Further Improvement on Maximum Independent Set in Graphs with Maximum Degree 4

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

We present a simple algorithm for the maximum independent set problem in an n-vertex graph with degree bounded by 4, which runs in $O^*(1.1446^n)$ time and improves all previous algorithms for this problem. The algorithm is analyzed by using the "Measure and Conquer" method. We use some good reduction and branching rules with a new idea on setting weights to obtain the improved time bound without increasing the number of branching rules in the algorithm.

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

1 Introduction

The *maximum independent set* problem (MIS), to find a maximum set of vertices in a graph such that there is no edge between any two vertices in the set, is not only a basic problem introduced in Garey and Johnson's work [10] on NP-completeness, but also one of the most important problems in the line of research on worst-case analysis of algorithms for NP-hard optimization problems. Since Tarjan and Trojanowski [16] published the first nontrivial $O^*(2^{n/3})$-time algorithm in 1977, the bound of the running time to exactly solve the problem has been improved frequently [11, 14, 15, 7, 12, 3]. One of the most important results among them is that due to Fomin *et al.* [7], in which they use a new method called "Measure and Conquer" to analyze simple algorithms. By using this method together with other techniques, recently Kneis *et al.* [12] and Bourgeois *et al.* [3] improved the running time bound to $O^*(1.2132^n)$ and $O^*(1.2127^n)$, respectively. Up to till now, these are the best published results for MIS.

One of the most important subcases to solve MIS is the problem in low-degree graphs. Since we can simply branch on a high-degree vertex by including it into the independent set or excluding it from the independent set, and then reduce the graph greatly in the subbranches, sometimes the problem in degree-i graphs (graphs with maximum degree i) for small i will become the bottleneck for solving the problem in general graphs. Bourgeois *et al.* [3] also used a bottom-up method to get improvements on MIS. In this method, the running time bounds for MIS in degree-3 graphs (MIS3) and MIS in degree-4 graphs (MIS4) will directly affect the algorithms and running time bounds for the problem in other degree bounded graphs and general graphs. Motivated by these, researchers have great interests in designing fast algorithms for MIS3 and MIS4. For MIS3, we quote the $O^*(1.1259^n)$-time algorithm by Beigel [1], the $O^*(1.1225^n)$-time algorithm by Fomin and Høie [8], the $O^*(1.1120^n)$-time algorithm by Fürer [9], the $O^*(1.1034^n)$-time algorithm by Xiao *et al.* [19], the $O^*(1.0892^n)$-time algorithm by Razgon [13], the $O^*(1.0885^n)$-time algorithm by Xiao [18], the $O^*(1.0854^n)$-time algorithm by Bourgeois *et al.* [2], and and finally the recent $O^*(1.0836^n)$-time algorithm by Xiao and Nagamochi [20]. For MIS4, MIS in degree-5 graphs (MIS5) and MIS in

[1]Technical report 2012-003, May 8, 2012

degree-6 graphs (MIS6), the best previous results are $O^*(1.1571^n)$, $O^*(1.1918^n)$ and $O^*(1.2071^n)$, respectively [3], which are designed by using a bottom-up method based on a fast algorithm for MIS3. In this paper, we will design a simple $O^*(1.1446^n)$-time algorithm for MIS4 by using "Measure and Conquer," which improves the previous best bound $O^*(1.1571^n)$ on MIS4.

Most fast exponential-time algorithms are based on a branch-and-reduce paradigm, which contains two main steps. We first check whether we can get partial solution and reduce the current problem directly according the *reduction rules*, and then branch the problem instance into several smaller instances according to the *branching rules*. To scale the size of the instance, we may need to use a parameter, such as the number of vertices or edges for graph problems, as a measure of the size of the instance. By bounding the size of the search tree to a function of the parameter, we will get a running time bound relating to the parameter for the problem. For MIS, we branch on the current graph G into several graphs $G_1, G_2, \ldots, G_l$ such that the parameter w_i of each graph G_i is less than the parameter w of graph G, and a maximum independent set in G can be found in polynomial time if a maximum independent set in each of the l graphs $G_1, G_2, \ldots, G_l$ is known. Usually, G_i $(i = 1, 2, \ldots, l)$ are obtained by deleting some vertices in G. We can build up a search tree according to our branching rules, and the exponential part of the running time of the algorithm corresponds to the size of the search tree. The running time analysis leads to a linear recurrence for each node in the search tree that can be solved by using standard techniques. By letting $C(w)$ denote the worst-case size of the search tree when the parameter of graph G is w, we get the recurrence relation $C(w) \leq \sum_{i=1}^{l} C(w - w'_i)$, where $w - w'_i = w_i$. Solving the recurrence, we get $C(w) = [\alpha(w'_1, w'_2, \ldots, w'_l)]^w$, where $\alpha(w'_1, w'_2, \ldots, w'_l)$ is the largest root of the function $f(x) = 1 - \sum_{i=1}^{l} x^{-w'_i}$. As for the measure (the parameter w), we should guarantee that when parameter $w \leq 0$ the problem can be solved directly and the parameter will not increase in each step (applying reduction rules or branching rules). A natural measure is the number of vertices or edges in the graph. Note that to get fast algorithms by this method, we hope that the decrease w'_i $(i = 1, 2, \ldots, l)$ in the above recurrence are as large as possible. To get large values of them, some proposed algorithms check a large number of local structures of the graph and get numerous branching rules. However, the "Measure and Conquer" method tries to improve the recurrences in another way. In this method, we set a weight to each vertex in the graph according to the degree of the vertex (usually vertices of the same degree receive the same weight) and use the sum of the weights in the graph as the measure. Note that when a vertex v is deleted, we may decrease the measure not only from v but also from the neighbors of v (the degrees of the neighbors will decrease by 1). Compared to traditional measures, the weighted measure may catch more structural information of the graph and may yield a further improvement without modifying the algorithms. Currently, the best exact algorithms for many NP-hard problems are designed by this method. However, we should choose a good weight setting on vertices, which is an important step in this method. To do this, we may need to solve a quasiconvex program. In this paper we also use the branch-and-reduce paradigm and the "Measure and Conquer" method to design our algorithm.

The rest of the paper is organized as follows. Section 2 gives the notation that may be used in the paper. Sections 3 and 4 introduce the reduction rules and branching rules, respectively. Section 5 presents our simple algorithm. Section 6 analyzes the running time bound. Finally Section 7 makes some concluding remarks.

2 Notation System

Given a graph $G = (V, E)$, the total number of vertices in the graph is denoted by n. For a vertex v in a graph, $d(v)$ is the degree of v, $N(v)$ the set of all neighbors of v, and $N_2(v)$ the set of vertices with distance exactly 2 from v. Denote $N[v] = N(v) \cup \{v\}$ and $N_2[v] = N_2(v) \cup N[v]$. We may also use $N(V')$ to denote the neighbors of a set V' of vertices, i.e., $N(V') = \cup_{v \in V'} N(v) - V'$. For $k \geq 3$ vertices $v_1, v_2, \ldots, v_k$ in G, we say that $v_1 v_2 \cdots v_k$ is a *k-cycle* in G if for each $i = 1, 2, \ldots, k$

v_i and v_{i+1} are adjacent, where we interpret $v_{k+1} = v_1$. A *line graph* of graph G is the graph whose vertices are corresponding to the edges of G, and two vertices are adjacent if and only if the corresponding two edges sharing a same endpoint in G. In our algorithm, when we remove a set of vertices, we also remove all the edges that are incident on it. 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 $f(n) = g(n)\text{poly}(n)$, where $\text{poly}(n)$ is a polynomial in n.

3 Reduction Rules

Reduction rules are used to decrease the size of instances of the problem directly before applying the branching rules. Reduction rules will not exponentially increase the size of our search tree. Furthermore, the reduction operations will reduce some special local structures of the graph, and then the branching rules can apply effectively in the resulted graphs. Some of the reduction rules used here are well-known in the literature. Some of them are newly introduced.

Let $\eta(G)$ denote the size of a maximum independent set of graph G. Clearly if there are degree-0 vertices, then any maximum independent set includes all these vertices and we remove them from the graph. Thus for the set V_0 of degree-0 vertices in G, $\eta(G) = \eta(G - V_0) + |V_0|$.

Dominance

We say that a vertex u dominates another vertex v if $N[u] \subseteq N[v]$. If there is a vertex v dominated by another vertex, we remove v from the graph.

We say that a vertex is a *dominated vertex* if it is dominated by another vertex. It is easy to see that we can remove a dominated vertex without losing a solution.

Lemma 1 *For a dominated vertex v in graph G,*

$$\eta(G) = \eta(G - \{v\}).$$

Next we assume that the above rule has been applied in the graph and then there is no dominated vertex. Note that the neighbor of any degree-1 vertex is a dominated vertex. If there is a triangle with a degree-2 vertex v in it, then the other two vertices in the triangle are dominated by v. When a graph has no dominated vertex, it will have neither degree-1 vertex nor a degree-2 vertex with two adjacent neighbors.

Roofs

A vertex v is called a roof, if it is in a 5-cycle $vabcd$ such that a and d are two adjacent degree-3 vertices. If there is a roof, we remove it from the graph.

Lemma 2 *Let v be a roof in graph G, then*

$$\eta(G) = \eta(G - \{v\}).$$

Proof. We only need to prove that there is a maximum independent set that does not contain v. Assume that v is in a maximum independent set S where $a, d \notin S$. Since there is an edge between b and c, at least one of b and c, say b without loss of generality, is not in S. Therefore, we can replace v with a in S to get another maximum independent set that does not contain v. ∎

Folding degree-2 vertices

Folding a degree-2 vertex v (with two nonadjacent neighbors a and b) means contracting v, a and b into a single vertex s.

Fig. 1 illustrates the operation of folding a degree-2 vertex.

A vertex v together with its neighbors $N(v)$ is called a *funnel* (or a $d(v)$-funnel) if $N[v] - \{a\}$ induces a complete graph for some $a \in N(a)$, and is denoted by a-v-$(N(v) - \{a\})$. In particular,

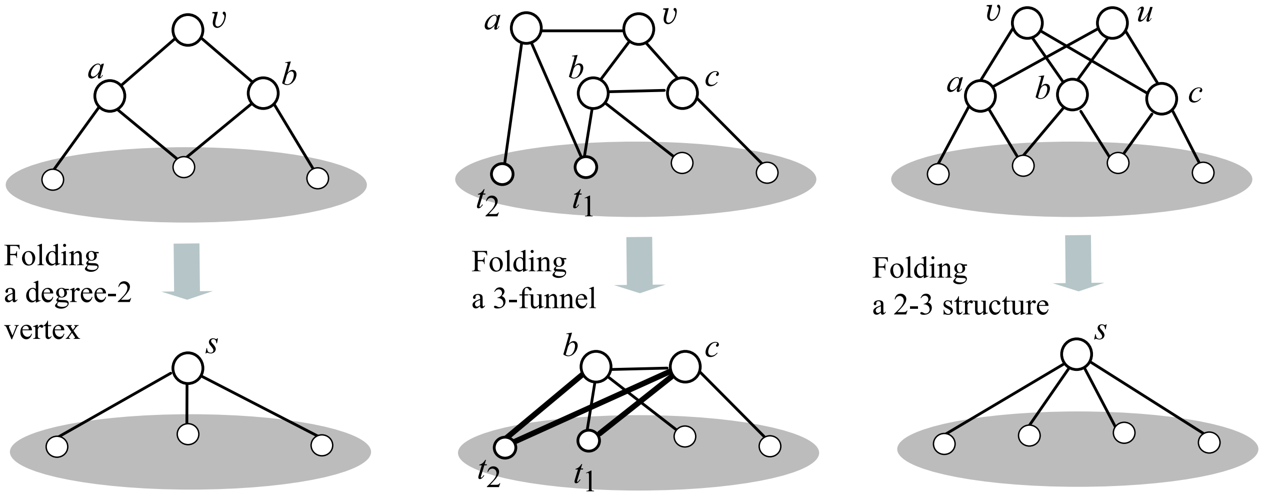

Figure 1: Illustrations of folding operations

for a degree-3 vertex v with $N(v) = \{a, b, c\}$ such that b and c are adjacent, the *3-funnel* is denoted by a-v-$\{b, c\}$, where $N(a) \cap N(v) = \emptyset$ holds when v dominates none of its neighbors. Note that 3-funnels are also called bottles in [18]. For a 3-funnel, we have the following reduction rule.

Folding funnels
In the operation of folding a 3-funnel a-v-$\{b, c\}$, we add edges between $N(a)$ and $\{b, c\}$ and remove $\{a, v\}$.

Fig. 1 illustrates the operation of folding a funnel. Let $G^\star(v)$ denote the graph after folding a degree-2 vertex v or a 3-funnel a-v-$\{b, c\}$ in G. Then we have the following lemma.

Lemma 3 *For a degree-2 vertex v or a funnel a-v-$\{b, c\}$ in graph G, we have $\eta(G) = 1 + \eta(G^\star(v))$.*

The correctness of the reduction rules has been discussed in many references [4, 7]. In fact, folding a funnel is a special case of a reduction rule introduced in [7]. We gave the new name of the local structure just for the convenience of the analysis. In general, folding a short funnel may increase our measure (defined in Section 6), which is unexpected in our algorithm. We call a 3-funnel a-v-$\{b, c\}$ in a graph with minimum degree 3 a *short funnel* if there are at least $d(a) - 2$ edges between $N(a) - \{v\}$ and $\{b, c\}$. In our algorithm, we will reduce short funnels only and leave some other funnels.

In our algorithm, we will also use the following reduction rules. If two independent degree-3 vertices v and u have three common neighbors a, b and c, then we say that the five vertices compose a 2-3 *structure* (see Fig. 1), and denote it by $\{v, u\}$-$\{a, b, c\}$. If there are three independent vertices v, u and z of degree ≥ 3 such that $|N(v) \cup N(u) \cup N(z)| = 4$, then we say that the seven vertices compose a 3-4 *structure* and denote it by $\{v, u, z\}$-$N(v) \cup N(u) \cup N(z)$.

Folding 2-3 and 3-4 structures
Let A-B be a 2-3 structure or 3-4 structure in graph G. Folding A-B means
(a) removing $A \cup B$ from the graph, when B is not an independent set in G; and
(b) contracting $A \cup B$ into a singe vertex and deleting parallel edges and self-loops from the graph, when B is an independent set in G.

Lemma 4 *If graph G has a k-$(k + 1)$ structure ($k = 2$ or 3), then $\eta(G) = k + \eta(G^\star)$, where $G^\star$ is the graph obtained from G by folding the k-$(k + 1)$ structure in G.*

The above reduction rule is a special case of the crown reduction introduced in [5]. The correctness of folding an A-B structure follows from this observation: When B is not an independent set, there is a maximum independent set that contains A. When B is an independent set, there is a maximum independent set that contains either B or A.

Line graphs
If graph G is a line graph of graph G', we find a maximum independent set of G directly by finding a maximum matching in G' and taking the corresponding vertex set in G as the solution.

Not every graph is a line graph. There are several good methods to check whether a graph is a line graph or not, which depend on characterizations of line graphs [17]. In this paper, we only need to check whether a graph is a line graph of a 3-regular graph, which can be easily done (note that a graph is a line graph of a 3-regular graph if and only if the graph has only degree-4 vertices and each of them is contained in two edge-disjoint triangles).

Bipartite graphs
If graph G is a bipartite graph, we find a maximum independent set of G in polynomial time by Hungarian Algorithm.

Definition 5 *A graph is called a* reduced graph*, if it contains none of dominated vertices, roofs, degree-2 vertices, short funnels, 2-3 structures, and 3-4 structures, and has no connected component which is a line graph of a 3-regular graph or a bipartite graph.*

4 Branching Rules

In the algorithm, we may branch on a vertex v of maximum degree by including it into the independent set or excluding it from the independent set. That is, in the first branch we will delete $N[v]$ from the graph and in the second branch we will delete v from the graph. Besides this branching rule, we also use two other branching rules, *branching on a funnel* and *branching on a 4-cycle*, introduced in [18].

We have introduced a reduction rule that can reduce 3-funnels. However, we just use the rule to reduce short funnels. For other kinds of funnels, we may use some branching rules to deal with them. A vertex $s \in N_2(u)$ is called a *satellite* of vertex u if there is a vertex $s' \in N(u)$ such that $N[s'] - N[u] = \{s\}$, where s' is also called the *parent* of satellite s. The concept of satellites is introduced in [12] and extended in [20]. We use S_u to denote the set of vertex u and all of its satellites.

Lemma 6 *Let a-v-$(N(v) - \{a\})$ be a funnel in graph G. Then there is a maximum independent set S in G such that either $v \in S$ or $S_a \subseteq S$.*

Proof. Let $k = d(v)$. First consider the case where G has a maximum independent set that does not contain a. Then we can directly remove a from G. In the remaining graph v becomes a degree-$(k-1)$ vertex in a clique of size k. In this case, there is a maximum independent set that contains v. Next consider the case where every maximum independent set S contains a. We show that S also contains any satellite s of a. Note that only $N[a] \cap S = \{a\}$. If s is not in S, then $N[w'] \cap S = \{a\}$ holds for the parent s' of s. We can replace a with s' in S to get another maximum independent set S that does not contain a, which is a contradiction. All satellites of a should be in S. ∎

Based on Lemma 6, we get the following branching rule.

Branching on a funnel
Branching on a funnel a-v-$(N(v) - \{a\})$ means branching by either including v in the independent set or including S_a in the independent set.

Lemma 7 *Let abcd be a 4-cycle in graph G. For any independent set S in G, either $a, c \notin S$ or $b, d \notin S$.*

Proof. Since any independent set contains at most two vertices in a 4-cycle and the two vertices cannot be adjacent, we know the lemma holds. ∎

Based on Lemma 7, we get the following branching rule.

Branching on a 4-cycle
Branching on a 4-cycle abcd means branching by either excluding a and c from the independent set or excluding b and d from the independent set.

5 The Algorithm for MIS4

The main idea of the algorithm is that: If the graph is not a reduced graph, we apply our reduction rules to reduce it; Elseif there is a vertex of degree ≥ 5, we branch on it; Elseif there is a 4-funnel or a triangle containing both degree-3 and degree-4 vertices (implying that there is always a 3-funnel), we branch on a 'good' funnel; Elseif there is a 4-cycle containing a degree-4 vertices, we branch on a 4-cycle; Elseif the graph still has a degree-4 vertex, we branch on a 'good' degree-4 vertex; Last, the graph is a degree-3 graph and we use a fast algorithm for MIS3 to solve it. Before presenting our algorithm, we give the definitions of 'good' funnels/degree-4 vertices.

A *good funnel* is defined to be one of the following: (i) a 4-funnel; (ii) a 3-funnel a-v-$\{b, c\}$ such that $d(a) = 4$ and at least one of b and c is of degree 4; and (iii) a 3-funnel a-v-$\{b, c\}$ such that the graph has no funnel in (i) or (ii) and at least one of b and c is of degree 4. Also a degree-4 vertex is called a *good degree-4 vertex* if it is not contained in two edge-disjoint triangles. Our algorithm for MIS4 is described in Figure 2.

6 The Analysis

To analyze the running time bound of the algorithm, 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 $i \geq 0$. Then we adopt $w = \sum_i w_i n_i$ as the measure of the graph, where n_i denotes the number vertices of degree i in the graph. We set $w_i > 0$ for $i \geq 3$ and $w_0 = w_1 = w_2 = 0$. Then when measure w is 0, the problem can be solved in polynomial time, since the graph with $w = 0$ 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

$$0 \leq w_3, w_4 \leq 1 \tag{1}$$

so that a given degree-4 graph satisfies $0 \leq w \leq n$. During our algorithm, a vertex with degree greater than 4 may be created by some reduction rules, but we set weights w_i, $i \geq 3$ so that the entire weight w never increases after any operation of reduction/branching rules. This is necessary to evaluate the time bound of our algorithm by analyzing how many instances will be generated until the measure becomes 0. We allow weight w_j, $j > 4$ to be larger than 1 as long as the entire weight w never increases.

6.1 Setting weights

As observed, we have set weights $w_0 = w_1 = w_2 = 0$. For the weight of a vertex of degree ≥ 7, we also set

$$w_i = w_6 + (i - 6)(w_4 - w_3), \quad i \geq 7.$$

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

1. **If** {The graph has a component P that has at most 15 vertices, or is the line graph of a 3-regular graph or a bipartite graph}, **return** $t + MIS4(G - P)$, where t is the size of a maximum independent set in P.
2. **Elseif** {$\exists v, u \in V$: $N[u] \subseteq N[v]$ or there is a roof v}, **return** $MIS4(G - \{v\})$.
3. **Elseif** {There is a degree-2 vertex v or a short funnel a-v-$(N(v) - \{a\})$, **return** $1 + MIS4(G^\star(v))$.
4. **Elseif** {There is a k-$(k+1)$ structure ($k = 2$ or 3)}, **return** $k + MIS4(G^\star)$.
5. **Elseif** {There is a vertex of degree ≥ 5}, pick up a vertex v of maximum degree, and **return** $\max\{MIS4(G - \{v\}), 1 + MIS4(G - N[v])\}$.
6. **Elseif** {There is a good funnel a-v-$(N(v) - \{a\})$, **return** $\max\{1 + MIS4(G - N[S_a]), 1 + MIS4(G - N[v])\}$.
7. **Elseif**{There is a 4-cycle $abcd$ that contains a degree-4 vertex}, **return** $\max\{MIS4(G - \{a, c\}), MIS4(G - \{b, d\})\}$.
8. **Elseif** {There are two adjacent degree-4 vertices}, pick up a good degree-4 vertex v that is adjacent to at least one degree-4 vertex, and **return** $\max\{MIS4(G - \{v\}), 1 + MIS4(G - N[v])\}$.
9. **Elseif** {There are still some degree-4 vertices}, pick up a degree-4 vertex v such that the number of degree-3 vertices in $N_2(v)$ is maximized, and **return** $\max\{MIS4(G - \{v\}), 1 + MIS4(G - N[v])\}$.
10. **Else** {The graph is a 3-regular graph}, we use a fast algorithm for MIS3 to solve the problem and return a solution.

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

Figure 2: The Algorithm $MIS4(G)$

We only need to assign the value to w_3, w_4, w_5 and w_6, which will decide the value of w_i for all other i's.

We here introduce several conditions on weights w_3, w_4, w_5 and w_6, where some conditions are necessary ones for analysis using measure while the other simplify our analysis. We will use Δw_i to denote $w_i - w_{i-1}$ for $i \geq 1$. To simplify our analysis, we assume that

$$0 \leq \Delta w_4 \leq \Delta w_i \leq \Delta w_3, \quad i \geq 5, \tag{2}$$

$$w_3 + w_4 \geq 4 \max\{\Delta w_5, \Delta w_6\} - \Delta w_4. \tag{3}$$

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

$$w_i \geq (i - 2)(w_4 - w_3), \quad i \geq 0.$$

To obtain the next lemma, we assume that

$$w_i + w_j \geq w_{i+j-2}, \quad 3 \leq i, j \leq 5. \tag{4}$$

Lemma 8 *$w_i + w_j \geq w_{i+j-2}$ holds for all $i, j \geq 1$*

Proof. If one of i and j, say i is at most 2, then $w_i + w_j = w_i \geq w_{i+j-2}$. Let $i, j \geq 3$. For $3 \leq i, j \leq 5$, we have $w_i + w_j \geq w_{i+j-2}$ by (4). Finally consider the case when at least one of i and j, say i, is greater than 5. Then we have that $w_{i+j-2} = w_i + (j-2)(w_4 - w_3)$ by the definition of w_i ($i \geq 7$). Since $w_j \geq (j-2)(w_4 - w_3)$, this implies $w_{i+j-2} = w_i + (j-2)(w_4 - w_3) \leq w_i + w_j$. ∎

Lemma 9 *The measure w of a graph will not increase, if we apply the reduction rules of folding a degree-2 vertex, a short funnel, a 2-3 structure or a 3-4 structure, or removing a dominated vertex, a roof, a connected component of a line graph of a 3-regular graph or a bipartite graph.*

Proof. In the operation of removing a dominated vertex, a roof, a connected component of a line graph of a 3-regular graph or a bipartite graph, we just delete some vertices from the graph, and this never increase the total weight w of the graph.

To fold a degree-2 vertex, we delete a degree-2 vertex v and contract the two nonadjacent neighbors a and b of v into a new vertex s, i.e., we delete two vertices of degree x and y from the graph and introduce a new vertex of degree at most $x + y - 2$. By Lemma 8, we have $w_x + w_y \geq w_{x+y-2}$, and the measure w does not increase.

To fold a 2-3 structure of Case (*b*), we delete vertices $\{v, u, a, b, c\}$ with $d(v) = d(u) = 3$ and introduce a new vertex s with $d(s) \leq d(a) + d(b) + d(c) - 6$. By Lemma 8, we have that $w_{d(s)} \leq w_{d(a)+d(b)+d(c)-6} \leq w_{d(a)+d(b)-4} + w_{d(c)} \leq w_{d(a)-2} + w_{d(b)} + w_{d(c)} \leq w_{d(a)} + w_{d(b)} + w_{d(c)}$. For folding a 3-4 structure, we can prove it in the same way.

To fold a short funnel a-v-$\{b, c\}$, we delete vertices $\{v, a\}$ with $d(v) = 3$. Let $d'(u)$ denote the degree of a vertex u after folding the funnel. Then it suffices to show that $\delta = [w_{d'(b)} + w_{d'(c)} + \sum_{t \in N(a)-\{v\}} d'(t)] - [w_{d(b)} + w_{d(c)} + \sum_{t \in N(a)-\{v\}} d(t)]$ is at most $w_3 + w_{d(a)}$. First consider the case of $d(a) = 3$. Let $N(a) = \{v, t_1, t_2\}$, where t_1 is adjacent to b or c, say b. Then $d'(b) \leq d(b)$, $d'(c) \leq d(c) + 1$, $d'(t_1) = d(t_1)$ and $d'(t_2) \leq d(t_2) + 1$ (see Fig. 1). Since the minimum degree in a reduced graph is 3 and $\Delta w_i \leq w_3$, $i \geq 3$ holds by (2), we have $\delta \leq 2w_3$. Next let $d(a) = 4$ and Let $N(a) = \{v, t_1, t_2, t_3\}$, where there are at least two edges between $\{t_1, t_2, t_3\}$ and $\{b, c\}$. We only consider the case where one of t_1, t_2 and t_3, say t_1, is adjacent to both b and c, and show that $\delta \leq 4 \max\{\Delta w_5, \Delta w_6\} - \Delta w_4$ holds in this case (we can show that $\delta \leq 3 \max\{\Delta w_5, \Delta w_6\}$ holds for the other case). Then $d'(b) \leq d(b) + 1$, $d'(c) \leq d(c) + 1$, $d'(t_1) = d(t_1) - 1$, $d'(t_2) \leq d(t_2) + 1$ and $d'(t_3) \leq d(t_3) + 1$. Therefore $\delta \leq 4 \max\{\Delta w_5, \Delta w_6\} - \Delta w_4$ (recall that $\Delta w_i = \Delta w_4 \leq \Delta w_j$ for $i \geq 7$ and $j = 5, 6$). By (3), we have $\delta \leq 4 \max\{\Delta w_5, \Delta w_6\} - \Delta w_4 \leq w_3 + w_4$, as required.

We have considered all cases and then finished the proof. ∎

Note that folding non-short funnels may increase w, which is not contained in our reduction rules.

To evaluate the weight decrease in folding degree-2 vertices, we define

$$\beta = \min_{3 \leq i,j \leq 4} w_i + w_j - w_{i+j-2}.$$

Lemma 10 *Let v be a degree-2 vertex with two nonadjacent neighbors a and b such that $3 \leq d(a), d(b) \leq 4$. Then folding a degree-2 vertex v decreases the measure w by at least β.*

Proof. Contracting a and b into a new vertex decreases w by at least $w_{d(a)} + w_{d(b)} - w_{d(a)+d(b)-2} \geq \beta$. ∎

To simplify our analysis, we further assume that

$$w_3 \le 2(w_4 - w_3). \tag{5}$$

Recall that $w_0 = w_1 = w_2 = 0$, $\Delta w_i \ge w_4 - w_3$, $3 \le i \le 4$, and $w_3 \le 2(w_4 - w_3)$ by (5). Let X be a subset of vertices in a reduce graph $G = (V, E)$ and p be the number of edges between X and $V - X$. When we remove X, the total weight in the remaining set $V - X$ decreases by at least $kw_3 + (w_4 - w_3)\delta$ for the integers k with $p = 3k + i$ ($i \in \{-1, 0, 1\}$) and δ such that $\delta = 1$ when $i = 1$, or $\delta = 0$ otherwise. In addition, if no degree-0 vertex is created in $G[V - X]$, then the total weight in the remaining set $V - X$ decreases by at least $k'w_3 + (w_4 - w_3)r$ for the integers k' and $r \in \{0, 1\}$ such that $p = 2k' + r$. In our analysis, we also use the following properties on a degree-3 vertex v in a reduced graph: (i) Removing $N[v]$ creates no degree-0 vertex u (otherwise $\{v, u\}$-$N(v)$ would be a 2-3 structure); and (ii) If there is no edge between two neighbors of degree-3 vertex v, then $|N_2(v)| \ge 4$ (otherwise $N(v)$-$N_2(v) \cup \{v\}$ would be a 3-4 structure).

In the following subsections, we focus on the analysis for branching rules.

6.2 Step 5

After Step 4, the graph is a reduced graph where the minimum degree is at least 3. In Step 5, the algorithm will branch on a vertex v of maximum degree by excluding it from the independent set or including it into the independent set. In the first branch, we will delete v from the graph. In the second branch, we will delete $N[v]$ from the graph. We use $\Delta_{out}(v)$ and $\Delta_{in}(v)$ to denote the decreased amount of w in the corresponding two branchings, respectively. Assume that v is of degree $d \ge 5$ and has d_i neighbors of degree i. Then $d = \sum_{i=3}^{d} d_i$. Note that each vertex $u \in N(v)$ is adjacent to a vertex in $N_2(v)$ (otherwise u would dominate v), and there are at least $|N(v)|$ edges between $N[v]$ and $N_2(v)$. To analyze how much w can decrease in each branch, we consider two cases.

(i) First let $d \ge 6$. Then we have

$$\Delta_{out}(v) = w_d + \sum_{i=3}^{d} d_i \Delta w_i \ge w_d + d(w_4 - w_3) \ge w_6 + 6(w_4 - w_3).$$

In the branch where v is included into the independent set, we will remove all vertices in $N[v]$, which decreases the degree of the vertices in $N_2(v)$. There are at least $|N(v)| \ge 6$ edges between $N[v]$ and $N_2(v)$, and deleting $N[v]$ decreases the weight of vertices in $N_2(v)$ by at least $2w_3$. Hence we get

$$\Delta_{in}(v) \ge w_d + \sum_{i=3}^{d} d_i w_i + 2w_3 \ge w_6 + 8w_3.$$

Let $C(w)$ denote the worst-case size of the search tree when the parameter of the graph is w. We have the following recurrence

$$\begin{aligned} C(w) &= C(w - \Delta_{out}(v)) + C(w - \Delta_{in}(v)) \\ &\le C(w - (w_6 + 6w_4 - 6w_3)) + C(w - (w_6 + 8w_3)). \end{aligned} \tag{6}$$

(ii) Next let $d = 5$. It holds $d = d_5 + d_4 + d_3$, and we get:

$$\Delta_{out}(v) = w_5 + d_5(w_5 - w_4) + d_4(w_4 - w_3) + d_3 w_3.$$

Hence there are at least $|N(v)| \ge 5$ edges between $N[v]$ and $N_2(v)$, and deleting $N[v]$ the weight of vertices in $N_2(v)$ by at least $2w_3$. We get

$$\Delta_{in}(v) \ \ge \ w_d + \sum_{i=3}^{5} d_i w_i + 2w_3 \ge w_5 + d_5 w_5 + d_4 w_4 + d_3 w_3 + 2w_3.$$

Therefore, we get the recurrence for the case $d = 5$:

$$\begin{aligned} C(w) &= C(w - \Delta_{out}(v)) + C(w - \Delta_{in}(v)) \\ &\leq C(w - (w_5 + d_5(w_5 - w_4) + d_4(w_4 - w_3) + d_3 w_3)) \\ &\quad + C(w - (w_5 + d_5 w_5 + d_4 w_4 + d_3 w_3 + 2w_3)) \\ &\qquad\qquad \text{for } 0 \leq d_3, d_4, d_5 \leq 5 \text{ with } d_5 + d_4 + d_3 = 5. \end{aligned} \tag{7}$$

This generates 21 recurrences.

6.3 Step 6

In this step, the graph is a reduced graph where there is no short funnel and each vertex is of degree 3 or 4. A triangle containing both degree-3 and degree-4 vertices is called *irregular*. If the graph contains an irregular triangle, then there exists a good funnel a-v-$(N(v) - \{a\})$, on which the algorithm will branch by removing either $N[S_a]$ or $N[v]$.

Case 1. The good funnel is a 4-funnel a-v-$\{b, c, d\}$: Since there is no dominated vertex, a is not adjacent to any of b, c and d. Note that $v \in N(a)$ and there are at least 5 edges between $N(a)$ and $N_2(a)$. After removing $N[a]$, the total weight of vertices in $N[a]$ decreases by at least $3w_3 + w_4$, that in $\{b, c, d\}$ by at least $3(w_4 - w_3)$, and that in the other vertices in $N_2(a)$ by at least $2(w_4 - w_3)$. Note that $N[a] \subseteq N[S_a]$. In the branching of removing $N[S_a]$, the measure w decreases by at least $6w_4 - 2w_3$ in total. In the other branch of removing $N[v]$, the total weight in $V' = \{v, b, c, d, a\}$ decreases by at least $w_3 + 4w_4$. There are at least 5 edges between V' and $V - V'$, and the total weight in $V - V'$ decreases by at least $2w_3$. Totally the measure w decreases by at least $4w_4 + 3w_3$. In Case 1, we get recurrence

$$C(w) \leq C(w - (6w_4 - 2w_3)) + C(w - (4w_4 + 3w_3)). \tag{8}$$

In the rest of cases, let a-v-$\{b, c\}$ be a good 3-funnel. Note that a is not adjacent to any of b and c, otherwise v would dominate b or c. Assume without loss of generality that $d(b) \geq d(c)$. Then $d(b) \geq 4$. Let p_a (resp., p_v) be the number of edges between $N[a]$ and $N_2(a)$ (resp., $N[v]$ and $N_2(v)$). We distinguish three cases.

Case 2. $d(a) = 4$: First we look at the branch where $N[S_a]$ is removed. In this branch at least $N[a] \subseteq N[S_a]$ is removed. Each neighbor $t \in N(a)$ of a is adjacent to a vertex in $N_2(a)$, otherwise t would dominate a. Hence $p_a \geq 5$ and $|N_2(a)| \geq 4$ (otherwise a-v-$\{b, c\}$ would be a short funnel). Removing $N[a]$ decreases the weights of vertices in $N[a]$ and $N_2(a)$ by at least $w_4 + 4w_3$ and $5(w_4 - w_3)$, respectively. Then this branch decreases w by at least $w_4 + 4w_3 + 5(w_4 - w_3) = 6w_4 - w_3$.

For the other branch where $N[v]$ is removed, we consider two cases: the degree of c is 3 or 4. When c is a degree-3 vertex, there are $p_v \geq 6$ edges between $N(v)$ and $N_2(v)$. Note that b and c have no common neighbor in $N_2(v)$, otherwise c would dominate b. Hence it is impossible to create a degree-0 vertex after removing $N[v]$. Then removing $N[v]$ decreases the weight of vertices in $N_2(v)$ by at least $3w_3$ (note that $w_3 \leq 2(w_4 - w_3)$). Then this branch decreases w by at least $2w_4 + 2w_3 + 3w_3 = 2w_4 + 5w_3$ in total. When c is a degree-4 vertex, there are $p_v \geq 7$ edges between $N(v)$ and $N_2(v)$. Note that b and c can have at most one common neighbor in $N_2(v)$, otherwise c would dominate b. Hence it is impossible to create two degree-0 vertices after removing $N[v]$. Then removing $N[v]$ decreases the total weight in $N_2(v)$ by at least $3w_3$. Totally this branch decreases w by at least $3w_4 + w_3 + 3w_3 = 3w_4 + 4w_3 > 2w_4 + 5w_3$.

In Case 2, we can branch with the following recurrence

$$C(w) \leq C(w - (6w_4 - w_3)) + C(w - (2w_4 + 5w_3)). \tag{9}$$

Case 3. $d(a) = 3$: Let $N(a) = \{v, t, t'\}$, where $d(t) \leq d(t')$ is assumed without loss of generality. Let $0 \leq \gamma_1 \leq 2$ be the number of degree-4 neighbors of a and $1 \leq \gamma_2 \leq 2$ be the number of degree-4 vertices in $\{b, c\}$.

First, we look at the first branch where $N[v]$ is removed. This decreases the total weight of vertices in $\{a,v\}$ by $2w_3$ and that in $\{b,c\}$ by $\gamma_2 w_4 + (2-\gamma_2)w_3$. Next we analyze how much weight in $N_2(v)$ decreases after removing $N[v]$. There are $4+\gamma_2$ edges between $N[v]$ and $N_2(v)$. We consider two cases on the first branch: the degree of c is 3 or 4.

(1-i) c is a degree-3 vertex (now $\gamma_2 = 1$): Let c' ($\neq v, b$) be the third neighbor of c. Then $c' \neq t, t'$ (otherwise b would be a roof) and the degree of c' is 3 (otherwise c'-c-$\{v,b\}$ would be a good funnel of Case (ii)). Let b' and b'' be the third and fourth neighbors b. Note that $b', b'' \neq c'$ otherwise c would dominate b. If $\{b', b''\} \cap \{t, t'\} = \emptyset$, then the weight of vertices in $N_2(v)$ decreases by at least $W_{N_2}(\gamma_1) = \gamma_1(w_4 - w_3) + (2-\gamma_1)w_3 + w_3 + 2(w_4 - w_3) = (2+\gamma_1)w_4 + (1-2\gamma_1)w_3$ after removing $N[v]$. Next consider the case of $\{b', b''\} \cap \{t, t'\} \neq \emptyset$. There is at most one edge between $\{t, t'\}$ and $\{b, c\}$ (otherwise a-v-$\{b,c\}$ would be a short funnel), and $\{b', b''\} \cap \{t, t'\}$ contains at most one vertex, say $b' = t$. Then removing $N[v]$ decreases the weight in $N_2(v)$ by at least $\gamma_1(w_4 - w_3) + (2-\gamma_1)w_3 + w_3 + (w_4 - w_3) + w_3 \geq W_{N_2}(\gamma_1)$ (when b' is a degree-4 vertex) or by $\gamma_1(w_4 - w_3) + (2-\gamma_1)w_3 + w_3 + (w_4 - w_3)$ (when b' is a degree-3 vertex) leaving b' as only one degree-1 vertex in the graph. For the latter case, we can further decrease w by at least $w_4 - w_3$ by folding degree-1 vertices and the first branch decreases the measure w by at least $W_{N_2}(\gamma_1)$ in total. Then in Case (1-i) removing $N[v]$ decreases the weight in $V - N[v]$ by at least $W_{N_2}(\gamma_1)$.

(1-ii) c is a degree-4 vertex (now $\gamma_2 = 2$): Let c'' be the fourth neighbor of c. Note that $\{c', c''\} \neq \{b', b''\}$ since otherwise c would dominate b. If $\{t, t'\} \cap \{c', c'', b', b''\} = \emptyset$, then the weight of vertices in $\{t, t'\}$ decreases by at least $\gamma_1(w_4 - w_3) + (2-\gamma_1)w_3$ and that in $\{c', c''\} \cap \{b', b''\}$ by at least $2(w_4 - w_3) + w_3$. Then removing $N[v]$ decreases the total weight of vertices in $N_2(v)$ by at least $W_{N_2}(\gamma_1)$. If there is an edge between $\{t, t'\}$ and $\{b, c\}$, then we see that the measure w decreases by at least $W_{N_2}(\gamma_1)$ by the same argument for c being a degree-3 vertex. Therefore removing $N[v]$ always decreases the weight in $V - N[v]$ by at least $W_{N_2}(\gamma_1)$.

Therefore, the first branch decreases the measure w by at least

$$\begin{aligned} W_1(\gamma_1, \gamma_2) &= 2w_3 + \gamma_2 w_4 + (2-\gamma_2)w_3 + W_{N_2}(\gamma_1) \\ &= (2+\gamma_1+\gamma_2)w_4 + (5-2\gamma_1-\gamma_2)w_3. \end{aligned}$$

For the second branch where $N[S_a]$ is removed, we consider two subcases.

Case 3.1. Vertex a is not in a triangle: For this case, we analyze how much we measure w will decrease by removing only $N[a]$ ($\subseteq N[S_a]$). Removing $N[a]$ decreases the weight of vertices in $\{a, v\}$ by $2w_3$ and that in $\{t, t'\}$ by $\gamma_1 w_4 + (2-\gamma_1)w_3$. We consider weight decrease of vertices in $N_2(a)$. There are $6+\gamma_1$ edges between $N[a]$ and $N_2(a)$. If no degree-1 vertex is created after removing $N[a]$, then the weight of vertices in $N_2(a)$ decreases by at least $W'_{N_2}(\gamma_1, \gamma_2) = (6+\gamma_1-(2-\gamma_2))(w_4 - w_3) + (2-\gamma_2)w_3 = (4+\gamma_1+\gamma_2)w_4 - (2+\gamma_1+2\gamma_2)w_3$ (note that there are at least $2-\gamma_2$ degree-3 vertices in $N_2(a)$). If a degree-1 vertex v' is created after removing $N[a]$, then the neighbor v'' of v' is a vertex of degree ≥ 3 in $G' = G - N[a]$, since otherwise v', v'' and a vertex in $\{t, t'\}$ (say t) will form a triangle and then t'-v'-$\{v'', t\}$ would be a short funnel. In this case, we can further decrease w by at least w_3 by removing dominated vertex v''; in total the measure w decreases by more than $W'_{N_2}(\gamma_1, \gamma_2)$. Hence the second branch decreases w by at least

$$\begin{aligned} W_2(\gamma_1, \gamma_2) &= 2w_3 + \gamma_1 w_4 + (2-\gamma_1)w_3 + W'_{N_2}(\gamma_1, \gamma_2) \\ &= (4+2\gamma_1+\gamma_2)w_4 + (2-2\gamma_1-2\gamma_2)w_3. \end{aligned}$$

We get recurrences:

$$C(w) \leq C(w - W_1(\gamma_1, \gamma_2)) + C(w - W_2(\gamma_1, \gamma_2)), \tag{10}$$

where $\gamma_1 \in \{0, 1, 2\}$ and $\gamma_2 \in \{1, 2\}$.

Case 3.2. Vertex a is in a triangle: Note that now v-a-$\{t, t'\}$ is also a 3-funnel. After removing $N[S_a] \supseteq N[a]$, by the above analysis (1-i) and (1-ii) of the branch where $N[v]$ is removed, we know that w will decrease by at least

$$W_2'(\gamma_1, \gamma_2) = W_1(\gamma_2, \gamma_1) = (2 + \gamma_1 + \gamma_2)w_4 + (5 - \gamma_1 - 2\gamma_2)w_3.$$

However, when $\gamma_1 = 0$ this is not good enough for our analysis. In fact, for this case vertex a has at least two satellites and we can show that the measure decreases more. Assume that $\gamma_1 = 0$ (i.e., $d(t) = d(t') = 3$). Let s and s' be the third neighbor of t and t', respectively. Note that s and s' are not adjacent otherwise a would be a roof. There are at least four edges between $\{s, s'\}$ and $N(\{s, s'\}) - \{t, t'\}$. It is impossible to have $N(\{s, s'\}) - \{t, t'\} = \{b, c\}$ otherwise the graph would contain only 8 vertices. Let $x \in N(\{s, s'\})$ be a vertex different from t, t', b and c. If one of b and c (say b) is in $N(\{s, s'\})$, then after removing $N[S_a]$, the degree of c will decrease by at least 2. In this case, removing $N[S_a]$ decreases the weight of vertices in $\{v, a, t, t', s, s', x, b, c\}$ by at least $7w_3 + \gamma_2 w_4 + (2 - \gamma_2)w_3 = \gamma_2 w_4 + (9 - \gamma_2)w_3$. If none of b and c is in $N(\{s, s'\})$, then $N(\{s, s'\}) - \{t, t'\}$ contains at least two vertices x and x' different from t, t', b and c, and removing $N[S_a]$ decreases the weight of vertices in $\{v, a, t, t', s, s', x, x'\}$ by at least $8w_3$ and that in $\{b, c\}$ by $\gamma_2(w_4 - w_3) + (2 - \gamma_2)w_3$; in total w decreases by at least $\gamma_2 w_4 + (10 - 2\gamma_2)w_3$ (where $\gamma_2 w_4 + (10 - 2\gamma_2)w_3 \le \gamma_2 w_4 + (9 - \gamma_2)w_3$ by $\gamma_2 \ge 1$). Hence the second branch decreases w by at least $\gamma_2 w_4 + (10 - 2\gamma_2)w_3$.

For Case 3.2, we get recurrences:

$$C(w) \le C(w - W_1(\gamma_1, \gamma_2)) + C(w - W_2'(\gamma_1, \gamma_2)), \tag{11}$$

where $\gamma_1 \in \{1, 2\}$ and $\gamma_2 \in \{1, 2\}$, and

$$C(w) \le C(w - W_1(0, \gamma_2)) + C(w - \gamma_2 w_4 - (10 - 2\gamma_2)w_3), \tag{12}$$

where $\gamma_2 \in \{1, 2\}$.

Note that after Step 6, the graph has no triangle that contains both degree-3 and degree-4 vertices.

6.4 Step 7

In this step, we will branch on 4-cycles that contain at least one degree-4 vertex. Without loss of generality, we assume that the algorithm will branch on 4-cycle $abcd$, where a is a degree-4 vertex. Note that if there is a degree-3 vertex in the cycle, a and c (b and d) are not adjacent, otherwise there would be an irregular triangle. According to the branching rule, our algorithm will branch by removing either $\{a, c\}$ or $\{b, d\}$ from the graph. We distinguish the following five cases.

Case 1. The other vertices than a in the 4-cycle are of degree 3: We assume that a' and a'' are the third and fourth neighbors of a, b' is the third neighbor of b, c' is the third neighbor of c, and d' is the third neighbor of d (see Fig. 3(a) for an illustration). Note that $b' \ne d'$, otherwise $\{b, d\}$-$\{a, c, b' = d'\}$ would be a 2-3 structure. Also $\{a', a'', c'\} \cap \{b', d'\} = \emptyset$, otherwise there would be an irregular triangle or a roof.

In the branch where $\{a, c\}$ is removed, b and d will become degree-1 vertices. The algorithm will apply the reduction rules to reduce degree-1 vertices immediately. Then b' and d' will be removed. Totally, at least 6 vertices a, b, c, d, b' and d' are removed from the graph. There are also at least 5 edges between $V' = \{a, b, c, d, b', d'\}$ and $V - V'$ (there may not be 7 edges when b' and d' are adjacent). We consider how much weight of vertices in $V - V'$ decreases after removing V'. If $|N(V')| \ge 3$, then the weight in $V - V'$ decreases by at least $w_3 + 2(w_4 - w_3) = 2w_4 - w_3$. If $|N(V')| = 2$, then after removing V' the weight in $V - V'$ decreases either by $w_4 + (w_4 - w_3)$ (at least one vertex in $N(V')$ is a degree-4 vertex) or by at least $2w_3$ (both vertices in $N(V')$ are degree-3 vertices) together with some degree-1 vertex created. For the latter case, we can further decrease w by at least $w_4 - w_3$ by removing dominated vertices adjacent to degree-1 vertices, and

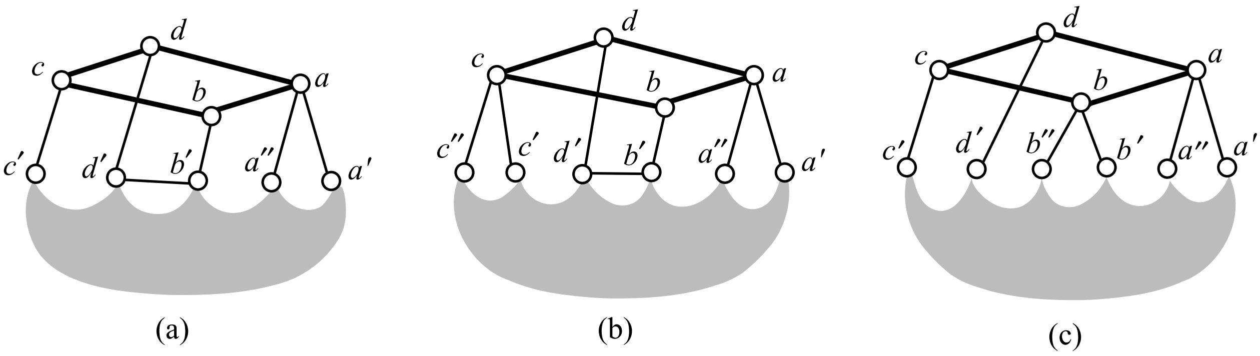

Figure 3: Branching on 4-cycles

thereby the weight in $V - V'$ decreases by at least $2w_3 + (w_4 - w_3) = w_4 + w_3$. For any case, the weight in $V - V'$ by at least $2w_4 - w_3$ after removing V' (recall that $w_3 \geq w_4 - w_3$). Then in the branch of removing $\{a, c\}$ the measure w decreases by at least

$$w_4 + 5w_3 + 2w_4 - w_3 = 3w_4 + 4w_3.$$

In the other branch where $\{b, d\}$ is removed, c will become a degree-1 vertex and we will further remove c' from the graph. Thus the branch will remove $N[c]$. Let us see how much weight in $V - N[c]$ will decrease by removing $N[c]$. There are at least 6 edges between $N[c]$ and $V - N[c]$. Note that $|N_2(c)| \geq 3$ since $\{a, d, d'\} \subseteq N_2(c)$. Note that no pair of neighbors of c are adjacent. If $|N_2(c)| = 3$, then $N(c)$-$N_2(c) \cup \{c\}$ would be a 3-4 structure. We know that $|N_2(c)| \geq 4$. No degree-0 vertex u is created after removing $N[c]$ otherwise $\{c, u\}$-$N(c)$ would be a 2-3 structure. If $|N_2(c)| \geq 5$, then the measure w decreases by at least $w_4 + 4(w_4 - w_3)$ (w_4 from a and $4(w_4 - w_3)$ from the other 4 vertices in $N_2(c)$). Now let $|N_2(c)| = 4$. If no degree-1 vertex is created after removing $N[c]$, then the weight in $N_2(c)$ still decreases by at least $w_4 + 4(w_4 - w_3)$. If a degree-1 vertex u is created, then the weight in $N_2(c)$ may only decrease by $w_4 + 2(w_4 - w_3) + w_3$. Note that u is the unique degree-1 vertex in the graph, and we can further decrease w by at least $(w_4 - w_3)$ by removing the dominated vertices adjacent to degree-1 vertices. Then for any case, the weight in $V - N[c]$ decreases by at least $w_4 + 4(w_4 - w_3)$. Totally, in the branch of removing $\{b, d\}$ the measure w decreases by at least

$$4w_3 + w_4 + 4(w_4 - w_3) = 5w_4.$$

In Case 1, we can always branch with the following recurrence

$$C(w) \leq C(w - (3w_4 + 4w_3)) + C(w - 5w_4). \tag{13}$$

Case 2. a and c are the two degree-4 vertices in the 4-cycle: Let b' and d' be the third neighbor of b and d, respectively. Note that $b' \neq d'$ holds, otherwise $\{b, d\}$-$\{a, c, b' = d'\}$ would be a 2-3 structure. Also b' (d') is not adjacent to a or c, otherwise there would be an irregular triangle. See Fig. 3(b) for an illustration of this case. Let $0 \leq k \leq 2$ be the number of vertices of degree 4 in $\{b', d'\}$.

It is easy to see that in the branch where $\{b, d\}$ is removed, the weight of b and d decreases by $2w_3$, that of a and c by $2w_4$, and that of b' and d' by at least $k(w_4 - w_3) + (2 - k)w_3$; totally the measure w decreases by $2w_3 + 2w_4 + k(w_4 - w_3) + (2 - k)w_3 = (2 + k)w_4 + (4 - 2k)w_3$.

In the other branch where $\{a, c\}$ is removed, b and d become degree-1 vertices and we will also further remove b' and d'. Let $V' = \{a, b, c, d, b', d'\}$. Removing V' decreases the sum of weights of vertices in V' by $(2+k)w_4 + (4-k)w_3$. We consider how much weight of vertices in $V - V'$ decreases after removing V'. Note that there are at least $6+k$ edges between V' and $V-V'$ (b' and d' may be adjacent) and $|N(V')| \geq 4$. Then the weight in $V-V'$ decreases by at least $w_3+(3+k)(w_4-w_3)$ for $k=0,1$ and by $2w_3+2(w_4-w_3)=2w_4$ for $k=2$. In Case 2, we can branch with at least one of the following recurrences:

$$C(w) \leq C(w-(2+k)w_4-(4-2k)w_3)+C(w-(5+2k)w_4-(2-2k)w_3) \tag{14}$$

for $k \in \{0,1\}$; and

$$C(w) \leq C(w-4w_4)+C(w-(6w_4+2w_3)) \quad \text{for } k=2. \tag{15}$$

Case 3. a and b (or a and d) are the two degree-4 vertices in the 4-cycle: Assume without loss of generality that a and b are the degree-4 vertices in the cycle. Define a', a'', b', c' and d' as in Case 1, and let b'' be the fourth neighbor of b (see Fig. 3(c) for an illustration). Since the graph has no irregular triangle, vertex c' (d') is different from any of b' and b'' (a' and a'') whereas $d' \in \{b', b''\}$ and $\{a', a''\} \cap \{b', b'', c'\} \neq \emptyset$. Also $c' \neq d'$, otherwise 5-cycle $c'cabd$ would contain a roof c'. We look at the branch where $\{a, c\}$ is removed. Vertex d will become a degree-1 vertex and we will further remove the dominated vertex d' and the degree-0 vertex d. Then in this branch we will remove $N[d]$. We consider how much weight of vertices in $V-N[d]$ decreases after removing $N[d]$. Note that no pair of vertices in $N(d)$ can be adjacent, otherwise there would be an irregular triangle or a roof. There are at least 7 edges between $N(d)$ and $N_2(d)$. It is impossible to create a degree-0 vertex v after removing $N[d]$, otherwise $\{d, v\}$-$\{a, c, d'\}$ would be a 2-3 stricture. Since $a', a'', b \in N_2(d)$, it holds $|N_2(d)| \geq 3$. If $|N_2(d)| \geq 4$, then the weight in $N_2(d)$ decreases by at least $w_4+w_3+2(w_4-w_3)$ (w_4 from b and $w_3+2(w_4-w_3)$ from the other vertices in $N_2(d)$). For the other case of $|N_2(d)|=3$, there are at least two degree-4 vertices in $N_2(d)$ (because no degree-0 vertex is created after removing $N[d]$), and the weight in $N_2(d)$ decreases by at least $w_4+w_4+w_3$ (w_4 from b and w_4+w_3 from the other vertices in $N_2(d)$); the weight in $V-N[d]$ still decreases by at least $w_4+w_3+2(w_4-w_3)=3w_4-w_3$. Therefore, the branch of removing $\{a, c\}$ decreases the measure w by at least $w_4+3w_3+3w_4-w_3=4w_4+2w_3$. This also holds for the other branch where $\{b, d\}$ is removed. In Case 3, we can branch with recurrence

$$C(w) \leq 2C(w-(4w_4+2w_3)). \tag{16}$$

Case 4. There are exactly three degree-4 vertices in the 4-cycle: Without loss of generality, we assume that the three degree-4 vertices are a, b and c. Note that the third neighbor d' of d is not adjacent to a or c. In the branch where $\{a, c\}$ is removed, vertex d becomes a degree-1 vertex and we will further remove $\{d'\}$. This decreases the weight of vertices in $\{a, b, c\}$ by $3w_4$, that in $\{d, d'\}$ by at least $2w_3$, and that in $V-\{a, b, c, d, d'\}$ by at least $3w_3$ (note that there are at least 6 edges between $\{a, c, d, d'\}$ and $V-\{a, b, c, d, d'\}$ and no degree-0 vertices will be created after removing $N[d]$). Totally the measure w decreases by at least $3w_4+5w_3$.

In the other branch, the measure w decreases by at least $3w_4+w_3+w_3+(w_4-w_3)=4w_4+w_3$. In Case 4, we get recurrence

$$C(w) \leq C(w-(3w_4+5w_3))+C(w-(4w_4+w_3)). \tag{17}$$

Case 5. All the vertices in the 4-cycle are degree-4 vertices: Note that for this case, a and c (also, b and d) may be adjacent to each other. Since no 4-funnels in a reduced graph, the vertices $abcd$ do not induce a clique of size 4. Assume without loss of generality a and c are not adjacent. The branch of removing $\{b, d\}$ decreases the weight of vertices in $\{a, b, c, d\}$ by $4w_4$ and that in $V-\{a, b, c, d\}$ by at least $2(w_4-w_3)$. The other branch of removing $\{a, c\}$ decreases the weight of vertices a, b, c and d by at least $4w_4$ and that in $V-\{a, b, c, d\}$ by at least $2w_3$. In Case 5 we get recurrence

$$C(w) \leq C(w-(6w_4-2w_3))+C(w-6w_4). \tag{18}$$

Note that after Step 7, no degree-4 vertex is contained in a 4-cycle.

6.5 Step 8

First of all, we show that there is always a good degree-4 vertex adjacent to another degree-4 vertex if the graph has two adjacent degree-4 vertices in this step. Note that if all vertices in a connected component are degree-4 vertices none of which is a good degree-4 vertex, then this component is the line graph of a 3-regular graph, which must have been reduced in Step 4. Otherwise, there is a degree-4 vertex v adjacent to both degree-3 and degree-4 vertices. If v is contained in two edge-disjoint triangles, then there is an irregular triangle, which will form a short funnel or a funnel satisfying the condition in Step 6. Then v is a good degree-4 vertex. Note that there is at least one degree-4 vertex in $N(v)$.

Note that when we remove v, a degree-3 neighbor $u \in N(v)$ becomes a degree-2 vertex, which will be removed by a reduction rule (folding or domination) and decreases the weight w by at least β by Lemma 10. Since v is not in any 4-cycle and no degree-3 neighbor is adjacent to another neighbor of v, each vertex in $N_2(v)$ is adjacent to exactly one neighbor of v, and if v has l degree-3 neighbors then after deleting $N[v]$, the weight of the vertices can decrease by at least $l\beta$ by folding these degree-2 vertices.

Let v be the good degree-4 vertex selected in this step. We will branch by either deleting v from the graph or deleting $N[v]$ from the graph. We distinguish the following five cases according to the number of degree-4 vertices in $N(v)$.

Case 1: There is only one degree-4 vertex in $N(v)$. Then $|N_2(v)| = 9$, otherwise there would be a 4-cycle containing a degree-4 vertex or a triangle containing both degree-3 and degree-4 vertices. The possible local structure in this case is showed in Fig. 4. In the branch where v is removed, the measure w decreases by $w_4 + 3w_3 + (w_4 - w_3) + 3\beta = 2w_4 + 2w_3 + 3\beta$. In the other branch where $N[v]$ is removed, the measure w decreases by at least $2w_4 + 3w_3 + 9(w_4 - w_3) = 11w_4 - 6w_3$. We get recurrence

$$C(w) \leq C(w - (2w_4 + 2w_3 + 3\beta)) + C(w - (11w_4 - 6w_3)). \tag{19}$$

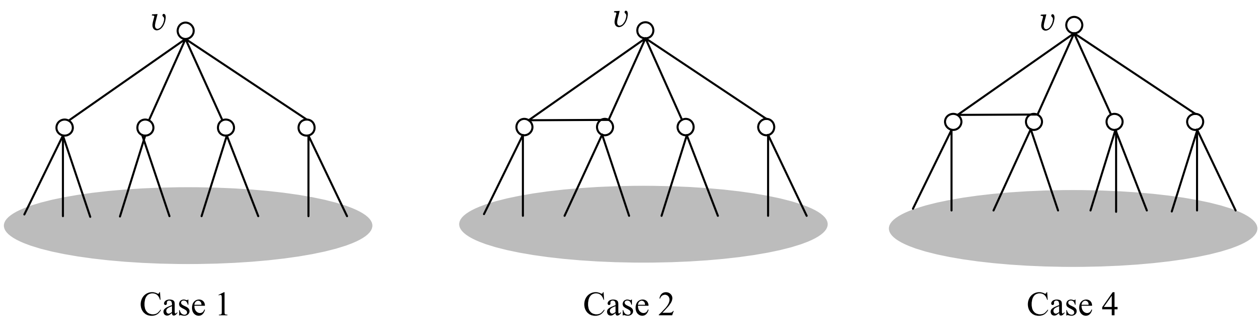

Figure 4: Some cases of branching on a degree-4 vertex

Case 2: There are two degree-4 vertices in $N(v)$. We have two subcases: the two degree-4 vertices are adjacent or not. It is easy to see that the case of adjacent will cover the other case. We assume that the two degree-4 vertices are adjacent to each other (see Fig. 4). Then $|N_2(v)| = 8$. When v is removed, the measure w decreases by $w_4 + 2w_3 + 2(w_4 - w_3) + 2\beta = 3w_4 + 2\beta$. When $N[v]$ is removed, the measure w decreases by at least $3w_4 + 2w_3 + 8(w_4 - w_3) = 11w_4 - 6w_3$. We get recurrence

$$C(w) \leq C(w - (3w_4 + 2\beta)) + C(w - (11w_4 - 6w_3)). \tag{20}$$

Case 3: There are three degree-4 vertices in $N(v)$. There is also at most one edge with two endpoints in $N(v)$. Then $|N_2(v)| \geq 9$. When v is removed, the measure w decreases by $w_4 + w_3 +$

$3(w_4 - w_3) + \beta = 4w_4 - 2w_3 + \beta$. When $N[v]$ is removed, the measure w decreases by at least $4w_4 + w_3 + 9(w_4 - w_3) = 13w_4 - 8w_3$. We get recurrence

$$C(w) \leq C(w - (4w_4 - 2w_3 + \beta)) + C(w - (13w_4 - 8w_3)). \tag{21}$$

Case 4: All vertices in $N(v)$ are degree-4 vertices. Since v is a good degree-4 vertex that is not contained in any 4-cycles. There is also at most one edge with two endpoints in $N(v)$ (see Fig. 4). For this case, $|N_2(v)| \geq 10$. When v is removed, the measure w decreases by $w_4+4(w_4-w_3) = 5w_4-4w_3$. When $N[v]$ is removed, the measure w decreases by at least $5w_4 + 10(w_4 - w_3) = 15w_4 - 10w_3$. We get recurrence

$$C(w) \leq C(w - (5w_4 - 4w_3)) + C(w - (15w_4 - 10w_3)). \tag{22}$$

6.6 Step 9

In this step, the set of degree-4 vertices is an independent set. Let v be a degree-4 vertex selected in this step. Then the neighbors of v are four degree-3 vertices. We show that there is at least one degree-3 vertex in $N_2(v)$. Assume to the contrary that for each vertex v', $N_2(v')$ contains only degree-4 vertices. Then the graph is a bipartite graph with one side of degree-3 vertices and the other degree-4 vertices, which must have been reduced by our reduction rules. Now we branch on v. In the branching where v is removed, the measure w decreases by $w_4 + 4w_3 + 4\beta$. Note that $|N_2(v)| = 8$ (v is not contained in any 3-cycle or 4-cycle). Then the other branching of removing $N[v]$ decreases the weight of vertices in $N_2(v)$ by at least $7(w_4 - w_3) + w_3 = 7w_4 - 6w_3$ and that in $N[v]$ by $w_4 + 4w_3$; totally the measure w decreases by at least $8w_4 - 2w_3$. We get recurrence

$$C(w) \leq C(w - (w_4 + 4w_3 + 4\beta)) + C(w - (8w_4 - 2w_3)). \tag{23}$$

6.7 Step 10

It is easy to see that if none of the first 9 steps can be executed, the graph is a 3-regular graph. We will use a fast algorithm for MIS3 to solve it. Here we use the $O^*(1.0836^n)$-time algorithm by Xiao and Nagamochi [20], and then this step will not be the bottleneck of our algorithm. For this step, we get running time bound

$$C(w) = O(1.0836^{\frac{w}{w_3}}), \tag{24}$$

which will generate the last constraint in our quasiconvex program.

6.8 Putting All Together

Recurrences (6) to (24) generate the constraints in our quasiconvex program. By solving this quasiconvex program under conditions (1), (2), and (4) according to the method introduced in [6], we get a running time bound of $O(1.14459^w)$ by setting $w_3 = 0.59933$, $w_4 = 1$, $w_5 = 1.40066$, $w_6 = 1.80132$, and $\beta = 0.19867$ for our problem. Now the bottlenecks are (1) with $w_4 \leq 1$, $\beta \leq w_3 + w_3 - w_4$, $\beta \leq w_4 + w_3 - w_5$, (20), and (22).

Theorem 11 *A maximum independent set in a degree-4 graph of n vertices can be found in $O^*(1.1446^n)$ time.*

7 Concluding Remarks

In this paper, we have designed a fast algorithm for the maximum independent set problem in graphs with degree bounded by 4, which is analyzed by the "Measure and Conquer" method. Different from most previous "Measure and Conquer" algorithms, our algorithm allows the weight of vertices greater than 1 and we carefully choose the weight of such vertices so that the measure w effectively decreases by the reduction rule for degree-2 vertices after branching on degree-4 vertices. In this paper, we have clearly listed out all constraints in our quasiconvex program and pointed out the bottlenecks of the algorithm.

References

[1] Beigel, R.: Finding maximum independent sets in sparse and general graphs. In: SODA 1999. ACM Press (1999) 856–857

[2] 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 (2010) 373–384

[3] Bourgeois, N., Escoffier, B., Paschos, V. T., van Rooij, J. M. M.: A bottom-up method and fast algorithms for max independent set. In: SWAT 2010, LNCS 6139 (2010) 62–73

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

[5] Chor, B., Fellows, M., Juedes, D. W.: Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps. In: WG 2004. LNCS 3353, Springer (2004) 257–269

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

[7] Fomin, F. V., Grandoni, F., Kratsch, D.: Measure and conquer: a simple $O(2^{0.288n})$ independent set algorithm. In: SODA, ACM Press (2006) 18–25

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

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

[10] Garey, M. R., Johnson, D. S.: Computers and intractability: A guide to the theory of NP-completeness. Freeman, San Francisco (1979)

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

[12] 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

[13] 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

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

[15] Robson, J.: Finding a maximum independent set in time $O(2^{n/4})$. Technical Report 1251-01, LaBRI, Univsersite Bordeaux I (2001)

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

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

[18] 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

[19] 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

[20] Xiao, M., Nagamochi, H.: Confining sets and avoiding bottleneck cases: A simple maximum independent set algorithm in degree-3 graphs. Technical Report 2012-002, May 8 (2012) `url: http://www-or.amp.i.kyoto-u.ac.jp/members/nag/Technical_report/TR2012-002.pdf`