Discrete Applied Mathematics 345 (2024) 170–189

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

s-Club Cluster Vertex Deletion on interval and well-partitioned chordal graphs ☆

Dibyayan Chakraborty [a,*], L. Sunil Chandran [b], Sajith Padinhatteeri [c], Raji R. Pillai [b]

[a] *School of Computing, University of Leeds, United Kingdom*
[b] *Indian Institute of Science, Bengaluru, India*
[c] *BITS-Pilani, Hyderabad, India*

ARTICLE INFO

Article history:
Received 17 February 2023
Received in revised form 27 November 2023
Accepted 29 November 2023
Available online 7 December 2023

Keywords:
Vertex deletion problem
Cluster Vertex Deletion
s-Club Cluster Vertex Deletion
Well-partitioned chordal graphs
Interval graphs

ABSTRACT

In this paper, we study the computational complexity of s-Club Cluster Vertex Deletion. Given a graph, s-Club Cluster Vertex Deletion (s-CVD) aims to delete the minimum number of vertices from the graph so that each connected component of the resulting graph has a diameter at most s. When $s = 1$, the corresponding problem is popularly known as Cluster Vertex Deletion (CVD). We provide a faster algorithm for s-CVD on *interval graphs*. For each $s \geq 1$, we give an $O(n(n+m))$-time algorithm for s-CVD on interval graphs with n vertices and m edges. In the case of $s = 1$, our algorithm is a slight improvement over the $O(n^3)$-time algorithm of Cao et al. (2018), and for $s \geq 2$, it significantly improves the state-of-the-art running time $\left(O\left(n^4\right)\right)$.

We also give a polynomial-time algorithm to solve CVD on *well-partitioned chordal graphs*, a graph class introduced by Ahn et al. (WG 2020) as a tool for narrowing down complexity gaps for problems that are hard on chordal graphs, and easy on split graphs. Our algorithm relies on a characterisation of the optimal solution and on solving polynomially many instances of the Weighted Bipartite Vertex Cover. This generalises a result of Cao et al. (2018) on split graphs. We also show that for any even integer $s \geq 2$, s-CVD is NP-hard on well-partitioned chordal graphs.

1. Introduction

Detecting "*highly-connected*" parts or "*clusters*" of a complex system is a fundamental research topic in network science [28,38] with numerous applications in computational biology [6,12,30,34,35], machine learning [5], image processing [37], etc. In a graph-theoretic approach, a complex system or a network is often viewed as an undirected graph G that consists of a set of *vertices* $V(G)$ representing the atomic entities of the system and a set of *edges* $E(G)$ representing a binary relationship among the entities. A *cluster* is often viewed as a dense subgraph (often a *clique*), and *partitioning* a graph into such clusters is one of the main objectives of *graph-based data clustering* [6,13,33].

Ben-Dor et al. [6] and Shamir et al. [33] observed that the clusters of certain networks might be retrieved by making a small number of modifications in the network. These modifications may be required to account for the errors introduced

☆ An extended abstract Chakraborty et al. (2022) [11] of this paper was presented at WG 2022, and this version contains all proofs missing from the conference version.

* Corresponding author.
E-mail address: D.chakraborty@leeds.ac.uk (D. Chakraborty).

https://doi.org/10.1016/j.dam.2023.11.048

during the construction of the network. In graph-theoretic terms, the objective is to modify (*e.g.* edge deletion, edge addition, vertex deletion) a given input graph as little as possible so that each component of the resulting graph is a cluster. When deletion of vertices is the only valid operation on the input graph, the corresponding clustering problem falls in the category of *vertex deletion* problems, a core topic in algorithmic graph theory. Many classic optimisation problems like Maximum Clique, Maximum Independent Set, Vertex cover are examples of vertex deletion problems. In this paper, we study popular vertex deletion problems called Cluster Vertex Deletion and its generalisation s-Club Cluster Vertex Deletion, both being important in the context of graph-based data clustering.

Given a graph G, the objective of Cluster Vertex Deletion (CVD) is to delete a minimum number of vertices so that the remaining graph is a set of disjoint cliques. Below we give a formal definition of CVD.

Cluster Vertex Deletion (CVD)
Input: An undirected graph G, and an integer k.
Output: Yes, if there is a set S of vertices with $|S| \leq k$, such that each component of the graph induced by $V(G) \setminus S$ is a clique. No, otherwise.

The term Cluster Vertex Deletion was coined by Gramm et al. [19] in 2004. However, NP-hardness of CVD, even on planar graphs and bipartite graphs, follows from the seminal works of Yannakakis [39] and Lewis & Yannakakis [24] from four decades ago. Since then many researchers have proposed *parameterised algorithms* and *approximation algorithms* for CVD on general graphs [3,8,15–17,20,31,36,40]. In this paper, we focus on polynomial-time solvability of CVD on special classes of graphs.

Cao et al. [9] gave polynomial-time algorithms for CVD on *interval* graphs (see Definition 2) and *split* graphs. Chakraborty et al. [10] gave a polynomial-time algorithm for CVD on *trapezoid* graphs. However, much remains unknown: Chakraborty et al. [10] pointed out that computational complexity of CVD on *planar bipartite* graphs and *cocomparability* graphs is unknown. Cao et al. [9] asked if CVD can be solved on chordal graphs (graphs with no induced cycle of length greater than 3) in polynomial-time.

Ahn et al. [1] introduced *well-partitioned chordal* graphs (see Definition 1) as a tool for narrowing down complexity gaps for problems that are hard on chordal graphs, and easy on split graphs. Since several problems (for example, transversal of longest paths and cycles, tree 3-spanner problem, geodetic set problem) which are either hard or open on chordal graphs become polynomial-time solvable on well-partitioned chordal graphs [1], the computational complexity of CVD on well-partitioned chordal graphs is a well-motivated open question.

In this paper, we also study a generalisation of CVD known as s-Club Cluster Vertex Deletion (s-CVD). In many applications, the equivalence of cluster and clique is too restrictive [2,4,29]. For example, in protein networks where proteins are the vertices and the edges indicate the interaction between the proteins, a more appropriate notion of clusters may have a diameter (maximum length of the shortest paths) of more than 1 [4]. Therefore researchers have defined the notion of *s-clubs* [4,11,26]. An s-club is a graph with *diameter* at most s. The objective of s-Club Cluster Vertex Deletion (s-CVD) is to delete the minimum number of vertices from the input graph so that all connected components of the resultant graph are an s-club. Below we give a formal definition of s-CVD.

s-Club Cluster Vertex Deletion (s-CVD)
Input: An undirected graph G, and integers k and s.
Output: Yes, if there is a set S of vertices with $|S| \leq k$, such that each component of the graph induced by $V(G) \setminus S$ has diameter at most s. No, otherwise.

Schäfer [32] introduced the notion of s-CVD and gave a polynomial-time algorithm for s-CVD on trees. Researchers have studied the particular case of 2-CVD as well [14,25]. In general, s-CVD remains NP-hard on planar bipartite graphs for each $s \geq 2$, APX-hard on split graphs for $s = 2$ [10] (contrasting the polynomial-time solvability of CVD on split graphs). Combination of the ideas of Cao et al. [9] and Schäfer [32], provides an $O(n^8)$-time algorithm for s-CVD on trapezoid graphs (intersection graphs of trapezoids between two horizontal lines) with n vertices [10]. This algorithm can be modified to give an $O(n^4)$-time algorithm for s-CVD on interval graphs with n vertices.

General notations: For a graph G, let $V(G)$ and $E(G)$ denote the set of vertices and edges, respectively. For a vertex $v \in V(G)$, the set of vertices adjacent to v is denoted by $N(v)$ and $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, let $G - S$ be an induced graph obtained by deleting the vertices in S from $V(G)$. For two sets S_1, S_2, let $S_1 - S_2$ denote the set obtained by deleting the elements of S_2 from S_1. The set $S_1 \Delta S_2$ denotes $(S_1 \cup S_2) - (S_1 \cap S_2)$. In this work, while writing equations, we use the symbol G and $V(G)$ interchangeably when the meaning is clear from the context. For example, in certain cases, set operations like, $G \cap S$ (or $G - S$) for a set S implies the set of vertices $V(G) \cap S$ (or $V(G) - S$). This difference would be clear to the reader from the context.

2. Our contributions

In this section, we state our results formally. We start with the definition of well-partitioned chordal graphs as given in [1].

Definition 1 (*[1]*)**.** A connected graph G is a well-partitioned chordal graph if there exists a partition $\mathcal{P}$ of $V(G)$ and a tree $\mathcal{T}$ having $\mathcal{P}$ as a vertex set such that the following hold.

(a) Each part $X \in \mathcal{P}$ is a clique in G.
(b) For each edge $XY \in E(\mathcal{T})$, there exist $X' \subseteq X$ and $Y' \subseteq Y$ such that edge set of the bipartite graph $G[X, Y]$ is $X' \times Y'$.
(c) For each pair of distinct $X, Y \in V(\mathcal{T})$ with $XY \notin E(\mathcal{T})$, there is no edge between a vertex in X and a vertex in Y.

The tree $\mathcal{T}$ is called a *partition tree* of G, and the elements of $\mathcal{P}$ are called its *bags* or *nodes* of $\mathcal{T}$.

Our first result is on CVD for well-partitioned chordal graphs, which generalises a result of Cao et al. [9] for split graphs. We prove the following theorem in Section 3.

Theorem 1. *Given a connected well-partitioned chordal graph G and its partition tree, there is an $O(m^2n)$-time algorithm to solve* CVD *on G, where n and m are the number of vertices and edges, respectively.*

Since a partition tree of a well-partitioned chordal graph can be obtained in polynomial time [1], the above theorem adds CVD to the list of problems that are open on chordal graphs but admits polynomial-time algorithm on well-partitioned chordal graphs.

Our algorithm relies on a characterisation of the solution set. We show that the optimal solution of a well-partitioned chordal graph with m edges can be obtained by finding minimum weighted vertex cover [23] of m many weighted bipartite graphs with weights at most n. Then standard *Max-flow* based algorithms [22,23,27] from the literature yield Theorem 1. On the negative side, we prove the following theorem in Section 5.

Theorem 2. *Unless the Unique Games Conjecture is false, for any even integer $s \geq 2$, there is no $(2 - \epsilon)$-approximation algorithm for s-*CVD *on well-partitioned chordal graphs.*

Our third result is a faster algorithm for s-CVD on *interval graphs*.

Definition 2. A graph G is an interval graph if there is a collection $\mathcal{I}$ of intervals on the real line such that each vertex of the graph can be mapped to an interval and two intervals intersect if and only if there is an edge between the corresponding vertices in G. The set $\mathcal{I}$ is an interval representation of G.

We prove the following theorem in Section 4.

Theorem 3. *For each $s \geq 1$, there is an $O(n(n + m))$-time algorithm to solve s-*CVD *on interval graphs with n vertices and m edges.*

We note that our techniques deviate significantly from the ones in the previous literature [9,10,32]. We show that the optimal solution (for s-CVD on interval graphs) must be one of "four types" and the optimum for each of the "four types" can be found by solving s-CVD on $O(m + n)$ many induced subgraphs. Furthermore, we exploit the "linear" structure of interval graphs to ensure that the optimal solution in each case can be found in $O(n)$-time. Our result significantly improves the state-of-the-art running time $\left(O\left(n^4\right), \text{see [10]}\right)$ for s-CVD on interval graphs.

3. Polynomial time algorithm for CVD on well-partitioned chordal graphs

In this section, we shall give a polynomial-time algorithm to solve CVD on well-partitioned chordal graphs. In the first subsection, we present the main ideas of our algorithm and describe our techniques for proving Theorem 1.

3.1. Overview of the algorithm

Let G be a well-partitioned chordal graph with a partition tree $\mathcal{T}$ rooted at an arbitrary node. For a node X, let $\mathcal{T}_X$ be the subtree rooted at X and G_X be the subgraph of G induced by the vertices in the nodes of $\mathcal{T}_X$. For two adjacent nodes X, Y of $\mathcal{T}$, *the boundary of X with respect to Y* is the set $bd(X, Y) = \{x \in X : N(x) \cap Y \neq \emptyset\}$ and the set $E(X, Y) = \{xy | x \in X, y \in Y \text{ and } xy \in E(G)\}$. Each edge $e \in E(X, Y)$ is called as an (X, Y)-edge. For a node X, $P(X)$ denotes the parent of X in $\mathcal{T}$. We denote minimum CVD sets of G_X and $G_X - bd(X, P(X))$ as $OPT(G_X)$ and $OPT(G_X - bd(X, P(X)))$, respectively. We shall use the above notations extensively in describing our algorithm and proofs.

Our dynamic programming-based algorithm traverses $\mathcal{T}$ in a post-order fashion and for each node X of $\mathcal{T}$, computes $OPT(G_X)$ and $OPT(G_X - bd(X, P(X)))$. A set S of vertices is a CVD *set* of G if $G - S$ is disjoint union of cliques. At the heart of our algorithm lies a characterisation of CVD sets of G_X, showing that any CVD set of G_X can be exactly one of two types, namely, X-*CVD set* or (X, Y)-*CVD set* where Y is a child of X (see Definitions 4 and 5). Informally, for a node X, a CVD set is an X-*CVD set* if it contains X or removing it from G_X creates a cluster all of whose vertices are from X. On the contrary, a CVD set is an (X, Y)-*CVD set* if its removal creates a cluster intersecting both X and Y, where Y is a child of X. In Lemma 4, we formally show that any CVD set of G_X must be one of the above two types.

To compute a minimum X-CVD set, first, we construct a weighted bipartite graph $\mathcal{H}$ which is defined in Section 3.3 and show that a minimum weighted vertex cover of $\mathcal{H}$ can be used to construct a minimum X-CVD set of G (see Eqs. (3), (4), (5), (6)). Then in Section 3.4, we show that the subroutine for finding minimum X-CVD sets can be used to get a minimum (X, Y)-CVD set for each child Y of X. Finally, in Section 3.5, we combine our tools and give an $O(m^2n)$-time algorithm to find a minimum CVD set of a well-partitioned chordal graph G with n vertices and m edges.

3.2. Definitions and lemma

In this section, we introduce some definitions and prove the lemma that facilitates the construction of a polynomial-time algorithm for finding a minimum CVD set of well-partitioned chordal graphs.

Definition 3. A *cluster* C of a graph G is a connected component that is isomorphic to a complete graph.

Definition 4. Let G be a well-partitioned chordal graph, $\mathcal{T}$ be its partition tree, and X be the root node of $\mathcal{T}$. A CVD set S of G is an "X-CVD set" if either $X \subseteq S$ or $G - S$ contains a cluster $C \subseteq X$.

Definition 5. Let G be a well-partitioned chordal graph, $\mathcal{T}$ be its partition tree, and X be the root node of $\mathcal{T}$. Let Y be a child of X.

A CVD set S is a "(X, Y)-CVD set" if $G - S$ has a cluster C such that $C \cap X \neq \emptyset$ and $C \cap Y \neq \emptyset$.

Lemma 4. *Let S be a CVD set of G. Then exactly one of the following holds.*

(a) The set S is a X-CVD set.
(b) There is exactly one child Y of X in $\mathcal{T}$ such that S is an (X, Y)-CVD set of G.

Proof. If $X \subseteq S$ or $G - S$ has a cluster contained in X, then S is an X-CVD set. Otherwise, $X^* = (G - S) \cap X \neq \emptyset$ and since X^* is a clique, $G - S$ must contain a cluster C such that $X^* \subset V(C) \nsubseteq X$. Therefore, C should intersect with at least one child of X. Let Y_1, Y_2 be children of X. If both $V(C) \cap Y_1 \neq \emptyset$ and $V(C) \cap Y_2 \neq \emptyset$, then C is not a cluster because Y_1 and Y_2 are non-adjacent nodes of $\mathcal{T}$. Hence C intersects exactly one child of X. □

3.3. Finding minimum X-CVD sets

In this section, we prove the following theorem.

Theorem 5. *Let G be a connected well-partitioned chordal graph and $\mathcal{T}$ be a partition tree of G rooted at X. Assume for each node $Y \in V(\mathcal{T}) - \{X\}$ both $OPT(G_Y)$ and $OPT(G_Y - bd(Y, P(Y)))$ are given. Then a minimum X-CVD set of G can be computed in $O(|E(G)| \cdot |V(G)|)$ time.*

For the remainder of this section, we denote by G a fixed well-partitioned chordal graph with a partition tree $\mathcal{T}$ rooted at X. Let $X_1, X_2, \ldots, X_t$ be the children of X. The main idea behind our algorithm for finding minimum X-CVD set of G is to construct an auxiliary vertex weighted bipartite graph $\mathcal{H}$ with at most $|V(G)|$ vertices such that the (minimum) vertex covers of $\mathcal{H}$ can be used to construct (minimum) X-CVD set. Below we describe the construction of $\mathcal{H}$.

Let $\mathcal{B} = \{bd(X_i, X) : i \in [t]\}$. The vertex set of $\mathcal{H}$ is $X \cup \mathcal{B}$ and the edge set of $\mathcal{H}$ is defined as

$$E(\mathcal{H}) = \{uB : u \in X, B \in \mathcal{B}, \forall v \in B, uv \in E(G)\} \tag{1}$$

The weight function on the vertices of $\mathcal{H}$ is defined as follows. For each vertex $u \in X$, define $w(u) = 1$ and for each set $B \in \mathcal{B}$ where $B = bd(X_j, X)$, define

$$w(B) = |B| + \left|OPT(G_{X_j} - B)\right| - \left|OPT(G_{X_j})\right| \tag{2}$$

Remark 1. Since $B \cup OPT(G_{X_j} - B)$ is a CVD set of G_{X_j}, we have $|OPT(G_{X_j})| \leq |B| + |OPT(G_{X_j} - B)|$ and therefore $w(B) \geq 0$.

Below we show how minimum weighted vertex covers of $\mathcal{H}$ can be used to compute the minimum X-CVD set of G. For an X-CVD set Z of G, define $Cov(Z) = (X \cap Z) \cup \{B \in \mathcal{B} : B \subseteq Z\}$.

Lemma 6. *Let Z be an X-CVD set of G. Then $Cov(Z)$ is a vertex cover of $\mathcal{H}$.*

Proof. Assume that $Cov(Z)$ is not a vertex cover of $\mathcal{H}$. Then there exists at least one edge $e = uB$ in $\mathcal{H} - Cov(Z)$. Hence from the definition of $Cov(Z)$, we infer that $u \in X - Z$ and $B \nsubseteq Z$. Let C_u be the cluster of $G - Z$ that contains the vertex u. Since X is a clique, $X - Z \subseteq C_u$. Observe that since uB is an edge of $\mathcal{H}$, there exists a vertex $w \in B$ such that $uw \in E(G)$. Then the definition of partition tree $\mathcal{T}$ and B implies that all vertices of B are contained in $N(u)$. Since $B \nsubseteq Z$ it follows that there exists at least one vertex $v \in B$ in $G - Z$ such that $uv \in E(G - Z)$ and hence $v \in C_u$. Therefore, the cluster C_u intersects the child of X that contains B, which contradicts the assumption that Z is an X-CVD set of G (see definition of X-CVD set). □

For a vertex cover D of $\mathcal{H}$, define

$$S_1(D) = D \cap X \tag{3}$$

$$S_2(D) = \bigcup_{\substack{B \in D \cap \mathcal{B} \\ B = bd(X_i, X)}} B \cup OPT(G_{X_i} - bd(X_i, X)) \tag{4}$$

$$S_3(D) = \bigcup_{\substack{B \in \mathcal{B} - D \\ B = bd(X_i, X)}} OPT(G_{X_i}) \tag{5}$$

$$Sol(D) = S_1(D) \cup S_2(D) \cup S_3(D) \tag{6}$$

Note that, by definition $S_i(D) \cap S_j(D) = \emptyset$, $1 \le i < j \le 3$. We have the following lemma.

Lemma 7. *Let D be a vertex cover of $\mathcal{H}$. Then $Sol(D)$ is an X-CVD set of G.*

Proof. Suppose for the sake of contradiction that $Sol(D)$ is not an X-CVD set of G. First, assume $Sol(D)$ is not a CVD set of G. Then there exists an induced path $P = uvw$ in $G - Sol(D)$. Consider the following cases.

1. $X \cap \{u, v, w\} = \emptyset$. Then there must exist a child Y of X such that u, v, w are vertices of G_Y. If $B = bd(Y, X) \in D$, then by Eqs. (4) and (6), $Sol(D)$ contains $B \cup OPT(G_Y - B)$. But then $B \cup OPT(G_Y - B)$ is not a CVD set of G_Y, a contradiction. If $B \notin D$, then by Eqs. (5) and (6), $Sol(D)$ contains $OPT(G_Y)$. But then $OPT(G_Y)$ is not a CVD set of G_Y, also a contradiction.
2. Otherwise, there always exist two adjacent vertices z_1, z_2 such that $\{z_1, z_2\} \subset \{u, v, w\}$ and $z_1 \in X$ and $z_2 \in Y$, where Y is a child of X. Observe that $z_2 \in B = bd(Y, X)$ and therefore z_1 is adjacent to B in $\mathcal{H}$. Since $\{z_1, z_2\} \cap Sol(D) = \emptyset$, $\mathcal{H} - D$ contains the edge z_1B, contradicting the fact that D is a vertex cover of $\mathcal{H}$.

Now assume that $Sol(D)$ is a CVD set but not an X-CVD set. Then there must exist a cluster C in $G - Sol(D)$ that contains an (X, Y)-edge uv where $u \in X$ and $v \in bd(Y, X)$. Therefore $u \notin D$ and $B = bd(Y, X) \notin D$. Then $\mathcal{H} - D$ contains the edge uB, contradicting the fact that D is a vertex cover of $\mathcal{H}$. □

A minimum weighted vertex cover D of $\mathcal{H}$ is also *minimal* if no proper subset of D is a vertex cover of $\mathcal{H}$. The minimality restriction avoids the inclusion of redundant vertices with weight 0 in the minimum vertex cover.

Observation 2. *Let D be a minimal minimum weighted vertex cover of $\mathcal{H}$. For any $i \in [t]$, either $bd(X, X_i) \subseteq D$ or $bd(X_i, X) \in D$, but not both.*

Proof. First assume $bd(X, X_i) \nsubseteq D$ and $B = bd(X_i, X) \notin D$. Observe that, the neighbourhood of B in $\mathcal{H}$ is $bd(X, X_i)$. Since $bd(X, X_i) \nsubseteq D$, there must exist a vertex $u \in (bd(x, X_i) - D) \subseteq X - D$. Then it follows that uB is an edge of $\mathcal{H} - D$. This contradicts the fact that D is a vertex cover of $\mathcal{H}$.

Now assume that both $bd(X, X_i) \subseteq D$ and $B = bd(X_i, X) \in D$. Since $\{x : xB \in E(\mathcal{H})\} = bd(X, X_i)$ the set $D - \{B\}$ is also a vertex cover of $\mathcal{H}$, a contradiction. □

From now on, D denotes a minimal minimum weighted vertex cover of $\mathcal{H}$, and Z denotes a fixed but arbitrary X-CVD set of G. Our goal is to show that $|Sol(D)| \le |Z|$. We need some more notations and observations.

First, we define the following four sets I_1, I_2, I_3, I_4. (Recall that $X_1, X_2, \ldots, X_t$ are children of the root X of the partition tree $\mathcal{T}$ of G.)

$$I_1 = \{i \in [t] : bd(X, X_i) \subseteq Sol(D) \text{ and } bd(X, X_i) \subseteq Z\} \tag{7}$$

$$I_2 = \{i \in [t] : bd(X, X_i) \subseteq Sol(D) \text{ and } bd(X, X_i) \nsubseteq Z\} \tag{8}$$

$$I_3 = \{i \in [t] - (I_1 \cup I_2) : bd(X_i, X) \subseteq Sol(D) \text{ and } bd(X_i, X) \subseteq Z\} \tag{9}$$

$$I_4 = \{i \in [t] - (I_1 \cup I_2) : bd(X_i, X) \subseteq Sol(D) \text{ and } bd(X_i, X) \nsubseteq Z\} \tag{10}$$

Note that $I_1 \cup I_2 \cup I_3 \cup I_4 = [t]$ and $(I_1 \cup I_2) \cap (I_3 \cup I_4) = \emptyset$. We have the following observations on the sets I_i, $1 \le i \le 4$.

Observation 3. *The sets I_1, I_2, I_3, I_4 form a partition of $[t]$.*

Proof. From the definition of I_i, $1 \le i \le 4$, it is clear that $I_i \cap I_j = \emptyset, i \ne j$. Assume that there exists an $i \in [t]$ such that $i \notin I_1 \cup I_2$. Hence, $bd(X, X_i) \nsubseteq Sol(D) \cap X = D \cap X$. Then by Observation 2, $bd(X_i, X) \in D$ and by Eq. (4) the set of vertices $bd(X_i, X) \subseteq Sol(D)$. Therefore each $i \in [t] - (I_1 \cup I_2)$ either belongs to the set I_3 or I_4. □

Lemma 8. *Let D be a vertex cover of $\mathcal{H}$ and $Sol(D)$ be an X-CVD set of G defined as in Eq. (6). For the sets I_i, $1 \le i \le 4$ defined by Eqs. (7)–(10), the following holds.*

(i) $\bigcup_{i \in I_1 \cup I_2} bd(X, X_i) = S_1(D)$
(ii) $\bigcup_{i \in I_3 \cup I_4} bd(X_i, X) \cup OPT(G_{X_i} - bd(X_i, X)) = S_2(D)$
(iii) $\bigcup_{i \in I_1 \cup I_2} OPT(G_{X_i}) = S_3(D)$.

Proof. First note that $S_1(D) = D \cap X = Sol(D) \cap X$ (by definition of $Sol(D)$). On the other hand, by definition of I_1 and I_2 we have $\bigcup_{i \in I_1 \cup I_2} bd(X, X_i) \subseteq Sol(D)$. Moreover, $\bigcup_{i \in I_1 \cup I_2} bd(X, X_i) \subseteq X$. Therefore, $\bigcup_{i \in I_1 \cup I_2} bd(X, X_i) \subseteq Sol(D) \cap X = S_1(D)$.

Now to prove the other side, $S_1(D) \subseteq \bigcup_{i \in I_1 \cup I_2} bd(X, X_i)$, suppose for the sake of contradiction that there exists a vertex $v \in S_1(D) - \bigcup_{i \in I_1 \cup I_2} bd(X, X_i)$. Let $J = \{j : v \in bd(X, X_j)\}$. Since $J \cap (I_1 \cup I_2) = \emptyset$, by definition of I_1 and I_2, for each $j \in J$, $bd(X, X_j) \nsubseteq Sol(D) \cap X = D \cap X$. Hence by Observation 2, $bd(X_j, X) \in D$, for all $j \in J$. Therefore, $D - \{v\}$ is also a vertex cover of $\mathcal{H}$, contradicting the minimality of D. By Observation 3, $I_3 \cup I_4 = [t] - I_1 \cup I_2$. Moreover, by the definition of I_1 and I_2, for each $i \in [t] - I_1 \cup I_2$ the set $bd(X, X_i) \nsubseteq Sol(D) \cap X = D \cap X$. Hence by Observation 2, we have $bd(X_i, X) \in D$ for each $i \in I_3 \cup I_4$ and $bd(X_i, X) \notin D, i \in I_1 \cup I_2$. Thus it follows from Observation 3 and the definition of S_2 and S_3 that $\bigcup_{i \in I_3 \cup I_4} bd(X_i, X) \cup OPT(G_{X_i} - bd(X_i, X)) = S_2(D)$ and $\bigcup_{i \in I_1 \cup I_2} OPT(G_{X_i}) = S_3(D)$. □

Based on the set I_1, we construct two sets D_1 and Z_1 from $Sol(D)$ and Z, respectively, which are defined as follows.

$$D_1 = \bigcup_{i \in I_1} bd(X, X_i) \cup (Sol(D) \cap G_{X_i}) \tag{11}$$

$$Z_1 = \bigcup_{i \in I_1} bd(X, X_i) \cup (Z \cap G_{X_i}) \tag{12}$$

Lemma 9. $|D_1| \le |Z_1|$.

Proof. From the definition of $Sol(D)$ and Eq. (3), for all $i \in I_1$, we infer that $bd(X, X_i) \subseteq D$. Hence, by Observation 2, $bd(X_i, X) \notin D, i \in I_1$ and from Eq. (5), $Sol(D) \cap G_{X_i} = OPT(G_{X_i})$. Since for each distinct $i, j \in I_1$, $G_{X_i} \cap G_{X_j} = \emptyset$ and $|Z \cap G_{X_i}| \ge |OPT(G_{X_i})|$, by the definitions of D_1 and Z_1 we have $|D_1| \le |Z_1|$. □

Based on the set I_2, we construct the following two sets $D_2 \subseteq Sol(D)$ and $Z_2 \subseteq Z$.

$$D_2 = \bigcup_{i \in I_2} bd(X, X_i) \cup (Sol(D) \cap G_{X_i}) - \bigcup_{i \in I_1} bd(X, X_i) \tag{13}$$

$$Z_2 = \bigcup_{i \in I_2} bd(X_i, X) \cup (Z \cap (G_{X_i} - bd(X_i, X))) \tag{14}$$

By the definition of the set I_2, the set of vertices $bd(X, X_i) \nsubseteq Z$, for all $i \in I_2$. By Lemma 6, recall that there exists a vertex cover, $Cov(Z)$ of $\mathcal{H}$ corresponding to every X-CVD-set Z. Since $bd(X, X_i) \nsubseteq Z$ and thus $bd(X, X_i) \nsubseteq Cov(Z)$, it is implicit in Observation 2 that $bd(X_i, X) \in Cov(Z)$. Hence $bd(X_i, X) \subseteq Z$ and the set $Z_2 \subseteq Z$.

Lemma 10. $|D_2| \le |Z_2|$.

Proof. By arguments similar to that in the proof of Lemma 9, for $i \in I_2$, $Sol(D) \cap G_{X_i} = OPT(G_{X_i})$. Hence, $D_2 = \bigcup_{i \in I_2} bd(X, X_i) \cup OPT(G_{X_i}) - \bigcup_{i \in I_1} bd(X, X_i)$. Suppose for contradiction that $|D_2| > |Z_2|$. Then by the definitions of D_2 and Z_2 we have

$$\left| \bigcup_{i \in I_2} bd(X, X_i) \cup OPT(G_{X_i}) - \bigcup_{i \in I_1} bd(X, X_i) \right| > \left| \bigcup_{i \in I_2} bd(X_i, X) \cup (Z \cap (G_{X_i} - bd(X_i, X))) \right|.$$

Since $X \cap V(G_{X_i}) = \emptyset$, for all $1 \leq i \leq t$ and $\left|Z \cap (G_{X_i} - bd(X_i, X))\right| \geq OPT(G_{X_i} - bd(X_i, X))$, we can rewrite the above inequality as follows.

$$\left|\bigcup_{i\in I_2} bd(X, X_i) - \bigcup_{i\in I_1} bd(X, X_i)\right| > \left|\bigcup_{i\in I_2} bd(X_i, X)\right| + \left|\bigcup_{i\in I_2} OPT(G_{X_i} - bd(X_i, X))\right| - \left|\bigcup_{i\in I_2} OPT(G_{X_i})\right|.$$

That is,

$$\left|\bigcup_{i\in I_2} bd(X, X_i) - \bigcup_{i\in I_1} bd(X, X_i)\right| > \sum_{i\in I_2}(|bd(X_i, X)| + \left|OPT(G_{X_i} - bd(X_i, X))\right| - \left|OPT(G_{X_i})\right|).$$

By Eq. (2), $|bd(X_i, X)| + \left|OPT(G_{X_i} - bd(X_i, X))\right| - \left|OPT(G_{X_i})\right| = w(bd(X_i, X))$ and hence,

$$\left|\bigcup_{i\in I_2} bd(X, X_i) - \bigcup_{i\in I_1} bd(X, X_i)\right| > \sum_{i\in I_2} w(bd(X_i, X)) \tag{15}$$

Recall that D is a minimal minimum weighted vertex cover of $\mathcal{H}$. By Lemma 8 we have $\bigcup_{i\in I_2} bd(X, X_i) \subseteq D$ and hence for each $i \in I_2$, the vertex $B = bd(X_i, X) \notin D$ by Observation 2. Now we show that if we delete the vertices in $\bigcup_{i\in I_2} bd(X, X_i) - \bigcup_{i\in I_1} bd(X, X_i)$ from D and add the set of vertices $\{bd(X_i, X) : i \in I_2\}$ then we get a vertex cover of smaller weight for $\mathcal{H}$ by inequality (15), a contradiction.

Claim 1. *Let D_1 be a set of vertices obtained from D by deleting the vertices in $\bigcup_{i\in I_2} bd(X, X_i) - \bigcup_{i\in I_1} bd(X, X_i)$ and by adding the set of vertices $\{bd(X_i, X) : i \in I_2\}$. Then, D_1 is a vertex cover of $\mathcal{H}$.*

Proof of Claim: Assume that there exists an edge $uB \in E(\mathcal{H} - D_1)$ where $B = bd(X_j, X)$, for some $j \in [t]$. Since $bd(X_j, X) \notin D_1$, by the definition of D_1 (given above) observe that $bd(X_j, X) \notin D$ and $j \notin I_2$. Note that the neighbourhood of $bd(X_j, X)$ in $\mathcal{H}$ is $bd(X, X_j)$ and hence $u \in bd(X, X_j)$. Since D is a vertex cover of $\mathcal{H}$, we have $bd(X, X_j) \subseteq D$. Now we show that $j \notin I_1$: By definition of D_1 we have $\bigcup_{i\in I_1} bd(X, X_i) \cap D \subseteq D_1$. Since $u \in bd(X, X_j)$ and $bd(X, X_j) \subseteq D$, if $j \in I_1$ then the vertex u remains in D_1. Thus no such edge uB exists in $\mathcal{H} - D_1$. Therefore, we infer that $j \notin I_1$. Since $j \notin I_1 \cup I_2$, from Lemma 8 we have $bd(X, X_j) \nsubseteq D \cap X$. Hence there exists a vertex $w \in bd(X, X_j)$ such that $w \in \mathcal{H} - D$. Moreover, by the definition of partition tree $\mathcal{T}$ and $bd(X, X_j)$ the edge $wB \in E(\mathcal{H} - D)$. This contradicts the assumption that D is a vertex cover of $\mathcal{H}$. ■

This completes the proof of the Lemma. □

Based on the set I_3, we construct the following two sets $D_3 \subseteq Sol(D)$ and $Z_3 \subseteq Z$.

$$D_3 = \bigcup_{i\in I_3} bd(X_i, X) \cup OPT(G_{X_i} - bd(X_i, X)) \tag{16}$$

$$Z_3 = \bigcup_{i\in I_3} bd(X_i, X) \cup (Z \cap (G_{X_i} - bd(X_i, X))) \tag{17}$$

Lemma 11. $|D_3| \leq |Z_3|$.

Proof. Since $\left|Z \cap (G_{X_i} - bd(X_i, X))\right| \geq OPT(G_{X_i} - bd(X_i, X))$, by the definitions of D_3 and Z_3 we have $|D_3| \leq |Z_3|$. □

Based on the set I_4, we construct the following two sets $D_4 \subseteq Sol(D)$ and $Z_4 \subseteq Z$.

$$D_4 = \bigcup_{i\in I_4} bd(X_i, X) \cup OPT(G_{X_i} - bd(X_i, X)) \tag{18}$$

$$Z_4 = \bigcup_{i\in I_4} bd(X, X_i) \cup (Z \cap (G_{X_i})) - \bigcup_{i\in I_1} bd(X, X_i) \tag{19}$$

By the definition of the set I_4, the set of vertices $bd(X_i, X) \nsubseteq Z, i \in I_4$. By Lemma 6, recall that there exists a vertex cover, $Cov(Z)$ of $\mathcal{H}$ corresponding to every X-CVD-set Z. Since $bd(X_i, X) \nsubseteq Z, i \in I_4$, by definition of $Cov(Z)$ we have $bd(X_i, X) \notin Cov(Z)$ and hence it is implicit in Observation 2 that $bd(X, X_i) \subseteq Cov(Z)$. Hence $bd(X, X_i) \subseteq Z$ and the set $Z_4 \subseteq Z$.

Lemma 12. $|D_4| \leq |Z_4|$.

Proof. Suppose for contradiction that $|D_4| > |Z_4|$. Then by the definitions of D_4 and Z_4 we have

$$\left|\bigcup_{i\in I_4}(bd(X_i,X)\cup OPT(G_{X_i}-bd(X_i,X)))\right| > \left|\bigcup_{i\in I_4} bd(X,X_i)\cup(Z\cap V(G_{X_i}))-\bigcup_{i\in I_1} bd(X,X_i)\right|$$

Since $G_{X_i}\cap G_{X_j}=\emptyset$ for $i,j\in I_4$ and $Z\cap V(G_{X_i})\geq OPT(G_{X_i})$, we have

$$\left|\bigcup_{i\in I_4} bd(X_i,X)\right| + \left|\bigcup_{i\in I_4} OPT(G_{X_i}-bd(X_i,X))\right| - \left|\bigcup_{i\in I_4} OPT(G_{X_i})\right|$$
$$> \left|\bigcup_{i\in I_4} bd(X,X_i)-\bigcup_{i\in I_1} bd(X,X_i)\right|$$

Note that by Eq. (2), $|bd(X_i,X)|+\left|OPT(G_{X_i}-bd(X_i,X))\right|-\left|OPT(G_{X_i})\right| = w(bd(X_i,X))$ and hence,

$$\sum_{i\in I_4} w(bd(X_i,X)) > \left|\bigcup_{i\in I_4} bd(X,X_i)-\bigcup_{i\in I_1} bd(X,X_i)\right| \tag{20}$$

Recall that D is a minimal minimum weighted vertex cover of $\mathcal{H}$. Observe that by definition of I_1 and $Sol(D)$, the set $\bigcup_{i\in I_1} bd(X,X_i)\subseteq D$. Now we show that if we delete the vertices in $\{bd(X_i,X): i\in I_4\}$ from D and adding the set of vertices $\bigcup_{i\in I_4} bd(X,X_i)-\bigcup_{i\in I_1} bd(X,X_i)$, then we get a vertex cover of smaller weight for $\mathcal{H}$ by inequality (20), a contradiction: By definition of I_1 and $Sol(D)$, the set $\bigcup_{i\in I_1} bd(X,X_i)\subseteq D$. Hence by the addition of the vertices $\bigcup_{i\in I_4} bd(X,X_i)-\bigcup_{i\in I_1} bd(X,X_i)$ to D we have the neighbourhood of each deleted vertex $bd(X_i,X)$ in D. □

Lemma 13. *$Sol(D)=\bigsqcup_{i=1}^4 D_i$ and for each distinct $i,j\subset[4]$, $Z_i\cap Z_j=\emptyset$.*

Proof. By Observation 3 it follows from the definition that for $1\leq i\neq j\leq 4$, the sets $D_i\cap D_j=\emptyset$ and $Z_i\cap Z_j=\emptyset$.

Now we show that $Sol(D)=D_1\cup D_2\cup D_3\cup D_4$. First consider the set $D_1\cup D_2=\bigcup_{i\in I_1\cup I_2}(bd(X,X_i)\cup(Sol(D)\cap G_{X_i}))$. By Lemma 8, $\bigcup_{i\in I_1\cup I_2} bd(X,X_i)=S_1(D)$ and $\bigcup_{i\in I_1\cup I_2}(Sol(D)\cap G_{X_i})=S_3(D)$. Hence $D_1\cup D_2=S_1(D)\cup S_3(D)$.

Now consider the set $D_3\cup D_4=\bigcup_{i\in I_3\cup I_4} bd(X_i,X)\cup OPT(G_{X_i}-bd(X_i,X))$. Hence by Lemma 8, $D_3\cup D_4=S_2(D)$. Therefore, the definition of $Sol(D)$ (Eq. (6)) implies $Sol(D)=\bigsqcup_{i=1}^4 D_i$. □

Proof of Theorem 5. Using Lemma 13, we have that $|Sol(D)|\leq|Z_1\cup Z_2\cup Z_3\cup Z_4|\leq|Z|$. Hence, $Sol(D)$ is a minimum X-CVD set of G. Furthermore, $\mathcal{H}$ has at most $|V(G)|$ vertices and $|E(G)|$ edges. Therefore minimum weighted vertex cover of $\mathcal{H}$ can be found in $O(|V(G)|\cdot|E(G)|)$-time [23,27] and $Sol(D)$ can be computed in total of $O(|V(G)|\cdot|E(G)|)$-time. □

Below we give a short pseudocode of our algorithm to find a minimum X-CVD set of G.

Algorithm 1: Pseudocode to find a minimum X-CVD set of a well-partitioned chordal graph

Input : A well-partitioned chordal graph G, a partition tree $\mathcal{T}$ of G rooted at the node X, for each node $Y\in\mathcal{T}-\{X\}$ both $OPT(G_Y)$ and $OPT(G_Y-bd(Y,P(Y)))$ are given as part of input
Output: A minimum X-CVD set

```
Construct a weighted bipartite graph H as described in Equations (1) and (2);
Find a minimum weighted vertex cover D of H;
Construct the sets S_1(D), S_2(D), S_3(D) and Sol(D) as described in Equations (3), (4), (5) and (6), respectively;
return Sol(D)
```

3.4. Finding minimum (X,Y)-CVD set of well-partitioned chordal graphs

In this section, we prove the following theorem.

Theorem 14. *Let G be a well-partitioned chordal graph; $\mathcal{T}$ be a partition tree of G rooted at X; Y be a child of X. Moreover, for each $Z\in V(\mathcal{T})-\{X\}$, assume both $OPT(G_Z)$ and $OPT(G_Z-bd(Z,P(Z)))$ are given ($P(Z)$ denotes the parent of Z in $\mathcal{T}$). Then a minimum (X,Y)-CVD set of G can be computed in $O\left(|E(G)|^2\cdot|V(G)|\right)$ time.*

For the remainder of this section, the meaning of G, $\mathcal{T}$, X and Y will be as given in Theorem 14. For an (X,Y)-edge e, we say that a minimum (X,Y)-CVD set A "preserves" the edge e if $G-A$ contains the edge e. Let $e\in E(X,Y)$ be an (X,Y)-edges of G. Then to prove Theorem 14, we use Theorem 15. First, we show how to construct a minimum (X,Y)-CVD set S_e that preserves the edge $e \in E(X, Y)$ and prove Theorem 15. Clearly, a minimum (X, Y)-CVD set S of G is the one that satisfies $|S| = \min_{e \in E(X,Y)} |S_e|$. Therefore, Theorem 14 will follow directly from Theorem 15. The remainder of this section is devoted to prove Theorem 15.

Theorem 15. *Assuming the same conditions as in Theorem 14, for $e \in E(X, Y)$, a minimum (X, Y)-CVD set of G that preserves e can be computed in $O(|E(G)| \cdot |V(G)|)$ time.*

First, we need the following observation about the partition trees of well-partitioned chordal graphs, which is easy to verify.

Observation 4. *Let G be a well-partitioned chordal graph with a partition tree $\mathcal{T}$. Let X, Y be two adjacent nodes of $\mathcal{T}$ such that $X \cup Y$ induces a complete subgraph in G and $\mathcal{T}'$ be the tree obtained by contracting the edge XY in $\mathcal{T}$. Now associate the newly created node with the subset of vertices $(X \cup Y)$ and retain all the other nodes of $\mathcal{T}$ and their associated subsets in $\mathcal{T}'$ as it is. Then $\mathcal{T}'$ is also a partition tree of G.*

We begin building the machinery to describe our algorithm for finding a minimum (X, Y)-CVD of G that preserves an (X, Y)-edge ab. Observe that any (X, Y)-CVD set that preserves the edge ab must contain the set $(N[a] \,\Delta\, N[b])$ as a subset. (Otherwise, the connected component of $G - S$ containing ab would not be a cluster, a contradiction).

Let H denote the graph $G - (N[a] \,\Delta\, N[b])$. Now consider the partition $\mathcal{Q}$ defined as $\{Z - (N[a] \,\Delta\, N[b]) : Z \in V(\mathcal{T})\}$. Now construct a graph $\mathcal{F}$ whose vertex set is $\mathcal{Q}$ and two vertices Z_1, Z_2 are adjacent in $\mathcal{F}$ if there is an edge $uv \in E(H)$ such that $u \in Z_1$ and $v \in Z_2$. Observe that $\mathcal{F}$ is a forest.

Now we have the following observation that relates the connected components of H with that of $\mathcal{F}$.

Observation 5. *There is a bijection f between the connected components of H and the connected components of $\mathcal{F}$, such that for a component C of H, $f(C)$ is the partition tree of C. Moreover, we can choose the root node of $f(C)$ as a subset of some node in $\mathcal{T}$.*

Proof. Recall that $\mathcal{Q}$ is a partition of $V(H)$, and the graph $\mathcal{F}$ is a forest. Let A be a connected component of H. We have the following cases.

1. A contains the vertices a and b. Observe that $\mathcal{Q}$ contains two sets $Z_1 = bd(X, Y)$ and $Z_2 = bd(Y, X)$. Hence, Z_1 and Z_2 are adjacent vertices in $\mathcal{F}$. Now define $f(A)$ to be the subgraph of $\mathcal{F}$ that contains Z_1 and Z_2. Clearly, $f(A)$ is a partition tree of A, and we can make the root node of $f(A)$ as $bd(X, Y)$ which is a subset of X, the root node of $\mathcal{T}$.
2. A contains a vertex v such that there exists an edge $e = uv$ with $u \in N[a] \setminus N[b]$ in G. In this case, observe that v must lie in some node Z of $\mathcal{T}$ and $Z - (N[a] \,\Delta\, N[b]) \in V(\mathcal{Q})$. Now define $f(A)$ to be the subgraph of $\mathcal{F}$ that contains $Z - (N[a] \,\Delta\, N[b])$. Clearly, $f(A)$ is a partition tree of A and make $Z - (N[a] \,\Delta\, N[b])$ as the root node of $f(A)$.
3. A contains a vertex v such that there exists an edge $e = uv$ with $u \in N[b] \setminus N[a]$ in G. In this case, observe that v must lie in some node Z of $\mathcal{T}$ and $Z - (N[a] \,\Delta\, N[b]) \in V(\mathcal{Q})$. Hence, $f(A)$ can be defined as in Case 2.

Observe that any connected component of H belongs to one of the above cases and each connected component of $\mathcal{F}$ contains atmost one node corresponding to a set $Z - (N[a] \,\Delta\, N[b])$. □

Consider the connected component H^* of H which contains a and b and let $\mathcal{F}' = f(H^*)$ where f is the function given by Observation 5. Observe that the root R' of $\mathcal{F}'$ is actually $bd(X, Y)$. Moreover, R' has a child R'', which is actually $bd(Y, X)$. Observe that $R' \cup R''$ induces a complete subgraph in H^*. Hence, due to Observation 4, the tree $\mathcal{F}^*$ obtained by contracting the edge $R'R''$ is a partition tree of H^*. Moreover, $R^* = R' \cup R'' = bd(X, Y) \cup bd(Y, X)$ is the root node of $\mathcal{F}^*$. Recall that our objective is to find a minimum (X, Y)-CVD set that preserves the edge ab. We have the following lemma.

Lemma 16. *Let $H^*, H_1, H_2, \dots, H_{k'}$ be the connected components of H. Let S^* be a minimum (R^*)-CVD set of H^*, $S_0 = (N[a] \,\Delta\, N[b])$, and for each $j \in [k']$, let S_j denote a minimum CVD set of H_j. Then $(S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_{k'} \cup S^*)$ is a minimum (X, Y)-CVD set of G that preserves the edge ab.*

Proof. Observe that any vertex adjacent to a or b lies in R^*. Since S^* is a minimum (R^*)-CVD set, $S^* \cap \{a, b\} = \emptyset$ and therefore $H^* - S^*$ has a cluster that contains the edge ab. Hence $S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_{k'} \cup S^*$ is an (X, Y)-CVD set that preserves the edge ab.

Let Z be any (X, Y)-CVD set of G that preserves the edge ab. For any vertex $u \in S_0$, observe that a, b, u induces a path of length 3. Hence, $S_0 \subseteq Z$. Let C be a connected component of $G - S_0$. Observe that $Z \cap V(C)$ must be a CVD set of C. Therefore, for each $i \in [k']$, $|Z \cap V(H_i)| \geq |S_i|$.

Since Z is an (X, Y)-CVD set of G that preserves the edge ab, $\{a, b\} \cap Z = \emptyset$. Since a, b are vertices of H^*, $(Z \cap V(H^*)) \cap \{a, b\} = \emptyset$. Now suppose $(Z \cap V(H^*))$ is not an (R^*)-CVD set of H^*. Then due to Lemma 4, $(Z \cap V(H^*))$ must be a (R^*, R)-CVD set of G^* for some child R of R^* in $\mathcal{T}^*$. Hence, there exists a (R^*, R)-edge cd which is preserved by $(Z \cap V(H^*))$. Without loss of generality, assume $c \in R^*$ and $d \in R$. Observe that d is adjacent to neither a nor b. Hence, a, c, d induce a path of length 3 in $H^* - (Z \cap V(H^*))$, a contradiction. Hence $|Z \cap V(H^*)| \geq |S^*|$. Therefore $|Z| \geq |S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_{k''} \cup S^*|$. □

Lemma 16 provides a way to compute a minimum (X, Y)-CVD set of G that preserves the edge ab. Clearly, the set $S_0 = (N[a] \mathbin{\Delta} N[b])$ can be computed in polynomial time. The following observation provides a way to compute a minimum CVD set of all connected components that are different from H^*.

Observation 6. *Let A be a connected component of H which is different from H^*. Then a minimum CVD set of A can be computed in polynomial time.*

Proof. Recall that A was obtained by deleting $(N[a] \mathbin{\Delta} N[b])$ from G, and that $\mathcal{T}$ is the partition tree of G where the root is X. Due to Observation 5, there is a function f between the connected components of H and the connected components of $\mathcal{F}$ such that $f(A)$ is the partition tree of A and there is a node $R \in \mathcal{T}$ such that the vertices in the root node of $f(A)$ are a subset of R. Now consider the following cases.

1. Consider the case when $\{a, b\} \cap bd(P(R), R) = \emptyset$. This implies no vertex of R is adjacent to a or b. Moreover, since A is different from H^*, $bd(P(R), R) \cap (bd(X, Y) \cup bd(Y, X)) = \emptyset$. This further implies that, either $bd(P(R), R) \subseteq N[a] - N[b]$ or $bd(P(R), R) \subseteq N[b] - N[a]$. In either case, $R \cap (N[a] \cup N[b]) = \emptyset$. This implies R is a node of $\mathcal{T}$ distinct from X such that A is isomorphic to G_R. Hence, due to the assumption given in Theorem 15, $OPT(G_R - bd(R, P(R)))$ is known and therefore a minimum CVD set of A is known.
2. Consider the case when there is a vertex $z \in \{a, b\}$ such that $z \in bd(P(R), R)$. Let z' be the vertex among a and b distinct from z. Since A is different from H^*, $z' \notin bd(P(R), R)$. Hence, $bd(R, P(R)) \subset N(z)$ and therefore $bd(R, P(R)) \subset (N[a] \mathbin{\Delta} N[b])$. This implies that R is a node of $\mathcal{T}$ distinct from X such that A is isomorphic to $G_R - bd(R, P(R))$. Hence, due to the assumption given in Theorem 15, $OPT(G_R - bd(R, P(R)))$ is known and therefore a minimum CVD set of A is known.

Clearly, distinguishing between the above cases takes $O(|E(G)|)$ time. This completes the proof. □

Let $H_1, H_2, \ldots, H_{k'}$ be the connected components of H, all different from H^*. Applying Observation 6 repeatedly on each component, it is possible to obtain, for each $j \in [k']$, a minimum CVD set S_j of H_j. The following observation provides a way to compute a minimum (R^*)-CVD set of H^*.

Observation 7. *Let R be a child of R^* in $\mathcal{F}^*$. Then both $OPT(H_R^*)$ and $OPT(H_R^* - bd(R, R^*))$ are known.*

Proof. Since no vertex of R is adjacent to a or b in G, there must exist a node $Q \in \mathcal{T}$ such that the vertices in the node Q are the same as that in R, $\mathcal{T}_Q = \mathcal{T}_R^*$ and $G_Q = H_R^*$. Moreover, $bd(R, R^*) = bd(Q, P(Q))$, where $P(Q)$ is the parent of Q in $\mathcal{T}$. Hence, due to the assumption given in Theorem 15, both $OPT(H_R^*)$ and $OPT(H_R^* - bd(R, R^*))$ are known. □

Due to Observation 7 and Theorem 5, it is possible to compute a minimum (R^*)-CVD set S^* of H^* in $O(|V(G)| \cdot |E(G)|)$ time. Now due to Lemma 16, we have that $(S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_{k'} \cup S^*)$ is a minimum (X, Y)-CVD set of G that preserves the edge ab. This completes the proof of Theorem 15 and therefore of Theorem 14. In Algorithm 2, we give a short pseudocode of our algorithm to find a minimum (X, Y)-CVD set of G that preserves an (X, Y)-edge ab. Using Algorithm 2, in Algorithm 3 we provide a short pseudocode to find a minimum (X, Y)-CVD set of G.

Algorithm 2: Pseudocode to find a minimum (X, Y)-CVD set of a well-partitioned chordal graph that preserves an (X, Y)-edge

Input : A well-partitioned chordal graph G, a partition tree $\mathcal{T}$ of G rooted at the node X, a child node Y, an (X, Y)-edge ab, for each node $Z \in \mathcal{T} - \{X\}$ both $OPT(G_Z)$ and $OPT(G_Z - bd(Z, P(Z)))$ are given as part of input

Output: A minimum (X, Y)-CVD set of G that preserves the edge ab

Construct the set $S_0 = (N[a] \mathbin{\Delta} N[b])$;
Construct the graph $H = G - (N[a] \mathbin{\Delta} N[b])$;
Let H^* be the connected component of H containing a and b. Let $H_1, H_2, \ldots, H_{k'}$ be the remaining connected components of H.
for $i = 1$ *to* k' **do**
  Compute a minimum CVD set S_i of H_i (Observation 6);
Find the partition tree of $\mathcal{T}^*$ of G^* whose root is $X^* = bd(X, Y) \cup bd(Y, X)$;
Compute a minimum (X^*)-CVD set S^* of G^* using Algorithm 1;
$Sol = S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_{k'} \cup S^*$;
return Sol;

Algorithm 3: Pseudocode to find a minimum (X, Y)-CVD set of a well-partitioned chordal graph.

Input : A well-partitioned chordal graph G, a partition tree $\mathcal{T}$ of G rooted at the node X, a child node Y, for each node $Z \in \mathcal{T} - \{X\}$ both $OPT(G_Z)$ and $OPT(G_Z - bd(Z, P(Z)))$ are given as part of input

Output: A minimum (X, Y)-CVD set of G.

For each (X, Y)-edge e, compute a minimum (X, Y)-CVD set that preserves the edge e using Algorithm 2;
Let S be a set among all S_e's that has the least cardinality;
return S;

3.5. Main algorithm

From now on, G denotes a fixed well-partitioned chordal graph with a partition tree $\mathcal{T}$ whose vertex set is $\mathcal{P}$, a partition of $V(G)$. We will process $\mathcal{T}$ in the post-order fashion, and for each node X of $\mathcal{T}$, we give a dynamic programming algorithm to compute both $OPT(G_X)$ and $OPT(G_X - bd(X, P(X)))$ where $P(X)$ is the parent of X (when exists) in $\mathcal{T}$. Due to Observation 4, we can assume that $bd(X, P(X)) \subsetneq X$. In the remaining section, X is a fixed node of $\mathcal{T}$, A has a fixed value (which is either $\emptyset$ or $bd(X, P(X))$), G_X^A denotes the graph $G_X - A$. Since well-partitioned chordal graphs are closed under vertex deletion, G_X^A is a well partitioned chordal graph which may be disconnected. Now consider the partition $\mathcal{P}^A$ defined as $\{Y - A : Y \in V(\mathcal{T}_X)\}$. Observe that, apart from the set X all other sets of the partitions $\mathcal{P}$ have remained in $\mathcal{P}^A$. Now construct a graph $\mathcal{T}'$ whose vertex set is the partition sets of $\mathcal{P}^A$ and two vertices X, Y are adjacent in $\mathcal{T}'$ if there is an edge $uv \in E(G_X^A)$ such that $u \in X$ and $v \in Y$ (since the graph induced by the union of the sets in $\mathcal{P}^A$ is G_X^A, the definition of $\mathcal{T}'$ is valid). Now we have the following observation whose proof is similar to that of Observation 5.

Observation 8. *There is a bijection f between the connected components of G_X^A and the connected components of $\mathcal{T}'$, such that for a component C of G_X^A, $f(C)$ is a partition tree of C, and the root of $f(C)$ is a child of X.*

Since the vertices of $X - A$ induce a clique in G_X^A, there exists at most one component G^* in G_X^A that contains a vertex from $X - A$. Due to Observation 5 there exists a unique connected component $f(G^*) = \mathcal{T}^*$ of $\mathcal{T}'$, which is a partition tree of G^*. Let the remaining connected components of G_X^A be $G_1, G_2, \ldots, G_k$ and for each $i \in [k]$, let $f(G_i) = \mathcal{T}_i$ and X_i is the root of $\mathcal{T}_i$. Let X^* denote the root node of $\mathcal{T}^*$ and $X_1^*, X_2^*, \ldots, X_t^*$ be the children of X^* in $\mathcal{T}^*$. We have the following observation.

Observation 9. *For each $j \in [t]$, there is a child Y_j of X in $\mathcal{T}$ such that $Y_j = X_j^*$ and $G_{Y_j} = G^*_{X_j^*}$.*

Proof. Observe that the root of $\mathcal{T}^*$ is $X^* = X - A$. Since $A \subsetneq X$, any child of X^* must be a child of X. □

We have the following lemma.

Lemma 17. $OPT(G_X^A) = \left(\bigsqcup_{i=1}^{k} OPT(G_{X_i}) \right) \sqcup OPT(G^*)$

Proof. The lemma follows directly from the fact that $G_{X_1}, G_{X_2}, \ldots, G_{X_k}$ and G^* are connected components of G_X^A. □

Due to Observation 8, $OPT(G_{X_i})$ is already known for each $i \in [k]$. Due to Lemma 4, any CVD set S of G^* is either a (X^*)-CVD set or there exists a unique child Y of X^*, such that S is a (X^*, Y)-CVD set of G^*. By Theorem 5, it is possible to compute a minimum (R^*)-CVD set S_0 of G^* in polynomial time. Due to Observation 9, for any node Y of $\mathcal{T}^*$ which is different from X^*, both $OPT(G_Y)$ and $OPT(G_Y - bd(Y, P(Y)))$ are known, where $P(Y)$ is the parent of Y in $\mathcal{T}^*$. Hence, by Theorem 14 for each child X_i^*, $i \in [t]$, computing a minimum (X^*, X_i^*)-CVD set S_i is possible in $O(|V(G^*_{X_i^*})| \cdot |E(G^*_{X_i^*})|^2)$ time. Let $S^* \in \{S_0, S_1, S_2, \ldots, S_t\}$ be a set with the minimum cardinality. Due to Lemma 4, S^* is a minimum CVD set of G^* that can be obtained in $O(m^2 n)$ time. Finally, due to Lemma 17, we have a minimum CVD set of G_X^A.

4. $O(n(n+m))$-Time algorithm for s-CVD on interval graphs

In this section, we shall give an $O(n(n+m))$-time algorithm to solve s-CVD on interval graph G with n vertices and m edges. For a set $X \subseteq V(G)$, if each connected component of $G - X$ is an s-club, then we call X as an *s-club vertex deleting set* (s-CVD set). In the next section, we present the main ideas of our algorithm to find a minimum cardinality s-CVD set of an interval graph.

4.1. Overview of the algorithm

The heart of our algorithm lies in a characterisation of s-CVD sets of an interval graph. We show (in Lemma 18) that any s-CVD set must be one of four types, defined in Definitions 9–12. Hence, the problem boils down to computing a minimum s-CVD set of each type. To do this, first we arrange the maximal cliques in the order of its *Helly region*. Let $Q_1, Q_2, \ldots, Q_k$ be the ordering of the cliques. Then for each $1 \le a \le k$, we find a minimum cardinality s-CVD set of the graph $G[1, a]$ which is the subgraph induced by the vertices in $(Q_1 \cup Q_2 \cup \cdots \cup Q_a)$. Moreover, to facilitate future computations we also find a minimum s-CVD set of the graph $G[1, a] - A$ where $A = Q_a \cap Q_b$ for some $a < b \le k$. The trick is to show that, by solving s-CVD on $O(n+m)$ many different "induced subgraphs" of G, it is possible to solve s-CVD on G. In other words, by solving $O(n+m)$ many different subproblems, it is possible to solve s-CVD on G. Moreover, it is possible to solve a subproblem in $O(n)$ time. In Section 4.2, we define four types of s-CVD sets and state that any optimal solution must be one of those four types. In Section 4.4, we give a sketch of our algorithm and analyse the time complexity in Section 4.5.

4.2. Definitions and main lemma

Let G denote a connected interval graph with n vertices and m edges. The set $\mathcal{I}$ denotes a fixed interval representation of G where the endpoints of the representing intervals are distinct. Let $l(v)$ and $r(v)$ denote the left and right endpoints, respectively, of an interval corresponding to a vertex $v \in V(G)$. Then the interval assigned to the vertex v in $\mathcal{I}$ is denoted by $I(v) = [l(v), r(v)]$.

Observe that, intervals on a real line satisfy the Helly property; hence, for each maximal clique Q of G there is an interval $I = \bigcap_{v\in Q} I(v)$. We call I as the *Helly* region corresponding to the maximal clique Q. Let $Q_1, Q_2, \ldots, Q_k$ denote the set of maximal cliques of G ordered with respect to their *Helly* regions I_a, $1 \le a \le k$ on the real line. That is, $I_1 < I_2 < \cdots < I_k$. Observe that, for any two integers a, b we have $I_a \cap I_b = \emptyset$ as both Q_a and Q_b are maximal cliques. Moreover, for any $a \le b \le c$ if a vertex $v \in Q_a \cap Q_c$, then $v \in Q_b$.

With respect to an ordering of maximal cliques $Q_1, Q_2, \ldots, Q_k$ of G, we define the following. In the following discussions, we consider Q_0 as an empty set.

Definition 6.

(i) For integers a,b where $1 \le a < b \le k$, let $S_a^b = Q_a \cap Q_b$.
(ii) For an integer a, let $\mathcal{S}(Q_a) = \left\{S_a^b : a < b \le k \text{ and } S_a^b \ne S_a^{b'}, a < b' < b\right\} \cup \emptyset$. (Note that, the members of the set $\mathcal{S}(Q_a)$ are distinct.)
(iii) For $A \in \mathcal{S}(Q_a)$, let $Y_A^a = (Q_a - Q_{a-1}) - A$.
(iv) For a vertex $v \in V(G)$, the index $q_v^- = \min\{a : v \in Q_a\}$. That is, the minimum integer a such that v belongs to the maximal clique Q_a.
(v) For a vertex $v \in V(G)$, the index $q_v^+ = \max\{a : v \in Q_a\}$. That is, the maximum integer a such that v belongs to the maximal clique Q_a.
(vi) For a vertex $v \in V(G)$, $N_{\text{left}}(v) = \{u : u \in N(v) \text{ and l(u) < l(v)}\}$.

We use the following observation to prove our main lemma.

Observation 10. *Let $X \subseteq V(G)$ and u, v be two vertices with $r(u) < l(v)$ such that u and v lie in different connected components in $G - X$. Then there exists an integer a with $q_u^+ \le a < q_v^-$, such that $S_a^{a+1} \subseteq X$.*

Proof. Let $\mathcal{C}$ be the set of all connected components of $G - X$. For a connected component $C \in \mathcal{C}$, define $\hat{r}(C) = \max\{r(v) : v \in C\}$ and $\hat{l}(C) = \min\{l(v) : v \in C\}$. Note that the interval $[\hat{l}(C), \hat{r}(C)] = \bigcup_{v\in V(C)} I(v)$ and we call it as the *span*(C). Observe that for two distinct connected components $C, C' \in \mathcal{C}$ we have *span*$(C) \cap$ *span*$(C') = \emptyset$. Therefore, $\mathcal{C}$ can be ordered with respect to the order in which the span of components appears on the real line. Let $C_1, \ldots, C_x$ be this ordering. We define *gap*$(C_i, C_{i+1}) = (\hat{r}(C_i), \hat{l}(C_{i+1}))$, $1 \le i \le x - 1$. Note that any vertex whose corresponding interval contains a point in *gap*(C_i, C_{i+1}) should be a member of X: otherwise that vertex belongs to another component in between C_i and C_{i+1} (by definition of *gap*(C_i, C_{i+1})) which contradicts the ordering of components. Let $C^u = C_t$ and $C^v = C_{t'}$ denote the connected components of $G - X$ that contain u and v, respectively. Since $r(u) < l(v)$, we have $t < t'$.

Let $p \in V(G)$ be such that $r(p) = \max\{r(w) : w \in V(G), r(w) < \hat{l}(C^v)\}$. Now take $a = q_p^+$, the maximum index i such that $p \in Q_i$, $1 \le i \le k$. For the index a, we will show that $q_u^+ \le a < q_v^-$ and $S_a^{a+1} \subseteq X$.

(i) $q_u^+ \le a < q_v^-$: It is immediate from the definition of $r(p)$ that $r(u) \le \hat{r}(C^u) \le r(p)$ and $r(p) < \hat{l}(C^v) \le l(v)$. Since $r(p) < l(v)$, observe that the *Helly* region corresponding to the clique containing the vertex p comes before that of v on the real line. Moreover, since the maximal cliques are numbered with respect to the order in which their *Helly* regions appear on the real line, we can infer that $q_p^+ = a < q_v^-$. Similarly, since $r(u) \le r(p)$, by similar arguments as above, we have $q_u^+ \le a$. Therefore we have proved $q_u^+ \le a < q_v^-$.

(ii) $S_a^{a+1} \subseteq X$: Consider the component $C_{t'-1}$ which comes in the immediate left of C^v in the ordering of the components in $\mathcal{C}$. Since $r(p) < \hat{l}(C^v)$, the *Helly* region of Q_a ends before *span*(C^v). Observe that $r(p) \ge \hat{r}(C_{t'-1})$. Moreover, the *Helly* region of Q_{a+1} starts after that of Q_a. Since $p \notin Q_{a+1}$ by definition of a it follows that *Helly* region of Q_{a+1} is after the *span*$(C_{t'-1})$. Therefore, the intervals corresponding to those vertices common to both Q_a and Q_{a+1} contain some points of *gap*$(C_{t'-1}, C^v)$. This implies $S_a^{a+1} \subseteq X$. □

For two integers a, b with $1 \le a \le b \le k$, let $G[a, b]$ denote the subgraph induced by the set $\{Q_a \cup Q_{a+1} \cup \cdots \cup Q_b\}$.

Definition 7. For an induced subgraph H of G, a vertex $v \in V(H)$ and an integer a, let $L_H(a, v)$ denote the set of vertices in H that lie at distance a from v in H.

In the remainder of this section, we use the notation $L_H(s + 1, v)$ where $H = G[1, a] - A$ for some integer a and $v \in Y_A^a$ (see Definition 6, (iii)) several times.

Definition 8. For an integer a, $1 \leq a \leq k-1$ and a set $A \in \mathcal{S}(Q_a)$ consider the induced subgraph $H = G[1, a] - A$ and the sub-interval representation $\mathcal{I}' \subseteq \mathcal{I}$ of H. We define the "frontal component" of the induced graph as the connected component of $G[1, a] - A$ containing the vertex with the rightmost endpoint in $\mathcal{I}'$.

Note that for an integer a and $A \in \mathcal{S}(Q_a)$, the vertices of Y_A^a, if any, lie in the *frontal* component of $G[1, a] - A$. Below we categorise an s-CVD set X of $G[1, a] - A$ into four types. In the following definitions, we consider an integer a, $1 < a \leq k$ and a set $A \in \mathcal{S}(Q_a)$.

Definition 9. An s-CVD set X of $G[1, a] - A$ is of "type-1" if $Y_A^a \subseteq X$.

Definition 10. An s-CVD set X of $H = G[1, a] - A$ is of "type-2" if there is a vertex $v \in Y_A^a$ such that $L_H(s+1, v) \subseteq X$.

Definition 11. An s-CVD set X of $H = G[1, a] - A$ is of "type-3" if there exists an integer c, $1 \leq c < a$ such that $S_c^{c+1} - A \subseteq X$ and $G[c+1, a] - (S_c^{c+1} \cup A)$ is connected and has diameter at most s.

Definition 12. An s-CVD set X of $H = G[1, a] - A$ is of "type-4" if there exists an integer c, $1 \leq c < a$ such that $S_c^{c+1} - A \subseteq X$ and $G[c+1, a] - (S_c^{c+1} \cup A)$ is connected and has diameter exactly $s+1$.

The following lemma is crucial for our algorithm.

Lemma 18 (*Main Lemma*)**.** *Consider an integer $1 \leq a \leq k$ and a set $A \in \mathcal{S}(Q_a)$. Then at least one of the following holds:*

1. *Every connected component of $G[1, a] - A$ has diameter at most s.*
2. *Any s-CVD set of $G[1, a] - A$ is of some type-j where $j \in \{1, 2, 3, 4\}$.*

Proof. Assume that the *frontal* component of $H = G[1, a] - A$ has diameter at least $s+1$ and the set $Y_A^a \neq \emptyset$. Otherwise, any s-CVD set X of H is of either type-1 or type-2: Type-1 is obvious when $Y_A^a = \emptyset$ because $\emptyset \subseteq X$. If the diameter of *frontal* component is at most s then the set $L_H(s+1, v) = \emptyset$ and hence any s-CVD set of H is of type-2.

Let H have an s-CVD set X that is not of type-j for any $j \in \{1, 2\}$ and v be a vertex in Y_A^a. Since X is not of type-2, $H - X$ contains a vertex u such that $u \in L_H(s+1, v)$. Now choose a vertex $u \in L_H(s+1, v)$ such that $q_u^+ = \max\{q_{u'}^+ : u' \in L_H(s+1, v) - X\}$.

Let $X' = X \cup A$. Then observe that $G[1, a] - X' = H - X$ and hence, $u \in V(G[1, a]) - X'$. Since X is an s-CVD set of H and the distance between u and v in H is $s+1$, the vertices u and v must lie in different connected components in $G[1, a] - X'$. Therefore, by Observation 10, there is an integer b such that $S_b^{b+1} \subseteq X'$ and $q_u^+ \leq b < q_v^-$. Let b be the maximum among all b' such that $q_u^+ \leq b' < q_v^-$ and $S_{b'}^{b'+1} \subseteq X'$. Note that $S_b^{b+1} \subseteq X'$ implies $S_b^{b+1} - A \subseteq X$. To complete the proof, we need the following claim.

Claim. Let Y be a subset of H such that $S_b^{b+1} \subseteq Y \subseteq X$ where b is the maximum among all b' such that $S_{b'}^{b'+1} \subseteq X$. Then $G[b+1, a] - (Y \cup A)$ is connected.

Proof of Claim: Suppose $G[b+1, a] - (Y \cup A)$ is not connected. Let $Z = Y \cup A$ and C_v be the connected component of $G[b+1, a] - Z$ containing a vertex $v \in Y_A^a$ (Note that $Y_A^a \neq \emptyset$). Since $G[b+1, a] - Z$ is not connected, there exists a vertex $u' \in G[b+1, a] - Z$ such that $u' \notin C_v$. Since u' and v are in different components and $v \in Y_A^a$, we have $q_{u'}^+ < q_v^- = a$. Note that $u' \in G - Z$ and $G - Z$ is also not connected. Then, by Observation 10, there exists an integer b^* such that $S_{b^*}^{b^*+1} \subseteq Z$ and $q_{u'}^+ \leq b^* < q_v^-$. Since $u' \in G[b+1, a] - Z$, the index $q_{u'}^+ > b$. Thus it follows that $b < q_u^+ \leq b^* < q_v^-$, which contradicts the maximality of the index b. ■

Let $H_b = G[b+1, a] - (S_b^{b+1} \cup A)$. Now we show that H_b has a diameter at most $s+1$. Otherwise, H_b contains vertices at a distance greater than $s+1$ from the vertex v. Let $Q_{b''}$ be the highest indexed maximal clique containing a vertex x such that distance between x and v in H_b is exactly $s+2$. By the definition of H_b, observe that $b'' > b$. Now we show that $S_{b''}^{b''+1} \subseteq X$ which contradicts the maximality of b (see the definition of b defined in the above paragraph).

For that, since $S_{b''}^{b''+1} \subseteq Q_{b''+1}$, the maximality of b'' implies that the vertices in $S_{b''}^{b''+1}$ are at distance $s+1$ from v in H_b. Note that by the above claim, the induced subgraphs H_b and $G[b+1, a] - (X \cup A)$ are connected. Moreover, since X is an s-CVD set of $H = G[1, a] - A$, when $S_b^{b+1} - A \subseteq X$ all vertices at distance greater than $s+1$ from the vertex v in H_b must be in $X - (S_b^{b+1} - A)$. Therefore, $S_{b''}^{b''+1} \subseteq X$ and $S_{b''}^{b''+1} \cup A \subseteq X'$. This contradicts the maximality of b. If the diameter of H_b is exactly $s+1$, then X is of type-4. Otherwise, X is of type-3. □

4.3. Some more observations

Let H be an induced subgraph of G, and u, v be two vertices of H. The distance between u and v in H is denoted by $d_H(u, v)$.

Observation 11. *Consider two integers a, b with $1 \leq a < b \leq k$ and a set $A \in \mathcal{S}(Q_b)$. Let $H = G[1, b] - A$ and u, v, w be three vertices of H such that $\{u, v\} \subseteq Q_b - Q_{b-1}$ and $w \in Q_a$. Then $d_H(u, w) = d_H(v, w)$.*

Proof. Suppose for contradiction that $d_H(u, w) \neq d_H(v, w)$. Without loss of generality assume that $d_H(u, w) < d_H(v, w)$. Let P be a shortest path between u and w in H and u' be the vertex in P which is adjacent to u. Observe that $u' \in Q_b \cap Q_{b-1}$ (this is because u is not intersecting with the *Helly* region of Q_{b-1}, $a < b$ in the ordering and P is a shortest path). Therefore u' is adjacent to v and $P' = (P - \{u\}) \cup \{v\}$ is a path between v and w such that $d_H(v, w) \leq |P'| = |P| = d_H(u, w)$, a contradiction. □

Observation 12. *Let C_f^H be the frontal component of $H = G[1, a] - A^*$, $A^* \subseteq V(G)$. Let $Y_{A^*}^a = (Q_a - Q_{a-1}) - A^*$. If $Y_{A^*}^a \neq \emptyset$, then any vertex $v \in Y_{A^*}^a$ is an end vertex of a diametral path (a shortest path whose length is equal to the diameter of a graph) of C_f^H.*

Proof. Suppose that $Y_{A^*}^a \neq \emptyset$ and a vertex $v \in Y_{A^*}^a$ is not an end vertex of a diametral path of C_f^H. Let P be a diametral path of C_f^H and x, y be the end vertices. Observe that neither x nor y is in $Y_{A^*}^a$. Without loss of generality, assume that $q_x^- \leq q_y^-$. Let P' be a shortest path between x and v where $v \in Y_{A^*}^a$. Since P have the maximum size among the shortest paths and P' is not a diametral path, we have $|P'| < |P|$. Since $v \in Y_{A^*}^a$ and $x, y \notin Y_{A^*}^a$ we have $a = q_v^- > q_y^- \geq q_x^-$. Hence the path P' contains a vertex w such that $w \neq v$ and $q_w^- \leq q_y^- \leq q_w^+$ (That is, any path from v to x should cross the cliques containing y). This implies w is a neighbour of y and there exists a path P'' between x and y via w such that $|P''| \leq |P'|$ (the path P'' is obtained by adding the edge wy to the subpath from x to w in P'). Since $|P'| < |P|$, this contradicts the assumption that P is a shortest path between x and y. Therefore, there exists at least one vertex $v \in Y_{A^*}^a$, which is an end vertex of a diametral path of C_f^H. Then by Observation 11, each vertex in $Y_{A^*}^a$ is an end vertex of a diametral path of C_f^H. □

4.4. The algorithm

Our algorithm constructs a table Ψ iteratively whose cells are indexed by two parameters. For an integer a, $1 \leq a \leq k$ and a set $A \in \mathcal{S}(Q_a)$, the cell $\Psi[a, A]$ contains a minimum s-CVD set of $G[1, a] - A$. Clearly, $\Psi[k, \emptyset]$ contains a minimum s-CVD set of G.

Now we start the construction of Ψ. Since $G[1, 1]$ is a clique, we set $\Psi[1, A] = \emptyset$ for all $A \in \mathcal{S}(Q_1)$.

Observation 13. *For any $A \in \mathcal{S}(Q_1)$, $\Psi[1, A] = \emptyset$.*

From now on assume $a \geq 2$ and A be a set in $\mathcal{S}(Q_a)$. Let H be the graph $G[1, a] - A$ and F be the graph $G[1, a-1] - (A \cap Q_{a-1})$. Observe that for any two integers a, b, $1 \leq a < b \leq k$ the set $S_{a-1}^b = S_a^b \cap Q_{a-1}$. Then, for any $A \in \mathcal{S}(Q_a)$ we have $(A \cap Q_{a-1}) \in \mathcal{S}(Q_{a-1})$ and $\Psi[a-1, A \cap Q_{a-1}]$ is defined. Note that $H - F = Y_A^a$.

In the following lemma we show that $\Psi[a, A] = \Psi[a-1, A \cap Q_{a-1}]$ if the *frontal* component of H has diameter at most s.

Lemma 19. *Let $H = G[1, a] - A$, for $A \in \mathcal{S}(Q_a)$, $1 < a \leq k$. If the* frontal *component of H has diameter at most s, then $\Psi[a, A] = \Psi[a-1, A \cap Q_{a-1}]$.*

Proof. Let F denote the graph $G[1, a-1] - (A \cap Q_{a-1})$. Since $H - F = Y_A^a$, if $Y_A^a = \emptyset$ then $H = F$ and hence, $\Psi[a, A] = \Psi[a-1, A \cap Q_{a-1}]$. Now assume that $Y_A^a \neq \emptyset$. Observe that the connected components of H and F are the same except the *frontal* components. The *frontal* components of H and F differ depending on the set S_{a-1}^a as follows.

(i) If $S_{a-1}^a \cap H = \emptyset$ then the *frontal* component of H is Y_A^a.
(ii) If $S_{a-1}^a \cap H \neq \emptyset$ then the *frontal* component of H is the union of the *frontal* component of $G[1, a-1] - A$ and Y_A^a.

If the *frontal* component of H is Y_A^a then $\Psi[a, A] = \Psi[a-1, A \cap Q_{a-1}]$ because diameter of Y_A^a is 1. Hence assume that the *frontal* component of H belongs to the case (ii) defined above. Let C_f^H be the *frontal* component of H and C_f^F be the *frontal* component of F. Then $C_f^H = C_f^F \cup Y_A^a$. We have the following claim.

Claim. Let $C_f^H = C_f^F \cup Y_A^a$. If the diameter of C_f^H is at most s, then the diameter of C_f^F is also at most s.

Proof of Claim: Suppose not, then C_f^F contains two vertices u and v such that the distance between u and v in C_f^F is at least $s+1$. Without loss of generality, assume that $l(u) < l(v)$. Let P be a shortest path between u and v in C_f^F. Observe that since $C_f^F = C_f^H - Y_A^a$, no vertex $w \in Y_A^a$ belongs to $V(P)$. Moreover, for any vertex $w \in Y_A^a$ we have $l(u) < l(v) < l(w)$ in the interval representation. Therefore, any shortest path between u and v in C_f^H does not contain a vertex $w \in Y_A^a$. Hence the shortest path between u and v in C_f^H is also at least $s+1$, which contradicts the assumption that the diameter of C_f^H is at most s. ■

Hence by the minimality of $\Psi[a-1, A \cap Q_{a-1}]$, no vertices of C_f^F are in $\Psi[a-1, A \cap Q_{a-1}]$. Thus it follows that $\Psi[a, A] = \Psi[a-1, A \cap Q_{a-1}]$. □

Now assume that the *frontal* component of $H = G[1, a] - A$ has diameter at least $s+1$. Recall that if $Y_A^a = \emptyset$, we have $\Psi[a, A] = \Psi[a-1, A \cap Q_{a-1}]$. Hence assume that $Y_A^a \neq \emptyset$. Due to Lemma 18, any s-CVD set of H has to be one of the four types defined in Section 4.2.

First, for each $j \in \{1, 2, 3, 4\}$, we find an s-CVD set of minimum cardinality, which is of type-j. We begin by showing how to construct a minimum cardinality s-CVD set X_1 of type-1 of $G[1, a] - A$. We define X_1 as below.

$$X_1 = Y_A^a \cup \Psi[a-1, A \cap Q_{a-1}] \tag{21}$$

Lemma 20. *The set X_1 is a minimum cardinality s-CVD set of type-1 of $G[1, a] - A$.*

Proof. Observe that the graph $H = G[1, a] - Y_A^a$ is isomorphic to $G[1, a-1] - (A \cap Q_{a-1})$. Hence $X_1 = Y_A^a \cup \Psi[a-1, A \cap Q_{a-1}]$ is an s-CVD set of H. By definition, Y_A^a is included in an s-CVD set of type-1. Hence the minimality of $\Psi[a-1, A \cap Q_{a-1}]$ implies that X_1 is a minimum cardinality set of type-1. □

Let v be some vertex in Y_A^a and $b < a$ be the maximum integer such that $(Q_b \cap L_H(s+2, v)) \neq \emptyset$. We construct a minimum cardinality s-CVD set of type-2 of $G[1, a] - A$ defined as follows.

$$X_2 = L_H(s+1, v) \cup \Psi[b, S_b^{b+1}] \tag{22}$$

Lemma 21. *The set X_2 is a minimum cardinality s-CVD set of type-2 of $G[1, a] - A$.*

Proof. By the maximality of b we have $S_b^{b+1} \subseteq L_H(s+1, v)$. Moreover, the graph $(G[b+1, a] - A) - L_H(s+1, v)$ is connected: otherwise, if $L_H(s+1, v)$ is a separator of $(G[b+1, a] - A)$ then $(S_{b'}^{b'+1} - A) \subseteq L_H(s+1, v)$ for some $b' > b$. Since $Q_{b'}$ is a maximal clique, there exists at least one vertex $w \in Q_{b'}$ and $w \notin Q_{b'+1}$. Hence the distance between w and v is $s+2$ and $(Q_{b'} \cap L_H(s+2, v)) \neq \emptyset$. Since $b' > b$, this contradicts the maximality of b.

Since $(G[b+1, a] - A) - L_H(s+1, v)$ is connected we have $((G[b+1, a] - A) - L_H(s+1, v))$ is a frontal component of $G[1, a] - (A \cup L_H(s+1, v))$. Let $A' = A \cup L_H(s+1, v)$. Note that $Y_{A'}^a = Y_A^a \neq \emptyset$. Observe that the distance between $v \in Y_{A'}^a$ and any other vertex in $(G[b+1, a] - A) - L_H(s+1, v)$ is at most s. Hence by Observation 12, $(G[b+1, a] - A) - L_H(s+1, v)$ has diameter at most s.

Note that any vertex of $G[1, b]$ that belongs to A is also in S_b^{b+1}. Hence $G[1, b] - (A \cup S_b^{b+1}) = G[1, b] - S_b^{b+1}$. Since $\Psi[b, S_b^{b+1}]$ is a minimum cardinality s-CVD set of $G[1, b] - S_b^{b+1}$ the set $X_2 = L_H(s+1, v) \cup \Psi[b, S_b^{b+1}]$ is an s-CVD set of H. By definition, $L_H(s+1, v)$ is included in an s-CVD set of type-2. Observe that any vertex of $G[1, b]$ that belongs to $L_H(s+1, v)$ is also in S_b^{b+1} and hence the minimality of $\Psi[b, S_b^{b+1}]$ implies that X_2 is a minimum cardinality set of type-2. □

Now we show how to construct a minimum cardinality s-CVD set X_3 of type-3 of $G[1, a] - A$. Let $B \subseteq \{1, 2, \ldots, a-1\}$ be the set of integers such that for any $i \in B$ the graph $H_i = G[i+1, a] - (S_i^{i+1} \cup A)$ is connected and has a diameter at most s. By definition, a type-3 s-CVD set X of H contains S_c^{c+1} for some $c \in B$. We call each such type-3 s-CVD set as type-3(c). Now we define minimum type-3(c) s-CVD set as follows.

$$\text{For each } c \in B, \quad Z_c = (S_c^{c+1} - A) \cup \Psi[c, S_c^{c+1}] \tag{23}$$

Claim. The set Z_c is a minimum cardinality s-CVD set of type-3(c) of $G[1, a] - A$.

Proof of Claim: Note that any vertex of $G[1, c]$ that belongs to A is also in S_c^{c+1}. By definition, S_c^{c+1} separates the connected component $G[c+1, a] - (S_c^{c+1} \cup A)$ from the rest of the graph namely, $G[1, c] - (S_c^{c+1})$. Since the diameter of $G[c+1, a] - (S_c^{c+1} \cup A)$ is at most s and $\Psi[c, S_c^{c+1}]$ is the minimal cardinality s-CVD set of $G[1, c] - S_c^{c+1}$ the set $Z_c = (S_c^{c+1} - A) \cup \Psi[c, S_c^{c+1}]$ is a minimum cardinality s-CVD set of H of type-3(c). ■

We define X_3 as below.

$$X_3 = \min\{Z_c : c \in B\} \tag{24}$$

Lemma 22. *The set X_3 is a minimum cardinality s-CVD set of type-3 of $G[1, a] - A$.*

Proof. The minimality of each Z_c implies that the set X_3 is a minimum cardinality type-3 s-CVD set. □

Finally, we show the construction of a minimum cardinality s-CVD set X_4 of type-4 of $G[1, a] - A$. Let $C \subseteq \{1, 2, \ldots, a-1\}$ be the set of integers such that for any $i \in C$ the graph $H_i = G[i+1, a] - (S_i^{i+1} \cup A)$ is connected and has diameter exactly $s+1$. By definition, a type-4 s-CVD set X of H contains S_i^{i+1} for some $i \in C$. We call each such type-4 s-CVD set as type-4(c). Now we define minimum type-4(c) s-CVD set as follows. Note that $Y_A^a \neq \emptyset$. Let v be some vertex in Y_A^a and $Y_i = L_{H_i}(s+1, v)$.

$$\text{For each } i \in C, \quad Z_i = (S_i^{i+1} - A) \cup Y_i \cup \Psi[i, S_i^{i+1}] \tag{25}$$

Claim. The set Z_i is a minimum cardinality s-CVD set of type-4(c) of $G[1, a] - A$.

Proof of Claim: Recall that H_i is connected and we claim that the graph $H_i - Y_i$ is also connected: otherwise, if Y_i is a separator of H_i then there exists a vertex w in $H_i - Y_i$ such that w does not belong to the component containing v in $H_i - Y_i$. Since any path from v to w in H_i passes through Y_i, the distance of w from v in H_i is at least $s + 2$ contradicting the assumption that H_i has diameter exactly $s + 1$.

Since $H_i - Y_i$ is connected, it is the *frontal* component of $G[1, a] - A - (S_i^{i+1} \cup Y_i)$. Let $A' = A \cup S_i^{i+1} \cup Y_i$. Note that $Y_{A'}^a = Y_A^a \neq \emptyset$. Hence the distance between $v \in Y_{A'}^a$ and any other vertex in $H_i - Y_i$ is at most s. Thus by Observation 12 the graph $H_i - Y_i$ has diameter at most s. Note that $\Psi[i, S_i^{i+1}]$ is the minimal cardinality s-CVD set of $G[1, i] - S_i^{i+1}$ and any vertex of $G[1, i] - S_i^{i+1}$ that belongs to Y_i or A is also in S_i^{i+1}. Hence, the set $Z_i = (S_i^{i+1} - A) \cup Y_i \cup \Psi[i, S_i^{i+1}]$ is a minimum s-CVD set of H of type-4(c). ■

Now define X_4 as follows.

$$X_4 = \min\{Z_i : i \in C\} \tag{26}$$

Lemma 23. *The set X_4 is a minimum cardinality s-CVD set of type-4 of $G[1, a] - A$.*

Proof. The minimality of each Z_i implies that the set X_4 is a minimum cardinality type-4 s-CVD set. □

Now we define a minimum s-CVD set of $G[1, a] - A$ as the one with minimum cardinality among the sets X_i, $1 \leq i \leq 4$. That is,

$$\Psi[a, A] = \min\{X_1, X_2, X_3, X_4\} \tag{27}$$

A pseudocode of the procedure to find Eq. (27) is given by Procedure 4.

Procedure 4: Compute_sCD(G, a, A)

Let $H = G[1, a] - A$ and $Y_A^a = (Q_a - Q_{a-1}) - A$
Set $X_1 = Y_A^a \cup \Psi[a - 1, A \cap Q_{a-1}]$
For a vertex $v \in Y_A^a$, find the maximum integer b such that $b < a$ and $Q_b \cap L_H(s + 2, v) \neq \emptyset$
Set $X_2 = L_H(s + 1, v) \cup \Psi[b, S_b^{b+1}]$
Set $B = C = \emptyset$
for $c = 1$ *to* $a - 1$ **do**
  if Diam[c][a][A] $\leq s$ **then**
    $Z_c = (S_c^{c+1} - A) \cup \Psi[c, S_c^{c+1}]$ ▷ $H_c = G[c + 1, a] - (S_c^{c+1} \cup A)$
    $B = B \cup \{c\}$
  if Diam[c][a][A] $= s + 1$ **then**
    $W_c = (S_c^{c+1} - A) \cup L_{H_c}(s + 1, v) \cup \Psi[c, S_c^{c+1}]$
    $C = C \cup \{c\}$
Set $X_3 = \min\{Z_i : i \in B\}$
Set $X_4 = \min\{W_i : i \in C\}$
Set $\Psi[a, A] = \min\{X_1, X_2, X_3, X_4\}$
Return $\Psi[a, A]$

We formally summarise the above discussion in the following lemma.

Lemma 24. *For $1 < a \leq k$, if the diameter of the* frontal *component of $G[1, a] - A$ is at least $s + 1$, then $\Psi[a, A] = \min\{X_1, X_2, X_3, X_4\}$.*

Proof. The proof follows from Lemma 18 and the above discussion on the minimality of the sets X_i, $1 \leq i \leq 4$, in their respective types. □

The proof of correctness of the algorithm follows from the Lemmas 13, 19, and 24. A pseudocode of the algorithm for finding a minimum s-CVD set of an interval graph is given in Algorithm 5. In the following section, we discuss the time complexity of the algorithm.

Algorithm 5: s-CVD(G, s):G is an interval graph and s is a positive integer

Input : An interval graph G and a positive integer s
Output: $\Psi[k, \emptyset]$

1 Using algorithm in [7] find the ordered set of maximal cliques of G, say $Q_1, Q_2, \dots, Q_k$ and $N_{\text{left}}(v)$, q_v^+ and q_v^- for each vertex $v \in V(G)$
2 Find $\mathcal{S}(Q_1)$
3 **for** *all* $A \in \mathcal{S}(Q_1)$ **do**
4     $\Psi[1, A] = \emptyset$
5 **for** $a = 2$ *to* k **do**
6   Find $\mathcal{S}(Q_a)$
7   **for** $A \in \mathcal{S}(Q_a)$ **do**
8   Set $Y_A^a = (Q_a - Q_{a-1}) - A$
9   **if** $Y_A^a = \emptyset$ **then**
10             $\Psi[a, A] = \Psi[a - 1, A \cap Q_{a-1}]$
11   **else**
12   **for** $c = 1$ *to* $a - 1$ **do**
13   Find the diameter of the induced subgraph $H_c = G[c + 1, a] - (A \cup S_c^{c+1})$ using $N_{\text{left}}(v)$, $v \in Y_A^a$ and store it in Diam$[c][a][A]$.
14   **if** *diameter* Diam$[1][a][A]$ *of the* frontal *component of* $H_0 = G[1, a] - A \leq s$ **then**
15                 $\Psi[a, A] = \Psi[a - 1, A \cap Q_{a-1}]$
16   **else**
17                 $\Psi[a, A] =$ `Compute_sCD(G,a,A)`
18 **return**

4.5. Time complexity

For a given interval graph G with n vertices and m edges, the algorithm first finds the ordered set of maximal cliques of G as described in Section 4.2. Such an ordered list of the maximal cliques of G can be produced in linear time as a byproduct of the linear ($O(n + m)$)-time recognition algorithm for interval graphs due to Booth and Leuker [7]. For each vertex $v \in G$, the algorithm gathers the following information during the enumeration of maximal cliques: (i) the values q_v^- and q_v^+ and (ii) $N_{\text{left}}(v)$ and are ordered with respect to the left endpoints.

Let $Q_1, Q_2, \dots, Q_k$ be the ordered set of maximal cliques of G. From the ordered set of cliques, the algorithm constructs the set $\mathcal{S}(Q_a)$ (**steps 2, 6, Algorithm 5**) for each Q_a, $1 \leq a < k$. For an integer a, $1 \leq a < k$ the set $\mathcal{S}(Q_a)$ can be constructed by adding a vertex $v \in Q_a$ to each $S_a^b \in \mathcal{S}(Q_a)$ for $a < b \leq q_v^+$. For the computation of each $\Psi[a, A]$, $1 \leq a \leq k$, $A \in \mathcal{S}(Q_a)$ the algorithm needs to compute the following: (i) the set of vertices, Y_A^a (**step 8, Algorithm 5**); (ii) the diameter of the frontal component of the graph $H = G[1, a] - A$ (**step 14, Algorithm 5**) and (iii) the diameter of the induced subgraphs $H_c = G[c + 1, a] - (A \cup S_c^{c+1})$, $1 \leq c \leq a - 1$ (**steps 12–13, Procedure 4**).

The set Y_A^a can be obtained from the vertex set of Q_a in linear time by checking the q_v^- and q_v^+ values of each vertex $v \in Q_a$. That is, $Y_A^a = \{v \in Q_a : q_v^- = a \text{ and } q_v^+ < b, A = S_a^b\}$. Let Diam$[1][a][A]$ be the diameter of the frontal component of $H = G[1, a] - A$. By Observation 12, the diameter of the frontal component of H is equal to the eccentricity of a vertex $v \in Y_a^A$. That is, the maximum distance of v from other vertices in H which we denote by $ecc_H(v)$. Hence, Diam$[1][a][A]$ $= ecc_H(v)$. Let v_l be the *leftmost* neighbour of v in H such that $q_{v_l}^- = a'$ and $ecc_{H'}(v_l)$ be the eccentricity of v_l in $H' = G[1, a'] - (Q_{a'} \cap Q_b)$. Then observe that $ecc_H(v) = ecc_{H'}(v_l) + 1$. Therefore, Diam$[1][a][A] =$ Diam$[1][a'][Q_{a'} \cap Q_b] + 1$. Since the *leftmost* neighbour of v in H can be found in linear time from $N_{\text{left}}(v)$ by checking the q_v^- and q_v^+ values of each vertex $u \in N_{\text{left}}(v)$, the diameter of the frontal component of H can be found in $O(n)$ time. Similarly, diameter of the induced subgraphs $H_c = G[c + 1, a] - (A \cup S_c^{c+1})$ in **steps 12–13, Procedure 4** together can be found in $O(n)$ time by similar arguments as above and the following observation; $N_{\text{left}}(v) - (A \cup S_c^{c+1}) \supseteq N_{\text{left}}(v) - (A \cup S_{c+1}^{c+2})$.

To compute the overall time complexity of our algorithm, we have the following claims.

Claim 2. *Total number of subproblems computed by the algorithm, Algorithm 5 is at most $O(|V| + |E|) = O(n + m)$.*

Proof of Claim: Note that with respect to the ordering of maximal cliques of G, the elements of the set $\mathcal{S}(Q_a)$ have the following relation. For each $b, a < b \leq k$ we have $S_a^{b+1} \subseteq S_a^b$. Hence the number of distinct subproblems computed by the algorithm corresponding to each maximal clique Q_a is at most $|S_a^{a+1}| + 1$ (recall that one of the subproblems corresponds to $\emptyset \in \mathcal{S}(Q_a)$). Since the number of maximal cliques in G is at most $|V| = n$ and $|S_a^{a+1}| \leq degree(v)$, $v \in Q_a - Q_{a+1}$, the total number of subproblems computed by the algorithm is at most $\sum_{v \in Q_a - Q_{a+1}} degree(v) + |V| \leq O(|V| + |E|) = O(n + m)$. ■

Claim 3. *The procedure* Compute_sCD(G, a, A) *computes the minimum cardinality s-CVD set of $H = G[1, a] - A$ in $O(n)$ time.*

Proof of Claim: Observe that the time complexity of the procedure *Compute_sCD*(G, a, A) depends mainly on building the sets X_i, $1 \leq i \leq 4$. Since the set Y^a_A, $1 \leq a < k, A \in \mathcal{S}(Q_a)$ is obtained in $O(n)$ time, the set X_1 can be computed in $O(n)$ time.

The set $L_H(s+1, v)$ can be computed from the leftmost neighbour of v in H, say v_l in linear time by s iterations: In the first iteration, find the leftmost vertex of v_l in $N_{\text{left}}(v_l) - A$, in the second iteration, find the leftmost vertex in the second neighbourhood and so on. Moreover, the leftmost neighbour of v in H can be obtained by a linear search of $N_{\text{left}}(v)$.

Since the number of induced subgraphs H_c is at most $O(n)$, the sets X_3 and X_4 can be constructed in $O(n)$ time. Hence the claim follows. ■

Therefore, by the above claims, the overall time complexity of our algorithm is $O(n \cdot (n+m))$, and Theorem 3 follows.

5. Hardness for well-partitioned chordal graphs

In this section, we prove Theorem 2. We shall use the following observation.

Observation 14. *Let H be a well-partitioned chordal graph. Let H' be a graph obtained from H by adding a vertex of degree 1. Then H' is a well-partitioned chordal graph.*

Let $s \geq 2$ be an even integer and let $s = 2t$. We shall reduce Minimum Vertex Cover (MVC) on general graphs to s-CVD on well-partitioned chordal graphs. Let $\langle G, k \rangle$ be an instance of Minimum Vertex Cover such that the maximum degree of G is at most $n-3$. Let $\overline{G}$ denote the complement of G. Now construct a well-partitioned chordal graph G_{well} from G as follows. For each vertex of $v \in V(G)$, we introduce a new path P_v with $t-1$ edges and let x_v, x'_v be the endpoints of P_v. For each edge $e \in E(\overline{G})$ we introduce a new vertex y_e in G_{well}. For each pair of edges $e_1, e_2 \in E(\overline{G})$ we introduce an edge between y_{e_1} and y_{e_2} in G_{well}. For each edge $e = uv \in E(\overline{G})$, we introduce the edges $x_u y_e$ and $x_v y_e$ in G_{well}. Observe that $C = \{y_e\}_{e \in E(\overline{G})}$ is a clique, $I = \{x_v\}_{v \in V(G)}$ is an independent set of G_{well}. Therefore $C \cup I$ induces a split graph, say G', in G_{well}. Moreover, each vertex of I has at least two neighbours in C. (This is due to the fact that minimum degree of $\overline{G}$ is at least two.) Since G_{well} can be obtained from G' by adding vertices of degree 1, due to Observation 14, we have that G_{well} is a well-partitioned chordal graph. We shall show that G has a vertex cover of size k if and only if G_{well} has a s-CVD set of size k.

Observation 15. *For each vertex $v \in C$, $|N(v) \cap I| = 2$ and for each vertex $u \in I$, $|N(u) \cap C| \geq 2$.*

Lemma 25. *Let D be a subset of I and let $T = \{u \in V(G) : x_u \in D\}$. The set D is an s-CVD set of G_{well} if and only if T is a vertex cover of G.*

Proof. Let $D' = \{x'_v : x_v \in I - D\}$ and $T' = \{u \in V(G) : x_u \in D'\}$ (note that $T = V(G) - T'$). Note that there is one single component G' of $G_{well} - D$ that contains vertices from C since there are no isolated vertices by Observation 15. Observe that G' contains $I - D$. Therefore, for any two vertices $x'_u, x'_v \in D'$, the distance between x'_u, x'_v is s if and only if there is an edge between u, v in $\overline{G}$. Therefore, the distance between any two pairs of vertices in D' is s if and only if T' induces a clique in $\overline{G}$ and, therefore, an independent set in G. Since $T = V(G) - T'$, we have that distance between any two pairs of vertices in D' is s if and only if T is a vertex cover of G. Since $|D'| = |I - D|$, we have that D is an s-CVD set of G_{well} if and only if T is a vertex cover of G. □

Lemma 26. *There is a subset of I, which is a minimum s-CVD set of G_{well}.*

Proof. Let S be a minimum s-CVD set of G_{well} such that $|S \cap I|$ is maximum. We claim that $S \subseteq I$. Suppose for contradiction this is not true. Let $I' = \bigcup_{u \in V(G)} V(P_u) - \{x_u\}$. Then we must have that $S \cap I' \neq \emptyset$ or $S \cap C \neq \emptyset$. Let a be a vertex of $S \cap I'$. Observe that there must be a vertex $u \in V(G)$ such that $a \in V(P_u)$ and that $(S - \{a\}) \cup \{x_u\}$ is an s-CVD set of G_{well}. This contradicts the assumption that S is a minimum s-CVD set of G_{well} with $|S \cap I|$ maximum.

Now consider the collection $\mathcal{C}$ of connected components of $G_{well} - S$. First, observe that there exists at most one connected component in $\mathcal{C}$ that intersects C (the clique of G_{well}). We shall call such a component as the *big component* and let X be the set of vertices of the big component. Note that, by Observation 15, the total number of edges incident on the vertices of I in the induced subgraph $I \cup C$ of G_{well} is atleast $2 \cdot |I|$ and on the vertices of C in the induced subgraph $I \cup C$ is exactly $2 \cdot |C|$ which implies $|I| \leq |C|$. In fact I itself is a s-CVD set. Therefore, without loss of generality, we can assume that $C \not\subseteq S$ and, indeed such a big component exists.

Let Y denote those vertices of $G_{well} - S$ that belong to $I - X$. Let $S_C = S \cap C$ and $S_I = S \cap I$. Recall that by assumption, $S_C \neq \emptyset$.

If there is a vertex $v \in S_C$ such that $|N[v] \cap Y| = 0$, then $S - \{v\}$ is a s-CVD set with $X \cup \{v\}$ as corresponding big component with diameter less than or equal to s. This contradicts the minimality of S. Similarly, if there exists a vertex $v \in S_C$ such that $N[v] \cap Y = \{u\}$, a singleton set then $S' = S \cup \{u\} - \{v\}$ is a new s-CVD set with $X \cup \{v\}$ as corresponding new big component. This contradicts the assumption that S is a minimum s-CVD set with $|S \cap I|$ is maximum. Hence together with Observation 15, we infer that $|N(v) \cap Y| = 2$, for each $v \in S_C$. Note that, by the definition of Y, all the neighbours of Y in C belong to S_C. Hence Observation 15 implies that for each vertex $u \in Y$, $|N(u) \cap S_C| \geq 2$. Therefore, the total number of edges incident on the vertices of Y in the induced subgraph $Y \cup S_C$ is atleast $2 \cdot |Y|$ and on the vertices of S_C in $Y \cup S_C$ is exactly $2 \cdot |S_C|$. This implies $|Y| \leq |S_C|$. Hence, $S' = (S - S_C) \cup Y$ is a minimum 2-CVD set with $X \cup S_C$ as the corresponding new big component and $|S' \cap I| > |S \cap I|$. This contradicts the assumption for S.

Hence we conclude that S is indeed a minimum s-CVD set such that $S \subseteq I$. □

Lemmas 25 and 26 imply that G has a vertex cover of size k if and only if G_{well} has a s-CVD set of size k. Now Theorem 2 follows from a result of Khot and Regev [21], where they showed that unless the Unique Games Conjecture is false, there is no $(2 - \epsilon)$-approximation algorithm for Minimum Vertex Cover on general graphs, for any $\epsilon > 0$.

6. Conclusion

In this paper, we studied the computational complexity of s-CVD on well-partitioned chordal graphs, a subclass of chordal graphs that generalises split graphs [18]. We gave a polynomial-time algorithm for $s = 1$, and we proved that for any even integer $s \geq 2$, s-CVD is NP-hard on well-partitioned chordal graphs. We also provide a faster algorithm for s-CVD on interval graphs for each $s \geq 1$. This raises the following questions.

Question 1. *What is the time complexity of Cluster Vertex Deletion on chordal graphs?*

Question 2. *What is the time complexity of* s-CVD *on chordal graphs for odd values of* s*?*

Question 3. *Is there a constant factor approximation algorithm for s-CVD,* $s \geq 2$ *on chordal graphs?*

Another generalisation of interval graphs is the class of *cocomparability* graphs. It would be interesting to investigate the following question.

Question 4. *What is the time complexity of* s-CVD *on cocomparability graphs for each* $s \geq 1$*?*

Data availability

No data was used for the research described in the article.

References

[1] Jungho Ahn, Lars Jaffke, O joung Kwon, Paloma T. Lima, Well-partitioned chordal graphs, Discrete Math. 345 (10) (2022) 112985.
[2] Richard D. Alba, A graph-theoretic definition of a sociometric clique, J. Math. Sociol. 3 (1) (1973) 113–126.
[3] Manuel Aprile, Matthew Drescher, Samuel Fiorini, Tony Huynh, A tight approximation algorithm for the cluster vertex deletion problem, Math. Program. 197 (2) (2023) 1069–1091.
[4] Balabhaskar Balasundaram, Sergiy Butenko, Svyatoslav Trukhanov, Novel approaches for analyzing biological networks, J. Combinat. Optim. 10 (1) (2005) 23–39.
[5] Nikhil Bansal, Avrim Blum, Shuchi Chawla, Correlation clustering, Mach. Learn. 56 (1) (2004) 89–113.
[6] Amir Ben-Dor, Ron Shamir, Zohar Yakhini, Clustering gene expression patterns, J. Comput. Biol. 6 (3–4) (1999) 281–297.
[7] Kellogg S. Booth, George S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (3) (1976) 335–379.
[8] Anudhyan Boral, Marek Cygan, Tomasz Kociumaka, Marcin Pilipczuk, A fast branching algorithm for cluster vertex deletion, Theory Comput. Syst. 58 (2) (2016) 357–376.
[9] Yixin Cao, Yuping Ke, Yota Otachi, Jie You, Vertex deletion problems on chordal graphs, Theoret. Comput. Sci. 745 (2018) 75–86.
[10] Dibyayan Chakraborty, L Sunil Chandran, Sajith Padinhatteeri, Raji R Pillai, Algorithms and complexity of s-club cluster vertex deletion, in: International Workshop on Combinatorial Algorithms, Springer, 2021, pp. 152–164.
[11] Dibyayan Chakraborty, L Sunil Chandran, Sajith Padinhatteeri, Raji R Pillai, s-Club cluster vertex deletion on interval and well-partitioned chordal graphs, in: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 2022, pp. 129–143.
[12] Frank Dehne, Michael A Langston, Xuemei Luo, Sylvain Pitre, Peter Shaw, Yun Zhang, The cluster editing problem: Implementations and experiments, in: International Workshop on Parameterized and Exact Computation, Springer, 2006, pp. 13–24.
[13] Michael R Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, Johannes Uhlmann, Graph-based data clustering with overlaps, Discrete Optim. 8 (1) (2011) 2–17.
[14] Aleksander Figiel, Anne-Sophie Himmel, André Nichterlein, Rolf Niedermeier, On 2-clubs in graph-based data clustering: Theory and algorithm engineering, J. Graph Algorithms Appl. 25 (1) (2021) 521–547.
[15] Samuel Fiorini, Gwenaël Joret, Oliver Schaudt, Improved approximation algorithms for hitting 3-vertex paths, Math. Program. 182 (1) (2020) 355–367.
[16] Fedor V Fomin, Serge Gaspers, Daniel Lokshtanov, Saket Saurabh, Exact algorithms via monotone local search, J. ACM 66 (2) (2019a) 1–23.
[17] Fedor V Fomin, Tien-Nam Le, Daniel Lokshtanov, Saket Saurabh, Stéphan Thomassé, Meirav Zehavi, Subquadratic kernels for implicit 3-hitting set and 3-set packing problems, ACM Trans. Algor. (TALG) 15 (1) (2019b) 1–44.
[18] Martin Charles Golumbic, Algorithmic Graph Theory and Perfect Graphs, Elsevier, 2004.
[19] Jens Gramm, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Automated generation of search tree algorithms for hard graph modification problems, Algorithmica 39 (4) (2004) 321–347.
[20] Falk Hüffner, Christian Komusiewicz, Hannes Moser, Rolf Niedermeier, Fixed-parameter algorithms for cluster vertex deletion, Theory Comput. Syst. 47 (1) (2010) 196–217.
[21] S. Khot, O. Regev, Vertex cover might be hard to approximate to within 2- ε, J. Comput. System Sci. 74 (3) (2008) 335–349.

[22] Valerie King, Satish Rao, Rorbert Tarjan, A faster deterministic maximum flow algorithm, J. Algorithms 17 (3) (1994) 447–474.
[23] Jon Kleinberg, Eva Tardos, Algorithm Design, Pearson Education India, 2006.
[24] J. M Lewis, M. Yannakakis, The node-deletion problem for hereditary properties is NP-complete, J. Comput. System Sci. 20 (2) (1980) 219–230.
[25] H. Liu, P. Zhang, D. Zhu, On editing graphs into 2-club clusters, in: Frontiers in Algorithmics and Algorithmic Aspects in Information and Management, Springer, 2012, pp. 235–246.
[26] Robert J. Mokken, Cliques, clubs and clans, Qual. Quant. 13 (1979) 161–173.
[27] James B. Orlin, Max flows in $O(nm)$ time, or better, in: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, 2013, pp. 765–774.
[28] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, Ploutarchos Spyridonos, Community detection in social media, Data Min. Knowl. Discov. 24 (3) (2012) 515–554.
[29] Srinivas Pasupuleti, Detection of protein complexes in protein interaction networks using n-clubs, in: European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer, 2008, pp. 153–164.
[30] Sven Rahmann, Tobias Wittkop, Jan Baumbach, Marcel Martin, Anke Truss, Sebastian Böcker, Exact and heuristic algorithms for weighted cluster editing, in: Computational Systems Bioinformatics, Vol. 6, World Scientific, 2007, pp. 391–401.
[31] Ignasi Sau, Uéverton dos Santos Souza, Hitting forbidden induced subgraphs on bounded treewidth graphs, in: 45th International Symposium on Mathematical Foundations of Computer Science, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
[32] A. Schäfer, Exact Algorithms for S-Club Finding and Related Problems (Ph.D. thesis), Friedrich-Schiller-University Jena, 2009.
[33] Ron Shamir, Roded Sharan, Dekel Tsur, Cluster graph modification problems, Discrete Appl. Math. 144 (1–2) (2004) 173–182.
[34] Roded Sharan, Ron Shamir, CLICK: a clustering algorithm with applications to gene expression analysis, in: Proceedings of the International Conference on Intelligent Systems in Molecular Biology, Vol. 8. No. 307, 2000, p. 16.
[35] Victor Spirin, Leonid A. Mirny, Protein complexes and functional modules in molecular networks, Proc. Natl. Acad. Sci. 100 (21) (2003) 12123–12128.
[36] Dekel Tsur, Faster parameterized algorithm for cluster vertex deletion, Theory Comput. Syst. 65 (2) (2021) 323–343.
[37] Zhenyu Wu, Richard Leahy, An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 15 (11) (1993) 1101–1113.
[38] Zhao Yang, René Algesheimer, Claudio J. Tessone, A comparative analysis of community detection algorithms on artificial networks, Sci. Rep. 6 (1) (2016) 1–18.
[39] M. Yannakakis, Node-and edge-deletion NP-complete problems, in: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, 1978, pp. 253–264.
[40] Jie You, Jianxin Wang, Yixin Cao, Approximate association via dissociation, Discrete Appl. Math. 219 (2017) 202–209.