHIFT* was called and, by Lemma 1, we conclude the proof.

If v_{l-1} is adjacent to v_{l-t-1} in P^θ (see Fig. 4), the position of v_{l-t-1} in P^θ is between $l-t+1$ and $\xi_{P^\theta}(v_{l-t-1})$,

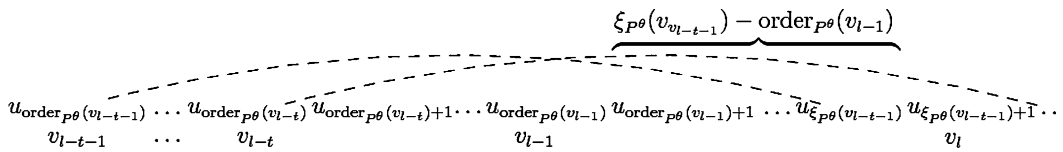

Fig. 4 Bracket indicates possible positions of vertices v_{l-1} and v_l

including them. Again, in order to minimize the number of vertices of P^θ, vertex v_l is inserted $(\xi_{P^\theta}(v_{l-t-1}) - \text{order}_{P^\theta}(v_{l-1}))$ vertices after vertex v_{l-1} in P^θ. Thus,

$$\begin{aligned}
&d_{P_{n_\theta}}(v_{l-t-1}, v_l) \\
&\quad = d_{P_{n_\theta}}(v_{l-t-1}, v_{l-1}) \\
&\qquad + \big(\xi_{P^\theta}(v_{l-t-1}) - \text{order}_{P^\theta}(v_{l-1})\big) + 1 \\
&\quad = \big(\text{order}_{P^\theta}(v_{l-1}) - \text{order}_{P^\theta}(v_{l-t-1})\big) \\
&\qquad + \big(\xi_{P^\theta}(v_{l-t-1}) - \text{order}_{P^\theta}(v_{l-1})\big) + 1 \\
&\quad = \big(\xi_{P^\theta}(v_{l-t-1}) - \text{order}_{P^\theta}(v_{l-t-1})\big) + 1 = \theta + 1.
\end{aligned}$$

Since $d_{P_{n_\theta}}(v_{l-t}, v_l) < d_{P_{n_\theta}}(v_{l-t-1}, v_l) = \theta + 1$, we have $d_{P_{n_\theta}}(v_{l-t}, v_l) \leq \theta$. So, v_l was inserted in P^θ without changing θ, and the number of vertices of P^θ became $n_\theta + (\xi_{P^\theta}(v_{l-t-1}) - \text{order}_{P^\theta}(v_{l-1})) + 1$. This insertion was minimum, because $d_{P_{n_\theta}}(v_{l-t-1}, v_l) = \theta + 1$.

This concludes the proof of the Lemma 2. □

In order to show that the Algorithm *CPP* returns the smallest power of a path containing G as an induced subgraph, we present two results with a given power of a path P^σ containing G as an induced subgraph. First, we shall give some notation from [3]. Given an unit interval graph G and an unit interval model associated to its vertices $I = \{v_1, v_2, \ldots, v_n\}$, we recall that the interval associated to vertex v is (a_v, b_v). We say that $v_1, v_2, \ldots, v_n$ is a *natural labeling* for the vertices of G, if $a_{v_i} \leq a_{v_{i+1}}$, for each $1 \leq i \leq n-1$. The ordering $v_1 < v_2 < \cdots < v_n$ is a *natural ordering*, if $v_1, v_2, \ldots, v_n$ is a natural labeling for $V(G)$. A vertex is a *left anchor* if it can receive the label v_1 in some natural labeling for $V(G)$. Consider the model I' obtained by mirroring an unit interval model I (that is, replacing each interval (a, b) by $(-b, -a)$). Model I' is also a valid unit interval model for G, so the rightmost interval in I is also a left anchor.

In the next results, we show properties of the ordering of $V(G)$ induced by a natural ordering that is generated by the subscripts of a natural labeling of a power of a path.

Lemma 3 *Let $P^\sigma_{n_\sigma}$ be a power of a path that contains G as an induced subgraph. The ordering of the vertices of $V(G)$ induced by a natural ordering of the vertices of $V(P^\sigma)$ satisfies Property 2.*

Proof Suppose that this ordering of $V(G)$ does not satisfy Property 2. Then, there exist three vertices $v_r, v_s, v_t \in V(G)$ such that $v_r < v_s < v_t$ with $v_r v_s \notin E(G)$ and $v_r v_t \in E(G)$. It follows that $v_r v_t \in E(G) \subset E(P^\sigma)$. Therefore, $1 \leq |\text{order}_{P^\sigma}(v_r) - \text{order}_{P^\sigma}(v_t)| \leq \sigma$. Since $v_r < v_s < v_t$ in $V(P^\sigma)$, we have $|\text{order}_{P^\sigma}(v_r) - \text{order}_{P^\sigma}(v_s)| \leq |\text{order}_P(v_r) - \text{order}_P(v_t)|$, and then $1 \leq |\text{order}_{P^\sigma}(v_r) - \text{order}_{P^\sigma}(v_s)| \leq \sigma$. Consequently, $v_r v_s \in E(P^\sigma)$ and $v_r v_s \notin E(G)$, i.e., $P^\sigma_{n_\sigma}$ does not contain G as an induced subgraph. □

Vertices $v, \overline{v} \in V(G)$ are *indistinguishable vertices* (*twin vertices*) if $N_G[v] = N_G[\overline{v}]$. The next result states that it is possible to change the position, between indistinguishable vertices of $V(G)$ in a natural ordering of $V(P^\sigma)$.

Lemma 4 *Let $v, \overline{v} \in V(G)$ such that $N_G[v] = N_G[\overline{v}]$ with $v = u_i$ and $\overline{v} = u_j$ in $V(P^\sigma)$. If we change the positions of vertices v and $\overline{v}$ in P^σ, i.e., $v = u_j$ and $\overline{v} = u_i$, graph G will still be an induced subgraph of P^σ.*

Proof Without loss of generality, suppose $i < j$. By Lemma 3, the ordering of $V(G)$ induced by a natural ordering of $V(P^\sigma)$ satisfies Property 2. So, $N_G[v] = \{v_{\eta_G(v)}, \ldots, v_{\xi_G(v)}\}$ and $N_G[\overline{v}] = \{v_{\eta_G(\overline{v})}, \ldots, v_{\xi_G(\overline{v})}\}$. Since $N[v] = N[\overline{v}]$, we have $\xi_G(\overline{v}) = \xi_G(v)$ and $\eta_G(\overline{v}) = \eta_G(v)$. Then, $v_{\xi_G(v)} = v_{\xi_G(\overline{v})}$, $v_{\eta_G(v)} = v_{\eta_G(\overline{v})}$, $v_{\xi_G(v)+1} = v_{\xi_G(\overline{v})+1}$ and $v_{\eta_G(v)-1} = v_{\eta_G(\overline{v})-1}$. Thus, by changing the positions of vertices v and $\overline{v}$ in P^σ, we have $\text{order}_{P^\sigma}(\overline{v}) - \text{order}_{P^\sigma}(v_{\eta_G(\overline{v})-1}) \geq \sigma + 1$; then edge $\overline{v} v_{\eta_G(\overline{v})-1} \notin E(P^\sigma)$. Also, for any $v' \in V(G)$ with $\text{order}_G(v') < \text{order}_G(v_{\eta_G(\overline{v})-1})$, edge $\overline{v} v' \notin E(P^\sigma)$. Similarly, $\text{order}_{P^\sigma}(v_{\xi_G(v)+1}) - \text{order}_{P^\sigma}(v) \geq \sigma + 1$, i.e., edge $v v_{\xi_G(v)+1} \notin E(P^\sigma)$ and also, for any $v' \in V(G)$ with $\text{order}_G(v_{\xi_G(v)+1}) < \text{order}_G(v')$, edge $v v' \notin E(P^\sigma)$.

Analogously, $\sigma \geq \text{order}_{P^\sigma}(\overline{v}) - \text{order}_{P^\sigma}(v_{\eta_G(v)})$, i.e., edge $\overline{v} v_{\eta_G(\overline{v})} \in E(P^\sigma)$ and, for any $v' \in V(G)$ with $\text{order}_G(v_{\eta_G(\overline{v})}) < \text{order}_G(v') < \text{order}_G(\overline{v})$, edge $\overline{v} v' \in E(P^\sigma)$. Similarly $\sigma \geq \text{order}_{P^\sigma}(v_{\xi_G(\overline{v})}) - \text{order}_{P^\sigma}(v)$, i.e., edge $v v_{\xi_G(v)} \in E(P^\sigma)$ and, for any $v' \in V(G)$ with $\text{order}_G(v) < \text{order}_G(v') < \text{order}_G(v_{\xi_G(v)})$, edge $v v' \in E(P^\sigma)$. □

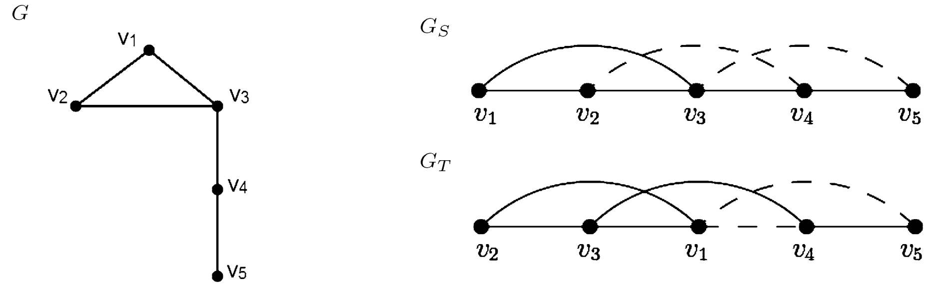

Fig. 5 Graph G is not induced subgraph of G_S and G_T

In what follows, we denote by $v_i <_B v_j$ if $\text{order}_G(v_i) < \text{order}_G(v_j)$ considering the ordering of $V(G)$ given by Algorithm *Recognize* [3]. First, Theorem 4, we need two results.

Theorem 3 (Theorem 2.2 [3]) *Let I be an unit interval model of an unit interval graph G with natural labeling $v_1, \dots, v_n$. Then, for all vertices $\bar{v}$, $v \in V(G)$, if $a_{\bar{v}} < a_v$ but $v <_B \bar{v}$, we have $N_G[v] = N_G[\bar{v}]$.*

As consequence of Theorem 2.3 of [3], we have the following result.

Lemma 5 ([3]) *Let $v'_1 <_B v'_2 <_B \cdots <_B v'_n$ be an ordering of $V(G)$ given by Algorithm* Recognize [3] *of an unit interval graph G. Given a natural labeling $v_1, \dots, v_n$ then $N_G[v'_1] = N_G[v_1]$ or $N_G[v'_1] = N_G[v_n]$.*

Finally, the correctness of Algorithm *CPP* is given by theorem below.

Theorem 4 *Let G be an unit interval graph. Algorithm* CPP *returns the smallest power of a path $P^{\theta}_{n_\theta}$ with respect to θ and n_θ, that contains G as an induced subgraph.*

Proof Let $P^{\sigma}_{n_\sigma}$ be the smallest power of a path that contains G as an induced subgraph. Let $\overline{u}_1 < \cdots < \overline{u}_{n_\sigma}$ be a natural ordering of $V(P^\sigma)$ and let $\overline{v}_1 < \cdots < \overline{v}_n$ be the ordering of $V(G)$ induced by the natural ordering of $V(P^\sigma)$. Clearly, $\overline{v}_1, \dots, \overline{v}_n$ is a natural labeling of $V(G)$. Let I be a family of intervals for this labeling of $V(G)$, such that each $v \in V(G)$ is associated to $(a_v, b_v) \in I$.

If we prove $\overline{v}_1 < \overline{v}_2 < \cdots < \overline{v}_n$ is equal to $v'_1 <_B v'_2 <_B \cdots <_B v'_n$ up to indistinguishable vertices, we have $\theta = \sigma$ and $n_\theta = n_\sigma$. In fact, since P^σ is the smallest power of a path that contains G as an induced subgraph, then $\sigma \le \theta$ and $n_\sigma \le n_\theta$. On the order hand, by Lemma 2, the power of a path P^θ returned by Algorithm *CPP* is the smallest power of a path that contains G as an induced subgraph with respect to the ordering, $v'_1 <_B v'_2 <_B \cdots <_B v'_n$. So, if this ordering is equal to $\overline{v}_1 < \overline{v}_2 < \cdots < \overline{v}_n$, up to indistinguishable vertices, by Lemma 4, P^σ contains G as an induced subgraph with respect to the ordering $v'_1 <_B v'_2 <_B \cdots <_B v'_n$. Then, by minimality of θ and n_θ with respect to $v'_1 <_B v'_2 <_B \cdots <_B v'_n$, we have $\sigma \ge \theta$ and $n_\sigma \ge n_\theta$.

First, suppose that the left anchor $\overline{v}_1$ is equal to v'_1. Suppose, by absurd, that there exist v, $\tilde{v} \in V(G)$, such that $v < \tilde{v}$, $\tilde{v} <_B v$ and $N_G[v] \ne N_G[\tilde{v}]$. Since $v < \tilde{v}$ then $a_v \le a_{\tilde{v}}$. If $a_v = a_{\tilde{v}}$, since all intervals of I have the same length, we have $b_v = b_{\tilde{v}}$ and hence $N_G[v] = N_G[\tilde{v}]$ a contradiction to the hypothesis. If $a_v < a_{\tilde{v}}$, since $\tilde{v} <_B v$ then, by Theorem 3, $N_G[v] = N_G[\tilde{v}]$, a contradiction to the hypothesis. Thus, for all pair of vertices $v, \tilde{v} \in V(G)$ such that $v < \tilde{v}$ and $\tilde{v} <_B v$, then $N_G[v] = N_G[\tilde{v}]$. Consequently, we have $\sigma = \theta$ and $n_\sigma = n_\theta$.

Now, suppose that the left anchor $\overline{v}_1$ is different from v'_1. By Lemma 5, either $N_G[\overline{v}_1] = N_G[v'_1]$ or $N_G[\overline{v}_n] = N_G[v'_1]$. If $N_G[\overline{v}_1] = N_G[v'_1]$, by Lemma 4, we can change the positions of these vertices in $V(P^\sigma)$, i.e., $\overline{u}_{\text{order}_{P^\sigma}(v'_1)} = \overline{v}_1$ and $\overline{u}_{\text{order}_{P^\sigma}(\overline{v}_1)} = v'_1$ and G will still be an induced subgraph of P^σ. After this change $v'_1 < \overline{v}_2 < \cdots < \overline{v}_1 < \cdots < \overline{v}_n$ is the new ordering of $V(G)$ induced by the ordering of $V(P^\sigma)$. We repeat the same argument used in the previous case, where $\overline{v}_1$ is equal to v'_1 and we conclude the proof. If $N_G[\overline{v}_n] = N_G[v'_1]$, since $\overline{v}_n$ is the left anchor of the natural labeling $\overline{v}_n < \overline{v}_{n-1} < \cdots < \overline{v}_1$ of $V(G)$ induced by the natural ordering $\overline{u}_{n_\sigma} < \cdots < \overline{u}_1$ of $V(P^\sigma)$ then, we can repeat the previous argument for the natural labeling $\overline{v}_n < \overline{v}_{n-1} < \cdots < \overline{v}_1$ and so we conclude the proof. □

The Algorithm *CPP* analyzes each vertex of G in the ordering returned by Algorithm *Recognize* [3] a single time. In the worst case, the Algorithm *CPP* calls Procedure *SHIFT* for each vertex $v_l \in V(G)$ only once. Since for each vertex v_l the Procedure *SHIFT* analyzes the set of vertices of G_l at most once, the complexity of the Algorithm *CPP* is $\mathcal{O}(n^2)$.

4 G is not an induced subgraph of G_S and G_T

If we relax the constraint that G must be an induced subgraph of G_S or G_T then even for unit interval graphs it is possible to find two powers of paths, whose intersection

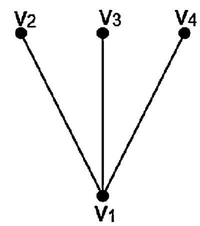

Fig. 6 Claw

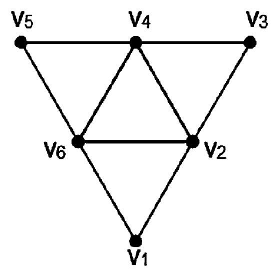

Fig. 7 3-sun (S_3)

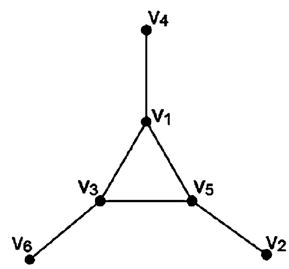

Fig. 8 Net ($\overline{S}_3$)

contains G as an induced subgraph, smaller than the answer given by Algorithm *CPP*. See an example in Fig. 5.

If graph G is an unit interval graph then G contains no induced Claw (Fig. 6), S_3 (Fig. 7), $\overline{S}_3$ (Fig. 8) and Cycle (C_n), $n \geq 4$. If G is a cycle C_n, $n \geq 4$. Then the smallest θ-powers of paths, G_S and G_T, such that $G_S \cap G_T$ contains C_n as induced subgraph can be obtained as follows. First, we construct G_S: for $1 \leq j \leq \lceil \frac{n}{2} \rceil$, $u_{2j-1} := v_j$; and $1 \leq j \leq \lfloor \frac{n}{2} \rfloor$, $u_{2j} := v_{n+1-j}$. Now, we construct G_T: for $1 \leq j \leq \lceil \frac{n}{2} \rceil$, $w_{2j-1} := v_{j+1}$; and $1 \leq j \leq \lfloor \frac{n}{2} \rfloor$, $w_{2j} := v_k$, where $k = (n + 2 - j) \bmod n$. See an example when G is a C_6 in Fig. 9.

Theorem 5 *Let G_S and G_T be 2-powers of paths with n vertices constructed by the previous method. Then $G_S \cap G_T$ is C_n, $n \geq 4$.*

Proof Let G_S be the 2-power of path $P_S = u_1, \ldots, u_n$, and let G_T be the 2-power of path $P_T = w_1, \ldots, w_n$ constructed by the previous method. Since the distance between consecutive vertices of G in G_S (resp. G_T) is less than or equal to 2, G_S (resp. G_T) contains G as subgraph.

For each $v_i \in C_n$, $i \in \{2, \ldots, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil + 2, \ldots, n\}$, and $3 \leq j \leq n-2$, if $u_j = v_i$ with j odd then $w_{j-2} = v_i$; if j is even, we have $w_{j+2} = v_i$.

Now, let $v_i \in C_n$, if $u_j = v_i$, $3 \leq j \leq n-2$ with j odd (resp. even), then $w_{j-2} = v_i$ (resp. $w_{j+2} = v_i$), and its neighbors $u_{j-1} = v_k = w_{j-1+2}$ (resp. $u_{j-1} = v_{k-1} = w_{j-1-2}$) and $u_{j+1} = v_{k-1} = w_{j+1+2}$ (resp. $u_{j+1} = v_k = w_{j+1-2}$). We conclude that $d_{P_S}(v_i, v_k) = d_{P_S}(v_i, v_{k-1}) = 1$, $d_{P_T}(v_i, v_k) = 3$ and $d_{P_T}(v_i, v_{k-1}) = 5$, i.e., $v_i v_k, v_i v_{k-1} \in E(G_S)$ and $v_i v_k, v_i v_{k-1} \notin E(G_T)$. Hence, $v_i v_k, v_i v_{k-1} \notin G_S \cap G_T$. □

5 Conclusion

In this work, we developed an $\mathcal{O}(n^2)$ time algorithm that generates, from a connected unit interval graph G, an explicit representation of the smallest θ-power of path G_S (with respect to θ and to the number of vertices) that contains G as an induced subgraph. We construct G_T, a θ-power of a path with the same number of vertices of G_S, such that the intersection $G_S \cap G_T$ contains G as an induced subgraph.

We remark that θ can be greater than or equal to the size of a maximum clique of the graph G, $\omega(G)$. We present in Fig. 10 an example where G has $\omega(G) = 4$ and Algorithm *CPP* returns $\theta = 5$, but the difference between θ and $\omega(G)$ can be greater than 1.

In case graph G is not an induced of G_S and G_T, we show a method that generates G_S and G_T, 2-powers of paths with n vertices, whose intersection is C_n, $n \geq 4$.

As future work, we intend to investigate this problem for other classes of graphs. We remark that all remaining forbidden induced subgraphs of unit interval graphs (Figs. 6, 7 and 8), have answer YES to Question 1.

For a Claw graph, we see that G_S is the 2-power of path $P_S = v_2, a, v_1, v_3, v_4$; and G_T is the 2-power of path $P_T = v_3, b, v_1, v_2, v_4$. For a 3-sun graph, we find that G_S and

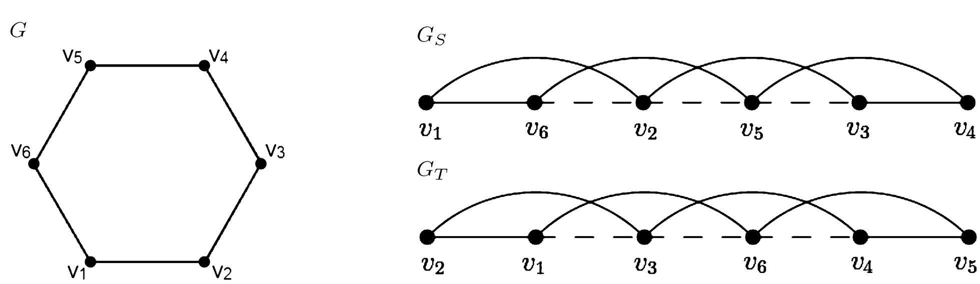

Fig. 9 Graph $G = C_6$ and the respective G_S and G_T, 2-powers of paths with 6 vertices

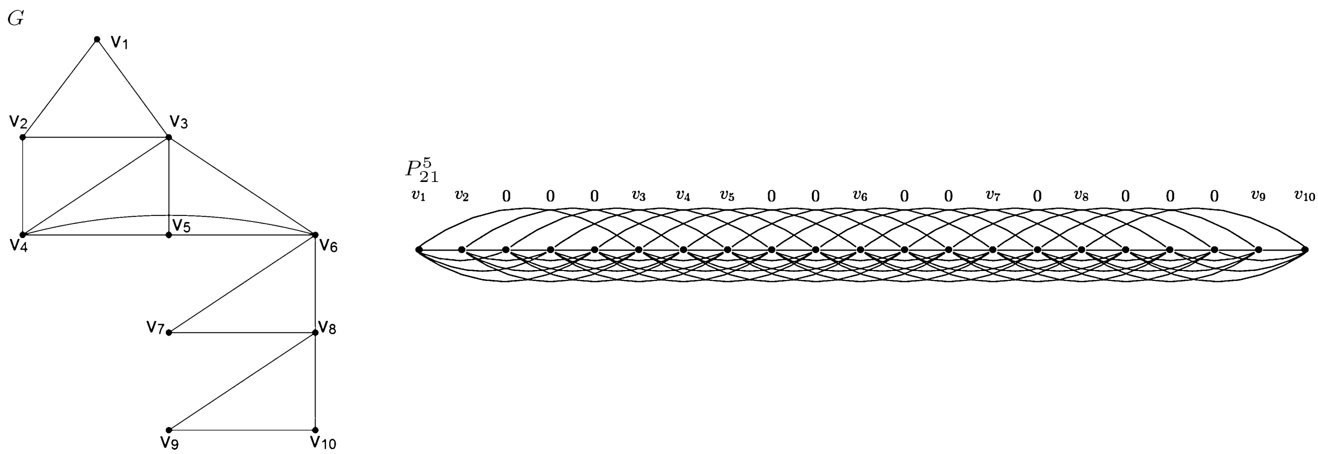

Fig. 10 Graph G with $n = 10$ and $\omega(G) = 4$ and the output returned by Algorithm *CPP*: $P^{\theta}_{n_\theta}$ with $n_\theta = 21$ and $\theta = 5$

G_T are 4-powers of paths $P_S = v_5, a, b, v_4, v_6, v_3, v_1, v_2$ and $P_T = v_1, v_6, x, v_5, v_2, v_4, y, v_3$, respectively. For a Net graph, we see that G_S and G_T are 2-powers of paths $P_S = v_4, v_2, v_1, v_5, v_3, a, v_6$ and $P_T = v_4, b, v_1, v_5, v_3, v_2, v_6$, respectively.

Acknowledgements This research was supported by CNPq and FAPERJ.

We are really grateful to professor Jayme Szwarcfiter for having presented to us the paper [5] in the very beginning of our work and for fruitful discussions on this topic. We are also thankful to the anonymous referees for their careful reading and valuable contributions.

References

1. Adam Z, Choi V, Sankoff D, Zhu Q (2008) Generalized gene adjacencies, graph bandwidth and clusters in yeast evolution. In: Lecture Notes in Bioinformatics, vol 4983, pp 134–145
2. Brandstädt A, Hundt C, Mancini F, Wagner P (2010) Rooted directed path graphs are leaf powers. Discrete Math 310:897–910
3. Corneil DG, Kim H, Natarajan S, Olariu S, Sprague A (1995) Simple linear time recognition of unit interval graphs. Inf Process Lett 55:99–104
4. Figueiredo CMH, Meidanis J, Mello CP (1995) A linear-time algorithm for proper interval graph recognition. Inf Process Lett 56:179–184
5. Lin MC, Rautenbach D, Soulignac FJ, Szwarcfiter JL (2011) Powers of cycles, powers of paths, and distance graph. Discrete Appl Math 159:621–627
6. Lin MC, Soulignac FJ, Szwarcfiter JL (2009) Short models for unit interval graphs. Electron Notes Discrete Math 35:247–255
7. Roberts FS (1968) Representations of indifference relations. Stanford University, Stanford
8. Sankoff D, Xu X (2008) Tests for gene clusters satisfying the generalized criterion. Lect Notes Comput Sci 5167:152–160
9. Soulignac FJ (2010) On proper and helly circular-arc graphs. Universidad de Buenos Aires, Buenos Aires