Check for updates

Chapter 41
Approximating Maximum Independent Set in Bounded Degree Graphs

Piotr Berman *   Martin Fürer †

Abstract

For every $\Delta > 2$ and $\varepsilon > 0$ we present a polynomial time approximation algorithm for the Maximum Independent Set problem, that in a graph of degree Δ approximates an optimal solution within ratio $\frac{5}{\Delta+3} - \varepsilon$ for even Δ and within ratio $\frac{5}{\Delta+3.25} - \varepsilon$ for odd Δ.

1 Introduction

It is well known that different NP-complete combinatorial optimization problems behave very differently with respect to their approximability [GJ]. In this paper we are interested exclusively in approximation ratios attainable by polynomial time algorithms. Therefore, by *approximating* we shall mean *approximating by a polynomial time algorithm.*

The most successful attempt to classify problems from this perspective has been the introduction of the class SNP by Papadimitriou and Yannakakis [PY]. Each SNP problem can be approximated with some positive constant performance ratio. In this paper, we use the inverse performance ratio (i.p.r.), which is a natural measure of the approximation quality for maximization problems. The i.p.r. of an approximation algorithm is the worst case (i.e., the minimum) ratio between the sizes of the approximate and optimal solutions. The bounded versions of many NP-complete problems are complete for the SNP class with respect to approximation preserving reducibilities. Among examples, one may mention MAX kSAT (MAX SAT with clause size bounded by k), minimum dominating set in graphs of degree Δ, MIS-Δ (maximum independent set in graphs of degree Δ).

The recent series of exciting lower bound results for approximation problems culminated in the proof that for each SNP-hard maximization problem there exists $\rho < 1$ such that every approximation algorithm has an i.p.r. at most ρ, unless P=NP (see [ALMSS], also Johnson's NP-completeness column [J]).

Currently, the upper and lower bounds on the attainable approximation ratios for SNP-complete problems are rather far apart. Closing this gap, even for some of these problems, would be a significant achievement. If this goal is ever to be reached, impossibility results would have to be improved substantially. The current ones, while technically impressive, do not even come close to intuitively "natural" values. For example, in a recent paper Bellare *et al.* [BGLR] show that MAX-3SAT cannot be approximated with an i.p.r. of $1 - \frac{1}{113}$ unless P=NP. The matching positive result, due to Yannakakis [Y], shows how to approximate this problem with an i.p.r. of $1-\frac{1}{4}$.

One may observe that a typical positive result is based on relatively simple combinatorial concepts and the progress, if achieved at all, comes in substantial increments. On the other hand, the negative results apply simultaneously quite an array of techniques, and each of them may be subject of further refinements, thus progress may be expected to be more tedious.

Approximating the MIS-Δ problem has already an interesting history. There is an obvious trivial approximation algorithm, which just

*Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, E-mail: berman@cse.psu.edu. Supported in part by NSF grant CCR-9114545.

†Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, E-mail: furer@cse.psu.edu.. Supported in part by NSF grant CCR-9218309.

chooses a maximal independent set. It always finds an independent set of size at least $\frac{n}{\Delta+1}$. In the case of a $\Delta+1$-clique, this bound is tight. Repeating this algorithm produces a $\Delta+1$ coloring.

If a connected graph contains a node of degree less than Δ, one can repeatedly choose a vertex of minimal degree and delete all its neighbors. In each step one node is chosen and at most $\Delta-1$ are deleted. Because the neighbors of the deleted nodes will have their degrees reduced, this Greedy Algorithm never chooses a node with Δ neighbors. It obtains an independent set of size $\frac{n}{\Delta}$.

When a component is Δ-regular and $\Delta>2$, we must be more careful. For example, in a 3-regular graph of 10 nodes we may pick one node, eliminate three others and the remaining nodes will form two triangles. As a result, we pick only three nodes, while we want to assure $\lceil 10/3\rceil=4$. Except when every connected component is a $\Delta+1$-clique one can still get an independent set of size $\frac{n}{\Delta}$ and a Δ coloring as follows. Brooks [Br] has shown that such a graph is Δ-colorable and the simple proof of Lovász [L] translates into a linear time algorithm based on depth-first search to find 2-connected components. A largest color class can be selected as the desired independent set. Alternatively we could apply the theorem that for $\Delta\geq 3$ the $\Delta+1$-clique is the only connected Δ-regular graph not containing an even length cycle with at most one chord [B]. One half of the vertices of such a cycle form a good independent set to start the Greedy Algorithm. Again, such a cycle can be found with depth-first search in linear time.

To obtain a reasonable i.p.r., one has to use good upper bounds on the size of a maximum independent set. A quick way to obtain such a bound is to look at a maximal matching. If it has k edges, then the MIS size is at most $n-k$. If $k<n(\Delta-1)/2\Delta$, then the complement of the matching has $n-2k>n/\Delta$ elements, and it yields the i.p.r. $\frac{n-2k}{n-k}>\frac{2}{\Delta+1}$; otherwise the independent set of size n/Δ yields the i.p.r. of $\frac{n/\Delta}{n-k}\geq\frac{n/\Delta}{n(\Delta+1)/2\Delta}=\frac{2}{\Delta+1}$.

A better i.p.r. can be obtained by Hochbaum [H], using the old method of Nemhauser and Trotter [NT]. They consider the fractional independent set problem obtained by relaxing the integer programming problem MIS to a linear programming problem. Besides 0 (not in the MIS) and 1 (in the MIS), basic feasible solutions can also contain the value 1/2. Furthermore, in polynomial time they can find an optimal linear programming solution with a maximal set of vertices assuming the value 1. And this set of vertices is guaranteed to be part of a MIS. First choosing this set of vertices optimally, and then choosing a $\frac{1}{\Delta}$-fraction of the vertices valued 1/2, results in an i.p.r. of $\frac{\Delta}{2}$.

Halldórsson and Radhakrishnan [HR1] have analyzed the Greedy Algorithm further and could actually determine its i.p.r. to be $\frac{3}{\Delta+2}$. An algorithm based on clique removals has been proposed by Paschos [P] and further investigated by Boppana and Halldórsson [BH]. The best i.p.r. obtained so far with this technique is $\frac{3.74}{\Delta+O(1)}$ [HR2]. Several researchers have also studied the closely related vertex cover problem for bounded degree graphs [MS,H,HR1].

In this paper, we use a relatively simple iterative local optimization procedure, combined with what one may call a complementation method, to increases the i.p.r. drastically. In particular, we show that one can approach the ratios of $\frac{5}{\Delta+3}$ for even Δ's and $\frac{5}{\Delta+3.25}$ for odd Δ's arbitrarily closely.

THEOREM 1.1. *For every $\varepsilon>0$ and $\Delta\geq 3$, there is a polynomial time algorithm approximating the MIS-Δ problem with an i.p.r. of $\frac{5}{\Delta+3}-\varepsilon$ for even Δ and with an i.p.r of $\frac{5}{\Delta+3.25}-\varepsilon$ for odd Δ.*

Our method reduces the approximation of MIS-Δ to the approximation of MIS-$(\Delta-2)$. In the process, the reciprocal of the i.p.r. increases by 2/5 plus an arbitrarily small constant. For even Δ's this ultimately reduces the problem to approximating MIS-2 within ratio $\frac{5}{2+3}$, which of course is possible. For odd Δ's we are not so lucky: MIS-1 cannot be approximated within ratio $\frac{5}{1+3}$, as this would mean finding approximated solutions larger than the optimal ones. Thus the bottleneck of our method is the problem of approximating MIS-3. A simpler version of our reduction technique reduces the approximation of MIS-Δ to MIS-$(\Delta-1)$ in such a way that the reciprocal of the i.p.r. increases by 1/4 (again, plus an arbitrarily small constant). Given that we have ratio 1 for MIS-2, this allows to approach the ratio of $\frac{4}{5} = \frac{5}{3+3.25}$ for MIS-3.

Nevertheless, we conjecture that an i.p.r. 5/6 for MIS-3 is attainable. In particular, one can reduce the problem MIS-3 to instances where there are no nodes of degree 0 or 1 and the nodes of degree 2 form an independent set. It appears that the i.p.r. of our algorithm is about 6/7 for 3-regular graphs. On the other hand, in examples where our heuristic has an i.p.r. very close to 4/5, we observed that almost all nodes of degree 2 belong to the optimal solution. One can modify our heuristic as follows: if the nodes of degree 2 are few, run the heuristic without change; otherwise make also a run that follows the selection of all nodes of degree 2 and the elimination of all their neighbors. We believe that a proper analysis of this modified algorithm will prove our conjecture.

2 Description of the algorithm

Before we describe the algorithm, we state the necessary definitions.

DEFINITION 2.1. *Below, $G = \langle V, E\rangle$ is an undirected graph or multigraph, and A, B, I are sets of nodes.*

(i) The subgraph induced by A *is* $G(A) = \langle A, E(A)\rangle$, *where $E(A)$ is the set of edges from E between vertices of A.*

(ii) A is an independent set *iff $E(A) = \emptyset$. MIS of G, a maximum independent set, is an independent set in V of maximal size.*

(iii) I is an improvement of A *iff both A and $A \oplus I$ are independent sets, $G(I)$ is connected and $A \oplus I$ is larger than A. (The operator $\oplus$ denotes the symmetric difference.)*

(iv) Comp$(A) = G(B)$, *where B equals $V - A$ if the degree of G is 3, and otherwise B is the set of nodes that have at least two neighbors in A.*

Our approach to approximating the Maximum Independent Set is fairly straightforward: we start from any independent candidate set A, and then we try to increase its size, using two possible ways. First, we check whether a "small" improvement I exists, if yes, we replace A by $A \oplus I$ and try again. If no such improvement exists, we apply an approximation algorithm to *Comp*(A) to find an independent set B; if B is larger than A, we replace A by B. If this does not succeed either, we terminate and return A, otherwise we look for small improvements again.

The first question that one may ask about this approach is the following: how large can "small" be so that the algorithm still runs in a polynomial time. In other words, what kind of size bound σ can we tolerate, so that the number of all conceivable improvements containing at most σ nodes is polynomial. The second question is: how large must σ be so that the algorithm delivers the promised approximation ratio.

The first question is easy to answer. Assume the neighbors of every vertex are sorted in an arbitrary way. Then every connected subgraph can be represented by any one of its depth-first traversals, which can be described by a start vertex and a sequence of choices of neighbors. The frequently used decision to go back in the tree can be encoded by one bit. This method shows that the number of connected subgraphs of G with at most σ nodes is smaller than $n(4\Delta)^\sigma$, where n is the size of G and Δ is the degree of G. Given that we consider the case where Δ is a constant, it suffices that σ is $O(\log n)$.

The answer to the next question is the subject of the next two sections. However, before we even attempt to address it, we shall pose this problem more precisely. For every $\Delta > 2$ and $k > 0$ we will show that there exists a polynomial time algorithm $SIC_{\Delta,k}$ that, given a graph of degree Δ having a MIS of size m, will find an independent set in G of size a, so that $\frac{m}{a} \leq \rho_\Delta + \frac{1}{k}$, where ρ_Δ equals $\frac{\Delta+3}{5}$ if Δ is even, and $\frac{\Delta+3.25}{5}$ if Δ is odd.

The algorithm $SIC_{\Delta,k}$ will have the form described above, with two clarifications. "Small" improvement shall mean an improvement with at most $c_{\Delta,k} \log n$ nodes, where the constant $c_{\Delta,k}$ will be found in Section 4. The second clarification concerns the algorithm used to approximate MIS in *Comp*(A) (to find B, a possible replacement of A). If $\Delta > 4$, then the degree of *Comp*(A) is at most $\Delta - 2$, and we will use $SIC_{\Delta-2,k}$ to find B. If

Δ equals 3 or 4, then $Comp(A)$ is a graph of degree at most 2, hence in linear time we can compute B to be an MIS of $Comp(A)$.

ALGORITHM 2.1. $SIC_{\Delta k}$:

```
If Δ ≤ 2 then
 compute an MIS exactly and stop
Let A be any maximal independent set
Repeat
 Do all possible improvements of size c_{Δ,k}
 If Δ = 3 then ℓ = 1 else ℓ = 2
 Recursively apply SIC_{Δ-ℓ,k} to Comp(A)
  and select the resulting independent
  set if it is bigger
Until no more improvements are found
```

3 Preliminaries to the algorithm analysis

This section contains graph-theoretical facts and definitions that will be used in the analysis of the algorithm. As multiple edges and loops would not affect independent sets, we usually don't allow them. Nevertheless, for some auxiliary constructions used for the analysis of our algorithm, we use multigraphs with loops and multiple edges. In all lemmas we will assume that $G = \langle V, E\rangle$ is a graph or multigraph with n nodes. Moreover, log will denote the logarithm with base 2, $\#S$ will denote the cardinality of set S.

DEFINITION 3.1. *A* lariat *is a connected subgraph induced by a set of nodes $A \subseteq V$ such that $\#E(A) \geq \#A > 0$. A binocular is a lariat with $\#E(A) > \#A$.*

Note that a typical lariat forms a loop with a chain attached to it, while two loops connected with a chain or two loops grown together form a binocular.

LEMMA 3.1. *Assume that every node in a multigraph $G = \langle V, E\rangle$ has degree at least 3. Then every node $v \in V$ belongs to a binocular with at most $4\log n - 1$ nodes.*

Proof. Consider a breadth first search tree of G where v is the root. If every node in this tree in distance less than m from v has at least two children, then this tree contains at least 2^m nodes. Hence there must be a node u in distance at most $\log n - 1$ from v, with at most one child. Because there are at least three edges incident to u, one of them is a cross edge, a loop or a multiple edge. Let this edge e connect u to a vertex w. Then the tree paths from v to u and w, taken together with the edge e, form a lariat L with at most $2\log n$ nodes. If L does not already define a binocular, we may shrink L to a single node $[L]$, and all nodes in the modified graph will still have degree at least 3. Hence, in the modified graph we can find a lariat $K \cup \{[L]\}$ with at most $2\log n$ nodes. $L \cup K$ is a binocular with at most $4\log n - 1$ nodes. □

LEMMA 3.2. *Assume that $\#E \geq \frac{k+1}{k}n$ in a multigraph $G = \langle V, E\rangle$. Then G contains a binocular with less than $4k\log n$ nodes.*

Proof. Let $G(U)$ be a smallest induced subgraph of G such that $\#E(U) \geq \frac{k+1}{k}\#U$. Then $G(U)$ cannot contain any degree 1 nodes. It is easy to see that any maximal chain of degree 2 nodes in U has less than k elements. Replace every such chain with a single edge connecting its ends. The resulting graph satisfies the assumption of Lemma 3.1; hence it contains a binocular B with some $m \leq 4\log n - 1$ nodes. Select $m+1$ edges contained in B, and replace those that are chain replacements with the respective chains of degree 2 nodes. The resulting graph is a binocular in G with at most $m + (m+1)(k-1) = (m+1)k - 1 < 4k\log n$ nodes. □

The preceding lemma allows us to provide sufficient conditions for the existence of improvements of a candidate independent set. First we need an additional definition. Below, A, B are two disjoint independent sets in V.

DEFINITION 3.2. *Let $\mathrm{Sol}(A, B)$ be the set of those elements of B that have exactly one neighbor in A, $\mathrm{Im}(A, B)$ is the set of those elements of A that have a neighbor in $\mathrm{Sol}(A, B)$, $\mathrm{Pair}(A, B)$ is the set of these elements of B that have exactly two neighbors in A and $E(A, B)$ is a set of edges containing one edge e from $u \in A$ to $v \in A$ for every common neighbor w in $\mathrm{Pair}(A, B)$ of u and v. In this case we say that w is a* representation *of the edge e.*

LEMMA 3.3. *If B is an independent set and $\#\mathrm{Pair}(A, B) \geq \frac{k+1}{k}\#A$, then A has an improvement contained in $A \cup B$ of size less than $8k\log n$.*

Proof. Let $s = 4k \log n - 1$. The number of edges $\#E(A,B)$ in the multigraph $G = \langle A, E(A,B) \rangle$ is at least $\frac{k+1}{k}\#A$. By Lemma 3.2, the multigraph G contains a binocular D with $t \leq s$ nodes that are joined by $t+1$ edges. The union of the nodes in D with the set of representations of these edges forms an improvement of A of size at most $2s+1$. □

LEMMA 3.4. *Let A and B be independent sets. Let $A_0 = A - \mathrm{Im}(A,B)$ and $B_0 = B - \mathrm{Sol}(A,B)$. Assume A_0 has an improvement contained in $A_0 \cup B_0$ of size s. Then A has an improvement contained in $A \cup B$ of size at most $max(s\Delta, \Delta^2+\Delta+1)$.*

Proof. Assume first that two nodes u, v of $Sol(A,B)$ have the same neighbor w in A. Then $\{u,v,w\}$ is an improvement of size 3 and we are done. Thus we may assume that every node of $Im(A,B)$ has exactly one neighbor in $Sol(A,B)$. Now we consider a minimum size improvement I of A_0 contained in $A_0 \cup B_0$; let $s = \#I$. Let I_A be the set of those nodes in $A - I$ that have neighbors in I. It is easy to see that $I_A \subseteq A - A_0 = Im(A,B)$. Let I_B be the set of nodes in $Sol(A,B)$ that have neighbors in I_A. It is also easy to see that $\#I_A = \#I_B$ and that I_B is disjoint from I. Therefore $I \cup I_A \cup I_B$ is an improvement of A contained in $A \cup B$.

Because I is connected, $E(A_0 \cup B_0)$ contains at least $s-1$ edges, all of them incident to $I \cap B_0$. Observe also that $\#(I \cap B_0) = (s+1)/2$. Thus the number of edges from $I \cap B_0$, and hence the size of I_A, is bounded by $\Delta(s+1)/2 - (s-1)$. Consequently, the size of the improvement $I \cup I_A \cup I_B$ is bounded by $s + \Delta(s+1) - 2s + 2 = \Delta s + \Delta + 2 - s$. For $s \geq \Delta + 2$ this is at most Δs, for smaller values of s this is at most $\Delta^2 + \Delta + 1$. □

The two previous lemmas allow to prove the following result, which is crucial in the analysis of the algorithm.

LEMMA 3.5. *Let A_0, B_0 be as in the previous lemma. Assume also that there are no improvements of A in $A \cup B$ of size below $32k\Delta^{4k}\log n$. Then*

$$\frac{\#B_0}{\#A_0} \leq \frac{\Delta+1}{3} + \frac{1}{3k}$$

Sketch of Proof. We will show the converse implication. Fix t to be $\#A_0/4k$. While $\#Sol(A_0,B_0) > t$, remove $Sol(A_0,B_0)$ from B_0 and $Im(A_0,B_0)$ from A_0. If at any step we remove more elements from A_0 than from B_0, that means that we have found an improvement and we are done. Note that if these removals were repeated $4k$ times, we would have removed all nodes from A_0 and obtain an improvement of the final A_0 of size 1. Otherwise, when we terminate and remove $Sol(A_0,B_0)$ from B_0, we have

$$\frac{\#B_0}{\#A_0} \geq \frac{\Delta + 1 + 1/4k}{3}$$

Note that at this point every node of B_0 is connected to at least two nodes of A_0. This fact, the above inequality and the degree bound of Δ imply that $\#Pair(A_0,B_0) \geq \frac{4k+1}{4k}\#A_0$. By Lemma 3.3 this implies that the final A_0 has an improvement in the final $A_0 \cup B_0$ of size at most $32k\log n$. Now assume $n \geq 2^{\frac{\Delta^2+\Delta+1}{32k\Delta}}$. Then $32k\Delta \log n \geq \Delta^2+\Delta+1$, and we can apply Lemma 3.4 repeatedly to infer that A has an improvement in $A \cup B$ of size at most $32k\Delta^{4k}\log n$. □

4 Algorithm analysis

We first describe the analysis for even Δ. The remaining cases are similar. We fix $c_{\Delta,k}$ to be $32k\Delta^{4k}$. Assume that A' is the result of our algorithm, and that B' is an MIS. Let $C = A' \cap B'$, $A = A' - C$, $B = B' - C$. Define A_0 and B_0 as in Lemma 3.4, and let $A_1 = A - A_0$, $B_1 = B - B_0$. Finally, denote the cardinalities of these sets by respective lower case letters.

Note that an improvement of A in $A \cup B$ is an improvement of A'. This follows from the fact that there are no edges between B and $A' - A = C$. Because A' is returned by the algorithm $SIC_{\Delta,k}$, it does not have improvements of size $32k\Delta^{4k}\log n$. From Lemma 3.5 we get the inequality

$$b_0 \leq \left(\frac{\Delta+1}{3} + \frac{1}{3k}\right) a_0 \qquad (1)$$

Note that B_0 is an independent set contained in $Comp(A')$. Because we have applied $SIC_{\Delta-2,k}$ to $Comp(A')$ and haven't found an independent set larger than A', we get the inequality

$$b_0 \leq \left(\frac{\Delta+1}{5} + \frac{1}{k}\right)(a_0 + a_1 + c) \qquad (2)$$

Note that $B_1 = Sol(A, B)$ and $A_1 = Im(A, B)$. As we have discussed before, $Sol(A, B)$ has not more elements than $Im(A, B)$. Otherwise, some two nodes of B had together only one neighbor in A, which led to a size 3 improvement of A, and hence of A'. Therefore

$$b_1 + c \leq a_1 + c \qquad (3)$$

By multiplying (1) by $\frac{3}{\Delta+1}$, (2) by $\frac{\Delta-2}{\Delta+1}$, (3) by 1 and adding them together, we obtain

$$b_0 + b_1 + c < \left(\frac{\Delta+3}{5} + \frac{1}{k}\right)(a_0 + a_1 + c) \qquad (4)$$

If Δ is odd and greater than 3, we just replace the inequality (2) by

$$b_0 \leq \left(\frac{\Delta+1.25}{5} + \frac{1}{k}\right)(a_0 + a_1 + c) \qquad (5)$$

We still multiply (1) by $\frac{3}{\Delta+1}$, (5) by $\frac{\Delta-2}{\Delta+1}$, (3) by 1, and add them together to obtain

$$b_0 + b_1 + c < \left(\frac{\Delta+3.25}{5} + \frac{1}{k}\right)(a_0 + a_1 + c) \quad (6)$$

Finally, if $\Delta = 3$, inequality (2) is replaced by

$$b_0 + b_1 \leq a_0 + a_1 + c \qquad (7)$$

Again applying the same weights to these inequalities, we conclude

$$b_0 + b_1 + c < (\frac{5}{4} + \frac{1}{k})(a_0 + a_1 + c) \qquad (8)$$

5 Open Problems

Obviously, the ultimate goal is to match upper and lower bounds on the i.p.r. for this basic SNP-complete problem. We conjecture that a slightly modified version of our algorithm, which gives vertices of degree 2 a higher chance of being included in the independent set A, can approach an i.p.r. of $\frac{5}{6}$ instead of $\frac{4}{5}$ for graphs of degree 3. This would extend the better i.p.r. of $\frac{5}{\Delta+3} - \varepsilon$ from graphs of bounded even degree to graphs of arbitrary bounded degree.

Currently, the running time is bounded by a polynomial of huge degree (depending on Δ and k). This poses several interesting problems. Is there a practical algorithm for $\Delta = 3$ and other small values of Δ with k reasonably large? Is there an approximation algorithm whose running time is a polynomial of a degree independent of Δ? The dependence of the running time on $k \approx \frac{1}{\varepsilon}$ is not yet of theoretical interest. Currently, we pay a high price in terms of running time depending on k, as we approach the best known performance ratio. If in the far future the true limit for the approximation ratio is known, it would be interesting to see whether one has still to pay such a high price when approaching the truly best performance ratio. In other words, how suddenly does this approximation problem get more difficult when the performance ratio is pushed to the limit?

Our algorithm is combinatorial in nature and uses specific methods, especially designed for the MIS problem. On the other hand we know that all SNP-complete problems are fairly closely related. Can our method be extended to similar SNP-complete problems?

References

[ALMSS] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, Proof verification and hardness of approximation problems, *Proc. 33rd FOCS*, (1992), 14-23.

[BGLR] M. Bellare, S. Goldwasser, C. Lund and A. Russell, Efficient Probabilistically Checkable Proofs and Applications to Approximation, *Proc. 25th STOC*, (1993), 294–304.

[B] P. Berman, Manuscript 1993.

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

[Br] R. L. Brooks, On Coloring the Nodes of a Network *Proc. Cambridge Philos. Soc.* 37, (1941), 194–197.

[GJ] M. Garey and D. S Johnson, *Computers and intractability*, W.H. Freeman and Company, 1979.

[HR1] M. M. Halldórsson and J. Radhakrishnan. Better bounded-degree independent sets via clique removal. In preparation.

[HR2] M. M. Halldórsson and J. Radhakrishnan. The relative size of greedy independent sets in sparse and bounded-degree graphs. In preparation.

[H] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover, and set packing problems. *Disc. Applied Math.* 6, (1982), 243–254.

[J] D.S. Johnson, The NP-Completeness Column: An Ongoing Guide, On multi-prover interactive proofs, probabilistically checkable proofs and approximation algorithms, *J. of Algorithms* 13, (1992), 502–524.

[L] L. Lovász, Three Short Proofs in Graph Theory, *J. Combin. Theory (B)* 19, (1975), 269–271.

[MS] B. Monien and E. Speckenmeyer. Some further approximation algorithms for the vertex cover problem. In *Lecture Notes in Computer Science* 159, Springer Verlag, (1983), 341–349.

[NT] G. L. Nemhauser and W. T. Trotter. Vertex packing: structural properties and algorithms. *Math. Programming* 8, (1975), 232–248.

[P] V. T. Paschos. A $(\delta/2)$-approximation algorithm for the maximum independent set problem. *Inf. Process. Lett.* 44, (1992), 11–13.

[PY] C. H. Papadimitriou and M. Yannakakis, Optimization, approximation and complexity classes, *JCSS* 43, (1991), 425–440.

[Y] M. Yannakakis, On the approximation of maximum satisfiability, *Proc. 3rd SODA*, (1992), 1–9.