A tight approximation algorithm for the cluster vertex deletion problem

Manuel Aprile · Matthew Drescher · Samuel Fiorini · Tony Huynh

Received: 26 May 2021 / Accepted: 22 November 2021 / Published online: 7 January 2022
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2021

Abstract
We give the first 2-approximation algorithm for the cluster vertex deletion problem. This approximation factor is tight, since approximating the problem within any constant factor smaller than 2 is UGC-hard. Our algorithm combines previous approaches, based on the local ratio technique and the management of twins, with a novel construction of a “good” cost function on the vertices at distance at most 2 from any vertex of the input graph. As an additional contribution, we also study cluster vertex deletion from the polyhedral perspective, where we prove almost matching upper and lower bounds on how well linear programming relaxations can approximate the problem.

Keywords Approximation algorithm · Cluster vertex deletion · Linear programming relaxation · Sherali-Adams hierarchy.

This paper appeared as an extended abstract in the proceedings of IPCO 2021. See the end of Sect. 1 for a detailed comparison.

This project was supported by ERC Consolidator Grant 615640-ForEFront. Samuel Fiorini and Manuel Aprile are also supported by FNRS grant T008720F-35293308-BD-OCP. Tony Huynh is also supported by the Australian Research Council.

Manuel Aprile
manuelf.aprile@gmail.com

Matthew Drescher
knavely@gmail.com

Samuel Fiorini
sfiorini@ulb.ac.be

Tony Huynh
tony.bourbaki@gmail.com

1 Dipartimento di Matematica, Università degli Studi di Padova, Padua, Italy
2 Département de Mathématique, Université libre de Bruxelles, Brussels, Belgium
3 School of Mathematics, Monash University, Melbourne, Australia
\section{1 Introduction}

A \textit{cluster graph} is a graph that is a disjoint union of complete graphs. Let $G$ be any graph. A set $X \subseteq V(G)$ is called a hitting set if $G - X$ is a cluster graph. Given a graph $G$ and (vertex) cost function $c : V(G) \rightarrow \mathbb{Q}_{\geq 0}$, the \textit{cluster vertex deletion} problem (\textsc{Cluster-VD}) asks to find a hitting set $X$ whose cost $c(X) := \sum_{v \in X} c(v)$ is minimum. We denote by $\text{OPT}(G, c)$ the minimum cost of a hitting set.

If $G$ and $H$ are two graphs, we say that $G$ contains $H$ if some induced subgraph of $G$ is isomorphic to $H$. Otherwise, $G$ is said to be $H$-free. Denoting by $P_k$ the path on $k$ vertices, we easily see that a graph is a cluster graph if and only if it is $P_3$-free.

\textsc{Cluster-VD} has applications in graph modeled data clustering in which an unknown set of samples may be contaminated. An optimal solution for \textsc{Cluster-VD} can recover a clustered data model, retaining as much of the original data as possible [25]. Vertex deletion problems such as \textsc{Cluster-VD}, where one seeks to locate vertices whose removal leaves a graph with desirable properties, often arise when measuring robustness and attack tolerance of real-life networks [1,4,5,26].

From what precedes, \textsc{Cluster-VD} is a hitting set problem in a 3-uniform hypergraph, and as such has a “textbook” 3-approximation algorithm (see for instance the introduction of [19]). Moreover, the problem has an approximation-preserving reduction from \textsc{Vertex Cover}. By adding a pendant edge to each vertex of $G$, one checks that solving \textsc{Cluster-VD} on the new graph is equivalent to solving \textsc{Vertex Cover} on the original graph (see Proposition 1 for more details). Hence, obtaining a $(2 - \varepsilon)$-approximation algorithm for some $\varepsilon > 0$ would contradict either the Unique Games Conjecture or $P \neq \text{NP}$.

The first non-trivial approximation algorithm for \textsc{Cluster-VD} was a $5/2$-approximation due to You, Wang and Cao [40]. Shortly afterward, Fiorini, Joret and Schaudt gave a $7/3$-approximation [19], and subsequently a $9/4$-approximation [20].

\subsection{1.1 Our contribution}

In this paper, we close the gap between 2 and $9/4 = 2.25$ and prove the following tight result.

\textbf{Theorem 1} \textsc{Cluster-VD} has a 2-approximation algorithm.

All previous approximation algorithms for \textsc{Cluster-VD} are based on the local ratio technique. See the survey of Bar-Yehuda, Bendel, Freund, and Rawitz [23] for background on this standard algorithmic technique. Our algorithm is no exception, see Algorithm 1 below. However, it significantly differs from previous algorithms in its crucial step (see Step 14 below). In fact, almost all our efforts in this paper focus on that particular step of the algorithm (see Theorem 2), which searches for a special type of induced subgraph of $G$, which we now describe.

Let $H$ be an induced subgraph of $G$, and let $c_H : V(H) \rightarrow \mathbb{Q}_{\geq 0}$. The weighted graph $(H, c_H)$ is said to be $\alpha$-\textit{good in} $G$ (for some factor $\alpha \geq 1$) if $c_H$ is not identically
0 and \( c_H(X \cap V(H)) \leq \alpha \cdot \text{OPT}(H, c_H) \) holds for every (inclusionwise) minimal hitting set \( X \) of \( G \). We overload terminology and say that an induced subgraph \( H \) is \( \alpha \)-good in \( G \) if there exists a cost function \( c_H \) such that \( (H, c_H) \) is \( \alpha \)-good in \( G \). We stress that the local cost function \( c_H \) is defined obliviously of the global cost function \( c : V(G) \to \mathbb{Q}_{\geq 0} \).

A pair of vertices \( u, u' \) of \( G \) are called twins if \( uu' \in E(G) \), and for all \( v \in V(G - u - u') \), \( uv \in E(G) \) if and only if \( u'v \in E(G) \). We say that \( G \) is twin-free if \( G \) has no twins. As in [19,20], if \( G \) has a pair of twins \( u, u' \), then CLUSTER-VD admits an easy reduction step (see Steps 8–12). The idea is simply to add the cost of \( u' \) to that of \( u \) and delete \( u' \). This works since \( u' \) belongs to a minimal hitting set of \( G \) if and only if \( u \) does (see [19] for a complete proof). Therefore, when searching for \( \alpha \)-good induced subgraphs, we may assume that \( G \) is twin-free, which is crucial for our proofs.

![Algorithm 1 CLUSTER-VD-APX (G, c)](image)

We will use two methods to establish \( \alpha \)-goodness of induced subgraphs. We say that \((H, c_H)\) is strongly \( \alpha \)-good if \( c_H \) is not identically \( 0 \) and \( c_H(V(H)) \leq \alpha \cdot \text{OPT}(H, c_H) \).

Clearly, if \((H, c_H)\) is strongly \( \alpha \)-good then \((H, c_H)\) is \( \alpha \)-good in \( G \), for every graph \( G \) which contains \( H \). We say that \( H \) itself is strongly \( \alpha \)-good if \((H, c_H)\) is strongly \( \alpha \)-good for some cost function \( c_H \).

Let \( N_{\leq i}(v) \) (resp. \( N_i(v) \)) be the set of vertices at distance at most (resp. equal to) \( i \) from vertex \( v \). We abbreviate \( N(v) := N_1(v) \) and \( N[v] := N_{\leq 1}(v) \). If we cannot find a strongly \( \alpha \)-good induced subgraph in \( G \), we will find an induced subgraph \( H \) that has a special vertex \( v_0 \) such that \( N[v_0] \) is entirely contained in \( H \), and a cost

\[1\] We warn the reader that, in other papers, twins are usually called true twins, whereas two vertices which have the same set of neighbours are called false twins (note that false twins are not adjacent). Since we have no need of false twins in this paper, we have chosen to use twins in place of true twins.
function $c_H : V(H) \to \mathbb{Z}_{\geq 0}$ such that $c_H(v) \geq 1$ for all vertices $v \in N[v_0]$ and $c_H(V(H)) \leq \alpha \cdot \text{OPT}(H, c_H) + 1$. Notice that no minimal hitting set $X$ can contain all the vertices of $N[v_0]$, since if $X$ contains $N(v_0)$, then $v_0$ is an isolated clique. Hence, $c_H(X \cap V(H)) \leq c_H(V(H)) - 1 \leq \alpha \cdot \text{OPT}(H, c_H)$ and so $(H, c_H)$ is $\alpha$-good in $G$. We say that $(H, c_H)$ (sometimes simply $H$) is centrally $\alpha$-good (in $G$) with respect to $v_0$. Moreover, we call $v_0$ the root vertex.

In order to illustrate these ideas, consider the following two examples (see Fig. 1). First, let $H$ be a $C_4$ (that is, a 4-cycle) contained in $G$ and $1_H$ denote the unit cost function on $V(H)$. Then $(H, 1_H)$ is strongly 2-good, since $\sum_{v \in V(H)} 1_H(v) = 4 = 2 \text{OPT}(H, 1_H)$. Second, let $H$ be a $P_3$ contained in $G$, starting at a vertex $v_0$ that has degree-1 in $G$. Then $(H, 1_H)$ is centrally 2-good with respect to $v_0$, but it is not strongly 2-good.

Each time we find a 2-good weighted induced subgraph $H$ in $G$, the local ratio technique allows us to recurse on an induced subgraph $G'$ of $G$ in which at least one vertex of $H$ is deleted from $G$. For example, the 2-good induced subgraphs mentioned above allow us to reduce to input graphs $G$ that are $C_4$-free and have minimum degree at least 2.

The crux of our algorithm, Step 14, relies on the following structural result.

**Theorem 2** Let $G$ be a twin-free graph, let $v_0$ be any vertex of $G$, and let $H$ be the subgraph of $G$ induced by $N_{\leq 2}[v_0]$. There exists a cost function $c_H : V(H) \to \mathbb{Z}_{\geq 0}$ such that $(H, c_H)$ is either strongly 2-good, or centrally 2-good in $G$ with respect to $v_0$. Moreover, $c_H$ can be constructed in polynomial time.

We also study CLUSTER-VD from the polyhedral point of view. In particular we investigate how well linear programming (LP) relaxations can approximate the optimal value of CLUSTER-VD. As in [10, 14, 17], we use the following notion of LP relaxations which, by design, allows for extended formulations.

Fix a graph $G$. Let $d \in \mathbb{Z}_{\geq 0}$ be an arbitrary dimension. A system of linear inequalities $Ax \geq b$ in $\mathbb{R}^d$ defines an LP relaxation of CLUSTER-VD on $G$ if the following hold: (i) For every hitting set $X \subseteq V(G)$, we have a point $\pi^X \in \mathbb{R}^d$ satisfying $A\pi^X \geq b$; (ii) For every cost function $c : V(G) \to \mathbb{Q}_{\geq 0}$, we have an affine function $f_c : \mathbb{R}^d \to \mathbb{R}$; (iii) For all hitting sets $X \subseteq V(G)$ and cost functions $c : V(G) \to \mathbb{Q}_{\geq 0}$, the condition $f_c(\pi^X) = c(X)$ holds.

The size of the LP relaxation $Ax \geq b$ is defined as the number of rows of $A$. For every cost function $c$, the quantity $\text{LP}(G, c) := \min\{f_c(x) \mid Ax \geq b\}$ gives a lower
bound on $\text{OPT}(G, c)$. The integrality gap of the LP relaxation $Ax \geq b$ is defined as $\text{sup}\{\text{OPT}(G, c) / \text{LP}(G, c) \mid c \in \mathbb{Q}_{\geq 0}^{|V(G)|}\}$.

Letting $\mathcal{P}_3(G)$ denote the collection of all vertex sets $\{u, v, w\}$ that induce a $P_3$ in $G$, we define

$$P(G) := \{x \in [0, 1]^{|V(G)|} \mid \forall\{u, v, w\} \in \mathcal{P}_3(G) : x_u + x_v + x_w \geq 1\}.$$ 

We let $\text{SA}_r(G, c)$ denote the relaxation obtained from $P(G)$ by applying $r$ rounds of the Sherali-Adams hierarchy [34], a standard procedure to derive strengthened LP relaxations of binary linear programming problems. If a cost function $c : V(G) \to \mathbb{Q}_{\geq 0}$ is provided, we let

$$\text{SA}_r(G, c) := \min \left\{ \sum_{v \in V(G)} c(v)x_v \mid x \in \text{SA}_r(G) \right\}$$

denote the optimum value of the corresponding linear programming relaxation.

It is not hard to see that the straightforward LP relaxation $P(G)$ has worst case integrality gap equal to 3 (by worst case, we mean that we take the supremum over all graphs $G$). Indeed, for a random $n$-vertex graph, $\text{OPT}(G, 1_G) = n - O(\log^2 n)$ with high probability. This can be easily proved via the probabilistic method. A similar proof can be found, for instance, in the introduction of [2]. On the other hand, $\text{LP}(G, 1_G) \leq n/3$, since the vector with all coordinates equal to 1/3 is feasible for $P(G)$.

On the positive side, we show how applying one round of the Sherali-Adams hierarchy gives a relaxation with integrality gap at most $5/2 = 2.5$, see Theorem 3. To complement this, we prove that the worst case integrality gap of the relaxation is precisely $5/2$, see Theorem 4. We then show that the integrality gap decreases to $2 + \varepsilon$ after applying $\text{poly}(1/\varepsilon)$ rounds, see Theorem 5.

On the negative side, applying known results on VERTEX COVER [10], we show that no polynomial-size LP relaxation of CLUSTER-VD can have integrality gap at most $2 - \varepsilon$ for some $\varepsilon > 0$. As is the case for similar lower bounds (see [9,32]), this result is unconditional: it does not rely on $P \neq \text{NP}$ nor the Unique Games Conjecture.

1.2 Comparison to previous works

We now revisit all previous approximation algorithms for CLUSTER-VD [19,20,40]. The presentation given here slightly departs from [19,40], and explains in a unified manner what is the bottleneck in each of the algorithms.

Fix $k \in \{3, 4, 5\}$, and let $\alpha := (2k - 1)/(k - 1)$. Notice that $\alpha = 5/2$ if $k = 3$, $\alpha = 7/3$ if $k = 4$ and $\alpha = 9/4$ if $k = 5$. In [20, Lemma 3], it is shown that if a twin-free graph $G$ contains a $k$-clique, then one can find an induced subgraph $H$ containing the $k$-clique and a cost function $c_H$ such that $(H, c_H)$ is strongly $\alpha$-good.

Therefore, in order to derive an $\alpha$-approximation for CLUSTER-VD, one may assume without loss of generality that the input graph $G$ is twin-free and has no $k$-clique. Let $v_0$ be a maximum degree vertex in $G$, and let $H$ denote the subgraph of
$G$ induced by $N_{\leq 2}[v_0]$. In [20], it is shown by a tedious case analysis that one can construct a cost function $c_H$ such that $(H, c_H)$ is 2-good in $G$, using the fact that $G$ has no $k$-clique.

The simplest case occurs when $k = 3$. Then $N(v_0)$ is a stable set. Letting $c_H(v_0) := d(v_0) - 1$, $c_H(v) := 1$ for $v \in N(v_0)$ and $c_H(v) := 0$ for the other vertices of $H$, one easily sees that $(H, c_H)$ is (centrally) 2-good in $G$. For higher values of $k$, one has to work harder.

In this paper, we show that one can always, and in polynomial time, construct a cost function $c_H$ on the vertices at distance at most 2 from $v_0$ that makes $(H, c_H)$ 2-good in $G$, provided that $G$ is twin-free, see Theorem 2. This result was the main missing ingredient in previous approaches, and single-handedly closes the approximability status of Cluster-VD.

1.3 Other related works

Cluster-VD has also been widely studied from the perspective of fixed parameter tractability. Given a graph $G$ and parameter $k$ as input, the task is to decide if $G$ has a hitting set $X$ of size at most $k$. A $2^k n^{O(1)}$-time algorithm for this problem was given by Hüffner, Komusiewicz, Moser, and Niedermeier [25]. This was subsequently improved to a $1.911^k n^{O(1)}$-time algorithm by Boral, Cygan, Kociumaka, and Pilipczuk [12], and a $1.811^k n^{O(1)}$-time algorithm by Tsur [38]. By the general framework of Fomin, Gaspers, Lokshtanov, and Saurabh [21], these parametrized algorithms can be transformed into exponential algorithms which compute the size of a minimum hitting set for $G$ exactly, the fastest of which runs in time $O(1.488^n)$.

For polyhedral results, Hosseinian and Butenko [24] gives some facet-defining inequalities of the Cluster-VD polytope, as well as complete linear descriptions for special classes of graphs.

Another related problem is the feedback vertex set problem in tournaments (FVST). Given a tournament $T$ with costs on the vertices, the task is to find a minimum cost set of vertices $X$ such that $T - X$ does not contain a directed cycle.

For unit costs, note that Cluster-VD is equivalent to the problem of deleting as few elements as possible from a symmetric relation to obtain a transitive relation, while FVST is equivalent to the problem of deleting as few elements as possible from an antisymmetric and complete relation to obtain a transitive relation.

In a tournament, hitting all directed cycles is equivalent to hitting all directed triangles, so FVST is also a hitting set problem in a 3-uniform hypergraph. Moreover, FVST is also UGC-hard to approximate to a constant factor smaller than 2. Cai, Deng, and Zang [15] gave a $5/2$-approximation algorithm for FVST, which was later improved to a $7/3$-approximation algorithm by Mnich, Williams, and Végh [31]. Lokshtanov, Misra, Mukherjee, Panolan, Philip, and Saurabh [30] recently gave a randomized 2-approximation algorithm, but no deterministic (polynomial-time) 2-approximation algorithm is known. For FVST, one round of the Sherali-Adams hierarchy actually provides a $7/3$-approximation [6].

Among other related covering and packing problems, Fomin, Le, Lokshtanov, Saurabh, Thomassé, and Zehavi [22] studied both Cluster-VD and FVST from
the kernelization perspective. They proved that the unweighted versions of both problems admit subquadratic kernels: $O(k^{5/3})$ for CLUSTER-VD and $O(k^{1/2})$ for FVST.

1.4 Overview of the proof

We give a sketch of the proof of Theorem 2. Recall that $H = G[N_{\leq 2}[v_0]]$. If the subgraph induced by $N(v_0)$ contains a hole (that is, an induced cycle of length at least 4), then $H$ is strongly 2-good by Lemma 1. If the subgraph induced by $N(v_0)$ contains an induced $2P_3$ (that is, two disjoint copies of $P_3$ with no edges between each other), then $H$ is strongly 2-good by Lemma 2. This allows us to reduce to the case where the subgraph induced by $N(v_0)$ is chordal and $2P_3$-free.

Lemma 5 then gives a direct construction of a cost function $c_H$ which certifies that $(H, c_H)$ is centrally 2-good, provided that the subgraph induced by $N[v_0]$ is twin-free. This is the crucial step of the proof. It serves as the base case of the induction. Here, we use a slick observation due to Lokshtanov [29]: since the subgraph induced by $N(v_0)$ is chordal and $2P_3$-free, it has a hitting set that is a clique. In a previous version, our proof of Theorem 2 was slightly more complicated.

We show inductively that we can reduce to the case where the subgraph induced by $N[v_0]$ is twin-free. The idea is to delete vertices from $H$ to obtain a smaller graph $H'$, while preserving certain properties, and then compute a suitable cost function $c_H$ for $H$, given a suitable cost function $c_{H'}$ for $H'$. We delete vertices at distance 2 from $v_0$. When this creates twins in $H$, we delete one vertex from each pair of twins. At the end, we obtain a twin-free induced subgraph of $H[N[v_0]]$, which corresponds to our base case.

We conclude the introduction with a brief description of the different sections of the paper. Section 2 is entirely devoted to the proof of Theorem 2. The proof of Theorem 1 is given in Sect. 3, together with a complexity analysis of Algorithm 1. Section 4 presents our polyhedral results. A conclusion is given in Sect. 5. There, we state a few open problems for future research.

Conference version. An abstract of this paper appeared in the proceedings of IPCO 2021 [7]. The current paper is an extended version with full proofs and detailed discussions. In particular, the running time analysis of our algorithm in Sect. 3 is new, and the polyhedral results of Sect. 4 appear without proof in the conference version.

2 Finding 2-good induced subgraphs

The goal of this section is to prove Theorem 2. Our proof is by induction on the number of vertices in $H := G[N_{\leq 2}[v_0]]$. First, we quickly show that we can assume that the subgraph induced by $N(v_0)$ is chordal and $2P_3$-free. Using this, we prove the theorem in the particular case where the subgraph induced by $N[v_0]$ is twin-free. Finally, we prove the theorem in the general case by showing how to deal with twins.
2.1 Restricting to chordal, 2P₃-free neighborhoods

A vertex of G is apex if it is adjacent to all the other vertices of G. A wheel is a graph obtained from a cycle by adding an apex vertex (called the center).

As pointed out earlier in the introduction, 4-cycles are strongly 2-good. This implies that the wheel on 5 vertices is strongly 2-good (putting a zero cost on the center). We now show that all wheels on at least 5 vertices are strongly 2-good. This allows our algorithm to restrict to input graphs such that the subgraph induced on each neighborhood is chordal. In a similar way, we show that we can further restrict such neighborhoods to be 2P₃-free.

Lemma 1 Let H := Wₖ be a wheel on k ≥ 5 vertices and center v₀, let c_H(v₀) := k−5 and c_H(v) := 1 for v ∈ V(H − v₀). Then (H, c_H) is strongly 2-good.

Proof Notice that OPT(H, c_H) ≥ k − 3 since a hitting set either contains v₀ and at least 2 more vertices, or does not contain v₀ but contains k − 3 other vertices. Hence, ∑v∈V(H) c_H(v) = k − 5 + k − 1 = 2(k − 3) ≤ 2OPT(H, c_H).

Lemma 2 Let H be the graph obtained from 2P₃ by adding an apex vertex v₀. Let c_H(v₀) := 2 and c_H(v) := 1 for v ∈ V(H − v₀). Then (H, c_H) is strongly 2-good.

Proof It is easy to check that OPT(H, c_H) ≥ 4. Thus, ∑v∈V(H) c_H(v) = 8 ≤ 2OPT(H, c_H).

2.2 When H is twin-free

Throughout this section, we assume that H is a twin-free graph with an apex vertex v₀ such that H − v₀ is chordal and 2P₃-free. Our goal is to construct a cost function c_H that certifies that H is centrally 2-good.

It turns out to be easier to define the cost function on V(H − v₀) = N(v₀) first, and then adjust the cost of v₀. This is the purpose of the next lemma. Below, ω(G, c) denotes the maximum weight of a clique in weighted graph (G, c).

Lemma 3 Let H be a graph with an apex vertex v₀ and H′ := H − v₀. Let c_H′ : V(H′) → ℤ≥1 be a cost function such that
(i) c_H′(V(H′)) ≥ 2ω(H′, c_H′) and
(ii) OPT(H′, c_H′) ≥ ω(H′, c_H′) − 1.

Then we can extend c_H′ to a function c_H : V(H) −> ℤ≥1 such that c_H(V(H)) ≤ 2OPT(H, c_H) + 1. In other words, (H, c_H) is centrally 2-good with respect to v₀.

Proof Notice that

OPT(H, c_H) = min(c_H(v₀) + OPT(H′, c_H′), c_H′(V(H′)) − ω(H′, c_H′)),

since if X is hitting set of H that does not contain v₀, then H − X is a clique.
Let $a = \max(1, c_{H'}(V(H')) - 2\OPT(H', c_{H'}) - 1)$ and $b = c_{H'}(V(H')) - 2\omega(H', c_{H'}) + 1$. Choose $c_{H}(v_0) \in \mathbb{Z}_{\geq 1}$ such that $a \leq c_{H}(v_0) \leq b$. Note that $c_{H}(v_0)$ exists since $a \leq b$ by conditions (i) and (ii).

Suppose $\OPT(H, c_{H}) = c_{H}(v_0) + \OPT(H', c_{H'})$. Since $a \leq c_{H}(v_0)$,

$$c_{H}(V(H)) \leq 2c_{H}(v_0) + 2\OPT(H', c_{H'}) + 1 = 2\OPT(H, c_{H}) + 1.$$ 

Suppose $\OPT(H, c_{H}) = c_{H}(V(H')) - \omega(H', c_{H'})$. Since $c_{H}(v_0) \leq b$,

$$c_{H}(V(H)) \leq 2c_{H'}(V(H')) - 2\omega(H', c_{H'}) + 1 = 2\OPT(H, c_{H}) + 1.$$ 

In either case, $c_{H}(V(H)) \leq 2\OPT(H, c_{H}) + 1$, as required. \hfill \qed

We abuse notation and regard a clique $X$ of a graph as both a set of vertices and a subgraph. We call a hitting set $X$ of a graph $G$ a hitting clique if $X$ is also a clique.

**Lemma 4** Every chordal, $2P_3$-free graph contains a hitting clique.

**Proof** Let $G$ be a chordal, $2P_3$-free graph. Since $G$ is chordal, $G$ admits a clique tree $T$ (see [11]). In $T$, the vertices are the maximal cliques of $G$ and, for every two maximal cliques $K$, $K'$, the intersection $K \cap K'$ is contained in every clique of the $K$–$K'$ path in $T$. For an edge $e := KK'$ of $T$, let $T_1$ and $T_2$ be the components of $T - e$ and $G_1$ and $G_2$ be the subgraphs of $G$ induced by the union of all the cliques in $T_1$ and $T_2$, respectively. It is easy to see that deleting $K \cap K'$ separates $G_1$ from $G_2$ in $G$. Now, since $G$ is $2P_3$-free, at least one of $G'_1 := G_1 - (K \cap K')$ or $G'_2 := G_2 - (K \cap K')$ is a cluster graph. If both $G'_1$ and $G'_2$ are cluster graphs, we are done since $K \cap K'$ is the desired hitting clique. Otherwise, if $G'_1$ is not a cluster graph, then we can orient $e$ towards $T_1$. Applying this argument on each edge, we define an orientation of $T$, which must have a sink $K_0$. But then removing $K_0$ from $G$ leaves a cluster graph, and we are done. Since the clique tree of a chordal graph can be constructed in polynomial time [11], the hitting clique can be found in polynomial time. \hfill \qed

We are ready to prove the base case for Theorem 2. For a graph $H$, we let $|H|$ denote the number of vertices of $H$.

**Lemma 5** Let $H$ be a twin-free graph with an apex vertex $v_0$ such that $H - v_0$ is chordal and $2P_3$-free. There exists a cost function $c_{H}$ such that $(H, c_{H})$ is centrally 2-good with respect to $v_0$. Moreover, $c_{H}$ can be found in time $O(|H|^3)$.

**Proof** By Lemma 4, some maximal clique of $H - v_0$, say $K_0$, is a hitting set.

We claim that there is a family of stable sets $S = \{S_v \mid v \in K_0\}$ of $H - v_0$ satisfying the following properties:

(P1) every vertex of $H - v_0$ is contained in some $S_v$;
(P2) for each $v \in K_0$, $S_v$ contains $v$ and at least one other vertex;
(P3) for every two distinct vertices $v, v' \in K_0$, $H[S_v \cup S_{v'}]$ contains a $P_3$. 

© Springer
Before proving the claim, we prove that it implies the lemma. Consider the cost function \( c_{H'} := \sum_{v \in K_0} x_{S_v} \) on the vertices of \( H' := H - v_0 \) defined by giving to each vertex \( u \) a cost equal to the number of stable sets \( S_v \) that contain \( u \) (see Fig. 2). It suffices to show that \( c_{H'} \) satisfies the conditions of Lemma 3 and can therefore be extended to a cost function \( c_H \) on \( V(H) \) such that \( (H, c_H) \) is centrally 2-good with respect to \( v_0 \).

First, by (P1), we have \( c_{H'}(u) \in \mathbb{Z}_{\geq 1} \) for all \( u \in V(H') \). Second, condition (i) of Lemma 3 follows from (P2) since each stable set \( S_v \) contributes at least two units to \( c_{H'}(V(H')) \) and at most one unit to \( \omega(H', c_{H'}) \). Third, (P3) implies that every hitting set of \( H' \) meets every stable set \( S_v \), except possibly one. Hence, \( \text{OPT}(H', c_{H'}) \geq |K_0| - 1 \). Also, every clique of \( H' \) meets every stable set \( S_v \) in at most one vertex, implying that \( \omega(H', c_{H'}) \leq |K_0| \), and equality holds since \( c_{H'}(K_0) = |K_0| \). Putting the last two observations together, we see that \( \text{OPT}(H', c_{H'}) \geq |K_0| - 1 = \omega(H', c_{H'}) - 1 \) and hence condition (ii) of Lemma 3 holds.

Now, we prove that our claim holds. Let \( K_1, \ldots, K_t \) denote the clusters (maximal cliques) of cluster graph \( H - v_0 - K_0 \). For \( i \in [t] \), consider the submatrix \( A_i \) of the adjacency matrix \( A(H) \) with rows indexed by the vertices of \( K_0 \) and columns indexed by the vertices of \( K_i \).

Notice that \( A_i \) contains neither \( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \) nor \( \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \) as a submatrix, as this would give a \( C_4 \) contained in \( H - v_0 \), contradicting the chordality of \( H - v_0 \). Hence, after permuting its rows and columns if necessary, \( A_i \) can be assumed to be staircase-shaped. That is, every row of \( A_i \) is nonincreasing and every column nondecreasing. Notice also that \( A_i \) does not have two equal columns, since these would correspond to two vertices of \( K_i \) that are twins.

For each \( K_i \) that is not complete to \( v \in K_0 \) (i.e., not all the edges between \( v \) and a vertex of \( K_i \) are present), define \( \varphi_i(v) \) as the vertex \( u \in K_i \) whose corresponding column in \( A_i \) is the first containing a 0 in row \( v \). Now, for each \( v \), let \( S_v \) be the set including \( v \), and \( \varphi_i(v) \), for each \( K_i \) that is not complete to \( v \).

Because \( K_0 \) is maximal, no vertex \( u \in K_i \) is complete to \( K_0 \). Since no two columns of \( A_i \) are identical, we must have \( u = \varphi_i(v) \) for some \( v \in K_0 \). This proves (P1).

Notice that \( v \in S_v \) by construction and that \( |S_v| \geq 2 \) since otherwise, \( v \) would be apex in \( H \) and thus a twin of \( v_0 \). Hence, (P2) holds.
Finally, consider any two distinct vertices \( v, v' \in K_0 \). Since \( v, v' \) are not twins, the edge \( vv' \) must be in a \( P_3 \) contained in \( H - v_0 \). Assume, without loss of generality, that there is a vertex \( u \in K_i \) adjacent to \( v \) and not to \( v' \) for some \( i \in \{ t \} \). Then \{ \( v, v', \varphi_i(v') \) \} induces a \( P_3 \) contained in \( H[S_v \cup S_{v'}] \), proving (P3). This concludes the proof of the claim.

We observe that the collection \( S \) can be computed in \( O(|H|^3) \) time. This yields the restriction \( c_H \) to \( H' \). Since \( H' \) is chordal, \( \omega(H', c_{H'}) \) can be computed in \( O(|H'|^2) \) time. We then just let \( c_H(v_0) := c_{H'}(V(H')) - 2\omega(H', c_{H'}) + 1 = c_{H'}(V(H')) - 2\sum_{i=0}^{k} (K_i) + 1 \). This sets \( c_H(v_0) \) equal to the value \( b \) in the proof of Lemma 3. Therefore, \( c_H \) can be constructed in \( O(|H|^3) \) time.

\[ \square \]

### 2.3 Handling twins in \( G[N[v_0]] \)

We now deal with the general case where \( G[N[v_0]] \) contains twins. We start with an extra bit of terminology relative to twins. Let \( G \) be a twin-free graph, and \( v_0 \in V(G) \). Suppose that \( u, u' \) are twins in \( G[N[v_0]] \). Since \( G \) is twin-free, there exists a vertex \( v \) that is adjacent to exactly one of \( u, u' \) in \( G \). We say that \( v \) is a distinguisher for the edge \( uu' \) (or for the pair \( \{u, u'\} \)). Notice that either \( uu'v \) or \( uu'v \) is an induced \( P_3 \). Notice also that \( v \) is at distance 2 from \( v_0 \).

Now, consider a graph \( H \) with a special vertex \( v_0 \in V(H) \) (the root vertex) such that

- (H1) every vertex is at distance at most 2 from \( v_0 \), and
- (H2) every pair of vertices that are twins in \( H[N[v_0]] \) has a distinguisher.

Let \( v \) be any vertex that is at distance 2 from \( v_0 \). Consider the equivalence relation \( u \equiv u' \) whenever \( u = u' \) or \( u, u' \) are twins in \( H - v \). Observe that the equivalence classes of \( \equiv \) are of size at most 2 since, if \( u, u', u'' \) are distinct vertices with \( u \equiv u' \equiv u'' \), then \( v \) cannot distinguish every edge of the triangle on \( u, u' \) and \( u'' \). Hence, two of these vertices are twins in \( H \), which contradicts (H2).

From what precedes, the edges contained in \( N[v_0] \) that do not have a distinguisher in \( H-v \) form a matching \( M := \{u_1u'_1, \ldots, u_ku'_k\} \) (possibly, \( k = 0 \)). Let \( H' \) denote the graph obtained from \( H \) by deleting \( v \) and exactly one endpoint from each edge of \( M \). Notice that the resulting subgraph is the same, up to isomorphism, no matter which endpoints are chosen.

The lemma below states how we can obtain a cost function \( c_H \) that certifies that \( H \) is centrally 2-good from a cost function \( c_{H'} \) that certifies that \( H' \) is centrally 2-good. It is inspired by [20, Lemma 3]. See Fig. 3 for an example.

**Lemma 6** Let \( H \) be any graph satisfying (H1) and (H2) for some \( v_0 \in V(H) \). Let \( v \in N_2(v_0) \). Let \( M := \{u_1u'_1, \ldots, u_ku'_k\} \) be the matching formed by the edges in \( N[v_0] \) whose unique distinguisher is \( v \), where \( u'_i \neq v_0 \) for all \( i \) (we allow the case \( k = 0 \)). Let \( H' := H - u'_1 - \cdots - u'_k - v \). Given a cost function \( c_{H'} \) on \( V(H') \), define a cost function \( c_H \) on \( V(H) \) by letting \( c_H(u'_i) := c_{H'}(u'_i) \) for \( i \in [k] \), \( c_H(v) := \sum_{i=1}^{k} c_{H'}(u'_i) = \sum_{i=1}^{k} c_H(u'_i) \), and \( c_H(u) := c_{H'}(u) \) otherwise. First, \( H' \) satisfies (H1) and (H2). Second, if \( (H', c_{H'}) \) is centrally 2-good, then \( (H, c_H) \) is centrally 2-good.
Fig. 3 Here $v_0$ is the gray vertex and $v$ is the red vertex. $H - v$ violates (H2), and contains two pairs of twins, indicated by the red edges. Lemma 6 applies. We see that $H'$ is a $P_3$, for which Lemma 5 gives $c_{H'} = 1$. In $(H, c_H)$, all vertices get a unit cost except $v$, which gets a cost of 2, since there are 2 pairs of twins in $H - v$. Thus, $c_H = (1, 2, 1, 1, 1, 1)$, where the entries corresponding to $v_0$ and $v$ are bold and red, respectively.

**Proof** For the first part, notice that $H'$ satisfies (H1) by our choice of $v$. Indeed, deleting $v$ does not change the distance of the remaining vertices from $v_0$.

We now prove that $H'$ satisfies (H2). First notice that in $H - v$ the twins in $N[v_0]$ are exactly $(u_1, u'_1), \ldots, (u_k, u'_k)$. Next, for each edge $e \in E(H[N[v_0]])$, $e$ has at least one distinguisher different than $v$ unless $e \in M$. Moreover, each $u'_i$ is a distinguisher for $e$ if and only if $u_i$ is a distinguisher for $e$. Thus, each edge of $H'[N[v_0]]$ still has at least one distinguisher, which proves (H2).

For the second part, notice that $c_H(u) \geq 1$ for all $u \in N[v_0]$ since $c_{H'}(u) \geq 1$ for all $u \in N[v_0] \setminus \{v, u'_1, \ldots, u'_k\}$. To argue that $c_H(V(H)) \leq 2 \text{OPT}(H, c_H) + 1$, consider a minimal hitting set $X$ of $H$. Then $X$ must either contain $v$, or at least one endpoint of each edge $u_i, u'_i \in M$: but one checks that, in the latter case, $X$ has to contain both $u_i, u'_i$, since $u_i, u'_i$ are twins. Hence $\text{OPT}(H, c_H) \geq \sum_{i=1}^{k} c_{H'}(u_i) + \text{OPT}(H', c_{H'})$.

Since $(H', c_{H'})$ is centrally 2-good, $c_{H'}(V(H')) \leq 2 \text{OPT}(H', c_{H'}) + 1$. It follows that

$$c_H(V(H)) = c_H(v) + \sum_{i=1}^{k} c_H(u'_i) + c_{H'}(V(H')) \leq 2 \text{OPT}(H, c_H) + 1.$$

\[\square\]

### 2.4 Proof of Theorem 2

We are ready to prove Theorem 2.

**Proof** We can decide in polynomial time (see for instance [36]) if $H[N(v_0)]$ is chordal, and if not, output a hole of $H[N(v_0)]$. If the latter holds, we are done by Lemma 1. If the former holds, we can decide in polynomial time (see [35], and the proof of Lemma 4) whether $H$ contains a $2P_3$. If it does, we are done by Lemma 2.
From now on, assume that the subgraph induced by \( N(v_0) \) is chordal and 2\( P_3 \)-free. This is done without loss of generality. Notice that hypotheses (H1) and (H2) from Sect. 2.3 hold for \( H \). This is obvious for (H1). To see why (H2) holds, remember that \( G \) is twin-free. Hence, every edge \( uu' \) contained in \( N[v_0] \) must have a distinguisher in \( G \), which is in \( N_{\leq 2}[v_0] \). (In fact, notice that if \( u \) and \( u' \) are twins in \( H[N[v_0]] \) then the distinguisher is necessarily in \( N_2(v_0) \).)

We repeatedly apply Lemma 6 in order to delete each vertex of \( N_2(v_0) \) one after the other and reduce to the case where \( H \) is a twin-free graph for which \( v_0 \) is apex. We can then apply Lemma 5. The whole process takes polynomial time.  

\[ \square \]

### 3 Running-time analysis

We now analyse the running-time of Algorithm 1. We assume that input graphs are given by their adjacency matrix. We need the following easy lemma, whose proof we include for completeness.

**Lemma 7** Given a matrix \( N \in \{0, 1\}^r \times c \), the set of all equivalence classes of equal rows of \( N \) can be found in time \( O(rc) \).

**Proof** Let \( R_0 \) and \( R_1 \) be the set of rows of \( N \) whose first entry is 0 and 1, respectively. We can determine \( R_0 \) and \( R_1 \) by reading the first column of \( N \), which takes time \( O(r) \). We then recurse on \( N'_0 \) and \( N'_1 \), where \( N'_i \) is the submatrix of \( N \) induced by \( R_i \) and the last \( c-1 \) columns of \( N \).  

Before proving the next lemma we remark that, given a graph \( H \) with \( n \) vertices and \( m \) edges, one can check whether \( H \) is a cluster graph by checking that each of its components is a clique, which takes \( O(n^2) \) time.

**Lemma 8** Let \( G \) be an \( n \)-vertex, twin-free graph. In \( O(n^3) \) time, we can find an induced subgraph \( H \) of \( G \) and a cost function \( c_H \) on \( V(H) \) such that \( (H, c_H) \) is 2-good.

**Proof** We fix any vertex \( v_0 \in V(G) \), and let \( H = G[N_{\leq 2}[v_0]] \). We can check in \( O(n^2) \) time whether \( H[N(v_0)] \) is chordal by using the algorithm from [36]. If \( H[N(v_0)] \) is not chordal this algorithm returns, as a certificate, a hole \( C \). By Lemma 1, \( H[V(C) + v_0] \) is strongly 2-good and the corresponding function \( c_H \) can be computed straightforwardly, hence we are done. Suppose now that \( H[N(v_0)] \) is chordal. We can construct the clique tree of \( H[N(v_0)] \) (see for instance [35]) in \( O(n^2) \) time. Each edge of the tree induces a separation of \( H[N(v_0)] \), and we can check if each side is a cluster graph in \( O(n^2) \) time. If neither side is a cluster graph, then we have found a 2\( P_3 \) in \( H[N(v_0)] \). Hence, \( H[V(2P_3) + v_0] \) is strongly 2-good and the corresponding function \( c_H \) can be computed straightforwardly. Since the clique tree has at most \( |H| \leq n \) vertices, by orienting its edges as in the proof of Lemma 4 we find, in \( O(n^3) \) time, a hitting clique. By applying Lemma 6 to get rid of twins, which can be done in \( O(n^2) \) time, we obtain a subgraph satisfying the hypotheses of Lemma 5. Finally, by Lemma 5, the cost function in the statement of Lemma 5 can be constructed in \( O(n^3) \) time.  

\[ \square \]

**Lemma 9** Algorithm 1 runs in \( O(n^4) \)-time.
For all \( r \in \mathbb{N} \), we define a polytope \( P \supseteq \mathcal{S}A_1(P) \supseteq \cdots \supseteq \mathcal{S}A_r(P) \supseteq P_1 \) as follows. Let \( N_r \) be the \emph{nonlinear} system obtained from \( P \) by multiplying each constraint by \( \prod_{i \in I} x_i \prod_{j \in J}(1 - x_j) \) for all disjoint subsets \( I, J \) of \([n]\) such that \( 1 \leq |I| + |J| \leq r \). Note that if \( x_i \in \{0, 1\} \), then \( x_i^2 = x_i \). Therefore, we can obtain a \emph{linear} system \( L_r \) from \( N_r \) by setting \( x_i^2 := x_i \) for all \( i \in [n] \) and then \( x_i := \prod_{j \in J} x_i \) for all \( I \subseteq [n] \) with \( |I| \geq 2 \). We then let \( \mathcal{S}A_r(P) \) be the projection of \( L_r \) onto the variables \( x_i, i \in [n] \).

Let \( \mathcal{P}_3(G) \) denote the collection of all vertex sets \( \{u, v, w\} \) that induce a \( P_3 \) in \( G \) and let \( \mathcal{S}A_r(G) := \mathcal{S}A_r(P(G)) \), where

\[
P(G) := \{ x \in [0, 1]^{|V(G)|} \mid \forall \{u, v, w\} \in \mathcal{P}_3(G) : x_u + x_v + x_w \geq 1 \}.
\]

If a cost function \( c : V(G) \to \mathbb{R}_{\geq 0} \) is provided, we let

\[
\mathcal{S}A_r(G, c) := \min \left\{ \sum_{v \in V(G)} c(v)x_v \mid x \in \mathcal{S}A_r(G) \right\}
\]

denote the optimum value of the corresponding linear programming relaxation. For the sake of simplicity, we sometimes denote by \( \mathcal{S}A_r(G, c) \) the above linear program itself.
We say vertices $a$ and $b$ form a diagonal if there are vertices $u$, $v$ such that $\{u, v, a\} \in \mathcal{P}_3(G)$ and $\{u, v, b\} \in \mathcal{P}_3(G)$. We say that a path contains a diagonal if any of its pairs of vertices are diagonals. Note that a diagonal pair in a path need not be an edge in the path.

Our first results concern $\text{SA}_1(G)$. For later use, we list here the inequalities defining $\text{SA}_1(G)$. For all $\{u, v, w\} \in \mathcal{P}_3(G)$ and $z \in V(G - u - v - w)$, we have the inequalities

\[
x_u + x_v + x_w \geq 1 + x_{uv} + x_{uw}, \tag{1}
\]
\[
x_{uz} + x_{vz} + x_{wz} \geq x_z \quad \text{and} \quad \tag{2}
\]
\[
x_u + x_v + x_w + x_z \geq 1 + x_{uz} + x_{vz} + x_{wz}. \tag{3}
\]

In addition, there are the inequalities

\[
1 \geq x_v \geq x_{vu} \geq 0 \tag{4}
\]

for all distinct $u$, $v \in V(G)$. The polytope $\text{SA}_1(G)$ is the set of all $(x_v)$ such that there exists $(x_{uv})$ such that inequalities (1)–(4) are satisfied. Note that by definition, $x_{uv}$ and $x_{vu}$ are the same variable.

In order to establish the integrality gap of $\text{SA}_1(G)$, we need two preliminary lemmas.

**Lemma 10** Let $x \in \text{SA}_1(G)$. If $G$ contains a $P_3$ which has a diagonal, then $x_v \geq 2/5$ for some vertex $v$ of $G$.

**Proof** Assume by contradiction that $a$, $b$ form a diagonal and all components of $x$ are less than $2/5$. By the definition of diagonal, there exist $u$, $v \in V(G)$ with $\{u, v, a\}, \{u, v, b\} \in \mathcal{P}_3(G)$. In particular, from (1) we have $x_a + x_u + x_v \geq 1 + x_{au} + x_{av}$ and from (2) $x_{ab} + x_{au} + x_{av} \geq x_a$.

Adding these two inequalities, we obtain $x_a + x_u + x_{ab} \geq 1$. We must have $x_{ab} \geq 1/5$ since otherwise max$(x_a, x_v) \geq 2/5$. Now let $c$ be the third vertex of the $P_3$ containing $a$, $b$ (notice that $c$ can be the middle vertex of the $P_3$). By (1) and (4), we have $x_a + x_b + x_c \geq 1 + x_{ab} + x_{ac} \geq 6/5$ which means that max$(x_a, x_b, x_c) \geq 2/5$, a contradiction which concludes the proof. $\square$

**Lemma 11** Let $G$ be a path or a cycle, and $c : V(G) \rightarrow \mathbb{R}_{\geq 0}$. Then:

(i) there is an efficient algorithm that solves $\text{CLUSTER-VD}$ for $(G, c)$;

(ii) the basic LP $\text{SA}_0(G)$ has integrality gap at most $2$.

**Proof** First, let $G$ be a path. We notice that the coefficient matrix of the basic LP is totally unimodular, by the consecutive ones property [33]. Hence solving the LP yields an integral optimal solution, which corresponds to a hitting set $X$ of $G$ of cost equal to $\text{OPT}(G, c) = \text{SA}_0(G, c)$. This proves (i), (ii) when $G$ is a path.

Now, let $G$ be a cycle, and let $v \in V(G)$. Suppose that $v$ belongs to a minimum cost hitting set $X$ of $G$. Then $X \setminus \{v\}$ is a minimum cost hitting set of $G - v$, hence it can be found efficiently since $G - v$ is a path. By iterating over all $v \in V(G)$ and taking the hitting set of minimum cost, we efficiently solve $\text{CLUSTER-VD}$ for $(G, c)$. This concludes the proof of (i).
Finally, let $\tilde{x}$ be an optimal solution of $SA_0(G, c)$. First, assume that there is some $v \in V(G)$ such that $\tilde{x}_v \geq 1/2$. Since $G - v$ is a path, the optimal hitting set $X'$ in $G - v$ has cost $c(X') \leq \sum_{u \neq v} c(u)\tilde{x}_u$. Hence, we see that $X := X' + v$ is a hitting set of $H$ with $c(X) = c(v) + c(X') \leq (c(v) + \sum_{u \neq v} c(u)\tilde{x}_u) \leq 2 \sum_u c(u)\tilde{x}_u = 2SA_0(G, c)$. On the other hand, if $\tilde{x}_v < 1/2$ for all vertices $v \in V(G)$, then the constraint $\tilde{x}_u + \tilde{x}_v + \tilde{x}_w \geq 1$ (where $u, w$ are the neighbors of $v$) implies that there can be no vertex $v$ with $\tilde{x}_v = 0$. So $0 < \tilde{x}_v < 1/2$ for all $v \in V(G)$. Therefore, extreme point $\tilde{x}$ is the unique solution of $|G|$ equations of the form $x_u + x_v + x_w = 1$ for $\{u, v, w\} \in P_3(G)$. Hence $\tilde{x}_v = 1/3$ for all vertices. Thus $SA_0(G, c) = 1/3 \cdot c(V(G))$. Now notice that since $G$ is a cycle we can partition its vertices into two disjoint hitting sets $X$ and $Y$. Without loss of generality assume that $c(X) \leq 1/2 \cdot c(V(G))$. Then $c(X) \leq 3/2 \cdot SA_0(G, c)$. This concludes the proof of (ii).

**Theorem 3** There is a polynomial-time algorithm that, given a graph $G$ and $c : V(G) \to \mathbb{R}_{\geq 0}$, outputs a hitting set $X$ of $G$ such that $c(X) \leq 5/2 \cdot SA_1(G, c)$. In particular, the integrality gap of $SA_1(G)$ is at most 5/2.

**Proof** Let $\bar{x}$ be an optimal solution of $SA_1(G, c)$. Let $U = \{v \in V(G) : \bar{x}_v \geq 2/5\}$, and $H = G \setminus U$. Notice that the restriction of $\bar{x}$ to $V(H)$ is a feasible solution for $SA_1(H, c)$ whose components are all strictly less than 2/5. Hence, by Lemma 10, $H$ cannot contain a $P_3$ which has a diagonal. We will now show that, after possibly getting rid of twins, $H$ is a disjoint union of paths and cycles. Then, by applying Lemma 11, we will obtain a minimum cost hitting set of $H$, which, together with $U$, will form our desired hitting set.

First, if $H$ contains twins $u$ and $v$, we can delete $v$, and set $c'(u) := c(u) + c(v)$, $c'(w) := c(w)$ for $w \in V(H) \setminus \{u, v\}$ to obtain a smaller weighted graph $(H', c')$. We claim that $SA_1(H', c') \leq SA_1(H, c)$. To see this, we show that one can turn any feasible solution $x$ of $SA_1(H, c)$ into a feasible solution of $SA_1(H', c')$ without increasing the cost. Let $x'_u := \min(x_u, x_v)$ and $x'_w := x_w$ for $w \neq u, v$. It is easy to check that this defines a feasible solution $x'$ to $SA_1(H', c')$, whose cost is

$$\sum_{w \neq v} c'(w)x'_w = (c(u) + c(v))\min(x_u, x_v) + \sum_{w \neq u, v} c(w)x_w \leq \sum_w c(w)x_w.$$  

This proves the claim. Moreover, a hitting set $X$ for $H$ can be immediately obtained from a hitting set $X'$ of $H'$ by adding $v$ if and only if $u \in X'$. Finally, notice that there is a feasible solution for $SA_1(H', c')$ obtained from the restriction of $\bar{x}$ whose components are all strictly less than 2/5. Hence, from now on we assume that $H$ is twin-free.

Now, we claim that $H$ is triangle-free and claw-free. Suppose that $H$ contains a triangle with vertices $u, v$ and $w$. Since $H$ is twin-free, every edge of the triangle has a distinguisher. Without loss of generality, $P_3(H)$ contains $\{u, v, y\}$, $\{u, w, y\}$, $\{v, w, z\}$ and $\{u, v, z\}$ where $y, z$ are distinct vertices outside the triangle. It is easy to see that, for instance, edge $uw$ is a diagonal contained in a $P_3$, a contradiction. A similar argument shows that $H$ cannot contain a claw. This proves the claim, and implies that $H$ has maximum degree at most 2. That is, $H$ is a disjoint union of
paths and cycles. By part (i) of Lemma 11, (applied to each component of \( H \)), we obtain a minimum cost hitting set \( X \) of \( H \) which, thanks to (ii) of Lemma 11, satisfies 
\[
c(X) \leq 2 \text{SA}_0(H, c) \leq \frac{5}{2} \text{SA}_1(H, c).
\]

Finally, consider \( X \cup U \). Clearly, it is a hitting set of \( G \). Moreover, one has 
\[
c(X \cup U) = c(X) + c(U) \leq \frac{5}{2} \left( \text{SA}_1(H, c) + \frac{2}{5}c(U) \right)
\]
\[
\leq \frac{5}{2} \left( \text{SA}_1(G, c) - \sum_{u \in U} c(u)\bar{x}_u + \frac{2}{5}c(U) \right) \leq \frac{5}{2} \cdot \text{SA}_1(G, c).
\]

It follows that the integrality gap of \( \text{SA}_1(G) \) is at most \( 5/2 \), concluding the proof.

We now complement the result above by showing a lower bound on the integrality gap of \( \text{SA}_1(G, c) \).

**Theorem 4** For every \( \epsilon > 0 \) there is some instance \((G, c)\) of CLUSTER-VD such that \( \text{OPT}(G, c) \geq (5/2 - \epsilon) \text{SA}_1(G, c) \).

**Proof** We show there is a graph \( G \) for which every hitting set \( X \) has 
\[
c(X) \geq (5/2 - \epsilon) \text{SA}_1(G, c)
\] 
for \( c := 1/G \). Let \( G \) be a graph whose girth is at least \( k \) for some constant \( k \geq 5 \) and with the independence number \( \alpha(G) \leq n/k \) where \( n := |G| \). It can be shown via the probabilistic method that such a \( G \) exists, see [2]. Set \( c(v) := 1 \) for all \( v \in V(G) \). We have 
\[
c(X) \geq n(1 - 2/k) \text{ for every hitting set } X.
\]
To see this observe that since \( G \) is triangle-free and \( \alpha(G) \leq n/k \), when we remove \( X \) we will get at most \( n/k \) components each of size at most 2. Thus there are at most \( 2n/k \) vertices in \( G - X \), so \( |X| \geq n - 2n/k \). Therefore, \( \text{OPT}(G, c) \geq (1 - 2/k)n \).

In order to show \( \text{SA}_1(G, c) \leq 2n/5 \), we construct the following feasible solution \( x \) to \( \text{SA}_1(G, c) \). Set \( x_v := 2/5 \) for all \( v \in V(G) \) and \( x_{uv} := 0 \) if \( uv \in E(G) \) and \( x_{uv} := 1/5 \) if \( uv \notin E(G) \). The inequalities defining \( \text{SA}_1(G) \) are all satisfied by \( x \). This is obvious for inequalities (1), (3) and (4). For inequality (2), notice that at most one of \( uz, vz, wz \) can be an edge of \( G \), since otherwise \( G \) would have a cycle of length at most 4. Thus (2) is satisfied too, \( x \in \text{SA}_1(G) \) and \( \text{SA}_1(G, c) \leq 2n/5 \).

This completes the proof since, by taking \( k \geq 5/\epsilon \), we have 
\[
\text{OPT}(G, c) \geq n(1 - 2/k) \geq (5/2 - \epsilon)2n/5 \geq (5/2 - \epsilon)\text{SA}_1(G, c).
\] 

We now show that the integrality gap decreases to \( 2 + \epsilon \) after applying poly(1/\( \epsilon \)) rounds of Sherali-Adams. We first need the following lemma.

**Lemma 12** Fix \( \alpha \geq 1 \) and \( r \in \mathbb{Z}_{\geq 0} \). Let \((G, c)\) be a minimum order weighted graph such that \( \text{OPT}(G, c) > \alpha \cdot \text{SA}_r(G, c) \). The following two assertions hold:

(i) if \( x \) is an optimal solution to \( \text{SA}_r(G, c) \), then \( x_v < 1/\alpha \) for all \( v \in V(G) \);

(ii) \( G \) is connected and twin-free.

**Proof** (i) Suppose for a contradiction that \( x_v \geq 1/\alpha \), for some \( v \in V(G) \). Note that \( x \) restricted to \( V(G) \setminus \{v\} \) is a feasible solution to \( \text{SA}_r(G - v, c) \). Thus \( \text{SA}_r(G - v, c) \leq \frac{r}{r+1} \omega \). Thus the integrality gap decreases to \( 2 + \epsilon \) after applying poly(1/\( \epsilon \)) rounds of Sherali-Adams. We first need the following lemma.
be a clique of size \( r \) from Section 2.3, one can obtain a cost function for cliques given above, except that this time we let a vertex \((v)\) neighborhood \(N(v)\) since otherwise we invoke Theorem 3 (taking \(c_H\) for \(SA_r\)\(, \beta=\alpha\)).

(ii) Note that \(G\) is connected, otherwise there exists a connected component \(H\) of \(G\) such that \(OPT(H, c_H) > \alpha \cdot SA_r(H, c_H)\), where \(c_H\) is the restriction of \(c\) to \(V(H)\), contradicting the minimality of \(G\). To show that \(G\) is twin-free, we proceed exactly as in the proof of Theorem 3. □

Theorem 5 For every fixed \(\varepsilon > 0\), performing \(r = \text{poly}(1/\varepsilon)\) rounds of the Sherali-Adams hierarchy produces an LP relaxation of \textsc{cluster-\text{vd}} whose integrality gap is at most \(2 + \varepsilon\). That is, \(OPT(G, c) \leq (2 + \varepsilon)SA_r(G, c)\) for all weighted graphs \((G, c)\).

Proof In order to simplify the notation below, let us assume that \(2/\varepsilon\) is integer. For instance, we could restrict to \(\varepsilon = 2^{-l}\) for some \(l \in \mathbb{Z}_{>1}\). This does not hurt the generality of the argument. We take \(r := 1 + (2/\varepsilon)^4\). We may assume that \(\varepsilon < 1/2\) since otherwise we invoke Theorem 3 (taking \(r = 1\) suffices in this case).

Let \((G, c)\) be a counterexample to the theorem, with \(|G|\) minimum. By Lemma 12.(i), for every optimal solution \(x\) to \(SA_r(G, c)\), every vertex \(v \in V(G)\) has \(x_v < 1/(2 + \varepsilon)\). By Lemma 12.(ii), \(G\) is twin-free (and connected).

We will use the following fact several times in the proof: for all \(R \subseteq V(G)\) with \(|R| \leq r\) and every \(x \in SA_r(G)\), the restriction of \(x\) to the variables in \(R\) is a convex combination of hitting sets of \((G|R)\). This is easy to see since, denoting by \(x_R\) the restriction of \(x\), we get \(x_R \in SA_r(G|R)\) and the Sherali-Adams hierarchy is known to converge in at most “dimension-many” rounds, see for instance [18].

First, we claim that \(G\) has no clique of size at least \(2/\varepsilon\). Suppose otherwise. Let \(C\) be a clique of size \(k := 2/\varepsilon\) and let \(D\) be a minimal set such that each edge of \(C\) has a distinguisher in \(D\). Let \(H := G[C \cup D]\). Then, following the construction from Section 2.3, one can obtain a cost function \(c_H\) such that \(c_H(H) = 2k - 1\), and \(c_H(X) \geq k - 1\) for any hitting set \(X\) of \(H\). See [20, Lemma 3] for the full construction, whose proof also shows that \(|H| \leq 2k - 1 \leq r\). Since every valid inequality supported on at most \(r\) vertices is valid for \(SA_r(G)\), the inequality \(\sum_v c_H(v)x_v \geq k - 1\) is valid for \(SA_r(G)\). Since \(c_H(H) = 2k - 1\), this implies that for all \(x \in SA_r(G)\), there is some vertex \(a \in V(H)\) with \(x_a \geq (k - 1)/(2k - 1)\). Since \((k - 1)/(2k - 1) \geq 1/(2 + \varepsilon)\), we get a contradiction. This proves our first claim.

Second, we claim that for every \(v_0 \in V(G)\), the subgraph of \(G\) induced by the neighborhood \(N(v_0)\) has no stable set of size at least \(2/\varepsilon\). The proof is similar to that for cliques given above, except that this time we let \(H\) be the induced star \(K_{1,k}\) with apex \(v_0\) and \(k = 2/\varepsilon\). The cost function \(c_H\) given by Lemma 5 has \(c_H(v_0) = k - 1\) and \(c_H(v) = 1\) for all \(v \in S\). Notice that once again \(c_H(H) = 2k - 1\). The star inequality \(\sum_v c_H(v)x_v \geq k - 1\) is valid for \(SA_r(G)\), which guarantees that for every \(x \in SA_r(G)\) there is some \(a \in V(H)\) which has \(x_a \geq (k - 1)/(2k - 1)\). This establishes our second claim.

Third, we claim that the neighborhood of every vertex \(v_0\) induces a chordal subgraph of \(G\). Suppose that \(C\) is a hole in \(G[N(v_0)]\). We first deal with the case \(|C| \leq r - 1 = (2/\varepsilon)^4\). We can repeat the same proof as above, letting \(H\) be the induced wheel on
\[ V(C) + v_0 \text{ and using the cost function } c_H \text{ defined in the proof of Lemma 1. Consider the wheel inequality } \sum_v c_H(v)x_v \geq k - 3, \text{ where } k := |H| = |C| + 1. \]

Since the wheel has at most \( r \) vertices, the wheel inequality is valid for \( \text{SA}_r(G) \). Since \( c_H(H) = 2k - 6 = 2(k - 3) \), for every \( x \in \text{SA}_r(G) \), there is some \( a \in V(H) \) which has \( x_a \geq 1/2 \geq 1/(2 + \varepsilon) \). This concludes the case where \( |C| \) is “small”.

Now, assume that \( |C| \geq r \), and consider the wheel inequality with right-hand side scaled by \( 2/(2 + \varepsilon) \). Suppose this inequality is valid for \( \text{SA}_r(G) \). This still implies that some vertex \( a \) of \( H \) has \( x_a \geq 1/(2 + \varepsilon) \), for all \( x \in \text{SA}_r(G) \), which produces the desired contradiction. It remains to prove that the scaled wheel inequality is valid for \( \text{SA}_r(G) \).

Let \( F \) denote any \( r \)-vertex induced subgraph of \( H \) that is a fan.\(^2\) Hence, \( F \) contains \( v_0 \) as an apex vertex, plus a path on \( r - 1 \) vertices. Letting \( c_F(v_0) := r - 3 - [(r - 1)/3] \) and \( c_F(v) := 1 \) for \( v \in V(F - v_0) \), we get the inequality \( \sum_v c_F(v)x_v \geq r - 3 \), which is valid for \( \text{SA}_r(G) \). By taking all possible choices for \( F \), and averaging the corresponding inequalities, we see that the inequality

\[
\left( r - 3 - \left\lfloor \frac{r - 1}{3} \right\rfloor \right)x_{v_0} + \frac{r - 1}{k - 1} \sum_{v \in V(H - v_0)} x_v \geq r - 3
\]

\[
\iff \frac{k - 1}{r - 1} \left( r - 3 - \left\lfloor \frac{r - 1}{3} \right\rfloor \right)x_{v_0} + \sum_{v \in V(H - v_0)} x_v \geq \frac{k - 1}{r - 1}(r - 3)
\]

is valid for \( \text{SA}_r(G) \). It can be seen that this inequality dominates the scaled wheel inequality, in the sense that each left-hand side coefficient is not larger than the corresponding coefficient in the scaled wheel inequality, while the right-hand side is not smaller than the right-hand side of the scaled wheel inequality. Therefore, the scaled wheel inequality is valid for \( \text{SA}_r(G) \). This concludes the proof of our third claim.

By the first, second and third claim\(^3\), \(|N(v_0)| \leq \omega(G[N(v_0)]) \cdot \alpha(G[N(v_0)]) \leq 4/\varepsilon^2\) for all choices of \( v_0 \). This implies in particular that \(|N_{2}[v_0]| \leq 1 + 16/\varepsilon^4 = r\). Now let \( H := G[N_{2}[v_0]] \). Theorem 2 applies since \( G \) is twin-free, by our second claim. Let \( c_H \) be the corresponding cost function. The inequality \( \sum_v c_H(v)x_v \geq \text{OPT}(H, c_H) \) is valid for \( \text{SA}_r(G) \).

Let \( \lambda^* \) be defined as in Step 15 of Algorithm 1, and let \( a \in V(G) \) denote any vertex such that \( (c - \lambda^*c_H)(a) = 0 \). By minimality of \( G \), there exists in \( (G', c') := (G - a, c - \lambda^*c_H) \) a minimal hitting set \( X' \) of cost \( c'(X') \leq (2 + \varepsilon) \text{SA}_r(G', c') \). We let \( X := X' \) in case \( X' \) is a hitting set of \( G \), and \( X := X' + a \) otherwise. Assume that \( X = X' + a \), the other case is easier. We have

\[
c(X) = c'(X') + \lambda^*c_H(X)
\]

\[
\leq (2 + \varepsilon)\text{SA}_r(G', c') + \lambda^*(c_H(H) - 1)
\]

\[
\leq (2 + \varepsilon)\text{SA}_r(G', c') + 2\lambda^*\text{OPT}(H, c_H)
\]

\[
\leq (2 + \varepsilon)\left(\text{SA}_r(G', c') + \lambda^*\text{OPT}(H, c_H)\right).
\]

\(^2\) A fan is a graph obtained from a path by adding an apex vertex.

\(^3\) The first inequality follows since \(|H| \leq \alpha(H) \cdot \omega(H)\), for every perfect graph \( H \).
By LP duality, we have $\text{SA}_{r}(G, c) \geq \text{SA}_{r}(G', c') + \lambda^{*} \text{OPT}(H, c_{H})$. This implies that $c(X) \leq (2 + \varepsilon) \text{SA}_{r}(G, c)$, contradicting the fact that $(G, c)$ is a counterexample. This concludes the proof. \hfill \Box

We now complement the result above by showing that every LP relaxation of $\mathsf{Cluster-VD}$ with (worst case) integrality gap at most $2 - \varepsilon$ must have super-polynomial size. The result is a simple consequence of an analogous result of [10] on the integrality gap of $\mathsf{Vertex-Cover}$, and of the straightforward reduction from $\mathsf{Vertex-Cover}$ to $\mathsf{Cluster-VD}$.

**Proposition 1** For infinitely many values of $n$, there is a graph $G$ on $n$ vertices such that every size-$n^{o(\log n / \log \log n)}$ LP relaxation of $\mathsf{Cluster-VD}$ on $G$ has integrality gap $2 - o(1)$.

**Proof** In [10] a similar result is proved for LP-relaxations of $\mathsf{Vertex-Cover}$: for infinitely many values of $n$, there is a graph $G$ on $n$ vertices such that every size-$n^{o(\log n / \log \log n)}$ LP relaxation of $\mathsf{Vertex-Cover}$ on $G$ has integrality gap at least $2 - \varepsilon$, where $\varepsilon = \varepsilon(n) = o(1)$ is a non-negative function.

Let $G$ be such a graph, and let $G^{+}$ be the graph obtained from $G$ by attaching a pendant edge to every vertex. It is easy to see that $U \subseteq V(G)$ is a hitting set for $G^{+}$ if and only if $U$ is a vertex cover of $G$.

Toward a contradiction, suppose that $Ax \geq b$ is a size-$n^{o(\log n / \log \log n)}$ LP relaxation of $\mathsf{Cluster-VD}$ on $G^{+}$ with integrality gap at most $2 - \delta$, for a fixed $\delta > \varepsilon$ (where $x \in \mathbb{R}^{d}$ for some dimension $d$ depending on $G$). For every $c^{+} \in Q_{\geq 0}^{V(G^{+})}$ there exists a hitting set $X$ of $G^{+}$ such that $c^{+}(X) \leq (2 - \delta) \text{LP}(G^{+}, c^{+})$.

We can easily turn $Ax \geq b$ into an LP relaxation for $\mathsf{Vertex-Cover}$. For every vertex cover $U$ of $G$, we let the corresponding point be the point $\pi^{U} \in \mathbb{R}^{d}$ for $U$ seen as a hitting set in $G^{+}$. For every $c \in Q_{\geq 0}^{V(G)}$, we define $c^{+} \in Q_{\geq 0}^{V(G^{+})}$ via $c^{+}(v) := c(v)$ for $v \in V(G)$, and $c^{+}(v) := \sum_{u \in V(G)} c(u)$ for $v \in V(G^{+}) \setminus V(G)$. Then, we let the affine function $f_{c}$ for $c$ be the affine function $f_{c^{+}}$ for $c^{+}$.

Since the integrality gap of $Ax \geq b$, seen as an LP relaxation of $\mathsf{Cluster-VD}$, is at most $2 - \delta$, for every $c \in Q_{\geq 0}^{V(G)}$ there exists a hitting set $X$ of $G^{+}$ whose cost is at most $(2 - \delta) \text{LP}(G^{+}, c^{+})$, where $c^{+}$ is the cost function corresponding to $c$. If $X$ contains any vertex of $V(G^{+}) \setminus V(G)$, we can replace this vertex by its unique neighbor in $V(G)$, without any increase in cost. In this way, we can find a vertex cover $U$ of $G$ whose cost satisfies $c(U) \leq c^{+}(X) \leq (2 - \delta) \text{LP}(G^{+}, c^{+}) = (2 - \delta) \text{LP}(G, c)$. Hence, the integrality gap of $Ax \geq b$ as an LP relaxation of $\mathsf{Vertex-Cover}$ is also at most $2 - \delta < 2 - \varepsilon$. As the size of $Ax \geq b$ is $n^{o(\log n / \log \log n)}$, this provides the desired contradiction. \hfill \Box

We point out that the size bound in the previous result can be improved. Kothari, Meka and Raghavendra [27] have shown that for every $\varepsilon > 0$ there is a constant $\delta = \delta(\varepsilon) > 0$ such that no LP relaxation of size less than $2^{\varepsilon^{2}}$ has integrality gap less than $2 - \varepsilon$ for Max-CUT. Since Max-CUT acts as the source problem in [10], one gets a $2^{\varepsilon^{2}}$ size lower bound for $\mathsf{Vertex-Cover}$ in order to achieve integrality gap $2 - \varepsilon$. This also follows in a black-box manner from [27] and [13]. The proof of Proposition 1 shows that the same bound applies to $\mathsf{Cluster-VD}$. \hfill \Box
5 Conclusion

In this paper we provide a tight approximation algorithm for the cluster vertex deletion problem (CLUSTER-VD). Our main contribution is the efficient construction of a local cost function on the vertices at distance at most 2 from any vertex $v_0$ such that every minimal hitting set of the input graph has local cost at most twice the local optimum. If the subgraph induced by $N(v_0)$ (the first neighborhood of $v_0$) contains a hole, or a $2P_3$, then this turns out to be straightforward. The most interesting case arises when the local subgraph $H$ is twin-free, has radius 1, and moreover $H[N(v_0)] = H - v_0$ is chordal and $2P_3$-free. Such graphs are very structured, which we crucially exploit.

Lemma 3 allows us to define the local cost function on the vertices distinct from $v_0$ and then later adjust the cost of $v_0$. We point out that condition (ii) basically says that the local cost function should define a hyperplane that “almost” separates the hitting set polytope and the clique polytope of the chordal, $2P_3$-free graph $H - v_0$. This was a key intuition which led us to the proof of Theorem 2. If these polytopes were disjoint, this would be easy. But actually it is not the case since they have a common vertex (as we show, $H - v_0$ has a hitting clique).

One natural question arising from our approach of CLUSTER-VD in general graphs is the following: is the problem polynomial-time solvable on chordal graphs? This seems to be a non-trivial open question, also mentioned in [16], where similar vertex deletion problems are studied for chordal graphs. It could well be that CLUSTER-VD in general chordal graphs is hard. Now, what about chordal, $2P_3$-free graphs? We propose this last question as our first open question.

Our second contribution are polyhedral results for the CLUSTER-VD problem, in particular with respect to the tightness of the Sherali-Adams hierarchy. Our results on Sherali-Adams fail to match the 2-approximation factor of our algorithm (by epsilon), and we suspect this is not by chance. We believe that, already for certain classes of triangle-free graphs, the LP relaxation given by a bounded number of rounds of the Sherali-Adams hierarchy has an integrality gap strictly larger than 2. Our intuition goes as follows. Consider the star inequality $(k - 1)x_{v_0} + \sum_{i=1}^{k} x_{v_i} \geq k - 1$, valid when $N(v_0) = \{v_1, \ldots, v_k\}$ is a stable set. Capturing all star inequalities is sufficient to achieve an integrality gap of at most 2 for all triangle-free graphs [20, Algorithm 1]. However, we suspect that Sherali-Adams will have a hard time recovering these in a constant number of rounds. The star inequality is very similar to the clique inequality $\sum_{i=1}^{k} x_{v_i} \geq k - 1$, which is valid for VERTEX COVER when $\{v_1, \ldots, v_k\}$ is a clique. It is known that Sherali-Adams is unable to capture all clique inequalities in a constant number of rounds of the VERTEX COVER relaxation (see [28, Section 6.1] for an equivalent statement on clique inequalities for the stable set polytope). Whether this intuition is accurate is our second open question.

As mentioned already in the introduction, we do not know any polynomial-size LP or SDP relaxation with integrality gap at most 2 for CLUSTER-VD. In order to obtain such a relaxation, it suffices to derive each valid inequality implied by Lemmas 1, 2, 5 and also somehow simulate Lemma 6. Here, different techniques to construct extended formulations (see for instance [3,8,37]) could be used. A partial result in this direction is that the star inequality has a bounded-degree sum-of-squares proof. This implies that a bounded number of rounds of the Lasserre hierarchy provides an SDP
relaxation for \textsc{Cluster-VD} with integrality gap at most 2, whenever the input graph is triangle-free. This should readily generalize to the wheel inequalities of Lemma 1 and of course to the inequality of Lemma 2 (since the underlying graph has bounded size). However, we do not know how for instance to derive the inequalities of Lemma 5. We leave this for future work as our third open question.

Our fourth open question: what is the best running time for Algorithm 1? We think that it is possible to improve on our $O(n^4)$ upper bound.

Another intriguing problem is to what extent our methods can be adapted to hitting set problems in other 3-uniform hypergraphs. We mention an open question due to László Végh [39]: for which classes of 3-uniform hypergraphs and which $\varepsilon > 0$ does the hitting set problem admit a $(3-\varepsilon)$-approximation algorithm?

As mentioned in the introduction, \textsc{FVST} (feedback vertex set in tournaments) is another hitting set problem in a 3-uniform hypergraph, which is also UGC-hard to approximate to a factor smaller than 2. There is a recent randomized 2-approximation algorithm [30], but no deterministic (polynomial-time) algorithm is known. Let us repeat here the relevant open question from [30]: does \textsc{FVST} admit a deterministic 2-approximation algorithm?

\textbf{Acknowledgements} We are grateful to Daniel Lokshtanov for suggesting Lemma 4, which allowed us to simplify our algorithm and its proof. We also thank two anonymous referees for their helpful comments, which improved the presentation of the paper.

\textbf{References}

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. Nature 406(6794), 378–382 (2000)
2. Alon, N., Spencer, J.H.: The Probabilistic Method, 4th edn. Wiley, New York (2016)
3. Aprile, M.: Extended formulations for matroid polytopes through randomized protocols. \textit{arXiv preprint arXiv:2106.12453}, (2021)
4. Aprile, M., Castro, N., Ferreira, G., Piccini, J., Robledo, F., Romero, P.: Graph fragmentation problem: analysis and synthesis. Int. Trans. Oper. Res. 26(1), 41–53 (2019)
5. Aprile, M., Castro, N., Robledo, F., Romero, P.: Analysis and complexity of node-immunization under natural disasters. In: DRCN 2017-Design of Reliable Communication Networks; 13th International Conference, pages 1–8. VDE, (2017)
6. Aprile, M., Drescher, M., Fiorini, S., Huyhn, T.: A simple 7/3-approximation algorithm for feedback vertex set in tournaments. \textit{arXiv preprint arXiv:2008.08779}, (2020)
7. Aprile, M., Drescher, M., Fiorini, S., Huyhn, T.: A tight approximation algorithm for the cluster vertex deletion problem. In: International Conference on Integer Programming and Combinatorial Optimization, pages 340–353. Springer, (2021)
8. Aprile, M., Faenza, Y.: Extended formulations from communication protocols in output-efficient time. Math. Programm. 183(1), 41–59 (2020)
9. Aprile, M. F., Faenza, Y., Fiorini, S., Huyhn, T., Macchia, M.: Extension complexity of stable set polytopes of bipartite graphs. Lecture Notes in Computer Science book series (LNCS), 10520(CONF):75–87, (2017)
10. Bazzi, A., Fiorini, S., Pokutta, S., Svensson, O.: No small linear program approximates vertex cover within a factor $2 - \varepsilon$. Math. Oper. Res. 44(1), 147–172 (2019)
11. Blair, J.R., Peyton, B.: An introduction to chordal graphs and clique trees. In: Graph theory and sparse matrix computation, pp. 1–29. Springer, (1993)
12. Boral, A., Cygan, M., Kociumaka, T., Pilipczuk, M.: A fast branching algorithm for cluster vertex deletion. Theory Comput. Syst. 58(2), 357–376 (2016)
13. Braun, G., Pokutta, S., Roy, A.: Strong reductions for extended formulations. Math. Program. 172(1–2), 591–620 (2018)
1091

14. Braun, G., Pokutta, S., Zink, D.: Inapproximability of combinatorial problems via small LPs and SDPs.
   In: Proceedings of STOC 2015, pp. 107–116, New York, NY, USA. ACM (2015)

15. Cai, M.-C., Deng, X., Zang, W.: An approximation algorithm for feedback vertex sets in tournaments.
   SIAM J. Comput. 30(6), 1993–2007 (2001)

16. Cao, Y., Ke, Y., Otachi, Y., You, J.: Vertex deletion problems on chordal graphs. Theor. Comput. Sci.
   745, 75–86 (2018)

17. Chan, S.O., Lee, J.R., Raghavendra, P., Steurer, D.: Approximate constraint satisfaction requires large
   LP relaxations. J. ACM (JACM) 63(4), 1–22 (2016)

18. Conforti, M., Cornuéjols, G., Zambelli, G.: Integer Programming, vol. 271. Springer, Berlin (2014)

19. Fiorini, S., Joret, G., Schaudt, O.: Improved approximation algorithms for hitting 3-vertex paths. In: 
   International Conference on Integer Programming and Combinatorial Optimization, pp. 238–249.
   Springer, (2016)

20. Fiorini, S., Joret, G., Schaudt, O.: Improved approximation algorithms for hitting 3-vertex paths. Math.
   Program. 182(1–2, Ser. A), 355–367 (2020)

21. Freund, A., Bar-Yehuda, R., Bendel, K.: Local ratio: a unified framework for approximation algorithms.
   ACM Trans. Algorithms 36, 422–463 (2005)

22. Giovannaz, S., Batenko, S.: Polyhedral properties of the induced cluster subgraphs. Discrete Appl.
   Math. 297, 80–96 (2021)

23. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster
   vertex deletion. Theory Comput. Syst. 47(1), 196–217 (2010)

24. Jahanpour, E., Chen, X.: Analysis of complex network performance and heuristic node removal
   strategies. Commun. Nonlinear Sci. Numer. Simul. 18(12), 3458–3468 (2013)

25. Kothari, P.K., Meka, R., Raghavendra, P.: Approximating rectangles by juntas and weakly-exponential
   lower bounds for LP relaxations of CSPs. In: Hatami, H., McKenzie, P., King, V. (eds), Proceedings
   of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC,
   Canada, June 19-23, 2017, pages 590–603. ACM, (2017)

26. Laurent, M.: A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1
   programming. Math. Oper. Res. 28(3), 470–496 (2003)

27. Lokshtanov, D.: Personal communication

28. Lokshtanov, D., Misra, P., Mukherjee, J., Panolan, F., Philip, G., Saurabh, S.: 2-approximating feedback
   vertex set in tournaments. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on
   Discrete Algorithms, pages 1010–1018. SIAM, (2020)

29. Mnich, M., Williams, V.V., Végh, L.A.: A 7/3-approximation for feedback vertex sets in tournaments. In:
   Proceedings of the 24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016,
   Aarhus, Denmark, volume 57 of LIPIcs, pages 67:1–67:14. Schloss Dagstuhl - Leibniz-Zentrum
   für Informatik, (2016)

30. Rothvoß, T.: The Lasserre hierarchy in approximation algorithms. Lecture Notes for the MAPSP, pp.
    1–25, (2013)

31. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1998)

32. Sherali, H.D., Adams, W.P.: A hierarchy of relaxations between the continuous and convex hull rep-
   resentations for zero-one programming problems. SIAM J. Discrete Math. 3(3), 411–430 (1990)

33. Tarjan, R.E.: Decomposition by clique separators. Discrete Math. 55(2), 221–232 (1985)

34. Tarjan, R.E., Yannakakis, M.: Simple linear-time algorithms to test chordality of graphs, test acyclicity
   of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM J. Comput. 3(3), 566–579 (1984)

35. Tiwary, H. R., Koutecký, M., Kolman, P.: Extension complexity, MSO logic, and treewidth. Discrete
   Math. Theor. Computer Sci., 22, (2020)

36. Tsug, D.: Faster parameterized algorithm for cluster vertex deletion. CoRR, arXiv:1901.07609, (2019)

37. Végh, L.A.: Personal communication

38. You, J., Wang, J., Cao, Y.: Approximate association via dissociation. Discret. Appl. Math. 219, 202–209
   (2017)