An Improved Algorithm for Parameterized Edge Dominating Set Problem

Ken Iwaide and Hiroshi Nagamochi

Department of Applied Mathematics and Physics,
Graduate School of Informatics,
Kyoto University, Japan,
{iwaide,nag}@amp.i.kyoto-u.ac.jp

Abstract. An edge dominating set of a graph $G = (V, E)$ is a subset $M \subseteq E$ of edges such that each edge in $E \setminus M$ is incident to at least one edge in M. In this paper, we consider the parameterized edge dominating set problem which asks us to test whether a given graph has an edge dominating set with size bounded from above by an integer k or not, and we design an $O^*(2.2351^k)$-time and polynomial-space algorithm. This is an improvement over the previous best time bound of $O^*(2.3147^k)$. We also show that a related problem: the parameterized weighted edge dominating set problem can be solved in $O^*(2.2351^k)$ time and polynomial space.

1 Introduction

An *edge dominating set* of a graph $G = (V, E)$ is a subset $M \subseteq E$ of edges in the graph such that each edge in $E \setminus M$ is incident with at least one edge in M. The *edge dominating set problem* (EDS) is to find a minimum edge dominating set of a given graph. The problem is one of the basic problems highlighted by Garey and Johnson [4] in their work on NP-completeness. Yanakakis and Gavril [13] showed that EDS is NP-hard even in planar or bipartite graphs of maximum degree 3. Randerath and Schiermeyer [6] designed an $O^*(1.4423^m)$-time and polynomial-space algorithm for EDS, where $m = |E|$ and O^* notation suppresses all polynomially bounded factors. The result was improved to $O^*(1.4423^n)$ by Raman *et al.* [5], where $n = |V|$. Considering the treewidth of the graph, Fomin *et al.* [3] obtained an $O^*(1.4082^n)$-time and exponential-space algorithm. With the measure and conquer method, Rooij and Bodlaender [7] designed an $O^*(1.3226^n)$-time and polynomial-space algorithm and an improved $O^*(1.3160^n)$-time and polynomial-space algorithm was presented by Xiao and Nagamochi [11]. For EDS in graphs of maximum degree 3, the best algorithm is an $O^*(1.2721^n)$-time and polynomial-space algorithm due to Xiao and Nagamochi [12].

The *parameterized edge dominating set problem* (PEDS) is, given a graph $G = (V, E)$ with an integer k, to decide whether there is an edge dominating set of size up to k. It is known that there is an FPT algorithm for PEDS; we can design an algorithm with the running time $f(k)poly(n)$ to solve the problem, where $f(k)$ is a function of k and $poly(n)$ is a polynomial of the number of vertices in G. For PEDS, an $O^*(2.6181^k)$-time and polynomial-space algorithm was given by Fernau [2]. Fomin *et al.* [3] obtained an $O^*(2.4181^k)$-time and exponential-space algorithm based on dynamic programming on treewidth-bounded graphs. With the measure and conquer method, Binkele-Raible and Fernau [1] designed an $O^*(2.3819^k)$-time and polynomial-space algorithm. Xiao *et al.* [9] give an $O^*(2.3147^k)$-time and polynomial-space branching algorithm. For PEDS in graphs of maximum degree 3, the best parameterized algorithm is an $O^*(2.1479^k)$-time and polynomial-space algorithm due to Xiao and Nagamochi [10].

Technical Report 2014-004, September 9, 2014, Department of Applied Mathematics and Physics, Kyoto University

EDS and PEDS are related to the *vertex cover problem*. A *vertex cover* of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex in the set. The set of endpoints of all edges in any edge dominating set is a vertex cover. To find an edge dominating set of a graph, we may enumerate vertex covers of the graph and construct edge dominating sets from the vertex covers. Many previous algorithms are based on enumeration of vertex covers. We enumerate candidates of such edge dominating sets by branching on a vertex: fixing it as a vertex incident on at least one edge in an edge dominating set with a bounded size or not. In the $O^*(2.3147^k)$-time algorithm to PEDS, Xiao *et al*. [9] observed that branching on vertices in a local structure called "2-path component" is the most inefficient among branchings on other local structures, and that reducing the number of branchings on 2-path components leads to an improvement over the time complexity. For this, they retained branching on 2-path components until no other structure remains, and effectively skipped subinstances that will not deliver edge dominating sets with a bounded size by systematically treating the set of 2-path components. In this paper, identifying new local structures, called "bi-claw," "leg-triangle" and "tri-claw components" and establishing a refined lower bound on the size of edge dominating sets, we design an $O^*(2.2351^k)$-time and polynomial-space algorithm.

Section 2 gives some terminologies and notations and introduces our branching operations of our algorithm. After Section 3 describes our algorithm that consists of three major stages, Section 4 analyzes the time complexity by deriving an upper bound on the number of all subinstances. Section 5 discusses a weighted variant of PEDS. Section 6 makes some concluding remarks. For space limitation, the proofs of lemmata are moved into Appendix A.

2 Preliminaries

2.1 Terminology and notation

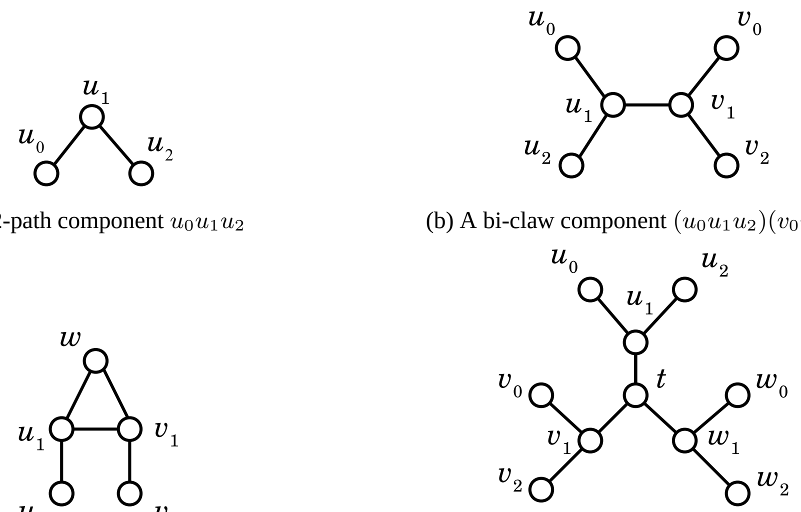
For non-negative integers $k_1, k_2, \ldots, k_m$, a multinomial coefficient $\frac{(\sum_{i=1}^m k_i)!}{k_1!\cdots k_m!}$ is denoted by $\binom{\sum_{i=1}^m k_i}{k_1,\ldots,k_m}$.

Lemma 1. *Let $k_1, k_2, \ldots, k_m$ be non-negative integers, where $m \geq 1$. Then for any positive reals γ_1, γ_2, …, γ_m such that $\sum_{i=1}^m 1/\gamma_i \leq 1$, it holds that*

$$\binom{\sum_{i=1}^m k_i}{k_1, k_2, \ldots, k_m} \leq \prod_{i=1}^m \gamma_i^{k_i}.$$

The set of vertices and edges in a graph H is denoted by $V(H)$ and $E(H)$, respectively. For a vertex v in a graph, let $N(v)$ denote a set of neighbors of v and let $N[v]$ denote a set of v and its neighbors (i.e., $N[v] = \{v\} \cup N(v)$). A vertex of degree d is called a *degree-d* vertex. The degree of a vertex v in a graph H is denoted by $d(v; H)$. For a set F of edges, we use $V(F)$ to denote a set of vertices incident on at least one edge in F, and we say that F *covers* a vertex set $S \subseteq V$ if $V(F) \supseteq S$. For a subset $S \subseteq V$ of vertices, $G[S]$ denote the subgraph of G induced by S. A cycle of length ℓ is called an *ℓ-cycle*, and is denoted by the sequence $v_1v_2 \ldots v_\ell$ of vertices in it, where the cycle contains edges $v_1v_2, \ldots, v_{\ell-2}v_{\ell-1}$ and $v_\ell v_1$. A connected component containing only one vertex is called *trivial*. We define five types of connected components as follows:
a *clique component*, a connected component that is a complete subgraph;
- a *2-path component*, a connected component consisting of a degree-2 vertex u_1 and its two degree-1 neighbors $u_0, u_2 \in N(u_1)$, denoted by $u_0u_1u_2$, as illustrated in Fig. 1(a);
- a *bi-claw component*, a connected component consisting of two adjacent degree-3 vertices u_1 and v_1 and their four degree-1 neighbors $u_0, u_2 \in N(u_1)$ and $v_0, v_2 \in N(v_1)$, denoted by $(u_0u_1u_2)(v_0v_1v_2)$, as illustrated in Fig. 1(b);
- a *legged triangle component* (or *leg-triangle component*), a connected component consisting of two adjacent degree-3 vertices u_1 and v_1, their two degree-1 neighbors $u_0 \in N(u_1)$ and $v_0 \in N(v_1)$ and one common degree-2 neighbor $w \in N(u_1) \cap N(v_1)$, denoted by $u_0(u_1wv_1)v_0$, as illustrated in Fig. 1(c); and
- a *tri-claw component*, a connected component consisting of three degree-3 vertices u_1, v_1 and w_1, their six degree-1 neighbors $u_0, u_2 \in N(u_1)$, $v_0, v_2 \in N(v_1)$ and $w_0, w_2 \in N(w_1)$ and their common degree-3 neighbor $t \in N(u_1) \cap N(v_1) \cap N(w_1)$, denoted by $t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$, as illustrated in Fig. 1(d).

The last four types of components, 2-path, bi-claw, leg-triangle and tri-claw components are called *bad components* collectively.



(a) A 2-path component $u_0u_1u_2$

(b) A bi-claw component $(u_0u_1u_2)(v_0v_1v_2)$

(c) A leg-triangle component $u_0(u_1wv_1)v_0$

(d) A tri-claw component $t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$

Fig. 1. The four types of bad components

2.2 Instances with covered and discarded vertices

Throughout our algorithm, we do not modify a given graph $G = (V, E)$ or a parameter k, but fix vertices to *covered* vertices or *discarded* vertices so that a pair of the sets C and D of covered and discarded vertices gives an instance (C, D) that asks to find an edge dominating set M of G such that $C \subseteq V(M) \subseteq V \setminus D$. We call such an edge dominating set a (C, D)-*eds* for short. An instance (C, D) is called *feasible* if it admits a (C, D)-eds, and is called k-*feasible* if it admits a (C, D)-eds M of size $|M| \leq k$. We call vertices in $V \setminus (C \cup D)$ *undecided* and denote by U the set of undecided vertices.

We use two kinds of fundamental branching operations. One is to branch on an undecided vertex $v \in U$ in (C, D): fix v as a new covered vertex in the first branch or as a new discarded vertex in the second branch. This is based on the fact that there is a (C, D)-eds M with $v \in V(M)$ or there is no such (C, D)-eds. Then we also fix all the vertices in $N(v)$ as covered vertices in the second branch, since any edge $e = vw$ incident to v needs to be incident to an edge dominating set at the vertex w. The other is to branch on a 4-cycle $v_0v_1v_2v_3$ over undecided vertices: fix vertices v_0 and v_2 as new covered vertices or fix vertices v_1 and v_3 as new covered vertices. This is based on the fact that for any edge dominating set M, the set $V(M)$ is a vertex cover and one of $\{v_0, v_2\}$ and $\{v_1, v_3\}$ is contained in any vertex cover [8]. Rooij and Bodlaender [7] found the following solvable case.

Lemma 2. [7] *A minimum (C, D)-eds of an instance (C, D) such that $G[U]$ contains only clique components can be found in polynomial time.*

We denote by U_1 the set of vertices of all clique components in $G[U]$, and let $U_2 = U \setminus U_1$. An instance (C, D) is called a *leaf instance* if $U_2 = \emptyset$. By Lemma 2, we only need to select vertices from U_2 to apply branching operations until all instances become leaf instances.

The next lower bound on the size of (C, D)-edses is immediate since for each clique component Q in $G[U]$, it holds that $|V(Q) \cap V(M)| \geq |V(Q)| - 1$.

Lemma 3. *For any (C, D)-eds M in a graph G, it holds that*

$$|V(M)| \geq |C| + \sum\{|V(Q)| - 1 \mid \text{clique components } Q \text{ in } G[U]\}.$$

Based on this, we define the *measure* μ of an instance (C, D) to be

$$\mu(C, D) = 2k - |C| - \sum\{|V(Q)| - 1 \mid \text{clique components } Q \text{ in } G[U]\}.$$

We do not need to generate any instances (C, D) with $\mu(C, D) < 0$ since they are not k-feasible. In this paper, we introduce the following new lower bound.

Lemma 4. *Let M be a (C, D)-eds in a graph G. Then for any subset $S \subseteq C$ it holds that*

$$|M| \geq \sum\{\lceil |V(H)|/2 \rceil \mid \text{components } H \text{ in } G[S]\} \geq \lceil |S|/2 \rceil.$$

Branching on a bad component H in $G[U_2]$ means to keep branching on vertices in $U_2 \cap V(H)$ until all vertices in $V(H)$ are contained in $C \cup D \cup U_1$. We treat a series of such branchings as an operation of branching on H that generates r new instances defined as follows. For each type of a bad component H, we define the number r and $C^{(j)}(H)$ (resp., $D^{(j)}(H)$), $j = 1, 2, \ldots, r$ to be a set of vertices of H fixed as covered (resp., discarded) vertices in the j-th branch:

For a 2-path component $H_1 = u_0u_1u_2$, by branching on u_1, we can branch on H_1 into $r = 2$ branches:
1. $C^{(1)}(H_1) = \{u_1\}$ and $D^{(1)}(H_1) = \emptyset$; and
2. $C^{(2)}(H_1) = \{u_0, u_2\}$ and $D^{(2)}(H_1) = \{u_1\}$.

For a bi-claw component $H_2 = (u_0u_1u_2)(v_0v_1v_2)$, where at least one of adjacent vertices u_1 and v_1 must be in $V(M)$ of any (C, D)-eds M, we can branch on this component into $r = 3$ branches:
1. $C^{(1)}(H_2) = \{u_1, v_1\}$ and $D^{(1)}(H_2) = \emptyset$;
2. $C^{(2)}(H_2) = \{u_0, u_2, v_1\}$ and $D^{(2)}(H_2) = \{u_1\}$; and
3. $C^{(3)}(H_2) = \{u_1, v_0, v_2\}$ and $D^{(3)}(H_2) = \{v_1\}$.

For a leg-triangle component $H_3 = u_0(u_1wv_1)v_0$, where at least one of adjacent vertices u_1 and v_1 must be in $V(M)$ of any (C, D)-eds M, we can branch on this component into $r = 3$ branches:
1. $C^{(1)}(H_3) = \{u_1, v_1\}$ and $D^{(1)}(H_3) = \emptyset$;
2. $C^{(2)}(H_3) = \{u_0, v_1, w\}$ and $D^{(2)}(H_3) = \{u_1\}$; and
3. $C^{(3)}(H_3) = \{u_1, v_0, w\}$ and $D^{(3)}(H_3) = \{v_1\}$.

For a tri-claw component $H_4 = t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$, we can branch on u_1, v_1 and w_1 sequentially to generate the following $r = 8$ branches:
1. $C^{(1)}(H_4) = \{u_1, v_1, w_1\}$ and $D^{(1)}(H_4) = \emptyset$;
2. $C^{(2)}(H_4) = \{t, u_0, u_2, v_1, w_1\}$ and $D^{(2)}(H_4) = \{u_1\}$;
3. $C^{(3)}(H_4) = \{t, u_1, v_0, v_2, w_1\}$ and $D^{(3)}(H_4) = \{v_1\}$;
4. $C^{(4)}(H_4) = \{t, u_1, v_1, w_0, w_2\}$ and $D^{(4)}(H_4) = \{w_1\}$;
5. $C^{(5)}(H_4) = \{t, u_0, u_2, v_0, v_2, w_1\}$ and $D^{(5)}(H_4) = \{u_1, v_1\}$;
6. $C^{(6)}(H_4) = \{t, u_1, v_0, v_2, w_0, w_2\}$ and $D^{(6)}(H_4) = \{v_1, w_1\}$;
7. $C^{(7)}(H_4) = \{t, u_0, u_2, v_1, w_0, w_2\}$ and $D^{(7)}(H_4) = \{u_1, w_1\}$; and
8. $C^{(8)}(H_4) = \{t, u_0, u_2, v_0, v_2, w_0, w_2\}$ and $D^{(8)}(H_4) = \{u_1, v_1, w_1\}$.

For each of the above branch, we define two kinds of values α and β which will be summed up to give lower bounds on the size of a (C', D')-eds of a leaf instance (C', D'). For each (i, j), let

$$\alpha_{i,j} = |C^{(j)}(H_i)| \text{ and } \beta_{i,j} = \sum\{\lceil |V(T)|/2 \rceil \mid \text{components } T \text{ in } G[C^{(j)}(H_i)]\}.$$

Observe that $\beta_{i,j}$ is a lower bound on the size of a $(C^{(j)}(H_i), \emptyset)$-eds by Lemma 4. For $(i, j) \in \{(1, 1), (1, 2), (2, 2), (2, 3), (3, 2), (4, 8)\}$, the graph $G[C^{(j)}(H_i)]$ contains only isolated vertices, and $\beta_{i,j} = |C^{(j)}(H_i)| = \alpha_{i,j}$. For other (i, j), the graph $G[C^{(j)}(H_i)]$ consists of exactly one nontrivial component of size $p \in \{2, 3\}$ and $|C^{(j)}(H_i)| - p$ isolated vertices, and $\beta_{i,j} = \lceil p/2 \rceil + (|C^{(j)}(H_i)| - p) = |C^{(j)}(H_i)| - 1 = \alpha_{i,j} - 1$.

In this paragraph, we introduce criteria in choosing 4-cycle/vertices to branch on used in our algorithm. For a subset $S \subseteq U_2$ of vertices, we let q_S and b_S denote the sum of $|V(Q)| - 1$ over all clique components Q and the number of bad components newly generated by removing S from $G[U_2]$, respectively. A 4-cycle $v_0v_1v_2v_3$ in $G[U_2]$ is called *admissible* if $b_{\{v_0,v_2\}} + b_{\{v_1,v_3\}} \leq 1$. A vertex v in $G[U_2]$ such that $b_v = x$ and $b_{N[v]} = y$ is called an *(x, y)-vertex*. A vertex v in $G[U_2]$ is called *optimal* if it satisfies a condition (c-i) below with the minimum i over all vertices in $G[U_2]$:

(c-1) v is a degree-3 $(0, 0)$-vertex;

(c-2) v is a degree-2 (x, y)-vertex with $x + y \leq 1$ and $q_v \geq 1$;

(c-3) (i) v is in an admissible 4-cycle;
(ii) v is a degree-d (x, y)-vertex such that $2 \leq d \leq 3$, $x + y \leq 1$ and $q_v + q_{N[v]} \geq 4 - d$;
(iii) v is a degree-d (x, y)-vertex such that $2 \leq d \leq 3$, $x + y \leq 1$, $q_{N[v]} = 3 - d$ and removing each of v and $N[v]$ produces no new 2-path component; or
(iv) v is a degree-3 $(0, 1)$-vertex such that $G[U_2 \setminus \{v\}]$ contains at least one degree-3 $(0, 0)$-vertex and removing $N[v]$ produces exactly one new 2-path component;
(c-4) v is a degree-2 vertex with $q_v = 1$;
(c-5) v is a degree-3 vertex; and
(c-6) v is a degree-2 vertex.

3 The Algorithm

Given a graph G and an integer k, our algorithm returns TRUE if it admits an edge dominationg set of size $\leq k$ or FALSE otherwise. The algorithm is designed to be a procedure that returns TRUE if a given instance (C, D) is k-feasible or FALSE otherwise, by branching on a vertex/4-cycle/bad component in (C, D) to generate new smaller instances $(C^{(1)}, D^{(1)}), \ldots, (C^{(r)}, D^{(r)})$, to each of which the procedure is recursively applied. The procedure is initially given an instance $(\emptyset, \emptyset)$, and always returns FALSE whenever $\mu(C, D) < 0$ holds.

Our algorithm takes three stages. The first stage keeps branching on vertices of degree ≥ 4, and retains the set $\mathcal{B}$ of all the produced bad components without branching on them. The second stage keeps branching on optimal vertices of degree ≤ 3, immediately branching on any newly produced bad component before it chooses the next optimal vertex to branch on. The third stage generates leaf instances by fixing all undecided vertices in the bad components in $\mathcal{B}$, where we try to decrease the number of leaf instances to be generated based on some lower bound on the size of solutions of leaf instances. To derive the lower bounds in the third stage, we let C_i store all vertices fixed to covered vertices during branching operations in the i-th stage. Formally EDSSTAGE1 is described as follows.

Algorithm EDSSTAGE1(C, D)

Input: A graph $G = (V, E)$ with an integer k, and subsets C and D of V (initially, $C = D = \emptyset$).
Output: TRUE if (C, D) is k-feasible or FALSE otherwise.

if $\mu(C, D) < 0$ **then**
  return FALSE
else if there is a vertex v of degree ≥ 4 in $G[U_2]$ **then**
  return EDSSTAGE1$(C \cup \{v\}, D) \vee$ EDSSTAGE1$(C \cup N(v), D \cup \{v\})$
else
  $C_1 := C$; $C_2 := \emptyset$;
  Let $\mathcal{B}$ store all bad components in $G[U_2]$;
  return EDSSTAGE2$(C_1, C_2, \mathcal{B}, D)$
end if

For a given instance (G, k) of PEDS, let $\mathcal{I}_1$ denote the set of all instances constructed immediately after the first stage. Let $V(\mathcal{B})$ denote the set of vertices in the bad components in $\mathcal{B}$. Given an instance $(C_1, C_2, \mathcal{B}, D) \in \mathcal{I}_1$, the second stage EDSSTAGE2 fixes all vertices in $U_2 \setminus V(\mathcal{B})$ to covered/discarded vertices by repeatedly branching on optimal vertices or any newly produced bad component in $G[U_2 \setminus V(\mathcal{B})]$ if it exists. During the second stage, the sets C_1 and $\mathcal{B}$ obtained in the first stage never change. When no vertex is left in $U_2 \setminus V(\mathcal{B})$, we switch to the third stage. Formally EDSSTAGE2 is described as follows.

Algorithm EDSSTAGE2$(C_1, C_2, \mathcal{B}, D)$

Input: A graph $G = (V, E)$ with an integer k, disjoint subsets $C_1, C_2, D \subseteq V$ and a set of bad components $\mathcal{B}$ in $G[U_2]$.
Output: TRUE if $(C_1 \cup C_2, D)$ is k-feasible or FALSE otherwise.

if $\mu(C_1 \cup C_2, D) < 0$ **then**
  return FALSE
else if there is a 2-path component H_1 in $G[U_2 \setminus V(\mathcal{B})]$ **then**
  return $\bigvee_{j=1,2}$ EDSSTAGE2$(C_1, C_2 \cup C^{(j)}(H_1), \mathcal{B}, D \cup D^{(j)}(H_1))$

else if there is a bi-claw component H_2 in $G[U_2 \setminus V(\mathcal{B})]$ **then**
  return $\bigvee_{1 \le j \le 3}$ EDSSTAGE2$(C_1, C_2 \cup C^{(j)}(H_2), \mathcal{B}, D \cup D^{(j)}(H_2))$
else if there is a leg-triangle component H_3 in $G[U_2 \setminus V(\mathcal{B})]$ **then**
  return $\bigvee_{1 \le j \le 3}$ EDSSTAGE2$(C_1, C_2 \cup C^{(j)}(H_3), \mathcal{B}, D \cup D^{(j)}(H_3))$
else if there is a tri-claw component H_4 in $G[U_2 \setminus V(\mathcal{B})]$ **then**
  return $\bigvee_{1 \le j \le 8}$ EDSSTAGE2$(C_1, C_2 \cup C^{(j)}(H_4), \mathcal{B}, D \cup D^{(j)}(H_4))$
else if $U_2 \setminus V(\mathcal{B}) \neq \emptyset$ **then**
  Choose an optimal vertex v in $G[U_2 \setminus V(\mathcal{B})]$;
  if v is in an admissible 4-cycle $v_0v_1v_2v_3$ of condition (c-4) **then**
    return EDSSTAGE2$(C_2 \cup \{v_0, v_2\}, D, \mathcal{B}, C_1)$ $\vee$ EDSSTAGE2$(C_1, C_2 \cup \{v_1, v_3\}, \mathcal{B}, D)$
  else
    return EDSSTAGE2$(C_1, C_2 \cup \{v\}, \mathcal{B}, D)$ $\vee$ EDSSTAGE2$(C_1, C_2 \cup N(v), \mathcal{B}, D \cup \{v\})$
  end if
else /* Now $U_2 = V(\mathcal{B})$ */
  return EDSSTAGE3$(C_1, C_2, \mathcal{B}, D)$
end if

Let $\mathcal{I}_2$ denote the set of all instances constructed immediately after the second stage. Consider an instance $I = (C_1, C_2, \mathcal{B}, D) \in \mathcal{I}_2$, where the graph $G[U_2]$ consists of the bad components in $\mathcal{B}$ retained at the first stage. Let $\mathcal{B}_1$ (resp., $\mathcal{B}_2, \mathcal{B}_3$ and $\mathcal{B}_4$) be the sets of 2-path (resp., bi-claw, leg-triangle and tri-claw) components in $\mathcal{B}$, and $y_i = |\mathcal{B}_i|$, $i = 1, 2, 3, 4$ in $I \in \mathcal{I}_2$. To obtain a leaf instance from the instance I, we need to fix all vertices in $V(\mathcal{B})$. The number of all leaf instances that can be constructed from the instance $I \in \mathcal{I}_2$ is $\prod_{i=1}^{4} r_i^{y_i} = 2^{y_1} \cdot 3^{y_2} \cdot 3^{y_3} \cdot 8^{y_4}$, where r_i is the number of subinstances generated by branching on a bad component $H \in \mathcal{B}_i$.

In the third stage, we avoid constructing of some "k-infeasible" leaf instances among all leaf instances. For a leaf instance $I' = (C' = C_1 \cup C_2 \cup C_3, D')$ obtained from the instance $I \in \mathcal{I}_2$, where C_3 denotes the set of undecided vertices in $V(\mathcal{B})$ that are fixed to covered vertices in I', we let $w_{i,j}$ be the number of bad components in $\mathcal{B}_i$ to which the j-th branch is applied to generate I', and call the vector $\boldsymbol{w}$ with these 16 entries $w_{i,j}$ the *occurrence vector* of I'. Note that $\sum_{i,j} \alpha_{i,j} w_{i,j} = |C_3|$ holds, and that $\sum_{i,j} \beta_{i,j} w_{i,j}$ is a lower bound on the size of (C_3, D')-eds by Lemma 4, since no edge in G joins two components in $\mathcal{B}$. We derive two necessary conditions for a vector $\boldsymbol{w}$ to be the occurrence vector of a k-feasible leaf instance $I' = (C', D')$. One is that $2k \ge 2|M| \ge |V(M)| \ge |C_1| + |C_2| + |C_3|$, i.e.,

$$2k \ge |C_1| + |C_2| + \sum_{i,j} \alpha_{i,j} w_{i,j}. \tag{1}$$

Observe that there is no edge between C_3 and C_2 in I', since any vertex in C_2 is contained in some component in $G[U_2 \setminus V(\mathcal{B})]$ during an execution of EDSSTAGE2. Hence $\sum_{i,j} \beta_{i,j} w_{i,j} + \lceil |C_2|/2 \rceil$ is a lower bound on the size of a $(C_3 \cup C_2, D')$-eds by Lemma 4, and another necessary condition is given by

$$k \ge |C_2|/2 + \sum_{i,j} \beta_{i,j} w_{i,j}. \tag{2}$$

Note that the number $\ell(\boldsymbol{w})$ of leaf instances I' whose occurrence vectors are given by $\boldsymbol{w}$ is

$$\ell(\boldsymbol{w}) = \binom{y_1}{w_{1,1}, w_{1,2}} \binom{y_2}{w_{2,1}, w_{2,2}, w_{2,3}} \binom{y_3}{w_{3,1}, w_{3,2}, w_{3,3}} \binom{y_4}{w_{4,1}, w_{4,2}, \ldots, w_{4,8}}. \tag{3}$$

For each instance $I = (C_1, C_2, \mathcal{B}, D) \in \mathcal{I}_2$, the third stage EDSSTAGE3 generates an occurrence vector $\boldsymbol{w}$ satisfying the conditions (1) and (2) and $\sum_j w_{i,j} = y_i$, $1 \le i \le 4$, and constructs all leaf instances $I' = (C_1 \cup C_2 \cup C_3, D')$ from $I \in \mathcal{I}_2$ with the vector $\boldsymbol{w}$, before it returns TRUE if one of the leaf instances is k-feasible or FALSE otherwise. Formally EDSSTAGE3 is described as follows.

Algorithm EDSSTAGE3$(C_1, C_2, \mathcal{B}, D)$

Input: A graph $G = (V, E)$ with an integer k, disjoint subsets $C_1, C_2, D \subseteq V$ and a set of bad components $\mathcal{B}$ in $G[U_2]$.
Output: TRUE if $(C_1 \cup C_2, D)$ is k-feasible or FALSE otherwise.
Let $\mathcal{B}_1$ (resp., $\mathcal{B}_2, \mathcal{B}_3$ and $\mathcal{B}_4$) be a set of 2-path (resp., bi-claw, leg-triangle and tri-claw) components in $\mathcal{B}$, and $y_i := |\mathcal{B}_i|$, $i = 1, 2, 3, 4$;

for each occurrence vector $\boldsymbol{w}$ that satisfies the conditions (1) and (2) and $\sum_j w_{i,j} = y_i$, $1 \le i \le 4$ **do**
  for each combination of partitions of $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ and $\mathcal{B}_4$ into
    $\mathcal{B}_1^{(1)} \cup \mathcal{B}_1^{(2)} = \mathcal{B}_1$, $\mathcal{B}_2^{(1)} \cup \mathcal{B}_2^{(2)} \cup \mathcal{B}_2^{(3)} = \mathcal{B}_2$, $\mathcal{B}_3^{(1)} \cup \mathcal{B}_3^{(2)} \cup \mathcal{B}_3^{(3)} = \mathcal{B}_3$, and
    $\mathcal{B}_4^{(1)} \cup \mathcal{B}_4^{(2)} \cup \cdots \cup \mathcal{B}_4^{(8)} = \mathcal{B}_4$ such that $|\mathcal{B}_j^{(j)}| = w_{i,j}$ for all i and j; **do**
    for each $j = 1, 2$ and each 2-path component $H_1 \in \mathcal{B}_1^{(j)}$ **do**
      $C_3 := C^{(j)}(H_1)$; $D := D \cup D^{(j)}(H_1)$
    end for;
    for each $j = 1, 2, 3$ and each bi-claw component $H_2 \in \mathcal{B}_2^{(j)}$ **do**
      $C_3 := C^{(j)}(H_2)$; $D := D \cup D^{(j)}(H_2)$
    end for;
    for each $j = 1, 2, 3$ and each leg-triangle component $H_3 \in \mathcal{B}_3^{(j)}$ **do**
      $C_3 := C^{(j)}(H_3)$; $D := D \cup D^{(j)}(H_3)$
    end for;
    for each $j = 1, 2, \ldots, 8$ and each tri-claw component $H_4 \in \mathcal{B}_4^{(j)}$ **do**
      $C_3 := C^{(j)}(H_4)$; $D := D \cup D^{(j)}(H_4)$
    end for; /* Now $U_2 = \emptyset$ and $(C_1 \cup C_2 \cup C_3, D)$ is a leaf instance */
    Test whether $(C = C_1 \cup C_2 \cup C_3, D)$ is k-feasible or not by computing a minimum (C, D)-eds by Lemma 2
  end for
end for;
if there is a k-feasible leaf instance $(C_1 \cup C_2 \cup C_3, D)$ in the for loop **then**
  return TRUE
else
  return FALSE
end if

4 The Analysis

For a given instance (G, k) of PEDS, let $\mathcal{I}_i$, $i = 1, 2, 3$ be the set of all instances constructed immediately after the i-th stage during the execution of EDSSTAGE1$(\emptyset, \emptyset)$, where $\mathcal{I}_3$ is the set of all leaf instances, which correspond to the leaf nodes of the search tree of the execution. To analyze the time complexity of our algorithm, it suffices to derive an upper bound on $|\mathcal{I}_3|$.

Lemma 5. *For any non-negative integer x_1, the number of instances $I = (C_1, \emptyset, \mathcal{B}, D) \in \mathcal{I}_1$ with $|C_1| = x_1$ is $O(1.380278^{x_1})$.*

Lemma 6. *For any non-negative integer x_2 and an instance $I = (C_1, \emptyset, \mathcal{B}, D) \in \mathcal{I}_1$, the number of instances $I' = (C_1, C_2, \mathcal{B}, D') \in \mathcal{I}_2$ with $|C_2| = x_2$ that can be generated from I is $O(1.494541^{x_2})$.*

From these, we obtain the next.

Lemma 7. *For any non-negative integers x_1 and x_2, the number of instances $(C_1, C_2, \mathcal{B}, D) \in \mathcal{I}_2$ such that $|C_1| = x_1$ and $|C_2| = x_2$ is $O(1.380278^{x_1} \cdot 1.494541^{x_2})$.*

Note that the number of combinations (x_1, x_2) for $(|C_1|, |C_2|)$ is $O(n^2)$. For a given instance $(C_1, C_2, \mathcal{B}, D) \in \mathcal{I}_2$, the number of possible occurrence vectors $\boldsymbol{w}$ satisfying the conditions (1) and (2) and $\sum_j w_{i,j} = y_i$, $1 \le i \le 4$ is also bounded by a polynomial of n. To show that our algorithm runs in $O^*(2.2351^k)$ time, it suffices to prove that the number of leaf instances generated from an instance $I = (C_1, C_2, \mathcal{B}, D) \in \mathcal{I}_2$ with specified size $|C_1| = x_1$ and $|C_2| = x_2$ and a specified occurrence vector $\boldsymbol{w}$ is $O^*(2.2351^k)$. Let $\mathcal{I}_3(x_1, x_2, \boldsymbol{w})$ denote the set of all such leaf instances. By Lemma 7 and (3), we see that $|\mathcal{I}_3(x_1, x_2, \boldsymbol{w})| = O(1.380278^{x_1} \cdot 1.494541^{x_2} \cdot \ell(\boldsymbol{w}))$.

In what follows, we derive an upper bound on $O(1.380278^{x_1} \cdot 1.494541^{x_2} \cdot \ell(\boldsymbol{w}))$ under the constraints (1) and (2). For this, we merge some entries in $\boldsymbol{w}$ into ten numbers by $z_{1,1} = w_{1,1}$, $z_{1,2} = w_{1,2}$, $z_{2,1} = w_{2,1}$, $z_{2,2} = w_{2,2} + w_{2,3}$, $z_{3,1} = w_{3,1}$, $z_{3,2} = w_{3,2} + w_{3,3}$, $z_{4,1} = w_{4,1}$, $z_{4,2} = w_{4,2} + w_{4,3} + w_{4,4}$, $z_{4,3} = w_{4,5} + w_{4,6} + w_{4,7}$ and $z_{4,4} = w_{4,8}$. Then $\ell(\boldsymbol{w})$ is restated as

$$\binom{z_{1,1}+z_{1,2}}{z_{1,1}, z_{1,2}} \cdot \binom{z_{2,1}+z_{2,2}}{z_{2,1}, z_{2,2}} \cdot 2^{z_{2,2}} \cdot \binom{z_{3,1}+z_{3,2}}{z_{3,1}, z_{3,2}} \cdot 2^{z_{3,2}} \cdot \binom{z_{4,1}+z_{4,2}+z_{4,3}+z_{4,4}}{z_{4,1}, z_{4,2}, z_{4,3}, z_{4,4}} \cdot 3^{z_{4,2}+z_{4,3}},$$

which is bounded from above by an exponential function in Lemma 1

$$\gamma_{1,1}^{z_{1,1}}\gamma_{1,2}^{z_{1,2}}\cdot\gamma_{2,1}^{z_{2,1}}\gamma_{2,2}^{z_{2,2}}\cdot\gamma_{3,1}^{z_{3,1}}\gamma_{3,2}^{z_{3,2}}\cdot\gamma_{4,1}^{z_{4,1}}\gamma_{4,2}^{z_{4,2}}\gamma_{4,3}^{z_{4,3}}\gamma_{4,4}^{z_{4,4}}$$

for any positive reals $\gamma_{1,1}, \gamma_{1,2}, \gamma_{2,1}, \gamma_{2,2}, \gamma_{3,1}, \gamma_{3,2}, \gamma_{4,1}, \gamma_{4,2}, \gamma_{4,3}$ and $\gamma_{4,4}$ such that $1/\gamma_{1,1}+1/\gamma_{1,2}\le 1$, $1/\gamma_{2,1}+2/\gamma_{2,2}\le 1$, $1/\gamma_{3,1}+2/\gamma_{3,2}\le 1$ and $1/\gamma_{4,1}+3/\gamma_{4,2}+3/\gamma_{4,3}+1/\gamma_{4,4}\le 1$. Then we have

$$|\mathcal{I}_3(x_1,x_2,\boldsymbol{w})| = O(1.380278^{x_1}\cdot 1.494541^{x_2}\gamma_{2,1}^{z_{2,1}}\gamma_{2,2}^{z_{2,2}}\gamma_{3,1}^{z_{3,1}}\gamma_{3,2}^{z_{3,2}}\gamma_{4,1}^{z_{4,1}}\gamma_{4,2}^{z_{4,2}}\gamma_{4,3}^{z_{4,3}}\gamma_{4,4}^{z_{4,4}}),$$

which is bounded by

$$\begin{aligned} O(\max\{&1.380278^{1/c_1}, 1.494541^{1/c_2}, \gamma_{11}^{1/c_{1,1}}, \gamma_{12}^{1/c_{1,2}}, \gamma_{21}^{1/c_{2,1}}, \gamma_{22}^{1/c_{2,2}},\\ &\gamma_{31}^{1/c_{3,1}}, \gamma_{32}^{1/c_{3,2}}, \gamma_{41}^{1/c_{4,1}}, \gamma_{42}^{1/c_{4,2}}, \gamma_{43}^{1/c_{4,3}}, \gamma_{44}^{1/c_{4,4}}\}^k\ ) \end{aligned} \tag{4}$$

for any constants c_1, c_2 and $\{c_{i,j}\}$ such that

$$\begin{aligned} k \ge\ & c_1x_1 + c_2x_2 + c_{1,1}z_{1,1} + c_{1,2}z_{1,2} + c_{2,1}z_{2,1} + c_{2,2}z_{2,2}\\ &+ c_{3,1}z_{3,1} + c_{3,2}z_{3,2} + c_{4,1}z_{4,1} + c_{4,2}z_{4,2} + c_{4,3}z_{4,3} + c_{4,4}z_{4,4}. \end{aligned} \tag{5}$$

Conditions (1) and (2) are restated as

$$\begin{aligned} k \ge\ & x_1/2 + x_2/2 + (z_{1,1}+2z_{1,2})/2 + (2z_{2,1}+3z_{2,2})/2\\ &+ (2z_{3,1}+3z_{3,2})/2 + (3z_{4,1}+5z_{4,2}+6z_{4,3}+7z_{4,4})/2; \end{aligned} \tag{6}$$

$$\begin{aligned} k \ge\ & x_2/2 + (z_{1,1}+2z_{1,2}) + (z_{2,1}+3z_{2,2})\\ &+ (z_{3,1}+2z_{3,2}) + (3z_{4,1}+4z_{4,2}+5z_{4,3}+7z_{4,4}). \end{aligned} \tag{7}$$

As a linear combination of (6) and (7) with λ and $(1-\lambda)$, we get (5) for constants $c_1 = \lambda/2$, $c_2 = 1/2$, $c_{1,1} = 1-\lambda/2$, $c_{1,2} = 2-\lambda$, $c_{2,1} = 1$, $c_{2,2} = 3-3\lambda/2$, $c_{3,1} = 1$, $c_{3,2} = 2-\lambda/2$, $c_{4,1} = 3-3\lambda/2$, $c_{4,2} = 4-3\lambda/2$, $c_{4,3} = 3-2\lambda$ and $c_{4,4} = 7-7\lambda/2$.

From (4), we obtain $|\mathcal{I}_3(x_1,x_2,\boldsymbol{w})| = O\left(2.2351^k\right)$ by setting $\lambda = 0.80142$, $\gamma_{1,1} = 1.61804$, $\gamma_{1,2} = 2.61804$, $\gamma_{2,1} = 2.10457$, $\gamma_{2,2} = 3.81068$, $\gamma_{3,1} = 2.23510$, $\gamma_{3,2} = 3.61931$, $\gamma_{4,1} = 3.60818$, $\gamma_{4,2} = 7.36647$, $\gamma_{4,3} = 11.29854$ and $\gamma_{4,4} = 19.96819$. This establishes the next theorem.

Theorem 1. *Algorithm* EDSSTAGE1, *accompanied by Algorithm* EDSSTAGE2 *and* EDSSTAGE3, *can solve the parameterized edge dominating set problem in $O^*(2.2351^k)$ time and polynomial space.*

5 A Related Problem: The Parameterized Weighted Edge Dominating Set Problem

We also consider a weighted variant of PEDS. The *weighted edge dominating set problem* (WEDS) is, given a graph $G = (V, E)$ with an edge weight function $\omega : E \to \mathbb{R}_{\ge 0}$, to find an edge dominating set M of minimum total weight $\omega(M) = \sum_{e\in M}\omega(e)$. The *parameterized weighted edge dominating set problem* (PWEDS) is, given a graph $G = (V, E)$ with an edge weight function $\omega : E \to \mathbb{R}_{\ge 1}$ and a positive real k, to test whether there is an edge dominating set M such that $\omega(M) \le k$. We show that a modification of our algorithm for PEDS can solve PWEDS in the same time and space complexities as our algorithm does PEDS.

For PWEDS we use the same terminologies and notations as for PEDS; for example, an instance of PWEDS is also denoted by (C, D). Rooij and Bodlaender [7] found the following solvable case for a weighted variant of EDS.

Lemma 8. [7] *A minimum (C, D)-eds of an instance (C, D) of WEDS such that $G[U]$ contains only clique components of size ≤ 3 can be found in polynomial time.*

Based on this lemma, for PWEDS we modify U_1 to be the set of vertices of clique components of size ≤ 3 in $G[U]$. We call our algorithm to which this modification is applied a modified algorithm. This modification brings the following corollary.

Corollary 1. *The modified algorithm can solve the parameterized weighted edge dominating set problem in $O^*(2.2351^k)$ time and polynomial space.*

Proof. We first show the correctness. If an edge dominating set M of G is k-feasible, i.e., $\omega(M) \leq k$, then it holds that $|V(M)| \leq 2k$ and $|M| \leq k$ since $\omega(e) \geq 1$ for any edge $e \in E$. This ensures the correctness of the measure $\mu(C, D)$ and the conditions (1) and (2) for an instance (C, D) of the weighted variant. Therefore we can solve PWEDS by the same branching method as PEDS.

Second we show the time complexity is the same as PEDS. Only difference between our algorithm for PEDS and one for PWEDS is treatment of clique components of size ≥ 4. In what follows, we describe the treatment by the modified algorithm and it guarantees that the time complexity is $O^*(2.2351^k)$. For a clique component H of size ≥ 5 of an instance (C, D), the degree of a vertex of H in $G[U_2]$ is $|V(H)| - 1 \geq 4$, on which therefore the modified algorithm branches in the first stage. For a clique component H of size 4 of an instance (C, D), a vertex of H satisfies condition (c-2), on which therefore the algorithm branches in the second stage. □

6 Conclusion

In this paper, we have presented an $O^*(2.2351^k)$-time and polynomial-space algorithm to PEDS. The algorithm retains bad components produced at the first stage for branching on vertices of degree ≥ 4, and branching on the remaining undecided vertices not in clique components by choosing 4-cycles/vertices to branch on carefully. Based on our new lower bound on the size of (C, D)-edses, we derived an upper bound on the number of leaf instances generated in the third stage. We have also shown that a modification of our algorithm can solve PWEDS in the same time and space complexities as PEDS.

For a possible achievement of further improved algorithms, it is still left to modify the first stage of our algorithm to branch on vertices of degree ≤ 4 in the second stage and to identify several new components as bad components.

References

1. Binkele-Raible, D., Fernau, H.: Enumerate and Measure: Improving Parameter Budget Management. IPEC 2010. LNCS 6478: 38-49 (2010)
2. Fernau, H.: Edge Dominating Set: Efficient Enumeration-Based Exact Algorithms. IWPEC 2006. LNCS 4169: 142-153 (2006)
3. Fomin, F., Gaspers, S., Saurabh, S., Stepanov, A.: On Two Techniques of Combining Branching and Treewidth. Algorithmica 54(2): 181-207 (2009)
4. Garey, M. R., Johnson, D. S.: Computers and Intractability: A Guide to The Theory of NP-Completeness. Freeman, San Francisco (1979)
5. Raman, V., Saurabh, S., Sikdar, S.: Efficient Exact Algorithms through Enumerating Maximal Independent Sets and Other Techniques. Theory of Computing Systems 42(3): 563-587 (2007)
6. Randerath, B., Sciermeyer, I.: Exact Algorithms for Minimum Dominating Set. Technical Report zaik 2005-501, Universität zu Köln, Cologne, Germany (2005)
7. van Rooij, J. M. M., Bodlaender, H. L.: Exact Algorithms for Edge Domination. Algorithmica 64(4): 535-563 (2012)
8. Xiao, M.: A Simple and Fast Algorithm for Maximum Independent Set in 3-Degree Graphs. WALCOM 2010. LNCS 5942: 281-292 (2010)
9. Xiao, M., Kloks, T., Poon, S.-H.: New Parameterized Algorithms for the Edge Dominating Set Problem. TCS 511: 147-158 (2013)
10. Xiao, M., Nagamochi, H.: Parameterized Edge Dominating Set in Cubic Graphs. FAW-AAIM 2011. LNCS 6681: 100-112 (2011)
11. Xiao, M., Nagamochi, H.: A Refined Exact Algorithm for Edge Dominating Set. TAMC 2012. LNCS 7287: 360-372 (2012)
12. Xiao, M., Nagamochi, H.: Exact Algorithms for Annotated Edge Dominating Set in Graphs with Degree Bounded by 3. IEICE TRANSACTIONS on Information and Systems E96-D(3): 408-418 (2013)
13. Yanakakis, M., Gavril., F.: Edge Dominating Set in Graphs. SIAM J. Appl. Math. 38(3): 364-372 (1980)

Appendix A

Lemma 1. *Let $k_1, k_2, \ldots, k_m$ be non-negative integers, where $m \geq 1$. Then for any positive reals γ_1, γ_2, $\ldots$, γ_m such that $\sum_{i=1}^m 1/\gamma_i \leq 1$, it holds that*

$$\binom{\sum_{i=1}^m k_i}{k_1, k_2, \ldots, k_m} \leq \prod_{i=1}^m \gamma_i^{k_i}.$$

Proof. We proceed by an induction on $\sum_{i=1}^m k_i$ to prove the lemma.
I. The lemma holds when $\sum_{i=1}^m k_i = 0$, since the both sides of the inequality in the lemma become 1.
II. Assume that the lemma holds for any instance $\{k'_1, k'_2, \ldots, k'_m\}$ such that $\sum_{i=1}^m k_i \leq K$ for some integer $K \geq 0$. We show that the lemma holds for any instance $\{k_1, k_2, \ldots, k_m\}$ with $\sum_{i=1}^m k_i = K+1$. If $k_j = 0$ for some j, where $m \geq 2$ by $\sum_{i=1}^m k_i = K+1 > 0$, then it suffices to show that the lemma holds for the instance $\{k_1, k_2, \ldots, k_m\} \setminus \{k_j\}$, since $\gamma_j^{k_j} = 1$ for any choice of $\{\gamma_1, \gamma_2, \ldots, \gamma_m\}$. Hence we assume without loss of generality that $k_i \geq 1$ for all $i = 1, 2, \ldots, m$. Let $\gamma_1, \gamma_2, \ldots, \gamma_m$ satisfy $\sum_{i=1}^m 1/\gamma_i \leq 1$. Using Pascal's rule and the inductive hypothesis, we obtain the following inequality:

$$\begin{aligned}
&\binom{K+1}{k_1, k_2, \ldots, k_m} \\
&= \binom{K}{k_1 - 1, k_2, \ldots, k_m} + \binom{K}{k_1, k_2 - 1, \ldots, k_m} + \cdots + \binom{K}{k_1, k_2, \ldots, k_m - 1} \\
&\leq \gamma_1^{k_1-1}\gamma_2^{k_2} \cdots \gamma_m^{k_m} + \gamma_1^{k_1}\gamma_2^{k_2-1} \cdots \gamma_m^{k_m} + \cdots + \gamma_1^{k_1}\gamma_2^{k_2} \cdots \gamma_m^{k_m-1} \\
&= \gamma_1^{k_1}\gamma_2^{k_2} \cdots \gamma_m^{k_m} \left(\frac{1}{\gamma_1} + \frac{1}{\gamma_2} + \cdots + \frac{1}{\gamma_m}\right) \leq \gamma_1^{k_1}\gamma_2^{k_2} \cdots \gamma_m^{k_m}.
\end{aligned}$$

This proves that the lemma also holds for any instance $\{k_1, k_2, \ldots, k_m\}$ with $\sum_{i=1}^m k_i = K+1$. □

Lemma 4. *Let M be a (C, D)-eds in a graph G. Then for any subset $S \subseteq C$ it holds that*

$$|M| \geq \sum\{\lceil |V(H)|/2 \rceil \mid \text{components } H \text{ in } G[S]\} \geq \lceil |S|/2 \rceil.$$

Proof. For each component H in $G[S]$ with a subset $S \subseteq C$, the minimal subset $M_H \subseteq M$ that covers $V(H)$ contains at least $\lceil |V(H)|/2 \rceil$ edges. Since there is no edge between two components in $G[S]$, minimal subsets M_H for all components H in $G[S]$ are disjoint, indicating that $|M| \geq \sum\{|M_H| \mid$ components H in $G[S]\} \geq \sum\{\lceil |V(H)|/2 \rceil \mid$ components H in $G[S]\}$, which is clearly at least $\lceil |S|/2 \rceil$. □

In what follows, we prove Lemmata 5 and 6. Let $T(\mu)$ be the maximum number of leaf instances that can be generated from an instance I with measure μ.

Lemma 5. *For any non-negative integer x_1, the number of instances $I = (C_1, \emptyset, \mathcal{B}, D) \in \mathcal{I}_1$ with $|C_1| = x_1$ is $O(1.380278^{x_1})$.*

Proof. At the first stage, the algorithm branches on a vertex v of degree ≥ 4 in $G[U_2]$. When the algorithm branches on v by fixing it as a covered vertex or a discarded vertex, $\{v\}$ (resp., $N(v)$) is added to the set C, and the measure μ decreases by 1 (resp., $|N(v)| \geq 4$). Hence we have the following recurrence:

$$T(\mu) \leq T(\mu - 1) + T(\mu - 4),$$

which solves to $T(\mu) = O(1.380278^{\mu})$. This proves the lemma. □

We here restate Lemma 6.

Lemma 6. *For any non-negative integer x_2 and an instance $I = (C_1, \emptyset, \mathcal{B}, D) \in \mathcal{I}_1$, the number of instances $I' = (C_1, C_2, \mathcal{B}, D') \in \mathcal{I}_2$ with $|C_2| = x_2$ that can be generated from I is $O(1.494541^{x_2})$.*

We use U'_2 to denote $U_2 \setminus V(\mathcal{B})$. To prove Lemma 6, we derive recurrences for branchings executed by Algorithm EDSSTAGE2. We first show recurrences for branching on bad components only.

Lemma 9. *Assume that Algorithm* EDSSTAGE2 *branches on a bad component H in $G[U_2']$. If H is a 2-path component, then the algorithm branches on H with the following recurrence:*

$$T(\mu) \le T(\mu - 1) + T(\mu - 2),$$

which solves to $T(\mu) = O(1.6181^\mu)$. If H is a bi-claw or leg-triangle component, then the algorithm branches on H with the following recurrence:

$$T(\mu) \le T(\mu - 2) + 2T(\mu - 3),$$

which solves to $T(\mu) = O(1.5214^\mu)$. If H is a tri-claw component, then the algorithm branches on H with the following recurrence:

$$T(\mu) \le T(\mu - 3) + 3T(\mu - 5) + 3T(\mu - 6) + T(\mu - 7),$$

which solves to $T(\mu) = O(1.5042^\mu)$.

Proof. In the i-th branch of each bad component H, all vertices in $C^{(i)}(H)$ are fixed as covered vertices and thereby the measure decreases by $|C^{(i)}(H)|$. Therefore we have the above recurrences. □

Observe that Algorithm EDSSTAGE2 branches on a bad component with the recurrence shown in Lemma 9, which is not good enough to establish Lemma 6. In our analysis, we combine a branching on a bad component together with the branching on the optimal vertex v (or the admissible 4-cycle on it) that produces the bad component, which yields a recurrence better than those in Lemma 9. In the case where the branching on v and the all bad components produced by any of the branchings to v yields a recurrence even not good enough to establish Lemma 6, we further combine it with a possible branching on a vertex of condition (c-1), (c-2) or (c-3)(iv) produced by the branching to v. In what follows, for each $i = 1, 2, \dots, 6$ in this order, we analyze the branching of an optimal vertex v satisfying condition (c-i) to derive such a recurrence.

Lemma 10. *Algorithm* EDSSTAGE2 *branches on a vertex v satisfying condition* (c-1) *in $G[U_2']$ together with possible branchings on the resulting new bad components with the following recurrence:*

$$T(\mu) \le 2T(\mu - 3) + 2T(\mu - 4), \tag{8}$$

which solves to $T(\mu) = O(1.494541^\mu)$.

Proof. Since v is a vertex satisfying condition (c-1), v is a degree-3 $(0, 0)$-vertex in $G[U_2']$. Neither of the first and second branches produces a new bad component. Therefore the algorithm branches on v with the following recurrence:

$$T(\mu) \le T(\mu - 1) + T(\mu - 3),$$

which solves to $T(\mu) = O(1.4656^\mu)$ and is better than the recurrence (8). □

Lemma 11. *Algorithm* EDSSTAGE2 *branches on an optimal vertex satisfying condition* (c-2) *in $G[U_2']$ together with possible branchings on the resulting new bad components with a recurrence not worse than* (8).

Proof. Since v is an optimal vertex satisfying condition (c-2), v is a degree-2 (x, y)-vertex with $x+y \le 1$ and $q_v \ge 1$ in $G[U_2']$. We distinguish two cases: Case 1. $x + y = 0$; and Case 2. $x + y = 1$.

Case 1. $x = y = 0$: In any of the first and second branches, no bad component is newly produced. Therefore the algorithm branches on v with the following recurrence:

$$T(\mu) \le T(\mu - 2) + T(\mu - 2),$$

which solves to $T(\mu) = O(1.4143^\mu)$.

Case 2. $x + y = 1$: In one of the first and second branches, exactly one bad component H is newly produced, and then the algorithm branches on it; and in the other branch, no bad component is newly produced. In the following, we derive recurrences for branching on v together with branching on H. When H is a 2-path component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-2) + T(\mu-2-1) + T(\mu-2-2) \\ &= T(\mu-2) + T(\mu-3) + T(\mu-4), \end{aligned} \tag{9}$$

which solves to $T(\mu) = O(1.4656^{\mu})$. When H is a bi-claw or leg-triangle component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-2) + T(\mu-2-2) + 2T(\mu-2-3) \\ &= T(\mu-2) + T(\mu-4) + 2T(\mu-5), \end{aligned} \tag{10}$$

which solves to $T(\mu) = O(1.4560^{\mu})$. When H is a tri-claw component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-2) + T(\mu-2-3) + 3T(\mu-2-5) + 3T(\mu-2-6) + T(\mu-2-7) \\ &= T(\mu-2) + T(\mu-5) + 3T(\mu-7) + 3T(\mu-8) + T(\mu-9), \end{aligned} \tag{11}$$

which solves to $T(\mu) = O(1.4634^{\mu})$.

Since all the recurrences obtained in Cases 1 and 2 are better than the recurrence (8), the lemma holds. □

Lemma 12. *Algorithm* EDSSTAGE2 *branches on an optimal vertex satisfying condition* (c-3) *in $G[U_2']$ together with possible branchings on the resulting new bad components with a recurrence not worse than* (8)*.*

Proof. Since v is an optimal vertex satisfying condition (c-3), v is in one of the following four cases: (i) v is in an admissible 4-cycle; (ii) v is a degree-d (x, y)-vertex such that $2 \le d \le 3$, $x + y \le 1$ and $q_v + q_{N[v]} \ge 4 - d$; (iii) v is a degree-d (x, y)-vertex such that $2 \le d \le 3$, $x + y \le 1$, $q_{N[v]} = 3 - d$ and removing each of v and $N[v]$ produces no new 2-path component; and (iv) v is a degree-3 $(0, 1)$-vertex such that removing $N[v]$ produces exactly one new 2-path component, and $G[U_2 \setminus \{v\}]$ contains at least one degree-3 $(0, 0)$-vertex. We distinguish three cases: Case (i) or (ii); Case (iii); and Case (iv).

Case (i) or (ii): When the algorithm branches on v (or the admissible 4-cycle on it) in $G[U_2']$, we have one of the following two recurrences:

$$T(\mu) \le T(\mu-2) + T(\mu-2), \tag{12}$$

which solves to $T(\mu) = O(1.4143^{\mu})$; and

$$T(\mu) \le T(\mu-1) + T(\mu-4), \tag{13}$$

which solves to $T(\mu) = O(1.3803^{\mu})$, and at most one bad component H is newly produced in one of the first and second branches. We consider three subcases (a)-(c).

Case (a). The algorithm branches on v (or the admissible 4-cycle on it) in $G[U_2']$ with the recurrence (12) and exactly one bad component H is produced in one of the first and second branches: When H is a 2-path component, we have the recurrence (9). When H is a bi-claw or leg-triangle component, we have the recurrence (10). When H is a tri-claw component, we have the recurrence (11).

Case (b). The algorithm branches on v in $G[U_2']$ with the recurrence (13) and exactly one bad component H is produced in the first branch: When H is a 2-path component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1-1) + T(\mu-1-2) + T(\mu-4) \\ &= T(\mu-2) + T(\mu-3) + T(\mu-4), \end{aligned}$$

which solves to $T(\mu) = O(1.4656^{\mu})$. When H is a bi-claw or leg-triangle component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1-2) + 2T(\mu-1-3) + T(\mu-4) \\ &= T(\mu-3) + 3T(\mu-4), \end{aligned}$$

which solves to $T(\mu) = O(1.4527^{\mu})$. When H is a tri-claw component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1-3)+3T(\mu-1-5)+3T(\mu-1-6)+T(\mu-1-7)+T(\mu-4) \\ &= 2T(\mu-4)+3T(\mu-6)+3T(\mu-7)+T(\mu-8), \end{aligned}$$

which solves to $T(\mu) = O(1.4629^{\mu})$.

Case (c). The algorithm branches on v in $G[U_2']$ with the recurrence (13) and exactly one bad component H is produced in the second branch: When H is a 2-path component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1)+T(\mu-4-1)+T(\mu-4-2) \\ &= T(\mu-1)+T(\mu-5)+T(\mu-6), \end{aligned}$$

which solves to $T(\mu) = O(1.4197^{\mu})$. When H is a bi-claw or leg-triangle component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1)+T(\mu-4-2)+2T(\mu-4-3) \\ &= T(\mu-1)+T(\mu-6)+2T(\mu-7), \end{aligned}$$

which solves to $T(\mu) = O(1.4190^{\mu})$. When H is a tri-claw component, we have the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1)+T(\mu-4-3)+3T(\mu-4-5)+3T(\mu-4-6)+T(\mu-4-7) \\ &= T(\mu-1)+T(\mu-7)+3T(\mu-9)+3T(\mu-10)+T(\mu-11), \end{aligned}$$

which solves to $T(\mu) = O(1.4320^{\mu})$.

Case (iii): When $x = y = 0$; i.e., neither of the first and second branches produces a new bad component, the algorithm branches on v with the following recurrence:

$$T(\mu) \le T(\mu-1)+T(\mu-3),$$

which solves to $T(\mu) = O(1.4656^{\mu})$.

Consider the case where $x + y = 1$; i.e., one of the first and second branches produces exactly one new bad component H other than a 2-path component whereas the other branch produces no new bad component. The algorithm branches on v together with branching on H with one of the following four recurrences. When $x = 1, y = 0$ and H is a bi-claw or leg-triangle component, we have

$$\begin{aligned} T(\mu) &\le T(\mu-1-2)+2T(\mu-1-3)+T(\mu-3) \\ &= 2T(\mu-3)+2T(\mu-4), \end{aligned}$$

which solves to $T(\mu) = O(1.494541^{\mu})$. When $x = 1, y = 0$ and H is a tri-claw component, we have

$$\begin{aligned} T(\mu) &\le T(\mu-1-3)+3T(\mu-1-5)+3T(\mu-1-6)+T(\mu-1-7)+T(\mu-3) \\ &= T(\mu-3)+T(\mu-4)+3T(\mu-6)+3T(\mu-7)+T(\mu-8), \end{aligned}$$

which solves to $T(\mu) = O(1.4914^{\mu})$. When $x = 0, y = 1$ and H is a bi-claw or leg-triangle component, we have

$$\begin{aligned} T(\mu) &\le T(\mu-1)+T(\mu-3-2)+2T(\mu-3-3) \\ &= T(\mu-1)+T(\mu-5)+2T(\mu-6), \end{aligned}$$

which solves to $T(\mu) = O(1.4841^{\mu})$. When $x = 0, y = 1$ and H is a tri-claw component, we have

$$\begin{aligned} T(\mu) &\le T(\mu-1)+T(\mu-3-3)+3T(\mu-3-5)+3T(\mu-3-6)+T(\mu-3-7) \\ &= T(\mu-1)+T(\mu-6)+3T(\mu-8)+3T(\mu-9)+T(\mu-10), \end{aligned}$$

which solves to $T(\mu) = O(1.4842^{\mu})$.

Case (iv): In the first branch, no bad component and a degree-3 $(0, 0)$-vertex u are newly produced, and then the algorithm branches on u, since u satisfies condition (c-1) after fixing v as a covered vertex.

In the second branch, exactly one 2-path component is newly produced. Therefore the algorithm branches on v together with branching on u and the 2-path component with the following recurrence:

$$\begin{aligned} T(\mu) &\le T(\mu-1-1) + T(\mu-1-3) + T(\mu-3-1) + T(\mu-3-2) \\ &= T(\mu-2) + 2T(\mu-4) + T(\mu-5), \end{aligned} \tag{14}$$

which solves to $T(\mu) = O(1.4865^{\mu})$.

Since all the recurrences obtained in Cases (i)-(iv) are not worse than the recurrence (8), the lemma holds. □

We say that an instance (C, D) is *reduced up to* (c-i) if $G[U_2']$ in (C, D) has no vertices of degree ≥ 4, no vertices satisfying any of conditions (c-1) to (c-i) and no bad components.

Lemma 13. *Let (C, D) be an instance reduced up to* (c-3).

(i) *After removing any vertex $v \in U_2'$ in (C, D), the set of newly produced bad components in $G[U_2' \setminus \{v\}]$ is a set of three 2-path components or an empty set.*

(ii) *Every degree-2 vertex u in $G[U_2']$ with $q_u = 1$ in (C, D) has a degree-3 neighbor $v \in U_2'$ removal of which produces exactly three 2-path components in $G[U_2' \setminus \{u\}]$. Conversely, every degree-3 vertex v in $G[U_2']$ of (C, D) removal of which produces exactly three 2-path components in $G[U_2' \setminus \{v\}]$ has a degree-2 neighbor u in $G[U_2']$ with $q_u = 1$.*

Proof. (i) Now the degree of every vertex in U_2' is at most 3 in $G[U_2']$ by the assumption on (C, D). We first prove the next claim.

Claim No vertex $v \in U_2'$ in (C, D) produces any bad components other than 2-path components in $G[U_2' \setminus \{v\}]$.

PROOF. Assuming that there is a bi-claw, leg-triangle or tri-claw component H in $G[U_2' \setminus \{v\}]$, we show that v or a vertex in H satisfies one of conditions (c-1) to (c-3) in $G[U_2']$ to prove the claim. Let $k = |N(v) \cap V(H)|$ in $G[U_2']$, where $1 \leq k \leq 3$. We distinguish three cases $k = 1, 2, 3$.

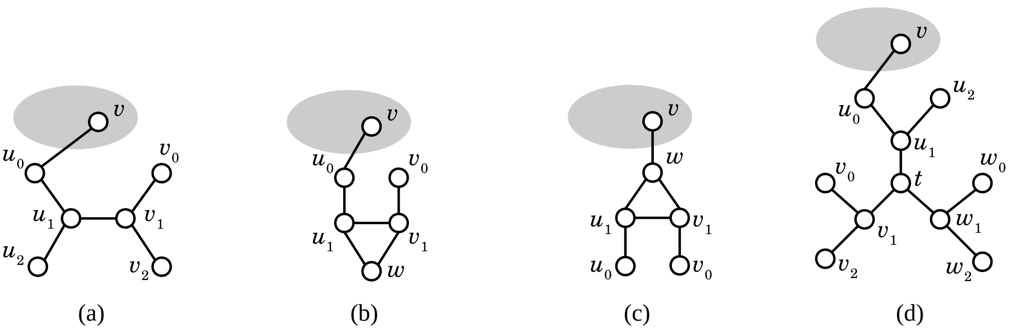


Fig. 2. Components containing v in $G[U_2']$ such that a bi-claw, leg-triangle or tri-claw component H is produced by removing v and $k = |N(v) \cap V(H)| = 1$ in $G[U_2']$

Case 1. $k = 1$: Without loss of generality there are four cases: (a) H is a bi-claw component $(u_0u_1u_2)(v_0v_1v_2)$ and u_0 is adjacent to v; (b) H is a leg-triangle component $u_0(u_1wv_1)v_0$ and u_0 is adjacent to v; (c) H is a leg-triangle component $u_0(u_1wv_1)v_0$ and w is adjacent to v; and (d) H is a tri-claw component $t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$ and u_0 is adjacent to v, where these four cases are illustrated in Fig. 2. If v is a degree-2 vertex and has a degree-1 neighbor in Case (a), (b) or (d), then u_0 is a vertex with $q_{u_0} = 1$ in $G[U_2']$, which satisfies (c-2). Assume that v is not such a vertex. We show that the degree-3 vertex $v_1 \in V(H)$ furthest from v satisfies (c-1) or (c-3).

Cases (a), (b) and (c): The degree-3 vertex v_1 satisfies both of the following two conditions: removing v_1 from $G[U_2']$ produces no bad component; and removing $N[v_1]$ from $G[U_2']$ produces at most one bad component other than a 2-path component. Therefore v_1 satisfies (c-1) or (c-3)(iii).

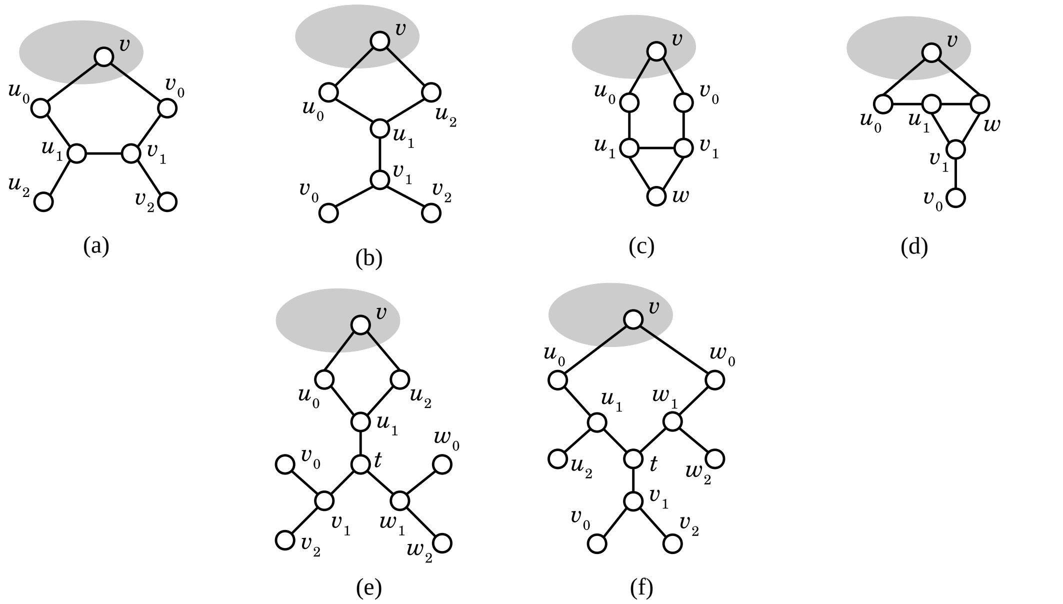


Fig. 3. Components containing v such that a bi-claw, leg-triangle or tri-claw component H is produced by removing v and $k = |N(v) \cap V(H)| = 2$ in $G[U_2']$

Case (d): The degree-3 vertex v_1 satisfies both of the following two conditions: removing v_1 from $G[U_2']$ produces a degree-3 $(0, 0)$-vertex w_1; and removing $N[v_1]$ from $G[U_2']$ produces exactly one 2-path component. Thus v_1 satisfies (c-3)(iv).

Case 2. $k = 2$: Without loss of generality there are six cases: (a) H is a bi-claw component $(u_0u_1u_2)(v_0v_1v_2)$ and $u_0, v_0 \in N(v)$; (b) H is a bi-claw component $(u_0u_1u_2)(v_0v_1v_2)$ and $u_0, u_2 \in N(v)$; (c) H is a leg-triangle component $u_0(u_1wv_1)v_0$ and $u_0, v_0 \in N(v)$; (d) H is a leg-triangle component $u_0(u_1wv_1)v_0$ and $u_0, w \in N(v)$; (e) H is a tri-claw component $t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$ and $u_0, u_2 \in N(v)$; and (f) H is a tri-claw component $t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$ and $u_0, w_0 \in N(v)$, where these six cases are illustrated in Fig. 3. If v has a degree-1 neighbor in $G[U_2']$, then v is a degree-3 $(1, 0)$-vertex such that removing v from $G[U_2']$ produces exactly one bad component, i.e., H, which is not a 2-path component. Hence v satisfies (c-3)(iii). Assume that v is not such a vertex. We show that the degree-3 vertex $v_1 \in V(H)$ furthest from v satisfies (c-1) or (c-3).

Cases (a), (b), (c) and (d): The degree-3 vertex v_1 satisfies both of the following two conditions: removing v_1 from $G[U_2']$ produces no bad component; and removing $N[v_1]$ from $G[U_2']$ produces at most one bad component other than a 2-path component. Therefore v_1 satisfies (c-1) or (c-3)(iii).

Cases (e): v_1 satisfies both of the following two conditions: removing v_1 from $G[U_2']$ produces a degree-3 $(0, 0)$-vertex w_1; and removing $N[v_1]$ from $G[U_2']$ produces exactly one 2-path component. Thus v_1 satisfies (c-3)(iv).

Case (f): v_1 is a degree-3 $(0, 0)$-vertex in $G[U_2']$. Hence v_1 satisfies (c-1).

Case 3. $k = 3$: Now $N(v) \subseteq V(H)$, and there is only one bad component other than a 2-path component in $G[U_2' \setminus \{v\}]$. In the case where H is a leg-triangle or tri-claw component, removing $N[v]$ produces no bad component, and v is a degree-3 $(1, 0)$-vertex, which satisfies (c-3)(iii). In the other case where H is a bi-claw component $(u_0u_1u_2)(v_0v_1v_2)$ and without loss of generality $\{u_0, u_2, v_0\} = N(v)$, we see that u_1 is a degree-3 $(0, 0)$-vertex, which satisfies (c-1).

This prove the claim. □

Next we prove that the set of new bad components in $G[U_2' \setminus \{v\}]$ is a set of three 2-path components. Let $P_1, P_2, \ldots, P_{b_v}$ be the new bad components produced in $G[U_2' \setminus \{v\}]$, all of which are 2-path components. To prove the property (i) of the lemma, we assume that $b_v \in \{1, 2\}$, and prove that some neighbor of v satisfies one of conditions (c-1) to (c-3) in $G[U_2']$. Without loss of generality for the 2-path component $P_1 = v_0v_1v_2$, there are the following five cases: (a) $N(v) \cap V(P_1) = \{v_0\}$; (b) $N(v) \cap V(P_1) = \{v_1\}$; (c) $N(v) \cap V(P_1) = \{v_0, v_1\}$; (d) $N(v) \cap V(P_1) = \{v_0, v_2\}$; and (e) $N(v) \subseteq V(P_1)$, as illustrated in Fig. 4.

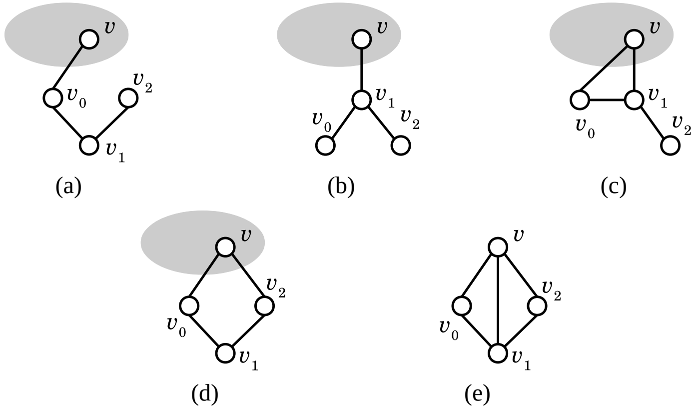


Fig. 4. Components containing v such that a 2-path component $v_0v_1v_2$ is produced by removing v

For Case (d) or (e), there is an admissible 4-cycle $vv_0v_1v_2$ in $G[U_2']$, implying that v satisfies condition (c-3)(i). Assume that neither of Case (d) and (e) holds for P_2 if any.

Next consider Case (a). We see that $G[U_2' \setminus N[v_0]]$ contains $b_v - 1$ (≤ 1) new 2-path components, where $b_v = 1$ if $b_{v_0} \geq 1$; i.e., removing v_0 produces new 2-path components. Hence v_0 is a degree-2 (x, y)-vertex with $x + y \leq 1$ and $q_{v_0} \geq 1$ in $G[U_2']$, satisfying condition (c-2). Assume that Case (a) does not hold for P_2 if any.

Finally consider Case (b) or (c). Let H denote the component containing u in $G[U_2']$. Removing v_1 from $G[U_2']$ produces no 2-path component, since H is not a bi-claw or leg-triangle component. Removing $N[v_1]$ from $G[U_2']$ produces $b_v - 1$ (≤ 1) new 2-path components. Hence if $b_v = 1$, then v_1 is a degree-3 $(0, 0)$-vertex, satisfying condition (c-1). Assume that $b_v = 2$, and denote P_2 by $w_0w_1w_2$, where $w_1 \in N(u)$ and P_2 satisfies configuration (b) or (c). We show that v_1 satisfies condition (c-3)(iv) in $G[U_2']$. Removing $N[v_1]$ from $G[U_2']$ produces only one 2-path component $P_2 = w_0w_1w_2$, and removing v_1 from $G[U_2']$ produces no 2-path component. We see that w_1 is a degree-3 vertex such that $b_{w_0} = b_{N[w_0]} = 0$ in $G[U_2' \setminus \{u\}]$. Hence v_1 is a vertex satisfying condition (c-3)(iv), as required.

(ii) Let u be a degree-2 vertex with $q_u = 1$ in $G[U_2']$. By $q_u = 1$, $G[U_2' \setminus \{u\}]$ contains a clique Q of size 2. The degree-2 vertex $u \in U_2'$ has one neighbor in Q and the other neighbor $v \in U_2' \setminus V(Q)$. Removing v from $G[U_2']$ produces a 2-path component H with $V(H) = \{u\} \cup V(Q)$, we see that removing v from $G[U_2']$ produces a set of three 2-path components by (i), which also indicates that v is of degree 3 in $G[U_2']$.

Conversely let v be a degree-3 vertex removal of which produces exactly three 2-path components in $G[U_2']$. Since there is no tri-claw component in $G[U_2']$, removing v from $G[U_2']$ produces at least one 2-path component $u_0u_1u_2$ such that $u_0 \in N(v)$ in $G[U_2']$. Then u_0 is a degree-2 vertex with $q_{u_0} = 1$ in $G[U_2']$ since removing u_0 produces the clique component consisting of $\{u_1, u_2\}$. □

Lemma 14. *Algorithm* EDSSTAGE2 *branches on an optimal vertex v satisfying condition* (c-4) *in $G[U_2']$ together with possible branchings on the resulting new bad components with a recurrence not worse than* (8).

Proof. Since v is an optimal vertex satisfying condition (c-4), v is a degree-2 vertex with $q_v = 1$ in $G[U_2']$ in an instance (C, D) reduced up to (c-3). Thus removing v from $G[U_2']$ produces exactly two components: the component H' containing u and the clique component Q of size 2. Now Lemma 13 holds for (C, D), and v has a degree-3 neighbor u removal of which produces exactly three 2-path components P_1, P_2 and P_3. We see that the component H containing v is a graph consisting of P_1, P_2 and P_3 and the degree-3 vertex u adjacent to all these 2-path components, one of which say P_3 is given by $vv'v''$ for $\{v', v''\} = V(Q)$. Let w_i, $i = 1, 2$, be the neighbor of u in P_i. In what follows, we show that the algorithm continues to branch on one of w_1 and w_2, say w after fixing v as a covered vertex, and branches on the other of them after fixing w as a covered vertex, and then derive recurrences for branching on v together with branchings on w, w' and all newly produced bad components. Without loss of generality, we distinguish three cases: (a) $d(w_1;H)=d(w_2;H)=3$; (b) $d(w_1;H)=2$ and $d(w_2;H)=3$; and (c) $d(w_1;H)=d(w_2;H)=2$, where these three components are illustrated in Fig. 5.

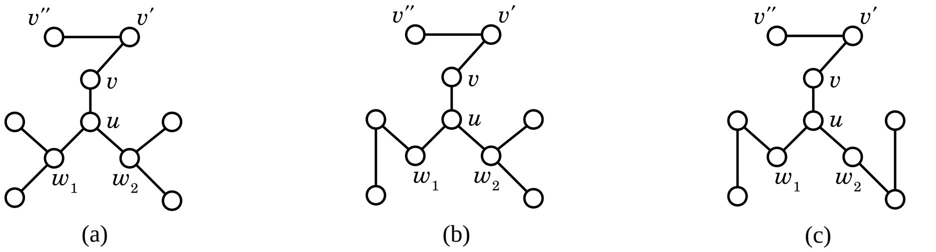


Fig. 5. Components containing a degree-2 vertex v with $q_v=1$ under the assumption in Lemma 13, which contain a degree-3 vertex u adjacent to v such that exactly three new 2-path components are produced by removing u

Case (a). $d(w_1;H)=d(w_2;H)=3$: From the structure of H, we see that w_1 is a degree-3 $(0,1)$-vertex in $G[U_2' \setminus \{v\}]$ such that removing w_1 from $G[U_2' \setminus \{v\}]$ changes w_2 to a degree-3 $(0,0)$-vertex satisfying condition (c-1); and removing $N[w_1]$ from $G[U_2' \setminus \{v\}]$ produces exactly one 2-path component. Hence w_1 satisfies condition (c-3)(iv) in $G[U_2' \setminus \{v\}]$. Since no vertex in H' satisfies any of conditions (c-1), (c-2) and (c-3)(i)-(iii) in $G[U_2' \setminus \{v\}]$, each of w_1 and w_2 is an optimal vertex in $G[U_2' \setminus \{v\}]$. After v is fixed as a covered vertex, the algorithm branches on one of them, say w and continues to branch on the other of them after fixing w as a covered vertex with the recurrence (14). Therefore we have the following recurrence:

$$
\begin{aligned}
T(\mu) &\le T(\mu-2-2)+2T(\mu-2-4)+T(\mu-2-5)\\
&\quad +T(\mu-2-1-1)+2T(\mu-2-1-2)+T(\mu-2-2-2)\\
&=2T(\mu-4)+2T(\mu-5)+3T(\mu-6)+T(\mu-7),
\end{aligned}
$$

which solves to $T(\mu)=O(1.4941^{\mu})$.

Case (b). $d(w_1;H)=2$ and $d(w_2;H)=3$: From the structure of H, we see that w_1 is a degree-2 $(0,1)$-vertex with $q_{w_1}=1$ in $G[U_2' \setminus \{v\}]$ such that removing w_1 from $G[U_2' \setminus \{v\}]$ changes w_2 to a degree-3 $(0,0)$-vertex; and removing $N[w_1]$ from $G[U_2' \setminus \{v\}]$ produces exactly one 2-path component, where w satisfies condition (c-2) in $G[U_2' \setminus \{v\}]$. Since no vertex in H' other than w_1 satisfies any of conditions (c-1) and (c-2) in $G[U_2' \setminus \{v\}]$, w_1 is the unique optimal vertex in $G[U_2' \setminus \{v\}]$. After fixing w_1 as a covered vertex, the algorithm branches on w_2, since w_2 satisfies condition (c-1) in $G[U_2' \setminus \{v,w\}]$. Therefore we have the following recurrence:

$$
\begin{aligned}
T(\mu) &\le T(\mu-2-2-1)+T(\mu-2-2-3)\\
&\quad +T(\mu-2-2-1)+T(\mu-2-2-2)\\
&\quad +T(\mu-2-1-1)+2T(\mu-2-1-2)+T(\mu-2-2-2)\\
&=T(\mu-4)+4T(\mu-5)+2T(\mu-6)+T(\mu-7),
\end{aligned}
$$

which solves to $T(\mu)=O(1.4876^{\mu})$.

Case (c). $d(w_1;H)=d(w_2;H)=2$: From the structure of H, we see that w_1 is a degree-2 $(0,1)$-vertex with $q_{w_1}=1$ in $G[U_2' \setminus \{v\}]$ such that removing w_1 from $G[U_2' \setminus \{v\}]$ changes w_2 to a degree-2 $(0,0)$-vertex with $q_{w_2}=1$; and removing $N[w_1]$ from $G[U_2' \setminus \{v\}]$ produces exactly one 2-path component. Hence w_1 (resp., w_2) satisfies condition (c-2) in $G[U_2' \setminus \{v\}]$ (resp., $G[U_2' \setminus \{v,w_1\}]$). Since no vertex of H' other than w_1 and w_2 satisfies any of conditions (c-1) and (c-2) in $G[U_2' \setminus \{v\}]$, each of w_1 and w_2 is an optimal vertex in $G[U_2' \setminus \{v\}]$. After v is fixed as a covered vertex, the algorithm branches on one of them, say w and continues to branch on the other of them say w' after fixing w as a covered vertex, since there is no vertex satisfying condition (c-1) in $G[U_2' \setminus \{v,w\}]$. Therefore we have the following recurrence:

$$\begin{aligned}
T(\mu) \le\ & T(\mu-2-2-2)+T(\mu-2-2-2) \\
& +T(\mu-2-2-1)+T(\mu-2-2-2) \\
& +T(\mu-2-1-1)+2T(\mu-2-1-2)+T(\mu-2-2-2) \\
=\ & T(\mu-4)+3T(\mu-5)+4T(\mu-6),
\end{aligned}$$

which solves to $T(\mu)=O(1.4833^{\mu})$.

Since all the recurrences obtained in Cases (a)-(c) are not worse than (8), the lemma holds. $\square$

Lemma 15. *Let (C,D) be an instance reduced up to* (c-4)*.*

(i) *For any vertex v in $G[U_2']$ of (C,D), removing v from $G[U_2']$ produces no bad component, and removing $N[v]$ from $G[U_2']$ produces no bad component other than 2-path components.*
(ii) *Every degree-3 vertex v in $G[U_2']$ of (C,D) is a $(0,1)$-vertex or a $(0,2)$-vertex.*
(iii) *For any degree-3 $(0,1)$-vertex v in $G[U_2']$ of (C,D), the component H containing v in $G[U_2']$ contains a 6-cycle such that either*
   (a) *$vu_0u_1u_2u_3u_4$ consisting of v and five degree-2 vertices u_i, $i=0,1,2,3,4$; or*
   (b) *$vv'u_0u_1u_2u_3$ consisting of v, another degree-3 $(0,1)$-vertex v', and four degree-2 vertices u_i, $i=0,1,2,3$.*
(iv) *For any degree-3 $(0,2)$-vertex v in $G[U_2']$ of (C,D), the component H containing v in $G[U_2']$ consists of either*
   (c) *two 6-cycles $vv'u_0u_1u_2u_3$ and $vv'v_0v_1v_2v_3$ that share an edge vv' between v and another degree-3 $(0,2)$-vertex v' and pass through degree-2 vertices u_i and v_i, $i=0,1,2,3$; or*
   (d) *a 4-cycle $vv_0v_1v_2$ of v and three other degree-3 $(0,2)$-vertices v_i, $i=0,1,2$ and two paths $vu_0u_1u_2v_1$ and $v_0w_0w_1w_2v_2$ joining two vertices in the 4-cycle and passing through degree-2 vertices u_i and w_i, $i=0,1,2$.*

The four types (a)-(d) *of components containing v are illustrated in Fig 6.*

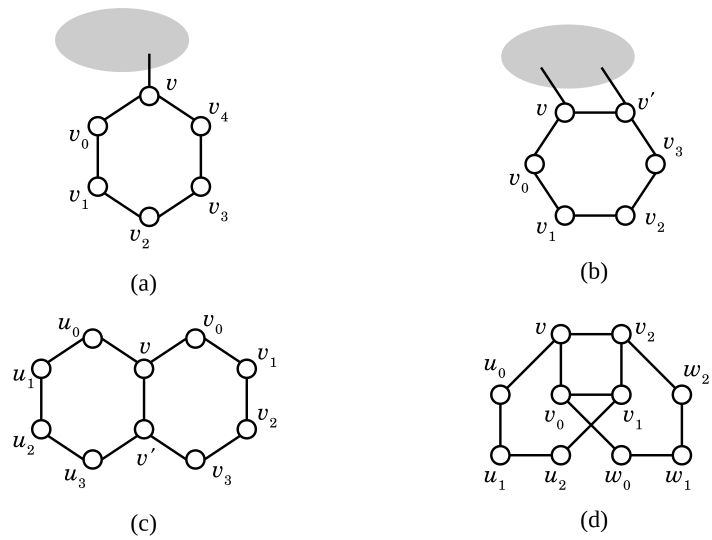


Fig. 6. Components containing a degree-3 vertex v under the assumption in Lemma 15

Proof. Now the degree of every vertex in U_2' is at most 3 in $G[U_2']$ by the assumption on (C,D).

(i) Lemma 13 holds due to the assumption, and there is no degree-2 vertex u with $q_u=1$ in $G[U_2']$. Therefore for any vertex v in $G[U_2']$, removing v from $G[U_2']$ produces no bad component.

To show that removing $N[v]$ produces no bad component other than 2-path components, we prove a slightly more general property as follows, where we can set $(z, S) = (v, N(v))$ to prove (i).

Claim Let $z \in U_2'$ and $S \subseteq U_2' \setminus \{z\}$ be a subset of vertices such that z and each vertex $s \in S$ are connected by a path in $G[U_2' \setminus (S \setminus \{z, s\})]$ of (C, D). Then removing S from $G[U_2']$ produces no bad component other than 2-path components or the component H_z containing z.

PROOF. Assuming that there exists a bi-claw, leg-triangle or tri-claw component H $(\neq H_z)$ in $G[U_2' \setminus S]$, we show that some vertex in H satisfies one of conditions (c-1) and (c-3) in $G[U_2']$ to prove the claim. Since removing z from $G[U_2']$ produces no bad component, at least two vertices in S, say a and b are adjacent to $V(H)$ in $G[U_2']$. Also every bad component H other than 2-path components contains a cut-vertex v^* removal of which leaves a 2-path component P, which we call a 2-*path subgraph* of H. Hence some vertex in S must be adjacent to each 2-path subgraph of H, since otherwise removing the cut-vertex v^* would produce a bad component. Therefore we only need to consider the following four cases:

(1) H is a bi-claw component $(u_0u_1u_2)(v_0v_1v_2)$ such that $u_0 \in N(a)$ and $v_0 \in N(b)$ in $G[U_2']$;
(2) H is a leg-triangle component $u_0(u_1wv_1)v_0$ such that $u_0 \in N(a)$ and $v_0 \in N(b)$ in $G[U_2']$;
(3) H is a leg-triangle component $u_0(u_1wv_1)v_0$ such that $w \in N(a)$ and $v_0 \in N(b)$ in $G[U_2']$; and
(4) H is a tri-claw component $t(u_0u_1u_2)(v_0v_1v_2)(w_0w_1w_2)$ such that u_0, v_0 and w_0 are adjacent to S in $G[U_2']$.

We show that vertex u_1 in cases (1)-(3) and vertex t in case (4) satisfy condition (c-1) or (c-3)(iii). Note that each path that connects two vertices in S and passes through z contains no vertex in H, since H does not contain z in $G[U_2' \setminus S]$. In cases (1)-(3), removing $N[u_1]$ from $G[U_2']$ produces only one nontrivial component H', which cannot be a 2-path component, since H' has a path of length ≥ 3 containing z, a, b and v_0. Therefore u_1 in cases (1)-(4) is a degree-3 $(0, y)$-vertex with $y \leq 1$ such that removing $N[u_1]$ produces no new 2-path component; that is, u_1 satisfies condition (c-1) or (c-3)(iii). In case (4), removing $N[t]$ from $G[U_2']$ produces only one nontrivial component H'' containing $\{u_0, v_0, w_0\} \cup S$, which cannot be a 2-path component, and we see that t satisfies condition (c-1) or (c-3)(iii). This proves the claim.

(ii) Note that v is a $(0, y)$-vertex with $y \geq 0$ by Lemma 13. Since there is no degree-3 $(0, 0)$-vertex in $G[U_2']$, v is a $(0, y)$-vertex with $y \geq 1$. Now removing $N[v]$ from $G[U_2']$ produces no bad components other than 2-path components. For any 2-path component H produced by removing $N[v]$ from $G[U_2']$, at least two neighbors of v are adjacent to $V(H)$; thus there are at least two edges between $N(v)$ and $V(H)$ in $G[U_2']$. Therefore there are at most six edges between $N(v)$ and $U_2' \setminus N[v]$ in $G[U_2']$. Thus removing $N[v]$ can produce at most three 2-path components; and thereby v is a $(0, y)$-vertex with $1 \leq y \leq 3$. Assuming that v is a degree-3 $(0, 3)$-vertex in $G[U_2']$, we show that there is a vertex satisfying condition (c-1) or (c-3)(iii) in $G[U_2']$. Let a, b and c denote the three neighbors of v in $G[U_2']$. Let P_1, P_2 and P_3 be the three 2-path components produced by removing $N[v]$ from $G[U_2']$. Without loss of generality, we assume that a and b are adjacent to $V(P_1)$, both b and c are adjacent to $V(P_2)$ and both c and a are adjacent to $V(P_3)$ in $G[U_2']$. Then $G[U_2']$ has a path that contains $\{b, c\}$ and some vertex in P_2 but does not contain v. Therefore removing $N[a]$ from $G[U_2']$ produces only one component H' containing $\{b, c\} \cup V(P_2)$ other than clique components of size ≤ 2, where H' cannot be a 2-path component. Thus a is a degree-3 $(0, 0)$- or $(0, 1)$-vertex in $G[U_2']$, which satisfies condition (c-1) or (c-3)(iii). Consequently, every degree-3 vertex v in $G[U_2']$ is a $(0, y)$-vertex with $1 \leq y \leq 2$. This proves (ii).

(iii) Let v be a degree-3 $(0, 1)$-vertex in $G[U_2']$. In what follows, we show that the component H containing v in $G[U_2']$ satisfies condition (a) or (b) of the lemma. Let a, b and c denote the neighbors of v in $G[U_2']$, and $P = u_0u_1u_2$ be the 2-path component produced by removing $N[v]$ from $G[U_2']$. Note that at least two vertices in $N(v) = \{a, b, c\}$ are adjacent to P since otherwise removing the unique vertex in $N(v)$ adjacent to P would produce a bad component, contradicting (i). We distinguish two cases: $N(u_1) \cap N(v) \neq \emptyset$; and $N(u_1) \cap N(v) = \emptyset$.

Case 1. u_1 is adjacent to a vertex in $N(v)$: Without loss of generality, let $u_0 \in N(a)$ and $u_1 \in N(b)$, where $u_0 \notin N(c)$ and $u_2 \notin N(a)$ since otherwise u_0cva or $u_2au_0u_1$ would be an admissible 4-cycle. By (ii), degree-3 vertex u_1 is a $(0, 1)$- or $(0, 2)$-vertex such that removing $N[u_1]$ produces at least one 2-path component, where avc must be one of such 2-path components, where vertices a and c are not adjacent. This indicates that the component H containing v in $G[U_2']$ consists of the seven vertices, v, a, b, c, u_0, u_1 and u_2. If vertex b is of degree 3 in H, then removing $N[b]$ from H produces no 2-path component because $u_0, a \notin N(c)$ and $u_2 \notin N(a)$, contradicting (ii). Hence b is a degree-2 vertex, where we see that b is a $(0, 0)$-vertex with $q_{N[b]} \geq 1$ in $G[U_2']$ satisfying (c-3)(iii). This contradicts the assumption on (C, D), and Case 1 cannot occur.

Case 2. u_1 is not adjacent to any vertex in $N(v)$ in $G[U_2']$: If u_0 is not adjacent to $N(v)$, then u_2 has two neighbors in $N(v)$, which must be a degree-3 $(0, y)$-vertex with $q_{u_2} = 1$, where $y \leq 1$ since v is a $(0, 1)$-vertex. This would imply that u_2 satisfies condition (c-3)(ii). Hence u_0 is adjacent to $N(v)$. Analogously u_2 is also adjacent to $N(v)$. Without loss of generality, let $u_0 \in N(a)$ and $u_2 \in N(b)$. We let a' (resp., b') denote the third neighbor of a (resp., b) if any.

We show that if $c \in N(u_0)$ or $c \in N(u_2)$ in $G[U_2']$, then H contains a vertex satisfying condition (c-3)(i)-(ii). Without loss of generality we assume that $c \in N(u_0)$. Since removing $N[v]$ from $G[U_2']$ produces no bad component other than 2-path component $u_0u_1u_2$, removing $\{a, c\}$ produces no bad component. Since $G[U_2']$ contains a path which starts from v, passes through b, u_2 and u_1 and ends at u_0, removing $\{v, u_0\}$ from $G[U_2']$ produces no bad component other than 2-path components by Claim with $(z, S) = (b, \{v, u_0\})$. If $b_{\{v,u_0\}} \leq 1$, then 4-cycle vau_0c is admissible in $G[U_2']$, and every vertex on the cycle satisfies (c-3)(i). Let $b_{\{v,u_0\}} \geq 2$. Then removing $\{v, u_0\}$ from $G[U_2']$ produces a 2-path component P' other than 2-path component u_1u_2b. If P' contains only one of a and c, then removing $N[v]$ from $G[U_2']$ produces a clique component of size 2 consisting of $V(P') \setminus \{a\}$ or $V(P') \setminus \{c\}$, indicating that v satisfies (c-3)(ii). Let P' contain both of a and c; i.e., $P' = aa'c$. Since $b_{\{v,a'\}} = b_{\{a,c\}} = 0$, 4-cycle $vaa'c$ is admissible in $G[U_2']$, and every vertex on the cycle satisfies (c-3)(i). In the following we assume that $c \notin N(u_0) \cup N(u_2)$, where we observe that no degree-2 vertex is adjacent to two neighbors of the degree-3 $(0, 1)$-vertex v.

Since $a \in N(b)$ in $G[U_2']$ implies that a is a degree-3 $(0, 0)$-vertex satisfying (c-1), we have $a \notin N(b)$ in (C, D).

If $a, b \in N(c)$, then $vacb$ would be an admissible 4-cycle in $G[U_2']$ and any vertex on it would satisfy (c-3)(i). If $a \in N(c)$ and $d(b; H) = 2$, then we see that $b_{N[a]} = 1$ by $b_{N[v]} = 1$ and that $vau_0u_1u_2b$ is a 6-cycle satisfying condition (b) for H. If $a \in N(c)$, $b \notin N(c)$ and $d(b; H) = 3$, then b would be a degree-3 $(0, 0)$-vertex in $G[U_2']$ satisfying (c-1). Hence we assume that $a, b \notin N(c)$ in the following.

We here show that $a \notin N(u_2)$. Let $a \in N(u_2)$. Then a is a degree-3 $(0, y)$-vertex, where $y = 2$, since if a is a degree-3 $(0, 1)$-vertex then there cannot exist a degree-2 vertex u_1 adjacent to two neighbors of a. In this case, the graph $G[U_2' \setminus N[a]]$ has two new 2-path components, P_b containing b and P_c containing c, where P_c is not adjacent to any vertex in $\{a, b, u_0, u_1, u_2\}$ since $c \notin N(u_0) \cup N(u_2)$, contradicting that P_c will not be produced by removing v. Therefore we have $a \notin N(u_2)$, $b \notin N(u_0)$ and $d(u_0; H) = d(u_1; H) = d(u_2; H) = 2$.

Finally we show that if $d(a; H) = 3$ then removing $N[a]$ from $G[U_2']$ produces no 2-path component that does not contain vertex b. Assume that a 2-path component P'' not containing b is produced in $G[U_2' \setminus N[a]]$. Since removing a' from $G[U_2']$ produces no bad component, both a' and v are adjacent to $V(P'')$ in $G[U_2']$, and $V(P'')$ consists of vertex c and some vertices $e, f \in U_2' \setminus (N[v] \cup \{a', u_0, u_1, u_2\})$. Since v is a degree-3 $(0, 1)$-vertex in $G[U_2']$ from the assumption, there is no 2-path component consisting of $\{a', e, f\}$ in $G[U_2' \setminus N[v]]$. Hence removing $N[v]$ from $G[U_2']$ produces a clique component of size 2 consisting of two of $\{a', e, f\}$. Then v would be a degree-3 $(0, 1)$-vertex with $q_{N[v]} = 1$ in $G[U_2']$ satisfying condition (c-3)(ii), a contradiction. This proves that if $d(a; H) = 3$ (resp., $d(b; H) = 3$) then removing $N[a]$ (resp., $N[b]$) from $G[U_2']$ produces no 2-path component that does not contain vertex b (resp., a).

When $d(a; H) = d(b; H) = 2$, there is a 6-cycle which starts from v, passes through five degree-2 vertices a, u_0, u_1, u_2 and b in this order and ends at v, indicating that the the component H containing v satisfies condition (a).

Let $d(a; H) \neq d(b; H)$, say $d(a; H) = 3$ and $d(b; H) = 2$. Then removing $N[a]$ from $G[U_2']$ produces no 2-path component that does not contain vertex b; i.e., it produces only one 2-path component bu_2u_1, and thereby a is a degree-3 $(0, 1)$-vertex in $G[U_2']$. Since there is a 6-cycle which starts from v, passes through four degree-2 vertices b, u_2, u_1, u_0 and a in this order and ends at v, the component H containing v in $G[U_2']$ satisfies condition (b).

Let $d(a; H) = d(b; H) = 3$. Analogously with the case of $d(a; H) = 3$ and $d(b; H) = 2$, we see that each of a and b is a degree-3 $(0, 1)$-vertex in $G[U_2']$. Recall that degree-3 $(0, 1)$-vertex v has two degree-3 $(0, 1)$-neighbors joined by a path P_v passing through three degree-2 vertices. By applying this to degree-3 $(0, 1)$-vertex a, we see that $G[U_2']$ contains a path $P_a = aa's_0s_1s_2v$ passing through degree-2 vertices s_i, $i = 0, 1, 2$. Similarly there is a path $P_b = bb't_0t_1t_2v$ passing through degree-2 vertices t_i, $i = 0, 1, 2$. Since $s_2 = t_2$ must hold, such two paths cannot exist unless $a' = b'$. However, when $a' = b'$, we see that v is a $(0, 2)$-vertex, a contradiction.

Consequently, the component H containing v in $G[U_2']$ satisfies one of two conditions (a) and (b) of the lemma.

(iv) Let v be a degree-3 $(0,2)$-vertex in $G[U_2']$, a, b and c denote the three neighbors of v in $G[U_2']$, and H be the component containing $N[v]$ in $G[U_2']$. Let $P_1 = u_0u_1u_2$ and $P_2 = w_0w_1w_2$ be the two 2-path components produced by removing $N[v]$ from $G[U_2']$. In what follows, we show that there is a vertex satisfying condition (c-1) or (c-3) in $G[U_2']$ unless H is a graph that satisfies condition (c) or (d) of the lemma.

For each P_i, at least two neighbors of v are adjacent to $V(P_i)$. Hence at least one neighbor of v, say b is adjacent to both P_1 and P_2.

If u_1 is adjacent to a vertex in $N(v)$, then it is a degree-3 $(0,0)$-vertex, since removing u_0, u_1, u_2 and exactly one vertex in $N(v)$ produces no 2-path component; that is, u_1 satisfies condition (c-1). We assume that neither of u_1 and w_1 is adjacent to any vertex in $N(v)$. Let $\{u_2, w_2\} \subseteq N(b)$ without loss of generality.

If u_0 is not adjacent to any vertex in $N(v)$, then the degree-3 vertex b is a $(0,y)$-vertex with $y \leq 1$ and $q_{N[b]} \geq 1$, which satisfies condition (c-1) or (c-3)(ii). We further assume that each of u_0, u_2, w_0 and w_2 is adjacent to a vertex in $N(v)$.

If vertex a (resp., c) is not adjacent to $u_0u_1u_2$ or $w_0w_1w_2$ in $G[U_2']$, then another neighbor c (resp., a) of v is a degree-3 $(0,0)$-vertex in $G[U_2']$, which satisfies (c-1).

If b is a degree-3 $(0,1)$-vertex in $G[U_2']$, then by (iii) H must have a 6-cycle containing b and at most one more degree-3 vertex that is not the degree-3 $(0,2)$-vertex v. Since such a 6-cycle does not exists in H, b is a degree-3 $(0,2)$-vertex in $G[U_2']$, and hence removing $N[b]$ from $G[U_2']$ produces two 2-path components, which must be au_0u_1 and cw_0w_1 (or cu_0u_1 and aw_0w_1).

In the following we assume that $N(u_0) = \{a, u_1\}$, $N(w_0) = \{c, w_1\}$ and $a \notin N(c)$ without loss of generality.

Case 1. Both a and c are degree-2 vertices in $G[U_2']$: In this case, H satisfies condition (c) of the lemma.

Case 2. One of a and c, say a is a degree-3 vertex in $G[U_2']$: If $a \notin N(w_2)$ or $u_2 \in N(c)$, then removing $N[a]$ from $G[U_2']$ produces no 2-path component. Therefore we have $a \in N(w_2)$ and $u_2 \notin N(c)$. Symmetrically if c is a degree-3 vertex in $G[U_2']$, then $c \in N(u_2)$ and $w_2 \notin N(a)$. This means that exactly one of a and c can be a degree-3 vertex in $G[U_2']$, and H satisfies condition (d) of the lemma. □

Lemma 16. *Algorithm* EDSSTAGE2 *branches on an optimal vertex v satisfying condition* (c-5) *in $G[U_2']$ together with possible branchings on the resulting new bad components with a recurrence not worse than* (8).

Proof. Since v is an optimal vertex satisfying condition (c-5), v is a degree-3 vertex in $G[U_2']$. Let H be the component containing v in $G[U_2']$. There are no vertices satisfying any of conditions (c-1) to (c-4) in $G[U_2']$; therefore Lemma 15 holds, indicating that H satisfies one of the four conditions (a) to (d) in the lemma.

In what follows, we first show that after removing v, a vertex w satisfying condition (c-2) will become an optimal vertex, and then derive recurrences for branching on v together with branchings on the optimal vertex w and all newly produced bad components. Note that after removing v, there is no vertex satisfying condition (c-1) in $G[U_2' \setminus \{v\}]$, since v does not satisfy condition (c-3)(iv) in $G[U_2']$. We distinguish two cases: condition (a) or (b) in Lemma 15 holds; and condition (c) or (d) in Lemma 15 holds

Case (a) or (b): Now v is a degree-3 $(0,1)$-vertex in $G[U_2']$.

We first consider case (a); i.e., H contains a cycle of length 6 which starts from v, passes through five degree-2 vertices v_0, v_1, v_2, v_3 and v_4 in this order and ends at v. Then v_2 will be a degree-2 vertex that satisfies condition (c-2) in $G[U_2' \setminus \{v\}]$, since removing v_2 from $G[U_2' \setminus \{v\}]$ produces exactly two clique components of size 2: one consisting of $\{v_0, v_1\}$ and the other consisting of $\{v_3, v_4\}$. Hence v_2 will be an optimal vertex w in $G[U_2' \setminus \{v\}]$.

We next consider case (b); i.e., H contains a cycle which starts from v, passes through four degree-2 vertices v_0, v_1, v_2, v_3 and a degree-3 vertex v' in this order and ends at v. Then v_2 will be a degree-2 vertex that satisfies condition (c-2) in $G[U_2' \setminus \{v\}]$, since removing v_2 from $G[U_2' \setminus \{v\}]$ produces exactly two components: a clique component of size 2 consisting of $\{v_0, v_1\}$ and the component containing $\{v_3, v'\}$. Hence v_2 will be an optimal vertex w in $G[U_2' \setminus \{v\}]$. To derive a recurrence, we show that removing each of v_2 and $N[v_2]$ from $G[U_2' \setminus \{v_2\}]$ produces no bad component other than a 2-path component.

Removing v_2 from $G[U_2' \setminus \{v\}]$ produces no bad component other than a 2-path component, since v' is a degree-2 vertex in $G[U_2' \setminus \{v, v_2\}]$. Let u denote the other neighbor of v' in $G[U_2' \setminus \{v\}]$. In the case where u is of degree ≤ 2 in $G[U_2' \setminus \{v\}]$, removing $N[v_2]$ produces no a bad component other than a 2-path component. In the case where u is of degree 3 in $G[U_2' \setminus \{v\}]$, the component containing u produced by removing $N[v_2]$ is not a bad component, since u must satisfy one of conditions (a) to (d) in Lemma 15.

As a result, the optimal vertex w in $G[U_2' \setminus \{v\}]$ satisfies condition (c-2); that is, w is a degree-2 (x, y)-vertex with $x + y \leq 1$ and $q_w \geq 1$, and removing each of w and $N[w]$ from $G[U_2' \setminus \{v\}]$ produces no bad component other than a 2-path component. In the case where $x + y = 0$, we have the following recurrence:

$$\begin{aligned}T(\mu) &\leq T(\mu - 1 - 2) + T(\mu - 1 - 2) + T(\mu - 3 - 1) + T(\mu - 3 - 2)\\ &= 2T(\mu - 3) + T(\mu - 4) + T(\mu - 5),\end{aligned}$$

which solves to $T(\mu) = O(1.4656^{\mu})$. In the case where $x + y = 1$, we have the following recurrence:

$$\begin{aligned}T(\mu) &\leq T(\mu - 1 - 2 - 1) + T(\mu - 1 - 2 - 2) + T(\mu - 1 - 2)\\ &\quad + T(\mu - 3 - 1) + T(\mu - 3 - 2)\\ &= T(\mu - 3) + 2T(\mu - 4) + 2T(\mu - 5),\end{aligned}$$

which solves to $T(\mu) = O(1.4826^{\mu})$.

Case (c) or (d): Now v is a degree-3 $(0, 2)$-vertex in $G[U_2']$.

We first consider case (c); i.e., H consists of the following two paths between v and a degree-3 $(0, 2)$-vertex v': a path which starts form v, passes through degree-2 vertices u_0, u_1, u_2 and u_3 in this order and ends at v'; and a path which starts form v, passes through degree-2 vertices v_0, v_1, v_2 and v_3 in this order and ends at v'. Recall that after removing v from $G[U_2']$, no vertex in H satisfies condition (c-1). Removing v from H leaves only a path which starts from u_0, passes through degree-2 vertices $u_1, u_2, u_3, v', v_3, v_2$ and v_1 in this order and ends at v_0. We see that any vertex $w \in \{u_2, v_2\}$ is a degree-2 $(0, 0)$-vertex with $q_w = 1$ in $G[U_2' \setminus \{v\}]$, and becomes an optimal vertex satisfying condition (c-2).

We next consider case (d); i.e., H consists of a 4-cycle $vv_0v_1v_2$ of four degree-3 $(0, 2)$-vertices and the following two paths joining two diagonal vertices in the 4-cycle: a path which starts from v, passes through degree-2 vertices u_0, u_1 and u_2 and ends at v_1; and a path which starts from v_0, passes through degree-2 vertices w_0, w_1 and w_2 and ends at v_2. After removing v from $G[U_2']$, only one vertex v_1 becomes a degree-3 vertex in $G[U_2' \setminus \{v\}]$, which does not satisfy condition (c-1), as already observed. Here removing u_2 from $G[U_2' \setminus \{v\}]$ produces exactly two components: a clique component of size 2 consisting of $\{u_0, u_1\}$ and the component containing v_1, which is not a bad component. Removing $N[u_2]$ from $G[U_2' \setminus \{v\}]$ also produces exactly two components: an isolated vertex u_0 and the component containing $\{v_0, v_2\}$, which is not a bad component. Hence u_2 is a degree-2 $(0, 0)$-vertex with $q_{u_2} = 1$ in $G[U_2' \setminus \{v\}]$, and is an optimal vertex w satisfying condition (c-2).

As a result, any optimal vertex w in $G[U_2' \setminus \{v\}]$ is a degree-2 $(0, 0)$-vertex satisfying condition (c-2); that is, w is a degree-2 $(0, 0)$-vertex with $q_w = 1$. Thus we have the following recurrence:

$$\begin{aligned}T(\mu) &\leq T(\mu - 1 - 2) + T(\mu - 1 - 2)\\ &\quad + T(\mu - 3 - 1 - 1) + 2T(\mu - 3 - 1 - 2) + T(\mu - 3 - 2 - 2)\\ &= 2T(\mu - 3) + T(\mu - 5) + 2T(\mu - 6) + T(\mu - 7),\end{aligned}$$

which solves to $T(\mu) = O(1.4845^{\mu})$.

Since all the recurrences obtained in Cases (a) to (d) are not worse than (8), the lemma holds. □

A component in $G[U_2']$ is called a *cycle component* if it consists of a single cycle. The following lemma shown in [9] is used to analyze the case where Algorithm EDSSTAGE2 branches on an optimal vertex satisfying condition (c-6).

Lemma 17. [9] *Let L be a cycle component of length ≥ 4 in $G[U_2']$. Algorithm* EDSSTAGE2 *branches on vertices of L with a recurrence not worse than* (8) *until U_2' has no vertices in L.*

Lemma 18. *Algorithm* EDSSTAGE2 *branches on an optimal vertex v satisfying condition* (c-6) *in $G[U_2']$ together with possible branchings on the resulting new bad components with a recurrence not worse than* (8).

Proof. Since v is an optimal vertex satisfying condition (c-6), v is a degree-2 vertex in $G[U_2']$. Let H be the component containing v in $G[U_2']$. In the following, we show that H is a cycle component of length ≥ 4.

Since there is no vertex that satisfies condition (c-5), there are only vertices of degree ≤ 2 in $G[U_2']$. Furthermore there is no vertex of degree ≤ 1 in $G[U_2']$, since $G[U_2']$ has no clique component, no 2-path component and no degree-2 vertex u with $q_u \geq 1$, which satisfies condition (c-2). Therefore there are only degree-2 vertices in $G[U_2']$, indicating that the component containing v in $G[U_2']$ is a cycle component of length ≥ 4.

Algorithm EDSSTAGE2 branches on vertices of H until $G[U_2']$ has no more vertices of H, with a recurrence not worse than (8), by Lemma 17. □

Now we are ready to complete the proof of Lemma 6. Lemmata 10, 11, 12, 14, 16 and 18 guarantee that Algorithm EDSSTAGE2 branches on an admissible 4-cycle or an optimal vertex in $G[U_2']$ together with possible branchings on the resulting new bad components with a recurrence not worse than (8). □