A 7/3-Approximation Algorithm for Cluster Vertex Deletion

Samuel Fiorini[1], Gwenaël Joret[2], and Oliver Schaudt[3]

[1] Département de Mathématique, Université libre de Bruxelles, Brussels, Belgium, sfiorini@ulb.ac.be
[2] Département d'Informatique, Université libre de Bruxelles, Brussels, Belgium, gjoret@ulb.ac.be
[3] Institut für Informatik, Universität zu Köln, Köln, Germany, schaudto@uni-koeln.de

Abstract. The cluster vertex deletion problem is to delete a minimum cost set of vertices from a given vertex-weighted graph in such a way that the resulting graph has no induced path on three vertices. This problem is a special case of the vertex cover problem on a 3-uniform hypergraph, and thus has a straightforward 3-approximation algorithm. Very recently, You, Wang, and Cao [14] described an efficient 5/2-approximation algorithm for the unweighted version of the problem. Our main result is a 7/3-approximation algorithm for arbitrary weights, using the local ratio technique. We further conjecture that the problem admits a 2-approximation algorithm and give some support for the conjecture. This is in sharp constrast with the fact that the similar problem of deleting vertices to eliminate all triangles in a graph is known to be UGC-hard to approximate to within a ratio better than 3, as proved by Guruswami and Lee [7].

1 Introduction

Given a graph[4] G and cost function $c : V(G) \to \mathbb{R}_+$, the *cluster vertex deletion problem* (Cluster-VD) is to find a minimum cost set X of vertices such that $G-X$ is a disjoint union of cliques. Equivalently, $X \subseteq V(G)$ is a feasible solution if and only if $G - X$ contains no induced subgraph isomorphic to P_3, the path on three vertices.

It should be clear that the problem has a 3-approximation algorithm: Assuming unit costs for simplicity, build any inclusionwise maximal collection $\mathcal{C}$ of vertex-disjoint induced P_3's in G and include in X every vertex covered by some member of $\mathcal{C}$. If $\mathcal{C}$ contains k subgraphs then we get a lower bound of k on the optimum. On the other hand, the cost of X is $3k$.

The problem admits an approximation-preserving reduction from Vertex Cover: if H is any given graph, let G denote the graph obtained from H by adding a pendent edge to every vertex. Then solving Vertex Cover on H is

[4] Graphs in this paper are finite, simple, and undirected.

equivalent to solving Cluster-VD on G. Hence it is UGC-hard to approximate Cluster-VD within any ratio better than 2. We show that we can however come close to 2.

Theorem 1. Cluster-VD *admits a* 7/3*-approximation algorithm.*

We further conjecture that Cluster-VD can be 2-approximated in polynomial time, as is the case for Vertex Cover. We give some support for this conjecture in Section 6, where we report on a 2-approximation algorithm for the case where the input graph does not contain a diamond (K_4 minus an edge) as an induced subgraph.

In contrast, the problem of finding a minimum cost set of vertices X such that $G - X$ has no triangle is known to be UGC-hard to approximate to within any ratio better than 3, as proved by Guruswami and Lee [7] (see also [8] for related inapproximability results).

Previous Work. Cluster-VD was previously mostly studied in terms of fixed parameter algorithms. Hüffner *et al.* [9] first gave a $O(2^k k^9 + nm)$-time fixed-parameter algorithm, parameterized by the solution size k, where n and m denote the number of vertices and edges of the graph, respectively. This was subsequently improved by Boral *et al.* [3], who gave a $O(1.9102^k(n + m))$-time algorithm. See also Iwata and Oka [10] for related results in the fixed parameter setting.

As for approximation algorithms, nothing better than a 3-approximation was known until the very recent work of You, Wang, and Cao [14], who showed that the unweighted version of Cluster-VD admits a 5/2-approximation algorithm. They further showed that their algorithm could be implemented efficiently, in $O(nm + n^2)$-time, using fast modular decomposition.

We note that the work of You *et al.* [14] and ours have been done independently. While we obtained a better approximation ratio of 7/3, let us remark that the running time of our algorithm is much larger (though still polynomial). We leave it as an open question whether it could be brought down to a small polynomial using the techniques from [14].

Incidentally, there was recent activity on another restriction of the vertex cover problem on 3-uniform hypergraph, namely, the feedback vertex set problem in tournaments. For that problem, the 5/2-approximation algorithm by Cai, Deng and Zang [4] was the best known for many years, until the very recent work of Mnich, Vassilevska Williams and Végh [12] who found a 7/3-approximation algorithm for the problem.

Our approach. Our approximation algorithm is based on the *local ratio* technique. In order to illustrate the general approach, let us give a very simple 2-approximation algorithm for hitting all P_3*-subgraphs* (instead of induced subgraphs) in a given weighted graph (G, c), see Algorithm 1 below.

It can be easily verified that the set X returned by Algorithm 1 is an inclusionwise minimal feasible solution. The reason why the algorithm is a 2-approximation is that optimum cost for the weighted star (H, c_H) is $d(u) - 1$

Algorithm 1 HITTING-P_3-SUBGRAPHS-APX(G, c)

Input: (G, c) a weighted graph
Output: X an inclusionwise minimal set of vertices hitting all the P_3 subgraphs
  if G has no P_3 subgraph **then**
    $X \leftarrow \varnothing$
  else if (G, c) has some zero-cost vertex u **then**
    $X' \leftarrow$ HITTING-P_3-SUBGRAPHS-APX$(G - u, c$ restricted to $V(G - u))$
    $X \leftarrow X'$ if $G - X'$ has no P_3 subgraph; $X \leftarrow X' \cup \{u\}$ otherwise
  else
    $u \leftarrow$ vertex of degree $d(u) \geqslant 2$, and let (H, c_H) be the weighted star centered at u with $V(H) := N(u) \cup \{u\}$, $c_H(u) := d(u) - 1$ and $c_H(v) := 1$ for $v \in N(u)$
    $\lambda^* \leftarrow$ maximum scalar λ s.t. $c(v) - \lambda c_H(v) \geqslant 0$ for all $v \in V(H)$
    $X \leftarrow$ HITTING-P_3-SUBGRAPHS-APX$(G, c - \lambda^* c_H)$
  end if
  return X

while the solution X returned by algorithm misses at least one of the vertices of the star, and thus a local cost of at most $2(d(u) - 1)$.

We remark that a 2-approximation algorithm for the problem of hitting P_3-subgraphs can also be obtained via a straightforward modification of the primal/dual 2-approximation algorithm of Chudak *et al.* [5] for the feedback vertex set problem. (Indeed, this is exactly what was done by Tu and Zhou [13].) However, the resulting algorithm is nowhere near as simple as Algorithm 1.

It is perhaps worth pointing out that, in the case of triangle-free graphs, hitting P_3's or induced P_3's are the same problem. This was actually an important insight for the 5/2-approximation algorithm of You, Wang, and Cao [14]. However, for arbitrary graphs the induced version of the problem seems much more difficult. Nevertheless, we are tempted to take the simplicity of Algorithm 1 as a hint that the local ratio technique is a good approach to attack the problem.

From a high level point of view, the structure of our 7/3-approximation algorithm for CLUSTER-VD is as follows: As long as there is an induced P_3 in the graph, either we can apply a reduction operation (identifying *true twins*) that does not change the optimum, or we find some special weighted induced subgraph (H, c_H) and decrease the weights of its vertices in (G, c) proportionally to the weights chosen for the induced subgraph H, ensuring a local ratio of 7/3. The crux of our proof is showing that, if no reduction can be applied, then the aforementioned special induced subgraph always exists. The list of induced subgraphs that we look for is given in Fig. 1. Every graph on the list has at most 7 vertices, and thus we can test their existence in $O(n^7)$-time.

2 Definitions and Preliminaries

Let G be a graph. Recall that the feasible solutions to CLUSTER-VD in G are the sets of vertices X that intersect every induced subgraph isomorphic to P_3. For this reason, we call such sets X *hitting sets* of G. We denote by $\mathrm{OPT}(G)$

the minimum size of a hitting set of G. The definitions extend naturally in the weighted setting: Given a weighted graph (G, c), where $c : V(G) \to \mathbb{R}_+$, we let $\mathrm{OPT}(G, c)$ denote the minimum weight of a hitting set of G. As expected, the *cost* of set $X \subseteq V(G)$ is defined as $c(X) := \sum_{v \in X} c(v)$.

For $X \subseteq V(G)$, the subgraph of G induced by X is denoted by $G[X]$. When H is an induced subgraph of G or isomorphic to an induced subgraph of G, we sometimes say that G *contains* H. If G does not contain H, we also say that G is *H-free*.

For $v \in V(G)$, the neighborhood of v is denoted by $N(v)$. From time to time, to indicate that x is a neighbor of y, we simply say that x *sees* y.

In a few occasions in the paper we resort to *trigraphs*, which are graphs with a set of special edges called the *undecided edges* (in figures, these are typically represented by wiggly edges). A trigraph is an efficient way to represent several graphs, its *instantiations*. These are the corresponding graphs in which each undecided edge may become an edge or not. Much of the terminology we use for graphs can be extended to trigraphs in a natural way. In particular, we say that graph G contains trigraph H or that H is an induced subtrigraph of G if G contains some instantiation of H.

3 Tools

3.1 α-Good Induced Subgraphs

Given a graph G, an induced subgraph H of G, and a weighting $c_H : V(G) \to \mathbb{R}_+$, we say that (H, c_H) is *α-good in G* if for every inclusionwise minimal hitting set X of G we have

$$\sum_{v \in X \cap V(H)} c_H(v) \leqslant \alpha \cdot \mathrm{OPT}(H, c_H)\,. \tag{1}$$

Moreover, we say that an induced subgraph H of G is itself α-good in G if there exists a weighting c_H such that (H, c_H) is α-good. The first technical tool of our 7/3-approximation algorithm is the following lemma, which provides a list of α-good induced subgraphs where $\alpha \leqslant 7/3$, see Fig. 1. Due to length restrictions, the proof of the lemma can be found in the Appendix.

Lemma 1. *Let G be a graph and H be an induced subgraph of G. Then H is 7/3-good in G whenever*

(i) H is isomorphic to C_4, W_5, $K_{1,4}$, the dart, the turtle, H_1 or H_2;
(ii) H is isomorphic to an instantiation of H_3, H_4 or H_5;
(iii) H is isomorphic to P_3, $K_{1,3}$, the gem or the bull, and there exists some vertex of H that has no neighbor in $G - V(H)$.

For any graph G, let $\mathcal{H}(G)$ denote the collection of all weighted induced subgraphs (H, c_H) that are isomorphic to a weighted graph from Fig. 1 (the trigraph H_3 has 16 corresponding graphs). By Lemma 1, every $(H, c_H) \in \mathcal{H}(G)$ is 7/3-good in G. Notice that $\mathcal{H}(G)$ contains at most 45 isomorphism classes of

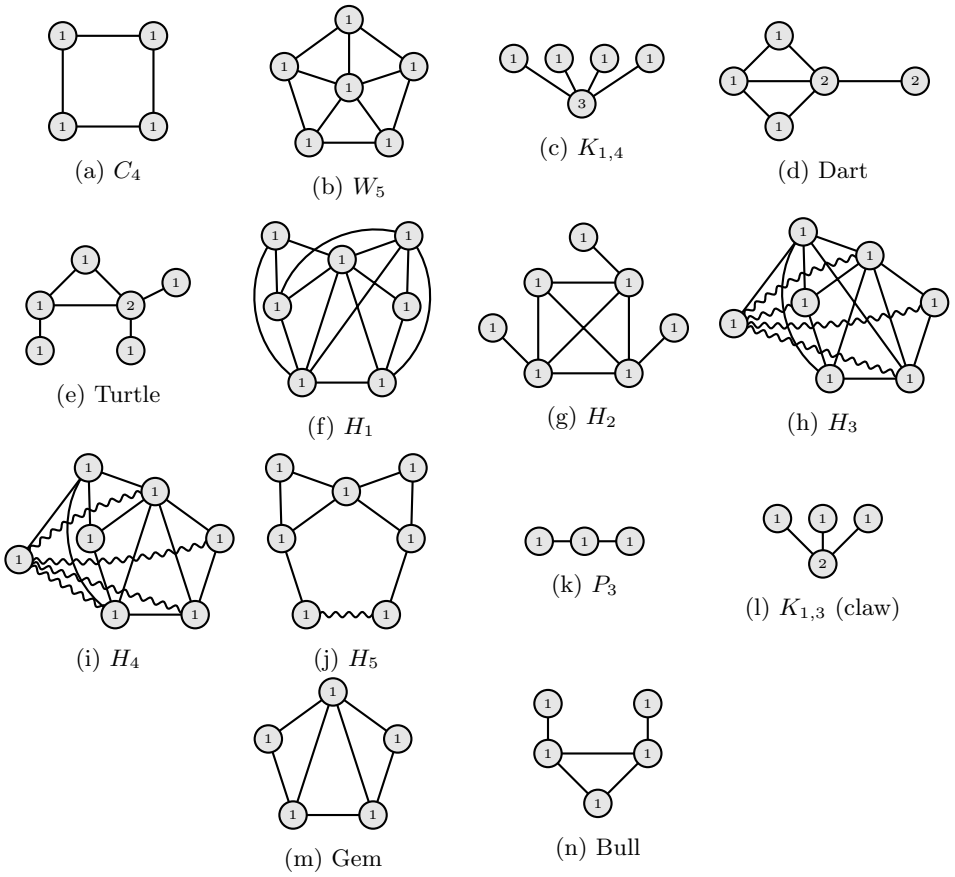

Fig. 1: The 7/3-good induced sub(tri)graphs of Lemma 1. For each corresponding induced subgraph H, a weighting c_H witnessing 7/3-goodness is shown.

weighted induced subgraphs, all of which involving graphs with at most 7 vertices. Notice also that in our collection of weighted induced subgraphs $\mathcal{H}(G)$, the induced subgraph H determines uniquely the weighting H. Thus $\mathcal{H}(G)$ contains $O(n^7)$ weighted induced graphs, where n denotes the number of vertices of G.

3.2 True Twins

Two vertices u, u' of a graph G are called *true twins* if they are adjacent and have the same neighborhood in $G - \{u, u'\}$. True twins have a particularly nice behavior regarding Cluster-VD, as proved in our next lemma. This our second main technical tool.

Lemma 2. *Let (G, c) be a weighted graph and $u, u' \in V(G)$ be true twins. Let (G', c') denote the weighted graph obtained from G by transferring the whole cost*

of u' to u and then deleting u', that is, let $G' := G - u'$ and $c'(v) := c(v)$ if $v \in V(H'), v \neq u$ and $c'(v) := c(u) + c(u')$ if $v = u$. Then $\mathrm{OPT}(G, c) = \mathrm{OPT}(G', c')$.

Proof. We have $\mathrm{OPT}(G, c) \leqslant \mathrm{OPT}(G', c')$ because every hitting set X' of G' yields a hitting set X of G with the same cost: we let $X := X' \cup \{u'\}$ if X contains u and $X := X'$ otherwise.

Conversely, we have $\mathrm{OPT}(G', c') \leqslant \mathrm{OPT}(G, c)$ because any inclusionwise minimal cost hitting set X of G either contains both of the true twins u and u', or none of them. □

If G does not contain any pair of true twins, we say that it is *twin-free*.

4 Algorithm

Our 7/3-approximation algorithm is described below, see Algorithm 2. Although we could have presented as a primal-dual algorithm, we chose to present it within the local ratio framework in order to avoid some technicalities, especially those related to the elimination of true twins.

The following lemma makes explicit a simple property of Cluster-VD that is key when using the local ratio technique. This property is common to many minimization problems, and is often referred to as the *Local Ratio Lemma*; see e.g. the survey of Bar-Yehuda *et al.* [2].

Lemma 3 (Local Ratio Lemma). *Let (G, c) be a weighted graph with c the sum of two cost functions c' and c'', and let $\alpha \geqslant 1$. If X is a hitting set of G such that $c'(X) \leqslant \alpha \cdot \mathrm{OPT}(G, c')$ and $c''(X) \leqslant \alpha \cdot \mathrm{OPT}(G, c'')$, then $c(X) \leqslant \alpha \cdot \mathrm{OPT}(G, c)$.*

Proof. Since $c(X) = c'(X) + c''(X)$, it is enough to show that $\mathrm{OPT}(G, c') + \mathrm{OPT}(G, c'') \leqslant \mathrm{OPT}(G, c)$. To see this, let X^* be an optimal hitting set for (G, c). Then $\mathrm{OPT}(G, c) = c(X^*) = c'(X^*) + c''(X^*) \geqslant \mathrm{OPT}(G, c') + \mathrm{OPT}(G, c'')$. □

Besides the Local Ratio Lemma, the analysis of Algorithm 2 relies on two lemmas. The first lemma guarantees that the algorithm terminates. That is, the algorithm is always able to find a 7/3-good weighted induced subgraph in Step 14. Since the number of weighted graphs in $\mathcal{H}(G)$ is polynomial, Algorithm 2 in fact runs in polynomial time. The proof of this lemma, which is the true heart of the algorithm, is given in Section 5.

Lemma 4 (Key Lemma). *If G is not a disjoint union of cliques and does not contain true twins, then $\mathcal{H}(G)$ is nonempty.*

Combined with Lemma 4, our second lemma shows that Algorithm 2 is a 7/3-approximation algorithm.

Lemma 5. *Suppose that Algorithm 2 terminates given some weighted graph (G, c) as input, and outputs a set X. Then X is an inclusionwise minimal hitting set of G and $c(X) \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G, c)$.*

Algorithm 2 CLUSTER-VD-APX(G, c)

Input: (G, c) a weighted graph
Output: X an inclusionwise minimal hitting set of G

1: **if** G is a disjoint union of cliques **then**
2: $\quad X \leftarrow \varnothing$
3: **else if** there exists $u \in V(G)$ with $c(u) = 0$ **then**
4: $\quad G' \leftarrow G - u$
5: $\quad c'(v) \leftarrow c(v)$ for $v \in V(G')$
6: $\quad X' \leftarrow$ CLUSTER-VD-APX(G', c')
7: $\quad X \leftarrow X'$ if X' is a hitting set of G; $X \leftarrow X' \cup \{u\}$ otherwise
8: **else if** there exist true twins $u, u' \in V(G)$ **then**
9: $\quad G' \leftarrow G - u'$
10: $\quad c'(v) \leftarrow c(u) + c(u')$ for $v = u$; $c'(v) \leftarrow c(v)$ for $v \in V(G') \setminus \{u\}$
11: $\quad X' \leftarrow$ CLUSTER-VD-APX(G', c')
12: $\quad X \leftarrow X'$ if X' does not contain u; $X \leftarrow X' \cup \{u'\}$ otherwise
13: **else**
14: $\quad$ pick any $(H, c_H) \in \mathcal{H}(G)$
15: $\quad \lambda^* \leftarrow \max\{\lambda \mid \forall v \in V(H) : c(v) - \lambda c_H(v) \geqslant 0\}$
16: $\quad G' \leftarrow G$
17: $\quad c'(v) \leftarrow c(v) - \lambda^* c_H(v)$ for $v \in V(H)$; $c'(v) \leftarrow c(v)$ for $v \in V(G) \setminus V(H)$
18: $\quad X \leftarrow$ CLUSTER-VD-APX(G', c')
19: **end if**
20: return X

Proof. The proof is by induction on the number of recursive calls. If the algorithm does not call itself, then it returns the empty set and in this case the statement trivially holds. Now assume that the algorithm calls itself at least once and that the output X' of the recursive call is an inclusionwise minimal hitting set of G' that satisfies $c'(X') \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G', c')$. There are three cases to consider.

Case 1: the recursive call occurs at Step 6. Then we have $c(X) = c'(X')$ and $\mathrm{OPT}(G, c) = \mathrm{OPT}(G', c')$ because (G', c') is simply (G, c) with one zero-cost vertex removed. By construction, X is an inclusionwise minimal hitting set of G. Moreover, by what precedes, $c(X) = c'(X') \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G', c') = \frac{7}{3} \cdot \mathrm{OPT}(G, c)$.

Case 2: the recursive call occurs at Step 11. Again, X is an inclusionwise minimal hitting set of G and $c(X) = c'(X') \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G', c') = \frac{7}{3} \cdot \mathrm{OPT}(G, c)$, where the last equality holds by Lemma 2.

Case 3: the recursive call occurs at Step 18. In this case, $G = G'$ and $X = X'$, thus X is automatically an inclusionwise minimal hitting set of G. Let c'' denote the weighting c_H extended to $V(G)$ by letting $c''(v) := 0$ for $v \in V(G) \setminus V(H)$. We have $c'(X) \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G, c')$ by induction and $\lambda^* c''(X) \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G, \lambda^* c'')$ since all the weighted induced subgraphs (H, c_H) in $\mathcal{H}(G)$ are 7/3-good in G (Lemma 1). Because $c = c' + \lambda^* c''$, Lemma 3 implies $c(X) \leqslant \frac{7}{3} \cdot \mathrm{OPT}(G, c)$. □

We are now ready to prove our main result.

Proof (of Theorem 1). By Lemmas 4 and 5, Algorithm 2 is a 7/3-approximation algorithm for Cluster-VD. □

5 Finding a 7/3-good Induced Subgraph

The aim of this section is to prove the Key Lemma, Lemma 4, which states that $\mathcal{H}(G)$ is nonempty for all twin-free graphs G that are not a disjoint union of cliques. Our approach is as follows: We consider a twin-free graph G such that our collection $\mathcal{H}(G)$ of 7/3-good induced subgraphs is empty. We first prove that, in this case, G contains no claw, then no gem, and then no cycle of length at least 4 as an induced subgraph. At that point, from a result of Kloks, Kratsch, and Müller [11], we know that G is the line graph of an acyclic multigraph, from which we show that G does not contain any P_3, as desired.

Lemma 6. *Let G be a twin-free graph such that $\mathcal{H}(G)$ is empty. Then G is claw-free.*

Proof. Assume that G contains a claw, say on the vertex set $\{x, u, v, w\}$, where x is the central vertex. Since $G[x, u, v, w]$ is not 2-good in G, there exists a neighbor y of x that is distinct from u, v and w. If y sees none of u, v, w, then $G[\{x, y, u, v, w\}]$ is a $K_{1,4}$, a contradiction. Thus we may assume that yu is an edge.

Suppose that yv is an edge. If yw is not and edge, then $G[\{x, y, u, v, w\}]$ is a dart, a contradiction, and thus yw is an edge. Since G is twin-free, there must be a vertex z in the symmetric difference $N(x)\Delta N(y)$. By symmetry, we may assume that $z \in N(x) \setminus N(y)$.

Since G is $K_{1,4}$-free, the set $\{z, u, v, w\}$ is not stable, and hence $|N(z) \cap \{u, v, w\}| \geqslant 1$. If $|N(z) \cap \{u, v, w\}| \geqslant 2$, say both zu and zv are edges, then $G[\{y, z, u, v\}]$ is a C_4, a contradiction. So, $|N(z) \cap \{u, v, w\}| = 1$, and we may assume that zw is an edge. But now $G[\{x, y, z, u, v\}]$ is a dart, a contradiction.

Summing up, we conclude that yv is not an edge and, by symmetry, yw is not an edge.

Since u and y are not true twins in G, there is some vertex z' in the symmetric difference $N(u)\Delta N(y)$. By symmetry, we may assume that $z' \in N(y) \setminus N(u)$. If xz' is not an edge, then z' sees none of v, w, because G is C_4-free. But then the graph $G[\{x, y, z', u, v, w\}]$ is a turtle, a contradiction. Hence, xz' is an edge.

To avoid an induced dart on the vertex sets $\{x, y, z', u, v\}$ or $\{x, y, z', u, w\}$, both vz' and wz' must be edges. But then $G[\{x, z', u, v, w\}]$ is a dart, a contradiction. This completes the proof. □

Lemma 7. *Let G be a twin-free graph such that $\mathcal{H}(G)$ is empty. Then G is gem-free.*

Proof. By Lemma 6, we know that G is claw-free. For a contradiction, assume that G contains a gem.

Let k be the maximum number of vertices of a gem contained in G that have a common neighbor outside of that gem, this maximum being taken over all gems contained in G.

Consider an induced gem in G, say with vertex set $\{v_1, v_2, v_3, v_4, v_5\}$, such that there is some vertex v outside of that gem with exactly k neighbors in the set $\{v_1, v_2, v_3, v_4, v_5\}$. Assume that the gem is made of the 5-cycle $v_1v_2v_3v_4v_5v_1$ and the two edges v_1v_3, v_1v_4. Now, we will distinguish some cases depending on the value of k. Notice that $k \geqslant 1$ since the gem $G[\{v_1, v_2, v_3, v_4, v_5\}]$ is not in $\mathcal{H}(G)$.

Case 1: $k = 5$. Since v and v_1 are not true twins in G, we may assume that there is some vertex u that sees v_1 and not v.

Case 1.1: uv_2 *is an edge.* Then neither uv_4 nor uv_5 is an edge of G, for otherwise $G[\{u, v_2, v, v_4\}]$ or $G[\{u, v_2, v, v_5\}]$ is a C_4. Moreover, uv_3 is an edge, since otherwise $G[\{v_1, u, v_3, v_5\}]$ is a claw. But now the graph $G[\{v_1 \ldots, v_5\} \cup \{v, u\}]$ is isomorphic to the special graph H_1 (see Fig. 1), and thus belongs to $\mathcal{H}(G)$, a contradiction.

Case 1.2: uv_2 *is not an edge.* Then uv_4 is an edge for otherwise $G[\{v_1, u, v_2, v_4\}]$ is a claw, and similarly uv_5 is an edge for otherwise $G[\{v_1, u, v_2, v_5\}]$ is a claw. Moreover, uv_3 is not an edge, because otherwise $G[\{u, v_3, v, v_5\}]$ is a C_4. But again $G[\{v_1, \ldots, v_5\} \cup \{v, u\}]$ is isomorphic to H_1 as in Case 1.1, a contradiction.

Case 2: $k = 4$. We may assume that v sees v_1, v_2, v_3, v_4 and not v_5. Otherwise, by symmetry, we may assume that v sees either all of v_2, v_3, v_4, v_5 and $G[\{v, v_2, v_1, v_5\}]$ is a C_4, or that v sees all of v_1, v_2, v_4, v_5 and $G[\{v, v_2, v_3, v_4\}]$ is a C_4. Since v and v_3 are not true twins, we may assume that there is a vertex u that sees v but not v_3. In this case, G contains the trigraph H_3 (see Fig. 1), a contradiction.

Case 3: $k = 3$. Without loss of generality, v sees v_1, v_2, v_3, because every other (that is, non-isomorphic) possibility leads to a contradiction. Indeed, if v sees v_1, v_2, and v_5, then $G[\{v_1, \ldots, v_5, v\}]$ is a W_5. If v sees v_3, v_4, and v_5, then $G[\{v, v_3, v_1, v_5\}]$ is a C_4, and similarly we have a C_4 if v sees v_2, v_4, and v_5, or v_1, v_3, and v_5. Moreover, if v sees v_1, v_3, and v_4, the dart $G[\{v, v_1, v_2, v_3, v_5\}]$ appears.

Since v and v_2 are not true twins, we may assume that there is a vertex u seeing v but not v_2. We get a contradiction, since G contains the trigraph H_4 (see Fig. 1).

Case 4: $k \leqslant 2$. Since $G[\{v_1, v_2, v_3, v_4, v_5\}]$ is not in $\mathcal{H}(G)$, we know that there is a neighbor w of v_1 outside the gem, and w has at most one neighbor in the set $\{v_2, \ldots, v_5\}$. If w sees neither v_3 nor v_4, then w is also non-adjacent to at least one of v_2, v_5, say v_2 using symmetry, and the graph $G[\{w, v_1, v_2, v_3, v_4\}]$ is a dart. Hence, we may assume that either wv_2 or wv_3 is an edge, say wv_2 by symmetry. Then $G[\{w, v_1, v_3, v_4, v_5\}]$ is a dart, a contradiction. This completes the proof. □

In the following, we need another small graph: a *diamond*, that is, a C_4 plus a crossing edge.

Lemma 8. *Let G be a twin-free graph such that such that $\mathcal{H}(G)$ is empty. Then G is diamond-free and K_4-free.*

Proof. We first prove that G is diamond-free. Suppose we have a diamond on the vertices u, v, w, and x, where ux is not an edge. By assumption, v and w are not true twins, and so we may assume that there is some $y \in N(v) \setminus N(w)$. To avoid a claw on the vertices u, v, x, and y, it must be that uy or xy is an edge. Both edges cannot be there, since otherwise $G[\{u, y, x, w\}]$ is a C_4. So we may assume that uy is an edge while xy is not. But now the graph $G[\{u, v, w, x, y\}]$ is a gem, which contradicts Lemma 7.

Next we prove that G is K_4-free. Assume not, and let u, v, w and x be four mutually adjacent vertices. Since u and x cannot be true twins, we may assume that there is some vertex u' adjacent to u but not to x. Since G is diamond-free, the only neighbor of u' in $\{u, v, w, x\}$ is u. Similarly, we obtain vertices v' and w', where v' is only adjacent to v in $\{u, v, w, x\}$ and w' is only adjacent to w in $\{u, v, w, x\}$. As G is C_4-free, the three vertices u', v', w' are pairwise non-adjacent. But now the graph $G[\{u, v, w, x, u', v', w'\}]$ is isomorphic to the special graph H_2 (see Fig. 1), a contradiction. □

A *hole* in a graph is an induced cycle of length at least four.

Lemma 9. *Let G be a twin-free graph such that $\mathcal{H}(G)$ is empty. Then G is hole-free.*

Proof. Thanks to Lemmas 6 and 8, we know that G is claw-free, diamond-free and K_4-free. By contradiction, assume that G contains a hole and let $H = v_1v_2 \dots v_kv_1$ be a shortest hole contained in G. Thus $k \geqslant 5$. If some vertex of H does not have a neighbor in $V(G) \setminus V(H)$, then G contains an induced P_3 whose middle vertex does not have neighbors outside the P_3, in contradiction to the assumption that $\mathcal{H}(G)$ is empty.

There cannot be a vertex outside of H having exactly one neighbor in H either, as G is claw-free. Moreover, if there is a vertex v outside of H having two or more neighbors H, they must appear consecutively. This is due to our assumption that H is a shortest hole in G, and also to the fact that G does not contain a C_4. Since G is diamond-free, this means that every vertex outside H that sees some vertex of H has exactly two neighbors in H, and they must appear consecutively on H.

Let $u \in V(G) \setminus V(H)$ be a neighbor of v_2. We may assume that u is adjacent to v_3 too. Since there is a bull on the vertices v_1, v_2, v_3, v_4, and u, there must be another neighbor v of v_2 outside of H. Note that uv is not an edge, because we cannot have a diamond or a K_4 on u, v, v_2, and v_3. Similarly, vv_3 is not an edge. Hence, v must be adjacent to v_1. But now $G[\{v_k, v_1, v_2, v_3, v_4, u, v\}]$ contains the trigraph H_5 (Fig. 1), the undecided edge being included if $k = 5$, and excluded if $k > 5$. This is a contradiction, which concludes the proof. □

To finalize the proof, we need the following theorem.

Theorem 2 (Kloks, Kratsch, and Müller [11]). *Let G be a graph that is hole-, claw- and gem-free. Then G is the line graph of an acyclic multigraph.*

Now, we are ready to prove our key lemma.

Proof (Proof of Lemma 4). We may assume that G is connected. (Indeed, if not, simply consider a component.) So $|V(G)| \geqslant 3$. By Lemmas 6, 7 and 9, Theorem 2 gives that G is a K_4-free line graph of an acyclic multigraph, say H. Since G is twin-free, H does not have parallel edges. Also H is clearly connected, thus H is a tree.

Root the tree H at an arbitrary vertex, and let x be a leaf at maximum distance from the root. Let y be the parent of x in H. Suppose that y has a child z distinct from x. Then z is also a leaf. However, the vertices of G corresponding to edges xy, zy are true twins in G, which is not possible. Hence x is the only child of y in H. But now the edge xy is a vertex v of degree one in G and, since $|V(G)| \geqslant 3$, the vertex v is contained in an induced P_3. Since v has no neighbor outside this P_3, this is a contradiction. This completes the proof. □

6 Conclusion

In this paper we presented a 7/3-approximation algorithm for the Cluster-VD problem, based on the local ratio technique. The main idea underlying the algorithm is that there exists a collection of small induced subgraphs that are on the one hand *good* in the sense that they guarantee a local ratio of at most 7/3, and on the other hand *sufficient* in the sense that one can always find and use one of them until the algorithm terminates.

As mentioned in the introduction, we conjecture that there is a 2-approximation algorithm for Cluster-VD. The following result gives some evidence to back up this conjecture.

Theorem 3. *There is a 2-approximation algorithm for* Cluster-VD *in the class of diamond-free graphs.*

The proof of Theorem 3 is given in the Appendix. The algorithm is modeled on the 7/3-approximation algorithm presented earlier, the main difference being the use of some infinite (but easy to detect) family of graphs that are 2-good. We note that Theorem 3 can be seen as a generalization of the fact that there is a 2-approximation for Cluster-VD in triangle-free graphs, a result that was used by You, Wang, and Cao [14] in their 5/2-approximation algorithm for (unweighted) Cluster-VD.

Finally, we point out that the analysis of our 7/3-approximation algorithm also proves that a certain $O(n^7)$-size LP relaxation for Cluster-VD has integrality gap at most 7/3, namely, the LP relaxation obtained by writing down at most $O(n^7)$ inequalities in the vertex variables x_v for each of the weighted graphs of Fig. 1. By considering graphs G with large girth and small stability number, we can see that the integrality gap is actually equal to 7/3, since in

these graphs $\mathrm{OPT}(G)$ is close to n and only the graphs $H \in \{P_3, K_{1,3}, K_{1,4}\}$ are induced subgraphs of G (see Lemma 10 in Appendix). Thus letting $x_v := 3/7$ for all vertices v gives a feasible fractional solution, of cost $3n/7$.

References

1. Hans-Jürgen Bandelt and Henry Martyn Mulder, *Distance-hereditary graphs*, J. Comb. Theory, Ser. B **41** (1986), no. 2, 182–208.
2. Reuven Bar-Yehuda, Keren Bendel, Ari Freund, and Dror Rawitz, *Local ratio: A unified framework for approximation algorithms*, ACM Comput. Surv. **36** (2004), no. 4, 422–463.
3. Anudhyan Boral, Marek Cygan, Tomasz Kociumaka, and Marcin Pilipczuk, *A fast branching algorithm for cluster vertex deletion*, Computer Science - Theory and Applications (Edward A. Hirsch, Sergei O. Kuznetsov, Jean-Éric Pin, and Nikolay K. Vereshchagin, eds.), Lecture Notes in Computer Science, vol. 8476, Springer International Publishing, 2014, arXiv:1306.3877, pp. 111–124.
4. Mao cheng Cai, Xiaotie Deng, and Wenan Zang, *An approximation algorithm for feedback vertex sets in tournaments*, SIAM Journal on Computing **30** (2001), no. 6, 1993–2007.
5. Fabián A Chudak, Michel X Goemans, Dorit S Hochbaum, and David P Williamson, *A primal–dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs*, Operations Research Letters **22** (1998), no. 4, 111–118.
6. Reinhard Diestel, *Graph theory*, third ed., Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Berlin, 2005.
7. Venkatesan Guruswami and Euiwoong Lee, *Inapproximability of feedback vertex set for bounded length cycles*, ECCC:TR14-006.
8. Venkatesan Guruswami and Euiwoong Lee, *Inapproximability of H-transversal/packing*, arXiv:1506.06302.
9. Falk Hüffner, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier, *Fixed-parameter algorithms for cluster vertex deletion*, Theory of Computing Systems **47** (2010), no. 1, 196–217.
10. Yoichi Iwata and Keigo Oka, *Fast dynamic graph algorithms for parameterized problems*, Algorithm Theory – SWAT 2014 (R. Ravi and Inge Li Gørtz, eds.), Lecture Notes in Computer Science, vol. 8503, Springer International Publishing, 2014, pp. 241–252 (English).
11. Ton Kloks, Dieter Kratsch, and Haiko Müller, *Dominoes*, Graph-Theoretic Concepts in Computer Science, 20th International Workshop, WG '94, Herrsching, Germany, June 16-18, 1994, Proceedings, 1994, pp. 106–120.
12. Matthias Mnich, Virginia Vassilevska Williams, and László A. Végh, *A 7/3-approximation for feedback vertex sets in tournaments*, arXiv:1511.01137.
13. Jianhua Tu and Wenli Zhou, *A primal–dual approximation algorithm for the vertex cover P_3 problem*, Theoretical Computer Science **412** (2011), no. 50, 7044–7048.
14. Jie You, Jianxin Wang, and Yixin Cao, *Approximate association via dissociation*, arXiv:1510.08276.

A Appendix

A.1 Proof of Lemma 1

Proof (of Lemma 1). Let c_H be as shown on Fig. 1. Let X be an inclusionwise minimal hitting set of G. We prove that, in each of the cases, (1) is satisfied for some $\alpha \leqslant 7/3$, from which it follows that (H, c_H) is 7/3-good in G.

(i) Let $\alpha = 7/3$. It can be easily verified that $\sum_{v \in V(H)} c_H(v) \leqslant \alpha \cdot \mathrm{OPT}(H, c_H)$ holds in each of the seven subcases (in fact, one can even take $\alpha = 2$ in the first three cases), which implies (1).

(ii) Let $\alpha = 7/3$. We claim that $\mathrm{OPT}(H, c_H) = \mathrm{OPT}(H) = 3$ for any instantiation H of H_3. To see this, suppose that H has a hitting set Y of size at most 2. We give the following names to the vertices of H, see Fig 1. Let u denote the leftmost vertex and v denote the highest vertex. Thus $H - \{u, v\}$ is a gem. We name the vertices of this gem $v_1, \ldots, v_5$ in counter-clockwise order, starting from the highest vertex.

Since every hitting set of H must contain two vertices in the gem $H - \{u, v\}$, we have $v_3 \in Y$, because of the P_3 induced by $\{u, v, v_3\}$, and also $\{v_4, v_5\} \cap Y \neq \varnothing$, because of the P_3 induced by $\{v, v_4, v_5\}$. But now $H - Y$ contains either of the P_3's induced by $\{v_2, v_1, v_4\}$ or $\{v_2, v_1, v_5\}$, a contradiction. Thus the claim holds. We conclude that $\mathrm{OPT}(H, c_H) = 3$, which implies (1).

The second and third cases, when H is an instantiation of H_4 or H_5, is similar and left to the reader.

(iii) Let $\alpha = 2$. This time we have $\sum_{v \in V(H), v \neq w} c_H(v) \leqslant \alpha \cdot \mathrm{OPT}(H, c_H)$ for any $w \in V(H)$, in each of the three subcases. By minimality of X, the intersection of X and $V(H)$ is a proper subset of $V(H)$. Thus there is some vertex $w \in V(H)$ such that $w \notin X$. By what precedes, this implies (1). □

A.2 Proof of Theorem 3

Proof (of Theorem 3, **sketch***).* Let (G, c) denote the diamond-free weighted graph given in input. We may assume that G is connected but not a clique. We may further assume that G contains a triangle (otherwise, simply use Algorithm 1 for hitting P_3-subgraphs described in the introduction). Instead of fully describing the 2-approximation algorithm, we will limit ourselves in this sketch of proof to showing how to detect 2-good induced subgraphs when there are no true twins in G. These are then used exactly as in our 7/3-approximation algorithm.

Assume thus that G has no true twins. Consider a connected subgraph H of G which is such that every edge of H is contained in a triangle in G, and inclusion-wise maximal with this property. Note that H is not necessarily an induced subgraph of G. However, H must be diamond-free: If there is a diamond $H[\{u, v, w, x\}]$ with $ux \notin E(H)$, it must be that $ux \in E(G)$ since G is diamond-free by assumption. But then ux is contained in a triangle in G and thus could have been added to H, a contradiction. Note also that the we may also assume

that H has been chosen so that it contains at least one triangle, since G has a triangle. It follows that every edge of H is in a triangle *in* H.

We may assume that G is C_4-free, as otherwise we have found a 2-good induced subgraph of G.

Suppose there is a hole $C = v_1v_2 \cdots v_kv_1$ in H. It might be that C has chords in G, but every such chord is contained in $E(G) \setminus E(H)$. By our choice of H, such a chord connects two vertices at distance at least 3 on C. For each edge v_iv_{i+1} of C, we pick a third vertex u_i adjacent to both endpoints of that edge. Such a vertex exists since C is a subgraph of H and every edge of H is contained in a triangle of G. Note that we might have $u_i = u_j$ for some $i \neq j$, and there might be edges between two distinct vertices u_i and u_j. Note also that the u_i's cannot be on the cycle C (indeed, otherwise some chord of C is in a triangle in G and thus could have been added to H).

Suppose that, for some $i \in \{1, \ldots, k\}$, we have $u_i = u_{i+1}$.[5] Then the subgraph of H induced by the vertices $v_i, v_{i+1}, u_i, v_{i+2}$ is a diamond in H, a contradiction. Moreover, $u_iu_{i+1} \notin E(G)$, since otherwise considering whether u_iv_{i+2} and/or $u_{i+1}v_i$ are edges of G, we get a diamond in G in all cases. This is because $v_iv_{i+2} \notin E(G)$, as argued above. Similarly, u_i is non-adjacent to v_{i+2} in G, since otherwise we again obtain a diamond in G. We leave to the reader the (not completely trivial) task of verifying that $V(C)$ together with the vertices u_i, $i = 1, \ldots, k$ induces a subgraph of G which is 2-good in G (when putting unit weights on its vertices).

We can test in polynomial time whether H contains a hole, and if it does find the corresponding 2-good induced subgraph. Thus we may assume that H is hole-free, that it is, chordal. But H is also diamond-free, and so H is a *block graph*: Every block of H is a clique [1].

Choose an end-block B of H. Thus B is a clique and exactly one vertex of B, say x, may have neighbors outside of B in H. Let us say $B = \{x, u_1, u_2, \ldots, u_k\}$. Note that $|B| \geqslant 3$, since every edge of H is in a triangle of H. Also, we remark that all vertices in $\{u_1, u_2, \ldots, u_k\}$ have some neighbor outside B in G, except perhaps for one. Indeed, otherwise two of them would be true twins in G.

Let x' be a neighbor of x in G that is not in B. (The vertex x' exists since G is not a clique.) Consider the subgraph B' of G induced by the union of $V(B) \cup \{x'\}$ with all vertices of $V(G) - V(B)$ that have a neighbor in $\{u_1, \ldots, u_k\}$.

We first observe the following facts which hold for all $i \in \{1, \ldots, k\}$. First, x' is not adjacent to u_i, since otherwise the edge $x'u_i$ should have been added to H because of the triangle $\{x, x', u_i\}$. Second, each neighbor of u_i outside B cannot see any other vertex of B, for a similar reason. Third, the neighbors of u_i outside B form a stable set (again, because of the maximality of H). Fourth, since G is C_4-free, every two vertices outside B that see different vertices in $\{u_1, \ldots, u_k\}$ are not adjacent.

It follows that B' is an induced subgraph of G consisting of a clique B plus a collection of pendent vertices attached to that clique. Moreover, B' contains all neighbors in G of each of the vertices $u_1, \ldots, u_k$.

[5] We assume that all indices are taken cyclically, e.g., $u_{k+1} = u_1$.

Define a weighting $c_{B'}$ of the vertices of B' as follows: Each vertex gets a weight of 1, except for $u_1, \ldots, u_k$: Each u_i is assigned its number of neighbors outside B as weight. Observe that $c_{B'}(V(B')) \geqslant 4$, since $k \geqslant 2$ and at most one vertex from $\{u_1, \ldots, u_k\}$ has no neighbor outside B. We claim that $(B', c_{B'})$ is 2-good in G. It can easily be checked that

$$\mathrm{OPT}(B', c_{B'}) = c_{B'}(\{u_1, \ldots, u_k\}) = \frac{c_{B'}(V(B')) - 2}{2}.$$

However, for any inclusionwise minimal hitting set X of G it holds that $c_{B'}(X \cap V(B')) \leqslant c_{B'}(V(B')) - 2$ because $|B| \geqslant 3$ and none of the vertices $u_1, \ldots, u_k$ has neighbors outside B'. The claim follows. □

A.3 Lower Bounding the Integrality Gap

We provide here details on how to construct the graphs used in the Conclusion to show that the integrality gap of the LP relaxation is at least 7/3. These are obtained straight from Erdős' "high χ high girth" proof.

Lemma 10. *For every integer $k \geqslant 1$, there exists an n-vertex graph G with girth at least k and $\mathrm{OPT}(G) \geqslant (1 - 1/k)n$.*

Proof. Fix some ϵ with $0 < \epsilon < 1/4k$. Consider the Erdős-Renyi random graph model $\mathcal{G}(n, p)$ with $p := n^{\epsilon}/n$. As in Erdős' proof (see e.g. Diestel [6]), we note that the following two probabilities tend to 0 as $n \to \infty$:

- probability that $G \in \mathcal{G}(n, p)$ has a stable set of size at least $n/4k$;
- probability that $G \in \mathcal{G}(n, p)$ has at least $n/2$ cycles of length at most k.

By the union bound, for n large enough we can find an n-vertex graph G with stability number less than $n/4k$, and less than $n/2$ cycles of length at most k. We can kill the latter cycles by removing at most $n/2$ vertices, giving a graph H with girth more than k and stability number less than $n/4k \leqslant |V(H)|/2k$.

Now, observe that if $X \subseteq V(H)$ induces a P_3-free subgraph of H, then H has a stable set of size at least $|X|/2$ since H is triangle free. Thus, every such set X has size at most $|V(H)|/k$. It follows that $\mathrm{OPT}(H) \geqslant (1 - 1/k)|V(H)|$, as desired. □