Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 19, no. 1, pp. 549–565 (2015)
DOI: 10.7155/jgaa.00377

# Connected $(s, t)$-Vertex Separator Parameterized by Chordality

*N. S. Narayanaswamy* [1] *N. Sadagopan* [2]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Madras, India
[2]Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram, India.

**Abstract**

We investigate the complexity of finding a minimum connected $(s, t)$-vertex separator ($(s, t)$-CVS) and present an interesting chordality dichotomy: we show that $(s, t)$-CVS is NP-complete on graphs of chordality at least 5 and present a polynomial-time algorithm for $(s, t)$-CVS on chordality 4 graphs. Further, we show that $(s, t)$-CVS is unlikely to have $\delta log^{2-\epsilon} n$-approximation algorithm, for any $\epsilon > 0$ and for some $\delta > 0$, unless NP has quasi-polynomial Las Vegas algorithms. On the positive-side of approximation, we present a $\lceil \frac{c}{2} \rceil$-approximation algorithm for $(s, t)$-CVS on graphs with chordality $c \geq 3$. Finally, in the parameterized setting, we show that $(s, t)$-CVS parameterized above the $(s, t)$-vertex connectivity is $W[2]$-hard.

| Submitted: | Reviewed: | Revised: | Reviewed: | Revised: |
|---|---|---|---|---|
| October 2012 | April 2013 | March 2014 | October 2014 | November 2014 |
| Reviewed: | Revised: | Accepted: | Final: | Published: |
| April 2015 | May 2015 | October 2015 | October 2015 | November 2015 |

Article type: Regular paper
Communicated by: M. Fürer

*E-mail addresses:* swamy@cse.iitm.ac.in (N. S. Narayanaswamy) sadagopan@iiitdm.ac.in (N. Sadagopan)

# 1 Introduction

The vertex or edge connectivity of a graph and the corresponding separators are of fundamental interest in Computer Science and Graph Theory. For a connected graph, a vertex separator is a subset of vertices whose removal disconnects the graph into two or more connected components and the vertex connectivity refers to the size of a minimum vertex separator. Many kinds of vertex separators, stable vertex separators [1], clique vertex separators [18], constrained vertex separators [13], and $\alpha$-balanced separators [13] are of interest to the research community.
As far as complexity results are concerned, finding a minimum vertex separator and a clique vertex separator are polynomial-time solvable, whereas, finding a stable vertex separator and other constrained separators reported in [13] are NP-hard. This shows that imposing an appropriate constraint on the well-studied vertex separator problem makes the problem NP-hard. Interestingly, constrained vertex separators have received much attention in parameterized complexity as well [13, 12]. In particular, Marx et al. in [13] considered the parameterized complexity of constrained separators satisfying some hereditary properties, for example, clique separators and stable separators. It is shown in [13] that the above problems have an algorithm whose running time is $f(k) \cdot n^{O(1)}$, where $k$ is the size of a constrained separator. Algorithms of this nature are popularly known as fixed-parameter tractable algorithms (FPT) with parameter as the solution size [15]. Subsequently, in [14], Marx et al. looked at the computational problem of finding a minimum $(s,t)$-vertex separator ($(s,t)$-CVS) satisfying some non-hereditary property, like connectedness. Interestingly, in [14] it is shown that $(s,t)$-CVS is in FPT.
When a computational problem is known to be NP-complete, it is natural to look at the complexity of the same in special graph classes such as chordal graphs, $P_5$-free graphs, planar graphs, etc. Well known problems such as maximum clique, maximum independent set, and minimum vertex cover have polynomial-time algorithms restricted to chordal graphs which are NP-complete in general graphs. Recent breakthrough due to Lokshtanov et al. [10] reveals that maximum independent set problem in $P_5$-free graphs is polynomial time. Essentially, classical problems which are known to be NP-complete in general graphs have nice polynomial-time algorithms when the input is restricted to graphs with forbidden subgraphs. Moreover, this line of research has received a significant attention in the past as it helps to identify the gap between the NP-Hardness and the polynomial-time solvable input instances. Having highlighted the importance of special graph classes, in this paper, we investigate the complexity of $(s,t)$-CVS in chordal graphs (graphs with no induced cycle of length at least 3) and its super classes. It is a well-known fact that in chordal graphs every minimal vertex separator is a clique [7]. It is clear that $(s,t)$-CVS is trivially solvable in chordal graphs. It is now natural to study $(s,t)$-CVS on graphs of higher chordality. A graph is said to have chordality $c$ ($c \geq 3$), if it does not contain any induced cycle of length at least $c+1$. To the best of our knowledge the complexity of $(s,t)$-CVS in graphs of higher chordality (henceforth, chordality

$c$ graphs) is open. With these motivations, in this paper, we focus our attention on the computational complexity of minimum connected $(s,t)$-vertex separator in chordality $c$ graphs.

**Remark:** The $(s,t)$-CVS can also be motivated from the theory of graph minors. We observe that there is an equivalence between the computational problems of finding a minimum connected $(s,t)$-vertex separator and a minimum set of edges whose contraction reduces the $(s,t)$-vertex connectivity to one. It is important to note that the analogous computational problem of reducing the $(s,t)$-edge connectivity to zero by a minimum number of edge deletions is polynomial-time solvable, because this is computationally equivalent to finding a minimum $(s,t)$-cut and deleting all edges in it.

**Our Results:** In this paper, we consider connected undirected unweighted non-complete simple graphs. For a graph $G$, let $(s,t)$ denote a fixed non-adjacent pair of vertices in $G$. Throughout this paper, when we refer to edge contraction, we do not contract edges incident on $s$ and edges incident on $t$.

1. As mentioned in the introduction, on chordal graphs every minimal vertex separator is a clique and therefore the $(s,t)$-CVS is immediately guaranteed in chordal graphs. Further, finding a minimum $(s,t)$-CVS in chordal graphs is equivalent to finding a minimum vertex separator which is polynomial-time solvable [7]. We show that deciding $(s,t)$-CVS is NP-complete on graphs of chordality 5 and on chordality 4 graphs $(s,t)$-CVS is polynomial-time solvable. This result is due to a very interesting structural property of minimal vertex separators in chordality 4 graphs and it says that every minimal vertex separator $S$ is either connected or there exist two vertices $u$ and $v$ such that both $u$ and $v$ have a neighbour to each connected component of $S$ in $G$.

2. As far as approximation algorithms are concerned, we present two results. We first present a $\lceil \frac{c}{2} \rceil$-approximation algorithm for $(s,t)$-CVS on graphs with chordality $c \geq 3$. We then establish an approximation preserving polynomial-time reduction from the Group Steiner Tree [9, 6] to $(s,t)$-CVS. Consequently, it follows that there is no polynomial-time approximation algorithm with approximation factor $\delta log^{2-\epsilon} n$ for some $\delta > 0$ and for any $\epsilon > 0$, unless NP has quasi-polynomial Las Vegas algorithms.

3. Our final result is from parameterized complexity theory. As mentioned before Marx et al. [14] have shown that $(s,t)$-CVS is in FPT with parameter as the size of the connected vertex separator. Since an important lower bound for $(s,t)$-CVS is the $(s,t)$-vertex connectivity itself. It is now natural to consider the following parameterization: the size of a $(s,t)$-CVS minus the $(s,t)$-vertex connectivity. This type of parameterization

is known as the above guarantee parameterization [11, 8]. We show that $(s,t)$-CVS parameterized above the $(s,t)$-vertex connectivity is unlikely to be fixed-parameter tractable under the standard parameterized complexity assumption, and in the terminology of parameterized hardness theory, it is hard for the complexity class $W[2]$ in the $W$-hierarchy.

**Graph Preliminaries:** Notation and definitions are as per [7, 16]. Let $G = (V,E)$ be a connected undirected unweighted simple graph where $V(G)$ is the set of vertices and $E(G)$ is the set of edges. For $S \subset V(G)$, $G[S]$ denote the graph induced on the set $S$ and $G \setminus S$ is the induced graph on the vertex set $V(G) \setminus S$. A vertex separator $S \subset V(G)$ is called a $(s,t)$-vertex separator if in $G \setminus S$, $s$ and $t$ are in two different connected components and $S$ is minimal if no proper subset of it is a $(s,t)$-vertex separator. A minimum $(s,t)$-vertex separator is a minimal $(s,t)$-vertex separator of least size. The $(s,t)$-vertex connectivity denote the size of a minimum $(s,t)$-vertex separator. A connected $(s,t)$-vertex separator $S$ is a $(s,t)$-vertex separator such that $G[S]$ is connected and such a set $S$ of least size is a minimum connected $(s,t)$-vertex separator. For a minimal $(s,t)$-vertex separator $S$, let $C_s$ and $C_t$ denote the connected components of $G \setminus S$ such that $s$ is in $C_s$ and $t$ is in $C_t$. We let $G \cdot e$ denote the graph obtained by contracting the edge $e = \{u,v\}$ in $G$ such that $V(G \cdot e) = V(G) \setminus \{u,v\} \cup \{z_{uv}\}$ and $E(G \cdot e) = \{\{z_{uv},x\} \mid \{u,x\}$ or $\{v,x\} \in E(G)\} \cup \{\{x,y\} \mid \{x,y\} \in E(G)$ and $x \neq u, y \neq v\}$. A graph is said to have chordality $c$, if it contains no induced cycle of length at least $c+1$. i.e., every cycle $C$ of length at least $c+1$ in $G$ has a chord (an edge joining a pair of non-consecutive vertices in $C$).

**Roadmap:** In Section 2, we analyze the complexity of $(s,t)$-CVS on chordality $c$ graphs and present our dichotomy result. We then present an approximation algorithm with approximation ratio as a function of chordality of the graph. In Section 3, we present a classical and an approximation hardness for $(s,t)$-CVS. We conclude Section 3 by presenting a parameterized hardness for the above guarantee $(s,t)$-CVS.

# 2 Complexity of $(s,t)$-CVS on Chordality $c$ graphs

The objective of this section is to look at the complexity of $(s,t)$-CVS with chordality as the parameter. Towards this end, we show that $(s,t)$-CVS is NP-complete on chordality 5 graphs and we present a polynomial-time algorithm for $(s,t)$-CVS on chordality 4 graphs. We conclude this section with a $\lceil \frac{c}{2} \rceil$-approximation algorithm for $(s,t)$-CVS on graphs of chordality $c \geq 3$. In our reduction, we choose Steiner tree problem as the candidate problem and it is defined as follows;

**Steiner tree problem:**
**Instance:** A graph $G$, a terminal set $R \subseteq V(G)$, and an integer $r$
**Question:** Is there a subtree in $G$ that contains all of $R$ with at most $r$ edges.

**Theorem 1** *$(s,t)$-CVS is NP-complete on chordality 5 graphs.*

**Proof:** **$(s,t)$-CVS is in NP:** Given an input instance $(G, s, t, q)$ of $(s,t)$-CVS, the certificate on Yes instances is a set $S \subseteq V(G)$ which is a connected $(s,t)$-vertex separator of cardinality at most $q$. Clearly, $S$ can be verified in polynomial time by standard reachability algorithms [2].
**$(s,t)$-CVS is NP-hard:** It is known from [17] that Steiner tree problem on split graphs is NP-complete and this can be reduced in polynomial time to $(s,t)$-CVS in chordality 5 graphs using the following construction. Note that any split graph $G$ can be seen as a graph with $V(G) = V_1 \cup V_2$ such that $G[V_1]$ is a clique and $G[V_2]$ is an independent set. Also, split graphs are a subclass of chordal graphs and hence have chordality 3. We map an instance $(G, R, r)$ of Steiner tree problem on split graphs to the corresponding instance $(G', s, t, q = r + 1)$ of $(s,t)$-CVS as follows: $V(G') = V(G) \cup \{s, t\}$ and $E(G') = E(G) \cup \{\{s, v\} \mid v \in R\} \cup \{\{t, v\} \mid v \in R\}$. An example is illustrated in Figure 1. We now show that instances created by this transformation have chordality 5. i.e., in $G'$, any cycle $C$ of length at least 6 has a chord. Clearly, $C$ must contain either $s$ or $t$ but not both. Let $\{s, u_1, \ldots, u_p\}, p \geq 5$ denote the ordering of vertices in $C$.

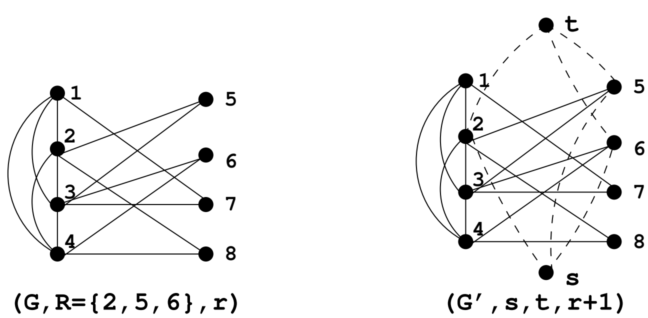


Figure 1: Reduction: Steiner tree in Split Graphs to $(s,t)$-CVS in Chordality 5 graphs

**Case 1:** $\{u_1, u_p\} \subseteq V_2$. Since $G$ is a split graph, $\{u_2, u_{p-1}\} \subset V_1$, and therefore, $\{u_2, u_{p-1}\} \in E(G)$ which is a chord in $C$.
**Case 2:** $u_1 \in V_2$ and $u_p \in V_1$. Clearly, $u_2 \in V_1$ and $\{u_2, u_p\} \in E(G)$, a chord in $C$.
Therefore, we conclude that chordality of $G'$ is at most 5. We now show that $(G, R, r)$ has a Steiner tree with at most $r$ edges if and only if $(G', s, t, q = r+1)$

has a $(s,t)$-CVS of size at most $r+1$. For *only if* claim, $G$ has a Steiner tree $T$ containing all vertices of $R$ and at most $r$ edges. By our construction of $G'$, to disconnect $s$ and $t$, we must remove the set $N_{G'}(s)$ which is $R$, as there is an edge from each element of $N_{G'}(s)$ to $t$. Since $G$ has a Steiner tree $T$ with at most $r$ edges, implies that $T$ has at most $r+1$ vertices. Clearly, in $G'$, $T$ guarantees a $(s,t)$-CVS of size at most $r+1$. For *if* claim, $G'$ has a $(s,t)$-CVS $S$ with at most $r+1$ vertices. Note that any spanning tree on at most $r+1$ vertices has at most $r$ edges. From our construction of $G'$, it follows that $N_{G'}(s) \subseteq S$ and the $(s,t)$-vertex connectivity is $|N_{G'}(s)|$. This implies that $G$ has a Steiner tree with at most $r$ edges containing $R = N_{G'}(s)$ as the terminal set. Hence the claim. $|V(G')| = |V(G)|+2$ and $|E(G')| \leq |E(G)|+2|V(G)|$ and the construction of $G'$ takes $O(|E(G)|)$. Hence, this is a polynomial-time reduction. As a consequence, it follows that $(s,t)$-CVS in chordality 5 graphs is NP-hard. Thus, we conclude $(s,t)$-CVS in chordality 5 graphs is NP-complete. □

## 2.1 $(s,t)$-CVS in Chordality 4 Graphs is Polynomial time

In this section, we present the other half of our dichotomy result which says that $(s,t)$-CVS in chordality 4 graphs is polynomial-time solvable. We now present a sequence of combinatorial results on the structure of minimal vertex separators in chordality 4 graphs, using which we show that $(s,t)$-CVS in chordality 4 graphs is polynomial-time solvable.

**Theorem 2** *Every minimal $(s,t)$-vertex separator $S$ in a chordality 4 graph $G$ satisfies one of the following properties:*

*(1) $G[S]$ is connected.*

*(2) Let $\{X_1,\ldots,X_r\}, r \geq 2$ denote the set of connected components in $G[S]$ and $V(X_i)$ denotes the vertex set of the component $X_i$. In $G \setminus S$, there exists $u$ in $C_s$ and there exists $v$ in $C_t$ such that for all $1 \leq i \leq r, N_G(u) \cap V(X_i) \neq \emptyset$ and $N_G(v) \cap V(X_i) \neq \emptyset$, where $C_s$ and $C_t$ denote the connected components in $G \setminus S$ containing $s$ and $t$, respectively.*

**Proof:** Our proof is by induction on $n = |V(G)|$. *Base:* $|V(G)| = 3$. The only non-complete chordality 4 graph on 3 vertices is a path on 3 vertices. Clearly, the lemma is true for the base case. Let us now assume all chordality 4 graphs on less than $n$, $n \geq 4$ vertices satisfy our claim. Consider a chordality 4 graph $G$ on $n \geq 4$ vertices. Let $S$ be a minimal $(s,t)$-vertex separator in $G$. If $|S| = 1$, then $S$ is a cut vertex $w$ and our claim is true. Since $w$ is a cut-vertex, $w$ has a neighbour $u$ in $C_1$ and $v$ in $C_2$, where $C_i, i \in \{1,2\}$ is a connected component in $G \setminus \{w\}$. For $|S| \geq 2$, we consider two cases to complete the induction. For clarity purpose, the case analysis is considered to complete the induction.
**Case 1:** $G[S]$ is not an independent set. Let $e = \{x,y\}$ be an edge contained in a connected component $X$ of $G[S]$. Consider the graph $G \cdot e$ obtained from $G$ by contracting $e$. Clearly, $|V(G \cdot e)| = n-1$. Let $S' = (S \setminus \{x,y\}) \cup \{z_{xy}\}$. Edges incident on $x$ or $y$ are now incident on $z_{xy}$. Observe that $S'$ is a minimal

$(s,t)$-vertex separator in $G \cdot e$. If $G[S']$ is connected in $G \cdot e$ then it implies that $G[S]$ is connected in $G$ as well. Otherwise, by the induction hypothesis, in $G \cdot e$, there exists $u$ and $v$ with the desired property. In particular, $V(X') \cap N_{G \cdot e}(u)$ and $V(X') \cap N_{G \cdot e}(v)$ are non empty where $X' = (X \setminus \{x,y\}) \cup \{z_{xy}\}$ and $X$ is the connected component in $S$ containing $x$ and $y$. Since $X'$ is obtained from $X$ and $\{x,y\} \in E(G)$, it follows that $u$ and $v$ are adjacent to $X$ in $G$. Thus, both $u$ and $v$ have the desired property in $G$ too. A snapshot is illustrated in Figure 2.

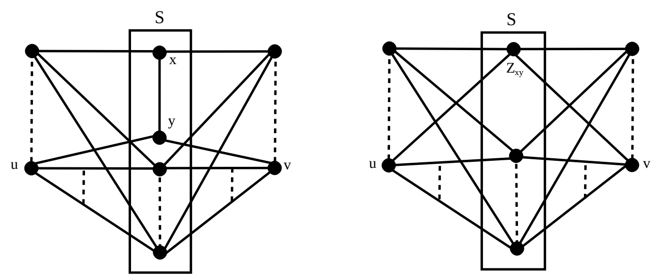


Figure 2: A snapshot illustrating Case 1 of Theorem 2

**Case 2:** $G[S]$ is an independent set. Now consider $x, y \in S$. Consider the graph $G \cdot xy$ obtained by contracting the non-adjacent pair $\{x,y\}$. Let $S' = (S \setminus \{x,y\}) \cup \{z_{xy}\}$ and edges incident on $x$ or $y$ are now incident on $z_{xy}$. Observe that $S'$ is a minimal $(s,t)$-vertex separator in $G \cdot xy$. Clearly, $|V(G \cdot xy)| = |V(G)| - 1$ and hence, by the induction hypothesis, in $G \cdot xy$, there exists $u$ in $C'_s$ and $v$ in $C'_t$ satisfying our claim where $C'_s$ and $C'_t$ are connected components in $(G \cdot xy) \setminus S'$ containing $s$ and $t$, respectively. Let $S = \{x, y, u_1, \ldots, u_p\}, p \geq 0$. We now prove in $G$ the existence of vertex $u$ in $C_s$ satisfying our claim. If $\{u,x\}, \{u,y\} \in E(G)$, then clearly $u \in C_s$ is the desired vertex in $G$. Otherwise, without loss of generality assume that $x \notin N_G(u)$. Thus, $S \setminus \{x\} \subset N_G(u)$. Let $P^s_{xu}$ denote a shortest path between $x$ and $u$ such that the internal vertices are in $C_s$. Consider the vertex $w$ in $P^s_{xu}$ such that $\{x,w\} \in E(G)$. Such a $w$ exists as $S$ is a minimal $(s,t)$-vertex separator in $G$. If for all $z \in S$, $\{w,z\} \in E(G)$, then $w$ is a desired vertex in $C_s$. Otherwise, there exists $z \in S$ such that $\{w,z\} \notin E(G)$. Let $P^s_{wu}$ denote the subpath of $P^s_{xu}$ on the vertex set $\{w = w_1, \ldots, w_q = u\}, q \geq 2$. Let $i$, $2 \leq i \leq q$ be the smallest integer such that, $\{z, w_i\} \in E(G)$. In this case, $P^s_{xw_i}\{w_i, z\}P^t_{xz}$ form an induced cycle of length at least 5 in $G$ where $P^s_{xw_i}$ denote the subpath of $P^s_{xu}$ on the vertex set $\{x, w = w_1, \ldots, w_i\}, 2 \leq i \leq q$. Note that $\{x,z\} \notin E(G)$ as $S$ is an independent set. However, this contradicts the fact that $G$ is a graph of chordality 4. Therefore, there exists a vertex $\hat{u} \in \{u,w\}$ in $C_s$ with the desired property. i.e., either $u$ or $w$ is adjacent to each element (connected component) in $S$. The proof for the existence of vertex $v$ in $C_t$ is symmetric. A snapshot is illustrated in Figure 3. □

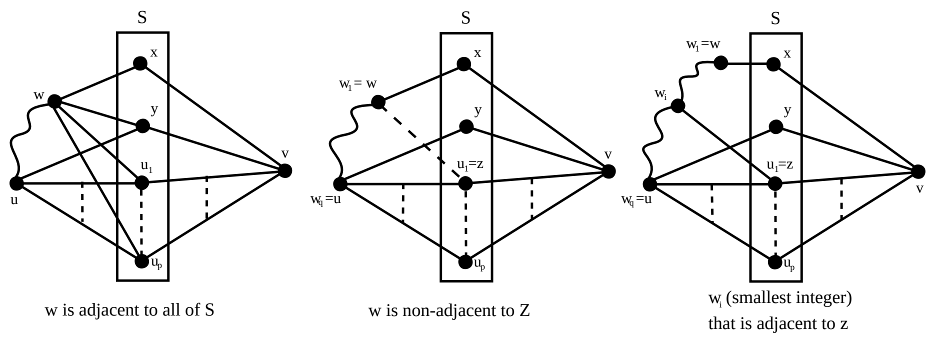


Figure 3: A snapshot illustrating Case 2 of Theorem 2

**Lemma 1** *Let $G$ be a chordality 4 graph with the $(s,t)$-vertex connectivity $k$. The size of any minimum $(s,t)$-CVS in $G$ is either $k$ or $k+1$.*

**Proof:** Note that any minimum $(s,t)$-CVS is of size at least $k$ as the $(s,t)$-vertex connectivity is $k$. If a minimum $(s,t)$-vertex separator itself is connected then we get a minimum $(s,t)$-CVS of size $k$. Otherwise, every minimum $(s,t)$-vertex separator $S$ is such that $G[S]$ is a collection of connected components. In this case, we know from Theorem 2, there exists a vertex $v$ in one of the components of $G \setminus S$ such that $v$ has a neighbour in each connected component of $S$. Therefore, $S \cup \{v\}$ is a minimum $(s,t)$-CVS of size $k+1$. Hence, the lemma is true. □

**Remark:** For a chordality 4 graph $G$ with the $(s,t)$-vertex connectivity $k$, asking for a minimum $(s,t)$-CVS of size $k$ is equivalent to checking whether $G$ contains a connected minimum $(s,t)$-vertex separator, i.e. a minimum $(s,t)$-vertex separator which itself is connected. The Lemma 2 shows that this equivalence checking is indeed polynomial-time solvable.

We now present two more combinatorial observations using which we can find a minimum $(s,t)$-CVS in chordality 4 graphs in polynomial time. We make use of the notion of *contractible edges*. Given a connected graph $G$ with the $(s,t)$-vertex connectivity $k$, an edge $e \in E(G)$ is said to be *contractible* if the $(s,t)$-vertex connectivity in $G \cdot e$ is at least $k$. Otherwise $e$ is called *non-contractible*. For a connected graph $G$ with the $(s,t)$-vertex connectivity $k \geq 2$, let $F = \{\{u,v\} \mid \{u,v\} \in E(G)$ and $\{u,v\}$ is contained in a minimum $(s,t)$-vertex separator $\}$. i.e., the set $F$ is the set of all non-contractible edges in $G$. We use $F$ to denote the set of non-contractible edges in $G$. By $G \cdot F$, we mean the graph obtained from $G$ by contracting all edges in $F$.
**Computing the set $F$:** The set $F$ can be computed in polynomial time. Given a graph $G$ with the $(s,t)$-vertex connectivity $k$, for each edge $e$ in $G$, compute $G \cdot e$ and check whether the $(s,t)$-vertex connectivity is $k-1$. If so, then $e \in F$.

Checking the vertex connectivity of a graph can be done in polynomial time using standard Max-flow Min-cut algorithm [2].

**Lemma 2** *$G \cdot F$ contains a cut-vertex if and only if there exists a minimum $(s,t)$-vertex separator $S$ such that $G[S]$ is connected.*

**Proof:** *If:* Suppose, $G \cdot F$ does not contain a cut-vertex. This implies that after contracting edges in $F$, in $G \cdot F$, every minimum vertex separator $S$ induces at least two connected components. Moreover, this is true even in $G$ as well, contradicting the fact that there exists a connected minimum $(s,t)$-vertex separator in $G$. *Only if:* Suppose every minimum $(s,t)$-vertex separator $S$ is such that $G[S]$ has at least two connected components. Since any edge contraction can not disconnect a graph which is already connected, any sequence of edge contractions of edges in $F$ results in a graph with the vertex connectivity at least two, contradicting the fact that $G \cdot F$ contains a cut-vertex. Hence, the claim follows. □

**Corollary 1** *For a connected graph $G$, deciding whether $G$ contains a connected minimum $(s,t)$-vertex separator is polynomial-time solvable.*

**Proof:** From Lemma 2, it is clear that checking for a connected minimum $(s,t)$-vertex separator in $G$ is equivalent to checking whether $G \cdot F$ contains a cut-vertex or not. This testing can be done using Depth First Search tree computed on $G \cdot F$ and hence, the claim. □

**Lemma 3** *For a chordality 4 graph $G$ with the $(s,t)$-vertex connectivity $k$, deciding whether $(s,t)$-CVS is of size $k$ or $k+1$ is polynomial-time solvable.*

**Proof:** The claim follows from Lemmas 1, 2 and Corollary 1. The decision algorithm *DECIDE-(s,t)-CVS(G,k)* performs the following two tasks, namely, contract all non-contractible edges in $G$ and check the $(s,t)$-vertex connectivity in the resulting graph $G'$. If $\kappa(G') \geq 2$, then the algorithm returns 'NO' which means that every minimum $(s,t)$-CVS is of size $k+1$. Otherwise, it returns 'YES' which means that there exists a minimum $(s,t)$-CVS of size $k$. Since the above tasks can be done using the standard depth first search algorithm, the decision algorithm runs in polynomial time. □

#### 2.1.1 Finding a minimum $(s,t)$-CVS in Chordality 4 graphs

Using *DECIDE-(s,t)-CVS()*, we now show that finding a minimum $(s,t)$-CVS in chordality 4 graphs is also polynomial-time solvable. The approach is to contract all non-contractible edges (edges in the set $F$) and check whether the resulting graph contains a cut-vertex or not. If there is no cut-vertex, then any minimum $(s,t)$-vertex separator in $G$ together with the vertex $v$ in one of the components in $G \setminus S$ (due to Theorem 2) yields a $(s,t)$-CVS of size $k+1$ in $G$. Otherwise, the given chordality 4 graph contains a $(s,t)$-CVS of size $k$. In such a case, we outline a procedure using which we can find a $(s,t)$-CVS $S$ of size $k$. Our procedure (Algorithm 1) makes polynomial number of calls to *DECIDE-(s,t)-CVS()* to output the desired set.

**Lemma 4** *Let $G$ be a chordality 4 graph with the $(s,t)$-vertex connectivity $k \geq 2$. $G$ has a $(s,t)$-CVS of size $k$ if and only if there exists a non-contractible edge $e$ in $G$ such that $G \cdot e$ has a $(s,t)$-CVS of size $k-1$.*

**Proof:** *If:* Let $S$ be a $(s,t)$-CVS of size $k$ in $G$. Since $G[S]$ is connected, there exists $u, v \in S$ such that $\{u,v\} \in E(G)$. Since the cardinality of $S$ is same as the $(s,t)$-vertex connectivity, the edge $e = \{u,v\}$ is non-contractible. Moreover, contracting $e$ leaves a graph $G \cdot e$ in which $S' = (S \setminus \{u,v\}) \cup \{z_{uv}\}$ is a vertex separator, where $z_{uv}$ is a new vertex created due to the contraction of $\{u,v\}$. Since $G[S]$ is connected and any edge contraction does not disconnect a subgraph which is already connected, $G[S']$ is a $(s,t)$-CVS of size $k-1$. Therefore, the necessity follows. *Only if:* Let $S$ be a $(s,t)$-CVS of size $k-1$ in $G \cdot e$. Clearly, $z_{uv} \in S$, the vertex corresponding to the contraction of the edge $\{u,v\}$. In $G$, $S' = (S \setminus \{z_{uv}\}) \cup \{u,v\}$ is a $(s,t)$-CVS of size $k$. Therefore, the sufficiency follows. □

The above combinatorial observation together with *DECIDE-(s,t)-CVS()*, we obtain a polynomial-time algorithm to find a minimum $(s,t)$-CVS of size $k$ and is presented in Algorithm 1.

**Lemma 5** *Let $G$ be a chordality 4 graph having a $(s,t)$-CVS of size $k$. Algorithm 2 outputs a $k$-sized $(s,t)$-CVS in polynomial time.*

**Proof:** The proof of this lemma follows from the fact that Algorithm 2 is an implementation of Lemma 4. The main purpose of Lemma 4 is to ensure that there is no backtracking on an edge $e$ whose contraction reduces the $(s,t)$-vertex connectivity by 1. □

**Algorithm 1** A Polynomial-time Algorithm to find a minimum $(s,t)$-CVS in Chordality 4 graphs

**Input:** Chordality 4 graph $G$ with the $(s,t)$-vertex connectivity $k$
If $k = 1$ then simply output any cut-vertex in $G$
**if** *DECIDE-(s,t)-CVS(G,k)* returns 'NO' **then**
    Find a minimum $(s,t)$-vertex separator $S$ in $G$ using classical vertex connectivity algorithm
    Output the set $S \cup \{v\}$ where $v$ is in one of the components of $G \setminus S$ such that $S \subseteq N_G(v)$, is a minimum $(s,t)$-CVS
**else**
    `/*--- there exists a` $k$`-size` $(s,t)$`-CVS. To obtain one such separator, perform the following; ---*/`
    *Find-$(s,t)$-CVS(G,k)*
**end if**

**Theorem 3** *Algorithm 1 outputs a minimum $(s,t)$-CVS in polynomial time.*

**Algorithm 2** Finding $k$-size $(s,t)$-CVS in Chordality 4 graphs *Find-*$(s,t)$*-CVS(G,k)*

```
/* Return value is 'fail' or a connected vertex separator */
If (DECIDE-(s,t)-CVS(G,k) returns 'NO') return 'fail'
for each non-contractible edge e in G do
    x = Find-(s,t)-CVS(G · e, k − 1)
    if (x == 'fail') continue
    /* continue goes to the beginning of the for-loop */
    return x
end for
```

**Proof:** From Lemma 1, we know that a minimum $(s,t)$-CVS is of size $k$ or $k+1$. To decide between $k$ and $k+1$, it is sufficient to check for a cut-vertex in $G \cdot F$ as per Lemma 2. This step can be implemented in polynomial time, by identifying the edges which are elements of $F$. Every edge whose contraction reduces the connectivity by 1 is in $F$. Then $G \cdot F$ is checked for the presence of a cut-vertex, and this can be done by a DFS. If the size of the minimum $(s,t)$-CVS is $k+1$, then steps 4 and 5 of Algorithm 1 outputs a $(s,t)$-CVS of size $k+1$ in polynomial time, by finding a minimum $(s,t)$-vertex separator. If the minimum $(s,t)$-CVS is of size $k$, then Algorithm 2 returns a minimum connected $(s,t)$-CVS. Overall, a minimum $(s,t)$-CVS can be obtained in polynomial time. □

## 2.2 $(\lceil \frac{c}{2} \rceil)$-Approximation for $(s,t)$-CVS on Graphs with Chordality $c$

**Lemma 6** *Let $G$ be a graph of chordality $c \geq 3$. For each minimal vertex separator $S$, for each $u, v \in S$ such that $\{u,v\} \notin E(G)$, there exists a path of length at most $\lceil \frac{c}{2} \rceil$ whose internal vertices are in $C_s$ or $C_t$, where $C_s$ and $C_t$ are components in $G \setminus S$ containing $s$ and $t$, respectively.*

**Proof:** Suppose for some non-adjacent pair $\{u,v\} \subseteq S$, both $P^1_{uv}$ and $P^2_{uv}$ are of length more than $\lceil \frac{l}{2} \rceil$, where $P^1_{uv}$ and $P^2_{uv}$ are shortest paths from $u$ to $v$ whose internal vertices are in $C_s$ and $C_t$, respectively. Now, there is an induced cycle $C$ containing $u$ and $v$ such that $|C| > \lceil \frac{l}{2} \rceil + \lceil \frac{l}{2} \rceil = l$. However, this contradicts the fact that $G$ is of chordality $l$. □

Let $OPT$ denote the size of any minimum $(s,t)$-CVS on chordality $c$ graphs. Clearly, $OPT \geq k$, where $k$ is the $(s,t)$-vertex connectivity. The description of approximation algorithm $ALG$ is as follows:

**Theorem 4** *Algorithm 3 outputs $(s,t)$-CVS in polynomial time with approximation ratio $\lceil \frac{c}{2} \rceil$.*

**Proof:** Observe that $S'$ is a $(s,t)$-CVS in $G$. The upper bound on the size of $S'$ output by $ALG$ is: $|S'| \leq k + (k-1)(\lceil \frac{c}{2} \rceil - 1)$. Therefore, approximation ratio $\beta$ is

**Algorithm 3** Approximation Algorithm for $(s,t)$-CVS on Chordality $c$ Graphs

1: Compute a minimum $(s,t)$-vertex separator $S$ in $G$. $S = \{v_1, \ldots, v_k\}$ be an arbitrary ordering of vertices in $S$
2: **for** each non-adjacent pair $\{v_i, v_{i+1}\} \subseteq S, 1 \leq i \leq k-1$, **do**
3: find a path $P_{v_iv_{i+1}}$ of length at most $\lceil \frac{c}{2} \rceil$ whose internal vertices are in $C_s$ or $C_t$. Such a path exists as per Lemma 6
4: $S' = \bigcup_{1 \leq i \leq k-1} V(P_{v_iv_{i+1}}) \cup S$
5: **end for**

$\beta \leq \frac{k+(k-1)(\lceil \frac{c}{2} \rceil - 1)}{k} = 1 + (1 - \frac{1}{k})(\lceil \frac{c}{2} \rceil - 1) < 1 + (\lceil \frac{c}{2} \rceil - 1) = \lceil \frac{c}{2} \rceil$

Step 1 of the Algorithm 3 incurs $O(n^3)$ time to output a minimum $(s,t)$-vertex separator in $G$. To implement step 3, we can make use of the standard reachability algorithm like Breadth First Search (BFS) to output $P_{v_iv_{i+1}}$ and this call is made for at most $O(n^2)$ time. Therefore, the overall time-complexity of the Algorithm 3 is $(mn^2)$, where $O(m)$ is the time incurred for BFS subroutine. □

# 3 Complexity of $(s,t)$-CVS: Hardness Results

The purpose of this section is two fold. Although in [14] it is shown that $(s,t)$-CVS is FPT, no explicit reduction is shown to establish NP-hardness result. In this section, we first establish a classical hardness of $(s,t)$-CVS by presenting a polynomial-time reduction from the Group Steiner tree to $(s,t)$-CVS. Moreover, the same reduction establishes an hardness of approximation for $(s,t)$-CVS. We conclude this section by showing that $(s,t)$-CVS parameterizing above the $(s,t)$-vertex connectivity is $W[2]$-hard.

## 3.1 Classical Hardness: A Reduction from Group Steiner tree to $(s,t)$-CVS

The decision version of $(s,t)$-CVS is given below

**Instance:** A graph $G$, a non-adjacent pair $(s,t)$, and $q \in \mathbb{Z}^+$
**Question:** Is there a $(s,t)$-vertex separator $S \subset V(G)$, $|S| \leq q$ and $G[S]$ is connected?

The Group Steiner tree problem can be stated as follows: given a connected undirected unweighted graph $G$, an integer $r$, and a collection of sets, which we call groups $g_1, g_2, \ldots, g_l \subseteq V(G)$, the objective is to find a subtree $T$ of $G$ with at most $r$ edges that contains at least one vertex from each group $g_i$. We assume that the groups are disjoint. The Group Steiner tree problem is a generalization of the Steiner tree problem [5] and therefore, it is NP-complete.
We transform an instance $I = (G, g_1, g_2, \ldots, g_l \subseteq V(G), r)$ of the Group Steiner tree to the corresponding instance $I' = (G', s, t, l+r+1)$ of $(s,t)$-CVS as follows:

$V(G') = V(G) \cup \{s,t\} \cup \{x_i \mid 1 \le i \le l\}$. $E(G') = E(G) \cup \{\{s,x_i\} \mid 1 \le i \le l\} \cup \{\{t,x_i\} \mid 1 \le i \le l\} \cup \{\{x_i,y\} \mid y \in g_i \text{ and } 1 \le i \le l\}$. An example is illustrated in Figure 4.

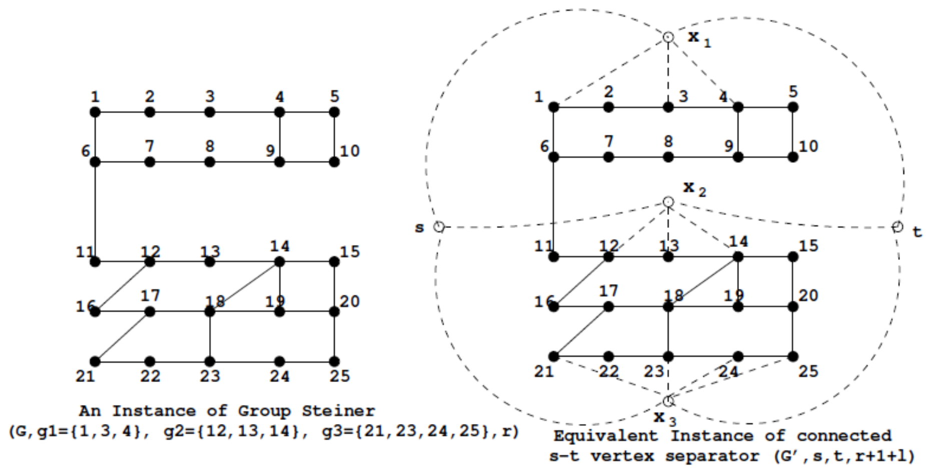


Figure 4: An instance of Group Steiner tree reduces to an instance of $(s,t)$-CVS

**Theorem 5** *$(s,t)$-CVS is NP-complete. Further, $(s,t)$-CVS is unlikely to have $\delta \log^{2-\epsilon} n$-approximation algorithm, for any $\epsilon > 0$ and for some $\delta > 0$, unless NP has quasi-polynomial Las Vegas algorithms.*

**Proof:** To establish NP-hardness result, we prove the following claim. For $I$ and $I'$ as defined above, $G$ has a Group Steiner tree with at most $r$ edges if and only if $G'$ has a $(s,t)$-CVS of size at most $r+1+l$. We first prove the necessity. Given that $G$ has a Group Steiner tree $T$ with at most $r$ edges that contains at least one vertex from each group $g_i$. By the construction of $G'$, it is clear that the $(s,t)$-vertex connectivity is $l$. Therefore, any $(s,t)$-CVS in $G'$ has at least $l$ vertices. Clearly, these $l$ new vertices together with at most $r+1$ vertices in $T$ form a $(s,t)$-CVS of size at most $r+1+l$ in $G'$. Conversely, by the construction of $G'$, any $(s,t)$-CVS $S$ of size at most $r+1+l$ must contain all $x_i$'s. i.e. $N_{G'}(s) \subset S$. This is true because $N_{G'}(s)$ is a $(s,t)$-vertex separator. Since $S$ is connected and $N_{G'}(s)$ is an independent set, it follows that by the construction $S \setminus N_{G'}(s)$ is connected. Moreover, $S$ must contain at least one element of $N_{G'}(x_i)$ for each $x_i$. Since $|S \setminus N_{G'}(s)| \le r+1$, any spanning tree on $S \setminus N_{G'}(s)$ is a Group Steiner tree with at most $r$ edges. As a consequence of the above claim, it follows that $(s,t)$-CVS is NP-hard and it is easy to verify that $(s,t)$-CVS is in NP as certificate testing can be done in polynomial time using standard graph traversals [2]. Therefore, $(s,t)$-CVS is NP-complete. □

We now show that our reduction establishes a stronger result: $(s,t)$-CVS is unlikely to have $\delta \log^{2-\epsilon} n$-approximation algorithm, for any $\epsilon > 0$ and for some $\delta > 0$, unless NP has quasi-polynomial Las Vegas algorithms.

**Hardness of Approximation of $(s,t)$-CVS:** The Group Steiner tree problem with $l$ groups is at least as hard as the Set Cover problem, thus can not be approximated to a factor $o(\log l)$, unless $P = NP$ [4]. On the hardness of approximation due to [9], the following result is known: there is no polynomial-time approximation algorithm for Group Steiner tree with approximation factor $\delta \log^{2-\epsilon} n$ for some $\delta > 0$ and for any $\epsilon > 0$, unless NP has quasi-polynomial Las Vegas algorithms. We now show that the above reduction is an approximation-ratio preserving reduction. Let $OPT_g$ and $OPT_c$ denote the size of any optimum solution of the Group Steiner tree problem and the $(s,t)$-CVS problem, respectively. Note that $OPT_c = OPT_g + l$ and $OPT_g \geq l$. Suppose there is an $(1+\alpha)$-approximation algorithm for $(s,t)$-CVS, where $\alpha \leq \delta \log^{2-\epsilon} n$, for some $\delta, \epsilon > 0$. Then the size of the output of the algorithm is $(1+\alpha)OPT_c = (1+\alpha)(OPT_g + l) \leq (1+\alpha)(OPT_g + OPT_g) = 2(1+\alpha)OPT_g$. This implies $2(1+\alpha)$-approximation algorithm for the Group Steiner tree problem, which is unlikely, unless NP has quasi-polynomial Las Vegas algorithms [9]. □

### 3.2 $(s,t)$-CVS Parameterized above the $(s,t)$-vertex connectivity is $W[2]$-hard

We consider the following parameterization which is the size of $(s,t)$-CVS minus the $(s,t)$-vertex connectivity. Since the size of every $(s,t)$-CVS is at least the $(s,t)$-vertex connectivity, it is natural to parameterize above the $(s,t)$-vertex connectivity and its parameterized version is defined below.

**$(s,t)$-CVS parameterized above the $(s,t)$-vertex connectivity:**
**Instance:** A graph $G$, a non-adjacent pair $(s,t)$ with the $(s,t)$-vertex connectivity $k$ and $r \in \mathbb{Z}^+$
**Parameter:** $r$
**Question:** Is there a $(s,t)$-vertex separator $S \subset V(G)$, $|S| \leq k + r$ such that $G[S]$ is connected?

We now show that there is no fixed-parameter tractable algorithm for $(s,t)$-CVS parameterized above the $(s,t)$-vertex connectivity. In order to characterize those problems that do not seem to admit a fixed-parameter tractable algorithms, Downey and Fellows defined a *parameterized reduction* and a hierarchy of intractable parameterized problem classes above FPT, the popular classes are $W[1]$ and $W[2]$. We refer [15] for details about parameterized reductions. We now present a parameterized reduction from parameterized Steiner tree problem to $(s,t)$-CVS parameterized above the $(s,t)$-vertex connectivity. This parameterized version of Steiner tree problem is shown to be $W[2]$-hard in [3].

| |
|---|
| **Parameterized Steiner tree problem:** |
| **Instance:** A graph $G$, a terminal set $R \subseteq V(G)$, and an integer $r$ |
| **Parameter:** $r$ |
| **Question:** Is there a set of vertices $T \subseteq V(G) \setminus R$ such that $\|T\| \leq r$ and $G[R \cup T]$ is connected? $T$ is called Steiner set (Steiner vertices). |

**Theorem 6** *$(s,t)$-CVS Parameterized above the $(s,t)$-vertex connectivity is $W[2]$-hard.*

**Proof:** Given an instance $(G, R, r)$ of Steiner tree problem, we construct the corresponding instance $(G', s, t, k, r)$ of $(s,t)$-CVS with the $(s,t)$-vertex connectivity $k = |R|$ as follows: $V(G') = V(G) \cup \{s,t\}$ and $E(G') = E(G) \cup \{\{s,v\} \mid v \in R\} \cup \{\{t,v\} \mid v \in R\}$. We now show that $(G, R, r)$ has a Steiner tree with at most $r$ Steiner vertices if and only if $(G', (s,t), k, r)$ has a $(s,t)$-CVS of size at most $k+r$. For *only if* claim, $G$ has a Steiner tree $T$ containing all vertices of $R$ and at most $r$ Steiner vertices. By our construction of $G'$, to disconnect $s$ and $t$, we must remove the set $N_{G'}(s)$ which is $R$, as there is an edge from each element of $N_{G'}(s)$ to $t$. Since $G$ has a Steiner tree with at most $r$ Steiner vertices, implies that in $G'$, it guarantees a $(s,t)$-CVS of size at most $k+r$. For *if* claim, $G'$ has a $(s,t)$-CVS $S$ with at most $k+r$ vertices. Since the $(s,t)$-vertex connectivity is $k$ and $S$ is a $(s,t)$-vertex separator, from our construction of $G'$ it follows that $N_{G'}(s) \subseteq S$ and $k = |N_{G'}(s)|$. This implies that $G$ has a Steiner tree with $R = N_{G'}(s)$ as the terminal set and $S \setminus N_{G'}(s)$ as the Steiner vertices of size at most $r$. Hence the claim. $|V(G')| = |V(G)| + 2$ and $|E(G')| \leq |E(G)| + 2|V(G)|$ and the construction of $G'$ takes $O(|E(G)|)$. Clearly, the reduction is a parameter preserving parameterized reduction. Therefore, we conclude that deciding whether a graph has a $(s,t)$-CVS is $W[2]$-hard with parameter $r$. $\square$

**Concluding Remarks and Further Research:** In this paper, we have investigated the complexity of minimum connected $(s,t)$-vertex separator ($(s,t)$-CVS) on graphs of higher chordality as finding a minimum $(s,t)$-CVS in chordal graphs is polynomial-time solvable. We have presented a chordality dichotomy which says that $(s,t)$-CVS is NP-complete on chordality 5 graphs and polynomial-time solvable on chordality 4 graphs. Further, we have presented a $\lceil \frac{c}{2} \rceil$-approximation algorithm on graphs with chordality $c \geq 3$. We also reported a non-approximiability result and in the parameterized-setting, we have established that parameterizing above the $(s,t)$-vertex connectivity is $W[2]$-hard. An interesting problem for further research is to parameterize $(s,t)$-CVS by the $(s,t)$-vertex connectivity.

# Acknowledgements

The authors wish to thank all reviewers for their comments and suggestions.

## References


[1] A. Brandstädt, F. F. Dragan, V. B. Le, and T. Szymczak. On stable cutsets in graphs. *Discrete Applied Mathematics*, 105(13):39 – 50, 2000. doi:10.1016/S0166-218X(00)00197-9.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2009.

[3] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer, 1999.

[4] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, July 1998. doi:10.1145/285055.285059.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[6] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66 – 84, 2000. doi:10.1006/jagm.2000.1096.

[7] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, volume 57 of *Annals of Discrete Mathematics*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2nd edition edition, 2004.

[8] G. Gutin and A. Yeo. Constraint satisfaction problems parameterized above or below tight bounds: A survey. In H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, editors, *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *Lecture Notes in Computer Science*, pages 257–286. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-30891-8_14.

[9] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 585–594, New York, NY, USA, 2003. ACM. doi:10.1145/780542.780628.

[10] D. Lokshantov, M. Vatshelle, and Y. Villanger. Independent set in P5-free graphs in polynomial time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 570–581. SIAM, 2014. URL: http://dl.acm.org/citation.cfm?id=2634074.2634117.

[11] M. Mahajan and V. Raman. Parameterizing above guaranteed values: Maxsat and maxcut. *Journal of Algorithms*, 31(2):335 – 354, 1999. doi:10.1006/jagm.1998.0996.

[12] D. Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394 – 406, 2006. Parameterized and Exact Computation-First International Workshop on Parameterized and Exact Computation 2004. doi:10.1016/j.tcs.2005.10.007.

[13] D. Marx, B. O'Sullivan, and I. Razgon. Treewidth Reduction for Constrained Separation and Bipartization Problems. In J.-Y. Marion and T. Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science*, volume 5 of *LIPIcs*, pages 561–572. Schloss Dagstuhl, 2010. doi:10.4230/LIPIcs.STACS.2010.2485.

[14] D. Marx, B. O'sullivan, and I. Razgon. Finding small separators in linear time via treewidth reduction. *ACM Trans. Algorithms*, 9(4):30:1–30:35, Oct. 2013. doi:10.1145/2500119.

[15] R. Niedermeier. *Invitation to fixed parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2006.

[16] D. West. *Introduction to Graph Theory*. Featured Titles for Graph Theory Series. Prentice Hall, 2001.

[17] K. White, M. Farber, and W. Pulleyblank. Steiner trees, connected domination and strongly chordal graphs. *Networks*, 15(1):109–124, 1985. doi:10.1002/net.3230150109.

[18] S. Whitesides. An algorithm for finding clique cut-sets. *Information Processing Letters*, 12(1):31 – 32, 1981. doi:10.1016/0020-0190(81)90072-7.