Improved Approximations of Independent Sets in Bounded-Degree Graphs

Magnús M. Halldórsson

School of Information Science
Japan Adv. Inst. of Science and Tech.
Ishikawa 923-12, JAPAN
magnus@jaist.ac.jp

Jaikumar Radhakrishnan

Theoretical Computer Science Group
Tata Institute of Fundamental Research
Bombay, India
jaikumar@tifrvax.bitnet

Abstract. Finding maximum independent sets in graphs with bounded maximum degree is a well-studied NP-complete problem. We study two approaches for finding approximate solutions, and obtain several improved performance ratios.
The first is a subgraph removal schema introduced in our previous paper. Using better component algorithms, we obtain an efficient method with a $\Delta/6(1+o(1))$ performance ratio. We then produce an implementation of a theorem of Ajtai et al. on the independence number of clique-free graphs, and use it to obtain a $O(\Delta/\log\log\Delta)$ performance ratio with our schema. This is the first $o(\Delta)$ ratio.
The second is a local search method of Berman and Fürer for which they proved a fine performance ratio but by using extreme amounts of time. We show how to substantially decrease the computing requirements while maintaining the same performance ratios of roughly $(\Delta+3)/5$ for graphs with maximum degree Δ. We then show that a scaled-down version of their algorithm yields a $(\Delta+3)/4$ performance, improving on previous bounds for reasonably efficient methods.

1 Introduction

An *independent set* in a graph is a set of vertices in which no two are adjacent. The problem of finding an independent set of maximum cardinality is of central importance in graph theory and combinatorial optimization.

Given that the problem is NP-hard, the approach that holds the greatest promise is in developing heuristics that find high-quality approximate solutions. The performance ratio of such an algorithm is defined to be the worst-case ratio of the size of the optimal solution to the size of the algorithm's solution. In spite of considerable effort, no algorithm is known for the independent set problem with a performance ratio less than $O(n/\log^2 n)$ [6], where n is the number of vertices in the input graph. Results in recent years on interactive proof systems, culminating in the celebrated paper of Arora et al [3], show that no constant factor approximation can be expected, and in fact, a $n^{1/4}$ ratio appears out of reach [4].

Given this apparent hardness of the general problem, it is natural to ask what restrictions make the problem easier to approximate. Perhaps the most natural and frequently occurring case is when the maximum vertex degree is bounded above by a constant. Just as the independent set (or clique) problem occurs in various context when modeling pairwise conflicts among elements, the bounded-degree variant occurs naturally when key parameters of the problems are fixed.

For the bounded-degree version (B-IS), the exact problem remains NP-complete, but the approximation problem becomes considerably easier. In fact, any algorithm that finds a *maximal* independent set has a performance ratio of Δ on graphs with maximum degree Δ. This problem is also among the original MAX SNP-complete problems [11], so the results of [3] imply that there is a constant $c > 1$ for which c-approximation becomes NP-hard, thus no polynomial-time approximation schema can exists (unless $P = NP$). This naturally leads us to ask for the best possible constant within which B-IS can be approximated.

We address this question by studying two recent approaches. The former is an algorithm schema that involves removing small cliques from the graph. The idea – which originates in [6], with traces back to Erdős[7] – comes from the observation that graphs without small cliques contain provably larger independent sets than general graphs do. Moreover, these larger solutions can be found effectively. For graphs with few disjoint cliques, we can manually remove all the cliques and find the promised improved solution on the remainder. On the other hand, graphs with many disjoint cliques cannot contain a very large independent set, providing an upper bound on the optimal solution. Hence, in either case, our performance ratio will be improved.

We previously used this schema to improve the approximation of the minimum degree greedy algorithm from $(\Delta+2)/3$ to $\Delta/3.81(1+o(1))$ [8]. This time, using new analysis of a simple local search algorithm of Khanna et al. [10], we obtain a surprisingly strong $\Delta/6(1+o(1))$ ratio.

We also use this schema to answer a tantalizing question: Given that all B-IS approximation results so far have merely improved the coefficient in front of Δ, is a $o(\Delta)$ performance ratio possible? The answer is affirmative; we present an algorithm with a $O(\Delta/\log\log\Delta)$ performance ratio. As a crucial step, we give a deterministic implementation of an existential theorem of Ajtai, Erdős, Komlós, and Szemerédi [1] on the independence number of sparse graphs containing no small cliques.

The latter approach that we consider is a method due to Berman and Fürer [5], that can be characterized as a local search algorithm that additionally searches in the complement of the current solution. Their algorithm yields excellent performance ratios on graphs of low maximum degree: $(\Delta+3)/5+1/h$, when Δ is even, and $(\Delta+3.25)/5+1/h$ when Δ is odd, for any fixed constant h.

Unfortunately, the method is extremely time consuming. The local search neighborhood, the set of solutions searched in for an improvement, involves all solutions whose difference (or distance) from the current one is a connected graph with σ vertices. In order to obtain their result, we must search a neighborhood of size $\sigma = 32h\Delta^{4h}\log n$. That implies a search complexity of $n(4\Delta)^{\sigma} \approx n^{64h\Delta^{4h}\log\Delta}$. In particular, the dependence of the complexity on the quality of the approximation h is doubly exponential. For instance, in order to obtain a ratio of 1.6 when $\Delta = 4$, this approach requires $n^{2^{90}}$ time complexity.

We address this feasibility concern by tightening the analysis, and thereby reducing the neighborhood size requirements considerably. In particular, we obtain identical performance ratios while shrinking the neighborhood size to $4h^{2.6}\Delta \log n$, eliminating the exponential dependence on h. While still not exactly practical, this is close to the limitations of this approach.

We also observe that small neighborhoods yield surprisingly good performance ratios. Using only 2Δ size neighborhood, which can easily be implemented in $O(n^2)$ time, we obtain a $(\Delta + 3)/4$ ratio. This improves on the previous best $(\Delta + 2)/3$ ratio for practical bounded-degree independent set algorithms [8].

Notation We use fairly standard graph terminology. For the graph in question, usually denoted by G, n denotes the number of vertices, Δ maximum degree, $\overline{d}$ average degree, α independence number (or size of the maximum independent set), and τ independence fraction (or the ratio of the independence number to the number of vertices). For a vertex v, $d(v)$ denotes the degree of v, and $N(v)$ the set of neighbors of v.

For an independent set algorithm Alg, $Alg(G)$ is the size of the solution obtained by the algorithm on graph G. The performance ratio of the algorithm in question is defined by

$$\rho = \max_G \frac{\alpha(G)}{Alg(G)}.$$

2 Subgraph Removal Approach

We present a strategy for approximating independent sets, based on removing cliques. Using this idea, we first obtain an asymptotically improved $O(\Delta/\log\log\Delta)$ performance ratio, via a constructive proof of a graph theorem. We then use two practical algorithms from the literature to obtain improved bounds for graphs of intermediate maximum degree.

Generic Clique Removal Schema

We present an algorithm schema, indexed by a cardinal k and a collection of subordinate procedures. One is algorithm General-BDIS-Algorithm for finding independent sets in general (bounded-degree) graphs. The others are methods for finding independent sets in ℓ-clique free graphs, possibly one for each value of ℓ, $3 \le \ell \le k$.

CliqueRemoval$_k(G)$
  $A_0 \leftarrow$ General-BDIS-Algorithm(G)
  for $\ell = k$ downto 2 do
    $S \leftarrow$ CliqueCollection(G,ℓ)
    $G \leftarrow G - S$

$A_\ell \leftarrow K_\ell$-free-BDIS-Algorithm(G)
od
Output A_i of maximum cardinality
end

The algorithm CliqueCollection finds in G a maximal collection of disjoint cliques of size ℓ; in other words, S is a set of mutually non-intersecting cliques of size ℓ such that the graph $G - S$ contains no ℓ cliques. Such a collection can be found in $O(\Delta^{\ell-1}n)$ time by searching exhaustively for a $(\ell-1)$-clique in the neighborhood of each vertex. That is polynomial whenever $\ell = O(\log_\Delta n)$.

We now present two instances of this schema.

AEKS

Ajtai, Erdős, Komlós and Szemerédi [1] proved the following result about K_ℓ free graphs.

Theorem 1 (AEKS) *There exists an absolute constant c_1 such that for any K_ℓ-free graph G,*

$$\alpha(G) \geq c_1 \frac{\log((\log \overline{d})/\ell)}{\overline{d}} n.$$

We have obtained an algorithm AEKS that constructs such an independent set in polynomial time, by derandomizing the parts of the proof of [1] where probabilistic existence arguments are used. For lack of space, its somewhat lengthy description is omitted here.

It suffices to use the following simplified algorithm to approximate independent sets. An independent set is *maximal* (MIS) if adding any further vertices to the set violates independence. An MIS is easy to find and provides a sufficient general upper bound of $n/(\Delta+1)$.

AEKS-SR(G)
$G' \leftarrow G$ − CliqueCollection$(G, c_1 \log\log \Delta)$
return max(AEKS(G'), MIS(G))
end

Theorem 2. *The performance ratio of* AEKS-SR *is* $O(\Delta/\log\log\Delta)$.

Proof. Let k denote $c_1 \log\log\Delta$, and let n' denote the size of $V(G')$. The independence number collects at most one from each k-clique, for at most

$$\alpha \leq n/k + n' \leq 2\max(n/k, n'),$$

while the size of the solution found by AEKS-SR is at least

$$\text{AEKS-SR}(G) \geq \max(\frac{1}{\Delta+1}n, \frac{k}{\Delta}n') \geq \frac{k}{\Delta+1}\max(n/k, n').$$

The ratio between the two clearly satisfies the claim. ∎

Observe that the combined method runs in polynomial time for Δ as large as $n^{1/\log\log n}$.

Effective method for moderately large maximum degree

While the clique removal method in combination with AEKS yields a good asymptotic performance ratio, Δ must be quite high for the gained $\log\log\Delta$ factor to overcome the large constants involved (implicit in the proof of [1]).

We now turn our attention to practical methods that can benefit from the clique removal schema. We present methods that combine to yield an asymptotic $\Delta/6(1+o(1))$ performance ratio, improving on the best previous known ratios for moderate to large values of Δ. This involves an algorithm of Shearer [12] for 3-clique-free graphs, and a simple local search algorithm for other k-clique-free graphs as well as for use as the general BDIS algorithm.

2-opt. Khanna et al. [10] studied a simple local search algorithm that we have named 2-opt. Starting with an initial maximal independent set, it tries all possible ways of adding two vertices and removing only one while retaining the independence property. We say that a triple $\langle v_1, v_2, u\rangle$ is a *2-improvement* of an independent set I iff vertices v_1, v_2 are outside of I, u is in I, and adding the former two to I while removing the latter retains the independence property. Since I can be assumed to be a maximal independent set, it suffices to look at pairs adjacent to a common vertex in I.

The method is a simplification of the algorithm presented in the following section (it omits the recursive call). Using proper data structures, it can be implemented in $O(poly(\Delta)n)$ time.

The following was shown by Khanna et al [10].

Lemma 3. 2-opt $\geq \dfrac{1+\tau}{\Delta+2}n.$

They proved a $\Delta/2.44(1+o(1))$ performance ratio of this algorithm combined with another simple algorithm. A better bound can be obtained via a technique of Nemhauser and Trotter (see [9] for application) which effectively allows one to assume without loss of generality that the independence fraction τ is at most 1/2. A ratio of $(\Delta+2)/3$ then follows easily, for the combination of these two methods. But we digress.

We can get improved bounds for k-clique free graphs.

Lemma 4. *On a k-clique free graph G,* 2-opt$(G) \geq \dfrac{2}{\Delta+k}n.$

Proof. Since A is a maximal independent set, each vertex in $V-A$ must have at least one edge coming into the set A. If the graph has no k-clique, then for each $u \in A$, at most $k-1$ vertices can be adjacent only to u and no other vertex in A. Thus, at most $(k-1)|A|$ vertices can be adjacent to only one vertex in A. Hence, if we sum up the necessary degrees of vertices from $V-A$ into A, we find that $|A|\Delta \geq (n-|A|)+(n-(k-1)|A|)$, which yields the lemma. ∎

Shearer. Shearer [12] proved the following theorem, improving a previous result of Ajtai, Komlós and Szemerédi [2].

Theorem 5 (Shearer [12]) *Let $f_s(d) = (d \log_e d - d + 1)/(d-1)^2$, $f_s(0) = 1$, $f_s(1) = \frac{1}{2}$. For a triangle-free graph G, $\alpha(G) \geq f_s(\overline{d})n$.*

Moreover, he gave a simple algorithm (which we name after him) attaining the bound, which repeatedly selects any vertex v of degree d_v satisfying

$$(d_v + 1) f_s(\overline{d}) \;\leq\; 1 + (\overline{d} d_v + \overline{d} - 2 \sum_{w \in N(v)} d(w)) f_s'(\overline{d}),$$

removes it and its neighbors from the graph, and repeats until the graph is empty. Using an appropriate data structure to maintain the f-values of the vertices, the algorithm can be implemented in time $O(poly(\Delta)n)$. In fact, the claim is also obtained in fully linear time by a simple randomized greedy algorithm, that randomly selects a non-adjacent vertex in each step. We shall only need the obvious corollary that $\mathsf{Shearer}(G) \geq f_s(\Delta)n \approx n(\log \Delta)/\Delta$.

Analysis. To improve the approximation further, we apply the method of Nemhauser and Trotter on each incarnation of G. That will allow us to assume that nothing will be left after the edges (2-cliques) are removed.

We obtain the following explicit, if less than compact, bound on the performance ratio. H_k is the k-th Harmonic number.

Theorem 6. $\mathsf{CliqueRemoval}_k$*, using* 2-opt *and* Shearer *attains a performance ratio of at most*

$$\left[\frac{\Delta}{2} + 2 + \frac{k}{2}\left(H_{k-1} + \frac{1}{3f_s(\Delta)} - \frac{3}{2} + \frac{\Delta}{3}\right)\right] / (k+1)$$

for graphs of maximum degree $\Delta \geq 5$.

Proof. Let n_t denote the number of vertices in the t-clique free graph. Thus, $n \geq n_k \geq \ldots \geq n_3 \geq n_2 \geq 0$. From applying Nemhauser-Trotter, we may assume $n_2 = 0$.

The size of the optimal solution is τn, which can be bounded by

$$\tau n \;\leq\; \tfrac{1}{2}(n_3 - n_2) + \cdots + \tfrac{1}{k}(n_{k+1} - n_k) \;=\; \sum_{i=3}^{k} \frac{1}{i(i-1)} n_i + \frac{1}{k} n. \qquad (1)$$

Our algorithm is guaranteed to output a solution of size at least

$$\max\left[\frac{1+\tau}{\Delta+2} n, \max_{4 \leq t \leq k} \frac{2}{\Delta + t} n_t, f_s(\Delta) n_3\right].$$

Thus, the performance ratio ρ attained by the algorithm is bounded by

$$\rho \;\leq\; \min\left[\frac{\tau n}{\frac{1+\tau}{\Delta+2} n}, \frac{\tau n}{\frac{2}{\Delta+t} n_t}, \frac{\tau n}{f_s(\Delta) n_3}\right].$$

From this we derive, respectively, that

$$\tau \geq \frac{\rho}{\Delta + 2 - \rho}, \tag{2}$$

$$n_t \leq \frac{\tau}{\rho} \frac{\Delta + t}{2} n, \quad t = 4, 5, \ldots, k \tag{3}$$

$$n_3 \leq \frac{\tau}{\rho} \frac{1}{f_s(\Delta)} n. \tag{4}$$

Combining (1), (4) and (3), we find that

$$\tau \leq \frac{\tau}{\rho} s_{\Delta,k} + \frac{1}{k}$$

where

$$s_{\Delta,k} = \frac{1}{6 f_s(\Delta)} + \sum_{i=4}^{k} \frac{\Delta + i}{2i(i-1)} = \frac{1}{2} \left[\frac{1}{3 f_s(\Delta)} + (H_{k-1} - \frac{3}{2}) + \Delta(\frac{1}{3} - \frac{1}{k}) \right].$$

Thus,

$$\tau \leq \frac{1}{k(1 - s_{\Delta,k}/\rho)}. \tag{5}$$

Hence, from (2) and (5)

$$\frac{\rho}{\Delta + 2 + \rho} \leq \frac{1}{k(1 - s_{\Delta,k}/\rho)}$$

which simplifies to

$$\rho \leq \frac{\Delta + 2 + k s_{\Delta,k}}{k+1}$$

and the claim follows. ∎

It is now easy to compute the ratio for particular values of Δ. Selected values are presented in section 4.

It is also easy to see that if Δ and k are assumed to be growing functions, then the $\Delta/3$ term will dominate for a $\Delta/6$ asymptotic ratio.

Corollary 7. CliqueRemoval, *using* 2-opt *and* Shearer, *attains a performance ratio of* $\Delta/6\ (1 + o(1))$.

3 Local Search Approach

The Algorithm

The algorithm, due to Berman and Fürer [5], is a type of a local search algorithm – with a twist. The locality, or neighborhood, is the natural one: the set of solutions that differ (in terms of symmetric set difference) from the current one in only few vertices. Starting with an arbitrary solution (e.g. a maximal independent set), we search through all neighboring solutions for one that is strictly larger. This process is repeated until no further improvements can be found.

The twist to the tale is a recursive application of this method on what can be thought of as the *complement* of the solution. Once a non-improvable solution A is found, the method is applied to $\text{COMP}(A)$, defined to be the subgraph induced by nodes with at least two neighbors in A (when $\Delta = 3$, one neighbor suffices). Since the maximum degree of this subgraph is two less, the recursion ends on degree two graphs which we solve optimally. Given this new alternative solution, if larger, we make it our current solution and continue to try to improve it; otherwise, we exit and proclaim optimality under this type of search.

```
LS(G)
  if (Δ ≤ 2) return MaximumIndependentSet(G)
  A ← MaximalIndependentSet(G)
  repeat
    try all possible t-improvements
    A2 ← LS(Comp(A))
    if (|A2| ≥ |A|)
      A ← A2
  until (no more improvements are found)
end
```

Paraphrasing, in each step we search for an *improvement* which is a vertex set whose symmetric difference with the current solution is larger and still independent. A em t-improvement adds t and removes $t-1$ vertices from the current solution. It is easy to see that it suffices to look for an improvement that induces a connected graph, and since the maximum degree is bounded by Δ, there are at most $\Delta^{2(2t-1)}n$ connected subgraphs of size $2t-1$. The size of the locality searched, indicated by t, is therefore the crucial factor in the complexity. We say that a solution is *t-optimal* if it has no t-improvement nor an improvement in Comp, and call an algorithm finding such a solution *t-opt*.

Tools for analysis

We analyze the guaranteed size of any t-optimal solution. Let A refer to such a solution, and B refer to some hypothetical optimal solution. The *lace* of a vertex in B is defined to be the set of adjacent vertices in A. The gist of the analysis is to show that few vertices can have a small lace, thus limiting the size of B given the limited adjacency capacity of A that results from the degree bound.

Notation. Let C denote the intersection of A and B. Let B_1 denote the subset of B of of vertices whose laces are unit size, (i.e. adjacent to precisely one vertex of A), and let A_1 be the set of vertices of A adjacent to the vertices of B_1. Let $A_0 = A - A_1 - C$ and $B_0 = B - A_1 - C$. We further split B_0 into B_2 and B_3, namely vertices with lace of size two, and three or more, respectively.

Finally, denote the cardinalities of the above sets by their respective lower case letter. Denote by ρ^{Δ} the performance ratio of this algorithm on graphs of maximum degree Δ.

Simple bounds. The sum of the lace sizes can add up to at most the adjacency capacity of A. This gives us the first crucial bound.

$$b_1 + 2b_2 + 3b_3 \leq \Delta a. \tag{6}$$

The second bound is on the size of B_1. By definition, $b_1 \geq a_1$. On the other hand, a vertex in A adjacent to two vertices in B_1 forms a 2-improvement. Thus the equality is strict under 2-opt.

$$b_1 = a_1 \tag{7}$$

We shall also use it in the form of $b_1 + c = a_1 + c$.

Bounds on the number of pairs. From lemma 3.3 of [5], $4h \log n$ + 2-opt guarantees

$$b_2 \leq (1 + \frac{1}{h})a_0 \tag{8}$$

This is shown by constructing a graph, whose vertex set is A_0 and whose edges are formed by the laces of nodes in B_2. At least one element of a lace must be inside A_0; those with only one endpoint in A_0 are modelled as a self-loop on that node. If the number of edges in the resulting multigraph exceeds the above bound, it can be shown that there exists an induced subgraph with at most $4h \log n - 1$ nodes containing strictly more edges. Since the graph may be assumed to be connected, it contains at most two self-loops. This then maps to a set of at most $4h \log n$ nodes in B_2, two nodes in B_1 and a corresponding set of nodes in A, that together constitute an improvement.

Bounds obtained by recursion. The recursive application provides us with a bound on b_0. If no larger solution can be found, the maximum independent set in $\mathsf{Comp}(A)$ cannot be greater than a by a factor of more than the performance ratio for $(\Delta - 2)$-degree graphs. But, this complement – the set of vertices adjacent to at least two vertices in A – must contain B_0 (and B_1 when $\Delta = 3$). Thus we obtain:

$$b_0 \leq \rho^{\Delta-2} a \tag{9}$$

$$b_1 + b_0 \leq \rho^{\Delta-1} a. \tag{10}$$

Derivative. We now introduce the concept of a *derivative* of the partitions A_0 and B_0, similar to the ideas of [5, lemma 3.5]. Removing A_1 from the graph splits B_0 and A_0 again into two parts: $B' = B_0 = B'_0 \cup B'_1$, $A' = A_0 = A'_0 \cup A'_1$. This procedure can be continued, producing second and third derivatives A''_0, $A^{(3)}_0$ etc.

The important observation is that inequalities (7), (6), and (8) hold equally for derivatives, with an additional Δ factor in neighborhood size.

Lemma 8. *A t-improvement of A' implies a $t\Delta$-improvement of A.*

Proof. Consider the t vertices of B' in a t-improvement of A'. Each lace contains some vertex in $A' = A_0$ and thus at most $\Delta - 1$ vertices in A_1. The vertices in A_1 are in one-to-one correspondence with vertices in B_1, whose laces in turn do not contain any further vertices in A. The combination of the abovementioned vertices includes at most $t\Delta$ vertices in B, and one fewer in A, and thus forms a $t\Delta$-improvement of A. ∎

We need to relate the sizes of the 'unit' sets, A_1, A_1' etc. Each vertex in B_1' is adjacent to a particular vertex in A_1'. In addition, it must be adjacent to some vertex in A_1, since otherwise it belongs to B_1. Each vertex in A_1 is adjacent to a vertex in B_1, and thus at most $\Delta - 1$ vertices in B_1'. Hence, we obtain the useful bound:

$$b_1' \leq (\Delta - 1)a_1. \tag{11}$$

Applying the tools

Fast local search. We first give a simple proof of a $(\Delta + 3)/4$ performance ratio for 2Δ-opt, which is an improvement over previously known bounds for practical methods.

Add (7) and (6) and take the derivative:

$$b_0 \leq \frac{\Delta + 1}{2} a_0. \tag{12}$$

For $\Delta = 3$, we add half (12), half (9), and once (7) to get

$$\rho^3 \leq 3/2.$$

In general, we add $2/(\Delta + 1)$ times (12), $(\Delta - 1)/(\Delta + 1)$ times (9), and once (7):

$$\rho^\Delta \leq 1 + \frac{\Delta - 1}{\Delta + 1}\rho^{\Delta - 2}.$$

This is bounded by $(\Delta + 3)/4$, and is slightly better for small even values of Δ.

Slow local search, faster. Let us now derive performance bounds identical to [5].

Adding (6), (8), and twice (7) yields

$$b \leq \frac{\Delta + 1 + 1/h}{3} a + (\frac{1 - 1/h}{3}) a_1$$

and s-th derivative gives

$$b^{(s)} = b_1^{(s)} + b_0^{(s)} \leq \frac{\Delta + 1 + 1/h}{3} a^{(s)} + \frac{1 - 1/h}{3} a_1^{(s)}.$$

That implies that by using derivatives of (7) and (11), we get

$$\begin{aligned}
b_0 &= b_1' + b_1'' + \ldots + b_1^{(s-1)} + b_1^{(s)} + b_0^{(s)} \\
&\leq a_1' + a_1'' + \ldots + a_1^{(s-1)} + \frac{\Delta + 1 + 1/h}{3} a^{(s)} + \frac{1 - 1/h}{3} a_1^{(s)} \\
&= \frac{\Delta + 1 + 1/h}{3} a_0 + \frac{1 - 1/h}{3} a_1^{(s)} - \frac{\Delta - 2 + 1/h}{3} (a_1' + a_1'' + \ldots + a_1^{(s-1)}) \\
&\leq \frac{\Delta + 1 + 1/h}{3} a_0 + \frac{1/(\Delta - 1)^{s-1} - 1/h}{3} a_1^{(s)}
\end{aligned}$$

Thus, as long as $h \leq (\Delta - 1)^{s-1}$, that is $s \geq 1 + \lceil \log h / \log(\Delta - 1) \rceil$, we obtain the essential inequality:

$$b_0 \;\leq\; \frac{\Delta + 1 + 1/h}{3} a_0. \tag{13}$$

We need s derivatives, resulting in an *improvement* requirement of

$$8h\Delta^s \log n \leq 8h\Delta h^{\log \Delta / \log \Delta - 1} \log n \;\leq\; 8h^{2.6} \Delta \log n.$$

The analysis of the performance guarantee now follows from the argument of [5, sec. 4]. Add $\frac{3}{\Delta+1}$ times (13), $\frac{\Delta-2}{\Delta+1}$ times (9) (for $\Delta = 3$, use (10)), and once (7) to obtain the recurrence:

$$\rho^{\Delta} \;\leq\; \left[1 + \frac{1}{h(\Delta + 1)} + \frac{\Delta - 2}{\Delta + 1}\right] \rho^{\Delta - 2}$$

which yields the desired ratios of $(\Delta + 3)/5 + 1/h$ ($(\Delta + 3.25)/5 + 1/h$) for even (odd) Δ.

In conclusion:

Theorem 9. LS *attains a $(\Delta + 3)/4$ performance ratio in time $O(\Delta^{\Delta} n^2)$, and a $(\Delta + 3)/5 + 1/h$ (plus 0.25 if Δ is odd) ratio in time $n^{O(h^{2.6} \Delta \log \Delta)}$.*

4 Comparison of Results

The results presented excel at different ranges of values of Δ. The local search approach is best for small values, the $\Delta/6$-ratio from subgraph removal is best for intermediate values, while eventually for large enough values, the asymptotically superior $O(\Delta / \log\log \Delta)$ bound wins. The intermediate one beats the $(\Delta+3)/5+1/h$ ratio of [5] for $\Delta \geq 613$, and the $(\Delta+3)/4$ ratio for $\Delta \geq 31$. We are left with four nearly incomparable results ruling the approximability landscape. Bounds for selected values of Δ are given below:

Δ	CliqRem	[5]	2Δ-opt	[8]
10	3.54	2.60	3.25	4.00
33	8.92	7.25	9.00	11.66
100	23.01	20.60	25.75	34.00
1024	201.57	205.40	256.75	342.00
8192	1535.20	1639.00	2048.75	2731.33

An interesting future topic would be to strengthen the lower bounds; currently, no lower bound is known to us that increases as a function of maximum degree.

References

1. M. Ajtai, P. Erdős, J. Komlós, and E. Szemerédi. On Turán's theorem for sparse graphs. *Combinatorica*, 1(4):313–317, 1981.
2. M. Ajtai, J. Komlós, and E. Szemerédi. A note on Ramsey numbers. *J. Combin. Theory Ser. A*, 29:354–360, 1980.
3. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. 33nd IEEE Symp. on Found. of Comp. Sci.*, pages 14–23, Oct. 1992.
4. M. Bellare and M. Sudan. Improved non-approximability results. To appear in STOC '94, May 1994.
5. P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *Proc. Fifth ACM-SIAM Symp. on Discrete Algorithms*, Jan. 1994.
6. R. B. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, June 1992.
7. P. Erdős. Some remarks on chromatic graphs. *Colloq. Math.*, 16:253–256, 1967.
8. M. M. Halldórsson and J. Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. To appear in STOC '94, May 1994.
9. D. S. Hochbaum. Efficient bounds for the stable set, vertex cover, and set packing problems. *Disc. Applied Math.*, 6:243–254, 1983.
10. S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computation views of approximability. Manuscript, Dec. 1993.
11. C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity. *J. Comput. Syst. Sci.*, 43:425–440, 1991.
12. J. B. Shearer. A note on the independence number of triangle-free graphs. *Discrete Math.*, 46:83–87, 1983.