IPSJ SIG Technical Report

Vol.2011-AL-134 No.1
2011/3/7

Approximating the path-distance-width for k-cocomparability graphs

Yota Otachi[*1], Toshiki Saitoh[*2], Katsuhisa Yamanaka[*3], Shuji Kijima[*4], Yoshio Okamoto[*5], Hirotaka Ono[*6], Yushi Uno[*7] and Koichi Yamazaki[*8]

1. Introduction

The path-distance-width is a graph parameter to measure how close a graph is to a path[19),20)]. There are several other such graph parameters including pathwidth and bandwidth. Roughly speaking, the classes of graphs of bounded path-distance-width, bounded bandwidth, and bounded pathwidth have *chain-like structures*. It is known that for any connected graph, pathwidth $\leq$ bandwidth $< 2 \cdot$ path-distance-width[13),20)]. By this relation, many useful properties for bounded pathwidth graphs and bounded bandwidth graphs also hold for bounded path-distance-width graphs. There are other graph classes which also have chain-like structures, such as interval graphs, AT-free graphs, and k-cocomparability graphs. It is known that there are relationships among those graph parameters and graph classes[2),4),13),20)].

This study is motivated by the research on bandwidth of AT-free graphs[10),14)]. To see the motivation, let us briefly review the history of the research on bandwidth for interval graphs and AT-free graphs. One may expect that if we restrict our input graphs to interval graphs or AT-free graphs, then we would be able to find easily its chain-like structure (such as its interval representation or a dominating pair), and then from the chain-like structure we might be able to compute the bandwidth. It had not been known, however, whether the bandwidth can be computed for interval graphs in polynomial time[12)]. But then it turned out that the decision problem can be solved in polynomial time (see[18)]). Since interval graphs are AT-free graphs, it would be natural to ask whether or not the bandwidth decision problem for AT-free graphs can be solved in polynomial time. Unfortunately, the bandwidth decision problem for AT-free graphs is NP-complete[14),16)]. However, it is known that for AT-free graphs, the bandwidth can be approximated within a factor 2 in $O(mn)$ time[14)], where m and n denote the number of edges and the number of vertices, respectively.

In a sense, bandwidth and path-distance-width have some common features. In fact, there is a similarity between the problem of computing the path-distance-width and the problem of computing the bandwidth: both problems do not admit any PTAS even for trees[1),19)]. Hence, it would be reasonable to ask the computational complexity of computing the path-distance-width for AT-free graphs. Unfortunately, as we will prove in this paper, the path-distance-width decision problem for AT-free graphs is also NP-complete. More precisely, we will show that the problem is NP-complete for cobipartite graphs. Thus we consider the problem of approximating the path-distance-width.

Although some techniques developed in the research on bandwidth can be carried over into the research on path-distance-width, the path-distance-width problem has a serious drawback which the bandwidth problem does not have: path-distance-width is *not* closed under the edge deletion. In many cases, this drawback makes the design and analysis of algorithms very difficult. In this study, however, it turns out that the restriction to AT-free graphs is enough to overcome the drawback for achieving a constant factor approximation. In this paper, we first present approximation algorithms with constant approximation ratios for the path-distance-width on a superclass of AT-free graphs, which is known as k-cocomparability graphs. Although this is already a constant factor approximation for AT-free graphs, we present another approximation algorithm for

*1 Graduate School of Information Sciences, Tohoku University. Sendai 980-8579, Japan. JSPS Research Fellow. otachi@dais.is.tohoku.ac.jp
*2 ERATO MINATO Discrete Structure Manipulation System Project, Japan Science and Technology Agency. North 14, West 9, Sapporo, Hokkaido, 060-0814, Japan. t-saitoh@erato.ist.hokudai.ac.jp
*3 Graduate School of Information Systems, University of Electro-Communications. Chofugaoka 1-5-1, Chofu, Tokyo, 182-8585 Japan. yamanaka@is.uec.ac.jp
*4 Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, 819-0395, Japan. kijima@inf.kyushu-u.ac.jp
*5 Center for Graduate Education Initiative, JAIST, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan. okamotoy@jaist.ac.jp
*6 Department of Economic Engineering, Kyushu University. 6-19-1 Hakozaki Higashi-ku, Fukuoka 812-8581, Japan. hirotaka@en.kyushu-u.ac.jp
*7 Department of Mathematics and Information Sciences, Graduate School of Science, Osaka Prefecture University. 1-1 Gakuen-cho, Naka-ku, Sakai 599-8531, Japan. uno@mi.s.osakafu-u.ac.jp
*8 Department of Computer Science, Gunma University. 1-5-1 Tenjin-cho, Kiryu, Gunma, 376-8515 Japan. koichi@cs.gunma-u.ac.jp

AT-free graphs, which has a better running time and a better approximation ratio. We also show that the problem is solvable for cochain graphs. The complexity for interval graphs and proper interval graphs remains open.

2. Preliminaries

In this paper, graphs are finite, simple, and connected.

Graph

Let G be a graph. We denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. The *distance between two vertices* $u, v \in V(G)$ *in* G, denoted by $d_G(u, v)$, is the length of a shortest u–v path in G. We define the *distance between a vertex subset* $S \subseteq V(G)$ *and a vertex* $v \in V(G)$ in G as $d_G(S, v) = \min_{u \in S} d_G(u, v)$. For $S \subseteq V(G)$, we define the *diameter of* S *in* G as $\mathrm{diam}_G(S) = \max_{u,v \in S} d_G(u, v)$. The *diameter of a graph* G is defined as $\mathrm{diam}(G) = \mathrm{diam}_G(V(G))$.

The *(open) neighborhood* of a vertex v in G, denoted by $N_G(v)$, is the set of vertices adjacent to v; that is $N_G(v) = \{u \mid \{u, v\} \in E(G)\}$. The *closed neighborhood* of v in G, denoted by $N_G(v)$, is the set $\{v\} \cup N_G(v)$. The *(open) neighborhood* of a vertex set $S \subseteq V(G)$ in G, denoted by $N_G(S)$, is the set of vertices not in S and adjacent to some vertex $u \in S$; that is $N_G(S) = \bigcup_{v \in S} N_G(v) \setminus S$.

The *compliment of a graph* G is the graph $\overline{G}$ such that $V\left(\overline{G}\right) = V(G)$ and two distinct vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G.

Path-distance-width

A sequence $(L_1, L_2, \ldots, L_t)$ of subsets of vertices is a *distance structure* of a graph G if $\bigcup_{1 \le i \le t} L_i = V(G)$ and $L_i = \{v \in V(G) \mid d_G(v, L_1) = i - 1\}$ for each $1 \le i \le t$. Each L_i is called a *level* and specially L_1 is called the *initial set*. The width of $(L_1, L_2, \ldots, L_t)$, denoted by $\mathrm{pdw}_{L_1}(G)$, is defined as $\max_{1 \le i \le t} |L_i|$. The *path-distance-width* of G, denoted by $\mathrm{pdw}(G)$, is defined as $\min_{S \subseteq V(G)} \mathrm{pdw}_S(G)$.

If the initial set of a distance structure of G is a set consists of only one vertex, then we say that it is a *rooted distance structure* of G. The *rooted path-distance-width* of G, denoted by $\mathrm{rpdw}(G)$, is the minimum width over all its rooted distance structures; that is, $\mathrm{rpdw}(G) = \min_{v \in V(G)} \mathrm{pdw}_{\{v\}}(G)$. Obviously, the rooted path-distance-width can be computed in polynomial time (see Lemma 2.1 for more details).

All-pairs shortest paths

The *all-pairs shortest paths problem* is literally the problem of finding a shortest path between each pair of vertices in a graph with n vertices and m edges. In some cases, all-pairs distances are needed instead of actual shortest paths. We consider this variant here; that is, we want to compute $d_G(u, v)$ for all pairs $u, v \in V(G)$. Clearly, by running breadth-first search from every vertex, the problem can be solved in $O(mn)$ time. The problem has been studied extensively, and there are some nontrivial improvements (see[3] and the references therein). Seidel[17] proved that the problem can be solved in $O(M(n) \log n)$ time by using fast matrix multiplication, where $M(n)$ is the time complexity to multiply two $n \times n$ matrices. The current fastest algorithm for matrix multiplication by Coppersmith and Winograd[5] implies that Seidel's algorithm runs in $O(n^{2.376})$ time. Recently, Chan[3] has presented a new algorithm for the all-pairs shortest path problem that runs in $o(mn)$ time.

For a graph G with n vertices and m edges, let $\mathrm{apd}(m, n)$ be the time complexity for computing the all-pairs distances and outputting the distance for each vertex pair. We can use any one of the above algorithms for the all-pairs distances. Note that $\mathrm{apd}(m, n) = \Omega(n^2)$ since we must output the distances for all $\binom{n}{2}$ pairs.

Lemma 2.1. *The rooted path-distance-width of a connected graph G with n vertices and m edges can be computed in $O(\mathrm{apd}(m, n))$ time.*

Proof. First, we compute $d_G(u, v)$ for all pairs $u, v \in V(G)$ in $O(\mathrm{apd}(m, n))$ time. By using the distance matrix d_G, we can compute $\mathrm{rpdw}(G)$ in $O(n^2)$ time. Since $\mathrm{apd}(m, n) = \Omega(n^2)$, the total running time is $O(\mathrm{apd}(m, n))$. □

Graph Classes

An *interval graph* is a graph whose vertices can be mapped to distinct intervals in the real line such that two vertices are adjacent in the graph if and only if their corresponding intervals overlap. We call the set of intervals representing a graph an *interval representation* of the graph. An interval representation is *proper* if no interval properly contains other intervals in it. A graph is a *proper interval graph* if it has a proper interval representation.

An independent set of three vertices is called an *asteroidal triple* if every two of them are connected by a path avoiding the neighborhood of the third. A graph is *asteroidal triple-free* (*AT-free* for short), if it contains no asteroidal triple.

A graph G is a *comparability graph* if there exists a linear ordering $<$ on $V(G)$ such that for any three vertices $u < v < w$, $\{u, v\} \in E(G)$ and $\{v, w\} \in E(G)$ implies $\{u, w\} \in E(G)$. A graph G is a *cocomparability graph* if G is the compliment of a comparability graph. It is known that G is a cocomparability graph if and only if it has a *cocomparability ordering*; that is, there exists a linear order $<$ on $V(G)$ such that for any three vertices $u < v < w$, $\{u, w\} \in E(G)$ implies $\{u, v\} \in E(G)$ or $\{v, w\} \in E(G)$.

Chang, Ho, and Ko[4] generalized cocomparability graphs to k-cocomparability graphs. Let G be a graph, and let k be a positive integer. A *k-cocomparability ordering* (*k-CCPO*) of G is an ordering on $V(G)$ such that for any three vertices $u < v < w$, $d_G(u, w) \leq k$ implies $d_G(u, v) \leq k$ or $d_G(v, w) \leq k$. A graph is a *k-cocomparability graph* if it admits a k-CCPO. Note that a 1-cocomparability ordering is just a cocomparability ordering.

A graph $G = (U, V; E)$ is a *cobipartite graph* if (U, V) is a nonempty partition of $V(G)$ and both U and V induce cliques. Thus a cobipartite graph is the complement of a bipartite graph. This implies that cobipartite graphs are cocomparability graphs, since bipartite graphs are comparability graphs. A cobipartite graph $H = (X, Y; E)$ is a *cochain graph* if the elements of X and Y can be ordered as $x_1, x_2, \ldots, x_{|X|}$ and $y_1, y_2, \ldots, y_{|Y|}$, respectively, so that $N_G[x_1] \subseteq N_G[x_2] \subseteq \cdots \subseteq N_G[x_{|X|}]$ and $N_G[y_1] \subseteq N_G[y_2] \subseteq \cdots \subseteq N_G[y_{|Y|}]$.

It is known that cochain graphs $\subset$ proper interval graphs $\subset$ interval graphs $\subset$ cocomparability graphs $\subset$ AT-free graphs $\subset$ 2-cocomparability graphs, and k-cocomparability graphs $\subset$ $(k+1)$-cocomparability graphs for any k (see[2),4]). It is easy to see that any graph G is a k_G-cocomparability graph for some large enough $k_G \leq \mathrm{diam}(G)$.

Summary of results

In this paper, we present some algorithms approximating the path-distance-width for k-cocomparability graphs and their subclasses such as AT-free graphs and proper interval graphs. Every algorithm has a constant approximation ratio (if k is a fixed constant), and runs in $O(\mathrm{apd}(m, n))$ or $O(m + n)$ time. See **Fig. 1**.

3. NP-hardness for cobipartite graphs

Before we present approximation algorithms, we show that the problem for determining the path-distance width is NP-hard even for a very restricted graph class, the class of cobipartite graphs. To this end, we first prove the NP-completeness of an intermediate problem, by constructing a polynomial time reduction from the following well-known NP-complete problem.

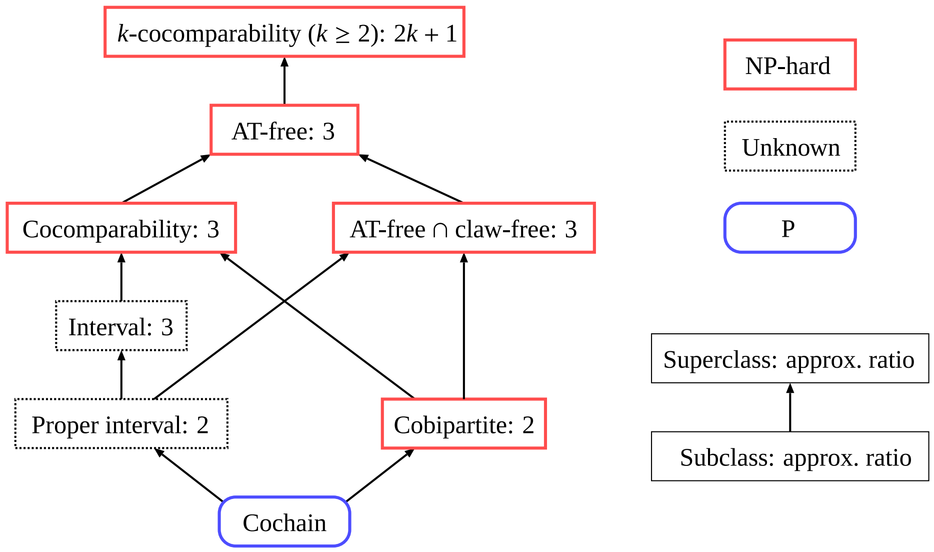

Fig. 1 Summary of results.

Problem: Set Cover[9] [SP5]
Instance: Set $C = \{c_1, \ldots, c_n\}$, family $\mathcal{F} = \{F_1, \ldots, F_m\} \subseteq 2^C$, positive integer $h \leq n$.
Question: Is there $X \subseteq \mathcal{F}$ such that $\bigcup_{F_i \in X} F_i = C$ and $|X| = h$?

In any instance of Set Cover, we can assume without loss of generality that for every $c_i \in C$, there is a subset $F_j \in \mathcal{F}$ such that $c_i \in F_j$, since otherwise the instance has no cover. We also assume $n > 1$ and $h < m$, since otherwise the problem is trivial.

Our intermediate problem is as follows.

Problem: Partial Cover in Bigraphs (PCB)
Instance: Bipartite graph $G = (U, V; E)$, positive integer $k \leq |V|$.
Question: Is there $Y \subseteq U$ such that $|N_G(Y)| = k$?

Kobayashi[15] pointed out that PCB is NP-complete. Here, we provide a full proof.

Lemma 3.1. *PCB is NP-complete even if $|V| > k + 2$ and G has no isolated vertex.*

Proof. From an instance $(C, \mathcal{F}, h)$ of Set Cover, we first construct a bipartite graph $G = (U, V; E)$ as follows: $U = \{u_1, \ldots, u_m\}$, $V = \{v_1, \ldots, v_n\}$, and $E = \{\{u_i, v_j\} \mid c_j \in F_i\}$.

The vertex sets U and V corresponds to the family $\mathcal{F}$ and the ground set C, respectively. The edge set E represents the containment relation between the elements of C and the subsets in $\mathcal{F}$. Next, by adding $n+1$ pendant vertices to each $u_i \in U$, we construct a bipartite graph $H = (U, V'; E')$. Clearly, this construction can be done in polynomial time. Note that $|V'| = n + (n+1)m > n + (n+1)h + 2$ since $n > 1$ and $m > h$. Also note that H has no isolated vertex.

Let $k = n + (n+1)h$. We shall prove that C has a cover $X \subseteq \mathcal{F}$ of size $|X| = h$ if and only if there is a set $Y \subseteq U$ such that $|N_H(Y)| = k$.

($\Longrightarrow$) Assume that there is $X \subseteq \mathcal{F}$ such that $\bigcup_{F_i \in X} F_i = C$ and $|X| = h$. We set $Y = \{u_i \mid F_i \in X\}$. Since X is a cover of C, $|N_H(Y) \cap V| = |V| = n$. Since $|N_H(Y) \setminus V| = (n+1)h$,

$$|N_H(Y)| = |N_H(Y) \cap V| + |N_H(Y) \setminus V| = n + (n+1)h = k.$$

($\Longleftarrow$) Assume that there exists $Y \subseteq U$ such that $|N_H(Y)| = k$. We first prove $|Y| = h$. If $|Y| \geq h+1$, then $|N_H(Y)| \geq |N_H(Y) \setminus V| \geq (n+1)(h+1) > k$. If $|Y| \leq h-1$, then $|N_H(Y)| \leq |V| + |N_H(Y)| \setminus V| \leq n + (n+1)(h-1) < k$. Thus $|Y| = h$. Now we have

$$|N_H(Y) \cap V| = |N_H(Y)| - |N_H(Y) \setminus V| = k - (n+1)h = n.$$

Therefore, if we set $X = \{F_i \mid u_i \in Y\}$, then $|X| = h$ and X covers the ground set C.

From the above observation the problem is NP-hard. Since the problem clearly belongs to NP, the lemma holds. □

Now we prove the NP-hardness of the path-distance-width problem for cobipartite graphs, by constructing a polynomial time reduction from PCB. We actually prove that deciding whether $\mathrm{pdw}(G) = |V(G)|/3$ is NP-complete for cobipartite graphs with diameter 2.

Theorem 3.2. *Given cobipartite graph H with* $\mathrm{diam}(H) = 2$, *it is NP-complete to decide whether* $\mathrm{pdw}(H) = |V(H)|/3$.

Proof. Clearly, the problem is in NP. Thus we prove the NP-hardness. From an instance $(G = (U, V; E), k)$ of PCB satisfying the conditions in Lemma 3.1, we construct a cobipartite graph $H = (U', V'; E')$ as follows (see **Fig. 2**). Let S and T be two sets of sizes $|S| = |U| + k$ and $|T| = |U| + 2|V| - k - 2$, where S, T, U, and V are pairwise non-intersecting. We set the vertex sets as $U' = U \cup T \cup \{a\}$ and $V' = V \cup S \cup \{b\}$, where a and b are new vertices. In H, both U' and V' induce cliques. Every edge in G is also in H. Additionally, a is adjacent to all vertices in S, and b is adjacent to all vertices in T. This construction can be done in polynomial time.

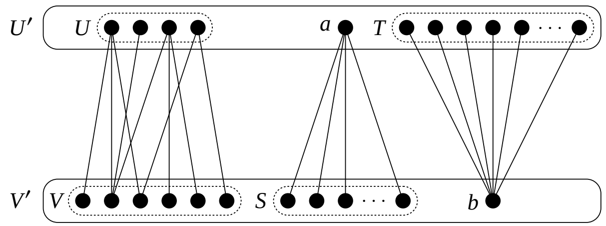

Fig. 2 Cobipartite graph $H = (U', V'; E')$.

Since G has no isolated vertex, $\mathrm{diam}(H) = 2$. It is easy to see that $|U'| = 2|U| + 2|V| - k - 1$ and $|V'| = |V| + |U| + k + 1$. Hence, $|V(H)| = |U'| + |V'| = 3(|U| + |V|)$. We shall show that (G, k) is a yes instance of PCB if and only if $\mathrm{pdw}(H) = |U| + |V|$. Note that $\mathrm{pdw}(H) \geq |V(H)|/(\mathrm{diam}(H) + 1) = |U| + |V|$.

($\Longrightarrow$) Assume that there exists $Y \subseteq U$ such that $|N_G(Y)| = k$. Let $X = Y \cup T'$, where T' is any subset of T such that $|T'| = |U| + |V| - |Y|$. Let $(L_1 = X, L_2, L_3)$ be the level structure with the initial set X. Clearly, $|L_1| = |X| = |U| + |V|$. The size of the second level is

$$|L_2| = |U' \setminus X| + |N_H(Y) \cap V'| + |N_H(T') \cap V'| = |U| + |V|. \tag{1}$$

This also implies $|L_3| = |V(H)| - |L_1| - |L_2| = |U| + |V|$. Therefore, $\mathrm{pdw}_X(H) = |U| + |V|$.

($\Longleftarrow$) Assume that $\mathrm{pdw}_X(H) = |U| + |V|$ for some $X \subseteq V(H)$. If X intersects both U' and V', then the distance structure has at most two levels, and thus $\mathrm{pdw}_X(H) \geq |V(H)|/2 > |U| + |V|$. Hence, X is included in either U' or V'. Suppose $X \subseteq V'$. Since $N_H(T) \cap V' = \{b\}$, all vertices in T belong to the same level. Since $|V| > k + 2$, this implies $\mathrm{pdw}_X(H) \geq |T| = |U| + 2|V| - k - 2 > |U| + |V|$, which is a contradiction. Thus we can conclude that $X \subseteq U'$.

Let $(L_1 = X, L_2, L_3)$ be the level structure with the initial set X. Since $|V(H)| = 3(|U| + |V|)$ and $\mathrm{pdw}_X(H) = |U| + |V|$, each level L_i has size $|L_i| = |U| + |V|$. If $a \in X$, then $S \subseteq L_2$. This implies $|L_3| \leq |V' \setminus S| = |V| + 1 < |U| + |V|$, a contradiction. Hence, $X \subseteq U \cup T$. Let $Y = X \cap U$ and $T' = X \cap T$. Clearly, $|N_H(T') \cap V'| = |\{b\}| = 1$. Since $|X| = |U| + |V|$, we have $|U' \setminus X| = |U| + |V| - k - 1$. Since Eq. (1) also holds here, we have $|N_H(Y) \cap V'| = k$. This implies $N_G(Y) = k$, and completes the proof. □

Here, we note that there is a trivial factor 2 approximation algorithm for cobipartite graphs. It is easy to see that a connected cobipartite graph G has diameter 3, and thus $\mathrm{pdw}(G) \geq \lceil |V(G)|/4 \rceil$. For any $S \subseteq V(G)$ with $|S| = \lceil |V(G)|/2 \rceil$, $\mathrm{pdw}_S(G) = \lceil |V(G)|/2 \rceil$. Therefore, $\mathrm{pdw}_S(G) \leq \lceil |V(G)|/2 \rceil \leq 2\lceil |V(G)|/4 \rceil \leq 2\mathrm{pdw}(G)$.

Proposition 3.3. *For a cobipartite graph with n vertices and m edges, the path-distance-width can be approximated within a factor* 2 *in* $O(m+n)$ *time.*

4. Approximating the path-distance-width

In this section, we present our main results. Namely, approximation algorithms for the path-distance-width. Our algorithms are based on a common idea: bounding the diameter of each level in distance structures. This yields the approximation guarantees. The algorithms also have a special feature: we use rooted distance structures only. Thus, our algorithms are very simple, and clearly run in polynomial time.

We first establish a general lower bound, which will be the main tool to guarantee the approximation ratios.

Proposition 4.1. *Let* $(L_1, \ldots, L_t)$ *be a distance structure of* G. *If* $u \in L_i$ *and* $v \in L_j$, *then* $d_G(u, v) \geq |i - j|$.

Proof. Assume $i \leq j$ without loss of generality. Let $(p_0, p_1, \ldots, p_\ell)$ be a shortest u–v path, where $p_0 = u$ and $p_\ell = v$. From the definition of distance structures, if $p_k \in L_h$, then $p_{k+1} \in L_{h-1} \cup L_h \cup L_{h+1}$. Since $p_0 \in L_i$, $p_\ell \in L_j$, and $i \leq j$, we need at least $j - i$ indices k such that $p_k \in L_h$ and $p_{k+1} \in L_{h+1}$. Thus $\ell \geq j - i$. □

Lemma 4.2. *Let* $S \subseteq V(G)$. *Then,* $\mathrm{pdw}(G) \geq |S|/(\mathrm{diam}_G(S) + 1)$.

Proof. Let $(L_1, \ldots, L_t)$ be an optimal distance structure of G; that is, $\mathrm{pdw}_{L_1}(G) = \mathrm{pdw}(G)$. Denote by I the set of the indices of levels having non-empty intersection with S; that is, $I = \{i \in \{1, \ldots, t\} \mid L_i \cap S \neq \emptyset\}$. By Proposition 4.1, $\max I - \min I \leq \mathrm{diam}_G(S)$. Thus, the vertices of S are included in at most $\mathrm{diam}_G(S) + 1$ levels $\{L_{\min I}, L_{\min I+1}, \ldots, L_{\max I}\}$. This implies that there exists a level L_i, $i \in I$, such that $|L_i \cap S| \geq |S|/(\mathrm{diam}_G(S) + 1)$. Hence, we have
$$\mathrm{pdw}(G) = \mathrm{pdw}_{L_1}(G) \geq |L_i| \geq |L_i \cap S| \geq |S|/(\mathrm{diam}_G(S) + 1),$$
as required. □

4.1 Approximating the path-distance-width for k-cocomparability graphs

By the property of k-CCPO, we are able to bound the diameter of each level in some distance structure of a k-cocomparability graph. Thus we have an approximation guarantee as follows.

Lemma 4.3. *Let* G *be a connected* k*-cocomparability graph, and* x *be the first vertex in a* k*-CCPO of* G. *Let* $(L_1, \ldots, L_t)$ *be the distance structure of* G *with the initial set* $L_1 = \{x\}$. *Then,* $\mathrm{diam}_G(L_i) \leq 2k$ *for all* i.

Proof. Let $y, z \in L_i$ for some i. Without loss of generality, we may assume that $x < y < z$ in the k-CCPO. We show that $d_G(y, z) \leq 2k$. Obviously, $d_G(x, y) = d_G(x, z)$. Let P be a shortest x–z path in G. Since $d_G(x, y) = d_G(x, z)$, y is not in P. Clearly, there exists an edge $\{v, w\}$ in P such that $v < y < w$. Since $d_G(v, w) = 1 \leq k$, we have $d_G(v, y) \leq k$ or $d_G(y, w) \leq k$. If $d_G(v, y) \leq k$, then $d_G(x, y) \leq d_G(x, v) + k$ and $d_G(y, z) \leq d_G(v, z) + k$. This implies
$$d_G(x, y) + d_G(y, z) \leq d_G(x, v) + d_G(v, z) + 2k = d_G(x, z) + 2k.$$
Then $d_G(y, z) \leq 2k$, since $d_G(x, y) = d_G(x, z)$. The case of $d_G(y, w) \leq k$ is almost the same. □

Combining Lemmas 2.1, 4.2, and 4.3, we have the following general approximation result.

Theorem 4.4. *For a connected* k*-cocomparability graph* G *with* n *vertices and* m *edges, the path-distance-width can be approximated within a factor* $2k+1$ *in* $O(\mathrm{apd}(m, n))$ *time.*

4.2 Approximating the path-distance-width for AT-free graphs

Chang, Ho, and Ko[4] showed that AT-free graphs are 2-cocomparability graphs. Hence, by Theorem 4.4, the path-distance-width of a connected AT-free graph with n vertices and m edges can be approximated within a factor 5 in $O(\mathrm{apd}(m, n))$ time. The aim of this subsection is to provide a better approximation algorithm for AT-free graphs by using some properties of AT-free graphs. More precisely, we present an $O(m+n)$ time 3-approximation algorithm for AT-free graphs. A *dominating pair* (u, v) of a graph G is a pair of vertices $u, v \in V(G)$ such that for any u–v path P in G, $V(P)$ is a dominating set of $V(G)$; that is, each vertex $v \in V(G) \setminus V(P)$ has a neighbor in $V(P)$.

Theorem 4.5 ([7,8])**.** *Any connected AT-free graph has a dominating pair. A dominating pair of a connected AT-free graph can be found in linear time.*

Lemma 4.6. *Let* (u, v) *be a dominating pair of an AT-free graph* G, *and let* $(L_1 = \{u\}, \ldots, L_t)$ *be the distance structure rooted at the vertex* u. *Then, for any* i, $\mathrm{diam}_G(L_i) \leq 2$.

Proof. Let $(p_1, \dots, p_\ell)$ be a shortest u–v path in G, where $p_1 = u$ and $p_\ell = v$. Clearly, $p_j \in L_j$ for all j. From the definition of distance structures and dominating pairs, a vertex in a level L_i must be adjacent to at least one of p_{i-1}, p_i, and p_{i+1}, and cannot be adjacent to any other p_j, $j \notin \{i-1, i, i+1\}$. Let $x, y \in L_i$ for some i. We assume $p_i \notin \{x, y\}$ since otherwise $d_G(x, y) \le 2$. Let $(q_1, \dots, q_i)$ is a shortest u–x path, where $q_1 = u$ and $q_i = x$. Obviously, $q_j \in L_j$ for all j. We now have three cases (see **Fig. 3**).

[Case 1] $\{\{x, p_{i+1}\}, \{y, p_{i+1}\}\} \cap E(G) \neq \emptyset$: By symmetry, we may assume $\{x, p_{i+1}\} = \{q_i, p_{i+1}\} \in E(G)$. Then, $(q_1, \dots, q_i, p_{i+1}, \dots, p_\ell)$ is a u–v path. Hence, y has a neighbor in $\{q_{i-1}, q_i, p_{i+1}\}$. Since $q_i = x$ and $\{q_{i-1}, q_i\}, \{q_i, p_{i+1}\} \in E(G)$, we have $d_G(x, y) \le 2$.

[Case 2] $\{\{x, p_i\}, \{y, p_i\}\} \cap E(G) \neq \emptyset$: By symmetry, we may assume $\{x, p_i\} = \{q_i, p_i\} \in E(G)$. Then, $(q_1, \dots, q_i, p_i, p_{i+1}, \dots, p_\ell)$ is a u–v path. Hence, y has a neighbor in $\{q_{i-1}, q_i, p_i, p_{i+1}\}$. By Case 1, if $\{y, p_{i+1}\} \in E(G)$, then $d_G(x, y) \le 2$. Otherwise, y has a neighbor in $\{q_{i-1}, q_i, p_i\}$. Since $q_i = x$ and $\{q_{i-1}, q_i\}, \{q_i, p_i\} \in E(G)$, we have $d_G(x, y) \le 2$.

[Case 3] $\{\{x, p_{i-1}\}, \{y, p_{i-1}\}\} \cap E(G) \neq \emptyset$: By Cases 1 and 2, it suffices to consider the case of $\{x, p_i\}, \{x, p_{i+1}\}, \{y, p_i\}, \{y, p_{i+1}\} \notin E(G)$. Clearly, this assumption implies $\{x, p_{i-1}\}, \{y, p_{i-1}\} \in E(G)$, and hence, $d_G(x, y) \le 2$. □

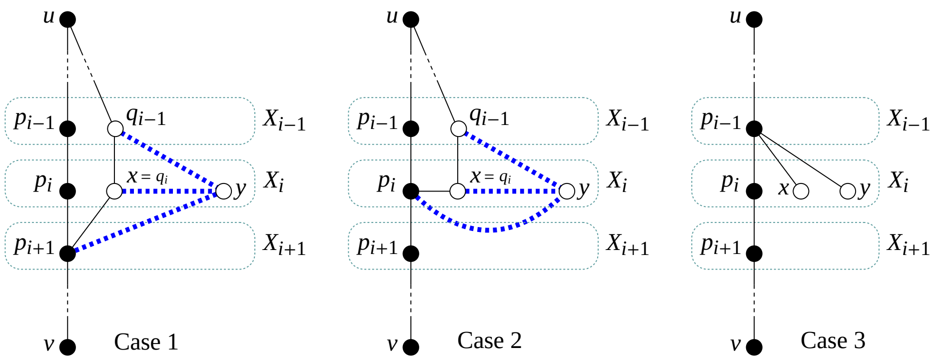

Fig. 3 The cases in the proof of Lemma 4.6.

Theorem 4.5 and Lemmas 4.2 and 4.6 imply the following better approximation result for AT-free graphs.

Theorem 4.7. *For a connected AT-free graph with n vertices and m edges, the path-distance-width can be approximated within a factor 3 in $O(m + n)$ time.*

We now show that the factor 3 is the best possible even for interval graphs (thus for AT-free graphs) if we use rooted distance structures.

Proposition 4.8. *The approximation ratio 3 of the path-distance-width for interval graphs cannot be improved if we select only one vertex as the initial set.*

Proof. The *friendship graph* F_d is the graph with $V(F_d) = \{c\} \cup \{u_i, v_i \mid 1 \le i \le d\}$ and $E(F_d) = \{\{u_i, v_i\} \mid 1 \le i \le d\} \cup \{\{c, w\} \mid w \in V(F_d) \setminus \{c\}\}$. For any d, F_d is an interval graph (see **Fig. 4**).

Let c be the center of F_{3d}, and let $w \in V(F_{3d}) \setminus \{c\}$. Clearly, $\mathrm{pdw}_{\{c\}}(F_{3d}) = 6d$ and $\mathrm{pdw}_{\{w\}}(F_{3d}) = 6d - 3$. On the other hand, if $S = \{u_i \mid 1 \le i \le 2d\}$, then

$$\begin{aligned}\mathrm{pdw}_S(F_{3d}) &= \max\{|\{u_i \mid 1 \le i \le 2d\}|, |\{c\} \cup \{v_i \mid 1 \le i \le 2d\}|, |\{u_i, v_i \mid 2d+1 \le i \le 3d\}|\}\\ &= \max\{2d, 2d+1, 2d\} = 2d + 1.\end{aligned}$$

Thus, if we use only one vertex of F_{3d} as an initial set, then the approximation ratio is at least $(6d - 3)/(2d + 1) = 3 - 6/(2d + 1)$. Since $6/(2d + 1)$ can be arbitrarily small by increasing d, the proposition holds. □

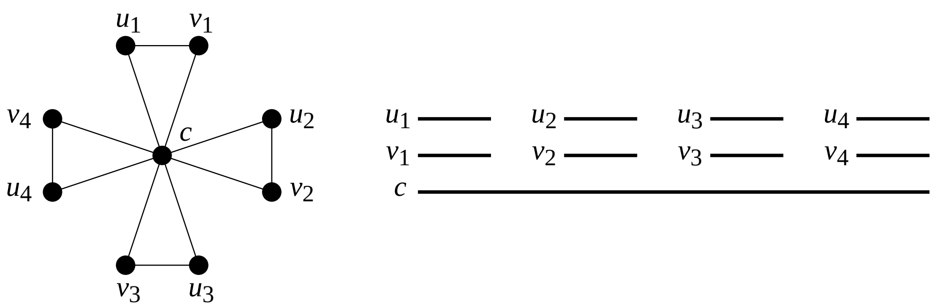

Fig. 4 Friendship graph F_4 and its interval representation.

4.3 Approximating the path-distance-width for proper interval graphs

Since proper interval graphs are AT-free, the result in the previous section provides an approximation algorithm for proper interval graphs as well. Fortunately, if we use proper interval representations, then we get a better approximation ratio.

Corneil, Kim, Natarajan, Olariu, and Sprague[6] [Proposition 2.1(2)] showed that in the rooted distance structure of a proper interval graphs rooted at the left most interval, every level is a clique.

Proposition 4.9 ([6]). *Let G be a connected proper interval graph, and let $u \in V(G)$ be the vertex with the left most starting point in some proper interval representation of G. Let L_i be the set of vertices of distance i from u; that is, $L_i = \{v \in V(G) \mid d_G(u, v) = i\}$. Then, for any i, $\mathrm{diam}_G(L_i) = 1$ if $L_i \neq \emptyset$.*

It is known that a proper interval representation of a proper interval graph can be computed in linear time (see e.g.[6]). Thus the left most vertex u in the above proposition and the rooted distance structure rooted at u can be found in linear time. Therefore, by Lemma 4.2, the next theorem holds.

Theorem 4.10. *For a connected proper interval graph G with n vertices and m edges, the path-distance-width can be approximated within a factor 2 in $O(m + n)$ time.*

Since the complete graph K_{2n} is a proper interval graph, $\mathrm{pdw}(K_{2n}) = n$, and $\mathrm{rpdw}(K_{2n}) = 2n - 1$, we can conclude that the factor 2 in the above theorem cannot be improved by any algorithm using rooted distance structures only.

Proposition 4.11. *The approximation ratio 2 of the path-distance-width for proper interval graphs cannot be improved if we select only one vertex as the initial set.*

5. Linear-time algorithm for cochain graphs

In this section, we present a linear-time algorithm to determine the path-distance-width of cochain graphs. Recall that every cochain graph is a proper interval graph.

Theorem 5.1 ([11]). *Given cochain graph G with n vertices and m edges, its bipartition (X, Y) and orderings on X and Y (which satisfies the definition) can be computed in $O(m + n)$ time.*

Theorem 5.2. *The path-distance-width of a connected cochain graph G with n vertices and m edges can be computed in $O(m + n)$ time.*

Proof. Assume G is a cochain graph with bipartition (X, Y). By Theorem 5.1, such a bipartition can be computed in $O(m + n)$ time. For convenience, let $\mathrm{pdw}(G, X) = \min\{\mathrm{pdw}_S(G) \mid S \subseteq X\}$ and $\mathrm{pdw}(G, Y) = \min\{\mathrm{pdw}_S(G) \mid S \subseteq Y\}$. If $S \subseteq V(G)$ intersects both X and Y, then $\mathrm{pdw}_S(G) \geq \lceil |V(G)|/2 \rceil$. It is easy to see that $\min\{\mathrm{pdw}(G, X), \mathrm{pdw}(G, Y)\} \leq \lceil |V(G)|/2 \rceil$. Therefore,

$$\mathrm{pdw}(G) = \min\{\mathrm{pdw}(G, X), \mathrm{pdw}(G, Y)\}.$$

By symmetry, it is sufficient to show that $\mathrm{pdw}(G, X)$ can be computed in $O(m + n)$ time.

Let $X = \{x_1, \ldots, x_p\}$ and $N_G[x_1] \subseteq N_G[x_2] \subseteq \cdots \subseteq N_G[x_p]$. By Theorem 5.1, such an ordering can be computed in linear time. We also compute in linear time $|X|$, $|Y|$, and $\mathrm{degree}_G(v)$ for each $v \in V(G)$. Let $Y_\emptyset = \{y \in Y \mid N_G(y) \cap X = \emptyset\}$. Clearly, $Y_\emptyset = \{y \in Y \mid \mathrm{degree}_G(y) = |Y| - 1\}$, and thus $|Y_\emptyset|$ can be obtained in linear time.

To compute $\mathrm{pdw}(G, X)$, we define $\mathrm{pdw}(G, X, i)$ as follows:

$$\mathrm{pdw}(G, X, i) = \min\{\mathrm{pdw}_S(G) \mid S \subseteq X,\ i = \max\{j \mid x_j \in S\}\}.$$

For $x_i \in X$, we denote $N_G(x_i) \cap Y$ by $N^Y_G(x_i)$. It is easy to see that $|N^Y_G(x_i)| = \mathrm{degree}_G(x_i) - (|X| - 1)$. If $i = \max\{j \mid x_j \in S\}$ for some $S \subseteq X$, then $N_G(x_i) \cap Y = N_G(S) \cap Y$ since $N_G[x_j] \subseteq N_G[x_i]$ for all $j < i$. Note that $N^Y_G(x_i)$ may be empty. We shall prove that $\mathrm{pdw}(G, X, i)$ can be computed in constant time by using $|X|$, $|Y|$, $|Y_\emptyset|$, and $|N^Y_G(x_i)|$. This will imply $\mathrm{pdw}(G, X)$ can be computed in linear time, since $\mathrm{pdw}(G, X) = \min_{1 \leq i \leq p} \mathrm{pdw}(G, X, i)$.

Let $S \subseteq \{x_1, \ldots, x_i\}$ and $x_i \in S$, and let D be the distance structure with the initial set S. We have the following three cases (see **Fig. 5**):

$$D = \begin{cases} (S,\ (X \setminus S) \cup N^Y_G(x_i),\ Y \setminus N^Y_G(x_i)) & \text{if } N^Y_G(x_i) \neq \emptyset, \\ (S,\ X \setminus S,\ Y) & \text{if } N^Y_G(x_i) = \emptyset \text{ and } Y_\emptyset = \emptyset, \\ (S,\ X \setminus S,\ Y \setminus Y_\emptyset,\ Y_\emptyset) & \text{if } N^Y_G(x_i) = \emptyset \text{ and } Y_\emptyset \neq \emptyset. \end{cases}$$

In any case, the average size of the first and second levels is $(|X| + |N^Y_G(x_i)|)/2$. Therefore, by setting $|S| = \min\{i, \lceil (|X| + |N^Y_G(x_i)|)/2 \rceil\}$, we can minimize the difference. One possible solution is $S = \{x_i\} \cup \{x_1, \ldots, x_{|S|-1}\}$. Since $\mathrm{pdw}_S(G)$ can be computed in constant time with $|S|$, $|X|$, $|Y|$, $|Y_\emptyset|$, and $|N^Y_G(x_i)|$, the theorem holds. Observe that, in any case, the location of the vertices in Y is solely determined by x_i. Thus the only thing we can do is to select the size of S arbitrarily from $\{1, \ldots, i\}$. Obviously, minimizing the difference of sizes between the first and second levels is the best solution here, since the vertices in X lie in these levels. □

6. Concluding remarks

We have considered the problem of determining the path-distance-width of graphs in important graph classes. It turned out that the problem is NP-hard even for cobipartite

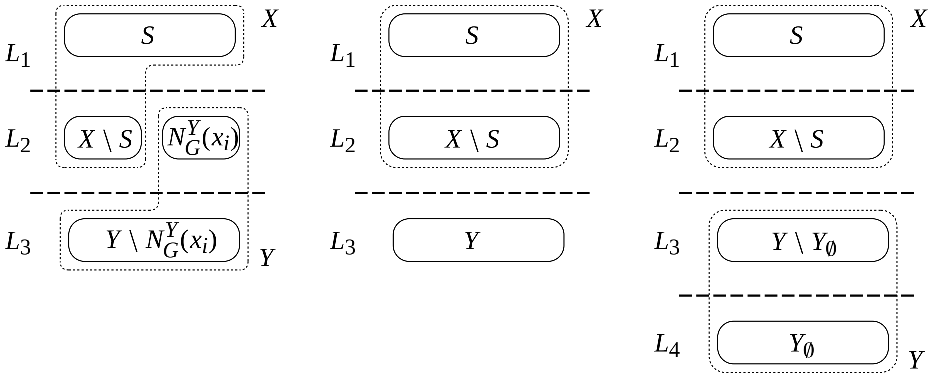

Fig. 5 Three cases in the proof of Theorem 5.2.

graphs, and thus for cocomparability graphs and AT-free graphs. However, using their chain-like structures, we were able to present constant-factor approximation algorithms. The algorithms are very simple and fast. We also present polynomial time (exact) algorithms for cochain graphs. The computational complexity of the problem for interval graphs and proper interval graphs remains open.

References

1) G.Blache, M.Karpinski, and J.Wirtgen. On approximation intractability of the bandwidth problem, 1998. ECCC TR98-014.
2) A.Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey*. SIAM, 1999.
3) T.M. Chan. All-pairs shortest paths for unweighted undirected graphs in $o(mn)$ time. In *SODA 2006*, pages 514–523, 2006.
4) J.-M. Chang, C.-W. Ho, and M.-T. Ko. Powers of asteroidal triple-free graphs with applications. *Ars Combin.*, 67:161–173, 2003.
5) D.Coppersmith and S.Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9:251–280, 1990.
6) D.G. Corneil, H.Kim, S.Natarajan, S.Olariu, and A.P. Sprague. Simple linear time recognition of unit interval graphs. *Inform. Process. Lett.*, 55:99–104, 1995.
7) D.G. Corneil, S.Olariu, and L.Stewart. Asteroidal triple-free graphs. *SIAM J. Discrete Math.*, 10:399–430, 1997.
8) D.G. Corneil, S.Olariu, and L.Stewart. Linear time algorithms for dominating pairs in asteroidal triple-free graphs. *SIAM J. Comput.*, 28:1284–1297, 1999.
9) M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
10) P.Golovach, P.Heggernes, D.Kratsch, D.Lokshtanov, D.Meister, and S.Saurabh. Bandwidth on AT-free graphs. In *ISAAC 2009*, volume 5878 of *Lecture Notes in Comput. Sci.*, pages 573–582. Springer-Verlag, 2009.
11) P.Heggernes and D.Kratsch. Linear-time certifying recognition algorithms and forbidden induced subgraphs. *Nordic J. Comput.*, 14:87–108, 2007.
12) D.S. Johnson. The NP-completeness column: An ongoing guide. *J. Algorithms*, 6:434–451, 1985.
13) H.Kaplan and R.Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques. *SIAM J. Comput.*, 25:540–561, 1996.
14) T.Kloks, D.Kratsch, and H.Müller. Approximating the bandwidth for asteroidal triple-free graphs. *J. Algorithms*, 32:41–57, 1999.
15) Y.Kobayashi. Private communication. September 2010.
16) A.Parra and P.Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Appl. Math.*, 79:171–188, 1997.
17) R.Seidel. On the all-pairs-shortest-path problem in unweighted undirected graphs. *J. Comput. System Sci.*, 51:400–403, 1995.
18) A.P. Sprague. An $O(n \log n)$ algorithm for bandwidth of interval graphs. *SIAM J. Discrete Math.*, 7:213–220, 1994.
19) K.Yamazaki. On approximation intractability of the path-distance-width problem. *Discrete Appl. Math.*, 110:317–325, 2001.
20) K.Yamazaki, H.L. Bodlaender, B.deFuiter, and D.M. Thilikos. Isomorphism for graphs of bounded distance width. *Algorithmica*, 24:105–127, 1999.