View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CiteSeerX

An Exact Algorithm for Maximum Independent Set in Degree-5 Graphs

Mingyu Xiao
School of Computer Science and Engineering, University of Electronic Science and Technology of China, China,
myxiao@gmail.com

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

Abstract

The maximum independent set problem is a basic NP-hard problem and has been extensively studied in exact algorithms. The maximum independent set problems in low-degree graphs are also important and may be bottlenecks of the problem in general graphs. In this paper, we present an $O^*(1.1737^n)$-time exact algorithm for the maximum independent set problem in an n-vertex graph with degree bounded by 5, improving the previous running time bound of $O^*(1.1895^n)$. In our algorithm, we introduce an effective divide-and-conquer procedure to deal with vertex cuts of size at most two in graphs, and design branching rules on some special structure of triconnected graphs of maximum degree 5. These result in an improved algorithm without introducing a large number of branching rules.

Key words. Exact Algorithm, Independent Set, Measure and Conquer

1 Introduction

In the line of research on worst-case analysis of exact algorithms for NP-hard problems, the *maximum independent set* problem (MIS) is one of the most important problems. It asks us to find a maximum set of vertices in a graph such that no pair of vertices in the set are adjacent to each other. The method of trivially checking all possible vertex subsets results in an $O^*(2^n)$-time algorithm. In the last half a century, great progresses have been made on exact exponential algorithms and their worst-case analysis for MIS. In 1977, Tarjan and Trojanowski [13] published the first nontrivial $O^*(2^{n/3})$-time algorithm. After this, many fast exact algorithms for MIS have been investigated. We quote the $O^*(1.2346^n)$-time algorithm by Jian in 1986 [9], the $O^*(1.2278^n)$-time polynomial-space and $O^*(1.2109^n)$-time exponential-space algorithms by Robson in 1986 [12], the $O^*(1.2210^n)$-time algorithm by Fomin *et al.* in 2006 [5], the $O^*(1.2132^n)$-time algorithm by Kneis *et al.* in 2009 [10] and the $O^*(1.2114^n)$-time algorithm by Bourgeois *et al.* in 2012 [2].

Most polynomial-space algorithms for MIS use the following simple idea to search a solution: branch on a vertex of maximum degree by either excluding it from the solution set or including it to the solution set. In the first branch we will delete the vertex from the graph and in the second branch we will delete the vertex together with all its neighbors from the graph. When the vertex to be branched on is of degree at lest 8, the simple branch is almost good enough to get the running time bound of all published polynomial-space algorithms for MIS. Then MIS in graphs with degree bounded by i ($i \in \{3, 4, 5, 6, 7\}$) may be the bottleneck cases of MIS. For most cases the running time bound for MIS-i (the maximum independent set problem in graphs with maximum degree i) is one of the bottlenecks to improve the running time bound for MIS-$(i+1)$, especially for small i. This holds in the many algorithms for MIS [2, 3, 5, 10, 18]. We look at the most recent two algorithms for MIS in general graphs. Kneis *et al.* [10] used a fast algorithm for MIS-3 by Razgon [11] and used a computer-aided method to check a huge number of cases for MIS-4, and then these two special

[1]Technical report 2013-003, April 11, 2013

case will not be the bottleneck cases in their algorithm for MIS in general graphs. In Bourgeois *et al.*'s paper [2], more than half is discussing algorithms for MIS-3 and MIS-4, and based on improved running time bounds for MIS-3 and MIS-4 they can improve the running time bounds for MIS-5, MIS-6 and then MIS in general graphs. We can see that MIS in low-degree graphs are important. In the literature, we can find a long list of contributions to fast exact algorithms for MIS in low-degree graphs [17, 1, 16, 11, 15, 8, 6, 18]. Currently, MIS-3 can be solved in $O^*(1.0836^n)$ time [17], MIS-4 can be solved in $O^*(1.1376^n)$ time [18], MIS-5 can be solved in $O^*(1.1895^n)$ time and MIS-6 can be solved in $O^*(1.2050^n)$ time [2]. In this paper, we will design an $O^*(1.1737^n)$-time algorithm for MIS-5, improving all previous running time bounds for this problem.

Our algorithm is mainly based on a branch-and-reduce paradigm, in each step of which we will branch on the current graph G with measure w into l subgraphs $G_{(i)}$ with measure $w_{(i)}$ ($i = 1, 2, \ldots, l$). Let $C(w)$ denote the worst-case size of the search tree in our algorithm when the measure of the graph is w. Then we get the recurrence relation $C(w) \leq \sum_{i=1}^{l} C(w - t_{(i)})$, where $t_{(i)} = w - w_{(i)}$ is the decrease of the measure in the i-th subinstance. Let $\tau(t_{(1)}, t_{(2)}, \ldots, t_{(l)})$ denote the unique positive real root of the function $f(x) = 1 - \sum_{i=1}^{l} x^{-t_{(i)}}$. Then $\tau(t_{(1)}, t_{(2)}, \ldots, t_{(l)})$ is called the *branching factor* of the recurrence. Let τ be the maximum branching factor among all branching factors in the search tree. Then the size $C(w)$ of the search tree is at most τ^w. When designing the exact algorithm, we need to make the worst branches in the algorithm as good as possible to improve the running time bound. More details about the analysis of the size of the search tree can be found in the monograph on exact algorithms [7].

To avoid some bad branches in the algorithm, we may need to reduce some special local structures of the graph. First, we apply our *reduction rules* to find a part of the solution when the graph has certain structures. Second, we design effective divide-and-conquer algorithms based on small cuts of the graph. By reducing the local structures in the above two steps, we can apply our *branching rules* on the graph to search a solution. In our algorithm, the divide-and-conquer methods are newly introduced and they can effectively reduce some bottleneck cases, and we design effective branching rules based on careful check on the structures of the graph and analysis of their properties. These are crucial techniques used in the paper to get the significant improvement on this problem.

2 Notation System

Let $G = (V, E)$ stand for a simple undirected graph with a set V of vertices and a set E of edges. Let $n = |V|$. We will use n_i to denote the number vertices of degree i in G, and $\alpha(G)$ to denote the size of a maximum independent set of G. The vertex set and edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. For simplicity, we may denote a singleton set $\{v\}$ by v and union $X \cup \{v\}$ of a subset X and an element by $X + v$.

For a subset $X \subseteq V$, let $\overline{X}$ denote the complement set $V \setminus X$, $N(X)$ denote the set of all vertices in $\overline{X}$ that are adjacent to a vertex in X, and $N[X] = X \cup N(X)$. Let $\delta(v) = |N(v)|$ denote the degree of a vertex v. For a subset $X \subseteq V$, $\delta(X)$ denote the sum of degree of vertices in X and $\delta_{\geq 3}(X)$ denote the sum of degree of vertices of degree ≥ 3 in X. We use $N_2(v)$ to denote the set of vertices with distance exactly 2 from v, and let $N_2[v] = N_2(v) \cup N[v]$. Let $G - X$ be the graph obtained from G by removing the vertices in X together with any edges incident to a vertex in X, $G[X] = G - (V \setminus X)$ be the graph induced from G by the vertices in X, and G/X denote the graph obtained from G by contracting X into a single vertex (removing self-loops and parallel edges). For a vertex v, let f_v denote the number of edges between $N[v]$ and $N_2(v)$. In a graph with maximum degree 5, we define the *gain* g_v of v to be

$$g_v = \sum_{t \in N_2(v)} (5 - \delta(t)) + (f_v - |N_2(v)|),$$

where the second term means the number of times two edges leaving $N[v]$ meet at a same vertex in $N_2(v)$. We denote $(f_v, g_v) \geq (a, b)$ when $f_v \geq a$ and $g_v \geq b$ hold.

A partition (V_1, Z, V_2) of the vertex set $V(G)$ of a graph G is called a *separation* if $V(G)$ is a disjoint union of nonempty subsets V_1, Z and V_2 and there is no edge between V_1 and V_2, where Z is called a *vertex cut*. In this paper, a vertex cut is always assumed to be a minimal vertex cut, i.e., no proper subset of a vertex cut is still a vertex cut. The *line graph* of a graph G is the graph whose vertices correspond to the edges of G, and two vertices are adjacent iff the corresponding two edges share a same endpoint in G. Throughout the paper we use a modified O notation that suppresses all polynomially bounded factors. For two functions f and g, we write $f(n) = O^*(g(n))$ if $f(n) = g(n)\text{poly}(n)$ holds for a polynomial $\text{poly}(n)$ in n.

3 Reduction Rules

First of all, we introduce the reduction rules, which can be applied in polynomial time and reduce the graph by finding a part of the solution. There are many reduction rules for MIS and the related *vertex cover* problem [3], from the simplest ones to deal with degree-1 and degree-2 vertices to the somewhat complicated unconfined vertices and crown reductions. They can be found in almost all exact algorithms and most of approximation and heuristic algorithms for MIS. We introduce some reduction rules that will be used our algorithm.

Reduction by eliminating easy instances

For a disconnected graph G with a component H, we see that $\alpha(G) = \alpha(H) + \alpha(G - V(H))$, and solve instances H and $G - V(H)$ independently. We will solve two kinds of components H directly:
(1) H has at most $\rho = 28$ vertices; and
(2) H is the line graph of a bipartite graph H' between the set of degree-3 vertices and the set of degree-4 vertices, which are call a $(3, 4)$-*bipartite graph*.

Case (1) can be solved in constant time since the size of the graph is constant. Case (2) is based on the following observation: if graph G is the line graph of a graph G', then we obtain a maximum independent set of G directly by finding a maximum matching M in G' and taking the corresponding vertex set V_M in G as a solution. There are several methods to check whether a graph is a line graph or not [14]. To identify a line graph $L(G')$ of a $(3, 4)$-bipartite graph G', we need to check if $L(H)$ is a union of 3-cliques and 4-cliques such that each vertex is a common vertex of a 3-clique and a 4-clique.

Reduction by removing unconfined vertices
A vertex v in an instance G is called *removable* if $\alpha(G) = \alpha(G - v)$. A sufficient condition for a vertex to be removable has been studied in [17]. In this paper, we only use a simple case of the condition. For a vertex v and its neighbor $u \in N(v)$, a vertex $s \in V \setminus V[v]$ adjacent to u is called an *out-neighbor* of u. A neighbor $u \in N(v)$ is called an *extending child* of v if u has exactly one out-neighbor $s_u \in V \setminus N[v]$, where s_u is also called an *extending grandchild* of v. Note that an extending grandchild s_u of v may be adjacent to some other neighbor $u' \in N(v) \setminus \{u\}$ of v. Let $N^*(v)$ denote the set of all extending children $u \in N(v)$ of v, and I_v be the set of all extending grandchildren s_u ($u \in N^*(v)$) of v together with v itself. We call v *unconfined* if there is a neighbor $u \in N(v)$ which has no out-neighbor or $I_v \setminus \{v\}$ is not an independent set (i.e., some two vertices in $I_v \cap N_2(v)$ are adjacent). It is known in [17] that any unconfined vertex is removable.

Lemma 1 [17] *For an unconfined vertex v in graph G, it holds that*

$$\alpha(G) = \alpha(G - v).$$

A vertex u *dominates* another vertex v if $N[u] \subseteq N[v]$, where v is called *dominated.* We see that dominated vertices are unconfined vertices.

Reduction by folding twins
The set $\{v_1, v_2\}$ of two nonadjacent degree-3 vertices is called a *twin* if $N(v_1) = N(v_2)$.

Lemma 2 [17] *For a twin $A = \{v_1, v_2\}$, we have that*

$$\alpha(G) = \alpha(G^\star) + 2,$$

where $G^\star = G/N[A]$ if $N(A)$ is an independent set and $G^\star = G - N[A]$ otherwise.

Folding a twin $A = \{v_1, v_2\}$ is to remove or contract $N[A]$ in the above way. See Fig. 3 (a) and (b) in Appendix for an illustration.

Reduction by folding short funnels
A degree-3 vertex v together with its neighbors $N(v) = \{a, b, c\}$ is called a *funnel* if $N[v]\backslash\{a\}$ induces a triangle for some $a \in N(v)$, and the funnel is denoted by a-v-$\{b, c\}$. Note that v dominates any vertex in $N(a) \cap N(v)$ if $N(a) \cap N(v)$ is not empty. When we assume that there are no dominated vertices anymore, then $N(a) \cap N(v) = \emptyset$.

Folding a funnel a-v-$\{b, c\}$ means that we add an edge between every non-adjacent pair (x, y) of vertices $x \in N(a) \setminus \{v\}$ and $y \in \{b, c\}$ and then remove vertices a and v.

Fig. 3(c) in Appendix illustrates the operation of folding a funnel. Let $G^\dagger$ denote the graph after folding a funnel a-v-$\{b, c\}$ in G. Then we have the following lemma.

Lemma 3 [17] *For any funnel a-v-$\{b, c\}$ in graph G, it holds that*

$$\alpha(G) = 1 + \alpha(G^\dagger).$$

We call a funnel a-v-$\{b, c\}$ in a graph with minimum degree 3 a *short funnel* if $\delta(a) = 3$ (resp., $\delta(a) = 4$) and between $N(a) \setminus \{v\}$ and $\{b, c\}$ there is at least one edge (resp., there are at least two edges meeting at the same vertex b or c). In our algorithm, we will reduce short funnels only and leave some other funnels.

Definition 4 *A graph is called a* reduced graph *if none of the above reduction operations is applicable.*

The algorithm in Figure 1 is a collection of all above reduction operations. When the graph is not a reduced graph, we can use the the algorithm in Figure 1 to reduce it and find a part of the solution.

4 Properties of vertex-cuts with size at most 2

For a disconnected graph G with a component H, we can solve instances H and $G - V(H)$ independently. Here we observe a similar property on graphs with vertex-connectivity 1 and 2.

Let v be a vertex cut in a graph G, which gives a separation $(V_1, \{v\}, V_2)$. Let $G_i = G[V_i]$, $i = 1, 2$, and $V_1^v = V_1 \setminus N(v)$. The induced graph $G[V_1^v]$ is denoted by G_1^v.

The following theorems provide a divide-and-conquer method for us to find a maximum independent set in G.

Input: A graph $G = (V, E)$ and the size s of the current partial solution (initially $s = 0$).
Output: A reduced graph $G' = (V', E')$ and the size s' of a partial solution S' with $N[S'] \cap V' = \emptyset$ in G.

1. **If** {Graph G has a component H that is a graph with at most $\rho = 28$ vertices or the line graph of a $(3, 4)$-bipartite graph}, **return** $(G', s') := \mathrm{RG}(G - V(H), s + \alpha(H))$.
2. **Elseif** {There is an unconfined vertex v}, **return** $(G', s') := \mathrm{RG}(G - v, s)$.
3. **Elseif** {There is a twin $A = \{u, v\}$}, **return** $(G', s') := \mathrm{RG}(G^\star, s + 1)$ for $G^\star = G - N[A]$ if $N(v)$ is an independent set, and $G^\star = G/N[A]$ otherwise.
4. **Elseif** {There is a short funnel}, **return** $(G', s') := \mathrm{RG}(G^\dagger, s + 1)$.
5. **Else return** $(G', s') := (G, s)$.

Figure 1: The Algorithm $\mathrm{RG}(G, s)$

Theorem 5 *For subgraphs G_1 and G_1^v defined on a separation $(V_1, \{v\}, V_2)$ in a graph G, it holds*

$$\alpha(G) = \alpha(G_1) + \alpha(G^\star),$$

where $G^\star = G - V_1$ if $\alpha(G_1) = \alpha(G_1^v)$, and $G^\star = G_2$ otherwise. A maximum independent set in a graph G can be constructed from any maximum independent sets to G_1, G_1^v and $G^\star$.
(See appendix for a proof of Theorem 5.)

For a separation $(V_1, \{u, v\}, V_2)$ of a graph G, let $G_i = G[V_i]$ $(i = 1, 2)$, $V_1^v = V_1 \backslash N(v)$, $V_1^u = V_1 \backslash N(u)$ and $V_1^{uv} = V_i \setminus N(\{u, v\})$, $i \in \{1, 2\}$. The induced graphs $G[V_1^v]$, $G[V_1^u]$ and $G[V_1^{uv}]$ are simply denoted by G_1^v, G_1^u and G_1^{uv} respectively. Let $\widetilde{G_2}$ denote the graph obtained from $G[V_2 \cup \{u, v\}]$ by adding an edge uv if v and u are not adjacent.

Theorem 6 *For subgraphs G_1, G_1^v, G_1^u and G_1^{uv} defined on a separation $(V_1, \{u, v\}, V_2)$ in a graph G, it holds*

$$\alpha(G) = \alpha(G_1) + \alpha(G^\star),$$

where

$$G^\star = \begin{cases} G[V_2 \cup \{u,v\}] & \text{if } \alpha(G_1^{uv}) = \alpha(G_1), \\ \widetilde{G_2} & \text{if } \alpha(G_1^{uv}) < \alpha(G_1^u) = \alpha(G_1^v) = \alpha(G_1), \\ G[V_2 + v] & \text{if } \alpha(G_1^u) < \alpha(G_1^v) = \alpha(G_1), \\ G[V_2 + u] & \text{if } \alpha(G_1^v) < \alpha(G_1^u) = \alpha(G_1), \\ G/(V_1 \cup \{u, v\}) & \text{if } \alpha(G_1^{uv}) + 1 = \alpha(G_1) \text{ and } \alpha(G_1^v) < \alpha(G_1), \\ G_2 & \text{otherwise } (\alpha(G_1^{uv}) + 2 \leq \alpha(G_1) \text{ and } \alpha(G_1^v) < \alpha(G_1)). \end{cases}$$

A maximum independent set in a graph G can be constructed from any maximum independent sets to G_1, G_1^v, G_1^u, G_1^{uv} and $G^\star$.
(See appendix for a proof of Theorem 6.)

Note that the above divide-and-conquer method can be used to deal with degree-1 and degree-2 vertices in the graph.

5 Branching Rules

We introduce our branching rules, which will only be applied on a graph that is reduced and connected component of it is triconnected.

Branching on a vertex
The simplest branching rule is to branch on a single vertex v by considering two cases: (i) there is a maximum independent set of G which does not contain v; (ii) every maximum independent set of G contains v. In (ii), it is shown that I_v is always contained in any maximum independent set of G [17].

Branching on a vertex v means creating two subinstances by excluding v from the independent set or including I_v to the independent set. In the first branch we will delete v from the instance whereas in the second branch we will delete $N[I_v]$ from the instance. Selecting vertices to branch on is important for efficiency of our algorithms.

A vertex cut Z of size $|Z| = 3$ is a *good* vertex cut if there is a separation (X_1, Z, X_2) such that

$$|X_1| \leq 24,\ \delta(X_1) \geq 17 \text{ and } X_1 \text{ induces a connected subgraph,}$$

and X_1 is maximal under the above two conditions. A pair of nonadjacent vertices u and v is called a *good* pair if one of u and v is a degree-5 vertex, and u and v share at least three common neighbors. A funnel a-v-$\{b, c\}$ is called a *good* funnel if vertex a has a neighbor u such that (v, u) is a good pair. The vertices on which we branch will be chosen as follows:

(i) the vertex is a vertex in a good vertex cut;

(ii) the vertex is a in a good funnel a-v-$\{b, c\}$; or

(iii) the vertex is a vertex of maximum degree $d \geq 5$.

When we branch on the vertex a of a funnel a-v-$\{b, c\}$ in (iii), we generate instance by excluding a or by including I_a. In the first branch, after removing a, vertex v becomes dominated, and we can include it in a solution. Therefore we get the following branching rule [17].

Branching on a funnel a-v-$\{b, c\}$ in a reduced instance G by either including v or including I_a in the independent set. Hence we generate the two subinstances by removing either $N[v]$ or $N[I_a]$ from G.

Branching on a complete bipartite subgraph

Lemma 7 *Let A and B be two disjoint vertex subsets in a graph G such that every two vertices $a \in A$ and $b \in B$ are adjacent. Then either $S \cap A = \emptyset$ or $S \cap B = \emptyset$ holds for any independent set S in G.*

Proof. If $S \cap A = \emptyset$ then we are done. If S contains a vertex $a \in A$, then $S \cap N(a)$ $(\supseteq S \cap B)$ is empty. ∎

For a good pair $\{u, v\}$, we have a bipartite graph between $A = \{u, v\}$ and $B = N(u) \cap N(v)$.

Branching on a good pair $\{u, v\}$ means branching by either excluding $\{u, v\}$ from the independent set or excluding $N(u) \cap N(v)$ from the independent set.

6 The Algorithm and Results

6.1 Framework for Analysis

We apply the Measure and Conquer method [5] to analyze our algorithm. In this method, we introduce a weight to each vertex in the graph according to the degree of the vertex, $w : \mathbb{Z}_+ \to \mathbb{R}_+$

(where $\mathbb{Z}_+$ and $\mathbb{R}_+$ denote the sets of nonnegative integers and nonnegative reals, respectively): we denote by w_i the weight $w(v)$ of a vertex v of each degree $i \geq 0$ and let $\Delta w_i = w_i - w_{i-1}$ for $i \geq 1$. We will assume that $\Delta w_5 \leq \Delta w_i$ for $i \geq 3$. Then we adopt $\mu(G) = \sum_i w_i n_i$ as the measure of the graph G. Usually, vertices of higher degree receive a larger weight (i.e., $\Delta w_i \geq 0$), and the weight of each vertex in the initial graph G is not greater than 1 (i.e., $\mu(G) \leq n$). We will construct a recurrence related to the measure $\mu = \mu(G)$ for each branch in our algorithm and analyze a bound for the worst ones.

For each branch operation, we will generate two subinstances G_1 and G_2 by deleting some vertices from the graph. After deleting some vertices, we can reduce the measure from two parts: the weight of the vertices being deleted and partial weight of the vertices adjacent to the deleted vertices since their degree will decrease. Let $t_{(i)}$ be a lower bound on the decrease of the measure in the subinstance (i.e., $\mu(G) - \mu(G_i) \geq t_{(i)}$). Then we get the recurrence

$$C(\mu) \leq C(\mu - t_{(1)}) + C(\mu - t_{(2)}). \tag{1}$$

The most important and complicated case in our algorithm is to branch on a vertex v of maximum degree. Let $\Delta_{out}(v)$ and $\Delta_{in}(v)$ to denote the decrease of the measure of μ in the branches of excluding v and including I_v, respectively. We get recurrence $C(\mu) = C(\mu - \Delta_{out}(v)) + C(\mu - \Delta_{in}(v))$. We give more details about lower bounds on $\Delta_{out}(v)$ and $\Delta_{in}(v)$. Let k_i denote the number of degree-i neighbors of v. Then $d = \sum_{i=3}^{d} k_i$. For the first branch, we get

$$\Delta_{out}(v) = w_d + \sum_{i=3}^{d} k_i \Delta w_i.$$

In the second branch, we will delete $N[I_v]$ from the graph. Let $\Delta(\overline{N[v]})$ denote the decrease of weight of vertices in $V(G) \setminus N[v]$ by removing $N[I_v]$ from G together with possibly weight decrease attained by reduction operations applied to $G - N[I_v]$. Then we have

$$\Delta_{in}(v) \geq w_d + \sum_{i=3}^{d} k_i w_i + \Delta(\overline{N[v]}).$$

We can branch on a vertex v of degree d with recurrence

$$\begin{aligned} C(\mu) &= C(\mu - \Delta_{out}(v)) + C(\mu - \Delta_{in}(v)) \\ &\leq C(\mu - (w_d + \textstyle\sum_{i=3}^{d} k_i \Delta w_i) + C(\mu - (w_d + \textstyle\sum_{i=3}^{d} k_i w_i + \Delta(\overline{N[v]}))). \end{aligned} \tag{2}$$

In our algorithm, we carefully select a vertex of maximum degree to branch on so that the worst recurrence (2) is as good as possible. To do so, we need to analyze lower bounds on $\Delta(\overline{N[v]})$ when the maximum degree of the graph is 5. If no vertex in $N_2(v)$ is adjacent to two vertices in $N_1(v)$, then $\Delta(\overline{N[v]}) \geq f_v \Delta w_5$, since we assume that $\Delta w_5 \leq \Delta w_i$ for $i \geq 3$. Otherwise, we have the following lower bound based on our weight setting (a proof can be found in Appendix 7.1)

$$\Delta(\overline{N[v]}) \geq f_v \Delta w_5 + g_v(\Delta w_4 - \Delta w_5). \tag{3}$$

When $N^*(v) \neq \emptyset$ (v has some extending children), the above bound may not be good enough since f_v may be small. For this case, $N[N^*(v)]$ will be removed from the graph and $\Delta(\overline{N[v]})$ can reach a desired bound.

6.2 The Algorithm

Our algorithm is simple in the sense that it consists of branching on three kinds of vertices and branching on good pairs except for how to select vertices of maximum degree 5 to branch on. A reduced degree-5 graph is a *proper graph* if it has neither good vertex cuts nor good pairs and each connected component of it has vertex connectivity at least 3. In fact, branching on a vertex v of maximum degree 5 in a proper graph will be bottlenecks in the analysis for the running time bound of our algorithms. We here identify degree-5 vertices v in proper graphs branching on which would efficiently reduce the current instance in terms of the degrees of neighbors of v. These vertices are called *optimal* vertices.

For a degree-5 vertex v in a proper graph, let $k(v) = (k_3, k_4, k_5)$, where k_i is the number of degree-i neighbors of v ($i = 3, 4, 5$). The vertex v is called *effective* if one of the following (a)-(f) holds:

(a) $(f_v, g_v) \geq (14, 0)$, $(12, 3)$ or $(10, 5)$, for $(k_4, k_5) = (0, 5)$;
(b) $(f_v, g_v) \geq (13, 0)$ or $(11, 2)$, for $(k_4, k_5) = (1, 4)$;
(c) $(f_v, g_v) \geq (12, 0)$ or $(10, 2)$, for $(k_4, k_5) = (2, 3)$;
(d) $(f_v, g_v) \geq (11, 0)$, for $(k_4, k_5) = (3, 2)$;
(e) $(f_v, g_v) \geq (12, 0)$ or $(10, 1)$, for $(k_4, k_5) = (4, 1)$; and
(f) $(f_v, g_v) \geq (10, 0)$, for $(k_4, k_5) = (5, 0)$.

Lemma 8 *Let G be a proper graph with at least one degree-5 vertex. Assume that $N^*(u) = \emptyset$ for all degree-5 vertices in G and that no degree-5 vertex is adjacent to a degree-3 vertex. Then there exists an effective vertex in G.*
(See appendix for a proof of Lemma 8.)

A degree-5 vertex v in a proper graph is called *optimal* if either (1) $k_3 \geq 1$ or (2) there is no degree-5 vertex adjacent to a degree-3 vertex, and v is effective or it holds $N^*(v) \neq \emptyset$. Our algorithm for MIS-5 is described in Figure 2.

6.3 The Result

In our algorithm, we set vertex weight as follows

$$w_i = \begin{cases} 0 & \text{for } i = 0, 1 \text{ and } 2 \\ 0.5091 & \text{for } i = 3 \\ 0.8243 & \text{for } i = 4 \\ 1 & \text{for } i = 5 \\ 1.5091 & \text{for } i = 6 \\ 1.7482 & \text{for } i = 7 \\ 1.9722 & \text{for } i = 8 \\ w_8 + (i-8)(w_5 - w_4) & \text{for } i \geq 9. \end{cases} \tag{4}$$

Lemma 9 *With the vertex weight setting (4), each recurrence generated by the algorithm in Figure 2 has an amortized branching factor not greater than* 1.1737.

The proof of this analytical lemma is moved to Appendix. From the lemma we know that the size of the search tree generated by our algorithm is not greater than 1.1737^{μ}, where μ is not greater than the number n of vertices in the initial graph since it has maximum degree 5.

Theorem 10 *A maximum independent set in a degree-5 graph of n vertices can be found in $O^*(1.1737^n)$ time.*

Input: A graph G.
Output: The size of a maximum independent set in G.

1. **If** {Graph G has a vertex cut v with a separation $(V_1, \{v\}, V_2)$ such that $\delta_{\geq 3}(V_1) \leq \delta_{\geq 3}(V_2)$}, **return** $\text{MIS5}(G[V_1]) + \text{MIS5}(G^\star)$.
2. **Elseif** {Graph G has a vertex cut $\{u, v\}$ with a separation $(V_1, \{u, v\}, V_2)$ such that $\delta_{\geq 3}(V_1) \leq \delta_{\geq 3}(V_2)$}, **return** $\text{MIS5}(G[V_1]) + \text{MIS5}(G^\star)$.
3. **Else** Let $(G, s) := \text{RG}(G, 0)$.
4. **If** {G has a vertex of degree ≥ 6}, pick up a vertex v of maximum degree, and **return** $s + \max\{\text{MIS5}(G - v), |I_v| + \text{MIS5}(G - N[I_v])\}$.
5. **Elseif** {G has a good vertex cut}, pick up a vertex v in a good vertex cut, and **return** $s + \max\{\text{MIS5}(G - v), |I_v| + \text{MIS5}(G - N[I_v])\}$.
6. **Elseif** {G has a good funnel a-v-$\{b, c\}$, **return** $s + \max\{1 + \text{MIS5}(G - N[I_a]), 1 + \text{MIS5}(G - N[v])\}$.
7. **Elseif**{G has a good pair (u, v)}, **return** $s + \max\{\text{MIS5}(G - \{u, v\}), \text{MIS5}(G - N(u) \cap N(v))\}$.
8. **Elseif** {G has a degree-5 vertex}, pick up an optimal degree-5 vertex v, and **return** $s + \max\{\text{MIS5}(G - v), |I_v| + \text{MIS5}(G - N[I_v])\}$.
9. **Else** {G is a degree-4 graph}, use an algorithm for MIS4 to solve the instance G and **return** $s + \alpha(G)$.

Note: With a few modifications, the algorithm can deliver a maximum independent set.

Figure 2: Algorithm MIS5(G)

References

[1] Bourgeois, N., Escoffier, B., Paschos, V.T., van Rooij, J.M.M.: Maximum independent set in graphs of average degree at most three in $O(1.08537^n)$. In: TAMC, LNCS 6108, Springer (2010) 373–384

[2] Bourgeois, N., Escoffier, B., Paschos, V. T., van Rooij, J. M. M., Fast algorithms for max independent set, Algorithmica 62(1-2), (2012) 382–415.

[3] Chen, J., Kanj, I.A., Xia, G.: Improved upper bounds for vertex cover. Theoretical Computer Science **411**(40-42) (2010) 3736–3756

[4] Eppstein D.: Quasiconvex analysis of backtracking algorithms. In: SODA, ACM Press (2004) 781–790

[5] Fomin, F. V., Grandoni, F., Kratsch, D.: A measure & conquer approach for the analysis of exact algorithms. J. ACM **56**(5) (2009) 1–32

[6] Fomin, F. V., Høie, K.: Pathwidth of cubic graphs and exact algorithms. Inf. Process. Lett. **97**(5) (2006) 191–196

[7] Fomin, F. V., Kratsch, D.: Exact Exponential Algorithms, Springer (2010)

[8] Fürer, M.: A faster algorithm for finding maximum independent sets in sparse graphs. In: LATIN 2006. LNCS 3887 , Springer (2006) 491–501

[9] Jian, T.: An $O(2^{0.304n})$ algorithm for solving maximum independent set problem. IEEE Transactions on Computers **35**(9) (1986) 847–851

[10] Kneis, J., Langer, A., Rossmanith, P.: A fine-grained analysis of a simple independent set algorithm. In Kannan, R., Kumar, K.N., eds.: FSTTCS 2009. V. 4 LIPIcs., Dagstuhl, Germany, (2009) 287–298

[11] Razgon, I.: Faster computation of maximum independent set and parameterized vertex cover for graphs with maximum degree 3. J. of Discrete Algorithms **7**(2) (2009) 191–212

[12] Robson, J.: Algorithms for maximum independent sets. J. of Algorithms **7**(3) (1986) 425–440

[13] Tarjan, R., Trojanowski, A.: Finding a maximum independent set. SIAM J. on Computing **6**(3) (1977) 537–546

[14] West, D.: Introduction to Graph Theory. Prentice Hall. 1996

[15] Xiao, M., Chen, J.E., Han, X.L.: Improvement on vertex cover and independent set problems for low-degree graphs. Chinese J. of Computers **28**(2) (2005) 153–160

[16] Xiao, M.: A simple and fast algorithm for maximum independent set in 3-degree graphs. In: M. Rahman and S. Fujita: WALCOM 2010, LNCS 5942, (2010) 281–292

[17] Xiao, M., Nagamochi, H.: Confining sets and avoiding bottleneck cases: A simple maximum independent set algorithm in degree-3 graphs. Theoretical Computer Science **469** (2013) 92–104

[18] Xiao, M., Nagamochi, H.: A refined algorithm for maximum independent set in degree-4 graphs. Technical report 2013-002, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University (2013) `url: http://www.amp.i.kyoto-u.ac.jp/tecrep/abst/2013/2013-002.html`

Appendix

7 The Analysis and Proof of Lemma 9

In the part, we show the details of the analysis, from which we can see how we set the vertex weight and prove Lemma 9.

7.1 Weight Setting

Let w_i denote the weight of a vertex v of degree $i \geq 0$, and define the *measure* of a graph G to be $\mu(G) = \sum_i w_i n_i$. We will use Δw_i to denote $w_i - w_{i-1}$ for $i \geq 1$. We set $w_i > 0$ for $i \geq 3$ and $w_0 = w_1 = w_2 = 0$, and

$$w_8 + (i-8)\Delta w_5, \quad i \geq 9.$$

Values of $w_i > 0$, $3 \leq i \leq 8$ will be determined after we analyze how the measure changes after each step of the algorithm.

Then an instance G with $\mu(G) = 0$ can be solved in polynomial time, since such a graph has only degree-0, degree-1 and degree-2 vertices and the maximum independent set problem can be solved in linear time. We also set

$$0 \leq w_3 \leq w_4 \leq w_5 \leq 1 \tag{5}$$

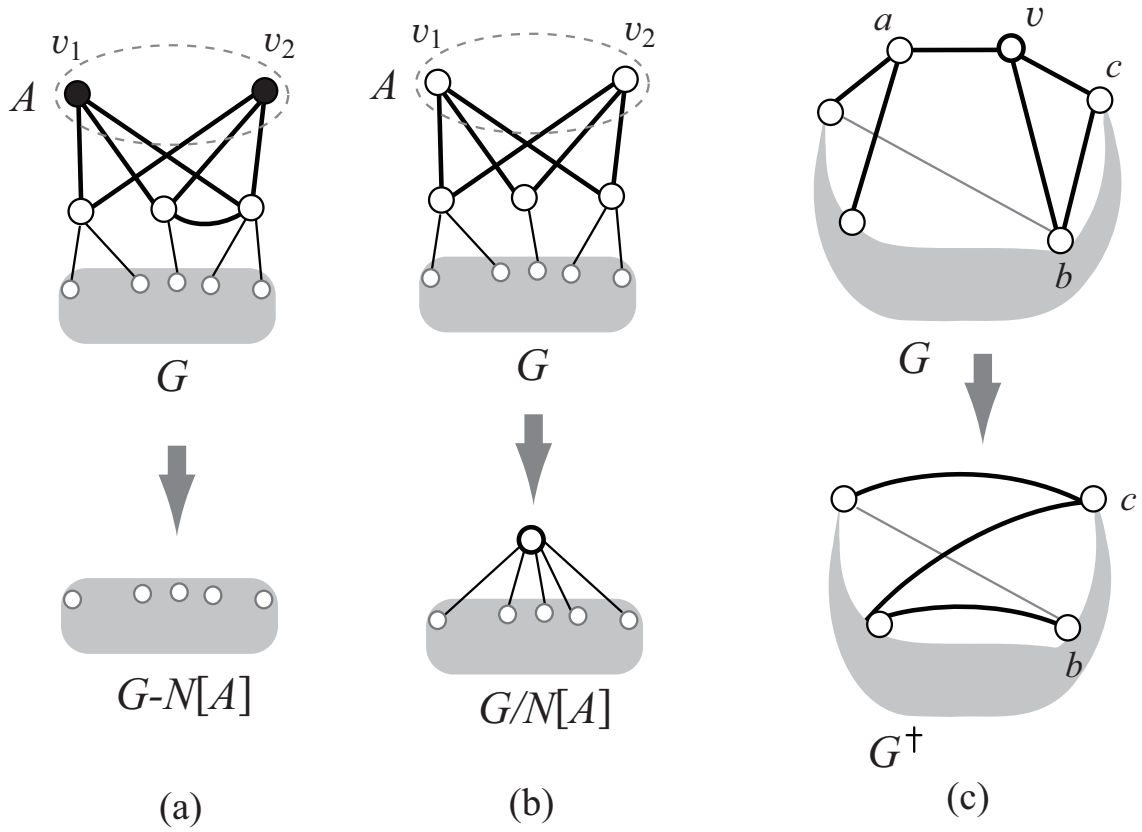

Figure 3: (a) Removing set $N[A]$ for a twin $A = \{v_1, v_2\}$; (b) Contracting set $N[A]$ for a twin $A = \{v_1, v_2\}$; (c) A short funnel b-a-$\{b, c\}$.

so that a given degree-5 graph satisfies $0 \le \mu(G) \le n$. We allow weight w_j, $j > 5$ to be larger than 1 as long as the entire weight $\mu(G)$ never increases.

We here introduce several conditions on weights w_i, $3 \le i \le 8$. To simplify our analysis, we assume that

$$0 \le \Delta w_5 \le \Delta w_i \le \Delta w_3, \quad i \ge 3. \tag{6}$$

This and $w_0 = w_1 = w_2 = 0$ imply

$$\Delta w_1 = \Delta w_2 = 0 \text{ and } w_i = \sum_{1 \le k \le i} \Delta w_k \ge (i-2)\Delta w_5, \quad i \ge 3.$$

For simplify argument, we assume

$$\Delta w_5 \le (5/9)\Delta w_4, \tag{7}$$

$$0.507 < w_3 \le 3\Delta w_5 < 0.525, \tag{8}$$

and

$$w_3 + \Delta w_3 \le w_i + \Delta w_i, \quad i \ge 3. \tag{9}$$

By the above assumptions, we can easily prove (3).

7.2 Weight Shift

To ease amortization on our analysis in this section, we introduce "shift" σ for some recurrence which is not a bottleneck in a final set of recurrences. In general, when a branch rule A generates two graphs G_1 and G_2 from a graph G, we get recurrence (1). Suppose that we know an instance G for which the branch rule A is applied by our algorithm. In this case, we temporarily increase the measure of G by $\sigma \geq 0$ (aiming to derive a better recurrence for the graph). Instead, we evaluate the branch rule A with a worse recurrence

$$C(\mu) \leq C(\mu - (t_{(1)} - \sigma)) + C(\mu - (t_{(2)} - \sigma)).$$

For analyzing the time bound of our algorithm, we introduce shift $\sigma = \Delta w_5$ in the recurrences for the divide-and-conquers on vertex cuts in Section 7.4 and branching on maximum degree ≥ 6 in Section 7.5.

7.3 Reduction Steps

We will show that the measure will not increase when we apply our reduction rules. This is important for us to construct our recurrences.

To obtain the next lemma, we assume that

$$w_i + w_j - w_{i+j-2} \geq 0, \quad 3 \leq i, j \leq 7. \tag{10}$$

$$w_3 + w_i - w_{i+1} \geq \sigma = \Delta w_5, \quad i = 3, 4. \tag{11}$$

Lemma 11 *$w_i + w_j \geq w_{i+j-2}$ holds for all $i, j \geq 1$, and $w_i + w_j \geq w_{i+j-2} + \Delta w_5$ if $i + j - 2 \leq 5$.*

Proof. If i or j is at most 2, say $i \leq 2$, then $w_i + w_j = w_j \geq w_{i+j-2}$. Let $i, j \geq 3$. For $3 \leq i, j \leq 7$, we have $w_i + w_j \geq w_{i+j-2}$ by (10). Let at least one of i and j, say i, be greater than 7. Then we have that $i + j - 2 \geq 8$ and $w_{i+j-2} = w_8 + (i + j - 2 - 8)\Delta w_5 = w_i + (j - 2)\Delta w_5$ by the definition of w_k ($k \geq 9$). Since $w_j \geq (j - 2)\Delta w_5$, this implies $w_{i+j-2} = w_i + (j - 2)\Delta w_5 \leq w_i + w_j$. Similarly, when $i + j - 2 \leq 5$, we see that $w_i + w_j \geq w_{i+j-2} + \Delta w_5$ by (11). ∎

We are ready to show that the measure never increases in $\mathrm{RG}(G, s)$. Note that any graph G after Step 2 is triconnected.

Lemma 12 *The measure μ of a graph G never increases in $RG(G, s)$. Moreover μ decreases by at least $\sigma = \Delta w_5$ after any step in $RG(G, s)$ if the maximum degree decreases by at least one after this step.*

Proof. Obviously the measure never increases by Steps 1-2 in $\mathrm{RG}(G, s)$, which simply removes some vertices from G. Furthermore, if the maximum degree of the graph decreases after Step 1 or 2, then the measure decreases by at least $\min_i \Delta w_i = \Delta w_5$. In Step 3 of $\mathrm{RG}(G, s)$, for a twin A, $G^\star$ is obtained from $G - A$ by removing or contracting $N(A)$. By Lemma 11, contracting $N(A)$ does not increase the measure. Hence the measure decreases by at least the weight in the twin $2w_3$ ($\geq \Delta w_5$).

Finally we consider Step 4 in $\mathrm{RG}(G, s)$ for folding a short funnel a-v-$\{b, c\}$. Let $\delta(a) = 4$ (the case of $\delta(a) = 3$ can be treated analogously). Let $N(a) = \{v, t_1, t_2, t_3\}$, where t_1 and t_2 are adjacent to b or c, say both of t_1 and t_2 are adjacent to b (the other cases can be treated analogously). By folding the funnel, we remove v and a from the graph and add at most four new edges bt_3, ct_1, ct_2 and ct_3. Removing vertices v and a decreases the weight of these vertices by $w_3 + w_4$, and adding the four edges increase the total weight at most by $\Delta w_{\delta(t_3)+1} + w_{\delta(c)+2} - \Delta w_{\delta(c)} \leq \max_{i \geq 4} \Delta w_i + \max_{j \geq 3}\{w_{j+2} - \Delta w_j\} \leq w_3 + w_4$ by (6) and (10), as required. Note that the degree of some vertex u in $\{b, c, t_1, t_2, t_3\}$ decreases only when $u = b$ and b is adjacent to all t_1, t_2 and t_3. In this case, the measure decreases by at least $\Delta w_{\delta(b)} \geq \Delta w_5$. ∎

Next, we will analyze the recurrences created in each step of the algorithm $\mathrm{MIS5}(G)$.

7.4 Divide-and-conquer on Vertex Cuts (Steps 1 and 2)

In fact, the number of instances generated by the divide-and-conquer procedures on vertex cuts in Steps 1-2 of MIS5(G) is not exponentially large.

Let (V_1, Z, V_2) be a separation with $\delta_{\geq 3}(V_1) \leq \delta_{\geq 3}(V_2)$ in Step 1 or 2. By (6) and (8), we know that $\mu(V_1) \leq 3\mu(V_2)$. Let $x = \mu(V_1)$, then $x \leq \frac{1}{4}\mu$. If V_1 contains only vertices of degree ≤ 2, then subinstances G_1, G_1^v, G_1^u and G_1^{uv} can be solved in constant time. The divide-and-conquer procedure reduces the instance G to an instance $G^\star$ in constant time, where clearly $\mu(G^\star) \leq \mu(G)$. Next, we assume that V_1 contains at least one vertex of degree ≥ 3. Then $w_3 \leq x \leq \frac{1}{4}\mu$.

In Step 1, we need to solve three subinstances G_1, G_1^v and $G^\star$, where $\mu(G_1^v) \leq \mu(G_1) \leq \mu(V_1) - \mu(Z) \leq x$ and $\mu(G^\star) \leq \mu(G) - x = \mu - x$. Then we get recurrence

$$C(\mu) \leq 2C(x) + C(\mu - x).$$

If we save shift $\sigma = \Delta w_5$ in the above recurrence, where $\Delta w_5 \leq 0.175$ by (8), we get

$$C(\mu) \leq 2C(x + \sigma) + C(\mu - x + \sigma) \leq 2C(x + 0.175) + C(\mu - x + 0.175). \tag{12}$$

In Step 2, we need to solve five subinstances G_1, G_1^v, G_1^u, G_1^{uv} and $G^\star$, where $\max(\mu(G_1), \mu(G_1^v), \mu(G_1^u), \mu(G_1^{uv})) \leq \mu(V_1) - \mu(Z) \leq x$ and $\mu(G^\star) \leq \mu(\widetilde{G_2}) \leq \mu(G) - x = \mu - x$ (note that we may add an edge uv in $\widetilde{G_2}$, however the degrees of u and v in $\widetilde{G_2}$ will not increase since the cut Z is minimal and u and v are adjacent to at least one vertex in V_1 in G). Then we get recurrence

$$C(\mu) \leq 4C(x) + C(\mu - x).$$

By saving shift $\sigma = \Delta w_5$ in the above recurrence, we still can get

$$C(\mu) \leq 4C(x + \sigma) + C(\mu - x + \sigma) \leq 4C(x + 0.175) + C(\mu - x + 0.175). \tag{13}$$

We can easily verify that $C(\mu) = 1.17^\mu$ satisfies the above two recurrences (12) and (13) by the substitution method (note that $w_3 \leq x \leq \frac{1}{4}\mu$).

We also analyze two special cases of applying the divide-and-conquer procedures.

Lemma 13 *Let G be a graph containing at least one vertex of degree ≥ 3. If G has a vertex cut of size 1, then after iteratively applying Step 1 in* MIS5(G) *until the graph has no vertex cut of size 1, the measure decreases by at least $\sigma = \Delta w_5$.*

Proof. Assume that the algorithm selects a vertex cut $Z = \{v\}$ of size 1 with a separation (V_1, Z, V_2) in Step 1. If V_1 contains at least one vertex of degree ≥ 3, then we can save σ from (12). If v is a vertex of degree ≥ 3, the measure will also decrease by at least $\Delta w_{\delta(v)} \geq \Delta w_5$ from v. Otherwise $V_1 \cup Z$ only contains vertices of degree ≤ 2 and $G[V_1 \cup Z]$ is a path. For this case, we reduce the G to $G^\star$ in constant time. We can see that either the measure decreases by at least Δw_5 from the neighbor of u in V_2 (u is of degree ≥ 3 in G) or $G^\star$ has a degree-1 vertex u (u is of degree 2 in G). For the later case, the graph still has a vertex cut of size 1. Therefore, the measure will finally decrease by at least $\sigma = \Delta w_5$ after iteratively applying Step 1. ∎

Lemma 14 *Let G be a graph containing at least one vertex of degree ≥ 3. If G has a degree-2 vertex not adjacent to any other degree-2 vertices, then after iteratively applying Steps 1 and 2 in* MIS5(G) *until the graph has no vertex cut of size 1 or 2, either the measure decreases by at least $\sigma = \Delta w_5$ or the resulting graph has a vertex of degree ≥ 6.*

Proof. If there is a vertex cut of size 1, then the measure can always decrease by σ by Lemma 13. Next we assume that the algorithm selects a vertex cut Z of size 2 with a separation (V_1, Z, V_2). If V_1 contains at least one vertex of degree ≥ 3, then we can save σ from (13). We only need to analyze the case that V_1 only contains vertices of degree 2 (note that G after Step 2 in MIS5(G) is biconnected).

Let v be a degree-2 vertex with two neighbors u_1 and u_2 of degree ≥ 3, where none of u_1 and u_2 is a degree-1 vertex since the graph is biconnected. If one of u_1 and u_2 is reduced to a vertex of degree ≤ 2 by these steps, the measure decreases by at least $\Delta w_3 \geq \Delta w_5$, since the operations create no new vertices (unless contracting two vertices into a vertex). Otherwise, in Step 2 we will finally select the separation $(V_1 = \{v\}, Z = \{u_1, u_2\}, V_2 = V \setminus \{v, u_1, u_2\})$. For this case, the algorithm reduces the graph G to $G^\star = G/\{v, u_1, u_2\}$ (this is exactly the operation of *folding a degree-2 vertex* in the literature [17, 3]). In this operation, we replace $\{v, u_1, u_2\}$ with a new vertex v^* with $\delta(v^*) \leq \delta(u_1) + \delta(u_2) - 2$ in the graph. If $\delta(v^*) \leq 5$ then we can save σ by Lemma 11; otherwise the graph has at least a vertex of degree ≥ 6. ∎

7.5 Branching on Vertices of Maximum Degree in Step 4

After Step 3 in MIS5(G), the current graph G is a reduced graph that is triconnected. In this step, the algorithm will branch on a vertex of maximum degree $d \geq 6$ with the recurrence (2).

Now we derive a lower bound $w_3 + 3\Delta w_5$ on $\Delta(\overline{N[v]})$ for $d \geq 6$. If no degree-0 or -1 vertex is created in $N_2(v)$ after deleting $N[v]$, the decrease of weight of vertices in $N_2(v)$ is at least $d\Delta w_5$, since $\Delta w_i \geq \Delta w_5$ by (6). If a vertex $t \in N_2(v)$ becomes of degree-1 after deleting $N[v]$, then at least two edges incident to t from $N(v)$ are removed in $G - N[v]$, and removing these two edges decrease the weight of t by w_3, where $2\Delta w_5 < w_3$ by (8). This implies that the weight decrease in $N_2(v)$ is minimized when no degree-1 vertex is created in $N_2(v)$. Let p be the number of vertices in $N_2(v)$ that become of degree-0 in $G - N[v]$. Since $|N(\overline{N_2[v]})| \geq 3$ by the triconnectivity, there are at least three vertices in $N_2(v)$ that cannot become of degree-0 in $G - N[v]$, and the weigh of them decreases by at least Δw_5. Hence we have

$$\Delta(\overline{N[v]}) \geq \min_{0 \leq p \leq f_v/3} \{pw_3 + \max\{f_v - 3p, 3\}\Delta w_5\}. \tag{14}$$

For each vertex $u \in N(v)$, there is an edge between u and a vertex in $N_2(v)$, otherwise u would dominate v. Hence we have $f_v \geq |N(v)| \geq d$. Since $\Delta(\overline{N[v]})$ never increases as f_v gets smaller, it holds $\Delta(\overline{N[v]}) \geq \min_{0 \leq p \leq d/3}\{pw_3 + \max\{d - 3p, 3\}\Delta w_5\}$, which is $qw_3 + (d - 3q)\Delta w_5$ for $q = \lfloor (d-3)/3 \rfloor$ since $w_3 \leq 3\Delta w_5$ by (8). We see that $qw_3 + (d - 3q)\Delta w_5$ is at least $w_3 + 3\Delta w_5$ for $d \geq 6$.

We also need the following inequality in the analysis. Using $\Delta w_5 \leq \Delta w_i$ and $w_3 + \Delta w_3 \leq w_i + \Delta w_i$ $(i \geq 3)$ by (6) and (9) and the property that $C(\mu - (a+b)) + C(\mu - (a+c)) \leq C(\mu - (a + b - \epsilon)) + C(\mu - (a + c + \epsilon))$ for $0 \leq a$, $0 \leq b \leq c$, and $0 \leq \epsilon \leq a + b$ (cf. [7]), we obtain the inequality

$$\begin{aligned} C(\mu) &\leq C(\mu - (b + \textstyle\sum_{u \in X} \Delta w_{\delta(u)}) + C(\mu - (c + \sum_{u \in X} w_{\delta(u)})) \\ &\leq C(\mu - (b + \textstyle\sum_{u \in X} \Delta w_5)) + C(\mu - (c + \sum_{u \in X} (w_{\delta(u)} + \Delta w_{\delta(u)} - \Delta w_5)) \\ &\leq C(\mu - (b + \textstyle\sum_{u \in X} \Delta w_5)) + C(\mu - (c + \sum_{u \in X} (2w_3 - \Delta w_5)). \end{aligned} \tag{15}$$

By (15), the recurrence (2) for $d \geq 6$ is given by

$$\begin{aligned} C(\mu) &\leq C(\mu - (w_d + \textstyle\sum_{i=3}^{d} k_i \Delta w_i) + C(\mu - (w_d + \sum_{i=3}^{d} k_i w_i + \Delta(\overline{N[v]}))) \\ &\leq C(\mu - (w_6 + 6\Delta w_5)) + C(\mu - (w_6 + 6(2w_3 - \Delta w_5) + w_3 + 3\Delta w_5)). \end{aligned}$$

We here introduce shift σ to the recurrence. Thus to save shift $\sigma = \Delta w_5$, we use a weaker recurrence

$$C(\mu) \leq C(\mu - (w_6 + 6\Delta w_5 - \sigma)) + C(\mu - (w_6 + 6(2w_3 - \Delta w_5) + w_3 + 3\Delta w_5 - \sigma)). \tag{16}$$

7.6 Branching on Vertices in Good Vertex Cuts in Step 5

Let v be a vertex in a good vertex cut Z with separation (X_1, Z, X_2), where $|X_1| \leq 24$, $\delta(X_1) \geq 17$ and X_1 is maximal under the first two conditions.

First, we assume that $|X_1| < 24$. By the maximality of X_1, we know that $|N(u) \cap X_2| \geq 2$ for any vertex $u \in Z$. Our algorithm branches on a vertex $u \in Z$ by removing u or $N[I_u]$. Note that the vertices in X_1 (resp., $X_1 - N[I_u]$) will be removed by Steps 1 and 2 in MIS5(G) on the vertex cut of size at most 2. Note that at least two vertices from X_2 will be removed in the second branch of removing $N[I_u]$ since $|N(u) \cap X_2| \geq 2$. Since $\delta(X_1) \geq 17$ implies $\sum_{x \in X_1} w_{\delta(x)} \geq \min\{w_5 + 4w_3, 2w_4 + 3w_3\} = w_5 + 4w_3$, where $w_3/3 \leq \Delta w_5 \leq \Delta w_i$ $(i \geq 3)$ by (6) and (8), we have recurrence

$$\begin{array}{rcl} C(\mu) & \leq & C(\mu - (\sum_{x \in X_1} w_{\delta(x)} + w_{\delta(u)} + \sum_{y \in N(u) \cap X_2} \Delta w_{\delta(y)})) \\ & & + C(\mu - (\sum_{x \in X_1} w_{\delta(x)} + w_{\delta(u)} + \sum_{y \in N(u) \cap X_2} w_{\delta(y)})) \\ & \leq & C(\mu - (w_5 + 5w_3 + 2\Delta w_5)) + C(\mu - (w_5 + 5w_3 + 2(2w_3 - \Delta w_5))) \\ & & \text{(by (15))}. \end{array} \tag{17}$$

When $|X_1| = 24$, in each branch the measure will decrease by at least $\sum_{x \in X_1} w_{\delta(x)} \geq 24w_3$. We can get a recurrence covered by the above one.

Let $N_2^+(v) = N(\overline{N_2[v]})$ be the set of vertices in $N_2(v)$ adjacent to at least one vertex in $V \setminus N_2[v]$. Then after Step 5, we see that

$$|N_2^+(v)| \geq 4 \text{ for all degree-5 vertices } v,$$

since otherwise ($|N_2^+(v)| = 3$) subset $X_1 = N_2[v] \setminus N_2^+(v)$ satisfies $|X_1| \leq 26 - 3 = 23$ and $\delta(X_1) \geq \delta(N[v]) \geq 20$, and then $N_2^+(v)$ would be a good vertex cut.

7.7 Branching on Good Funnels in Step 6

For a degree-5 vertex v with $N(v) = \{u_1, u_2, u_3, u_4, u_5\}$, let (v, t) is a good pair which has a good funnel u_1-t-$\{u_2, u_3\}$ for a degree-3 vertex $t \in N_2(v)$ such that $N(t) = \{u_1, u_2, u_3\}$. Note that u_2 and u_3 are adjacent. If $G[N(t)]$ contains two edges u_iu_j and u_ju_k, then t would dominate u_j, a contradiction. Hence $G[N(t)]$ contains no other edge than u_2u_3. Also $\delta(u_i) \geq 4$, $i = 2, 3$ (otherwise u_i would dominate u_3). Now there are two edges between $v \in N(u_1)$ and $\{u_2, u_3\}$. Hence $\delta(u_1) \geq 4$, since otherwise u_1-t-$\{u_2, u_3\}$ would be a short funnel. If $\delta(u_1) \geq 5$, then there is a vertex $z_1 \in (N(u_1) \setminus \{v, t\}) \cap N_2(v)$.

We branch on good funnel u_1-t-$\{u_2, u_3\}$ by removing $N[I_{u_1}]$ or removing $N[t]$. In the first branch of removing $N[I_{u_1}]$ decreases μ by at least $\sum_{u \in N[u_1]} w_{\delta(u)} + \sum_{x \in \{u_4, u_5\} \setminus N(u_1)} \Delta w_{\delta(x)} + \sum_{y \in \{u_2, u_3\}} (w_{\delta(y)} - w_{\delta(y)-2})$, which is at least $w_5 + w_4 + 3w_3 + 2\min\{w_5 - w_3, w_3\}$ for $\delta(u_1) = 4$ and $w_5 + w_4 + 4w_3 + 2\min\{w_5 - w_3, w_3\}$ for $\delta(u_1) \geq 5$, where $\min\{w_5 - w_3, w_3\} = w_5 - w_3$ by (8). The other branch of removing $N[t]$ decreases μ by at least $\sum_{a \in N[t]} w_{\delta(a)} + w_5 + \sum_{u \in N(u_1) \setminus \{t, v\}} \Delta w_{\delta(u)} \geq w_3 + 3w_4 + w_5 + 2\Delta w_5$. We get recurrence

$$\begin{array}{rcl} C(\mu) & \leq & C(\mu - (w_5 + w_4 + 3w_3 + 2(w_5 - w_3))) \\ & & + C(\mu - (w_3 + 3w_4 + w_5 + 2(w_5 - w_4))). \end{array} \tag{18}$$

7.8 Branching on Good Pairs in Step 7

In Step 5, our algorithm branches on a good pair. Let v be a degree-5 vertex such that (v, t) is a good pair for a vertex $t \in N_2(v)$, where $\delta(t) \geq 4$ or $G[N(t)]$ contains no edge (otherwise there

would be a good funnel). We denote $N(v) = \{u_1, u_2, u_3, u_4, u_5\}$, $d_i = \delta(u_i)$, $i = 1, 2, \ldots, 5$ and $N(t) \cap N(v) = \{u_i \mid i = 1, 2, \ldots, r\}$ $(r \geq 3)$. We branch on the good pair (v, t) by removing $A = \{v, t\}$ or $B = N(t) \cap N(v) = \{u_1, u_2, \ldots, u_r\}$. We distinguish three cases.

Case 1. $r = |N(t) \cap N(v)| \geq 4$: In the first branch of removing A, μ decreases by at least $w_4 + w_5 + \sum_{u \in N(t)} \Delta w_{\delta(u)} + \sum_{u \in N(t)} \Delta w_{\delta(u)-1} + \sum_{z \in N(v) \setminus N(t)} \Delta w_{\delta(z)} \geq w_4 + w_5 + \sum_{u \in N(t)} \Delta w_{\delta(u)} + 5\Delta w_5$. In the other branch of removing $N[t]$, we can always remove u_5 even when $r = 4$ since in this case v will be a degree-1 vertex and u_5 can be removed by the divide-and-conquer steps. Hence the measure μ decreases by at least $w_4 + w_5 + \sum_{u \in N(t)} w_{\delta(u)} + w_{\delta(z)} \geq w_5 + w_4 + \sum_{u \in N(t)} w_{\delta(u)} + w_3$. We have recurrence

$$\begin{aligned} C(\mu) &\leq C(\mu - (w_4 + w_5 + \textstyle\sum_{u \in N(t)} \Delta w_{\delta(u)} + 5(w_5 - w_4))) \\ &\quad + C(\mu - (w_5 + w_4 + \textstyle\sum_{u \in N(t)} w_{\delta(u)} + w_3)) \\ &\leq C(\mu - (w_4 + w_5 + 4(w_5 - w_4) + 5(w_5 - w_4))) \\ &\quad + C(\mu - (w_5 + w_4 + 4(2w_3 - (w_5 - w_4)) + w_3)) \text{ (by (15))}. \end{aligned} \tag{19}$$

Case 2. $r = 3$ and $\delta(t) \geq 4$. Denote $d_i = \delta(u_i)$, $i = 1, 2, \ldots, 5$, where we assume without loss of generality that $3 \leq d_1 \leq d_2 \leq d_3 \leq 5$. Let ℓ be the number of degree-3 neighbors in $N(t)$. For each $u_i \in N(t)$ with $i \leq \ell$, we let z_i $(\neq v, t)$ denote the third neighbor of u_i. Note that $z_i \notin B$ (otherwise u_i would dominate $z_i = u_j$), and for $1 \leq i < j \leq \ell$, it holds $z_i \neq z_j$ (otherwise $\{u_i, u_j\}$ would be a twin). We consider two subcases.

(a) $\delta(t) = 5$: Let $x_1, x_2 \in N(t) - B$ be the two neighbors of v, where $x_1, x_2 \notin N[v]$. In the first branch of removing A, the weight of vertices in $X = \{u_4, u_5, x_1, x_2\}$ decreases, and also u_i with $i \leq \ell$ becomes a degree-1 vertex, whose neighbor z_i will be removed by the divide-and-conquer steps. Hence the first branch decreases μ by at least $w_5 + w_{\delta(t)} + \sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \ell w_3 + \sum_{u \in X \setminus \{z_1, \ldots, z_\ell\}} \Delta w_{\delta(u)} \geq 2w_5 + \sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \ell w_3 + (4 - \ell)\Delta w_5$.

In the other branch of removing B, the total weight in the vertices in $\{v, t, u_1, u_2, u_3\}$ becomes zero. There are at least two edges between B and $G - \{v, t, u_1, u_2, u_3\}$ (otherwise a vertex in B dominates some other vertex in B), and the decrease of weight in the vertices in $G - \{v, t, u_1, u_2, u_3\}$ is then at least $2\Delta w_5$. The branch of removing B decreases μ by at least $w_5 + w_{\delta(t)} + \sum_{i=1,2,3} w_{d_i} + 2\Delta w_5$.

Therefore we get recurrences:

$$\begin{aligned} C(\mu) &\leq C(\mu - (6w_5 - 4w_4 + \textstyle\sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \ell(w_3 + w_4 - w_5))) \\ &\quad + C(\mu - (4w_5 - 2w_4 + \textstyle\sum_{i=1,2,3} w_{d_i})) \end{aligned} \tag{20}$$

for all $3 \leq d_1 \leq d_2 \leq d_3 \leq 5$.

(b) $\delta(t) = 4$: Let $x_1 \in N(t) - B$ be the neighbor of v, where $x_1 \notin N[v]$. Note that $z_i \notin \{u_4, u_5\}$ (otherwise t-u_i-$\{v, z_i\}$ would be a short funnel). In the first branch of removing A, the weight of vertices in $\{x_1, u_4, u_5\}$ decreases, and u_i with $i \leq \ell$ becomes a degree-1 vertex, whose neighbor z_i will be removed by the divide-and-conquer steps (possibly $x_1 = z_i$ for some $i \leq \ell$). Hence the first branch decreases μ by at least $w_5 + w_{\delta(t)} + \sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \max\{\ell w_3, \Delta w_{\delta(x_1)}\} + \Delta w_{d_4} + \Delta w_{d_5} \geq w_5 + w_4 + \sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \max\{\ell w_3, \Delta w_5\} + 2\Delta w_5$.

In the second branch of removing B, the total weight in the vertices in $\{v, t, u_1, u_2, u_3\}$ becomes zero, and t becomes a degree-1 vertex, whose neighbor x_1 will be removed by the divide-and-conquer steps. Hence the branch of removing B decreases μ by at least $w_5 + w_{\delta(t)} + \sum_{i=1,2,3} w_{d_i} + w_{\delta(x_1)}$. Therefore we get recurrences:

$$\begin{aligned} C(\mu) &\leq C(\mu - (3w_5 - w_4 + \textstyle\sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \max\{\ell w_3, w_5 - w_4\})) \\ &\quad + C(\mu - (w_5 + w_4 + w_3 + \textstyle\sum_{i=1,2,3} w_{d_i})) \end{aligned} \tag{21}$$

for all $3 \le d_1 \le d_2 \le d_3 \le 5$.

Case 3. $r = 3$, $N(t) \subseteq N(v)$ and $G[N(t)]$ contains no edge: Denote $d_i = \delta(u_i)$, $i = 1, 2, \ldots, 5$, where we assume without loss of generality that $3 \le d_1 \le d_2 \le d_3 \le 5$. Let ℓ be the number of degree-3 neighbors in $N(t)$. For each $u_i \in N(t)$ with $i \le \ell$, we let z_i ($\neq v, t$) denote the third neighbor of u_i, and let $\widetilde{d}_i = \delta(z_i)$. Note that $z_i \notin N(v)$ (otherwise t-u_i-$\{z_i, v\}$ would be a short funnel), and for $1 \le i < j \le \ell$, it holds $z_i \neq z_j$ (otherwise $\{u_i, u_j\}$ would be a twin).

In the first branch of removing A, the weight of u_4 and u_5 deceases by at least $2\Delta w_5$ in total, and u_i with $i \le \ell$ becomes a degree-1 vertex, whose neighbor z_i will be removed by the divide-and-conquer steps. Hence the first branch decreases μ by at least $w_5 + w_3 + \sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \sum_{i \le \ell} w_{\widetilde{d}_i} + 2\Delta w_5$. In the other branch of removing B, the total weight in the vertices in $\{v, t, u_1, u_2, u_3\}$ becomes zero, and the weight of vertices z_i, $i \le \ell$ decreases by $\sum_{i \le \ell} \Delta w_{\widetilde{d}_i}$. There are $p = \sum_{i=1,2,3}(d_i - 2)$ edges between B and $G - \{v, t, u_1, u_2, u_3\}$, and three of them can meet at the same vertex x of degree ≥ 4 (otherwise $\{t, x\}$ would be a twin). Since $w_4 = \Delta w_4 + w_3 \ge \Delta w_5 + 2\Delta w_5 > 3\Delta w_5$ by (8), the decrease of weight in the vertices in $G - \{v, t, u_1, u_2, u_3\}$ is then at least $\max\{\sum_{i \le \ell} \Delta w_{\widetilde{d}_i}, p\Delta w_5\}$. The second branch of removing B decrease μ by at least $w_5 + w_3 + \sum_{i=1,2,3} w_{d_i} + \max\{\sum_{i \le \ell} \Delta w_{\widetilde{d}_i}, p\Delta w_5\}$. Therefore we get recurrences:

$$\begin{aligned} C(\mu) \quad \le \quad & C(\mu - (3w_5 - 2w_4 + w_3 + \textstyle\sum_{i=1,2,3}(w_{d_i} - w_{d_i-2}) + \sum_{i \le \ell} w_{\widetilde{d}_i})) \\ & + C(\mu - (w_5 + w_3 + \textstyle\sum_{i=1,2,3} w_{d_i} \\ & \qquad + \max\{\textstyle\sum_{i \le \ell} \Delta w_{\widetilde{d}_i}, \sum_{i=1,2,3}(d_i - 2)(w_5 - w_4)\})) \end{aligned} \tag{22}$$

for all $3 \le d_1 \le d_2 \le d_3 \le 5$ and $3 \le \widetilde{d}_i \le 5$ with $i \le \ell$.

7.9 Branching on Optimal Vertices of Maximum Degree 5 in Step 8

After Step 7, if the graph still contains a degree-5 vertex, then the graph is a proper graph, and in this step the algorithm will select an optimal degree-5 vertex v to branch on with the recurrence (2) at least.

We consider the first branch of removing v such that $k_3 = 1$ or $N(v) \setminus N^*(v)$ contains a degree-3 neighbor of v. In this case, we can further decrease the measure by $\sigma = \Delta w_5$ after the first branch of removing v (see the following lemma).

Lemma 15 *Let v be an optimal vertex in a proper graph G such that $k_3 = 1$ or $N(v) \setminus N^*(v)$ contains a degree-3 neighbor of v. Then the measure decreases by at least $w_5 + \sum_{u \in N(v)} w_{\delta(u)} + \sigma$ in the branch of removing v when shift σ is save from Lemma 14, reduction operations or* (16) (*in the first four steps in* MIS5(G)).

Proof. In this case, each degree-3 neighbor u becomes a degree-2 vertex in $G - v$ in which u is not adjacent to any other degree-2 vertex. By Lemma 14, we know that after Steps 1 and 2, either the measure decreases by at least $\sigma = \Delta w_5$ or the resulting graph contains a vertex of degree ≥ 6. If the maximum degree of the graph decreases in Step 3, then the measure will decrease by at least $\sigma = \Delta w_5$ by Lemma 12. Otherwise our algorithm branches on a vertex of degree ≥ 6 with recurrence (16) in Step 4 wherein shift $\sigma = \Delta w_5$ is already saved. Therefore, for a degree-5 vertex v with $k_3 = 1$ or $N(v) \setminus N^*(v)$ contains a degree-3 neighbor of v, we can further decrease the measure μ by $\sigma = \Delta w_5$. ∎

The following lemma provides a lower bound on $\Delta(\overline{N[v]})$ in the recurrence (2) when branching on a degree-5 vertex in a proper graph, The proof of which is delayed to Appendix 8.3.

Lemma 16 *Let v be an optimal degree-5 vertex in a proper graph G. Then it holds*

$$\Delta(\overline{N[v]}) \geq \lambda(k_3, k_4, k_5),$$

where

$$\lambda(k_3,k_4,k_5) = \begin{cases} 14\Delta w_5 & \text{if } (k_3,k_4,k_5)=(0,0,5) \\ 9\Delta w_5+2\Delta w_4 & \text{if } (k_3,k_4,k_5)=(0,1,4) \\ 8\Delta w_5+2\Delta w_4 & \text{if } (k_3,k_4,k_5)=(0,2,3) \\ 9\Delta w_5+\Delta w_4 & \text{if } (k_3,k_4,k_5)=(0,4,1) \\ 11\Delta w_5 & \text{if } (k_3,k_4,k_5)=(0,3,2) \\ 10\Delta w_5 & \text{if } (k_3,k_4,k_5)=(0,5,0) \\ 2w_3+6\Delta w_5 & \text{if } k_3 \geq 2 \text{ and } N(v)\setminus N^*(v) \text{ contains} \\ & \text{no degree-3 neighbor of } v \\ 10\Delta w_5 & \text{if } k_3 = 1 \text{ or } N(v)\setminus N^*(v) \text{ contains} \\ & \text{a degree-3 neighbor of } v. \end{cases}$$

By applying Lemma 15 and Lemma 16 to the weight decreases in the first and second branches of (2) for $d = 5$, we get recurrence

$$\begin{aligned} C(\mu) \quad \leq \quad & C(\mu-(w_5+k_3w_3+k_4(w_4-w_3)+k_5(w_5-w_4)+\epsilon\sigma)) \\ & +C(\mu-(w_5+k_3w_3+k_4w_4+k_5w_5+\lambda(k_3,k_4,k_5))) \end{aligned} \tag{23}$$

for all nonnegative integers (k_3, k_4, k_5) with $k_3 + k_4 + k_5 = 5$, where $\sigma = \Delta w_5$, and $\epsilon = 1$ if $k_3 = 1$ or $N(v) \setminus N^*(v)$ contains a degree-3 neighbor of v and $\epsilon = 0$ otherwise.

7.10 Reducing to Degree-4 Graphs in Step 9

When the maximum degree of the current graph is at most 4, we invoke an exact algorithm [18] that runs in $O^*(1.137567^{\mu'(G)})$ time for a degree-4 graph G with measure $\mu'(G) = \sum_{1\leq i\leq 4} w'_i n_i$ where n_i is the number of degree-i vertices in G, and w'_i is a weight of a degree-i vertex defined by $w'_0 = w'_1 = w'_2 = 0$, $w'_3 = 0.62225$ and $w'_4 = 1$. Then it holds $C(\mu(G)) = O(1.137567^{\mu'(G)})$. Since $\mu'(G)/\mu(G) = \sum_{3\leq i\leq 4} w'_i n_i / \sum_{3\leq i\leq 4} w_i n_i \leq \max\{w'_3/w_3, w'_4/w_4\}$, we have $C(\mu(G)) = O(1.137567^{\mu'(G)})$ $= O(1.137567^{\max\{0.62225/w_3, 1/w_4\}\mu(G)})$. Hence we only need to consider the following two cases.

$$C(\mu) = O(1.137567^{(1/w_4)\mu}), \tag{24}$$

and

$$C(\mu) = O(1.137567^{(0.62225/w_3)\mu}). \tag{25}$$

7.11 Final Solution to Weights

Recurrences (16) to (25) generate the constraints in our quasiconvex program. By solving the quasiconvex program under conditions (5) to (11) and (24), and (25) according to the method introduced in [4], we get a bound 1.17367 of the branching factor for all recurrences by setting $w_3 = 0.50906$, $w_4 = 0.82426$, $w_5 = 1$, $w_6 = 1.509050$, $w_7 = 1.748217$ and $w_8 = 1.972241$. This verifies Lemma 9.

Now recurrences (23) with $(k_4, k_5) = (1, 4)$ and $(2, 3)$ and constraints $\Delta w_5 \leq (5/9)\Delta w_4$ in (7), $w_8 \leq w_5 + w_5$ in (10) and (24) attain the tight branching factor 1.1737.

8 Some proofs

8.1 Proofs of the theorems in Section 4

Theorem 5 For subgraphs G_1 and G_1^v defined on a separation $(V_1, \{v\}, V_2)$ in a graph G, it holds

$$\alpha(G) = \alpha(G_1) + \alpha(G^\star),$$

where $G^\star = G - V_1$ if $\alpha(G_1) = \alpha(G_1^v)$, and $G^\star = G_2$ otherwise. A maximum independent set in a graph G can be constructed from any maximum independent sets to G_1, G_1^v and $G^\star$.

Theorem 5 follows from the next lemma and its proof.

Lemma 17 *It holds that $\alpha(G_1) \geq \alpha(G_1^v)$ and*

$$\alpha(G) = \begin{cases} \alpha(G_1^v) + \alpha(G[V_2 + v]) & \text{if } \alpha(G_1) = \alpha(G_1^v) \\ \alpha(G_1) + \alpha(G_2) & \text{if } \alpha(G_1) > \alpha(G_1^v). \end{cases}$$

Proof. Since G_1^v is an induced subgraph of G, we know that $\alpha(G_1) \geq \alpha(G_1^v)$ holds. Let S be a maximum independent set of G. Then we have $|S \cap V_1| \leq \alpha(G_1)$ and $|S \setminus V_1| \leq \alpha(G[V_2 + v])$.

First assume that $\alpha(G_1) = \alpha(G_1^v)$. Then we have $|S \cap V_1| \leq \alpha(G_1) = \alpha(G_1^v) = |S_1^v|$ for any maximum independent set S_1^v of G_1^v. Then $\alpha(G) = |S \cap V_1| + |S \setminus V_1| \leq \alpha(G_1^v) + \alpha(G[V_2 + v])$. On the other hand, $|S_1^v \cup (S \setminus V_1)| \leq \alpha(G)$ since $S_1^v \cup (S \setminus V_1)$ is an independent set of G since $G[V_2 + v]$ and G_1^v are separated by cut $V_1 \setminus V_1^v$. Therefore, $\alpha(G) = \alpha(G_1^v) + \alpha(G[V_2 + v])$.

Next, we consider the case of $\alpha(G_1) > \alpha(G_1^v)$. Let $\hat{S}_1$ be a maximum independent set of $G[V_1 + v]$. We have that $|\hat{S}_1| \leq \min\{\alpha(G_1^v) + 1, \alpha(G_1)\} \leq \alpha(G_1)$. We have that $|S \cap V_2| \leq \alpha(G_2)$ and $|S \setminus V_2| \leq |\hat{S}_1| \leq \alpha(G_1)$. Then $\alpha(G) = |S \cap V_2| + |S \setminus V_2| \leq \alpha(G_2) + \alpha(G_1)$. On the other hand, $\alpha(G) \geq |(S \cap V_2) \cup S_1| = |S \cap V_2| + \alpha(G_1)$ for any maximum independent set S_1 of G_1, since $(S \cap V_2) \cup S_1$ is also an independent set of G since G_2 and G_1 are separated by cut $\{v\}$. Therefore, $\alpha(G) = \alpha(G_1) + \alpha(G_2)$. ∎

Theorem 6 For subgraphs G_1, G_1^v, G_1^u and G_1^{uv} defined on a separation $(V_1, \{u, v\}, V_2)$ in a graph G, it holds

$$\alpha(G) = \alpha(G_1) + \alpha(G^\star),$$

where

$$G^\star = \begin{cases} G[V_2 \cup \{u,v\}] & \text{if } \alpha(G_1^{uv}) = \alpha(G_1), \\ \widetilde{G_2} & \text{if } \alpha(G_1^{uv}) < \alpha(G_1^u) = \alpha(G_1^v) = \alpha(G_1), \\ G[V_2 + v] & \text{if } \alpha(G_1^u) < \alpha(G_1^v) = \alpha(G_1), \\ G[V_2 + u] & \text{if } \alpha(G_1^v) < \alpha(G_1^u) = \alpha(G_1), \\ \widehat{G_2} & \text{if } \alpha(G_1^{uv}) + 1 = \alpha(G_1) \text{ and } \alpha(G_1^v) < \alpha(G_1), \\ G_2 & \text{otherwise } (\alpha(G_1^{uv}) + 2 \leq \alpha(G_1) \text{ and } \alpha(G_1^v) < \alpha(G_1)). \end{cases}$$

A maximum independent set in a graph G can be constructed from any maximum independent sets to G_1, G_1^v, G_1^u, G_1^{uv} and $G^\star$.

Theorem 6 follows from the next lemma and its proof.

Lemma 18 *Assume without loss of generality that $\alpha(G_1^u) \leq \alpha(G_1^v)$. Then it holds that $\alpha(G_1^{uv}) \leq \alpha(G_1^u) \leq \alpha(G_1^v) \leq \alpha(G_1)$ and*

$$\alpha(G) = \begin{cases} \alpha(G_1^{uv}) + \alpha(G[V_2 \cup \{u,v\}]) & \text{if } \alpha(G_1^{uv}) = \alpha(G_1), & \text{(i)} \\ \alpha(G_1) + \alpha(\widetilde{G_2}) & \text{if } \alpha(G_1^{uv}) < \alpha(G_1^u) = \alpha(G_1^v) = \alpha(G_1), & \text{(ii)} \\ \alpha(G_1^v) + \alpha(G[V_2 + v]) & \text{if } \alpha(G_1^u) < \alpha(G_1^v) = \alpha(G_1), & \text{(iii)} \\ \alpha(G_1) + \alpha(\widehat{G_2}) & \text{if } \alpha(G_1^{uv}) + 1 = \alpha(G_1) \text{ and } \alpha(G_1^v) < \alpha(G_1), & \text{(iv)} \\ \alpha(G_1) + \alpha(G_2) & \text{if } \alpha(G_1^{uv}) + 2 \leq \alpha(G_1) \text{ and } \alpha(G_1^v) < \alpha(G_1), & \text{(v)} \end{cases}$$

where $\widetilde{G_2}$ is the graph obtained from $G[V_2 \cup \{u,v\}]$ by adding an edge uv if v and u are not adjacent and $\widehat{G_2} = G/(V_1 \cup \{u,v\})$ is the graph obtained from G by contracting $V_1 \cup \{u,v\}$ into a single vertex z and deleting multi-edges and self-loops.

Proof. Since G_1^{uv} is an induced subgraph of G_1^u (resp., G_1^v), and G_1^u (resp., G_1^v) is an induced subgraph of G_1, we know that $\alpha(G_1^{uv}) \leq \alpha(G_1^u) \leq \alpha(G_1)$ and $\alpha(G_1^{uv}) \leq \alpha(G_1^v) \leq \alpha(G_1)$. We consider the other five possible relations among $\alpha(G_1^{uv})$, $\alpha(G_1^u)$, $\alpha(G_1^v)$ and $\alpha(G_1)$. In the following, S (resp., S_1^{uv}, S_1^u, S_1^v and S_1) denotes an arbitrary maximum independent set of G (resp., G_1^{uv}, G_1^u, G_1^v and G_1^u).

Case (i). $\alpha(G_1^{uv}) = \alpha(G_1)$: We partition $V(G)$ into V_1^{uv}, $Z = V_1 \setminus V_1^{uv}$ and $V_2 \cup \{u,v\}$ so that there is no edge between V_1^{uv} and $V_2 \cup \{u,v\}$. Hence we have $\alpha(G) \geq \alpha(G_1^{uv}) + \alpha(G[V_2 \cup \{u,v\}])$. The converse can be obtained by

$$\begin{aligned}
\alpha(G) = |S| &= |S \cap V_1| + |S \cap (V_2 \cup \{u,v\})| &&\leq \alpha(G_1) + |S \cap (V_2 \cup \{u,v\})| \\
&\leq \alpha(G_1^{uv}) + |S \cap (V_2 \cup \{u,v\})| &&= |S_1^{uv}| + |S \cap (V_2 \cup \{u,v\})| \\
&\leq \alpha(G_1^{uv}) + \alpha(G[V_2 \cup \{u,v\}]).
\end{aligned}$$

Case (ii). $\alpha(G_1^{uv}) < \alpha(G_1^u) = \alpha(G_1^v) = \alpha(G_1)$: This holds because $V_2 \cup \{u,v\}$ is a subgraph of $\widetilde{G_2}$. We first show that G has a maximum independent set S containing at most one vertex in $\{u,v\}$. If $u, v \in S$, then can replace $S \cap (V_1 \cup \{u\})$ with S_1^v in S to get another maximum independent set $S' = S_1^v \cup \{v\} \cup (S \cap (V_2 \setminus N(\{u,v\})))$ of G, since

$$\begin{aligned}
\alpha(G) = |S| &= |S \cap V_1^{uv}| + |\{u,v\}| + |S \cap (V_2 \setminus N(\{u,v\}))| \\
&\leq \alpha(G_1^{uv}) + |\{u,v\}| + |S \cap (V_2 \setminus N(\{u,v\}))| \\
&\leq \alpha(G_1^v) - 1 + |\{u,v\}| + |S \cap (V_2 \setminus N(\{u,v\}))| \\
&= |S_1^v| + |\{v\}| + |S \cap (V_2 \setminus N(\{u,v\}))| \\
&\leq \alpha(G).
\end{aligned}$$

Hence G has a maximum independent set S such that $|\{u,v\} \cap S| = 1$. Now we observe that

$$\begin{aligned}
\alpha(G) = |S| &= |S \cap V_1| + |S \cap (V_2 \cup \{u,v\})| \\
&\leq |S \cap V_1| + \alpha(\widetilde{G_2}) \\
&\leq \max\{\alpha(G_1^u), \alpha(G_1^v), \alpha(G_1)\} + \alpha(\widetilde{G_2}) \\
&= \min\{\alpha(G_1^u), \alpha(G_1^v), \alpha(G_1)\} + \alpha(\widetilde{G_2}) \\
&\leq \alpha(G),
\end{aligned}$$

indicating that $\alpha(G) = \alpha(G_1) + \alpha(\widetilde{G_2})$.

Case (iii). $\alpha(G_1^u) < \alpha(G_1^v) = \alpha(G_1)$: We partition $V(G)$ into V_1^v, $Z = \{u\} \cup (V_1 \setminus V_1^v)$ and $V_2 + v$ such that there is no edge between V_1^v and $V_2 + v$. Hence $\alpha(G) \geq \alpha(G_1^v) + \alpha(G[V_2 + v])$. If $u \notin S$, then $|S \cap (V_1 + u)| = |S \cap V_1| \leq \alpha(G_1) = \alpha(G_1^v)$. If $u \in S$, then $|S \cap (V_1 + u)| = |S \cap V_1^u| + 1 \leq \alpha(G_1^u) + 1 \leq \alpha(G_1^v)$. In any case we have

$$\begin{aligned}
\alpha(G) = |S| &= |S \cap (V_1 + u)| + |S \cap (V_2 + v)| \leq \alpha(G_1^v) + |S \cap (V_2 + v)| \\
&= |S_1^v| + |S \cap (V_2 + v)| \leq \alpha(G).
\end{aligned}$$

Case (vi). $\alpha(G_1^{uv}) + 1 = \alpha(G_1)$ and $\alpha(G_1^v) < \alpha(G_1)$: We first observe that assumption $\alpha(G_1^u) \leq \alpha(G_1^v) < \alpha(G_1)$ implies that G has a maximum independent set S with $|\{u,v\} \cap S| = 0$ or 2. If

$\{u,v\} \cap S = \{u\}$, then we can replace $S \cap (V_1 \cup \{u,v\})$ with S_1 in S to get another maximum independent $S' = S_1 \cup (S \cap V_2)$ set of G, where

$$\begin{aligned}\alpha(G) = |S| &= |S \cap V_1| + |S \cap \{u,v\}| + |S \cap V_2| \\ &\le \alpha(G_1^u) + 1 + |S \cap V_2| \\ &\le \alpha(G_1) + |S \cap V_2| \\ &= |S_1| + |S \cap V_2| \le \alpha(G).\end{aligned}$$

Symmetrically if $\{u,v\} \cap S = \{v\}$, then $\alpha(G_1^v) < \alpha(G_1)$ implies that we can replace $S \cap (V_1 \cup \{u,v\})$ with S_1 in S to get another maximum independent set $S' = S_1 \cup (S \cap V_2)$ of G. Hence G has a maximum independent set S with $|\{u,v\} \cap S| = 0$ or 2.

When $|\{u,v\} \cap S| = 2$, we have $\alpha(G) = \alpha(G_1^{uv}) + |S \cap (V_2 \cup \{u,v\})| \le \alpha(G_1^{uv}) + \alpha(\widehat{G_2}) + 1$. When $|\{u,v\} \cap S| = 0$, we have $\alpha(G) = \alpha(G_1) + |S \cap (V_2 \cup \{u,v\})| \le \alpha(G_1) + \alpha(\widehat{G_2})$. In any case, we have $\alpha(G) \le \alpha(G_1^{uv}) + 1 + \alpha(\widehat{G_2}) = \alpha(G_1) + \alpha(\widehat{G_2})$. We show the converse. For a maximum independent set S^* of $\widehat{G_2}$, if $z \in S^*$ (resp., $z \notin S^*$) then we have an independent set S' of G such that $S' = S_1^{uv} \cup (S^* \setminus \{z\}) \cup \{u,v\}$ (resp., $S' = S_1 \cup S^*$) and $\alpha(G) \ge |S'| = \alpha(G_1^{uv}) + \alpha(\widehat{G_2}) - 1 + 2$ (resp., $\alpha(G) \ge |S'| = \alpha(G_1) + \alpha(\widehat{G_2})$), where $\alpha(G_1^{uv}) + 1 + \alpha(\widehat{G_2}) = \alpha(G_1) + \alpha(\widehat{G_2})$ by assumption.

Case (v). $\alpha(G_1^v) < \alpha(G_1)$ and $\alpha(G_1^{uv}) + 2 \le \alpha(G_1)$: As in Case (iv), assumption $\alpha(G_1^u) \le \alpha(G_1^v) < \alpha(G_1)$ implies that G has a maximum independent set S with $|\{u,v\} \cap S| = 0$ or 2. If $|\{u,v\} \cap S| = 2$, then we can replace $S \cap (V_1 \cup \{u,v\})$ with S_1 in S to get another maximum independent set $S' = S_1 \cup (S \cap V_2)$ of G with $\{u,v\} \cap S' = \emptyset$, where

$$\begin{aligned}\alpha(G) = |S| &= |S \cap V_1^{uv}| + |S \cap \{u,v\}| + |S \cap V_2| \\ &\le \alpha(G_1^{uv}) + 2 + |S \cap V_2| \\ &\le \alpha(G_1) + |S \cap V_2| \\ &= |S_1| + |S \cap V_2| \le \alpha(G).\end{aligned}$$

Hence G has a maximum independent set S with $S \cap \{u,v\} = \emptyset$, indicating that This means that $\alpha(G) = \alpha(G_1) + \alpha(G_2)$. ∎

8.2 The proof of Lemma 8

Lemma 8 *Let G be a proper graph with at least one degree-5 vertex. Assume that $N^*(u) = \emptyset$ for all degree-5 vertices in G and that no degree-5 vertex is adjacent to a degree-3 vertex. Then there exists an effective vertex in G.*

Proof. Note that any degree-5 vertex v with $N^*(v) = \emptyset$ satisfies $f_v \ge 2\delta(v) = 10$, and we are done if there is a degree-5 vertex with $(k_4, k_5) = (5,0)$ or $(k_4, k_5) = (3,2)$ (note that for $(k_4, k_5) = (3,2)$ it holds $f_v \ge 2\delta(v) + 1 = 11$ by parity condition). To prove the lemma, we assume that G contains no degree-5 vertex v with $(k_4, k_5) = (5,0)$ or $(3,2)$ (also no degree-5 vertex with $k_3 \ge 1$ by the assumption in the lemma).

In G, we choose a degree-5 vertex v so that $(k_4, k_5) = (4,1)$ or $(1,4)$ holds (if any) before a degree-5 vertex v with $(k_4, k_5) = (2,3)$ is selected. Let $N(v) = \{u_i \mid 1 \le i \le 5\}$. Since $N^*(v) = \emptyset$, each $u_i \in N(v)$ has at least two neighbors in $N_2(v)$, which we denote by x_i and y_i.

We distinguish three cases.

(i) $(k_4, k_5) = (4,1)$: Assume $(f_v, g_v) \le (10,0)$ (otherwise $(f_v, g_v) \ge (12,0)$ or $(10,1)$ and we are done). Since $N^*(v) = \emptyset$, it holds $f_v = 2\delta(v) = 10$ and each neighbor $u_i \in N(v)$ has exactly two neighbors $x_i, y_i \in N_2(v)$. Also $N_2(v)$ contains only degree-5 vertices since $g_v = 0$. Hence the induced graph $G[N(v)]$ consists of an edge u_1u_2 and a path $u_4u_5u_3$ with $\delta(u_5) = 5$. Then vertex $a = u_5$ is a degree-5 vertex with $k(a) = (0,2,3)$, where the two neighbors $x_5, y_5 \in N(a) \setminus N[v]$ have no neighbors in $N[v]$. Hence $f_a \ge 12$ holds and a is an effective vertex.

(ii) $(k_4, k_5) = (1, 4)$: Assume $(f_v, g_v) \leq (11, 1)$ (otherwise $(f_v, g_v) \geq (13, 0)$ or $(11, 2)$ and we are done). Since $N^*(v) = \emptyset$, it holds $f_v = 2\delta(v) + 1 = 11$ by parity condition, and one neighbor $u' \in N(v)$ has exactly three neighbors in $N_2(v)$ and each of the other neighbors in $N(v) - u'$ has exactly two neighbors in $N_2(v)$. In this case, u' cannot be the degree-4 vertex, since otherwise $G[N(v)]$ contains a 4-cycle $u_2u_3u_4u_5$, where nonadjacent degree-5 vertices u_2 and u_4 would be a good pair. Hence $G[N(v)]$ is a single path $u_1u_2u_3u_4u_5$ from the degree-4 vertex u_1 to $u' = u_5$. We show that u_2 or u_4 is an effective vertex. Since $g_v \leq 1$, there is at most one vertex $t \in N_2(v)$ such that $\delta(t) = 4$ or t has two neighbors in $N(v)$. If each of x_4 and y_4 has no other neighbor in $N(v)$ than u_4, then $a = u_4$ is an effective vertex with $k(a) = (0, 0, 5)$ and $f_a \geq 14$ (when $\delta(x_4) = \delta(y_4) = 5$) or $k(a) = (0, 1, 4)$ and $f_a \geq 13$ (when $\delta(x_4)$ or $\delta(y_4)$ is 4).

Now assume that one of x_4 and y_4 has a neighbor $u_j \in N(v)$ other than u_4. Then all other vertices in $N_2(v) \setminus \{x_4, y_4\}$ are degree-5 vertices (otherwise $g_v \geq 2$). Note that $u_j \neq u_2$, since otherwise (u_2, u_4) would be a good pair. Then $a = u_2$ is an effective vertex with $k(a) = (0, 1, 4)$ and $f_a \geq 13$.

(iii) $(k_4, k_5) = (2, 3)$: Assume $(f_v, g_v) \leq (10, 1)$ (otherwise $(f_v, g_v) \geq (12, 0)$ or $(10, 2)$ and we are done). Since $N^*(v) = \emptyset$, it holds $f_v = 2\delta(v) = 10$ by parity condition, and each neighbor $u_i \in N(v)$ has exactly two neighbors $x_i, y_i \in N_2(v)$. We see that the configuration of $G[N(v)]$, which is either
(type-1): a single path; or
(type-2): a union of an edge and a triangle.
Since $g_v \leq 1$, there is at most one vertex $t \in N_2(v)$ such that $\delta(t) \leq 4$ or t has two neighbors in $N(v)$. We distinguish two subcases on the configuration of $G[N(v)]$.

(1) $G[N(v)]$ is a path $u_1u_2u_3u_4u_5$ from a degree-4 vertex u_1 to the other degree-4 vertex u_5: We show that u_2 or u_4 is an effective vertex. If each of x_4 and y_4 has no other neighbor in $N(v)$ than u_4, then $a = u_4$ is an effective vertex with $k(a) = (0, 1, 4)$ and $f_a \geq 13$ (when $\delta(x_4) = \delta(y_4) = 5$) or $k(a) = (0, 2, 3)$ and $f_a \geq 12$ (when $\delta(x_4)$ or $\delta(y_4)$ is 4).

Now assume that one of x_4 and y_4 has a neighbor $u_j \in N(v)$ other than u_4, and all other vertices in $N_2(v) \setminus \{x_4, y_4\}$ are degree-5 vertices (otherwise $g_v \geq 2$). Note that $u_j \neq u_2$, since otherwise (u_2, u_4) would be a good pair. Then $a = u_2$ is an effective vertex with $k(a) = (0, 1, 4)$ and $f_a \geq 13$.

(2) $G[N(v)]$ is a union of edge u_1u_2 between two degree-4 vertices and a triangle $u_3u_4u_5$ on three degree-5 vertices. Without loss of generality let $x_2 = t$ whenever there is a vertex $t \in N_2(v)$ with $\delta(t) \leq 4$ or $|N(t) \cap N(v)| = 2$. Hence each of x_1 and y_1 is a degree-5 vertex which has only one neighbor in $N(v)$. By the choice of v, we know that $N(x_1)$ contains at most two degree-4 vertices. If $N(x_1)$ contains no other degree-4 vertex than u_1, then x_1 is a degree-5 vertex with $k(x_1) = (0, 1, 4)$, which contradicts our choice of vertex v.

Assume that $N(x_1)$ contains two degree-4 vertices u_1 and t. In this case, u_1 and t cannot be adjacent since y_1 is of degree 5 now, and x_1 is a degree-5 vertex with $k(x_1) = (0, 2, 3)$ such that $G[N(x_1)]$ is not of type-2. This means that $G[N(x_1)]$ is of type-1 or $(f_v, g_v) > (10, 1)$ (i.e., $(f_v, g_v) \geq (12, 0)$ or $(10, 2)$), since $N^*(y) = \emptyset$ for all degree-5 vertices y in G. We are done in the latter. In the former, we can apply the argument in (iii)-(1) to obtain an effective vertex.

We next consider the case where G is 5-regular. Choose a vertex v in G. Let $N(v) = \{u_i \mid 1 \leq i \leq 5\}$. Each $u_i \in N(v)$ has at least two neighbors in $N_2(v)$, which are denoted by x_i and y_i. Also assume that $f_v \leq 12$, since otherwise we are done with $f_v \geq 14$. In this case, $G[N(v)]$ has three possible configurations:
(type-1): $f_u = 10$ and $G[N(u)]$ is a 5-cycle;
(type-2): $f_u = 12$ and $G[N(u)]$ is a path of length 4; and
(type-3): $f_u = 12$ and $G[N(u)]$ is a union of an edge and a triangle.
Note that for $f_u = 12$, $G[N(u)]$ cannot be a union of an isolated vertex and a 4-cycle, since otherwise some nonadjacent degree-5 vertices in the 4-cycle would be a good pair.

We fist claim that G always contains a type-2 degree-5 vertex. If G consists of only type-1

degree-5 vertices, then each pair of adjacent neighbors $u_i, u_{i+1} \in N(v)$ share a common vertex in $N_2(v)$ and this implies that $g_v = 5$ and v is an effective vertex with $(f_v, g_v) = (10, 5)$. If G consists of only type-3 degree-5 vertices, then it is the 5-regular line graph $L(H)$ of a $(3,4)$-bipartite graph H, again contradicting that G is a reduced graph. However, we easily check that a type-1 degree-5 vertex u cannot be adjacent to a type-3 degree-5 vertex u', since u will be of degree 1 or 3 in $G[N(u')]$. Therefore, the 5-regular graph G contains a type-2 degree-5 vertex.

Let v be a type-2 degree-5 vertex, where u_i $(i = 1, 5)$ has three neighbors $x_i, y_i, z_i \in N_2(v)$ and u_i $(i = 2, 3, 4)$ has three neighbors $x_i, y_i, z_i \in N_2(v)$ each of the other three has two neighbors $x_i, y_i \in N_2(v)$. Assume that $g_v \leq 2$ (otherwise we are done). We show that one of vertices u_2, u_3, u_4, x_3 and y_3 will be an effective vertex.

For each $i = 2, 3, 4$, if x_i and y_i are not adjacent or both x_i and y_i have no other neighbor in $N(v)$ than u_i, then we see that $a = u_i$ is an effective vertex with $f_a \geq 14$. Hence assume that, for each $i = 2, 3, 4$, x_i and y_i are adjacent and one of x_i and y_i has a neighbor $u_{i'} \in N(v)$ than u_i, Note that $\{x_2, y_2\} \cap \{x_4, y_4\} = \emptyset$, since otherwise (u_2, u_4) would be a good pair. Now $N_2(v)$ contains exactly two vertices $t, t' \in N_2(v)$ which have two neighbors in $N(v)$ (otherwise $g_v \geq 3$), where $\{x_i, y_i\} \cap \{t, t'\} \neq \emptyset$ for $i \in \{2, 3, 4\}$. Without loss of generality let $t = y_2 = x_3$ and $t' \in \{x_4, y_4\}$. Since u_1 is not adjacent to x_2 or y_2 (otherwise $g_v \geq 3$), we have $f_{u_2} = 12$, where we assume that $g_{u_2} \leq 2$ (otherwise we are done).

Now $g_{u_3} \geq 3$, and we assume that $f_{u_3} \leq 10$ and $g_{u_3} \leq 4$ (otherwise u_3 is an effective vertex). We see that $f_{u_3} \leq 10$ can hold only when $t' = y_3 = x_4$ (recall that x_3 cannot be adjacent to u_4) and that $g_{u_3} \geq 4$ holds only when the last neighbors z_1 and z_2 of x_3 and y_3 must be distinct.

Finally we show that $t = y_2 = x_3$ is an effective vertex. Since $g_t \geq 2$, it suffices to show that $f_t \geq 12$. Note that $g_{u_2} \leq 2$ implies that x_2 and z_1 are not adjacent and u_1 and $y_2 = x_3$ are not adjacent. Also x_2 and $y_3 = x_4$ are not adjacent (otherwise (x_2, u_3) would be a good pair). These indicate $f_t \geq 14$. ∎

8.3 The proof of Lemma 16

Lemma 16 *Let v be an optimal vertex in a proper graph G. Then it holds*

$$\Delta(\overline{N[v]}) \geq \lambda(k_3, k_4, k_5),$$

where

$$\lambda(k_3, k_4, k_5) = \begin{cases} 14\Delta w_5 & \text{if } (k_3, k_4, k_5) = (0, 0, 5) \\ 9\Delta w_5 + 2\Delta w_4 & \text{if } (k_3, k_4, k_5) = (0, 1, 4) \\ 8\Delta w_5 + 2\Delta w_4 & \text{if } (k_3, k_4, k_5) = (0, 2, 3) \\ 9\Delta w_5 + \Delta w_4 & \text{if } (k_3, k_4, k_5) = (0, 4, 1) \\ 11\Delta w_5 & \text{if } (k_3, k_4, k_5) = (0, 3, 2) \\ 10\Delta w_5 & \text{if } (k_3, k_4, k_5) = (0, 5, 0) \\ 2w_3 + 6\Delta w_5 & \text{if } k_3 \geq 2 \text{ and } N(v) \setminus N^*(v) \text{ contains} \\ & \text{no degree-3 neighbor of } v \\ 10\Delta w_5 & \text{if } k_3 = 1 \text{ or } N(v) \setminus N^*(v) \text{ contains} \\ & \text{a degree-3 neighbor of } v. \end{cases}$$

Lemma 16 is proven via the following two lemmas, one for optimal vertices v with $N^*(v) = \emptyset$ and the other for those with $N^*(v) \neq \emptyset$. The first lemma for optimal vertices v with $N^*(v) = \emptyset$ is as follows.

Lemma 19 *Let v be an optimal vertex with $N^*(v) = \emptyset$ in a proper graph G. Then it holds*

$$\Delta(\overline{N[v]}) \geq \lambda(k_3, k_4, k_5),$$

where if $k_3 \geq 1$, then $N(v) \setminus N^*(v)$ contains a degree-3 neighbor of v and $\lambda(k_3, k_4, k_5) = 10\Delta w_5$.

Proof. Since $N^*(v) = \emptyset$, there are at least $2|N(v)| = 10$ edges between $N(v)$ and $N_2(v)$ and then $\Delta(\overline{N[v]}) \geq f_v \Delta w_5 \geq 10\Delta w_5$. Note that when $k_3 \geq 1$, then $N(v) \setminus N^*(v)$ contains a degree-3 neighbor of v and $\lambda(k_3, k_4, k_5)$ is defined to be $10\Delta w_5$. Therefore, the lemma holds for the cases of $k_3 \geq 1$ and $(k_3, k_4, k_5) = (0, 5, 0)$. The other cases are proved by used (3) and the property of optimal vertex (v is an effective vertex since $N^*(v) = \emptyset$). For $k(v) = (0, 0, 5)$, it holds $(f_v, g_v) \geq (14, 0)$, $(12, 3)$ or $(10, 5)$ by the lower bounds on f_v and g_v in the definition of effective vertices, and we have $\min\{14\Delta w_5, 12\Delta w_5 + 3(\Delta w_4 - \Delta w_5), 10\Delta w_5 + 5(\Delta w_4 - \Delta w_5)\} = 14\Delta w_5 = \lambda(0, 0, 5)$ by (7). The other cases can be treated analogously. ∎

The second lemma for optimal vertices v with $N^*(v) \neq \emptyset$ is as follows. We easily see by (7) and (8) that $\lambda(0, 0, 5) = 14\Delta w_5 \leq w_4 + 2w_3 + 4\Delta w_5$, $\lambda(0, 1, 4) = 9\Delta w_5 + 2\Delta w_4 \leq 3w_3 + 4\Delta w_5$, $\lambda(0, 5, 0) = 10\Delta w_5 \leq 2w_3 + 5\Delta w_5$ and $\max\{9\Delta w_5 + \Delta w_4, 8\Delta w_5 + 2\Delta w_4, 2w_3 + 6\Delta w_5, 11\Delta w_5\} \leq 2w_3 + 6\Delta w_5$. Hence the next lemma and Lemma 19 imply Lemma 16.

Lemma 20 *Let v be an optimal vertex with $N^*(v) \neq \emptyset$ in a proper graph G. Then it holds*

$$\Delta(\overline{N[v]}) \geq \begin{cases} w_4 + 2w_3 + 4\Delta w_5 & \text{if } k(v) = (0, 0, 5) \\ 3w_3 + 4\Delta w_5 & \text{if } k(v) = (0, 1, 4) \\ 2w_3 + 5\Delta w_5 & \text{if } k(v) = (0, 5, 0),\ v \text{ has exactly one degree-3} \\ & \text{neighbor (i.e., } k_3 = 1), \text{ or } N(v) \setminus N^*(v) \\ & \text{contains a degree-3 neighbor of } v \\ 2w_3 + 6\Delta w_5 & \text{otherwise.} \end{cases}$$

Proof. Without loss of generality let $N(v) = \{u_1, u_2, \ldots, u_5\}$, $N_2(v) = \{t_1, t_2, \ldots, t_r\}$ ($r \geq |N(\overline{N_2[v]})| \geq 4$), and $u_1 \in N^*(v)$ and $t_1 \in N(u_1) \cap N_2(v)$. For each vertex $t \in N_2(v)$, let $e(t)$ be the number of neighbors of t in $N(v)$, where $e(t) \leq 2$ since otherwise (v, t) would be a good pair.

For $X = N[I_v]$, it holds $|N(X)| \geq 4$ since there is no good vertex cut, where $N(X) \subseteq \overline{N[v]}$. Hence if $|N[I_v] \setminus N[v]| \geq 3$, then we have

$$\Delta(\overline{N[v]}) \geq \sum_{x \in N[I_v] \setminus N[v]} w_{\delta(x)} + \sum_{y \in N(X)} \Delta w_{\delta(y)},$$

which is at least $w_4 + 2w_3 + 4\Delta w_5$ for $k(v) = (0, 0, 5)$ (note that $t_1 \in N[I_v] \setminus N[v]$ is a vertex adjacent to a degree-5 vertex u_1 and then $\delta(t_1) \geq 4$ by the choice of the optimal vertex v) and $3w_3 + 4\Delta w_5$ ($\geq 2w_3 + 6\Delta w_5$ by (8)) otherwise.

Then assume that $|N[I_v] \setminus N[v]| \leq 2$. In this case $\{t_1\} = I_v - \{v\}$, since $|I_v - \{v\}| = 2$ means that each vertex $t \in I_v - \{v\}$ has a neighbor in $V - N(v) - I_v$ (recall that $e(t) \leq 2$ and I_v is an independent set), implying a contradiction that $|N[I_v] \setminus N[v]| > 2$. Also t_1 is adjacent to two neighbors $u_1, u_2 \in N(v)$ and $\delta(t_1) = 3$ (otherwise $|N[I_v] \setminus N[v]| \geq 3$). Hence $k(v) \notin \{(0, 1, 4), (0, 0, 5)\}$, since otherwise u_1 or u_2 would be a degree-5 vertex adjacent to degree-3 neighbor t_1 and such a vertex u_i should have been chosen as an optimal vertex instead of v). Let u_1 and u_2 be the neighbors of t_1 in $N(v)$. We show that none of u_1 and u_2 is a degree-3 vertex in $N^*(v)$. If one of them, say $u_1 \in N^*(v)$ is a degree-3 vertex, then t_1-u_1-$\{v, u^*\}$ is a short funnel for the third neighbor $u^* \in N(v)$ of u_1, since t_1 is a degree-3 vertex and there is an edge $u_2 v$ between $N(t_1) \setminus \{u_1\}$ and $\{v, u^*\}$. Then, no degree-3 vertex is in $N^*(v)$. Hence $|I_v - \{v\}| = 1$ implies that if $k_3 \geq 1$ then $N(v) \setminus N^*(v)$ always contains a degree-3 neighbor of v. This indicates that we do not need to show $\Delta(\overline{N[v]}) \geq 2w_3 + 6\Delta w_5$ for the case where $k_3 \geq 2$ and $N(v) \setminus N^*(v)$ contains no degree-3 neighbor of v. So we only have to show that

$$\Delta(\overline{N[v]}) \geq 2w_3 + 5\Delta w_5 \text{ for the case where } k(v) = (0, 5, 0) \text{ or } k_3 \geq 1,$$

and

$$\Delta(\overline{N[v]}) \geq 2w_3 + 6\Delta w_5 \text{ otherwise } (k(v) \notin \{(0,5,0),(0,1,4),(0,0,5)\}).$$

Then each $u_i \in N(v)$ with $i = 3,4,5$ has at least two neighbors in $N_2(v)$, since otherwise u_i would be adjacent to another vertex in $I_v - \{v\}$ (note that t_1 can be adjacent to only u_1 and u_2 since $e(t_1) = 2$).

Let x be the third neighbor of t_1.

At least six edges between $\{u_3, u_4, u_5\}$ and $N_2(v) \setminus \{t_1\}$ are removed in $G - N[I_v]$.

(i) $x \notin N_2(v)$: In this case, we have

$$\Delta(\overline{N[v]}) \geq w_{\delta(t_1)} + w_{\delta(x)} + 6\Delta w_5 \geq 2w_3 + 6\Delta w_5,$$

as required.

(ii) $x \in N(\overline{N_2[v]})$: Analogously with (i), the degree of x_1 decreases in $G - N[I_v]$, where $x_1 \in (\overline{N_2[v]})$ is a neighbor of x and we have $\Delta(\overline{N[v]}) \geq w_{\delta(t_1)} + w_{\delta(x)} + \Delta w_{\delta(x_1)} + 5\Delta w_5$ if $e(x) \leq 1$; or $\Delta(\overline{N[v]}) \geq w_{\delta(t_1)} + w_{\delta(x)} + \Delta w_{\delta(x_1)} + 4\Delta w_5$ if $e(x) = 2$ (where $\delta(x) \geq 4$). In any case, we obtain

$$\Delta(\overline{N[v]}) \geq 2w_3 + 6\Delta w_5.$$

Now assume that $x \in N_2(v) \setminus N(\overline{N_2[v]})$.

(iii) $x = t_2 \in N_2(v) \setminus N(\overline{N_2[v]})$ and t_2 has a neighbor $t_3 \in N_2(v)$: Now when $\delta(t_2) \leq 4 - \epsilon$ ($\epsilon = 0, 1$), there are at least $4 + \epsilon$ edges between $\{u_3, u_4, u_5\}$ and $N_2(v) \setminus \{t_1, t_2\}$. Note that t_3 may receive two of these edges, and in this case, $\delta(t_3) \geq 4$ or $|N_2(v) \setminus \{t_1, t_2, t_3\}| \geq |N(\overline{N_2[v]})| \geq 4$ holds. In any case, the weight decrease in $N_2(v) \setminus \{t_1, t_2\}$ is at least $(6 - \epsilon)\Delta w_5$. By $w_{\delta(t_2)} \geq w_3 + \epsilon\Delta w_5$, we have

$$\Delta(\overline{N[v]}) \geq w_3 + w_{\delta(t_2)} + (6 - \epsilon)\Delta w_5 \geq w_3 + w_3 + 6\Delta w_5.$$

(iv) $x = t_2 \in N_2(v) \setminus N(\overline{N_2[v]})$ and t_2 has no neighbor in $N_2(v)$: In this case, $\delta(t_2) = 3$ since $e(t_2) \leq 2$. Let h be the number of edges between $\{u_2, u_3, u_4, u_5\}$ and $N_2(v) \setminus \{t_1, t_2\}$, where $h \geq 4$, since there are six edges between $\{u_3, u_4, u_5\}$ and $N_2(v) \setminus \{t_1\}$ and t_2 can receive at most two of them. We have

$$\Delta(\overline{N[v]}) \geq w_3 + w_3 + h\Delta w_5.$$

Since we are done if $h \geq 6$, we assume $h \leq 5$.

If $h \leq 4$, then u_2 is not adjacent to any vertex in $N_2 \setminus \{t_1, t_2\}$, then $Z = \{u_3, u_4, u_5\}$ is a vertex cut that separates $X = \{v, u_1, u_2, t_1, t_2\}$ from G, where $|X| \leq 24$ and $\delta(X) \geq 17$, contradicting that G has no good vertex cut. Therefore $h = 5$ holds, and in this case, u_2 is adjacent to t_1 and some vertex $t_j \in N_2(v)$ with $j \geq 3$, and t_2 is adjacent to two vertices in $\{u_3, u_4, u_5\}$, say u_3 and u_4.

Since $h = 5$ holds, we have obtained $\Delta(\overline{N[v]}) \geq 2w_3 + 5\Delta w_5$ for $k(v) = (0,5,0)$ and $k_3 \geq 1$.

Next, we assume that $k_3 = 0$ and $k(v) \neq (0,5,0)$. In this case, no degree-5 neighbor of v can be adjacent to degree-3 vertex t_1 or t_2 by the choice of optimal vertex v, and we have $\delta(u_1) = \delta(u_2) = \delta(u_3) = \delta(u_4) = 4$. We see that for $k(v) = (0,4,1)$, it cannot hold $f_v = 4 + h = 9$ due to the parity condition, a contradiction. ∎