Mathematical Programming (2020) 182:355–367
https://doi.org/10.1007/s10107-019-01395-y

FULL LENGTH PAPER

Series A

Improved approximation algorithms for hitting 3-vertex paths

Samuel Fiorini[1] · Gwenaël Joret[2] · Oliver Schaudt[3]

Received: 30 August 2018 / Accepted: 13 April 2019 / Published online: 22 April 2019

Abstract
We study the problem of deleting 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 often called *cluster vertex deletion* in the literature and admits a straightforward 3-approximation algorithm since it is a special case of the vertex cover problem on a 3-uniform hypergraph. Recently, You, Wang, and Cao described an efficient 5/2-approximation algorithm for the unweighted version of the problem. Our main result is a 9/4-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 contrast 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.

Mathematics Subject Classification 05C85 · 90C27 · 90C59 · 68W25

S. Fiorini is supported by ERC Consolidator Grant 615640-ForEFront. G. Joret is supported by an ARC grant from the Wallonia-Brussels Federation of Belgium.
A preliminary version of this paper appeared as an extended abstract in the Proceedings of the 18th Conference on Integer Programming and Combinatorial Optimization (IPCO ’16) [5].

✉ Gwenaël Joret
gjoret@ulb.ac.be

Samuel Fiorini
sfiorini@ulb.ac.be

Oliver Schaudt
schaudto@uni-koeln.de

[1] Département de Mathématique, Université Libre de Bruxelles, Brussels, Belgium

[2] Département d’Informatique, Université Libre de Bruxelles, Brussels, Belgium

[3] Institut für Informatik, Universität zu Köln, Cologne, Germany

1 Introduction

Graphs in this paper are finite, simple, and undirected. Given a graph 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 each component of $G - X$ is a complete graph. 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.

The problem admits a staightforward 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 also 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 pendant edge to every vertex. Then solving Vertex Cover on H is equivalent to solving Cluster- VD on G. Hence, known hardness and inapproximability results for Vertex Cover apply to Cluster- VD as well, and in particular it is UGC-hard to approximate Cluster- VD to within any ratio better than 2. We show that we can however come close to 2.

Theorem 1 Cluster- VD *admits a 9/4-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 Sect. 5, where we notice that our 9/4-approximation algorithms is in fact a 2-approximation algorithm for the case where the largest clique in the input graph has size at most 4, and can be easily modified to a 2-approximation algorithm if the input graph does not contain any 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 [6] (see also Guruswami and Lee [7] for related inapproximability results).

1.1 Previous work

Cluster- VD was previously mostly studied in terms of fixed parameter algorithms. Hüffner et al. [8] first gave an $O(2^k k^9 + nm)$-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. [2], who gave a $O(1.9102^k(n + m))$-time algorithm. See also Iwata and Oka [9] for related results in the fixed parameter setting.

As for approximation algorithms, nothing better than a 3-approximation was known until the recent work of You et al. [12], who showed that the unweighted version of Cluster- VD admits a 5/2-approximation algorithm.

In a previous version of this paper [5], we gave a 7/3-approximation algorithm for Cluster- VD. The algorithm in this version of the paper achieves a better approximation ratio and is at the same time much simpler.

Finally, we note that there has been 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 et al. [3] was the best known for many years, until the very recent work of Mnich et al. [10] who found a 7/3-approximation algorithm for the problem.

1.2 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.

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 $d(u) \geq 2$, 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 $\lambda$ s.t. $c(v) - \lambda c_H(v) \geq 0$ for all $v \in V(H)$
      $X \leftarrow$ HITTING- $P_3$- SUBGRAPHS- APX$(G, c - \lambda^* c_H)$
  end if
  return $X$
```

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 the optimum cost for the weighted star (H, c_H) is $d(u) - 1$ while the solution X returned by the algorithm misses at least one of the vertices of the star, and thus has 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. [4] for the feedback vertex set problem. (Indeed, this is exactly what was done by Tu and Zhou [11].) However, the resulting algorithm is much more complicated than 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 et al. [12]. 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 9/4-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 induced subgraph H and decrease the weights of its vertices in (G, c) proportionally to a carefully chosen weighting c_H for the vertices of H, ensuring a local ratio of 9/4. (We remark that c_H depends on H only and is thus independent of the weights of vertices in G, similarly as in Algorithm 1.)

The induced subgraphs we consider are as follows: *cycles of length* 4 (C_4's), 5-*cliques plus distinguishing sets* (K_5's plus distinguishing sets), and *second-neighborhood* subgraphs induced by the vertices at distance at most two from a maximum degree vertex of G. We note that the approximation algorithm in the preliminary version of this paper [5] has the same general structure but exploits a different set of induced subgraphs, namely a finite (but longish) list of graphs on at most 7 vertices. Using the new set of induced subgraphs results in both simpler proofs and a better approximation ratio of 9/4.

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 (cost) of a hitting set of G. As expected, the *weight* (or *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.

3 Tools

3.1 True twins and distinguishers

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 is our first 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 $v \in V(G')$, $v \neq u$ and $c'(v) := c(u) + c(u')$ if $v = u$. Then* $\mathrm{OPT}(G, c) = \mathrm{OPT}(G', c')$.

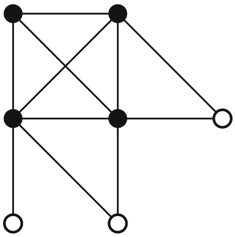

Fig. 1 The white vertices form a distinguishing set for the black vertices

Proof We have $\mathrm{OPT}(G, c) \leq \mathrm{OPT}(G', c')$ 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. Here we use that no induced P_3 in G contains both u and u'.

Conversely, we have $\mathrm{OPT}(G', c') \leq \mathrm{OPT}(G, c)$ 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 G is *twin-free*.

Notice that two adjacent vertices u and v are *not* true twins if and only if G has an induced P_3 containing u and v. The third vertex of such a P_3 is adjacent to one of u and v, and nonadjacent to the other. We say that it is a *distinguisher* for the edge uv, and call the induced P_3 a *distinguishing* P_3.

Now let $S \subseteq V(G)$. A set $D \subseteq V(G)$ disjoint from S is said to be a *distinguishing set* for S if for every edge uv whose endpoints are true twins in $G[S]$, the set D contains a distinguisher w for the edge uv. See Fig. 1 for an illustration.

Lemma 3 *Let H be a graph whose vertex set is partitioned into a clique C and a distinguishing set D for C. Then, there exists a weight function $c_H : V(H) \to \mathbb{Z}_{\geq 0}$ such that $c_H(v) = 1$ for all $v \in C$, $\sum_{v \in D} c_H(v) = |C| - 1$ and every set $X \subseteq V(H)$ hitting each distinguishing P_3 has weight $c_H(X) \geq |C| - 1$. In particular,* $\mathrm{OPT}(H, c_H) \geq |C| - 1$.

Proof First, we claim that for every fixed $w \in D$, the set of edges uv of C that are distinguished by w and by no other vertex of D forms a matching. Indeed, assume that C has two incident edges uv and uv' that are distinguished by w but are not distinguished by any other vertex of D. Then either w is adjacent to both v and v', or to none of them. Thus w does not distinguish the edge vv'. Let $w' \in D$ be any vertex distinguishing the edge vv'. Then w' is a distinguisher of uv or uv' that is distinct from w, a contradiction.

Next, we define the weight function c_H by the following iterative procedure.

- Pick any distinguisher $w \in D$.
- Let M denote the edges of C that are distinguished by w and by no other vertex of D. By the claim, M is a matching. Define $c_H(w) := |M|$.
- Let U be any set of $|M|$ vertices hitting each edge of M exactly once. Delete the vertices of U from C, delete w from D, and repeat until there are no more vertices in D.

Notice that at each step D remains a distinguishing set for C. Notice also that the graph obtained after deleting w from the distinguishing set and U from the clique does not depend on the particular choice of U. Indeed, all the possible choices for U lead to isomorphic graphs since w gets deleted.

Finally, we show that every set $X \subseteq V(H)$ hitting all the distinguishing P_3's has weight at least $|C| - 1$, by induction.

Let w denote the first distinguisher picked by the weighting procedure and the corresponding set U. If $w \in X$, consider the reduced instance $C' := C \backslash U$, $D' := D \backslash \{w\}$. It is true that $X - w$ hits all the distinguishing P_3's for this new instance. By induction, we get $c_H(X) = c_H(X \backslash \{w\}) + c_H(w) \geq |C'| - 1 + c_H(w) = |C| - c_H(w) - 1 + c_H(w) = |C| - 1$.

Now assume that $w \notin X$. Thus X meets each edge of M at least once. Let $R \subseteq X$ be any set meeting each edge of M exactly once. By the remark above, we may assume that $U = R$. As before, consider the reduced instance $C' := C \backslash U$, $D' := D \backslash \{w\}$. Clearly, $X \backslash U$ hits all the distinguishing P_3's for this instance. By induction, we get $c_H(X) = c_H(X \backslash U) + c_H(U) = c_H(X \backslash U) + c_H(w) \geq |C'| - 1 + c_H(w) = |C| - c_H(w) - 1 + c_H(w) = |C| - 1$. □

3.2 *α*-Good induced subgraphs

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

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

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.

We start by considering two different types of weighted induced subgraphs (H, c_H) that satisfy the stronger condition $\sum_{v \in V(H)} c_H(v) \leq \alpha \cdot \mathrm{OPT}(H, c_H)$, which obviously implies that they are α-good.

Lemma 4 *Let G be a graph. If H is an induced C_4 in G, then H is 2-good.*

Proof We let $c_H(v) := 1$ for all $v \in V(H)$. Then $\mathrm{OPT}(H, c_H) = 2$ and

$$\sum_{v \in V(H)} c_H(v) = 4 = 2 \cdot \mathrm{OPT}(H, c_H).$$

□

Lemma 5 *Let G be a twin-free graph, let C be a 5-clique in G and let D be a distinguishing set for C. The induced subgraph $H := G[C \cup D]$ is α-good in G for $\alpha = 9/4$.*

Proof With the weight function c_H defined in Lemma 3, we have

$$\sum_{v \in V(H)} c_H(v) = |C| + |C| - 1 = 9 \leq (9/4) \cdot \mathrm{OPT}(H, c_H).$$

□

The next lemma is our main tool for constructing α-good weighted induced subgraphs for $\alpha = 2$. This time, we use the minimality of the hitting set X to establish α-goodness, however in a very simple way.

Lemma 6 *Let G be a graph that is twin-free, C_4-free and K_5-free. Let v_0 be a vertex of maximum degree, and let A_1, ..., A_k denote the components of $G[N(v_0)]$. For $i \in [k]$, let B_i denote the set of vertices in $G - (\{v_0\} \cup N(v_0))$ that see at least one vertex in A_i. Let H denote the subgraph of G induced by $\{v_0\} \cup N(v_0) \cup \bigcup_{i=1}^{k} B_i$. Then there exists a weight function $c_H : V(H) \to \mathbb{Z}_{\geq 0}$ such that (H, c_H) is 2-good in G.*

Proof Notice that since G is C_4-free, the sets B_i are pairwise disjoint.

In all cases except in one sporadic case (part of Case 1.3 below), we let $c_H(v) := 1$ for all $v \in N(v_0)$, that is, we put unit weight on these vertices. The weights on the vertices in $\{v_0\} \cup \bigcup_{i=1}^{k} B_i$ will be determined later.

Let X denote a minimal hitting set of G. We wish to show that (1) always holds for our choice of weights and $\alpha = 2$. We split the discussion into two cases according to the number of components of $G[N(v_0)]$. Each of these cases is split into several subcases according to the structure of the induced subgraphs $G[A_i]$, $i \in [k]$.

In all the cases, we make sure that the weight on v_0 is at least 1, and hence

$$\sum_{v \in X \cap V(H)} c_H(v) \leq \sum_{v \in V(H)} c_H(v) - 1.$$

This follows from the assumption that X is minimal: X has to exclude at least one of the vertices of $\{v_0\} \cup N(v_0)$, and each of these vertices has weight at least 1. In order to prove 2-goodness, it suffices then to show that $c_H(V(H)) \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Case 1 $k = 1$. Then $A_1 = N(v_0)$.

Case 1.1 *A_1 is a clique.* We let $c_H(v_0) := 1$ and use Lemma 3 on the clique $C = \{v_0\} \cup A_1$ and distinguishing set $D = B_1$ to define weights on B_1. We get $c_H(V(H)) = 2|C| - 1$ and $\mathrm{OPT}(H, c_H) \geq |C| - 1$, and thus $c_H(V(H)) \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Case 1.2 *A_1 is not a clique and $G[A_1]$ has clique number* 2. If $|A_1| \geq 4$, we let $c_H(v_0) := |A_1| - 3 \geq 1$ and $c_H(v) := 0$ for $v \in B_1$. Then $\mathrm{OPT}(H, c_H) \geq |A_1| - 2$. This can be seen as follows. Let Y denote a minimum weight hitting set of (H, c_H). Either Y contains v_0 and at least one vertex of A_1, or Y does not contain v_0 and $A_1 \backslash Y$ is a clique. In both cases the weight of Y is at least $|A_1| - 2$. We have $c_H(V(H)) = 2|A_1| - 3 \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Otherwise, $|A_1| = 3$ and $G[A_1]$ is a P_3. Let v_1 denote the middle vertex of this P_3. Since G is twin-free, v_0 and v_1 are not true twins. Thus, there exists a vertex $v_2 \in B_1$ that sees v_1 and not v_0. We put unit weights on v_0 and v_2, and zero weights on the vertices of $B_1 \backslash \{v_2\}$. We get $c_H(V(H)) = 5 \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Case 1.3 *A_1 is not a clique and $G[A_1]$ has clique number* 3. First, assume that $|A_1| \geq 6$ and the minimum size of a hitting set of $G[A_1]$ is at least 2. We let $c_H(v_0) := |A_1| - 5 \geq 1$ and $c_H(v) := 0$ for $v \in B_1$. By an argument similar to that used in Case 1.2, we have $\mathrm{OPT}(H, c_H) \geq |A_1| - 3$. Then $c_H(v(H)) = 2|A_1| - 5 \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Second, assume that there is a vertex v_1 that is a hitting set of $G[A_1]$. Because $G[A_1]$ is connected, not a clique, and does not contain any 4-clique, one can check that the following holds for the graph $G[A_1]$: (1) v_1 has no true twin, (2) every pair of true twins lie in a triangle, (3) every triangle contains a pair of true twins, and (4) every two pairs of true twins are vertex-disjoint and there is no edge between them.

There is at least one pair of true twins in $G[A_1]$ (since $G[A_1]$ has a triangle), and each pair of true twins in $G[A_1]$ is distinguished in G by some vertex in B_1. Moreover, every two such pairs are distinguished by distinct vertices in B_1, since G is C_4-free. So there is a nonempty set $B_1' \subseteq B_1$ with the following properties: (i) every pair of true twins in $G[A_1]$ has a distinguisher in B_1', (ii) there are $|B_1'|$ vertex-disjoint induced P_3's with one endvertex in B_1' and the other two vertices in $A_1 \backslash \{v_1\}$.

Assume for now that $|A_1| - |B_1'| - 3 \geq 1$. Then, we put a weight of $|A_1| - |B_1'| - 3$ on v_0, unit weights on the vertices of B_1' and zero weights on the vertices of $B_1 \backslash B_1'$. Consider a minimum weight hitting set Y of (H, c_H). Either Y contains v_0 and at least $1 + |B_1'|$ further vertices in $A_1 \cup B_1'$, or Y does not contain v_0 and contains at least $|A_1| - 2$ vertices in $A_1 \cup B_1'$. Therefore, we have $\mathrm{OPT}(H, c_H) \geq |A_1| - 2$ and $c_H(V(H)) = 2|A_1| - 3 \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Otherwise, $|A_1| - |B_1'| - 3 \leq 0$ and using $|A_1| \geq 4$, $|B_1'| \geq 1$ and $|A_1| \geq 2|B_1'| + 1$, we have $(|A_1|, |B_1'|) \in \{(4, 1), (5, 2)\}$. In both cases, A_1 contains a vertex v_2 (possibly $v_2 = v_1$) that sees every vertex in $A_1 \backslash \{v_2\}$, in addition to v_0. Since v_0 has maximum degree, v_2 has exactly the same neighbors as v_0, and is thus a true twin of v_0, a contradiction.

Finally, the last case to consider is when the minimum size of a hitting set in $G[A_1]$ is at least 2 and $|A_1| \leq 5$. Using that $G[A_1]$ is C_4-free, one can check that $|A_1| = 5$ in this case, and that the minimum size of a hitting set in $G[A_1]$ is exactly 2. Then the maximum degree in $G[A_1]$ is at most 3 since otherwise by maximality of its degree, v_0 would have a true twin in A_1. Since $G[A_1]$ contains at least one triangle and is C_4-free, this leaves only one possibility: $G[A_1]$ is a *bull*, that is, a triangle with two extra vertices of degree 1, say v_1 and v_2, each seeing a different vertex in the triangle. We increase the weight of one of these vertices to 2, say v_1, put a unit weight on v_0, an zero weights on B_1. Then $\mathrm{OPT}(H, c_H) = 3$ and $c_H(V(H)) = 7 \leq 2\,\mathrm{OPT}(H, c_H) + 1$.

Case 2 $k \geq 2$. In this case the weight on v_0 is set implicitly. Remember that we require

$$c_H(v_0) \geq 1. \tag{2}$$

For $i \in [k]$, we let OPT_i denote the minimum weight of a hitting set of $H[A_i \cup B_i]$, and OPT_i' denote the minimum weight of a hitting set of $H[\{v_0\} \cup A_i \cup B_i]$ not containing v_0. Notice that these quantities depend on the weight function c_H, which is not fully determined at this point.

We claim that the following lower bound holds on $\mathrm{OPT}(H, c_H)$, regardless of how $c_H(v)$ is chosen for $v \in \{v_0\} \cup \bigcup_{i=1}^{k} B_i$:

$$\mathrm{OPT}(H, c_H) \geq \min\left(\left\{c_H(v_0) + \sum_i \mathrm{OPT}_i\right\} \cup \left\{\sum_{i \neq j} |A_i| + \mathrm{OPT}_j' \mid j \in [k]\right\}\right).$$

In order to verify that this claim is true, consider a hitting set Y of H. If Y contains v_0, then $Y \cap (A_i \cup B_i)$ is a hitting set of $G[A_i \cup B_i]$ for each $i \in [k]$. In this case, $c_H(Y) \geq c_H(v_0) + \sum_i \mathrm{OPT}_i$. Otherwise, Y does not contain v_0. Then, there exists an index $j \in [k]$ such that Y contains A_i for all $i \neq j$. Moreover, $Y \cap (A_j \cup B_j)$ is a hitting set of $G[\{v_0\} \cup A_i \cup B_i]$ not containing v_0. In this case, $c_H(Y) \geq \sum_{i \neq j} |A_i| + \mathrm{OPT}'_j$.

Thanks to the above lower bound on $\mathrm{OPT}(H, c_H)$, it suffices to satisfy the following $1 + k$ inequalities in order to guarantee that (H, c_H) is 2-good (remember that we put unit weights over the A_i's, thus $c_H(A_i) = |A_i|$ for every $i \in [k]$):

$$\begin{aligned} c_H(v_0) + \sum_i (|A_i| + c_H(B_i)) &\leq 2\left(c_H(v_0) + \sum_i \mathrm{OPT}_i\right) + 1 \\ \iff c_H(v_0) &\geq \sum_i (|A_i| + c_H(B_i) - 2\ \mathrm{OPT}_i) - 1 \end{aligned} \tag{3}$$

and, for all $j \in [k]$,

$$\begin{aligned} c_H(v_0) + \sum_i (|A_i| + c_H(B_i)) &\leq 2\left(\sum_{i \neq j} c_H(A_i) + \mathrm{OPT}'_j\right) + 1 \\ \iff c_H(v_0) &\leq \sum_{i \neq j} (|A_i| - c_H(B_i)) \\ &\quad + 2\ \mathrm{OPT}'_j - |A_j| - c_H(B_j) + 1. \end{aligned} \tag{4}$$

By eliminating the variable $c_H(v_0)$ from the system (2)–(4), we get the following $2k$ inequalities not involving $c_H(v_0)$. For all $j \in [k]$:

$$|A_j| + c_H(B_j) \leq \sum_{i \neq j} (\mathrm{OPT}_i - c_H(B_i)) + \mathrm{OPT}_j + \mathrm{OPT}'_j + 1 \tag{5}$$

and

$$|A_j| + c_H(B_j) \leq \sum_{i \neq j} (|A_i| - c_H(B_i)) + 2\ \mathrm{OPT}'_j\,. \tag{6}$$

If (5) and (6) are satisfied for all $j \in [k]$, then (H, c_H) is 2-good.

In order to simplify these constraints, we add the extra requirements that $c_H(B_i) \leq \mathrm{OPT}_i$ and $c_H(B_i) \leq |A_i| - 1$ for all $i \in [k]$. Since $k \geq 2$ and $\mathrm{OPT}'_j \geq \mathrm{OPT}_j$, both (5) and (6) follow if, for all $j \in [k]$:

$$|A_j| + c_H(B_j) \leq 1 + \mathrm{OPT}_j + \mathrm{OPT}'_j\,. \tag{7}$$

Fix any $j \in [k]$. We set the weights on the vertices of B_j by inspecting the structure of the induced graph $H[A_j]$. We consider three subcases, see below. In each of these cases, it is straightforward to check that the two extra requirements are satisfied for $i = j$.

Case 2.1 A_j is a clique. By Lemma 3, we may set weights on B_j to have $c_H(B_j) = |A_j| - 1$ and $\mathrm{OPT}_j = |A_j| - 1$. Now $\mathrm{OPT}'_j \geq \mathrm{OPT}_j = |A_j| - 1$, so that inequality (7) is satisfied, since

$$\begin{aligned}|A_j| + c_H(B_j) &= |A_j| + |A_j| - 1 = 1 + (|A_j| - 1) + (|A_j| - 1)\\ &\leq 1 + \mathrm{OPT}_j + \mathrm{OPT}'_j\,.\end{aligned}$$

Case 2.2 A_j is not a clique and $G[A_j]$ has clique number 2. In this case, we put zero costs on B_j. We get $\mathrm{OPT}_j \geq 1$ because A_j is not a clique and also $\mathrm{OPT}'_j \geq |A_j| - 2$, so that

$$|A_j| + c_H(B_j) = |A_j| = 1 + 1 + (|A_j| - 2) \leq 1 + \mathrm{OPT}_j + \mathrm{OPT}'_j\,.$$

Case 2.3 A_j is not a clique and $G[A_j]$ has clique number 3. If the minimum size of a hitting set in $G[A_j]$ is at least 2, we put zero weights on B_j. Thus (7) is satisfied, since then $\mathrm{OPT}_j \geq 2$ and

$$|A_j| + c_H(B_j) = |A_j| \leq 1 + 2 + |A_j| - 3 \leq 1 + \mathrm{OPT}_j + \mathrm{OPT}'_j\,.$$

Now, assume that there exists some vertex v_1 that hits all the induced 3-paths in A_j. As in Case 1.3, we see that there is a set $B'_j \subseteq B_j$ with the following properties: (i) every pair of true twins in $G[A_j]$ has a distinguisher in B'_j, (ii) among the distinguishing P_3's defined by the vertices in B'_j, there are $|B'_j|$ vertex-disjoint P_3's.

We put unit weights on the vertices of B'_j and zero weight on the vertices of $B_j \backslash B'_j$. We get $\mathrm{OPT}_j \geq |B'_j| + 1$ since a hitting set in $G[A_j \cup B_j]$ has to have one vertex on each of the $|B'_j|$ vertex-disjoint distinguishing P_3's but this is not enough to hit all the induced P_3's. And also $\mathrm{OPT}'_j \geq |A_j| - 2$ since every triangle in $G[A_j]$ has one pair of true twins, which is distinguished by some vertex of B'_j. We have

$$\begin{aligned}|A_j| + c_H(B_j) &= |A_j| + |B'_j|\\ &\leq 1 + (|A_j| - 2) + (|B'_j| + 1) \leq 1 + \mathrm{OPT}_j + \mathrm{OPT}'_j\,.\end{aligned}$$

□

4 Algorithm

Our 9/4-approximation algorithm is described below, see Algorithm 2. Although we could have presented it 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. [1].

Lemma 7 (Local Ratio Lemma for CLUSTER- VD) *Let* (G, c) *be a weighted graph with* c *the sum of two cost functions* c' *and* c''*, and let* $\alpha \geq 1$*. If* X *is a hitting set of* G *such that* $c'(X) \leq \alpha \cdot \mathrm{OPT}(G, c')$ *and* $c''(X) \leq \alpha \cdot \mathrm{OPT}(G, c'')$*, then* $c(X) \leq \alpha \cdot \mathrm{OPT}(G, c)$*.*

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

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

```
Input: $(G, c)$ a weighted graph
Output: $X$ an inclusionwise minimal hitting set of $G$
if $G$ is a disjoint union of cliques then
    $X \leftarrow \varnothing$
else if there exists $u \in V(G)$ with $c(u) = 0$ then
    $G' \leftarrow G - u$
    $c'(v) \leftarrow c(v)$ for $v \in V(G')$
    $X' \leftarrow$ CLUSTER-VD-APX$(G', c')$
    $X \leftarrow X'$ if $X'$ is a hitting set of $G$; $X \leftarrow X' \cup \{u\}$ otherwise
else if there exist true twins $u, u' \in V(G)$ then
    $G' \leftarrow G - u'$
    $c'(v) \leftarrow c(u) + c(u')$ for $v = u$; $c'(v) \leftarrow c(v)$ for $v \in V(G')\backslash\{u\}$
    $X' \leftarrow$ CLUSTER-VD-APX$(G', c')$
    $X \leftarrow X'$ if $X'$ does not contain $u$; $X \leftarrow X' \cup \{u'\}$ otherwise
else
    pick the first $(H, c_H)$ in $\mathcal{H}(G)$
    $\lambda^* \leftarrow \max\{\lambda \mid \forall v \in V(H) : c(v) - \lambda c_H(v) \geq 0\}$
    $G' \leftarrow G$
    $c'(v) \leftarrow c(v) - \lambda^* c_H(v)$ for $v \in V(H)$; $c'(v) \leftarrow c(v)$ for $v \in V(G)\backslash V(H)$
    $X \leftarrow$ CLUSTER-VD-APX$(G', c')$
end if
return $X$
```

Algorithm 2 uses an ordered list $\mathcal{H}(G)$ of weighted induced subgraphs (H, c_H) of G as defined in Lemmas 4, 5 and 6. We order the weighted induced subgraphs (H, c_H) in $\mathcal{H}(G)$ in order to make sure that the hypotheses of the corresponding lemma are satisfied when (H, c_H) is used. The first elements of the list are induced C_4's (if any), next come the induced K_5's (if any) each of them taken together with a distinguishing set, and finally the second neighborhood of any maximum degree vertex v_0. Notice that the list $\mathcal{H}(G)$ is always nonempty and of polynomial size. This ensures that Algorithm 2 has polynomial complexity.

We are now ready to prove our main result.

Proof of Theorem 1 By induction on the number of recursive calls, we prove the following claim:

($\star$) *The set X output by Algorithm 2 on input (G, c) is an inclusionwise minimal hitting set of G and $c(X) \leq \frac{9}{4} \cdot \mathrm{OPT}(G, c)$.*

If the algorithm does not call itself, then it returns the empty set and in this case claim ($\star$) 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 $c'(X') \leq \frac{9}{4} \cdot \mathrm{OPT}(G', c')$. 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, $c(X) = c'(X') \leq \frac{9}{4} \cdot \mathrm{OPT}(G', c') = \frac{9}{4} \cdot \mathrm{OPT}(G, c)$.

Case 2 The recursive call occurs at Step 11. Again, X is an inclusionwise minimal hitting set of G and $c(X) = c'(X') \leq \frac{9}{4} \cdot \mathrm{OPT}(G', c') = \frac{9}{4} \cdot \mathrm{OPT}(G, c)$, 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 $c'(X) \leq \frac{9}{4} \cdot \mathrm{OPT}(G, c')$ by induction and $\lambda^* c''(X) \leq \frac{9}{4} \cdot \mathrm{OPT}(G, \lambda^* c'')$ since all the weighted induced subgraphs (H, c_H) in $\mathcal{H}(G)$ are 9/4-good in G (see Lemmas 4, 5 and 6). Because $c = c' + \lambda^* c''$, Lemma 7 implies $c(X) \leq \frac{9}{4} \cdot \mathrm{OPT}(G, c)$. □

5 Conclusion

In this paper we presented a 9/4-approximation algorithm for the Cluster- VD problem, based on the local ratio technique. The main idea underlying the algorithm is that in a twin-free, (C_4, K_5)-free graph, one can define weights on the vertices of the second neighborhood of any maximum degree vertex in order to guarantee a local ratio of at most 2. Moreover, the input graph can be made twin-free and C_4-free without worsening the approximation ratio beyond 2. Making the graph K_5-free is what causes the approximation ratio to increase to 9/4. If the input graph is K_5-free, our algorithm is in fact a 2-approximation algorithm.

Furthermore, looking closely at the proof of Lemma 6, we see that one can also obtain a 2-approximation algorithm for diamond-free graphs. This is due to the fact that, if G is diamond-free, the open neighborhood of any vertex is a union of cliques.

Theorem 8 *There is a 2-approximation algorithm for* Cluster- VD *in the class of K_5-free graphs, and in the class of diamond-free graphs.*

We note that Theorem 8 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 et al. [12] in their 5/2-approximation algorithm for (unweighted) Cluster- VD.

Acknowledgements We thank the anonymous referees for their careful reading of the paper and helpful remarks.

References

1. Bar-Yehuda, R., Bendel, K., Freund, A., Rawitz, D.: Local ratio: a unified framework for approximation algorithms. ACM Comput. Surv. **36**(4), 422–463 (2004)
2. Boral, A., Cygan, M., Kociumaka, T., Pilipczuk, M.: A fast branching algorithm for cluster vertex deletion, computer Science—theory and applications. Lecture Notes in Computer Science, vol. 8476, pp. 111–124. Springer (2014). arXiv:1306.3877
3. Cai, M., Deng, X., Zang, W.: An approximation algorithm for feedback vertex sets in tournaments. SIAM J. Comput. **30**(6), 1993–2007 (2001)
4. Chudak, F.A., Goemans, M.X., Hochbaum, D.S., Williamson, D.P.: A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. Oper. Res. Lett. **22**(4), 111–118 (1998)
5. Fiorini, S., Joret, G., Schaudt, O.: Integer programming and combinatorial optimization. Lecture Notes in Computer Science, vol. 9682, pp. 238–249. Springer (2016)
6. Guruswami, V., Lee, E.: In Approximability of Feedback Vertex Set for Bounded Length Cycles, ECCC:TR14-006
7. Guruswami, V., Lee, E.: Inapproximability of H-transversal/packing. SIAM J. Discrete Math. **31**(3), 1552–1571 (2017). arXiv:1506.06302
8. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. Theory Comput. Syst. **47**(1), 196–217 (2010)
9. Iwata, Y., Oka, K.: Fast dynamic graph algorithms for parameterized problems, Algorithm theory—SWAT 2014. Lecture Notes in Computer Science, vol. 8503, pp. 241–252. Springer (2014)
10. Mnich, M., Williams, V.V., Végh, L.A.: A 7/3-approximation for feedback vertex sets in tournaments. In: 24th Annual European Symposium on Algorithms, LIPIcs. Leibniz International Proceedings in Informatics, vol. 57, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern (2016). http://drops.dagstuhl.de/opus/volltexte/2016/6409, pp. Art. No. 67, 14
11. Tu, J., Zhou, W.: A primal-dual approximation algorithm for the vertex cover P_3 problem. Theor. Comput. Sci. **412**(50), 7044–7048 (2011)
12. You, J., Wang, J., Cao, Y.: Approximate association via dissociation. Discrete Appl. Math. **219**, 202–209 (2017). arXiv:1510.08276