



# Parameterized Algorithms for Cluster Vertex Deletion on Degree-4 Graphs and General Graphs

Kangyi Tian<sup>1</sup>(✉), Mingyu Xiao<sup>1</sup>(ID), and Boting Yang<sup>2</sup>

<sup>1</sup> School of Computer Science and Engineering,  
University of Electronic Science and Technology of China, Chengdu, China  
tkysss@outlook.com

<sup>2</sup> University of Regina, Regina, Canada

**Abstract.** In the CLUSTER VERTEX DELETION problem, we are given a graph  $G$  and an integer  $k$ , and the goal is to determine whether we can delete at most  $k$  vertices from  $G$  to make the remaining graph a cluster, i.e., a graph with each connected component being a complete graph. In this paper, we show that CLUSTER VERTEX DELETION can be solved in  $O^*(1.7549^k)$  time, improving the previous result of  $O^*(1.811^k)$ . To obtain this result, one crucial step is to show CLUSTER VERTEX DELETION on graphs of maximum degree at most 4 can be solved in  $O^*(1.7485^k)$  time. After this, we know that the graph will always have a vertex of degree at least 5. Then by adopting the previous method of automated generation of searching trees, we can get the result on general graphs.

## 1 Introduction

In clustering problems, there is a set of elements and a value of similarity between each pair of elements in it. The goal is to partition these elements into distinct groups based on their similarities. This problem has wide-ranging applications in various fields, such as machine learning [1] and computational biology [2]. One effective approach to model this problem is by constructing a similarity graph, where the elements form the vertices and edges exist only if the similarity between two vertices exceeds a specified threshold. Therefore, the clustering problem transforms into partitioning the similarity graph into parts where each part induces a complete graph. Occasionally, the given similarity data may deviate slightly from the actual clustering model due to noise or other factors, resulting in improper no-instance. For such cases, modifications are permitted within the clustering problem, leading to cluster graph modification problems. Depending on the type of modifications allowed, different problems can be defined. If we can only delete edges to create a cluster, i.e., a graph where each connected component forms a complete graph, the problem is referred to as Cluster Edge Deletion [7, 8, 10, 11, 15]. If both deleting and adding edges are allowed to form a cluster, the problem is known as Cluster Editing [3, 4, 7, 10, 11]. When solely the deletion of vertices is allowed, the problem becomes Cluster Vertex Deletion (CVD) [5, 6, 10, 12, 14, 16]. These problems have been extensively examined

in the literature. In this paper, we focus on studying parameterized and exact algorithms for CVD.

**Table 1.** Parameterized algorithms for CLUSTER VERTEX DELETION

References	Results	Main Methods
Cai, 1996 [6]	$O^*(3^k)$	forbidden graphs
Gramm et al., 2004 [10]	$O^*(2.26^k)$	automated generation of searching trees
Wahlström, 2007 [16]	$O^*(2.076^k)$	3-hitting
Hüffer et al., 2010 [12]	$O^*(2^k)$	iterative compression
Boral et al., 2016 [5]	$O^*(1.911^k)$	auxiliary graph $H^v$
Tsur, 2021 [14]	$O^*(1.811^k)$	careful analysis by considering twins
<b>This paper</b>	$O^*(1.7549^k)$	dealing with low-degree graphs

In the CLUSTER VERTEX DELETION (CVD) problem, we are given a graph  $G = (V, E)$  and an integer  $k$ . The goal is to determine whether there is a set of vertices  $S$  of size at most  $k$  such that after deleting  $S$  from  $G$  the remaining graph is a cluster. One important property is that a graph is a cluster if and only if there is no induced 3-path (i.e., a path with three vertices such that there is no edge between the first and the last vertices) in the graph. In other words, induced 3-path is the forbidden graph of a cluster. By using this property and applying the general branching algorithm of Cai [6], we can get an  $O^*(3^k)$ -time branching algorithm for this problem. Later, Gramm et al. [10] improved this result to  $O^*(2.26^k)$  by using automated generation techniques for search trees. Since hitting all induced 3-paths is a special case of the 3-hitting problem, the  $O^*(2.076^k)$ -time algorithm for the 3-hitting problem [16] directly implies the same complexity for CVD. In 2010, Hüffer et al. [12] improved the runtime to  $O^*(2^k)$  using the iterative compression method. However, the iterative compression method naturally provides a lower bound of  $2^k$  in complexity, limiting further improvement. The next breakthrough came in 2016 when Boral et al. [5] proposed a novel idea of selecting a vertex  $v$  and considering vertex sets that hit all induced 3-paths containing  $v$ . They constructed an auxiliary graph and used a Python script, similar to the one in [10], to automate the generation of search trees, resulting in an improved  $O^*(1.911^k)$ -time algorithm. Tsur [14] further refined the algorithm, achieving the current best runtime of  $O^*(1.811^k)$ .

In this paper, we further improve the running time bound to  $O^*(1.7549^k)$  for CVD. To get this result, we first show that the problem can be solved in  $O^*(1.7485^k)$  when the input graph has maximum degree at most 4. After that, we adopt the technique of the auxiliary graph  $H^v$  and a method to automatically generate branching vectors used in [5, 14]. Since the degree of the vertex considered is at least 5, we have more vertices in the local structure and the previous bottlenecks can be avoided. The history of parameterized algorithms for CVD was listed in Table 1. We also remark that by using the general approach

to obtain exact algorithms via fast parameterized algorithms in [9], we can also solve CVD in  $O(1.4302^n)$  time, improving all previous results.

One of the most important contributions in this paper is to obtain the improved result for the problem in degree-4 graphs. For vertices of degree  $\geq 5$ , we just modify previous algorithms to get our desired result. Due to the space limitation, we mainly introduce our algorithm for degree-4 graphs in the main body and the part for high-degree vertices and general graphs can be found in the full version of this paper.

## 2 Notations

Let  $G = (V, E)$  denote an undirected graph. For a vertex  $v \in V$ , we use  $N_G(v)$  to represent the set of neighbors of  $v$  in graph  $G$ , i.e., vertices adjacent to  $v$  in  $G$ . Define  $\deg_G(v) = |N_G(v)|$  as the degree of the vertex  $v$ . For a subset  $V'$  of vertices, let  $N_G(V') = (\cup_{v \in V'} N_G(v)) \setminus V'$  and  $N_G[V'] = N_G(V') \cup V'$ . We also use  $N_G^2(v) = N_G(N_G(v)) \setminus \{v\}$  to represent the vertices with distance exactly 2 to vertex  $v$ . In the above notations, we may omit the subscript  $G$  if the graph  $G$  is clear from the context. A singleton  $\{v\}$  may be simply written as  $v$ . For a graph  $G'$ , we use  $V(G')$  and  $E(G')$  to denote the set of vertices and the set of edges in  $G'$ , respectively. For  $V' \subseteq V$  and  $E' \subseteq E$ , a graph  $G' = (V', E')$  is called an induced subgraph of  $G$  if any edge  $(x, y)$  in  $E$  is in  $E'$  iff  $x \in V'$  and  $y \in V'$ . Let  $G[V']$  denote the induced subgraph of  $G$  with vertices  $V'$  and  $G - V'$  denote  $G[V \setminus V']$ .

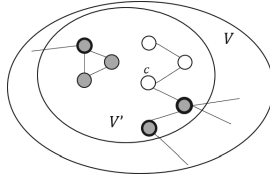
A vertex set  $S$  of a graph  $G$  is called a *vertex cover* if for any edge in  $G$  there is at least one endpoint in  $S$ . A *3-path* is a path of 3 vertices. A 3-path  $(u, v, w)$  is called an *induced 3-path* if there is no edge between  $u$  and  $w$ . A complete graph is also called a *clique*. A graph is called a *cluster* if each connected component is a clique. The CLUSTER VERTEX DELETION problem (CVD) is to check whether we can delete at most  $k$  vertices from a graph to make the remaining graph a cluster. An *instance* of CVD is denoted by  $(G, k)$ . A subset  $S$  of vertices is called a *CVD-set* if  $G - S$  is a cluster. A CVD-set is also called a *solution set* to CVD and a minimum CVD-set is called an *optimal solution set* to CVD. The following forbidden-graph property has been frequently used in algorithms for CVD [5, 6, 10, 12, 14, 16].

**Lemma 1.** *A graph is a cluster iff the graph contains no induced 3-path.*

## 3 Properties for Algorithm of Degree-4 Graph

### 3.1 The CVD-Dominating Family

**Definition 1.** *Let  $V'$  be a subset of vertices of the graph  $G$ . Define  $\Delta_G(V') = N_G(V \setminus V') \cap V'$ , i.e., the set of vertices in  $V'$  adjacent to some vertices not in  $V'$ . Let  $T$  be a CVD-set of  $G[V']$ . Define  $\Delta_G(V', T)$  as the set of vertices of cliques in  $G[V' - T]$  containing some vertices in  $\Delta_G(V')$ . See Fig. 1 for an illustration of these concepts.*



**Fig. 1.** An example for Definition 1.

In Fig. 1, vertices in  $\Delta_G(V')$  are bold. A CVD-set  $T$  of  $G[V']$  is  $\{c\}$ . After removing  $T$  from  $V'$ , there are 3 cliques and the vertices in  $\Delta_G(V', T)$  are all marked as gray vertices.

**Definition 2.** Let  $V'$  be a subset of vertices of the graph  $G$ . For two CVD-sets  $X$  and  $Y$  of induced graph  $G[V']$ , we say that  $Y$  dominates  $X$  under  $G[V']$  if

$$|\Delta_G(V', Y) \setminus \Delta_G(V', X)| \leq |X| - |Y|.$$

**Lemma 2.** Let  $V' \subseteq V$  be a vertex subset of  $G$ , and  $X$  and  $Y$  be two CVD-sets of  $G[V']$  such that  $Y$  dominates  $X$  under  $G[V']$ . If there is a CVD-set  $S$  of  $G$  containing  $X$  then there is a CVD-set  $S'$  of  $G$  containing  $Y$  such that  $|S'| \leq |S|$ .

*Proof.* Let  $S$  be a CVD-set of  $G$  containing  $X$ . Let  $Y' = Y \cup (\Delta_G(V', Y) \setminus \Delta_G(V', X))$ . We show that  $S' = (S \setminus X) \cup Y'$  is a CVD-set of  $G$  containing  $Y$  such that  $|S'| \leq |S|$ .

Since  $Y \subseteq Y' \subseteq S'$ , we know that  $S'$  contains  $Y$ . Next, we prove  $G - S'$  is a cluster. Assume to the contrary that  $G - S'$  is not a cluster. By Lemma 1, we know there is an induced 3-path  $P$  in graph  $G - S'$ .

Since  $Y$  is a CVD-set of  $G[V']$  and  $Y \subseteq S'$ , we know that  $G[V' \setminus S']$  is a cluster. Then the induced 3-path  $P$  must contain at least one vertex in  $V \setminus V'$ . Thus, all the three vertices of  $P$  must be in  $(V \setminus (V' \cup S')) \cup \Delta_G(V', Y')$ , which is a superset of  $(V \setminus (V' \cup S')) \cup \Delta_G(V', X)$ , by the definition of  $Y'$ . This implies that the induced 3-path  $P$  also exists in  $G - S$ , a contradiction. So  $G - S'$  is a cluster. Last, we consider the size of  $S'$ . We can see that  $|S'| = |(S \setminus X) \cup Y'| = |(S \setminus X) \cup Y \cup (\Delta_G(V', Y) \setminus \Delta_G(V', X))| \leq |S| - |X| + |Y| + |\Delta_G(V', Y) \setminus \Delta_G(V', X)| \leq |S|$ . So  $S'$  is a satisfied set.  $\square$

**Definition 3.** A CVD-dominating family  $\mathcal{F}_G(V')$  is a set of CVD-sets of  $G[V']$  such that for every CVD-set  $X$  of  $G[V']$ , there is a CVD-set in  $\mathcal{F}_G(V')$  dominating  $X$  under  $G[V']$ .

Based on Definition 3, to search for a solution, instead of listing all CVD-sets of  $V'$ , we can only consider a CVD-dominating family.

**Definition 4.** A vertex subset  $V' \subseteq V$  is called an important set if there is a minimum CVD-set  $C$  of induced subgraph  $G[V']$  such that  $\Delta_G(V') \subseteq C$ . The set  $C$  is called an important cut.

**Corollary 1 (of Lemma 2).** Let  $V' \subseteq V$  be an important set with an important cut  $C$ . There is a minimum CVD-set of  $G$  containing  $C$ .

### 3.2 Core Branching Processing

We introduce a technique to design branching rules. The idea is to relax the concept of important sets. For an instance  $(G, k)$ , we consider a set of vertices  $C$  that induces a clique. Let  $S$  be a CVD-set of  $(G, k)$ . Since  $G - S$  is a cluster, we know that  $C \cap (V - S)$  is either empty or forms a clique. When designing a branching rule, we can only consider all possible cliques in  $G - S$  intersecting with  $C$ . Furthermore, we can also reduce some dominated cases. This processing to generate branching rules is called core branching processing and the selected set  $C$  is called the *core clique*.

**Definition 5.** *Given an reduced instance  $(G, k)$ , the core branching processing will produce a branching rule by the following steps:*

1. *Choose an induced clique of  $G$  as the core clique  $C$ ;*
2. *Let  $\mathcal{T}$  be the set of cliques intersected with  $C$  and  $\mathcal{C} = \{C\} \cup \{N(T) \mid T \in \mathcal{T}\}$ ;*
3. *If there exist  $T_1$  and  $T_2 \in \mathcal{T}$  such that  $|N(T_1)| \geq |N(T_2)|$  and  $N[T_1] \subseteq N[T_2]$ , remove  $N(T_1)$  from  $\mathcal{C}$ ;*
4. *If there exist  $T_1 \in \mathcal{T}$  such that  $|N(T_1)| \geq |C|$  and  $N[T_1] \subseteq C$  (resp.,  $|C| \geq |N(T_1)|$  and  $C \subseteq N[T_1]$ ), remove  $N(T_1)$  (resp.,  $C$ ) from  $\mathcal{C}$ .*

The correctness of Step 3 of the core branching processing is based on the fact that if there is a CVD-set of  $G$  containing  $N(T_1)$ , then there must be a CVD-set containing  $N(T_2)$ . Here we use the concept of dominating. Let  $V' = N[T_2]$ . We know that  $\Delta_G(V') \subseteq N(T_2)$ . Assume that  $S$  is a CVD-set of  $G$  containing  $N(T_1)$ . Let  $X = S \cap V'$ . Then  $S$  is also a CVD-set of  $V'$ . Since  $N[T_1] \subseteq N[T_2]$ , we know that  $|X| \geq |N(T_1)| \geq |N(T_2)|$ . By Definition 2 and Lemma 2, we know that  $N(T_2)$  dominates  $X$  and there must be a CVD-set containing  $N(T_2)$ .

## 4 Reduction Rules

We first introduce some reduction rules that will be applied in our algorithms to reduce the instance in polynomial time.

**R-Rule 1.** *If  $k < 0$ , return ‘no’ to indicate that the instance is a no-instance.*

**R-Rule 2.** *If the graph is an empty graph, return ‘yes’ to indicate that the instance is a yes-instance.*

**R-Rule 3.** *If there is a connected component of  $G$  that is a clique, delete it from the graph.*

**R-Rule 4.** *If there is a connected component  $C$  of  $G$  and a vertex  $u \in C$  such that  $G[C] - u$  is a cluster, delete  $u$  from the graph and decrease  $k$  by 1.*

**R-Rule 5.** *If there is an important set  $V' \subseteq V$  with an important cut  $C$  such that  $|C| \leq 10$ , then return the instance  $(G - C, k - |C|)$ .*

In R-Rule 5, we require that  $|C| \leq 10$ . So the important sets  $V'$  and important cuts  $C$  can be found in polynomial time if they exist. We mention a special case of the important sets that will be reduced by this rule. Let  $u$  be a degree-2 vertex with one degree-1 neighbor  $v$  and another neighbor  $w$ . Then  $\{u, v, w\}$  is an important set satisfying the condition in R-Rule 5. We may delete  $w$  directly from the graph.

## 5 The Algorithm for Degree-4 Graphs

First of all, we present our algorithm for the case that the input graph is a graph with maximum degree at most 4. In this section, we mainly use the concept of the CVD-dominating family and the method of core branching processing. The algorithm contains the following seven main steps. When introduce one step, we will assume all previous steps can not be applied.

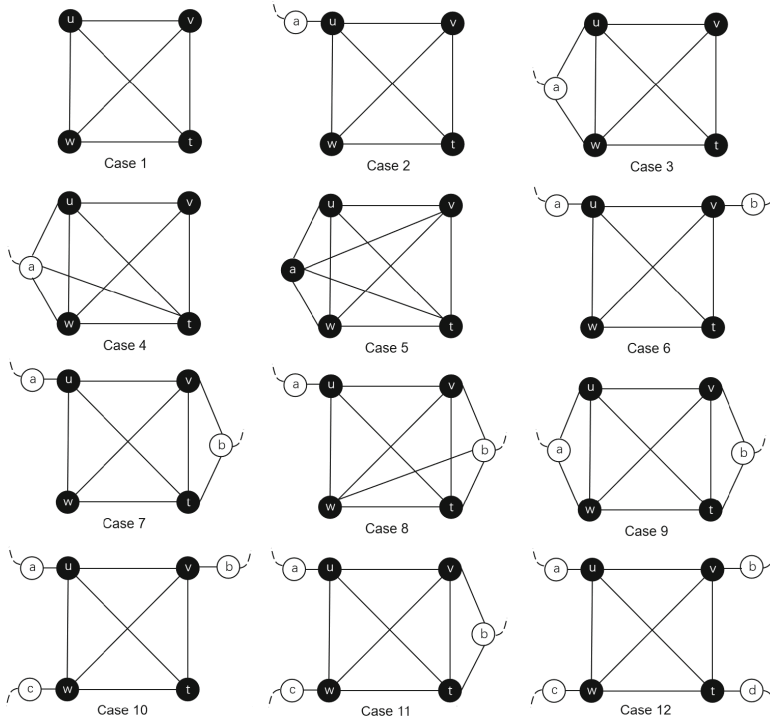
1. Iteratively apply the above five reduction rules to get a reduced instance.
2. If there is a degree-1 vertex  $v$ , execute the branching operations in Sect. 5.1.
3. If there is a 4-clique, execute the branching operations in Sect. 5.2.
4. If there is a degree-4 vertex  $v$ , execute the branching operations in Sect. 5.3.
5. If there are two triangles sharing one edge, execute the branching operations in Sect. 5.4.
6. If there is a triangle, execute the branching operations in Sect. 5.5.
7. Reduce the problem to the 3-path vertex cover problem.

### 5.1 Degree-1 Vertices

Assume there is a degree-1 vertex  $u$ . We let  $v$  denote the unique neighbor of  $u$ . We know that  $v$  is a vertex of degree at least three otherwise if  $v$  is of degree 2 then R-Rule 5 would be applied on the important set  $N[v]$  and if  $v$  is of degree 1 then R-Rule 3 would be applied. We pick  $\{u\}$  as the core clique and apply the core branching processing. The set of intersecting cliques is  $\mathcal{T} = \{\emptyset, \{u\}, \{u, v\}\}$  and thus we will get three initial branches: (1) deleting  $\{u\}$  and decreasing  $k$  by 1; (2) deleting  $\{v\}$  and decreasing  $k$  by 1; (3) deleting  $N(u, v)$  and decreasing  $k$  by  $|N(u, v)| \geq 2$ . However, the first branch is dominated by the second branch and then the first branch will be removed according to the last step of the core branching processing. Thus, we get a branching vector at least  $(1, 2)$ .

### 5.2 4-Cliques

Assume there is a 4-clique  $\{v, u, w, t\}$ . We show all possible configurations of the clique in Fig. 2, where dark vertices mean all neighbors of them are already drawn on the figure and a dotted edge incident on a vertex means there may be some further edges incident on this vertex. Note that a vertex has degree at most 4. Each vertex in  $\{v, u, w, t\}$  can be adjacent to at most one vertex not in this set. Thus  $|N(v, u, w, t)| \leq 4$ . Let  $N_c = |N(v, u, w, t)|$ . The cases in Fig. 2 are listed



**Fig. 2.** The cases with 4-cliques

according to the value of  $N_c$ . In Case 1,  $N_c = 0$ ; In Cases 2–5,  $N_c = 1$ ; In Cases 6–9,  $N_c = 2$ ; In Cases 10–11,  $N_c = 3$ ; In Case 12,  $N_c = 4$ . For Cases 1 and 5, the graph is a component of clique and it would be reduced by R-Rule 3. For Cases 2–4,  $\{v, u, w, t, a\}$  is an important set with the important cut  $\{a\}$  and it would be reduced by R-Rule 5. For Cases 6, 7 and 9,  $\{v, u, w, t, a, b\}$  is an important set with the important cut  $\{a, b\}$  and it would be reduced by R-Rule 5. For Case 8, we consider the subgraph  $G'$  induced by  $\{v, u, w, t, a, b\}$ . All CVD-sets in  $G'$  of size at least 2 are dominated by  $\{a, b\}$  and there is only one CVD-set  $\{u\}$  of size less than 2 in  $G'$ . Thus, we branch into two branches: deleting  $\{a, b\}$  and decreasing  $k$  by 2; deleting  $\{u\}$  and decreasing  $k$  by 1. We get a branching vector  $(2, 1)$ . For Case 10,  $\{v, u, w, t, a, b, c\}$  is an important set with the important cut  $\{a, b, c\}$  and it would be reduced by R-Rule 5. For Case 11, we consider the subgraph  $G'$  induced by  $\{v, u, w, t, a, b, c\}$ . All CVD-sets in  $G'$  of size at least 3 are dominated by  $\{a, b, c\}$  and there is only one CVD-set  $\{u, w\}$  of size less than 3 in  $G'$ . Thus, we branch into two branches: deleting  $\{a, b, c\}$  and decreasing  $k$  by 3; deleting  $\{u, w\}$  and decreasing  $k$  by 2. We get a branching vector  $(3, 2)$ . For Case 12, we first consider the subgraph  $G'$  induced by  $\{v, u, w, t, a, b, c, d\}$ . All CVD-sets in  $G'$  of size at least 4 are dominated by  $\{a, b, c, d\}$ . Next, we assume that  $S$  is a CVD-set such that  $S' = S \cap \{v, u, w, t, a, b, c, d\}$  contains at most 3 vertices. Then,

there are only four possibilities:  $S' = \{u, v, w\}, \{u, v, t\}, \{v, t, w\}$  or  $\{t, w, u\}$ . We consider  $S' = \{u, v, w\}$ . For this case, both  $t$  and  $d$  are left in a component of a clique of size 2 and thus  $N(t, d)$  is in  $S$ . After Sect. 4.1, there is no degree-1 vertex, and then  $d$  is adjacent to a vertex  $d' \neq t$ . Furthermore,  $d' \notin \{u, v, w\}$  since the graph is a degree-4 graph. For this case,  $\{u, v, w, d'\} \subseteq N(t, d)$  and at least  $|N(t, d)| \geq 4$  vertices can be deleted. We can decrease  $k$  by at least 4. By the same reason, for the cases  $S' = \{u, v, t\}, \{v, t, w\}$  and  $\{t, w, u\}$ , we can also decrease  $k$  by at least 4. Thus, we can branch into five branches and in each branch the parameter  $k$  decreases by at least 4. We get a branching vector  $(4, 4, 4, 4, 4)$ .

Among all the cases, the worst branching vector is  $(2, 1)$ , the branching factor of which is 1.6182.

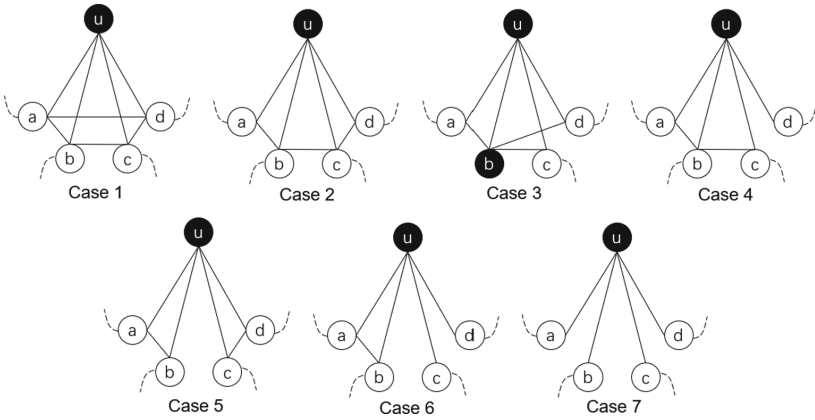


Fig. 3. The cases with 4-degree vertices

### 5.3 Degree-4 Vertices

Assume there is a degree-4 vertex  $u$  with four neighbors  $\{a, b, c, d\}$ . Let  $E_l$  denote the set of edges with both endpoints in  $\{a, b, c, d\}$  and  $l$  be the size of  $E_l$ . Note that the edges in  $E_l$  will not form a triangle, since there is no cliques of size 4 after Sect. 4.2. We consider different cases according to the value of  $l$ . All the cases are shown in Fig. 3.

**Case 1.**  $l = 4$ . Since there is no triangle, the four edges in  $E_l$  will form a cycle of size 4. Without loss of generality, we assume that  $E_l = \{(a, b), (b, c), (c, d), (d, a)\}$ . Let  $X$  be the set with the least neighbors among  $\{u, a, b\}, \{u, b, c\}, \{u, c, d\}$  and  $\{u, d, a\}$ . If  $|N(X)| \leq 3$ , then  $N[X]$  is an important set with the important cut being  $N(X)$ , because any CVD-set of the graph induced by  $N[X]$  will contain at least  $|N(X)|$  vertices. For this case, it will be reduced by R-Rule 5. Next, we assume that  $|N(X)| \geq 4$ . Let  $a', b', c'$  and  $d'$  be the four neighbor of  $a, b, c$  and  $d$ , respectively, where  $a' \neq b', b' \neq c', c' \neq d'$  and  $d' \neq a'$ .



We choose  $\{u\}$  as the core clique and apply the core branching processing. The intersecting cliques are  $\emptyset$ ,  $\{u\}$ ,  $\{u, a\}$ ,  $\{u, b\}$ ,  $\{u, c\}$ ,  $\{u, d\}$ ,  $\{u, a, b\}$ ,  $\{u, b, c\}$ ,  $\{u, c, d\}$ , and  $\{u, d, a\}$ . We get initial branches of deleting  $\{u\}$ ,  $N(u)$ ,  $N(u, a)$ ,  $N(u, b)$ ,  $N(u, c)$ ,  $N(u, d)$ ,  $N(u, a, b)$ ,  $N(u, b, c)$ ,  $N(u, c, d)$ , and  $N(u, d, a)$ . The branches of deleting  $N(u)$ ,  $N(u, a)$  and  $N(u, b)$  are dominated by the branch of deleting  $N(u, a, b)$ . The branches of deleting  $N(u, c)$  and  $N(u, d)$  are dominated by the branch of deleting  $N(u, c, d)$ . We can branch by deleting  $\{u\}$ ,  $N(u, a, b)$ ,  $N(u, b, c)$ ,  $N(u, c, d)$  or  $N(u, d, a)$ , where  $|N(u, a, b)|$ ,  $|N(u, b, c)|$ ,  $|N(u, c, d)|$  and  $|N(u, d, a)| = 4$  since  $|N(X)| \geq 4$  and the maximum degree of the graph is 4. The branching vector is  $(1, 4, 4, 4, 4)$  with a branching factor 1.7485.

**Case 2.**  $l = 3$ . There are three edges in  $E_l$  and they form a path of length 3. Without loss of generality, we assume that  $E_l = \{(a, b), (b, c), (c, d)\}$ . If  $\deg(b) = \deg(c) = 3$ , then  $\{u, a, b, c, d\}$  is an important set with an important cut  $\{a, d\}$ , which would be reduced by R-Rule 5. Next, we assume that  $|N(u, b, c)| \geq 3$ .

**Case 2.1.** Only one of  $\{b, c\}$  is of degree 4. We assume without loss of generality that  $\deg(b) = 4$ ,  $\deg(c) = 3$  and  $e$  is the fourth neighbor of  $b$ . We further distinguish two cases. If  $a$  is not adjacent to  $e$  or both of  $a$  and  $d$  are adjacent to  $e$ , then  $\{u, a, b, c, d, e\}$  is an important set with an important cut  $\{a, d, e\}$ , which would be reduced by R-Rule 5. Next, we assume that  $a$  and  $e$  are adjacent but  $d$  and  $e$  are not adjacent. If  $d$  is of degree 2, then  $\{u, a, b, c, d\}$  is an important set with an important cut  $\{a, b\}$ , which would be reduced by R-Rule 5. Thus, we assume that  $d$  is adjacent to a vertex  $f$  different from  $\{e, u, c\}$ .

Let  $V' = \{u, a, b, c, d, e, f\}$  and  $G' = G[V']$ . In  $G'$ , any CVD-set of size at least three is dominated by  $\{a, d, e\}$  or  $\{a, b, f\}$ . There is only one CVD-set of size less than three  $\{u, c\}$ . We can branch by either deleting  $\{a, d, e\}$  or  $\{a, b, f\}$  or  $\{u, c\}$ , and we get a branching vector  $(3, 3, 2)$ .

**Case 2.2.** Both of  $\{b, c\}$  are degree-4 vertices. Let  $e$  be the fourth neighbor of  $b$  and  $f$  be the fourth neighbor of  $d$ . We know that  $e, f \notin \{u, a, b, c, d\}$ . First we show that  $e \neq f$ . If  $e = f$ , then  $e$  is not adjacent to either  $a$  or  $d$ . Otherwise, this case will be solved in Case 1. Thus, there is an important set  $V' = \{u, a, b, c, d, e\}$  with an important cut  $\{a, e, d\}$ , which should be reduced by R-Rule 5. Next, we assume that  $e \neq f$ .

If  $a$  is not adjacent to a vertex different from  $u, b$  and  $e$ , then there is an important set  $\{u, a, b, c, d, e\}$  with an important cut  $\{c, d, e\}$ . If  $d$  is not adjacent to a vertex different from  $u, c$  and  $f$ , then there is an important set  $\{u, a, b, c, d, f\}$  with an important cut  $\{a, b, f\}$ . Thus, we can assume that  $N(u, a, b), N(u, c, d) \geq 4$ .

We choose  $\{u\}$  as the core clique and apply the core branching processing. The intersecting cliques are  $\emptyset$ ,  $\{u\}$ ,  $\{u, a\}$ ,  $\{u, b\}$ ,  $\{u, c\}$ ,  $\{u, d\}$ ,  $\{u, a, b\}$ ,  $\{u, b, c\}$ ,  $\{u, c, d\}$ , and  $\{u, d, a\}$ . The branches of deleting  $N(u)$ ,  $N(u, b)$  and  $N(u, c)$  are dominated by the branch of deleting  $N(u, b, c)$ . The branch of deleting  $N(u, a)$  is dominated by the branch of deleting  $N(u, a, b)$ . The branch of deleting  $N(u, d)$  is dominated by the branch of deleting  $N(u, c, d)$ . We can branch by deleting  $\{u\}$ ,

$N(u, a, b)$ ,  $N(u, b, c)$  or  $N(u, c, d)$ , where  $|N(u, a, b)|, |N(u, b, c)|, |N(u, c, d)| \geq 4$ . The branching vector is  $(1, 4, 4, 4)$  with a branching factor 1.6851.

**Case 3.**  $l = 3$  and the three edges in  $E_l$  have a common endpoint. Without loss of generality, we assume that  $E_l = \{(a, b), (b, c), (b, d)\}$ . Let  $X = \{u, a, b\}, \{u, b, c\}$ , or  $\{u, b, d\}$ . If  $|N(X)| \leq 2$ , then  $N[X]$  is an important set with the important cut being  $N(X)$ . It will be reduced by R-Rule 5. Next, we assume that  $|N(X)| \geq 3$ . Let  $a', c'$  and  $d'$  be the three neighbors of  $a, c$  and  $d$ , respectively.

We choose  $\{u\}$  as the core clique and apply the core branching processing. The intersecting cliques are  $\emptyset, \{u\}, \{u, a\}, \{u, b\}, \{u, c\}, \{u, d\}, \{u, a, b\}, \{u, b, c\}$ , and  $\{u, b, d\}$ . We get initial branches of deleting  $\{u\}, N(u), N(u, a), N(u, b), N(u, c), N(u, d), N(u, a, b), N(u, b, c)$  and  $N(u, b, d)$ . The branches of deleting  $N(u)$  and  $N(u, a)$  are dominated by the branch of deleting  $N(u, a, b)$ . The branches of deleting  $N(u, c)$  and  $N(u, d)$  are dominated by the branches of deleting  $N(u, b, c)$  and  $N(u, b, d)$ , respectively. The branch of deleting  $N(u, b)$  will be dominated by the branching deleting  $X$  if  $|N(X)| \leq 3$ . Notice that  $u$  and  $b$  are twins of each other and by Lemma 9 in [15], we can delete  $b$  at the same time if we delete  $u$ . We can branch by deleting  $\{u, b\}, N(u, b), N(u, a, b), N(u, b, c)$  or  $N(u, b, d)$ , where  $|N(u, a, b)|, |N(u, b, c)|$  and  $|N(u, b, d)| = 4$  or deleting  $\{u, b\}, N(u, a, b), N(u, b, c)$  or  $N(u, b, d)$ , where  $|N(u, a, b)|, |N(u, b, c)|$  or  $|N(u, b, d)| = 3$ . The branching vector is  $(2, 3, 4, 4, 4)$  or  $(2, 3, 3, 3)$  with a branching factor 1.6472 or 1.6717.

**Case 4.**  $l = 2$  and the two edges in  $E_l$  have a common endpoint. Without loss of generality, we assume that  $E_l = \{(a, b), (b, c)\}$ . After Sect. 5.1, there is no degree-1 vertex. We assume that  $d$  is adjacent to a vertex  $e \neq u$ .

**Case 4.1.**  $deg(b) = 3$ . For this case, neither  $a$  or  $c$  can be a degree-2 vertex, otherwise there would be an important set  $\{u, a, b, c, d\}$  with an important cut  $\{c, d\}$  or  $\{a, d\}$ . Thus, we have  $|N(a, b, u)| \geq 3$  and  $|N(c, d, u)| \geq 3$ . We choose  $\{b\}$  as the core clique and apply the core branching processing. The intersecting cliques are  $\emptyset, \{b\}, \{b, a\}, \{b, u\}, \{b, c\}, \{b, a, u\}$  and  $\{b, c, u\}$ . The branch of deleting  $N(b)$  and deleting  $N(b, a)$  is dominated by the branch of deleting  $N(b, a, u)$ . The branch of deleting  $N(b, c)$  is dominated by the branch of deleting  $N(b, c, u)$ . After ignoring the branching of deleting  $N(b), N(b, a)$  and  $N(b, c)$ , we may be able to get a branching vector  $(1, 3, 3, 3)$ . However, it is not good enough. We further analyze several cases. If  $deg(a) = 3$  (resp.,  $deg(c) = 3$ ), then the branching of deleting  $N(b, u)$  is dominated by the branch of deleting  $N(b, u, a)$  (resp.,  $N(b, u, c)$ ). For this case, we can branch with  $(1, 3, 3)$  at least, the branching factor of which is 1.6957. Otherwise,  $deg(a) \geq 4$  and  $deg(c) \geq 4$ . The branching vector is at least  $(1, 3, 4, 4)$  with a branching factor 1.7254.

**Case 4.2.**  $deg(b) = 4$ . Let  $f$  be the fourth neighbor of  $b$ . We choose  $\{u\}$  as the core clique and apply the core branching processing. The intersecting cliques are  $\emptyset, \{u\}, \{u, a\}, \{u, b\}, \{u, c\}, \{u, d\}, \{u, a, b\}$ , and  $\{u, b, c\}$ . The branch of deleting  $N(u)$  is dominated by the branch of deleting  $N(u, b)$ . The branch of deleting  $N(u, a)$  is dominated by the branch of deleting  $N(u, a, b)$ . The branch of deleting  $N(u, c)$  is dominated by the branch of deleting  $N(u, b, c)$ . We only

need to consider five branches of deleting  $\{u\}$ ,  $N(u, b)$ ,  $N(u, d)$ ,  $N(u, a, b)$ , and  $N(u, b, c)$ . We further consider several subcases. If  $\text{deg}(a) = \text{deg}(c) = 4$ , we have  $|N(a, u, b)| = |N(b, u, c)| = 5$  because  $a$  and  $b$  (resp.,  $b$  and  $c$ ) have no common neighbor other than  $u$  after Case 1. Then, we will get a branching vector  $(1, 4, 4, 5, 5)$  with a branching factor 1.6770. Next, we assume that either  $a$  or  $c$  is a vertex of degree at most 3. Then the branch of deleting  $N(u, b)$  is dominated by the branch of deleting  $N(u, a, b)$  or  $N(u, c, d)$ . There are four branches left: deleting  $\{u\}$ ,  $N(u, d)$ ,  $N(u, a, b)$ , and  $N(u, b, c)$ . If  $\text{deg}(a) = 3$  or  $\text{deg}(c) = 3$ , we can get a branching vector  $(1, 4, 4, 3)$  with a branching factor 1.7254. The only left case is that  $\text{deg}(a) = \text{deg}(c) = 2$ . For this case, the branch of deleting  $N(u, b, c)$  is dominated by the branch of deleting  $N(u, a, b)$ . we get a branching vector  $(1, 4, 3)$  with a branching factor 1.6181.

Please see the details of analysis for Cases 5,6 and 7 in the full version of this paper.

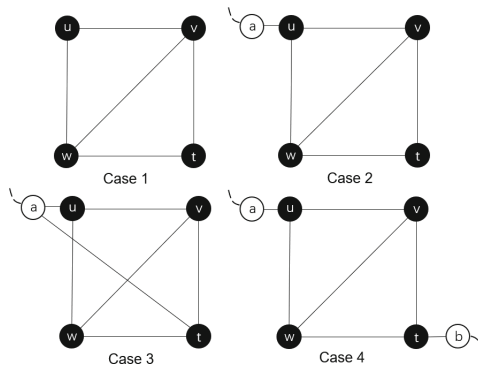


Fig. 4. two triangles sharing an edge

### 5.4 Two Triangles Sharing One Edge

In this step, there is no vertex of degree  $\geq 4$ . Assume there are two triangles  $\{u, w, v\}$  and  $\{t, w, v\}$  sharing one edge  $(w, v)$ . All possible configurations are shown in Fig. 4. Since the maximum degree is at most 3 now, we know that only  $u$  and  $t$  are possibly adjacent to a vertex out of  $\{u, v, w, t\}$ . Case 1 is a component of 4 vertices. We can simply put  $u$  to the solution. For Case 2, let  $V' = \{u, v, w, t, a\}$  and  $G' = G[V']$ . We can see that  $\{u\}$  is a CVD-set dominates all other CVD-sets in  $G'$ . Thus, we can also simply put  $u$  to the solution. For Case 3, let  $V' = \{u, v, w, t, a\}$  and  $G' = G[V']$ . We can see that  $\{a, u\}$  is a CVD-set dominates all other CVD-sets in  $G'$ . Thus, we can also simply put  $\{a, u\}$  to the solution. For Case 4, let  $V' = \{u, v, w, t, a, b\}$  and  $G' = G[V']$ . Any CVD-set in  $G'$  is dominated by either  $\{a, t\}$  or  $\{u, b\}$ . We can branch with a branching vector  $(2, 2)$ .

## 5.5 Triangles

Assume that there is still a triangle  $\{u, v, w\}$  in the graph. If  $|N(u, v, w)| \leq 2$ , then  $N[u, v, w]$  is an important set with an important cut  $N(u, v, w)$ , which should be reduced by R-Rule 5. Thus, it holds that  $|N(u, v, w)| = 3$ . Note that after Sect. 5.4, no two triangles share an edge and there are exact three edges between  $\{u, v, w\}$  and  $\{u', v', w'\}$ . Let  $\{u', v', w'\} = N(u, v, w)$ , where there are three edges  $(u, u')$ ,  $(v, v')$ ,  $(w, w')$  between  $\{u, v, w\}$  and  $\{u', v', w'\}$ . We choose  $\{u, v, w\}$  as the core clique and apply the core branching processing. The intersecting cliques will be  $\emptyset$ ,  $\{u\}$ ,  $\{v\}$ ,  $\{w\}$ ,  $\{u, v\}$ ,  $\{u, w\}$ ,  $\{v, w\}$ ,  $\{u, u'\}$ ,  $\{v, v'\}$ ,  $\{w, w'\}$  and  $\{u, v, w\}$ . Any branch will be dominated by one of the branches of deleting  $N(u, u')$ ,  $N(v, v')$ ,  $N(w, w')$  and  $\{u', v', w'\}$ . Since there is no degree-1 vertex, we can get a branching vector at least  $(3, 3, 3, 3)$  with the branching factor 1.5875.

## 5.6 The Remaining Case

Now the graph has the maximum degree at most 3 and there is no triangle in it. CVD is equal to delete at most  $k$  vertices from the graph to make the each connected component containing at most 2 vertices. The latter problem is the 3-path vertex cover problem. We directly use the  $O^*(1.713^k)$ -time algorithm for the 3-path vertex cover problem in [13] to solve our problem.

We have considered all the cases and the worst branching factor is 1.7485.

**Theorem 1.** CLUSTER VERTEX DELETION in graphs with maximum degree at most 4 can be solved in  $O^*(1.7485^k)$  time and polynomial space.

With this result, we can improve the running time bound for CLUSTER VERTEX DELETION in general graphs to  $O^*(1.7549^k)$ . The details can be found in the full version of this paper.

## 6 Conclusion

In this paper, we proposed an improved parameterized algorithm for the classic CLUSTER VERTEX DELETION. Except using previous techniques such as the auxiliary graph, twins, and automated generation of searching trees, we introduce two new concepts: CVD-dominating family and core branching processing. A significant contribution of our work is to tackle the problem in low-degree graphs as the first step. Subsequently, we focus solely on vertices with high degrees, which potentially contain more valuable information. This allows us to design specific branching rules to achieve better branching factors. It is worth noting that even for graphs with maximum degree 4, our algorithm still requires some new techniques and deep analyses to get the desired improvement.

**Acknowledgments.** The author is grateful to all the anonymous reviewers for fruitful and insightful comments to improve the presentation of the paper. The work is supported by the National Natural Science Foundation of China, under the grants 62372095 and 61972070.

## References

1. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. *Mach. Learn.* **56**(1–3), 89–113 (2004)
2. Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. *J. Comput. Biol.* **6**(3/4), 281–297 (1999)
3. Böcker, S.: A golden ratio parameterized algorithm for cluster editing. *J. Disc. Algor.* **16**, 79–89 (2012)
4. Böcker, S., Briesemeister, S., Bui, Q.B.A., Truß, A.: Going weighted: parameterized algorithms for cluster editing. *Theor. Comput. Sci.* **410**(52), 5467–5480 (2009)
5. Boral, A., Cygan, M., Kociumaka, T., Pilipczuk, M.: A fast branching algorithm for cluster vertex deletion. *Theory Comput. Syst.* **58**(2), 357–376 (2016)
6. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.* **58**(4), 171–176 (1996)
7. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-21275-3>
8. Damaschke, P.: Bounded-degree techniques accelerate some parameterized graph algorithms. In: Chen, J., Fomin, F.V. (eds.) *IWPEC 2009*. LNCS, vol. 5917, pp. 98–109. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-11269-0\\_8](https://doi.org/10.1007/978-3-642-11269-0_8)
9. Fomin, F.V., Gaspers, S., Lokshtanov, D., Saurabh, S.: Exact algorithms via monotone local search. In: Wicks, D., Mansour, Y. (eds.) *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, Cambridge, MA, USA, 18–21 June 2016, pp. 764–775. ACM (2016)
10. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica* **39**(4), 321–347 (2004)
11. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Graph-modeled data clustering: exact algorithms for clique generation. *Theory Comput. Syst.* **38**(4), 373–392 (2005)
12. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. *Theory Comput. Syst.* **47**(1), 196–217 (2010)
13. Tsur, D.: Parameterized algorithm for 3-path vertex cover. *Theor. Comput. Sci.* **783**, 1–8 (2019)
14. Tsur, D.: Faster parameterized algorithm for cluster vertex deletion. *Theory Comput. Syst.* **65**(2), 323–343 (2021)
15. Tsur, D.: Cluster deletion revisited. *Inf. Process. Lett.* **173**, 106171 (2022)
16. Wahlström, M.: Algorithms, measures and upper bounds for satisfiability and related problems. PhD thesis, Linköping University, Sweden (2007)