Theory Comput. Systems **32**, 115–132 (1999)

Theory of Computing Systems

On Approximation Properties of the Independent Set Problem for Low Degree Graphs*

P. Berman[1] and T. Fujito[2]

[1]Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, USA
berman@cse.psu.edu

[2]Department of Electrical Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527, Japan
fujito@huis.hiroshima-u.ac.jp

Abstract. The subject of this paper is the Independent Set problem for bounded node degree graphs. It is shown that the problem remains *MAX SNP*-complete even when graphs are restricted to being of degree bounded by 3 or to being 3-regular. Some related problems are also shown to be *MAX SNP*-complete at the lowest possible degree bounds. We next study a better polynomial time approximation of the problem for degree 3 graphs. The performance ratio is improved from the previous best of $\frac{5}{4}$ to arbitrarily close to $\frac{6}{5}$ for degree 3 graphs and to $\frac{7}{6}$ for cubic graphs. When combined with existing techniques this result also leads to approximation ratios, $(B+3)/5+\varepsilon$ for the independent set problem and $2-5/(B+3)+\varepsilon$ for the vertex cover problem on graphs of degree B, improving previous bounds for relatively small odd B.

1. Introduction

By virtue of recent remarkable developments in the theory of the polynomial time approximability it is now possible to classify many *NP*-hard optimization problems *qualitatively* by their approximation properties. The class *MAX SNP*, a subclass of *NP* optimization problems consisting solely of constant factor approximable problems, was introduced by Papadimitriou and Yannakakis, and shown to contain many natural complete

---

* A preliminary version was presented at the 4th Workshop on Algorithms and Data Structures (Canada, August 1995).

problems [16]. Arora et al. then established the inapproximability of *MAX SNP*-hard problems by proving that none of them can have a polynomial time approximation scheme (PTAS) unless $P = NP$ [1]. As a result a more *quantitative* classification of hard problems according to their approximability has become a research issue of even greater interest.

The main problem we treat in this paper is a well-studied one: the *bounded degree Independent Set problem*. An independent set in a graph is a set of nodes in which no two of them are adjacent, and in the independent set problem such a node set of maximum cardinality is sought. The general Independent Set problem (denoted MAX IS), being notorious for its apparent intractability, has been a source of many nontrivial lower bound results. When its approximability is concerned in particular (i.e., the approximation ratio which can be guaranteed in polynomial time), even the best known heuristic performs only slightly better than trivial (approx. ratio of $O(n/\log^2 n)$ [5]), and a recent result explains this phenomenon by providing a strong lower bound of n^ε for any $\varepsilon > 0$, under a reasonable complexity theoretic assumption [11]. On the other hand the bounded degree Independent Set problem (denoted MAX IS-B when the maximum node degree of graphs is bounded above by B) has an interesting history of its own. It is one of the original *MAX SNP*-complete problems given in [16], and its approximation ratio has been continuously improved over the years by a number of new techniques and analysis. The first nontrivial performance ratio of B appeared implicitly in Lovász's algorithmic proof [14] of Brooks's coloring theorem [6]. Hochbaum developed a heuristic with a ratio $B/2$ [12], which applies to the case of *weighted* graphs as well, using this coloring technique coupled with a method of Nemhauser and Trotter [15]. Halldórsson and Radhakrishnan recently showed that the greedy heuristic actually delivers a better ratio, $(B + 2)/3$ [9]. The best results known today are summarized as follows. Berman and Fürer designed new heuristics of which performance ratios are arbitrarily close to $(B + 3)/5$ for even B and $(B + 3.25)/5$ for odd B [4]. Halldórsson and Radhakrishnan then obtained soon afterward, via subgraph removal techniques, asymptotically better ratios, $B/6 + o(1)$ and $O(B/\log\log B)$ [8].

In this paper we pay special attention to MAX IS-3. MAX IS-B is *NP*-complete even when instance graphs are restricted to being cubic and planar [7]. It is also known that MAX IS (unbounded degree) admits a PTAS when graphs are planar [2]. In Section 2 we show, however, that MAX IS-3 and MAX IS restricted to cubic graphs (i.e., 3-regular graphs) are both *MAX SNP*-complete. As by-products a few other problems (such as MAX 3-SET PACKING-2 and MAX TRIANGLE PACKING-4) are shown to remain *MAX SNP*-complete at the lowest possible degree bounds.

We next study a better approximation of MAX IS-3 in Sections 3–5. In Section 3 our approximation algorithm is presented, and its performance ratios for degree 3 graphs and cubic graphs are derived in Section 4. The analysis proceeds centered around eight inequalities and equations, relating the sizes of various node subsets of a given graph, and their proofs are collectively given in Section 5. The previous best ratio for MAX IS-3 is $\frac{5}{4}$ [4], and we improve it to arbitrarily close to $\frac{6}{5}$ for degree 3 graphs and to $\frac{7}{6}$ for cubic graphs. The best performance guarantee for MAX IS-B is currently achieved by Berman–Fürer's algorithm when B is relatively small (up to around 613 [8]). The new ratio for MAX IS-3 provides a further improvement on their ratio for every odd degree B, matching their performance guarantee formula for even B. It is worth pointing out that with the method of Nemhauser and Trotter this result also leads to an approximation factor of $2 - 5/(B+3) + \varepsilon$ for the Minimum Vertex Cover problem with degree bound of B, improving previous bounds again for small B.

The heart of our algorithm is a type of local search which, taking advantage of degree boundedness, searches through better solutions in far distance. To accelerate its performance even further, however, some general reduction methods and other tricks are invented and incorporated. While it is our main concern to push the relative error ratio attainable in polynomial time as far down as possible, contributions of the current paper include demonstrating that the interplay of these techniques leads to a nontrivial improvement in the quality of approximate solutions, especially in the domain of low degree graphs (the related work of Halldórsson and Yoshihara [10] also indicates that some of our techniques are effective in improving the performance of simple greedy type heuristics for MAX IS-3).

2. *MAX SNP*-Completeness at the Lowest Degree Bounds

In this section we show that MAX IS remains *MAX SNP*-complete even for cubic graphs. MAX 3SAT-B is a restriction of MAX 3SAT such that in any instance the number of occurrences of any variable is bounded by B. We L*-reduce* MAX 3SAT-B to MAX IS-3 using the "ring of trees" construction of Kann [13]. The L-reduction [16] of an optimization problem Π_1 to another optimization problem Π_2 is a pair of polynomial time functions (f, g) such that

1. for every instance x of Π_1, $f(x)$ is an instance of Π_2 such that $\text{opt}(f(x)) \leq \alpha\, \text{opt}(x)$ for some positive constant α, and
2. for every feasible solution s of $f(x)$, $g(s)$ is a feasible solution of x such that $|\text{opt}(x) - c_1(g(s))| \leq \beta |\text{opt}(f(x)) - c_2(s)|$ for some positive constant β,

where $\text{opt}(x)$ is the optimum cost of an instance x, and c_i is the cost function of feasible solutions of Π_i for $i = 1, 2$.

Theorem 1. *MAX IS-*3 *is MAX SNP-complete.*

Proof (*Sketch*). Suppose a variable u occurs d times in a given 3SAT-B instance. Let K be a large enough power of two (it suffices to take $K = 2^{\lfloor \log_2((3/2)B+1) \rfloor}$). Construct K identical cycles of length $2d$ (called rings) and sequentially index the nodes of each cycle from 1 to $2d$. Also construct $2d$ complete binary trees with K leaves each. Join these rings and trees by overlapping, in the identical fashion, leaves of each tree with nodes of the same index from each ring. Label the roots of these trees as u_i and $\bar{u}_i$, $i = 1, \ldots, d$, alternatively in the order of their indices. Construct a ring of trees this way for every variable.

Each clause c is represented by a clique of size $|c|$. Suppose the ith occurrence of a variable u is in a clause c. Connect $\bar{u}_i$ or u_i, depending on whether u appears positive or not, to the corresponding node in the c-clique. This way clause cliques and rings of trees are connected together. Note that the degree of every node is bounded by 3.

Let A be the (disjoint) union of rings of trees corresponding to all variables. An independent set is said to be *consistent* if it includes all u_i's and none of $\bar{u}_i$'s, or vice versa, for every variable u. For a MAX IS-3 instance thus constructed it can be shown that

1. an independent set maximum in A is consistent, and
2. an independent set maximum in A is larger than an independent set not maximum in A.

We deduce that an optimal solution for MAX IS-3 consists of a maximum independent set in A plus a collection of nodes, one each from a clique corresponding to a satisfied clause in an optimal solution for MAX 3SAT-B. It follows that an optimal value is scaled up only by some constant factor because the size of A can be bounded by K and $\sum d \leq 3 \cdot$ (number of clauses) and at least half of the clauses can be always satisfied.

Secondly, from any MAX IS-3 solution we can find a consistent solution of no smaller size, and hence, a solution for MAX 3SAT-B. The value of a solution thus obtained is no further from the optimum than the original one for MAX IS-3. □

A graph is called *branchy* if it has no node of degree less than 2 and if any two nodes of degree 2 are not adjacent in it. The branchy reduction is one of the reduction methods used in our algorithm (see Section 3.1), and for its effect Lemma 4 asserts that it reduces an arbitrary graph to a branchy one without any loss of approximation quality.

Theorem 2. *MAX IS is MAX SNP-complete for cubic graphs.*

Proof. Apply first the branchy reduction to a degree 3 graph. We reduce MAX IS for branchy degree 3 graphs to MAX IS for cubic graphs by an L-reduction (f, g). Given a branchy degree 3 graph G with a set V_2 of degree 2 nodes and a set V_3 of degree 3 nodes, f does the following local replacement of every degree 2 node in V_2. Let u be a degree 2 node with degree 3 nodes v and w adjacent to it. Replace u by a graph H_u with the node set $\{u_1, u_1', u_2, u_2'\}$, where any two of these four nodes are adjacent except for the pair of u_2 and u_2'. Connect H_u to the rest of G by two edges, one between u_2 and v, and the other between u_2' and w. It is easily seen that every node of H_u now has its degree 3, and thus, $f(G)$ is indeed a cubic graph.

Let I_2 be an independent set in $f(G)$. Then there exists an independent set I_2' of no smaller size in $f(G)$ such that for each subgraph H_u, introduced by f corresponding to $u \in V_2$, either (i) $\{u_2, u_2'\} \subseteq I_2'$ or (ii) $\{u_2, u_2'\} \cap I_2' = \emptyset$ and $\{u_1, u_1'\} \cap I_2' \neq \emptyset$. From such an independent set I_2' in $f(G)$, g constructs an independent set I_1 in G such that (i) $I_1 \cap V_3 = I_2' \cap V_3$ and (ii) $u \in I_1 \cap V_2$ iff $\{u_2, u_2'\} \subseteq I_2'$ for every u corresponding to H_u. Thus we have $|g(S)| \geq |S| - |V_2|$ for any solution S of $f(G)$, and, in particular, $\text{opt}(G) \geq \text{opt}(f(G)) - |V_2|$. Conversely, $\text{opt}(f(G)) \geq \text{opt}(G) + |V_2|$ since, for any independent set I_1 in G, two nodes of H_u if $u \in I_1$, or one node of it otherwise, can be added to independent $I_1 \cap V_3$ in $f(G)$ for each u in V_2. So, $\text{opt}(f(G)) = \text{opt}(G) + |V_2|$, and

$$\text{opt}(G) - |g(S)| \leq \text{opt}(G) - |S| + |V_2| = \text{opt}(f(G)) - |S|$$

for any solution S of $f(G)$. Notice also that $|V_2| \le \mathrm{opt}(G)$ since V_2 is a feasible solution in G, and thus

$$\mathrm{opt}(f(G)) = \mathrm{opt}(G) + |V_2| \le 2\,\mathrm{opt}(G). \qquad \square$$

There are some other *MAX SNP*-complete problems structurally closely related to MAX IS-3 and MAX IS-B such as:

MAX 3-DIMENSIONAL MATCHING-*B*. Given three sets W, X, Y and a set $M \subseteq W \times X \times Y$ such that the number of occurrences of any element of W, X, or Y in M is bounded by B, find the largest matching, i.e., a subset $M' \subseteq M$ such that no two elements of M' agree in any coordinate.

MAX 3-SET PACKING-*B*. Given a collection C of subsets of a set S where every $c \in C$ contains at most three elements and every element $s \in S$ is contained in at most B of the subsets in C, find a largest collection of mutually disjoint subsets in C.

MAX TRIANGLE PACKING-*B*. Given a graph of maximum node degree bounded by B find a largest collection of mutually (node) disjoint 3-cliques.

Using *MAX SNP*-completeness of 3-DIMENSIONAL MATCHING-3, Kann showed that MAX IS-5, MAX 3-SET PACKING-3, and MAX TRIANGLE PACKING-6 are *MAX SNP*-complete as well [13]. Reducing MAX IS-3 to these problems instead, we can improve the degree bounds in these problems to the best possible ones.

Corollary 3. *MAX* 3-*SET PACKING*-2 *and MAX TRIANGLE PACKING*-4 *are MAX SNP-complete.*

Definitions and Notation. For a graph $G = (V, E)$ and a node set $U \subseteq V$, let $G(U)$ denote the subgraph of G induced by U. The *neighborhood set* $N(U)$ of U is the set of nodes (of V) adjacent to a node of U, and the *degree* $d(v)$ of a node $v \in V$ is $|N(\{v\})|$. Either of these can also be given with restriction to an arbitrary node set (instead of V), and, for $W \subseteq V$, $N_W(U)$ and $d_W(v)$ denote $N(U) \cap W$ and $|N_W(\{v\})|$, respectively. An acronym *MIS* is used for a maximum independent set. The *independence number*, $\alpha(G)$, of G is the cardinality of an MIS in G.

Our local search method is based on augmentation of an independent set U by a node set I called an *improvement*. Here, I is an improvement for U if $G(I)$ is connected and the symmetric difference $U \oplus I$ is a larger independent set. An *s-improvement* is the one that adds s and removes $s-1$ from U, and a solution is *s-optimal* if it has no s-improvement. The quality of an approximation algorithm is measured by its *performance ratio*, which is the worst case ratio of the optimal solution size to the size of an approximate solution returned by the algorithm from the same instance.

3. Approximation Algorithm

The mechanism of our algorithm $\mathsf{MIS3}_k$ for building a large independent set is a local search in two directions (as in Berman–Fürer's algorithm for MAX IS-B [4]). For some constant parameter k, a solution A is repeatedly augmented by applying improvements of size $O(k \log n)$ for $n = |V|$, until such an augmentation is no longer possible. When A becomes locally optimal its complement, $G(V - A)$, is searched as well. When G is a degree 3 graph and A is 1-optimal every node of $G(V - A)$ has degree of 2 or less, and, thus, an MIS A' in this space can be easily found. If A' is larger than A, the local search is started again from A'.

The resulting solution thus satisfies two properties: (i) A is $O(k \log n)$-optimal and (ii) $\alpha(G(V \backslash A)) \leq |A|$. Properties (i) and (ii) together assure that $(\frac{5}{4} + 1/k)|A| \geq \alpha(G)$, and, moreover, this inequality is tight. To improve the performance, we assure that the output satisfies several additional properties that enable us to prove a stronger inequality. To be more specific $\mathsf{MIS3}_k$ has the following additional features (see Figure 1 for a more formal description):

1. A given graph is preprocessed by approximation preserving reductions before a local search is initiated (by the first **Repeat**-loop). As a result the graph satisfies such properties that (1) there exists no node of degree < 2, (2) any two nodes of degree 2 are nonadjacent, (3) $\alpha(G) \leq \frac{1}{2}|V|$, and (4) every small (i.e., of size $\leq k$) node set U is *not* an MIS in $G(U \cup N(U))$.
2. An initial solution is constructed with preference given to degree 2 nodes over degree 3 ones. All the degree 2 nodes are collected unconditionally into an initial subset A (possible because of property (2) above). Attached to it is a solution recursively found from $G(V \backslash (A \cup N(A)))$, and this is the initial solution we start with.

$\mathsf{MIS3}_k$

Input: A degree 3 graph $G = (V, E)$.

Repeat
  oldsize $\leftarrow |V|$
  Do Branchy reduction
  Do Nemhauser–Trotter reduction
  Do Small Commitment reduction
until $|V| =$ oldsize   /* until no more reduction is applicable */
$A_1 \leftarrow \{v \in V : d(v) = 2\}$
$A_2 \leftarrow$ an independent set recursively computed in $G(V \backslash (A_1 \cup N(A_1)))$
$A \leftarrow A_1 \cup A_2$
Repeat
  oldsize $\leftarrow |A|$
  Do Acyclic Complement procedure
  Do all possible improvements of size $\max\{3k \log n, 4k + 2k \log n\}$ to A
  Find an optimal solution A_3 in $G(V \backslash A)$
  If A_3 is larger than A **then** $A \leftarrow A_3$
until $|A| =$ oldsize

Fig. 1. Algorithm $\mathsf{MIS3}_k$.

3. Every time an $O(k \log n)$-optimal solution is found, it is made sure that its complement is acyclic by replacing it, if necessary, with another of no smaller size (in the second **Repeat**-loop).

Each of these procedures is described in more detail in the following subsections.

3.1. *Branchy Reduction*

A graph G is called *branchy* if every node has degree 2 or more, and any two nodes of degree 2 are nonadjacent (the notion of branchy graphs and reduction was used before in the context of the weighted feedback vertex set problem, see [3]). A branchy graph G' can be obtained from an arbitrary graph G as follows:

1. While there is a node of degree 0 or 1, remove it (along with its neighbor, if any) from G and store it in a set S.
2. While there is a path $\langle v_1, v_2, v_3, v_4 \rangle$ where both v_2 and v_3 are degree 2 nodes, remove v_2 and v_3 from G, insert an edge $\{v_1, v_4\}$, and store the path in a set P.

The next lemma states that branchy reduction is a lossless reduction.

Lemma 4. *Let $G' = (V', E')$ be the branchy reduction of $G = (V, E)$, and let S and P be the sets produced in the process. Then*:

1. *From any independent set A' in G', an independent set of size $|A'| + |S| + |P|$ in G can be constructed in linear time.*
2. *There is an MIS A in G such that $A \cap V'$ is an independent set of size $|A| - |S| - |P|$ in G'.*

Proof. 1. Clearly, A' is an independent set in G. Let $\langle v_1, v_2, v_3, v_4 \rangle$ be a path found in Step 2. Since v_1 and v_4 are adjacent in G', at most one of them is contained in A'. Thus, one node for every path in P, and every node in S can be inductively added to A' to form an independent set in G.

2. It can be seen that there is an MIS A in G containing all the nodes of S, plus exactly one of v_2 and v_3 from every path $\langle v_1, v_2, v_3, v_4 \rangle$ found in Step 2. Removing all of such nodes from A results in $A \cap V'$, which is independent in G'. □

3.2. *Nemhauser-Trotter Reduction*

Nemhauser and Trotter studied solutions to a linear program relaxation of the MAX IS problem in [15], and showed that, for an arbitrary graph $G = (V, E)$, the partition $\{V_1, V_2, V_3\}$ of V can be computed, with time complexity of the bipartite matching problem, such that (i) there is an MIS containing all the nodes of V_1 but none of V_2, (ii) $N(V_1) \subseteq V_2$ (i.e., there is no edge between V_1 and V_3), and (iii) $\alpha(G(V_3)) \leq \frac{1}{2}|V_3|$. Hochbaum was the first to use the preprocessing method, based on this theorem, in approximation of the MAX IS problem [12]: compute V_1, V_2, and V_3 as above from a given graph, find an approximate solution S in $G(V_3)$, and return $S \cup V_1$ as an approximate solution for G. In this way an arbitrary graph G is reduced to one (i.e., $G(V_3)$) in which the independence number is at most half of the number of nodes.

3.3. *Small Commitment Reduction*

This reduction repetitively looks for a substructure of the following kind in G: for some node set A of size at most k, A is an MIS in $G(A \cup N(A))$. If found, we commit ourselves to A, remove $A \cup N(A)$ from G, and repeat as long as such a small independent set A (*small commitment*) remains in G. The effect of this reduction is again no loss in approximation quality.

3.4. *Acyclic Complement Procedure*

Suppose that I is a 2-optimal independent set in a connected degree 3 graph $G = (V, E)$ and G is not K_4. This procedure finds an independent set I' of size no smaller than I in polynomial time such that its complement induces an acyclic subgraph. We note that here G must be assumed to be a degree 3 graph (unlike the other reduction methods given earlier).

First a few easy observations. The subgraph $G(V \backslash I)$ consists of disjoint cycles and paths only, because its degree is at most 2 (otherwise, a 1-improvement). Also every node in $V \backslash I$ must be adjacent to some node in I (otherwise, a 1-improvement). Assume that there exists a cycle in $G(V \backslash I)$ and let C denote its node set. We describe below how to reduce the number of cycles in $G(V \backslash I)$ by one, depending on the nodes in $N(C) \backslash C \subseteq I$.

Case 1. There is a node $v \in N(C) \backslash C$ adjacent to three nodes, a, b, and c, of C. If $\{a, b\}, \{b, c\}, \{c, a\} \in E$ then G is K_4. Otherwise, if, say, $\{a, b\} \notin E$, then $\{a, b, v\}$ is a 2-improvement.

Case 2. There is a node $v \in N(C) \backslash C$ adjacent to two nodes, a and b, of C (note $\{a, b\}$ must be in E). Let $c \neq a$ be another node of C adjacent to b. Break this cycle by bringing b into I and v out of I. This will not create a new cycle in the complement because all the nodes which could form a new component are connected to c and c now has degree 1 in the complement.

Case 3. For each node $v \in N(C) \backslash C$ there exists exactly one node u in C adjacent to v. If $d(v) = 1$, then swap v and u. If $d(v) = 2$, then the other neighbor of v must be an endpoint of a maximal path in $G(V \backslash I)$ (otherwise, a 2-improvement); swapping v and u cannot create a new cycle. So assume $d(v) = 3$ and let the other two neighbors of v be w and x. Observe that either w or x must be an endpoint of a maximal path in $G(V \backslash I)$ (otherwise, a 2-improvement), and if w and x are of different paths, swapping u and v does not introduce a new cycle. So assume that they are the endpoints of a maximal path P in $G(V \backslash I)$. Destroy C by swapping v and u, and this creates a new cycle C' formed by P and v. Apply the same procedure to C', but never swap v and u again by considering nodes other than v in C'.

We continue as long as the current condition holds, and this way the procedure processes a sequence of cycles. Notice, however, that no cycle can repeat in this sequence since we always move onto a newly created cycle. So, the procedure must terminate, upon which the number of cycles in the complement is decreased by one.

4. Analysis of Performance Ratios

To estimate the quality of approximate solutions produced by $\mathsf{MIS3}_k$, we compare the relative sizes of node subsets defined by our solution and one fixed optimal solution. Given a graph $G = (V, E)$, let A' be an independent set found by $\mathsf{MIS3}_k$, and let B' be any MIS in G. Partition V into four sets A, B, C, and D such that

- $C \stackrel{\text{def}}{=} A' \cap B'$ (C for the "common" portion of approximate and optimal solutions),
- $A \stackrel{\text{def}}{=} A' \backslash C$ (A for "approximate") and $B \stackrel{\text{def}}{=} B' \backslash C$ (B for "best"),
- $D \stackrel{\text{def}}{=} V \backslash (A \cup B \cup C)$, i.e., the remaining portion of V.

Based on this partition of V, every node in D will be further classified according to its neighboring subsets. If a node is of degree 3 (resp. 2) and its neighbors belong to node sets X, Y, and Z (resp. X and Y), we say the node is of type $[XYZ]$ (resp. $[XY]$), where $X, Y, Z \in \{A, B, C, D\}$.

An easy observation tells us that there are possibly 10 node types for degree 2 nodes and 20 for degree 3 nodes, among which a total of 18 can be valid types for nodes in D. Moreover, D-nodes of type $[AB]$ do not occur in our graphs as will be explained later, which leaves us the following 17 types for D-nodes:

> **degree 2 nodes:** types of $[AC]$, $[BC]$, $[CC]$, and $[CD]$.
> **degree 3 nodes:** types of $[AAB]$, $[AAC]$, $[ABB]$, $[ABC]$, $[ABD]$, $[ACC]$, $[ACD]$, $[BBC]$ $[BCC]$, $[BCD]$, $[CCC]$, $[CCD]$, and $[CDD]$.

Denote the cardinalities of sets, A, B, C, and D, by respective lower case letters, and set $i = b - a$. Also denote the number of degree 2 nodes in sets A, B, and C by a_2, b_2, and c_2, respectively. The number of D-nodes of type $[XYZ]$ (resp. $[XY]$) is denoted by its lower case counterpart, $[xyz]$ (resp. $[xy]$). In Section 5 we prove the eight inequalities and equations listed in Figure 2, each concerning a cardinality relation to be satisfied by various node sets under our classification scheme. Assume for now that all these inequalities and equations hold.

Theorem 5.

1. $\mathsf{MIS3}_k$ *approximates MAX IS*-3 *with a ratio* $\frac{6}{5} + 1/5k$ *in time* $O(n^{2+3k \log 3})$.
2. $\mathsf{MIS3}_k$ *approximates MAX IS for cubic graphs with a ratio* $\frac{7}{6} + 1/6k$ *in time* $O(n^{2+3k \log 3})$.

Proof. The time complexity of $\mathsf{MIS3}_k$ is dominated by that of its local neighborhood search. A search space consists of all the connected subgraphs of node set size at most $\max\{3k \log n, 4k + 2k \log n\}$, and, hence, it takes $O(n^{1+3k \log 3})$ time to find a small improvement, as one can show that the number of such subgraphs of size at most s in a degree 3 graph of n nodes is smaller than $n3^s$. Since a solution is augmented at most n times, $\mathsf{MIS3}_k$ runs in time $O(n^{2+3k \log 3})$.

$$\frac{k+1}{k}c \geq 3[bc] + 2[bbc] + [bcd] + [cdd] + [cd] + [bcc] + [cc] - (2[acc] + [aac] + [abc] + [acd] + [ac]) \quad (1)$$
$$\frac{k+1}{k}c \geq [bbc] + [bc] + [bcc] + [cc] + \tfrac{1}{2}([bcd] + [cdd] + [cd] + [ccd]) \quad (2)$$
$$c \geq i + [ccc] + [acc] + [aac] + [ac] + [cc] + \tfrac{1}{2}([cdd] + [ccd] + [acd] + [cd]) \quad (3)$$
$$\frac{k+1}{k}a \geq 3i + a_2 + 2[aab] + 2[aac] + [abb] + [abd] + [acc] + [acd] + [abc] + [ac] \quad (4)$$
$$d \geq c + i \quad (5)$$
$$b_2 = 3i + a_2 + ([aab] + 2[aac] + [acc] + [acd] + [ac]) - ([abb] + 2[bbc] + [bcc] + [bcd] + [bc]) \quad (6)$$
$$3c - c_2 = 3[ccc] + 2([acc] + [bcc] + [ccd] + [cc]) + [aac] + [abc] + [bbc] + [acd] + [bcd] + [cdd] + [ac] + [bc] + [cd] \quad (7)$$
$$a_2 + [bc] + [cc] \geq \tfrac{1}{4}(b_2 + c_2 + [ac] + [cd]) \quad (8)$$

Fig. 2. List of inequalities relating node set sizes.

1. Sum up inequalities and equations (1)–(8) with respective multiplicative factors of $\frac{7}{13}$, $\frac{6}{13}$, $\frac{12}{13}$, 1, $\frac{3}{13}$, $\frac{4}{13}$, $-\frac{3}{13}$, and $\frac{16}{13}$, which yields the inequality

$$\frac{k+1}{k}(a+c) \geq 5\tfrac{1}{13}i + \tfrac{1}{13}(a_2 + c_2 + 27[aab] + 6[abb] + 10[abd] + 33[aac] + 6[acc] + 10[acd] + 13[cdd] + 6[bbc] + 20[ac] + [bc] + 14[cd]).$$

Since all the variables are of nonnegative value, we have $((k+1)/k)(a+c) \geq 5\frac{1}{13}i$, and hence,

$$\frac{|B'|}{|A'|} = \frac{b+c}{a+c} = \frac{a+c+i}{a+c} \leq 1 + \frac{1}{(k/(k+1))5\frac{1}{13}} < \frac{6}{5} + \frac{1}{5k}.$$

2. Setting the number of degree 2 nodes of any type to zero, (1) + (4) + (6) yields

$$\frac{k+1}{k}(a+c) \geq 6i + 3[aab] + 3[aac] + [abd] + [acd] + [cdd] \geq 6i,$$

and, hence,

$$\frac{|B'|}{|A'|} = \frac{a+c+i}{a+c} \leq 1 + \frac{1}{(k/(k+1))6} = \frac{7}{6} + \frac{1}{6k}. \qquad \square$$

When combined with $\mathsf{MIS3}_k$ to handle degree 3 graphs, Berman–Fürer's algorithm for MAX IS-B [4] does better for every odd B, giving the same performance guarantee formula as the one for even B.

Corollary 6. *MAX IS-B can be approximated in polynomial time within a ratio arbitrarily close to* $(B+3)/5$ *for all* $B \geq 2$.

5. Proofs of (1)–(8)

The purpose of the preprocessing and the acyclic complement procedures presented in Section 3 is to normalize a given graph and the final solution so that they possess some nice properties.

Lemma 7. *The node sets* A, B, C, *and* D *satisfy the following properties*:

1. C *is maximal* (*i.e.*, A' *and* B' *have a maximum overlap*).
2. *The subgraph* $G(B \cup D)$ *induced by the complement of a solution* A' *is acyclic*.
3. *The set of degree* 2 *nodes is an independent set in* G.
4. *Every node of* A *has at least two neighbors in* B.
5. *There exists no node of type* $[AB]$ *in* G.

Proof. 1. Clearly this can be assumed since the choice of an MIS $B \cup C$ is up to us.

2. This is a direct consequence of the acyclic complement procedure.

3. This is a direct consequence of the branchy reduction.

4. Suppose there is an A-node u with less than two neighbors in B. If no neighbor, then $B \cup C \cup \{u\}$ is a larger independent set and $B \cup C$ is not optimal. If u has only one neighbor w in B, then $(B \backslash \{w\}) \cup (C \cup \{u\})$ is an independent set as large as $B \cup C$, yet it has a larger intersection with $A \cup C$, contradicting property 1.

5. Let u be a node of type $[AB]$ and let w be its neighbor in B. Since $d_{B \cup C}(u) = 1$, $(B \backslash \{w\} \cup \{u\}) \cup C$ remains independent. Note that w is necessarily of degree 3 due to property 3, meaning that it cannot become a node of type $[AB]$. Thus, we can choose B' so that there is no degree 2 node of type $[AB]$ in D. □

5.1. *Proof of* (1)

We make use of local accounting here as well as in proving (2) and (4). Denote the set of D-nodes adjacent to C by D' $(= D \cap N(C))$. In a nutshell we assign potential to every node in $C \cup D'$ so that inequality (1) states that the total potential of $C \cup D'$ cannot be positive. We partition $C \cup D'$ into connected fragments to study each of them one at a time. We show that a fragment with a positive potential either contains a small improvement, or a k-commitment, or otherwise we will be able to "factor it out" and reduce the problem to a smaller one. There are 14 different node types for D'-nodes, and for the sake of the proof we name them as follows:

- *Enforcers* are
  —*solitaires*: nodes of type $[BC]$ or $[BBC]$,
  —*half-solitaires*: nodes of type $[BCD]$, $[CDD]$, or $[CD]$, and
  —*pairs*: nodes of type $[BCC]$ or $[CC]$.

- *Absorbers*: nodes of type $[ACC]$, $[AAC]$, $[ABC]$, $[ACD]$, or $[AC]$.
- *Neutrals*: nodes of type $[CCC]$ or $[CCD]$.

Each enforcer u is given a weight $w(u)$, which is initially 1, and larger weights result from reductions that factor out connected fragments (see below). Each node u of $C \cup D'$ is assigned with a potential $p(u)$ such that

$$p(u) = \left\{\begin{array}{l} c_\alpha - (w(u) - 1)/k \\ c_\alpha \quad (< 0) \\ -(k+1)/k \\ 0 \end{array}\right\} \text{ if } u \text{ is } \left\{\begin{array}{l} \text{an enforcer of type } [\alpha], \\ \text{an absorber of type } [\alpha], \\ \text{a } C\text{-node}, \\ \text{a neutral node}, \end{array}\right.$$

where c_α is the coefficient of a $[\alpha]$ term in inequality (1) for $\alpha \in \{a, b, c\}^3 \cup \{a, b, c\}^2$. For a node set U let $p(U)$ (the potential of U) and $w(U)$ (the weight of U) stand for $\sum_{u \in U} p(u)$ and $\sum_{u \in U} w(u)$, respectively. Now inequality (1) can be rewritten in the form $p(C \cup D') \le 0$.

Consider the subgraph G_C of G formed by C, the set of enforcers, and the set of edges between them (ignoring those edges inside D). Then the set of C-nodes and enforcers is partitioned by the connected components of G_C, in which each pair node plays the role of an "edge" connecting its neighbors in C. In what follows a connected component is meant to be that of G_C defined this way. For the node set X of such a component, let X_C, X_D, and $\mathrm{ab}(X)$ denote $X \cap C$, $X \cap D$ (i.e., the set of enforcers in X), and the number of edges connecting a node of X_C with an absorber, respectively. Then $p(C \cup D')$ is the sum of $p_X \stackrel{\text{def}}{=} p(X) - \mathrm{ab}(X)$ over the components X of G_C. If no p_X is positive we are done. Otherwise, we apply a case analysis to X with positive p_X.

Let $w_X \stackrel{\text{def}}{=} w(X_D)$ and $x \stackrel{\text{def}}{=} |X_C|$. We measure the size of X by w_X. First the case of a small component, that is, assume $w_X \le k$. Certainly X_D contains at least $x - 1$ pairs. If X_D contains $x + 1$ (or more) pairs, these pairs, together with X_C, form a k-improvement, a contradiction. We get a similar contradiction if X_D contains x pairs and a (half-)solitaire, or $x - 1$ pairs and either of: three half-solitaires, one solitaire and one half-solitaire, two solitaires, or two nonadjacent half-solitaires. If X_D consists of pairs only and $|X_D| \le x$, then $p_X < 0$. So, we need to consider only the cases when X_D consists of $x - 1$ pairs, plus either one solitaire or two half-solitaires adjacent to each other.

Case 1: X_D *consists* of $x - 1$ *pairs and a solitaire of type* $[BC]$ (*so,* $3 - (w - 1)/k$ *in its coefficient*). We remove X and all the adjacent edges from consideration, and at the same time shift the potential p_X to nodes of type $[CCC]$ or $[CCD]$ that must be adjacent to X_C.

As a technical preliminary, observe that at least two edges go from X_C to nonenforcers. If $X_C = \{u\}$ for some u, then u has three neighbors as a neighbor of a degree 2 node (Property 3), hence two nonenforcer neighbors.Otherwise, the node set X_C and the edge set of pairs form a tree with at least two leaves and each leaf has a nonenforcer neighbor; in particular, a leaf adjacent to $[BC]$ has three neighbors—one pair, one $[BC]$, and one nonenforcer.

Notice that currently $p_X = 2 - w_X/k - \mathrm{ab}(X)$, and so $\mathrm{ab}(X) \le 1$ since $p_X > 0$.

Case 1.1: $\mathrm{ab}(X) = 1$. An edge goes from X_C to a $[CCC]$ or $[CCD]$ node, say u. If u is of type $[CCC]$, then after X is removed, u appears to be of type $[CC]$, a pair (this way u is introduced as a "new pair" in the subsequent analysis). Suppose that after removal of X, some set $I \subseteq C \cup D$ becomes an improvement. If $u \notin I$, then I itself, and otherwise $I \cup X$, was an improvement even before the removal. Thus an s-improvement containing u after the removal translates into an $(s + w_X)$-improvement that is "genuine." Therefore, we can keep track of the improvement size correctly even after the removal if u is given the weight of $1 + w_X$. We also redefine $p(u)$ from 0 to $p_X = 1 - w_X/k$. This way the total potential is preserved, and the potential of u agrees with the formula for pairs.

We can handle identically the case when u is of type $[CCD]$. This time u becomes a half-solitaire after the removal, and the formula for its potential is same as the one for pairs. In either case we say that u is *promoted* (from a neutral to a pair or a half-solitaire).

Case 1.2: $\mathrm{ab}(X) = 0$. Then X_C is adjacent to two neutral nodes and both can be promoted. We do so by assigning potential to them *nondeterministically*; $1 = 1 + 1/k - 1/k$ to one and $1 + 1/k - (w_X + 1)/k$ to the other. Correspondingly, these newly introduced pairs and/or half-solitaires are given weights of 1 and $1 + w_X$, respectively. Here it is meant that if only one of them is later involved in any improvement, then that one will have its weight $= 1 + w_X$ in this promotion process while, if both are involved, it does not matter which one gets which weight.

In addition we note here that in the current case p_X could be positive even when $k \leq w_X$ if $w_X \leq 2k$. We simply handle this as above promoting two neutral nodes although this time one of them is given a negative potential.

Case 2: *X_D consists of $x - 1$ pairs and a solitaire of type $[BBC]$.* This case is similar to Case 1, only simpler. Now, $p_X = 1 - w_X/k - \mathrm{ab}(X)$ and $\mathrm{ab}(X) = 0$ since $p_X > 0$. So, at least one edge goes from X to a nonenforcer which must be a neutral node, and we can perform a promotion as before.

Case 3: *X_D consists of $x - 1$ pairs and two half-solitaires adjacent to each other*. This case is very similar to Case 2 (now $p_X = 1 + 1/k - w_X/k - \mathrm{ab}(X)$), but with one additional subcase—when no edge joins X_C with a nonenforcer, i.e., when $X_D = N(X_C)$. Let S be an MIS in X. If $|S| > x$, then $S \oplus X_C$ is a k-improvement, a contradiction. If $|S| = x$, then X_C is an MIS in $X_C \cup N(X_C)$. Since $|X_C| = x \leq k$, such a construct must have been eliminated by the Small Commitment reduction, again a contradiction. It should be noted here that this situation could not emerge as a result of performing some removals; removing a component may change the classification of its neighbors but not the neighbors themselves.

To prove (1) it remains to handle large components with a positive potential (i.e., the case $w_X > k$ and $p_X > 0$). For an easier analysis we "remove" half-solitaires first. Let H be a set of half-solitaires in X, and let I_H be an MIS in $G(H)$. Each node u of I_H is now given a new potential of $2 - (w(u) - 1)/k$ and treated as a new solitaire. Then, since $p(u) + p(v) = (1 - (w(u) - 1)/k) + (1 - (w(v) - 1)/k) \leq 2 - (w(u) - 1)/k$ for any two adjacent half-solitaires u and v in H, the total potential of new solitaires is no smaller than that of old half-solitaires, i.e., $p(I_H) \geq p(H)$. We may thus take only this set I_H of solitaires into account and ignore H; X_D is now an independent set.

Represent X by a graph G_X consisting of the node set X_C and the edge set E of pairs in X_D. We show below how to find a small improvement in it. (Note that we here pay attention only to enforcers, among those nodes adjacent to X_C, unlike the case of a small X.) A node in X_C is called a *solitaire point* if it is adjacent to a solitaire.

1. Remove all the leaf nodes (i.e., degree 1 nodes) recursively from G_X as long as they are not solitaire points. This pruning operation does not decrease the potential of X. Meanwhile, if the total weight of enforcers drops below k in the process, a k-improvement is found. To see it let S_1 and S_2 denote the sets of solitaires of type $[BBC]$ and those of type $[BC]$ in X_D, respectively. An easy calculation shows that

$$p(X) = \left(\frac{k+1}{k}\right)|E| + \left(\frac{2k+1}{k}\right)|S_1| + \left(\frac{3k+1}{k}\right)|S_2| - \frac{w(X_D)}{k} - \left(\frac{k+1}{k}\right)|X_C|,$$

and, by the assumptions that $p_X > 0$ and $w_X > k$,

$$\left(\frac{k+1}{k}\right)|E| + \left(\frac{2k+1}{k}\right)|S_1| + \left(\frac{3k+1}{k}\right)|S_2| > \frac{w(X_D)}{k} + \left(\frac{k+1}{k}\right)|X_C| > 1 + \left(\frac{k+1}{k}\right)|X_C|.$$

From this (and the fact that $|E| \geq |X_C| - 1$), one can show that $|X_D| > |X_C|$, or otherwise, X_D consists of $|X_C| - 1$ pairs and one solitaire of type $[BC]$. In the latter case, however, $p(X) > 0$ only if $w_X \leq 2k$, and this case was already eliminated (in Case 1.2). Thus, X_D is initially an independent set larger than X_C in G. Each pruning of a leaf node removes one node each from X_C and X_D. So, at any moment during this pruning process X_D provides an improvement of size $w(X_D)$.

Now every leaf node in G_X must be a solitaire point. A chain of degree 2 nodes in G_X is called *bare* if none of its intermediate nodes is a solitaire point.

2. If a leaf node v in G_X is connected to a maximal (i.e., nonextensible) bare chain Z of weight $\geq 2k$, then remove v, the solitaire adjacent to v, and Z. Its potential p' is nonpositive for, if Z is of length l and weight w, $p' = 2 + l(1+1/k) - w/k - l(1+1/k) = 2 - w/k \leq 0$ when $w \geq 2k$. Meanwhile, if the total weight of enforcers drops below $k+1$, then consider the node u in the remaining component, contacting Z before the removal.

Case 1: $d(u) \geq 2$ *before the removal of* Z. Then either u is a solitaire point or $d(u) = 3$ before the removal of Z. By the same argument given below in the analysis of 3, X now has an improvement of weight $\leq k$ after the removal of Z.

Case 2: $d(u) = 1$ *before the removal of* Z. Thus, X is simply a bare chain Z connecting two solitaire points (before the removal of Z). Since $p(X) = 2 + 2 + l(1 + 1/k) - w/k - (l+1)(1+1/k) = (3k-1-w)/k > 0$ it follows that $w(X_D) < 3k-1$. Hence, X is an improvement of weight $< 3k - 1$ before the removal of Z.

3. Consider a maximal bare chain Z in G_X not incident to a leaf node. It must be connecting a degree 3 node (or a degree 2 solitaire point) and another degree 3 node (or another degree 2 solitaire point). If the weight of Z is $\geq k+1$, then remove it. Its potential

$p(Z)$ is nonpositive for $p(Z) = l(1+1/k) - w/k - (l-1)(1+1/k) = (k+1-w)/k \leq 0$ when $w \geq k+1$. Note also that such a removal might create preconditions for operation 2.

Observe that while operations 1 or 2 never disconnects G_X, operation 3 could. Suppose it actually breaks G_X into two. If both are of weight $\geq k+1$, we continue with the one having a larger potential, which must be positive. Suppose such a removal introduces a component Y of weight $\leq k$ but with a positive potential. Let u be a node of Y contacting Z. It can be verified that if u is a degree 3 node before the removal of Z, Y must contain either (i) two solitaires, (ii) one solitaire and one cycle, or (iii) two cycles. Similarly, if u is a solitaire point, Y must contain either (i) or (ii). However, then either case leads to an improvement of weight $\leq k$. So, assuming that there exists no improvement of weight $\leq k$, any disconnected component of weight $\leq k$ must have a nonpositive potential, and it can be safely discarded.

Replace every maximal bare chain by an edge of the corresponding weight in G_X. Then each internal edge (i.e., incident with nonleaves only) has weight $\leq k$ while each external edge (i.e., incident with a leaf) has weight $\leq 2k-1$. Moreover, if X_C contains no solitaire point, the minimum degree of G_X must be 3.

Lemma 8. *In any graph with n nodes and minimum degree 3, there is a subgraph F with at most $3\log n$ nodes such that F contains more edges than nodes.*

Proof. Starting at any node r as a root, build a breadth first search (BFS) tree T with the following modification. We still mark nodes whenever they are visited, but even if a node is already marked, visit it again. We continue building T until some node, say u, is visited three times. The (cumulative) number of nodes appearing in T up to the kth level is at least $3 \cdot 2^k - 2$ while T may contain at most $2n+1$ nodes in it since any node can occur at most twice except for one with three appearances. So, if the depth of T is d, $2n+1 \geq 3 \cdot 2^d - 2$, that is, $2^d \leq \frac{2}{3}n + 1 \leq n$ for $n \geq 3$, and, hence, $d \leq \log n$.

Case 1. There is a node v occurring at least twice on a path running from r to some leaf. This means that v is contained in a simple cycle C_1 of length at most $\log n$. Collapse C_1 into a single node $[C_1]$, and every node in the modified graph will still have degree at least 3. Now construct an ordinary BFS tree rooted at $[C_1]$. It is easy to find another cycle $[C_1] \cup C_2$ of length at most $2\log n$ in the modified graph. Thus, $C_1 \cup C_2$ contains at most $3\log n$ nodes and more edges than nodes.

Case 2. No node occurs more than once on any path running from r down to a leaf. Take the smallest subtree T' of T containing all the three occurrences of u. It can be seen, by the assumption of the current case, that in T' there exist exactly three leaves, only at each of which u occurs, and exactly one node w of degree 3, at the position of the lowest common ancestor of two leaves. Assume that no parallel edges exist (the other case is similar). Then all the neighbors of w must be distinct, and this implies that there are three distinct (but not necessarily disjoint) paths running from w to u. However, then the union of these paths has at most $3\log n$ nodes in it, and contains more edges than nodes. □

Therefore, if X_C contains no solitaire point there is an improvement of weight $3k\log n$. Otherwise, pick any solitaire point in X_C as a root and start a BFS until either another solitaire point is found or a cycle is formed. This will find an improvement of weight $4k + 2k \log n$.

5.2. *Proof of* (2)

The proof of (2) is quite similar to that of (1), but simpler as we do not use promotions. Take the set of all the D-nodes with at most two neighbors in C, and select a maximum independent subset F in it. Then F contains all the nodes of types $[BBC]$, $[BC]$, $[BCC]$, and $[CC]$ (since none of them is adjacent to another D-node), and at least half of the nodes of types $[BCD]$, $[CDD]$, $[CCD]$, and $[CD]$ (since $G(D)$ contains no cycles, due to Property 2).

Give a potential $p(u) = 1$ to each node $u \in F$, and $p(u) = -(1 + 1/k)$ to each $u \in C$. It then suffices to show that $p(C \cup F) \leq 0$. As before, we divide $C \cup F$ into connected components defined by pairs. This time pairs are such nodes of F, each having two neighbors in C. Now it amounts to showing that $p(X) \leq 0$, for every component X of $C \cup F$. Assume otherwise (i.e., $p(X) > 0$). However, then, using the same reasoning as in the previous subsection, we can show

1. if $|X \cap F| \leq k$, then, for some $Y \subseteq X \cap F$ with $|X \cap C| + 1$ nodes, $(X \cap C) \cup Y$ is a $(|X \cap C| + 1)$-improvement, and
2. if $|X \cap F| > k$, then there exists an improvement of size at most $\max\{3k \log n, 4k + 2k \log n\}$,

reaching a contradiction in either case.

5.3. *Proof of* (4)

This inequality is proven in two stages. This time solitaires are such nodes of B with only one neighbor in A, and pairs those of B with exactly two. Denote by $a_{(j)}$, for $0 \leq j \leq 3$, the number of A-nodes with exactly j neighbors in B. Similarly, let $b_{(j)}$ denote the number of B-nodes with exactly j neighbors in A (so $b_{(1)}$ and $b_{(2)}$ are the numbers of solitaires and pairs, respectively). First we prove an analogue of inequality (1) in the context of A versus B.

Lemma 9. *If there is no improvement of size* $\max\{3k \log n, 4k + 2k \log n\}$,

$$2b_{(1)} + b_{(2)} \leq \frac{k+1}{k} a. \tag{9}$$

Proof. The proof is similar to that of (1), but simpler because we do not have half-solitaires, solitaires with potential 3, nor absorbers, and we have only one type of neutral nodes: $[AAA]$. The mechanism of promotions works as before, with one exception. Given a component X of $A \cup B$ with $|X \cap A| - 1$ pairs and one solitaire in it, we must argue that $X \cap A$ is adjacent to at least one neutral node. When we dealt with components of $C \cup D$, this was guaranteed by the Small Commitment reduction, but here this argument is of no use as $X \cap A$ may have neighbors in D, which cannot be promoted.

Nevertheless, we can reach a contradiction. Assume that $X \cap A$ is not adjacent to any neutral node. Then we have $N_B(X \cap A) = X \cap B$, and, consequently, $C \cup (X \cap A) \cup (B \setminus X)$ is an independent set as large as $C \cup B$, but with a larger overlap with our independent set $C \cup A$, contradicting the choice of $C \cup B$ (Property 1). $\square$

Consider the set $E(A, D)$ of all the edges between A and D. Any edge in it must be incident to a node u in A such that $d(u) = 3$ and $d_B(u) = 2$ (by Property 4). Conversely, any A-node u with $d(u) = 3$ and $d_B(u) = 2$ must be incident to an edge of $E(A, D)$. Thus we have $|E(A, D)| = |\{u \in A : d(u) = 3 \text{ and } d_B(u) = 2\}| = a_{(2)} - a_2$. On the other hand, $|E(A, D)| = 2|\{u \in D : d_A(u) = 2\}| + |\{u \in D : d_A(u) = 1\}| = 2([aab] + [aac]) + ([abb] + [abd] + [acc] + [acd] + [abc] + [ac])$ (there exists no D-node u with $d_A(u) \geq 3$), and, hence,

$$a_{(2)} = a_2 + 2([aab]+[aac]) + ([abb]+[abd]+[acc]+[acd]+[abc]+[ac]). \quad (10)$$

Recall that $a_{(0)} = 0$ (since $B \cup C$ is an MIS) and $a_{(1)} = 0$ (by Property 4), and consider the set of all the edges between A and B. Since $\sum_{u \in A} d_B(u) = 2a_{(2)} + 3a_{(3)} = 3a - a_{(2)}$ and $\sum_{u \in B} d_A(u) = b_{(1)} + 2b_{(2)} + 3b_{(3)} = 3b - b_{(2)} - 2b_{(1)} = 3(a + i) - b_{(2)} - 2b_{(1)}$, by equating them,

$$3i + a_{(2)} = 2b_{(1)} + b_{(2)}. \quad (11)$$

Combining (9), (10), and (11) yields

$$\begin{aligned} &3i + a_2 + 2[aab] + 2[aac] + [abb] + [abd] + [acc] + [acd] + [abc] + [ac] \\ &\quad = 2b_{(1)} + b_{(2)} \leq \frac{k+1}{k} a. \end{aligned}$$

5.4. *Proofs of the Remaining Inequalities*

Inequality (3) follows from the fact that our solution, $A \cup C$, is at least as large as the largest independent subset of $B \cup D$. Because $G(B \cup D)$ contains no cycles the size of the latter can be estimated by that of the set formed by B, plus all the D-nodes without neighbors in $B \cup D$ (types $[CCC]$, $[ACC]$, $[AAC]$, $[AC]$ and$[CC]$) plus half of the D-nodes without neighbors in B but with some in D (types $[CCD]$, $[CDD]$, $[ACD]$, and$[CD]$).

Inequality (5) follows from the fact that we use the Nemhauser–Trotter reduction; consequently, the size of an MIS, $a + i + c$, does not exceed one-half of the total number of nodes, $2a + i + c + d$.

Because $\sum_{u \in A} d_B(u) = \sum_{u \in B} d_A(u)$ and $b = a + i$ we have (6).

Similarly, (7) is the equation obtained from the relation $\sum_{u \in C} d(u) = \sum_{u \in C} d_D(u) = \sum_{u \in D} d_C(u)$.

Lastly, a degree 2 node u is called *good* if $u \cup N(u)$ contains only one node of $B \cup C$ while, if two nodes are contained, call it *bad*. Let g and h denote the respective numbers of good and bad degree 2 nodes in G. $\mathsf{MIS3}_k$ constructs an initial solution A from A_1, the set of all degree 2 nodes, and A_2, the solution recursively found from $G(V \setminus (A_1 \cup N(A_1)))$. Since $G(V \setminus (A_1 \cup N(A_1)))$ contains an independent set of size at least $(b + c) - (g + 2h)$, if it is inductively assumed that $\mathsf{MIS3}_k$ finds an independent set of size at least $\frac{5}{6}$ of the optimal one, $|A_2|$ is at least $\frac{5}{6}(b + c - g - 2h)$, and, hence, $|A|$ is at least as large as $(g+h)+\frac{5}{6}(b+c-g-2h)=\frac{5}{6}(b+c)+\frac{1}{6}(g-4h)$, attaining the desired ratio $\frac{5}{6}$ *if* $g \geq 4h$. Assuming otherwise (and Property 5), we have inequality (8).

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc*. 33*rd Annual Symp*. *on Foundations of Computer Science*, pages 14–23, 1992.

[2] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery*, 41:153–180, 1994.

[3] R. Bar-Yehuda, D. Geiger, J. Naor and R. M. Roth. Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and Bayesian inference. In *Proc*. 5*th Annual ACM–SIAM Symp*. *on Discrete Algorithms*, pages 344–354, 1994.

[4] P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *Proc*. 5*th Annual ACM–SIAM Symp*. *on Discrete Algorithms*, pages 365–371, 1994.

[5] R.B. Boppana and M.M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, 1992.

[6] R.L. Brooks. On coloring the nodes of a network. *Proceedings of the Cambridge Philosophical Society*, 37:194–197, 1941.

[7] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[8] M.M. Halldórsson and J. Radhakrishnan. Improved approximations of independent sets in bounded-degree graphs. *Nordic Journal of Computing*, 1(4):475–492, 1994.

[9] M.M. Halldórsson and J. Radhakrishnan. Greed is good: approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18:143–163, 1997.

[10] M.M. Halldórsson and K. Yoshihara. Greedy approximations of independent sets in low degree graphs. In *Proc*. 6*th Internat*. *Symp*. *on Algorithms and Computation* (*ISAAC*), LNCS #1004, Springer-Verlag, Berlin, pages 152–161, 1995.

[11] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proc*. 37*th Annual Symp*. *on Foundations of Computer Science*, pages 627–636, 1996.

[12] D.S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6:243–254, 1983.

[13] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37:27–35, 1991.

[14] L. Lovász. Three short proofs in graph theory. *Journal of Combinatorial Theory, Series B*, 19:269–271, 1975.

[15] G.L. Nemhauser and L.E. Trotter, Jr., Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[16] C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

Received March 1996, *and in final form May* 1998.