Confining Sets and Avoiding Bottleneck Cases: A Simple Maximum Independent Set Algorithm in Degree-3 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

We present an $O^*(1.0836^n)$-time algorithm for finding a maximum independent set in an n-vertex graph with degree bounded by 3, which improves all previous running time bounds for this problem. Our approach has the following two features. Without increasing the number of reduction/branching rules to get an improved time bound, we first successfully extract the essence from the previously known reduction rules such as domination, which can be used to get simple algorithms. More formally, we introduce a procedure for computing "confining sets," which unifies several known reducible subgraphs and covers new reducible subgraphs. Second we identify those instances that generate the worst recurrence among all recurrences of our branching rules as "bottleneck instances" and prove that bottleneck instances cannot appear consecutively after each branching operation.

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 one of the basic NP-hard optimization problems and has been extensively studied in the literature, in particular in the line of research on worst-case analysis of algorithms for NP-hard optimization problems. In 1977, Tarjan and Trojanowski [16] designed the first nontrivial algorithm for this problem, which runs in $O^*(2^{n/3})$ time and polynomial space. Later, the running time was improved to $O^*(2^{0.304n})$ by Jian [10]. Robson [14] obtained an $O^*(2^{0.296n})$-time polynomial-space algorithm and an $O^*(2^{0.276n})$-time exponential-space algorithm. In a technical report [15], Robson also claimed better running times. Fomin *et al.* [7] got an $O^*(2^{0.288n})$-time polynomial-space algorithm by using the "Measure and Conquer" method. Recently Kneis et al. [11] and Bourgeois et al. [2] improved the running time bound to $O^*(1.2132^n)$ and $O^*(1.2127^n)$ respectively. There is also a considerable amount of contributions to the maximum independent set problem in sparse graphs, especially in degree-3 graphs [1, 5, 19, 4]. Chen at al. [5] showed that MIS3 (the maximum dependent set problem in degree-3 graphs) can be solved in $O^*(1.1254^n)$ time. Xiao et al. [19] used the number of degree-3 vertices as a measure to analyze algorithms and got an $O^*(1.1034^n)$-time algorithm for MIS3. Razgon [12] also designed another $O^*(1.1034^n)$-time algorithm for this problem. Fürer [9] designed an algorithm for MIS3 by measuring the running time in terms of $m-n$, where m is the number of edges. Based upon a refined branching with respect to Fürer's algorithm, Bourgeois et al. [4] got an $O^*(1.0977^n)$-time algorithm for MIS3. Razgon [13] and Xiao [18] further improved the running time bound to $O^*(1.0892^n)$ and $O^*(1.0885^n)$ respectively. Currently, the best result on this problem is Bourgeois et al.'s $O^*(1.0854^n)$-time algorithm designed by carefully checking the worst cases [2]. See Table 1 for a summary on the currently published results on low-degree graphs as wellas general graphs.

[1]Technical report 2012-002, May 8, 2012

Authors	Running times	References	Notes
Tarjan & Trojanowski	$O^*(1.2600^n)$ for MIS	1977 [16]	n: number of vertices
Jian	$O^*(1.2346^n)$ for MIS	1986 [10]	
Robson	$O^*(1.2109^n)$ for MIS	1986 [14]	Exponential space
Beigel	$O^*(1.0823^m)$ for MIS $O^*(1.1259^n)$ for MIS3	1999 [1]	m: number of edges
Chen et al.	$O^*(1.1254^n)$ for MIS3	2003 [5]	
Xiao et al.	$O^*(1.1034^n)$ for MIS3	2005 [19]	Published in Chinese
Fomin et al.	$O^*(1.2210^n)$ for MIS	2006 [7]	
Fomin & Høie	$O^*(1.1225^n)$ for MIS3	2006 [8]	
Fürer	$O^*(1.1120^n)$ for MIS3	2006 [9]	
Razgon	$O^*(1.1034^n)$ for MIS3	2006 [12]	
Bourgeois et al.	$O^*(1.0977^n)$ for MIS3	2008 [4]	
Razgon	$O^*(1.0892^n)$ for MIS3	2009 [13]	
Kneis et al.	$O^*(1.2132^n)$ for MIS	2009 [11]	
Xiao	$O^*(1.0885^n)$ for MIS3	2010 [18]	
Bourgeois et al.	$O^*(1.2127^n)$ for MIS $O^*(1.0854^n)$ for MIS3	2012 [2]	
this paper	$O^*(1.0836^n)$ for MIS3	2012	

Table 1: Exact algorithms for the maximum independent set problem

One reason why MIS3 has been extensively studied is that MIS in low-degree graphs are usually the bottlenecks to get improvement for the problem in general graphs. Most previous result for MIS in general graphs are obtained by carefully analyzing the problems in low-degree graphs. Bourgeois et al. [2] presented a bottom-up method for MIS, which shows that the improvements on MIS for low-degree graphs can be used to derive improved algorithms for MIS in general graphs and get the current best result for MIS in general graphs by designing an improved algorithm for MIS3 and so on.

Most fast algorithms for the maximum independent set problem are obtained via careful examinations of the structures in the graph. In those algorithms, a long list of reduction and branching rules are used, which is derived from a somewhat complicated case analysis. In this paper, we introduce some uniform reduction and branching rules for the maximum independent set and vertex cover problems, which can be used to design simple algorithms. To catch more properties of the graphs, we use the sum of $\max\{0, \delta(v) - 2\}$ over all vertices v as the measure of a graph to analyze the algorithm, where $\delta(v)$ is the degree of a vertex v to analyze our algorithm. When the graph is a degree-3 graph, the measure is the number of degree-3 vertices in the graph. To get improvement on MIS3, we use an idea of avoiding the worse cases. Finally, our algorithm runs in $O^*(1.0836^n)$ time, which improves previous algorithms for MIS3 and can derive improved algorithm for MIS in general graph by using the bottom-up method introduced in [2].

Based on our new result on MIS3, we recently designed an $O^*(1.1446^n)$-time algorithm to MIS4 (the maximum dependent set problem in degree-4 graphs) [21], which improves the previous best bound $O^*(1.1571^n)$ on MIS4 [3].

2 Preliminaries

Let V denote the set of all vertices in an instance and let $n = |V|$. We may simply use v to denote the set $\{v\}$ of a single vertex v. For a set X of vertices, let $N(X)$ to denote the *neighbors* of X, i.e., the vertices $y \in V - X$ adjacent to a vertex $x \in X$, and denote $N(X) \cup X$ by $N[X]$. For a vertex $v \in V$, let $N_2(v)$ denote the set of vertices with distance exactly 2 from v, and $\delta(v)$ $(= |N(v)|)$

denote the degree of v. Define $\rho(v) = \max\{0, \delta(v) - 2\}$. For a graph $H = (V_H, E_H)$, we denote $\rho(H) = \sum_{v \in V_H} \rho(v)$. We also denote $\rho(X) = \sum_{v \in X} \rho(v)$ for a set X of vertices in G.

We say that an edge e is incident on a vertex set X, if at least one endpoint of e is in X. Let $G - X$ denote the graph obtained from G by removing the vertices in X and the edges incident to X. *Contracting* X is to identify all vertices in X as a single vertex s, where any resulting self-loops and multiple edges will be removed. Hence s is adjacent to a vertex $v \in V - X$ in the resulting graph if and only if v is adjacent to a vertex in X. Let G/X denote the graph obtained from G by contracting a subset X of vertices.

A subgraph of G is called a *k-path* (or *path*) if it consists of a sequence of $k+1$ distinct vertices $v_1, v_2, \ldots, v_{k+1}$ such that v_i and v_{i+1} are adjacent for each $i = 1, 2, \ldots, k$. A $(k-1)$-path $v_1, v_2, \ldots, v_k$ ($k \geq 3$) together with an edge $v_k v_1$ called a *k-cycle* (or *cycle*). A path $v_1, v_2, \ldots, v_{k+1}$ in a graph G is called a *pure path* if each non-endpoint v_i in the path has no neighbor other than v_{i-1} and v_{i+1} in G. A pure path is called an *o-path* (resp., *e-path*) if the two endpoints are of degree ≥ 3 and the number of non-endpoints (of degree 2) in it is odd (resp., even), where we allow the two endpoints being a same vertex. A component of a graph means a maximal connected subgraph of the graph.

Our algorithms are based on the branch-and-reduce paradigm. We will first apply some reduction rules to reduce the size of instances of the problem. Then we apply some branching rules to branch on the instance by including some vertices in the independent set or excluding some vertices from the independent set. In each branch, we will get a maximum independent set problem in a graph instance with a smaller measure. Next, we introduce the reduction rules and branching rules that will be used in our algorithm.

2.1 Reduction Rules

Let $\eta(G)$ denote the size of a maximum independent set of a graph G. For a subset X of vertices in G, let $\eta(X)$ denote $\eta(G')$ of the graph $G' = G - (V - X)$ induced by X.

Reduction by removing unconfined vertices

A vertex v in an instance G is called *removable* if $\eta(G) = \eta(G - v)$, i.e., there is a maximum independent set of G which does not contain v. We can tell that a vertex v is removable if a contradiction is obtained from an assumption that every maximum independent set of G contains $S = \{v\}$. Based on this idea, we introduce a sufficient condition for testing if a given is removable or not. This is also an extension of "satellite" proposed in [11].

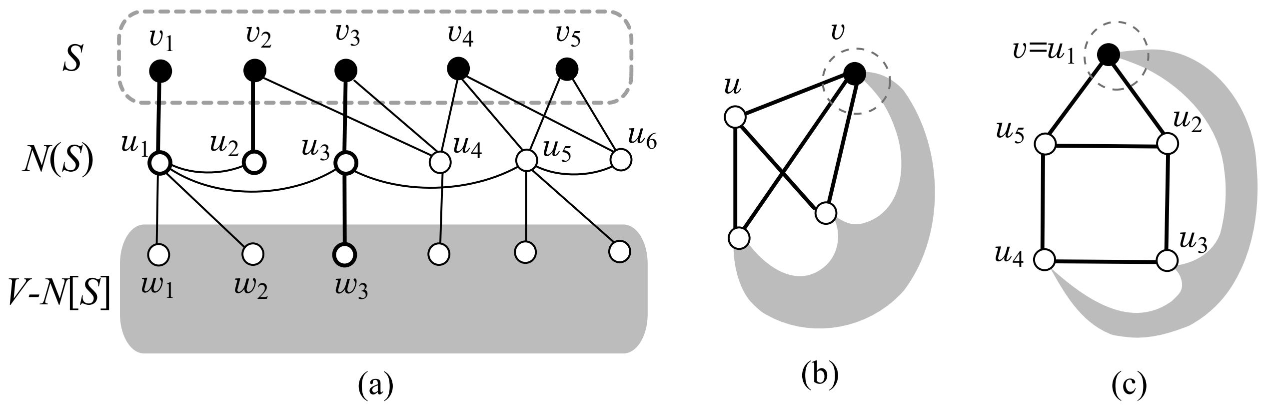

Figure 1: (a) A process of extending a confining set S; (b) A vertex v dominated by a neighbor u; (c) A roof v in a 5-cycle $u_1(=v)u_2u_3u_4u_5$.

For an independent set S of G, a vertex $u \in N(S)$ is called a *child* of S if it has a unique neighbor

$s \in S$ (i.e., $|N(u) \cap S| = 1$), where s is called the *parent* of u. (See Fig. 1(a), where vertices u_1, u_2 and u_3 are the children of $S = \{v_1, v_2, \ldots, v_5\}$.)

Lemma 1 *Let S be an independent set that is contained in any maximum independent set of G. Then every maximum independent set of G contains at least one vertex $w \in N(u) - N[S]$ for each child $u \in N(S)$.*

Proof. Assume that there is a maximum independent set S_G of G such that $S_G \cap (N(u) - N[S]) = \emptyset$ for some child $u \in N(S)$. The parent $u' \in S \cap N(u)$ of u belongs to S_G by the assumption on S. Hence we can replace the parent $u' \in S_G$ with its child u to obtain another set $S'_G = (S_G - u') \cup \{u\}$, which is an independent set of G by $N(u) \cap (S - u') = \emptyset$. However, S'_G does not entirely contain S, contradicting that S is always contained in a maximum independent set of G. ∎

Suppose that we wish to know if a given vertex v is removable or not, i.e., $\{v\}$ is an independent set that is contained in any maximum independent set of G or not. Starting with $S := \{v\}$, we repeatedly apply Lemma 1 as follows. If there is a child $u \in N(S)$ of S such that $|N(u) - N[S]| = 1$ (such as u_3 in Fig. 1), then we can add the vertex $w \in N(u) - N[S]$ to S to obtain a larger set $S \cup \{w\}$, which also needs to be contained in any maximum independent set of G. We call such a child u *extending*. On the other hand, if there is a child $u \in N(S)$ such that $N(u) - N[S] = \emptyset$ (such as u_2 in Fig. 1), then this implies that the assumption on S was false.

From these observation, we obtain the following sufficient condition for a vertex v to be removable. After starting with $S := \{v\}$, we repeat (i) until (ii) or (iii) holds:

(i) If S has any extending child in $N(S)$, then let W be the set of vertices $w \in N(u) - N[S]$ for all extending children $u \in N(S)$ of S. If W is an independent set in G, then $S := S \cup W$;

(ii) If W in (i) is not an independent set or there is a child $u \in N(S)$ such that $N(u) - N[S] = \emptyset$, then halt concluding that v is "unconfined" (see below);

(iii) If $|N(u) - N[S]| \geq 2$ for all children $u \in N(S)$, then halt by delivering S as the set S_v that "confines" v (see below).

Obviously the procedure can be executed in polynomial time for any starting set S of a vertex. If the procedure halts in (iii), then we say that the set S obtained in (iii) *confines* vertex v: vertex v is called *confined*. The set confining a vertex v is denoted by S_v, which is uniquely determined by the procedure with starting set $S = \{v\}$ (possibly $S_v = \{v\}$). On the other hand, vertex v is called *unconfined*. If v has no such set S in (iii), then it is called *unconfined*. Clearly any unconfined vertex is removable since $\eta(G) = \eta(G - v)$. As one of our reduction rules, we remove any unconfined vertex in an instance G.

We here observe two structures that involve unconfined vertices. We say that a vertex v is *dominated* by a neighbor u of it if v is adjacent to all neighbors of u, i.e., $N[v] \supseteq N[u]$. (see Fig. 1(b)). Clearly, any dominated vertex v is unconfined, since $S = \{v\}$ has a child u with $N(u) - N[S] = \emptyset$.

A *roof* is defined to be a vertex u_1 which belongs to a 5-cycle $u_1 u_2 \cdots u_5$ such that u_2 and u_5 are two adjacent degree-3 vertices. (see Fig. 1(c)). A roof $v = u_1$ is not confined, since children u_2 and u_5 of $S = \{v\}$ are extending, but $W = \{u_3, u_4\}$ is not an independent set, indicating that no set S can confine a roof.

After removing any dominated vertex, we can also remove all the resulting degree-0 vertices by including them into the solution directly. We will consider this operation as part of removing dominated vertices. For a vertex u dominated by a degree-1 vertex v, the operation of removing dominated vertex u is also called *folding* a degree-1 vertex v.

Reduction by folding degree-2 vertices and twins

We call a set $A = \{v_1, \ldots, v_k\}$ of k degree-$(k+1)$ vertices a *complete k-independent set* if they have common neighbors $N(v_1) = \cdots = N(v_k)$. If there is a complete k-independent set A, then we only need to look for a maximum independent set S_G of G such that $N[A] \cap S_G = A$ or $N[A] \cap S_G = N(A)$ (since $N[A] \cap S_G$ with $|N[A] \cap S_G| \leq k$ can be replaced with A to obtain another independent set). Then if $N(A)$ is not an independent set (the case of $N[A] \cap S_G = N(A)$ cannot occur), then the new instance $G - N[A]$ obtained by removing $N[A]$ satisfies $\eta(G) = \eta(G - N[A]) + k$ (see Fig. 2(a)); Otherwise ($N(A)$ is an independent set) the new instance $G/N[A]$ obtained by contracting $N[A]$ satisfies $\eta(G) = \eta(G/N[A]) + k$ (see Fig. 2(b)). *Folding* a complete k-independent set A is to eliminate the set $N[A]$ from an instance in the above way.

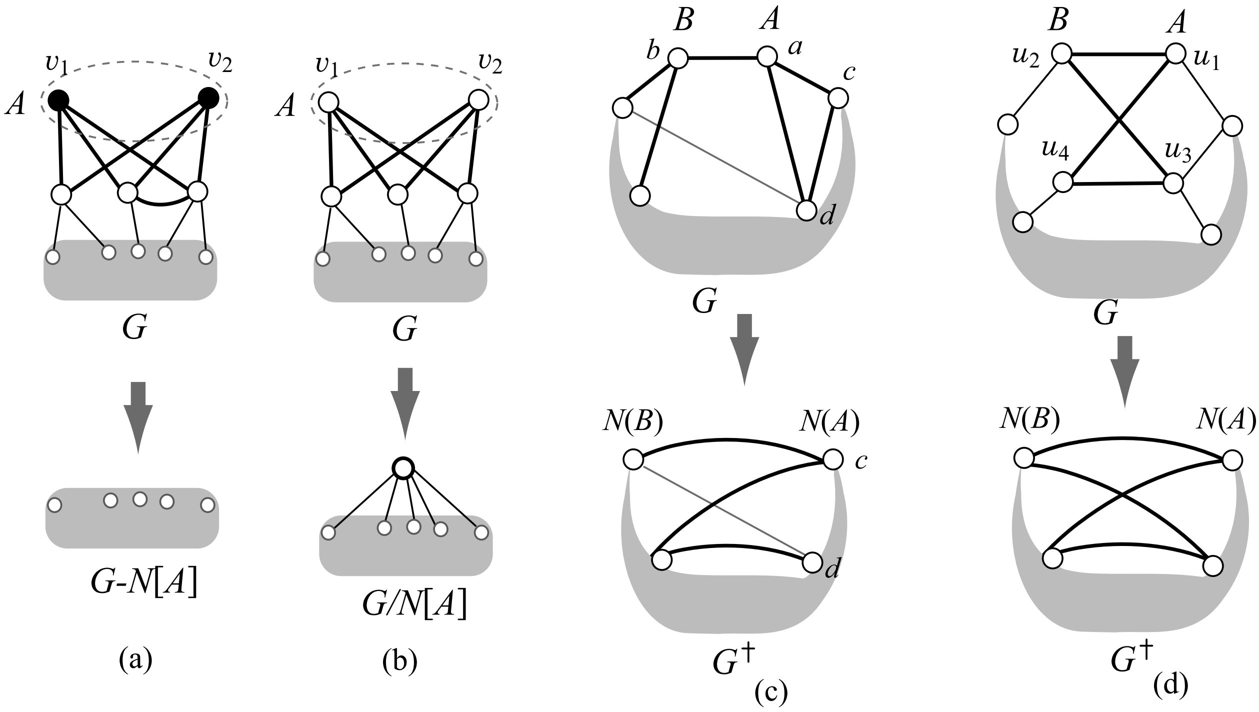

Figure 2: (a) Removing set $N[A]$ for a twin $A = \{v_1, v_2\}$; (b) Contracting set $N[A]$ for a twin $A = \{v_1, v_2\}$; (c) A short 3-funnel b-a-$\{b, c\}$; (d) A desk $u_1u_2u_3u_4$.

A complete 1-independent set $A = \{v\}$ consists of a degree-2 vertex v. *Folding* a degree-2 vertex v with no degree-1 neighbor means to fold the set $A = \{v\}$. We call a complete 2-independent set a *twin* (see Fig. 2(a) and (b)). Note that a twin is a special case of crown-reductions [6], and our algorithm does not need to rely on any other crown-reduction rules.

Reduction by folding short funnels and desks

Two disjoint independent subsets A and B in a graph G are called *alternative* if $|A| = |B| \geq 1$ and there is a maximum independent set S_G of G which satisfies $S_G \cap (A \cup B) = A$ or B. Let $G^\dagger$ be the graph obtained from G by removing $A \cup B \cup (N(A) \cap N(B))$ and adding an edge ab for every two nonadjacent neighbors $a \in N(A) - N[B]$ and $b \in N(B) - N[A]$.

Lemma 2 *For alternative subsets A and B in a graph G, $\eta(G) = \eta(G^\dagger) + |A|$.*

Proof. By definition, there is a maximum independent set S_G of G which satisfies $S_G \cap (N[A] \cup B) = A$ or $S_G \cap (N[B] \cup A) = B$. For such an independent set S_G, $S' = S_G - (A \cup B) \cup (N(A) \cap N(B))$ is an independent set in $G^\dagger$ since $S' \cap (N(A) - N[B]) = \emptyset$ or $S' \cap (N(B) - N[A]) = \emptyset$ holds and any newly added edge ab is adjacent to a vertex not in S'. Conversely for any independent set S' in $G^\dagger$, S' satisfies $S' \cap (N(A) - N[B]) = \emptyset$ or $S' \cap (N(B) - N[A]) = \emptyset$ (since $N(A) - N[B]$ and $N(B) - N[A]$ induce a complete bipartite graph between them in $G^\dagger$), and $S' \cup A$ or $S' \cup B$ is also an independent set in G. Therefore $\eta(G) = \eta(G^\dagger) + |A|$. ∎

A vertex a together with its neighbors $N(a)$ is called a *funnel* (or a $\delta(a)$-funnel) if $N[a] - b$ induces a complete graph for some $b \in N(a)$, and is denoted by b-a-$(N(a) - b)$ (see Fig. 2(c) for a 3-funnel b-a-$\{c, d\}$).

Lemma 3 *For a funnel b-a-$(N(a) - b)$, there are only two cases: (i) every maximum independent set of G contains b; (ii) there is a maximum independent set of G which contains a. Hence $A = \{a\}$ and $B = \{b\}$ are alternative.*

Proof. Assuming that there is a maximum independent set S_G of G which does not contain b, we show that (ii) holds. If $S_G \cap \{a, b\} = \emptyset$, then it must hold $|S_G \cap N(a) - b| \le 1$ (since $N(a) - b$ induces a clique). On the other hand, at least one of a's neighbors, say c, must be in S_G, otherwise we can add a into S_G to get a bigger independent set. Therefore, c is the only neighbor of a that is in S_G and we can replace c with a in S_G to obtain another maximum independent set which contains a. ∎

A funnel b-a-$(N(a) - b)$ is called *short* if $N(b) \cap N(a) = \emptyset$ and there are at most $\delta(b)$ pairs of nonadjacent vertices $u \in N(b) - a$ and $v \in N(a) - b$ (note that $N(b) \cap N(a) = \emptyset$ holds when a dominates no vertex in $N(b) - a$).

A chordless 4-cycle $u_1u_2u_3u_4$ with four vertices of degree at least 3 is called a *desk* if $A = \{u_1, u_3\}$ and $B = \{u_2, u_4\}$ have no common neighbor and each of them has at most two neighbors outside the cycle; i.e., $N(A) \cap N(B) = \emptyset$ and $|N(A) - B|$, $|N(B) - A| \le 2$ (see Fig. 2(d)).

Lemma 4 *For a desk $u_1u_2u_3u_4$ in a graph G, sets $A = \{u_1, u_3\}$ and $B = \{u_2, u_4\}$ are alternative.*

Proof. We show that G has a maximum independent set S_G satisfying $|S_G \cap \{u_1, u_2, u_3, u_4\}| = 2$, which clearly implies $S_G \cap \{u_1, u_2, u_3, u_4\} = \{u_1, u_3\}$ or $\{u_2, u_4\}$. Let S_G be a maximum independent set of G with $|S_G \cap \{u_1, u_2, u_3, u_4\}| \le 1$, where $S_G \cap \{u_2, u_3, u_4\} = \emptyset$ is assumed without loss of generality. Then we get another maximum independent set $S' = (S_G - N(A)) \cup A$ with $|S'| \ge |S_G|$ and $|S_G \cap \{u_1, u_2, u_3, u_4\}| = 2$ (note that if $u_1 \in S_G$ then $|S_G \cap (N(A) - B)| \le 1$ since the degree of u_1 is at least 3). ∎

Folding alternative subsets A and B is to replace an instance G with $G^\dagger$. In our algorithm, we use only alternative sets A and B such that $A = \{a\}$ and $B = \{b\}$ for a short funnel b-a-$(N(a) - b)$; or $A = \{u_1, u_3\}$ and $B = \{u_2, u_4\}$ for a desk $u_1u_2u_3u_4$, both of which can be found in polynomial time.

Reduction by folding line graphs

If a graph H is the line graph of a graph H', then a maximum independent set S of H can be obtained as the set of vertices that corresponds the set of edges in a maximum matching M in H'. 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].

Suppose that an instance G contains a component H which is the line graph of a 3-regular graph. Folding such a line graph H is to discard H from the instance after computing a maximum independent set of H. We can test whether a component H in G is such a line graph or not in polynomial time, since a graph is the 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.

Reduction by folding small components
A component H with at most 20 vertices in an instance G is called *small.* Folding small component H is to discard H from the instance after computing a maximum independent set of H.

Definition 5 *An instance is called* reduced*, if none of the above reduction rules is applicable.*

We can test whether each of the above reduction rules is applicable to an instance or not in polynomial time. Then an instance can be reduced to a reduced one in polynomial time.

2.2 Branching Rules

Next we introduce our branching rules in a reduced instance G. 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. *Branching on a vertex* v means creating two subinstances by excluding v from the independent set or including S_v into the independent set. In the first branch we will delete v from the instance whereas in the second branch we will delete $N[S_v]$ from the instance.

We also use the following two branching techniques, *branching on a* 4*-cycle* and *branching on a funnel*, which are simple and obvious, but can be used to avoid tedious branching rules in the algorithms.

Branching on 4**-cycles**

Lemma 6 *Let abcd be a* 4*-cycle in an instance* G*. Then 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 6, we get the following branching rule. *Branching* on a 4-cycle *abcd* means branching by either excluding vertices $\{a, c\}$ or excluding $\{b, d\}$ from the independent set. Hence we generate the two subinstances by removing either $\{a, c\}$ or $\{b, d\}$ from an instance G.

Branching on funnels
Based on Lemma 3, we get the following branching rule. *Branching* on a funnel b-a-$(N(a) - b)$ in a reduced instance G by either including a or including S_b in the independent set. Hence we generate the two subinstances by removing either $N[a]$ or $N[S_b]$ from G.

3 Preliminary Analysis

Our algorithm for the maximum independent set problem first applies reduction rules repeatedly until a reduced instance is obtained, and then branches on one of a funnel, a 4-cycle and a vertex of maximum degree. However, we choose a vertex/funnel/4-cycle which we branch on carefully so that the two resulting subinstances can have smaller 'measure.' Next, we first define our measure and analyze some properties of it.

3.1 The measure

The measure of the problem is used to analyze the time complexity of the algorithm. When we apply a branching rule, we will get a recurrence relation related to the measure. We will require the measure satisfying: (i) the problem can be solved directly (in polynomial time), when the measure is not greater than 0; (ii) the measure will not increase when any reduction rule is applied; and

(iii) when any branching rule is applied, the measure in each sub instance becomes smaller. For worst-case analysis, we will design the algorithm such that the worst recurrence in the algorithm is better and then get a better running time bound.

In this paper, we set $r = \rho(V)$ as the measure of graph $G = (V, E)$. Recall that $\rho(V) = \sum_{v \in V} \rho(v)$ and $\rho(v) = \max\{0, \delta(v) - 2\}$. When measure $r \leq 0$, the graph has only degree-0, degree-1 and degree-2 vertices and the maximum independent set problem can be solved in linear time. Next, we consider the reduction and branching operations. To make the measure reduction clear, we adopt a notation to indicate how much r decreases from a vertex or a set of vertices in an operation. For a subset X of vertices in G, we use $X \to t$ to mean that $\rho(X)$ decreases by at least t by an application of a reduction or branching operation.

3.2 Analysis on reduction operations

We call any of an o-path, a vertex of degree ≥ 4 and a cycle containing at most 4 vertices of degree ≥ 3 and at least one vertex of degree ≥ 3 a *fine local structure*. Note that a reduced graph may contain some fine local structures.

Lemma 7 *Let G be an instance (not necessarily reduced), and G' be a reduced instance obtained from G by applying all of our reduction rules in Section 2.1. Then:*

(i) $\rho(G) \geq \rho(G')$.

(ii) $\rho(G) - \rho(G') \geq 1$ *if G has a fine local structure, but G' has no longer any fine local structure.*

Proof. (i) Clearly the measure never increases by removing unconfined vertices or folding degree-1 and -2 vertices and twins. It suffices to show that folding short funnels and desks also does not increase the measure either. Consider a short funnel b-a-$(N(a) - b)$ (the case of desks can be treated analogously). Removing $\{a, b\}$ decreases the measure r by $2\delta(a) + 2\delta(b) - 6 \geq 2\delta(b)$ (since G has no degree-1, -2 or dominated vertices in Step 6), and adding new k edges between $N(b)$ and $N(a)$ increases the measure by at most $2k$. Since the funnel is short, it holds $k \leq \delta(b)$ and hence folding a short funnel does not increases r.

(ii) Clearly $\rho(G) \geq 1$ since G contains a fine local structure. Note that G' is a 3-regular graph with at least 21 vertices that has neither triangles nor 4-cycles. Removing unconfined vertices of degree ≥ 3 decreases the measure. We assume that only folding degree-1, -2 vertices, desks, funnels or discarding a component has been applied to G to obtain G'. Discarding a component H with $\rho(H) \geq 1$ during the application of reduction rules proves the lemma. Discarding a component H with $\rho(H) = 0$ (i.e., H is a component of path or cycle) still leaves any of the existing fine local structures. Folding desks or funnels always leaves a 4-cycle if the measure does not decrease. Folding a degree-1 vertex v cannot eliminate any of fine local structures unless the unique neighbor u of v is of degree ≥ 3 or u is a degree-2 vertex adjacent to a vertex of degree ≥ 3 (the measure will decrease in this case). Folding a degree-2 vertex v does not decrease the measure only when the two neighbors $u_1, u_2 \in N(v)$ are not adjacent and has no common neighbor other than v (note that when u_1 and u_2 are adjacent one of them is of degree ≥ 3, since otherwise $N[u]$ has been discarded as a small component). Folding such a degree-2 vertex v leaves a contracted vertex of degree $\delta(u_1) + \delta(u_2) - 2 \geq 4$ unless one of u_1 and u_2 is a degree-2 vertex. In this case, folding a degree-2 vertex just can decrease the number of degree-2 vertices in an o-path or e-path by exactly 2 without eliminating any of a vertex of degree ≥ 4, an o-path and a cycle containing $k \in [1, 4]$ vertices of degree ≥ 3. Therefore, the fine local structures of G cannot be eliminated without decreasing the measure. ∎

Observe that the measure of an instance with no degree-1 vertices is always an even integer. Hence for two such instances G and G' and an odd integer Δ, $\rho(G) - \rho(G') \geq \Delta$ always implies that $\rho(G) - \rho(G') \geq \Delta + 1$.

3.3 Analysis on basic branching operations

We use $C(r)$ to denote the worst-case size of the search tree in our algorithm when the measure of the graph is r, and consider how much the measure can decrease in each branch of our search tree. We analyze each branching rule applied to an instance G to obtain new instances G_1 and G_2 from the two branches, and derive a recurrence in the form of

$$C(r) \leq C(r - \Delta_1) + C(r - \Delta_2), \tag{1}$$

where Δ_i is a lower bound on $\rho(G) - \rho(G_i)$ for each i. For convenience, we always assume that G_1 is the instance with $\Delta_1 \leq \Delta_2$. Our first target is to make Δ_1 and Δ_2 in (1) as large as possible.

One possible way to do this is to require the algorithm first branching on funnels and 4-cycles if they exist and then branching on a vertex of maximum degree. In fact, funnels and 4-cycles catch some local structures of the graphs and we may get a good recurrence. However, when the reduced graph has none of funnels, 4-cycles and vertices of degree ≥ 4, we may meet a worst case. For this case, we are forced to branch on a degree-3 vertex v by removing either v or $N[S_v] = N[v]$, which may only decrease the measure r by $\rho(v) + |N(v)| = 4$ and $\rho(N[v]) + |N_2(v)| \geq 10$, respectively, as will be analyzed in Lemma 13. Then we can get only recurrence

$$C(r) \leq C(r - 4) + C(r - 10), \tag{2}$$

which solves to $C(r) = 1.1120^r$. Even if the recurrences from the other branching rules are better than (2), we cannot get our claimed bound $C(r) = 1.0836^r$ unless we device a way of maintaining instances so that such a "bottleneck branching" will not frequently occur. Our second target is to try to make the two subinstances G_1 and G_2 corresponding to (1) will not branch with the worst recurrence (2) directly.

A reduced instance with no fine local structure is called a *bottleneck instance*. Contrary to this, we call an instance G with no degree-1 vertices (not necessarily reduced) a *fine instance* if it contains at least one fine local structure. Note that it is possible that we can branch on a bottleneck graph with a recurrence better than (2) but it is sure that we can get a recurrence better than (2) on a fine instance. We also note that by Lemma 7 the measure of a fine instance G will decrease by at least 2 if G becomes a bottleneck instance after simply applying some reduction rules.

4 The Algorithm

We may reduce the instance directly by applying reduction rules. So when a reduction rule can be applied, we just apply it. For the worst case, we may only get reduced graph in each step and no reduction rule can be used. However, branching rules may affect the running time greatly. In a reduced graph, we should choose a 'good' vertex/funnel/4-cycle to branch such that the corresponding recurrence (1) satisfies the above two targets. We define the following special cases of vertices/funnels/4-cycles.

The vertex a of a 3-funnel b-a-$(N(a) - b)$ is called *effective* if a has three degree-3 neighbors and $\rho(G) - \rho(G - N[S_a]) \geq 20$.

(1) A funnel is called *optimal* if (i) it is a 4-funnel; (ii) when there is no 4-funnel, a 3-funnel b-a-$\{c, d\}$ is called *optimal* if $\delta(b) \geq 4$; (iii) when there is no funnel of (i) or (ii), a 3-funnel b-a-$\{c, d\}$ is called *optimal* if b is in a triangle; (iv) when there is no funnel of (i)-(iii), a 3-funnel b-a-$\{c, d\}$ is called *optimal* if there is a vertex of degree ≥ 4 in $N(\{a, b\})$; and (v) when there is no funnel of (i)-(iv), a 3-funnel b-a-$\{c, d\}$ is called if $G - N[b]$ is a fine instance (such a 3-funnel exists by Lemma 8 below);

(2) A 4-cycle $abcd$ is called *optimal* if (i) $\delta(a) = \delta(c) = 3$ (or $\delta(b) = \delta(d) = 3$); and (ii) when there is no such 4-cycle of (i), the number of degree-3 vertices in the cycle is maximized; and

(3) A vertex v of maximum degree d is *optimal* if (i) it maximizes $|N_2(v)|$ for $d \geq 4$; or (ii) it maximizes the number of o-paths in $G - N[v]$ for $d = 3$.

Lemma 8 *Let G be a reduced instance that has neither an effective vertex nor a 4-funnel. Assume that G has a 3-funnel and that for every 3-funnel b-a-$\{c, d\}$, b is not in a triangle and each vertex in $N[\{a, b\}]$ is of degree 3. Then there is a 3-funnel b-a-$\{c, d\}$ such that $G - N[b]$ is a fine instance.*

Proof. Consider a triangle $v_1v_2v_3$ containing a degree-3 vertex. Then each u_i-v_i-$(N(v_i) - u_i)$ is a 3-funnel, where u_i is the third neighbor of v_i. We show that one of $G_i = G - N[u_i]$ $(i = 1, 2, 3)$ is a fine instance or one of v_1, v_2 and v_3 is an effective vertex. Let u'_i and u''_i be the two neighbors of u_i other than v_i, where u'_i and u''_i are also degree-3 vertices by assumption. We observe the following properties for each $i = 1, 2, 3$:

(P1) There are 6 edges between $N(u_i)$ and $N_2(u_i)$;

(P2) There is no edge between $\{u'_i, u''_i\}$ and $\{v_1, v_2, v_3\}$ (otherwise one of v_1, v_2 and v_3 would be a dominated vertex or a roof); and

(P3) If u'_i and u''_i have a common neighbor $u^* \in N_2(v_i)$, the degree of u^* is at least 4 (otherwise $v_iu'_iu^*u''_i$ would be a desk).

First, we show that the lemma holds when u_1, u_2 and u_3 have a common neighbor $u'_1 = u'_2 = u'_3$. Let a and b be the two neighbors of u''_1 other than u_1. If the graph G_1 is not a fine instance, then there are only three cases in G: (i) both of a and b are degree-4 vertices; (ii) vertices a and b are two adjacent degree-3 vertices; and (iii) it holds $\{a, b\} = \{u''_2, u''_3\}$. The reason is based on the following observation: graph G_1 has at most two e-paths if it has no o-path. When G_1 has only one e-path of length 4, which is $u''_2u_2v_2v_3u_3u''_3$, we get Case (i). When G_1 has two e-paths, we get Case (ii). When G_1 has only one e-path of length 6, we get Case (iii).

Case (i): If neither of G_2 and G_3 contains a degree-4 vertex, then both of u''_2 and u''_3 are adjacent to a and b. However, we can see that there is a 4-cycle $u''_2au''_3b$ containing at least one degree-3 vertex in G_1, which implies that G_1 is a fine instance. Case (ii): Now u''_1ab is a triangle. If G_2 (resp., G_3) does not contain triangle u''_1ab, then at least one vertex in the triangle is in $N[u_2]$ (resp., $N[u_3]$). This means that $\{a, b\} = \{u''_2, u''_3\}$ holds and the component containing v_1 has only 10 vertices, contradicting that every component in a reduced instance has at least 21 vertices. Case (iii): We know that there are two edges $u''_1u''_2$ and $u''_1u''_3$. By switching the indices i of v_i, we can also get an edge $u''_2u''_3$. Therefore, the component containing v_1 has only 10 vertices, again a contradiction.

Second we show that the lemma holds when a pair of u_1, u_2 and u_3, say u_1 and u_2, has a common neighbor $u'_1 = u'_2$. Since we have already proved the case where u_3 is also adjacent to u'_1, we assume that u'_1 is not a neighbor u_3. We look at G_3. There is a 5-cycle $v_1v_2u_2u'_1u_1$ in G_3. We see that at least one of u_1, u_2 and u'_1 is of degree ≥ 3 in G_3, since otherwise each of them is adjacent to a or b, and a or b, say b is adjacent to "u_1 and u_2" but the case of $b = u'_3 = u'_1 = u'_2$ has been discussed in the first case (note that b is not adjacent to "u_1 and u'_1" since G has no triangle $bu_1u'_1$ by assumption). Then G_3 is a fine instance.

Now we assume that the graph G has the fourth property:

(P4) For $1 \leq i < j \leq 3$, it holds $N[u_i] \cap N[u_j] = \emptyset$.

Third, we show that if a pair of vertices in $\{u'_1, u'_2, u'_3, u''_1, u''_2, u''_3\}$ are adjacent, the lemma holds. Note that u'_i and u''_i are not adjacent, otherwise it becomes Case (ii) in the above. Next we assume that u'_1 and u'_2 are adjacent without loss of generality. It is clear that there is a 6-cycle $v_1u_1u'_1u'_2u_2v_2$ with at least two degree-2 vertices u_1 and u_2 in G_3. Therefore G_3 is a fine instance. Let $B_i = N(u'_i) \cup N(u''_i) - \{u_i\}$. We assume the fifth property on G:

(P5) For each $i = 1, 2, 3$, it holds $B_i \cap (N[u_1] \cup N[u_2] \cup N[u_3]) = \emptyset$.

Forth, we get two more properties on B_i. If u'_i and u''_i have a common neighbor $u^*_i \in B_i$, then graph G_j ($j \neq i$) is a fine instance since there is a 4-cycle $u_i u'_i u''_i u^*_i$ in G_j. Then we assume the following (P6).

(P6) For each $i = 1, 2, 3$, vertices u'_i and u''_i have no common neighbor other than u_i.

If there is a degree-3 vertex $u^*_i \in B_i$, then u^*_i should also be adjacent to a degree-3 vertex $u^{**}_i \in B_i$ otherwise u^*_i will become an o-path in G_i and we are done. If u^*_i is also adjacent to a vertex in B_j with $j \neq i$, then G_i has a 5-cycle $u_i u'_i u^*_i u^{**}_i u"_i$, where u^*_i is a degree-2 vertex and u_i is a degree-3 vertex in G_i, implying that G_i is a fine instance. Therefore, we know that if $B_i \cap B_j \neq \emptyset$ ($i \neq j$) then any vertex in $B_i \cap B_j$ should be a vertex of degree ≥ 4. We get the following (P7).

(P7) For some $1 \leq i < j \leq 3$, if a vertex a is adjacent to both of a vertex in $\{u'_i, u''_i\}$ and a vertex in $\{u'_j, u''_j\}$, then a is a vertex of degree ≥ 4.

Our last target is to prove that when the above five properties hold, vertex v_1 is an effective vertex. Note that $V_1 = \{v_1, u_2, u_3\} \subseteq S_{v_1}$. It is easy to see that $N[V_1] = \{v_1, v_2, v_3, u_1, u_2, u_3, u'_2, u'_3, u''_2, u''_3\}$ is a set containing 10 different vertices (by (P4)) and there are exact 10 edges between $N[V_1]$ and $N(N[V_1])$ in G (by (P5)). Furthermore, by (P6) and (P7) we know that for each vertex in $N(N[V_1])$ either it is adjacent to one vertex in $N[V_1]$ or it is a vertex of degree ≥ 4 and adjacent to two vertices in $N[V_1]$. Therefore, after removing $N[V_1]$ from G the measure$r$ decreases by 20 by $N[V_1] \to 10$ and $N(N[V_1]) \to 10$, which implies that v_1 is an effective vertex. ∎

An entire description of our algorithm is given in Figure 3. We first apply the reduction rules to reduce the current instance G (**Step** 1-6). Then when a reduced instance G contains an effective vertex in a 3-funnel, we branch on it in **Step** 7. When a reduced instance G contains no effective vertices, but has a 3- or 4-funnel, we branch on an optimal funnel in **Step** 8. When a reduced instance G has no 3- or 4-funnel, but contains a 4-cycle, we branch on an optimal 4-cycle in **Step** 9. Finally, a reduced instance G has neither 3-, or 4-funnels nor 4-cycles and we will select an optimal vertex of maximum degree to branch on (**Step** 10).

5 The Analysis

Now we are ready to analyze each branching operations in the algorithm.

Lemma 9 *Let v be an effective vertex in a reduced instance G. Then branching on it in Step 7 decreases the measure r of G at least with*

$$C(r) \leq C(r-4) + C(r-20), \tag{3}$$

where instance G_1 in (3) is a fine instance.

Proof. We branch on v by excluding it from the independent set or including S_v in the independent set. By definition, vertex v is in a triangle vuu' and the remaining degree-3 neighbor $w \in N(v)$ has no common neighbor with v (otherwise $N(v)$ would contain a dominated vertex). In the branch of removing v, measure r decreases by at least 4, and the neighbor w becomes an o-path in $G - \{v\}$ without leaving degree-1 vertices. Hence $G_1 = G - \{v\}$ is a fine instance. In the other branch of removing $N[S_v]$, measure r decreases by at least $\rho(G) - \rho(G - N[S_v]) \geq 20$ by the definition of effective vertices. ∎

Input: An instance G.
Output: The size of a maximum independent set in G.

1. **If** $\{G$ has a component $H = (V_H, E_H)$ that has at most 20 vertices or is the line graph of a 3-regular graph$\}$, **return** $MIS(G - V_H) + \eta(H)$.
2. **Elseif** $\{$there is a degree-1 vertex$\}$, **return** $MIS(G') + 1$ for the instance G' obtained by folding a degree-1 vertex.
3. **Elseif** $\{$there is a degree-2 vertex$\}$, **return** $MIS(G') + 1$ for the instance G' obtained by folding a degree-2 vertex.
4. **Elseif** $\{$there is an unconfined vertex $v\}$, **return** $MIS(G - v)$.
5. **Elseif** $\{$there is a twin$\}$, **return** $MIS(G') + 2$ for the instance G' obtained by folding a twin.
6. **Elseif**$\{$there is a short funnel or a desk$\}$, **return** $MIS(G^\dagger)$ for the instance $G^\dagger$ obtained by folding a short funnel or a desk.
7. **Elseif** $\{$there is an effective vertex $v\}$ $\max\{MIS(G - v), MIS(G - N[S_v]) + |S_v|\}$.
8. **Elseif** $\{$there is a 3- or 4-funnel$\}$, pick up an optimal funnel b-a-$(N(a) - b)$ and **return** $\max\{MIS(G - N[a]) + 1, MIS(G - N[S_b]) + |S_b|\}$.
9. **Elseif** $\{$there is a 4-cycle$\}$, pick up an optimal 4-cycle $abcd$ and **return** $\max\{MIS(G - \{a, c\}), MIS(G - \{b, d\})\}$.
10. **Else** pick up an optimal vertex v of maximum degree, and **return** $\max\{MIS(G - v), MIS(G - N[S_v]) + |S_v|\}$.

Note: The algorithm can be easily modified to deliver a maximum independent set of G.

Figure 3: The Algorithm $MIS(G)$

Lemma 10 *Let G be a reduced instance that has no effective vertex. Branching on an optimal funnel in G in Step 8 decreases the measure r with one of the following recurrences:*

$$C(r) \le C(r - 10) + C(r - 12), \tag{4}$$

$$C(r) \le C(r - 8) + C(r - 12), \tag{5}$$

$$C(r) \le C(r - 10) + C(r - 10), \ \textit{and} \tag{6}$$

$$C(r) \le C(r - 8) + C(r - 10), \tag{7}$$

where instance G_1 in (5) is a fine instance, at least one of G_1 and G_2, say G_1, in (6) is a fine instance, and both G_1 and G_2 in (7) are fine instances.

Proof. First observe properties on a funnel b-a-$(N(a) - b)$ in G:

(P1): $N(b) \cap N(a) = \emptyset$ (otherwise a would be dominated);

(P2): each neighbor $b' \in N(b)$ has a neighbor $w \in N_2(b)$ (otherwise b' would dominate b).

We branch on an optimal funnel b-a-$(N(a) - b)$ by removing either $N[a]$ or $N[S_b]$ from G. We distinguish five cases according to cases (i)-(v) in the definition of optimal funnels.

Case (i). b-a-$(N(a)-b)$ is a 4-funnel b-a-$\{c,d,e\}$: In the branch of removing $N[a]$, the measure r of G will decrease by at least 11 (hence by at least 12 after folding any resulting degree-1 vertices) by $N[a] \to 9$ and $N(b)-a \to 2$. By (P2), in the other branch of removing $N[S_b]$, measure r will decrease by at least 10 by $N[b] \to 5$ and $\{w\} \cup \{c,d,e\} \to 4$ (note that $\rho(w) \geq 2$ when $w \in \{c,d,e\}$).

In the following, let b-a-$\{c,d\}$ be the optimal 3-funnel called by the algorithm. Let c' and d' denote the third neighbor of c and d not in $\{a,b,c\}$, and c'' and d'' denote the fourth neighbor (if any) of c and d. We here observe the following properties:

(P3): If $\delta(c)=3$ then $c' \notin N(d)$ and $N[c] \cap N(d) - \{a,c\} = \emptyset$ (otherwise c would dominate d);

(P4): If $\delta(c)=3$ and $(N(c')-c) \cap N(d) \neq \emptyset$, then $\delta(c') \geq 4$ (otherwise c'-c-$\{a,d\}$ would be a short funnel);

(P5): If $\delta(c)=\delta(d)=3$ then no two of c', d' and b are adjacent (otherwise $\{a,b,c\}$ would contain a roof), and hence $N(N(b)-a)-b) \cap \{c,d\} = \emptyset$; and

(P6): at least *two* neighbors of $N(b)-a$ are not adjacent to any of c and d, and hence $N_2(b) - \{c,d\} \neq \emptyset$: This is because a non-short funnel b-a-$\{c,d\}$ has at least $\delta(b)+1$ pairs of nonadjacent vertices between $N(b)-a$ and $\{c,d\}$ (i.e., it has at most $2(\delta(b)-1)-(\delta(b)+1)=\delta(b)-3$ edges between them).

By (P6), removing $N[a]$ decreases r by at least $N[a] \to \delta(b)+\delta(c)+\delta(d)-5$, $N(b)-a \to 2$ and $N(\{c,d\})-a \to |N(\{c,d\})-a| \geq 2$. Note that $|N(\{c,d\})-a| \geq 3$ if $\delta(c)+\delta(d) \geq 7$ (otherwise $\{c,d\}$ would contain a dominated vertex). Hence removing $N[a]$ decreases r by at least 11 (hence 12) if "$\delta(c)+\delta(d) \geq 8$" or "$\delta(b) \geq 4$ and $\delta(c)+\delta(d) \geq 7$."

Case (ii). $\delta(b) \geq 4$: We branch by removing $N[a]$ or $N[S_b]$.

(ii-1) $\delta(b) \geq 4$ and $\delta(c)+\delta(d) \geq 7$: As observed in the above, the first branch of removing $N[a]$ decreases r by at least 12 in this case. By (P6), we get $N_2(b)-\{c,d\} \neq \emptyset$. In the second branch of removing $N[S_b]$, measure r decreases by at least 10 ($\rho(N[b])+|N_2(b)-\{c,d\}|+|\{c,d\}| \geq 7+2=9$). In this case, (4) holds.

(ii-2) $\delta(b) \geq 4$ and $\delta(c)=\delta(d)=3$: Analogously with (ii-1), we see that the first branch of removing $N[a]$ decreases r by at least 10. We show that the second branch of removing $N[S_b]$ decreases r by at least 12 to obtain (4). By (P5), $N(N(b)-a)-b) \cap \{c,d\} = \emptyset$. Note that $\rho(N[b]) \geq 6$ and $|N(N(b)-a)-b)| \geq 2$. If $\rho(N[b])+|N(N(b)-a)-b)| \geq 9$, then the second branch decreases r by at least $\rho(N[b])+|N(N(b)-a)-b)|+|\{c,d\}| \geq 9+2=11$. Assume that $\rho(N[b])+|N(N(b)-a)-b)| \leq 8$, i.e., $\delta(b)=4$, $|N(N(b)-a)-b)|=2$ and $N(b)-a$ consists of three degree-3 vertices, where there are a triangle bxx' and two 3-funnels z-x-$\{b,x'\}$ and z'-x'-$\{b,x\}$ for some $x,x' \in N(b)-a$ and $\{z,z'\}=N(N(b)-a)-b)$. Then $\delta(z),\delta(z') \geq 4$ (otherwise these 3-funnels would be short). Now removing $N[b]$ decreases r by at least 12 ($\rho(N[b]) \to 6$, $N(N(b)-a)-b) \to 4$ and $\{c,d\} \to 2$).

In Cases (iii)-(v), it holds $\delta(b)=3$, and we denote $N(b)=\{a,b',b''\}$, which always satisfies:

(P7): $N(\{b',b''\}) \cap \{c,d\} = \emptyset$ (otherwise b-a-$\{c,d\}$ would be a short funnel).

By the definition of optimal funnels, if $\delta(c)=3$ (resp., $\delta(d)=3$) then $\delta(c')=3$ (resp., $\delta(d')=3$).

Case (iii). $\delta(b)=\delta(a)=3$ and b is in a triangle: Note that a-b-$\{b',b''\}$ is also a 3-funnel. For notational convenience, we analyze branching on funnel a-b-$\{b',b''\}$ by removing (A) $N[S_a]$ or (B) $N[b]$ without loss of generality.

(A) The first branch of removing $N[S_a]$: We show that r decreases by at least 11 after removing $N[S_a]$ (hence by 12 by folding degree-1 vertices). As already observed, removing $N[a]$ decreases r by at least 12 if $\delta(c)+\delta(d) \geq 8$. We distinguish the two remaining cases.

(A-1) $\delta(c)+\delta(d)=7$, say $\delta(d)=4$: Then $\{a,c'\} \subseteq S_a$. We see that $\{b',b''\}-N(c') \neq \emptyset$ (otherwise child b of $S=\{a,c'\} \subseteq S_a$ would satisfy $N(b)-N[S]=\emptyset$, contradicting that a is confined). Recall that $\delta(c')=3$ and $N[c'] \cap N(d)-\{a,c\}=\emptyset$ by (P3) and (P4). In the first branch, we remove $N[\{a, c'\}] \subseteq N[S_a]$ from G and r decreases by at least $\rho(N[a]) + \rho(N[c'] - N(d)) + |\{b', b''\} - N(c')| + |N(d) - \{a, c\}| \geq 5 + 3 + 1 + 2 = 11$.

(A-2) $\delta(c) = \delta(d) = 3$: Then $\{a, c', d'\} \subseteq S_a$. We distinguish two subcases.

(A-2-i) One of b' and b'' (say b'') is adjacent to one of c' and d' (say c'): Now $\{a, c', d', b'\} \subseteq S_a$ holds and hence b' is not adjacent to d'. Also $\delta(b'') \geq 4$ (otherwise child b'' of $S = \{c, b, d'\} \subseteq S_c$ would satisfy $N(b'') - N[S] = \emptyset$). Then r decreases by at least $\rho(N[\{a, c', d', b'\}]) \geq \rho(\{a, b, c, d, c', d', b''\} \cup (N(d') - d)) \geq 10$ (note $\rho(b'') \geq 2$) and $|N(N[\{a, c', d', b'\}])| \geq |\{b'\}| = 1$.

(A-2-ii) Neither of b' and b'' is adjacent to any of c' and d': In this case, if $|N(\{c', d'\}) - \{c, d\}| \geq 3$ then the first branch removes $N[\{a, c', d'\}]$ and decreases r by at least $\rho(N[\{a, c', d'\}]) + |N(N[\{a, c', d'\}])| \geq 9 + |\{b', b''\}| = 11$. Assume that $|N(\{c', d'\}) - \{c, d\}| = 2$. We show that there exists a vertex $x \in N[\{a, c', d'\}] - \{b', b''\}$, which implies that $\rho(N[\{a, c', d'\}]) + |N(N[\{a, c', d'\}])| \geq 8 + |\{b', b'', x\}| = 11$. A neighbor $z \in N(\{c', d'\}) - \{c, d\}$ is a child of $S = \{c, d', b\} \subseteq S_c$ and must have a neighbor $x \in N(z) - N[S]$. Note that $x \notin \{b', b''\} \subseteq N[S]$ and $x \in N[\{a, c', d'\}] - \{b', b''\}$, as required.

(B) The second branch of removing $N[b]$: We distinguish two cases.

(B-1) $\delta(b') + \delta(b'') \geq 7$: Then the second branch decreases r by at least 10 ($\rho(N[b]) + |N(\{b', b''\}) - b| + |\{c, d\}| \geq 5 + 2 + 2 = 9$). In this case, (4) holds.

(B-2) $\delta(b') = \delta(b'') = 3$: Then $|N(\{b', b''\}) - b| \geq 2$. Then the second branch decreases r by at least $\rho(N[b]) + |N(\{b', b''\}) - b| + |\{c, d\}| \geq 4 + 2 + 2 = 8$. When r decreases only by 8, no degree-1 vertex appears in $G - N[b]$ by $|N(\{b', b''\}) - b| = 2$ and the vertices $x' \in N(b') - \{b', b''\}$ and $x'' \in N(b'') - \{b', b''\}$ are degree-3 vertices by the definition of optimality of funnels, which will form o-paths in $G - N[b]$. In this case, (5) holds.

Case (iv). $\delta(b) = 3$, vertex b is not in a triangle and one of b', b'', c' and d' is of degree ≥ 4: It holds that

(P8): $|N(\{b', b''\}) - b| \geq 3$ (otherwise $\{b', b''\}$ would be in a twin).

In each case, we branch by including either a or S_b, where $S_b = b$ holds. We distinguish three cases.

(iv-1) $\delta(c) + \delta(d) \geq 8$: As observed in the above, the first branch of removing $N[a]$ decreases r by at least 12 in this case. By (P8) and (P6), the second branch of removing $N[b]$ decreases r by at least $\rho(N[b]) + |N(\{b', b''\}) - b| + |\{c, d\}| \geq 4 + 3 + 2 = 9$ (hence 10). Hence (4) holds in (iv-1).

(iv-2) $\delta(c) + \delta(d) = 7$ (say $\delta(c) = 3$ and $\delta(d) = 4$): In the first branch of removing $N[a]$, r decreases by $\rho(N[a]) + |N(b) - a| + |N(\{c, d\}) - a| = 5 + 2 + 3 = 10$, leaving no degree-1 vertex by (P3). In the second branch of removing $N[b]$, r decreases by at least 10 ($\rho(N[b]) + |N(b) - a| + |\{c, d\}| \geq 4 + 3 + 2 = 9$). If r decreases by at least 12 in the second branch, we have (4). When the second branch decreases r only by 10, it holds $\delta(b') = \delta(b'') = 3$, and we shall claim that $G - N[a]$ or $G - N[b]$ is a fine instance, indicating that (6) holds.

Let $N(d) = \{a, c, d', d''\}$. Note that $\delta(c') = 3$ holds by $\delta(c) = 3$ and c' is not adjacent to any of d' and d'' by (P4). Assume that c' is not an o-path in $G - N[a]$ (otherwise we are done). Then c' is adjacent to one of b' and b'', say b'', and we see that $S_a \supseteq \{a, c', b'\}$ holds, which implies that c' is not adjacent to b'. Also assume that b' is not an o-path in $G - N[a]$, either. Then b' is adjacent to one of d' and d'', say d', and $\delta(d') = 3$ in G. Let $N(d') = \{d, b', y\}$, where $y \neq b''$ (otherwise $bb''d'b'$ would be a desk). We see that y is not adjacent to b'', since otherwise $S_c \subseteq \{c, b, y, b'\}$ would contain adjacent b and b', contradicting that c is confined. This implies that d' is an o-path in $G - N[b]$. This proves the claim.

(iv-3) $\delta(c) = \delta(d) = 3$ and $\delta(b') + \delta(b'') \geq 7$: Note that $\delta(d') = 3$ holds by $\delta(d) = 3$. In the first branch of removing $N[a]$, r decreases by at least $\rho(N[a]) + |N(b) - a| + |\{c', d'\}| = 4 + 2 + 2 = 8$. We show that $G - N[a]$ is a fine instance (without using $\delta(b') + \delta(b'') \geq 7$). In $G - N[a]$, there is no degree-1 vertex, and c and d are nonadjacent degree-2 vertices. If one of c' and d', say c' is adjacent to a degree-3 vertex of b' or b'', say b', then b is a child of $S = \{a, c, d\} \subseteq S_a$ and $\{a, c, d, b''\} \subseteq S_a$ must hold, and b'' cannot be adjacent to d, implying that d' (or $d'b'c'$) is an o-path in $G - N[a]$.

In the second branch of removing $N[b]$, we show that r decreases by at least 12. Let $\delta(b'') \geq 4$ without loss of generality. Removing $N[b]$ decreases r by at least $\rho(N[b]) + |N(\{b', b''\}) - b| + |\{c, d\}| \geq 5 + |N(\{b', b''\}) - b| + 2$, where $|N(\{b', b''\}) - b| \geq 3$ by (P8). Assume that r decreases by at most 10, i.e., $\delta(b'') = 4$, $\delta(b') = 3$, $|N(\{b', b''\}) - b| = 3$ and the two common neighbors $x_1, x_2 \in N(b') \cap N(b'') - b$ are both degree-3 vertices. We here claim that r will further decrease by 2 by folding the degree-1 vertices in $G - N[b]$. Now each x_i is a degree-1 vertex incident to a vertex y_i in $G - N[b]$. To prove the claim, it suffices to show that $y_1 \neq y_2$, $\{x_1, x_2\} \cap \{y_1, y_2\} = \emptyset$ and each y_i is of degree ≥ 3 in $G - N[b]$. We see that $x_i \neq y_j$ (otherwise x_i would be dominated) and $x_i \neq c', d'$ (otherwise, child b of $S = \{a, c', d'\} \subseteq S_a$ would satisfy $N(b) - N[S_a] = \emptyset$). Also $y_i \notin N(b'') - \{x_1, x_2\}$ (otherwise b'-x_i-$\{b'', y_i\}$ would be a short funnel). Therefore each y_i is of degree ≥ 3 in $G - N[b]$, as required.

Case (v). b is not in a triangle, $\delta(b) = \delta(c) = \delta(d) = \delta(b') = \delta(b'') = 3$ and $G - N[b]$ is a fine instance (recall that the definition of optimal funnels and Lemma 8): As observed in Case (iv-3), removing $N[a]$ decreases r by 8, leaving a fine instance $G - N[a]$. In the other branch of removing $N[S_b]$, r decreases by 10 ($\rho(N[b]) + |N(\{b', b''\}) - b| + |\{c, d\}| \geq 4 + 3 + 2 = 9$), leaving a fine instance $G - N[b]$. In this case, we have (7). ∎

Lemma 11 *Let G be a reduced instance with no 3-, 4-funnels. Branching on an optimal 4-cycle in G in Step 9 decreases the measure r at least with (5) leaving a fine instance G_1 or (4).*

Proof. We branch on an optimal 4-cycle $abcd$ by removing either $\{a, c\}$ or $\{b, d\}$ from G. We let a' denote the third neighbor of a not in the 4-cycle, and a'' be the fourth neighbor of a (if any). We use the same notation for b', b'', c', c'', d' and d''. We distinguish three cases

Case (i). There are two nonadjacent degree-3 vertices in the cycle: Let $\delta(b) \geq \delta(d) \geq \delta(a) = \delta(c) = 3$ without loss of generality. Then $a' \neq c'$ (otherwise $\{a, c\}$ would be a twin). Now no edge of $abcd$ is in a triangle (since G has no 3-funnel), and it holds that $\delta(b) \geq 4$ and $|N(\{b, d\}) - \{a, c\}| \geq 3$ (otherwise $abcd$ would be a desk). In the branch of removing $\{b, d\}$, we remove $X = \{b, d, a, a', c, c'\}$ by folding the degree-1 vertices c and d. If $\rho(X) \geq 8$ (hence $|N(X)| \geq 3$) or $|N(X)| \geq 4$, then removing X decreases r by at least $\rho(X) + |N(X)| \geq 8 + 3 = 11$ (or $7 + 4 = 11$). Assume otherwise, $\rho(X) = 7$ (i.e., $\delta(b) = 4$ and $\delta(d) = \delta(a') = \delta(c') = 3$) and $|N(X)| = |\{b', b'', d'\}| = 3$. In this case, d' is adjacent to a' or c' and hence $\delta(d') \geq 4$ (otherwise $d'a'ad$ or $d'c'cd$ would be a desk). This means that r still decreases by at least 11 (hence 12) ($X \to 7$ and $\{b', b'', d'\} \to 4$).

In the other branch of removing $\{a, c\}$, r decreases by $\{a, c\} \to 2$, $\{b, d\} \to \delta(b) + \delta(d) - 4$ and $\{a', c'\} \to 2$, totally $\delta(b) + \delta(d) \geq 7$ (by $\delta(b) \geq 4$). If $\delta(b) + \delta(d) \geq 9$ then r decreases by at least 10. Assume $\delta(b) + \delta(d) \leq 8$. If $\delta(d) = 3$ and $\delta(d') \geq 4$, then r decreases by at least 10 ($N[d] \to 5$, $b \to 2$ and $\{a', c'\} \to 2$). If $\delta(d) = \delta(d') = 3$, then there a neighbor $w \in N(d') - d$ such that $w \neq b$, where $\delta(w) \geq 4$ if $w \in \{a', c'\}$ (otherwise $wd'da$ or $wd'dc$ would be a desk). Hence r still decreases by at least 10 ($N[d] \to 4$, $b \to 2$ and $\{a', c', w\} \to 3$).

In the remaining case of $\delta(b) = \delta(d) = 4$, b and d will be o-paths in $G - \{a, c\}$, since neither of b and d in $G - \{a, c\}$ can be adjacent to other degree-2 vertices (no edge of $abcd$ is contained in a triangle). When r decreases by only 8, we have a fine instance $G_1 = G - \{a, c\}$. This proves that (4) or (5) holds in (i).

Case (ii). The cycle contains a degree-3 vertex, but there is no pair of nonadjacent degree-3 vertices in the cycle: Let $\delta(b) \geq \delta(d) \geq \delta(a) = 3$, and $\delta(c), \delta(b) \geq 4$ without loss of generality. In the branch of removing $\{a, c\}$, r decreases by at least 10 ($\{a, c\} \to 3$, $\{b, d\} \to 4$ and $N(c) - \{b, d\} \to 2$). In the branch of removing $\{b, d\}$, where $N[a]$ will be removed by folding the degree-1 vertex a. When b is adjacent to a neighbor $u \in N(d) - \{a, c\}$ or $u \in N(a') - a$, it holds $\delta(u) \geq 4$ (otherwise 4-cycle $badu$ or $baa'w$ must be optimal). Hence removing $N[a]$ decreases r by at least 12 ($N[a] \to 5$, $c \to 2$ and $N_2(a) - c \to 4$). This proves that (4) holds in (ii).

Case (iii). *abcd* is none of Cases (i)-(ii); i.e, *abcd* contains no degree-3 vertex: We distinguish two subcases.

(iii-a). vertices $\{a,b,c,d\}$ induce a 4-clique: Now $\delta(u) \geq 5$ for all $u \in \{a,b,c,d\}$ (otherwise a vertex u with $\delta(u) = 3,4$ would dominate another vertex or form a 4-funnel). In each of the two branches, say removing $\{a,c\}$, r will decrease by at least 12 ($\{a,c\} \to 6$, $\{b,d\} \to 4$ and $N(a) - \{b,c,d\} \to 2$). This proves that (4) holds in (iii-a).

(iii-b). $\{a,b,c,d\}$ contain two independent vertices, say a and c: By optimality of 4-cycles, G has no 4-cycle containing degree-3 vertices, and $\delta(u) \geq 4$ for all vertices $u \in N(a) \cap N(c) - \{b,d\}$. Hence in the branch of removing $\{a,c\}$, measure r decreases by at least 12 ($\{a,c\} \to 4$, $\{b,d\} \to 4$ and $N(\{a,c\}) - \{b,d\} \to 4$). In the other branch of removing $\{b,d\}$, r will decrease by at least 10 ($\{b,d\} \to 4$, $\{a,c\} \to 4$ and $N(\{b,d\}) - \{a,c\} \to 1$). This proves that (4) holds in (iii-b). ∎

In Step 10, the algorithm will branch an optimal vertex v of maximum degree. We will consider two case: either v is of degree ≥ 4 or v is of degree 3 (where G is a bottleneck instance).

Lemma 12 *Let G be a reduced instance which has neither 3,4-funnels nor 4-cycles. Then branching on an optimal vertex of maximum degree $d \geq 4$ in G decreases the measure r at least with one of the following recurrences:*

$$C(r) \leq C(r-6) + C(r-20); \tag{8}$$

$$C(r) \leq C(r-6) + C(r-14), \tag{9}$$

where instance G_1 in (9) is a fine instance.

Proof. We branch on v by excluding it from the independent set or including S_v in the independent set. In the branch of removing v, we can decrease the measure r by $d-2+d = 2d-2 \geq 6$.

In the other branch of removing $N[v] \subseteq N[S_v]$, we can decrease the measure r by $\rho(N[v]) + |N_2(v)|$. Let k be the number of edges between $N(v)$ and $N_2(v)$. Since there is no 4-cycle passing through v, no two of the k edges meet at the same vertex in $N_2(v)$, and we have $|N_2(v)| = k$.

Since there is no 4-cycle passing through v, at most $\lfloor d/2 \rfloor$ (disjoint) pairs of neighbors of v can be adjacent, and both adjacent neighbors must be of degree ≥ 4 (since no triangle contains v and a degree-3 neighbor). This implies that each neighbor of v has at least two neighbors in $N_2(v)$. Hence removing $N[v]$ decreases r by at least $\rho(N[v]) + k \geq d-2+d+2d \geq 14$.

(i) v has a degree-3 neighbor: In $G - v$, all degree-3 neighbors of v become degree-2 vertices, and no two of them are adjacent or share a common neighbor, since no triangle contains v and a degree-3 neighbor. Hence each degree-2 vertex in $G - v$ separately forms an o-path, and $G - v$ is a fine instance. This proves that (9) holds.

(ii) v has no degree-3 neighbor: For $d \geq 5$, it holds $\rho(N[v]) + k \geq d-2+1+d+2d+1 \geq 20$, where at least one neighbor of v needs to have at least three neighbors in $N_2(v)$ since there is no degree-3 neighbor of v. For $d = 4$, it suffices to show that $k \geq 10$, which implies $\rho(N[v]) + k \geq 2+8+10 = 20$ and (8). Since the reduced graph G has no component of the line graph of a 3-regular graph, there always exists a degree-4 vertex that is adjacent to a degree-3 vertex or not contained in two edge-disjoint triangles. Hence in this case, we see that at most one pair of its neighbors are adjacent by the optimality of v. This shows $k \geq 10$, as required. ∎

Lemma 13 *Let v be an optimal vertex in a bottleneck instance. Then branching on it decreases the measure r of G at least with (2) so that both $G_1 = G - v$ and $G_2 = G - N[v]$ are fine instances.*

Proof. We branch on v by excluding it from the independent set or including it in the independent set. In the branch where v is removed, three independent degree-2 vertices (three o-paths) are created. The remaining graph $G - v$ is always a fine instance. In the other branch where $N[v]$ is removed, the remaining graph $G - N[v]$ has exactly six degree-2 vertices and no degree-1 vertex. Since G has neither a triangle nor a 4-cycle, we see that $G - N[v]$ has no o-path if and only if the six degree-2 vertices form three e-paths, and this is possible only when vertex v is contained in three 5-cycles each pair of which shares exactly two vertices. It can be shown that any 3-regular connected graph with at least 21 vertices contains at least one vertex v for which the above condition does not hold (see Lemma 15 in Appendix for a proof). Since G is a reduced instance whose component has at least 21 vertices, an optimal vertex v has an o-path in $G - N[v]$. ∎

Finally we make an entire analysis over all branching operations. As we remarked, the key idea is how to prevent the "bottleneck branching" with recurrence (2) from successively occurring during an execution of our algorithm. Although our algorithm is described so that it starts from applying reduction rules, we evaluate the changes of measure based on the instant when instances become fine instances during the execution. Consider branching on an optimal funnel in a fine instance G. In this case, G is first reduced to a reduced instance G' (possibly without any decrease in the measure), and two instances G_1 and G_2 are generated with recurrence (4). For each $i = 1, 2$, G_i may not be a fine instance and possibly is reduced to a bottleneck instance G_i'. In G_i', we are forced to branch on a degree-3 vertex v_i in G_i' with (2) generating two instances $G_i' - v_i$ and $G_i' - N[v_i]$, which are both shown to be fine instances by Lemma 13. In total, from a fine instance G, four fine instances are generated with recurrence

$$C(r) \leq C(r - 10 - 4) + C(r - 10 - 10) + C(r - 12 - 4) + C(r - 12 - 10).$$

Analogously for (8), four fine instances are generated with recurrence

$$C(r) \leq C(r - 6 - 4) + C(r - 6 - 10) + C(r - 20 - 4) + C(r - 20 - 10).$$

For (3), (5), (6) and (9), the resulting instance G_1 is shown to be a fine instance, and we only branch on G_2' with (2), and three fine instances are generated with recurrences: $C(r) \leq C(r - 4) + C(r - 20 - 4) + C(r - 20 - 10)$; $C(r) \leq C(r - 8) + C(r - 12 - 4) + C(r - 12 - 10)$; $C(r) \leq C(r - 10) + C(r - 10 - 4) + C(r - 10 - 10)$; and

$$C(r) \leq C(r - 6) + C(r - 14 - 4) + C(r - 14 - 10). \tag{10}$$

Note that the resulting instances G_1 and G_2 in (7) are both fine instances.

We now consider the remaining branching of an optimal degree-3 vertex with (2). Recall that we are given a fine instance G. Then G is reduced to a bottleneck instance G'. Hence $\rho(G) - \rho(G') \geq 2$ by Lemma 7. Therefore branching on an optimal degree-3 vertex in G' generates two fine instances $G_1 = G - v$ and $G_2 = G - N[v]$. Hence from a fine instance, we can generate two fine instances with recurrence

$$C(r) \leq C(r - 2 - 4) + C(r - 2 - 10). \tag{11}$$

Among all above recurrences, the worst recurrences are (10) and (11), which solves to $C(r) = 1.0836^r$.

Theorem 14 *A maximum independent set in a graph with maximum degree* 3 *can be found in* $O^*(1.0836^n)$ *time.*

6 Concluding Remarks

In this paper, we have presented an $O^*(1.0836^n)$-time algorithm for the maximum independent set problem in degree-3 graphs, which improves all previous results on this problem without increasing the number of branching rules. The maximum independent set problem is one of the most extensively studied problems in exact algorithms. The best worst-case behavior of exact exponential solutions to it is an important issue in this area.

Based on the new result on MIS3, we recently improved our algorithm for MIS4 [20] to an $O^*(1.1446^n)$-time algorithm [21]. Combining our results on low-degree graphs with the bottom-up method in [2], we can also improve the best exact algorithm for the maximum independent set problem in general graphs.

In this paper, we use some reduction rules and branching rules, such as confining sets, alternative sets, and branching on a funnel/4-cycle, to avoid tedious examinations of the local structures. These rules catch the structural properties of small cycles in graphs, and simplify the algorithm. It is easy to see that many previous algorithms can apply the new rules to simplify the description and analysis.

Most previous exact algorithms got improvements by carefully checking what will happen next after a worst case. To get a light improvement, we may need to consider a large number of cases, which may make the algorithm too complicated and hard to check. In this paper, we get improvements by avoiding the cases instead of checking what after worst cases. This idea can also be used to design fast algorithms for other problems.

References

[1] Beigel, R., Finding maximum independent sets in sparse and general graphs, SODA, ACM Press, 1999.

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

[4] Bourgeois, N., Escoffier, B., Paschos, V. T., An $O^*(1.0977^n)$ exact algorithm for max independent set in sparse graphs,

[5] Chen, J. Kanj, I., Xia, G., Labeled search trees and amortized analysis: Improved upper bounds for NP-hard problems, Algorithmica 43 (4) (2005) 245–273.

[6] Chor, B., Fellows, M., Juedes, D., Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps, WG2004, LNCS 3353, Springer, 2005.

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

[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, LATIN2006, LNCS 3887, Springer, 2006.

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

[11] Kneis, J., Langer, A., Rossmanith, P., A fine-grained analysis of a simple independent set algorithm, FSTTCS 2009. V. 4 LIPIcs., Dagstuhl, Germany, (2009) 287–298.

[12] Razgon, I., A faster solving of the maximum independent set problem for graphs with maximal degree 3, in: H. Broersma, S. S. Dantchev, M. J. 0002, S. Szeider (eds.), ACiD, vol. 7 of Texts in Algorithmics, King's College, London, 2006.

[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, Journal 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 Journal 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, WALCOM 2010, LNCS 5942, Springer, (2010) 281–292.

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

[20] Xiao, M., Nagamochi, H., Further improvement on maximum independent set in degree-4 graphs. Wang, W., Zhu, X., and Du, D. (Eds.): COCOA'11. LNCS 6831 (2011) 163-178.

[21] Xiao, M., Nagamochi, H., Further improvement on maximum independent set in graphs with maximum degree 4, Technical Report 2012-003, May 8 (2012) url: http://www-or.amp.i.kyoto-u.ac.jp/members/nag/Technical_report/TR2012-003.pdf

Appendix

Lemma 15 *Let G be a 3-regular connected graph with neither a triangle nor a 4-cycle. If every two adjacent edges are contained in a 5-cycle, then the number of vertices in G is either 10 or 20.*

Proof. Let $N_3(v)$ denote the set of vertices whose distance from a vertex v is 3. For each vertex v in G, $|N(v)| = 3$ and $|N_2(v)| = 6$ hold by the assumption. We fix a vertex v, and denote $N(v) = \{a, b, c\}$ and $N_2(v) = \{a_c, a_b, b_a, b_c, c_b, c_a\}$, where vertex $x_y \in N_2(v)$ is adjacent to vertex $x \in N(v)$ and vertex $y_x \in N_2(v)$ for symbols $x, y \in \{a, b, c\}$. Let x'_y be the third neighbor of each vertex $x_y \in N_2(v)$, i.e., $N(x_y) = \{x, y_x, x'_y\}$. The assumption implies that a shortest path $P_{a,b}$ in the graph $G - v$ between two neighbors $a, b \in N(v)$ is of length 3 and unique, and the three shortest paths $P_{a,b}$, $P_{b,c}$ and $P_{a,c}$ form a 9-cycle $aa_bb_abb_cc_bcc_aa_c$.

Case 1. There is a vertex v such that $N_2(v)$ contains a pair of adjacent vertices $s, t \in N_2(v)$ other than a_bb_a, b_cc_b and c_aa_c: Assume $s = a_c$ without loss of generality. Since $t \in \{a_b, b_a, c_b\}$ implies that it would be in a 3- or 4-cycle, it holds $t = b_c$. Then the third 5-cycle C containing a other than aa_cc_acv and aa_bb_abv must pass through vertices $a_baa_cb_c$ and one of c_b and b, and the 9-cycle containing $N(a) \cup N_2(a)$ must be $a_cc_acvbb_aa_bc_bb_c$, indicating that a_b and c_b also need to be adjacent. Therefore by symmetry, c_a and b_a must be adjacent too. Now each vertex in $N[v] \cup N_2(a)$ already receives three edges, and the graph G has only 10 vertices in $N[v] \cup N_2(v)$.

Case 2. There is no vertex v of Case 1: In this case, for each vertex u, $N(v) \cup N_2(u)$ induces a chordless 9-cycle. We first show that $|N_3(v)| = 6$. Assume that $|N_3(v)| < 6$, where a'_c is equal to a vertex in $\{a'_b, b'_a, b'_c, c'_b, c'_a\}$ without loss of generality. If $a'_c = c'_a$ or a'_b then it would be in a 3- or 4-cycle. If $a'_c = c'_b$ or b'_a, say, $a'_c = c'_b$, then c_bc would be a chord of the 9-cycle induced by $N(a_c) \cup N_2(a_c)$. If $a'_c = b'_c$ then the distance a and a'_c in the graph $G - a_c$ is at least 4, contradicting the assumption. Therefore $|N_3(u)| = 6$ for all vertices u in G in Case 2. Since $N(v) \cup N_2(u)$ induces a chordless 9-cycle for each vertex u, G has edges $a'_ca'_b$, $b'_cb'_a$ and $c'_bc'_a$. Let a''_b (resp., a''_c) be the third neighbor of a'_b (resp., a'_c), i.e., $N(a'_b) = \{a_b, a'_c, a''_b\}$ and $N(a'_c) = \{a_c, a'_b, a''_c\}$. We prove $a''_b, a''_c \notin N[v] \cup N_2(v) \cup N_3(v)$. Assume that $a''_b \in N[v] \cup N_2(v) \cup N_3(v)$. Since $(N[v] \cup N_2(v) \cup N_3(v)) - (N[a] \cup N_2(a) \cup N_3(a)) = \{b'_c, c'_b\}$, we see that a''_b is equal to b'_c or c'_b. If $a''_b = c'_b$ then b'_a is also adjacent to c'_b (since $N(a_b) \cup N_2(a_b)$ induces a chordless 9-cycle) and c'_b would be adjacent to four vertices c_b, c'_a, a'_b and b'_a, a contradiction. If $a''_b = b'_c$, then the 9-cycle induced by $N(b_a) \cup N_2(b_a)$ would have a chord $a'_bb'_c$. Hence $a''_b \notin N[v] \cup N_2(v) \cup N_3(v)$. Symmetrically we have $a''_c \notin N[v] \cup N_2(v) \cup N_3(v)$. Now $N(a'_b) \cup N_2(a'_b)$ induces a 9-cycle $a''_bb'_ab_aa_baa_ca'_ca''_ca^*$ for a vertex a^*, and $N(b_c) \cup N_2(b_c)$ induces a 9-cycle $c'_bc_bcvbb_ab'_ab'_cv^*$ for a vertex v^*, where $a^* \neq v^*$ and $a^*, v^* \notin N[v] \cup N_2(v) \cup N_3(v) \cup N[a] \cup N_2(a)$ (otherwise such a vertex would be of degree 4). Finally $N(b'_c) \cup N_2(b'_c)$ must induce the 9-cycle $b_aa_ba'_ba''_ba^*v^*b'_cb_cb$. Then no more edge can be incident to any vertex in $\{a^*, v^*\} \cup N[v] \cup N_2(v) \cup N_3(v) \cup N[a] \cup N_2(a)$, and G has only these 20 vertices. ∎