Approximating Maximum Independent Sets by Excluding Subgraphs

Ravi Boppana[1] and Magnús M. Halldórsson[2]

Department of Computer Science
New York University
New York, NY 10012
and
Rutgers University

Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract

An approximation algorithm for the maximum independent set problem is given, improving the best performance guarantee known to $\mathcal{O}(n/(\log n)^2)$. We also obtain the same performance guarantee for graph coloring. The results can be combined into a surprisingly strong *simultaneous* performance guarantee for the clique and coloring problems.

The framework of *subgraph excluding* algorithms is presented. We survey the known approximation algorithms for the independent set (clique), coloring, and vertex cover problems and show how almost all fit into that framework. It is shown that among subgraph excluding algorithms, the ones presented achieve the optimal asymptotic performance guarantees.

1 Introduction

An *independent set* in a graph is a set of vertices with no edges connecting them. The problem of finding an independent set of maximum size is one of the classical $\mathcal{NP}$-hard problems. We consider polynomial time algorithms that find an approximation of guaranteed size. The quality of the approximation is given by the ratio of the size of the maximum independent set to the size of the approximation found, and the largest such ratio over all inputs gives the *performance guarantee* of the algorithm.

A few other problems are closely related to the independent set problem. A *clique* is a set of mutually connected vertices. Since finding the maximum size clique in a graph is equivalent to finding the maximum independent set in the complement of the graph, the clique problem is for our purposes the identical problem.

A *vertex cover* is a set of vertices with the property that every edge in the graph is incident to some vertex in the set. Note that vertices not in a given vertex cover must be independent, hence finding a maximum independent set is equivalent to finding a minimum vertex cover. Approximations of the two problems, however, are widely different.

The third related problem is *graph coloring*, namely finding an assignment of as few colors as possible to the vertices so that no adjacent vertices share the same color. Because the colors induce a partition of the graph into independent sets, the problems of approximating independent set and coloring are closely related.

[1]Supported in part by National Science Foundation Grant number CCR-8902522
[2]Supported in part by Center for Discrete Mathematics and Theoretical Computer Science fellowship

The analysis of approximation algorithms for these problems started with Johnson who showed that the greedy algorithm colors k-colorable graphs with $\mathcal{O}(n/\log_k n)$ colors, obtaining a performance guarantee of $\mathcal{O}(n/\log n)$. Several years later, Wigderson [20] introduced an elegant algorithm that colors k-colorable graphs with $\mathcal{O}(kn^{1/(k-1)})$ colors, which, when combined with Johnson's result, yields an $\mathcal{O}(n(\log\log n/\log n)^2)$ performance guarantee. Very recently, Berger and Rompel [3] gave an algorithm that improves on Johnson's idea to obtain an $\mathcal{O}(n/(\log_k n)^2)$ coloring. When combined with Wigderson's method, they obtain an $\mathcal{O}(n(\log\log n/\log n)^3)$ performance guarantee. Finally, Blum has improved the best ratio for small values of k, in particular for 3-coloring from the $\mathcal{O}(\sqrt{n})$ of Wigderson and the $\mathcal{O}(\sqrt{n}/\log n)$ of Berger and Rompel, to $n^{0.4+o(1)}$ [6] and later to $n^{0.375+o(1)}$ [5].

We shall present an efficient graph coloring algorithm that colors k-colorable graphs with $\mathcal{O}(n^{1/(k-1)}/k)$ colors when $k \leq 2\log n$, and $\mathcal{O}(\log n/\log\frac{k}{\log n})$ when $k \geq 2\log n$. The algorithm strictly improves on both Johnson's and Wigderson's method, and can be combined with the method of Berger and Rompel in place of Wigderson's method to improve the performance guarantee slightly (although not asymptotically).

Folklore (see [11, pp. 134] attributed to Gavril) tells us that any maximal matching approximates the minimum vertex cover by a factor of two. This was slightly improved independently by Bar-Yehuda and Even [2], and Monien and Speckenmeyer [17] to a factor of $2 - \Omega(\frac{\log\log n}{\log n})$, but no further improvements have yet been found.

Approximating the independent set problem has seen less success. No approximation algorithm yielding a non-trivial performance guarantee has been found in the literature. One of the main results of this paper is an algorithm that obtains an $\mathcal{O}(n/(\log n)^2)$ performance guarantee for the independent set problem on general graphs, as well as several results on graphs with a high independence number.

For all these problems the optimal approximation ratios are unknown, and the gaps between the upper and lower bounds are large. The vertex cover could possibly have a polynomial time approximation schema, i.e. being approximable within any fixed constant greater than one. The independent set problem might also have a polynomial time approximation schema, while otherwise it cannot be approximated within *any* fixed constant [11, pp. 146]. Recent results by Berman and Schnitger [4] give evidence that it may not be approximable within anything less than some fixed power of n. Finally, graph coloring cannot be approximated within less than a factor of two (assuming $\mathcal{P} \neq \mathcal{NP}$) [11, pp. 144], and the results of Linial and Vazirani [15] also suggest that some fixed power of n may be the best approximation we can hope for.

We will present lower bounds of a different kind, namely for a fixed class of algorithms, similar to the work of Chvátal [7]. We show how the best known approximation algorithms for these problems revolve around the concept of *excluding subgraphs* and how no algorithm within that framework can do significantly better than the algorithms presented here. The techniques used have a striking connection with graph Ramsey theory and the Ramsey-theoretic results may be of independent interest.

Some graph notation: For an undirected graph $G = (V, E)$, $|G|$ is the *order* of G or the number of vertices, $\alpha(G)$ is the *independence number* of the graph or the size of the largest independent set, $i(G)$ is the *independence ratio* or the independence number divided by the order of the graph, $cl(G)$ is the *clique number*, and $\chi(G)$ is the *chromatic number* (the number of colors needed to vertex color G). For a vertex v, $N(v)$ refers to the subgraph induced by the neighbors of v and $\overline{N}(v)$ similarly the subgraph induced by vertices non-adjacent to v. Unless otherwise stated, G is the input graph, n is the order of G, and H is a fixed forbidden (not necessarily induced) subgraph.

2 The Ramsey Algorithm

When one measure of a graph is fixed or restrained, others often become restricted as well. In particular, we are interested in constructive lower bounds on the independence number in terms of the clique number.

It was first proven by the English mathematician Frank Ramsey that for any integers s and t there is an integer n for which all graphs of order n contain either a clique of size s or an independent set of size t. Denote the minimal such value n by the function $R(s,t)$. A good upper bound for $R(s,t)$ was found by Erdös and Szekeres [9] in 1935. We shall see how their elegant proof will lead us to an efficient algorithm.

Theorem 1

$$R(s,t) \leq \binom{s+t-2}{t-1}$$

Proof. Trivially, $R(s,1) = R(1,s) = 1$. Also, since a graph with no edge must be an independent set, and, vice versa, a graph without non-adjacent nodes must be a clique, $R(s,2) = R(2,s) = s$.

Now consider a graph G of order $\binom{(s-1)+(t-1)}{t-1}$. Pick any vertex v in the graph and look at the subgraphs $N(v)$ and $\overline{N}(v)$. Since $\binom{x-1}{y} + \binom{x-1}{y-1} = \binom{x}{y}$, either the order of $N(v)$ is at least $\binom{(s-2)+(t-1)}{t-1}$ or the order of $\overline{N}(v)$ is at least $\binom{(s-1)+(t-2)}{t-2}$.

Hence by the inductive hypothesis, either $N(v)$ contains an $(s-1)$-clique or t-independent set, or $\overline{N}(v)$ contains an s-clique or $(t-1)$-independent set. In any case we are done because if $N(v)$ has a clique of size $s-1$ then adding v produces a clique of size s, and similarly if $\overline{N}(v)$ has an independent set of size $t-1$ then adding v will give us one of size t. ∎

For convenience, let us define $r(s,t)$ to be the above upper bound $\binom{s+t-2}{t-1}$ for $R(s,t)$ for all positive integers, and 1 if either s or t are non-positive. Also let $t_k(n)$ be the minimal t for which $r(k,t) \geq n$. We approximate $t_k(n)$ by $kn^{1/(k-1)}$ for $k \leq 2\log n$, and $\log n / \log \frac{k}{\log n}$ for $k \geq 2\log n$.

To produce an algorithm to find either the independent set or the clique guaranteed, consider a graph of order at least $r(s,t)$. Pick any vertex and look at the subgraphs induced by its neighbors and non-neighbors. At least one of those will satisfy our order constraints. If the order of the neighborhood subgraph is at least $r(s-1,t)$ we add the vertex to our clique, otherwise the order of the non-neighborhood subgraph is at least $r(s,t-1)$, in which case we add the vertex to the independent set. In either case, we restrict our attention to the subgraph chosen and continue this process until the graph has been exhausted. The last vertex, however, can be placed in both the clique and the independent set.

```
Ramsey (G, s)
begin
  I ← ∅; C ← ∅;
  while |G| > 1 do
    choose some v ∈ V(G)
    t ← t_s(|G|)
    if |N(v)| ≥ r(s − 1, t)
      then C ← C ∪ {v}; G ← N(v); s ← s − 1
      else I ← I ∪ {v}; G ← N̄(v)
  od
  C ← C ∪ V(G); I ← I ∪ V(G)
  output (C, I)
end
```

Algorithm 2.1: The Ramsey Algorithm

Algorithm 2.1, Ramsey, will find both an independent set I and a clique C, obeying the inequality $r(|C|,|I|) \geq n$. It follows that $|I||C| \geq c(\log n)^2$, for some constant c. It can be shown to hold for $c = \frac{1}{4}$.

When carefully implemented, algorithm 2.1 involves $\mathcal{O}(n + \sum_{v \in I \cup C} d(v)) \leq \mathcal{O}(n+m)$ work. When selecting a pivot node v, search sequentially starting from the last pivot node. When restricting the graph, mark the neighbors with some new "color" if the non-adjacent nodes are to be deleted, and otherwise mark the neighbors as deleted.

Clique removal

Suppose we are trying to find a large independent set. The Ramsey algorithm provides us with the means of obtaining either a sizable clique or a sizable independent set — the problem is that we have no direct control over which one we will get. One way of influencing the result is by eliminating choice: if the input graph contains no clique of size k, the independent set returned must be of size at least $t_k(n)$.

The naive bound of $t_{cl(G)+1}(n)$ is poor, but if we can remove all cliques of size k while retaining a good portion of the graph, the above bound holds on the derived graph. A key observation is that a clique and an independent set can share no more than a single vertex. If the independence number of the graph is at least $(1/k + \epsilon)n$ for some constant $\epsilon > 0$, then at least a fraction $\epsilon/(1-\frac{1}{k})$ of the vertices remain.

Algorithm 2.2 will find an independent set of size $t_k(N)$ for graphs with independence ratio

```
CliqueRemoval (G, k)
begin
  (C, I) ← Ramsey (G,k)
  while |C| ≥ k do
    G ← G − C
    (C, I) ← Ramsey (G,k)
  od
  output I
end
```

Algorithm 2.2: Algorithm for finding independent set in graphs with high independence ratio

strictly more than $1/k + \epsilon$, where $N \geq \epsilon n$ is the number of vertices remaining after the last clique has been removed. The algorithm repeats calling Ramsey and removing cliques, until a sizable independent set is found. Since no more than n/k cliques of size k can be removed from the graph, the complexity of the algorithm is $\mathcal{O}(nm/k)$.

It may appear that because ϵ can be arbitrarily small the approximation must be weak. We can however always make do with removing only $(k+1)$-cliques, in which case $\epsilon \geq (\frac{1}{k+1} - \frac{1}{k})$ and $N \geq (\frac{1}{k+1} - \frac{1}{k})\frac{1}{1-\frac{1}{k}}n \geq \frac{n}{k^2}$. Since the approximation is $N^{1/(k-1)} \geq (\frac{n}{k^2})^{1/(k-1)} = 2^{(\log n - 2\log k)/(k-1)}$, the factor of k^2 does not affect the asymptotic bound.

A technique by Ajtai, Komlós, and Szemerédi [1] can be used in addition to our method to improve the performance by some logarithmic factor. As treated by Shearer [18], the technique can be thought of as a *randomized* greedy algorithm, which we can be make deterministic. For a fixed k, it will find an independent set in k-clique-free graphs of size $\Omega(n^{1/(k-1)}(\log n)^{(k-2)/(k-1)})$ in polynomial time.

In general we don't know the k for which CliqueRemoval will yield the best possible approximation guarantee. Fortunately, a conservative estimate on the inverse independence ratio of the graph suffices. We can guess a value for k, and if CliqueRemoval(G,k) returns too small an approximation (less than $(\frac{n}{k^2})^{1/k}$), then the estimate was too low. If we find a k' such that CliqueRemoval fails on $k'-1$ but succeeds on k', then the estimate is conservative. To find such a k' we employ a form of binary search [20], thereby adding a $\log n$ factor to the time complexity. The performance guarantee of this algorithm is no worse than $\min_k \frac{n/k}{t_{k+1}(n)} = \mathcal{O}(n/(\log n)^2)$ with minima at $k \approx \log n$.

Graph coloring

Any approximation algorithm A_{IS} for independent set can be turned into an approximation algorithm for graph coloring using the ColorByExcavating heuristic [13]. This method of coloring is used (directly or indirectly) by all known approximate coloring methods.

Since a k-colorable graph contains no $(k+1)$-cliques, we can use Ramsey($G, k+1$) as the independent set approximation algorithm. When the chromatic number is not known, we simply use the size of the largest clique found so far plus one as a parameter. The complexity of the

```
ColorByExcavating (G)
begin
  p ← 0
  while G ≠ ∅ do
    I ← A_IS(G)
    Color I with color p
    G ← G − I
    p ← p + 1
  od
end
```

Algorithm 2.3: Coloring graphs by "excavating" independent sets

algorithm is clearly bounded by the the number of colors found times the number of edges and vertices in the graph.

The number of colors needed will be the inverse of the independent set approximation, namely $n/t_{k+1}(n) \approx n^{(k-1)/k}/k$, times some overhead factor. Johnson [14] showed that when using an optimal independent set algorithm the overhead is $\theta(\log n)$. In our case, however, the excavation is so piecemeal that the overhead is only a small constant. In fact, $\lim_{n\to\infty} \textsc{Colors} = \frac{n}{|I|}$.

The performance guarantee is $\min_k \frac{n^{(k-1)/k}/k}{k} = \mathcal{O}(n/(\log n)^2)$. This can also be obtained using the previously noted fact that $|I| \geq \frac{(\log n)^2}{4|C|}$, conservatively obtaining $\textsc{Colors} \leq \frac{5n}{(\log n)^2}|C|$. This was, in fact, noticed by Erdös in a rather obscure paper [8] from 1967.

Now consider the product of the performance guarantees for the clique problem R_{cl} and the coloring problem R_χ:

$$R_{cl} \cdot R_\chi \leq \frac{cl(G)}{|C|}\frac{\textsc{Colors}}{\chi(G)} \leq \frac{5n}{(\log n)^2}\frac{cl(G)}{\chi(G)}$$

Since $cl(G)$ is never greater than $\chi(G)$, this bound immediately yields the individual bounds. Clearly, if one approximation is as far off as the individual bound indicates, then the other one must be within a constant factor of optimal. Also, for classes of instances for which the measures are apart, the performance guarantee is even stronger. In particular, random graphs almost always have a clique number asymptotically $2\log n$ and chromatic number $n/(2\log n)$, and for graphs with these parameters $R_{cl} \cdot R_\chi = \mathcal{O}(1)$.

3 Subgraph-Excluding Algorithms

The algorithms presented in the previous section all fall into the category of subgraph-excluding (alias Ramsey-type) algorithms.

Definition 1 *A_H is a* Ramsey-type *algorithm if, given arbitrary graph G, it is of the form:*

1. *Ensure G contains no copy of the subgraph H, and*
2. *Find an independent set in G, using only the property that G contains no copy of H.*

There are a few ways in which such an algorithm can exclude a subgraph H:

Remove: All copies of the subgraph, or parts of it, can be pulled out of the graph sequentially. A necessary and sufficient precondition for the removal process to retain at least a constant fraction of the vertices is that $i(H) < i(G) + \epsilon$, for some constant $\epsilon > 0$.

Forbid: The exclusion of the subgraph can be built into the statement of the problem. This applies particularly to the graph coloring problem. For instance, the clique on $k+1$ vertices is forbidden in k-colorable graphs.

Merge: In certain cases, vertices can be fused together, causing a certain type of subgraphs to become non-existent.

The issue that remains is finding graphs H that force graphs free of H to contain large independent sets, as well as coming up with algorithms to actually finding those independent sets in H-free graphs. The previous section described algorithms that use cliques. Other subgraphs discussed in this section include odd cycles, wheels, and color-critical subgraphs. The following section will then illustrate that these subgraphs are in some sense the best of their kind.

Wheels

A *wheel*, denoted by $W_{p,m}$, is a graph that consists of an odd cycle of $m \geq 3$ nodes, and $p \geq 0$ *spokes* which are nodes that connect to all other nodes in the graph. A wheel with p spokes is referred to as a p-wheel. The clique number of an p-wheel is $p+2$ (except when $m = 3$), whereas the chromatic number is $p+3$.

Define $R(W_p, K_t)$ to be the minimal n such that all graphs of order n contain some p-wheel or an independent set of size t. Note that if a graph does not contain a p-wheel, then its neighborhood graph cannot contain a $(p-1)$-wheel nor can its non-neighborhood graph contain a p wheel. Hence we obtain the same characterization as for the regular Ramsey numbers: $R(W_p, K_t) \leq R(W_{p-1}, K_t) + R(W_p, K_{t-1})$.

Observe that $R(W_0, K_t) = 2t - 1$ and $R(W_p, K_2) = p + 3$. An inductive argument shows that $R(W_p, K_t) \leq 2r(p+2, t)$, only a factor of two from the upper bound of the regular Ramsey function. We use these properties to design a variation of the Ramsey algorithm with bipartite graphs as a base case.

Given a graph with no $(k-2)$-wheels, WheelFreeRamsey finds an independent set of size at least $\Omega(kn^{1/(k-1)})$. Applying algorithm 2.3, we can color a graph without $(k-2)$-wheels using $\mathcal{O}(n^{(k-2)/(k-1)}/k)$ colors.

Algorithm 3.1 is strongly related to Wigderson's coloring algorithm. By considering the whole uncolored portion of the graph in each iteration, instead of fully coloring the pivot nodes' neighborhoods before coloring their non-neighbors, WheelFreeRamsey improves the approximation by a factor of k. Also, by focusing alternately on neighborhoods and non-neighborhoods, another factor of k is gained. Wigderson's method, however, has the advantage of $\mathcal{O}(\chi(G)\ (n+m))$ time complexity, compared to $\mathcal{O}(\textsc{Colors}\ (n+m))$.

```
WheelFreeRamsey (G, s)
begin
  I ← ∅; C ← ∅;
  while s > 2 and G ≠ ∅ do
    choose some v ∈ V(G)
    t ← t_s(n/2)
    if |N(v)| ≥ 2 r(s − 1, t)
      then C ← C ∪ v; G ← N(v); s ← s − 1
      else I ← I ∪ v; G ← N̄(v)
  od
  if s ≤ 2  { G should be bipartite }
    then I ← I ∪ IndepSetInBipartiteGraph(G)
         C ← C ∪ {some edge in G, if it exists}
  output (C, I)
end
```

Algorithm 3.1: Ramsey Algorithm for Wheels

In comparison with the graph coloring algorithm deduced from the Ramsey algorithm for clique-free graphs, this algorithm improves the exponent from $(k-1)/k$ to $(k-2)/(k-1)$.

Short odd cycles

For graphs with independence ratio in the range of $(\frac{1}{3}+\epsilon, \frac{1}{2})$, the Ramsey algorithm obtains an independent set approximation of $\Omega(\sqrt{n})$ by removing triangles. Families of odd cycles as excluded subgraphs allow us to refine the approximations in this range.

The method starts by removing all odd cycles of size up to $2k+1$. Note that a cycle of length $2k+1$ has an independence ratio $\frac{k}{2k+1}$. So if $i(G) > \frac{k}{2k+1}$, we can remove these cycles and then apply algorithm 3.2.

```
OddCycleFreeApproximation (G, k)
 { Graph G contains no odd cycles of length 2k + 1 or shorter}
begin
  while G ≠ ∅ do
    choose any vertex v in V(G).
    V_i ← vertices of distance i from v.
    S_i ← V_i ∪ V_{i−2} ∪ ...
    Determine i such that |S_{i+1}| ≤ n^{1/(k+1)} |S_i|.
    I ← I ∪ S_i
    G ← G − S_i − S_{i+1}
  od
  return I
end
```

Algorithm 3.2: Algorithm for independent sets on graphs with no short odd cycles

Since each independent set S_i selected causes only $n^{1/(k+1)}$ times as many other nodes to be removed from the graph, the graph is not exhausted until an independent set of at least

$n^{k/(k+1)}$ has been collected. Assume there was no i satisfying $|S_{i+1}| \leq n^{1/(k+1)}|S_i|$. Then $|S_k| > n^{1/(k+1)}|S_{k-1}| > n^{2/(k+1)}|S_{k-2}| > \cdots > n^{k/(k+1)}|S_0| = n^{k/(k+1)}$, and the problem is solved.

Since each vertex and each edge are looked at only once, the algorithm runs in linear time. On the other hand, when applied to general graphs the algorithm must be run for many different values of k, in which case it may be useful to combine the cycle removal process (see [17]).

The technique of Ajtai, Komlós, and Szemerédi can also be applied here. When k is fixed, we can find an independent set of size $\Omega(n^{k/(k+1)}(\log n)^{1/(k+1)})$ in polynomial time for graphs with no odd cycles of length $2k+1$ or less.

Color-critical graphs

A. Blum [6, 5] recently introduced improved approximation algorithms for k-colorable graphs, where k is fixed, in particular improving approximate 3-coloring to $\tilde{O}(n^{3/8})$ colors. His complicated method can be summarized in the following three steps:

1. Destroy all copies of the subgraphs $K_4 - e$ and 1-2-3 graphs by collapsing certain pairs of nodes.
2. Classify vertices according to degree, producing a polynomial number of subgraphs, one of which has an independence ratio close to half.
3. Apply algorithm 3.2 on each of these subgraphs.

The graph $K_4 - e$ is the clique on 4 vertices with one edge removed. "1-2-3 graphs" is our terminology for the graphs with three specific parts: A, consisting of two disconnected nodes; B, an independent set of at least 3 nodes; and C, an odd cycle, where parts A and C are completely disconnected, A and B are completely connected, and the connections between B and C are such that each node in C is connected to some node in B. Since C requires three colors, B needs two, and thus the two nodes in A must have the same color under any legal 3-coloring of the subgraph, hence the name. Similarly, the two disjoint nodes in $K_4 - e$ must share the same color.

The first and the third steps are strictly Ramsey, whereas the second does use the size of the independent sets promised by the k-colorability property, hence, the algorithm appears to lack the "forgetfulness" property of Ramsey-style algorithms.

4 Limitation results

The main result of this section is that excluding subgraphs other than cliques and series of odd cycles does not help much in forcing a graph to contain a large independent set. This implies that no subgraph removal algorithms, even super-polynomial ones, can yield asymptotically better performance guarantees for the maximum independent set, graph coloring, and vertex cover problem than the algorithms given.

Extend the Ramsey function from cliques to arbitrary graphs. Let $R(H, K_t)$ denote the minimal n such that every graph on n vertices either contains a copy of the graph H or has an independent set of size t. Note that H does not need to be isomorphic to a *vertex induced* subgraph of G, only that all the edges of H be contained in such a subgraph. It immediately follows that $R(H, K_t) \geq R(H', K_t)$ whenever H' is an edge-subset of H. Obtaining an upper bound on $R(H, K_t)$ shows that not all H-free graphs contain all that large independent sets, showing a limitation on the power of excluding H.

A few definitions are in order. For a graph H, let $e(H)$ be the number of edges, and $\rho(H)$ denote the maximum of $e(H')/|H'|$ over all subgraphs H' of H. Extend these definitions to a collection $\mathcal{H}$ of graphs. Define $i(\mathcal{H})$ to be the maximum of $i(H)$ over all H in $\mathcal{H}$. Define $\rho(\mathcal{H})$ and $\chi(\mathcal{H})$ to be the minimum of $\rho(H)$ and $\chi(H)$, respectively, over all H in $\mathcal{H}$. Also, $R(\mathcal{H}, K_t)$ is the minimal n such that every graph on n vertices either contains a copy of some H in $\mathcal{H}$ or has an independent set of size t.

There are some well-known relations between these quantities. One relation is $\chi(H)i(H) \geq 1$, which holds because a coloring is just a partition into independent sets. Another relation is $\chi(H) \leq 2\rho(H) + 1$, which holds because H has a vertex of degree $2\rho(H)$ or less. Both relations generalize to a collection of graphs.

We will give the central theorem for a function slightly stronger than ρ. Define $\rho'(H) = \min \frac{e(H')-1}{|H'|-2}$ where H' is a subgraph of H on at least 3 vertices. Similarly extend it to a collection of graphs $\mathcal{H}$. The value of ρ' is always at least as large as ρ, and for small graphs the improvement makes a difference.

Theorem 2 $R(\mathcal{H}, K_t) = \Omega((\frac{t}{\log t})^{\rho'(\mathcal{H})})$

The omitted proof follows the probabilistic method using the Lovász local lemma (see [19]). We will first apply theorem 2 to obtain a limitation for forbidden subgraphs of a minimum chromatic value.

Recall that Blum's algorithm made use of subgraphs that contain two nodes that must be of the same color under any legal 3-coloring. A graph is *k-avoidable* iff it has a pair of vertices that get assigned the same color for every k-coloring of the graph. Note that this is vacuously true for non-k-colorable graphs. Alternatively, k-avoidable graphs can be characterized as being no more than one edge away from being $(k+1)$-chromatic. A collection $\mathcal{H}$ is k-avoidable iff every H in $\mathcal{H}$ is.

Corollary 1 *For every positive integer k, if $\mathcal{H}$ is k-avoidable, then* $R(\mathcal{H}, K_t) = \Omega((t/\log t)^{k/2})$.

Proof. If $H \in \mathcal{H}$ is k-avoidable, then $H + e$ is $(k+1)$-chromatic for some edge e. Hence $\rho(H+e) \geq \frac{k}{2}$. But $\rho'(H) \geq \min_{H' \in H+e} \frac{e(H')-2}{|H'|-2} \geq \rho(H+e)$, when $\rho(H+e) \geq 1$. This holds for all H in $\mathcal{H}$, hence the conclusion follows from theorem 2 ∎

This result implies that a Ramsey-type algorithm on a k-colorable graph that relies solely on the lack of some set of k-avoidable subgraphs, cannot guarantee finding an independent set of size more than $O(n^{2/k}\log n)$, and hence cannot guarantee a coloring with less than $\Omega(n^{1-2/k}/\log n)$ colors. As an example, no such algorithm can guarantee coloring a 3-colorable graph with less than $\Omega(n^{1/3}/\log n)$ colors.

We can make a stronger statement regarding the 3-coloring problem.

Theorem 3 *If $\mathcal{H}$ is 3-avoidable, then $\rho'(\mathcal{H}) \geq \frac{3}{2} + \frac{1}{26}$.*

Proof. Let H be a 3-avoidable graph in $\mathcal{H}$, and $H+e$ be 4-chromatic. A *4-critical* graph is a 4-chromatic graph with the property that removing any node will make it 3-colorable. Gallai [10] showed that 4-critical graphs, with the exception of K_4, have an edge-to-vertex ratio of at least $3/2 + 1/26$. If $H+e$ contains a K_4, then $\rho'(H) \geq \rho'(K_4 - e) = \frac{5-1}{4-2} = 2$. Otherwise, $\rho(H+e) \geq \frac{e(H^*)}{|H^*|} \geq \frac{3}{2} + \frac{1}{26}$, by Gallai's result. In either case, $\rho'(H) \geq \frac{3}{2} + \frac{1}{26}$ for any H in $\mathcal{H}$. ∎

As a result, Ramsey-type algorithms require at least $\Omega(n^{1-1/(\frac{3}{2}+\frac{1}{26})}/\log n) \approx \Omega(n^{.35})$ colors on 3-colorable graphs. Notice that Blum's technique also breaks down in the region of $n^{1/3}$ [5], even though it is not known to be of a subgraph excluding type.

Let us now derive a limitation for general graphs. It can be shown, in the spirit of the bounds on the diagonal Ramsey function $R(s,s)$, that if $\mathcal{H}$ is a t-avoidable collection then $R(\mathcal{H}, K_t) = 2^{\Omega(t)}$. Hence if all graphs of order n contain either a subgraph H in the t-chromatic collection $\mathcal{H}$ or an independent set of size t, then t must be no more than $O(\log n)$. Hence no Ramsey-type algorithm that relies solely on the lack of avoidable subgraphs can obtain a better than $\Omega(n/(\log n)^2)$ performance guarantee for graph coloring.

Our emphasis so far on graph coloring is for a good reason, namely because the lower bounds for graph coloring are also lower bounds for the independent set problem. Since $i(H) < 1/k$ implies that $\chi(H) \geq k+1$, corollary 1 holds as well for graphs with large independence ratio. Similarly, the limitation result on performance guarantees for the general coloring problem carries immediately over to the maximum independent set problem.

Limitation results for odd cycles

Our next goal is to show that our cycle-based algorithm is close to optimal for graphs with independence ratio near $\frac{1}{2}$. Our results will also imply that the algorithm of Monien and Speckenmeyer for approximately solving the vertex cover problem is essentially optimal among subgraph-removal algorithms.

We need the following structural result on graphs without short odd cycles.

Theorem 4 *For every positive integer k, if $\rho(H) \leq 1 + \frac{1}{4k+2}$, and H contains no odd cycles of length $2k-1$ or less, then $i(H) \geq \frac{k}{2k+1}$.*

Proof. By induction on the number of vertices in H. If there is a vertex v of degree 0 or 1, then remove v and its neighbor from the graph. By induction, the remaining graph has independence ratio at least $\frac{k}{2k+1}$. But adding v to the largest independent set of the remaining graph shows that H itself has independence ratio greater than $\frac{k}{2k+1}$.

Thus, assume that every vertex has degree 2 or more. Suppose there is a cycle passing through only vertices of degree exactly 2. Since H has no odd cycles of length $2k-1$ or less, the independence ratio of this cycle is at least $\frac{k}{2k+1}$. Then we could apply induction to the remainder of the graph and be finished.

Thus, assume there are no such cycles. Let H' be the subgraph induced by the vertices of degree exactly 2. The subgraph H' must be the disjoint union of paths, so $i(H') \geq \frac{1}{2}$. Since $\rho(H) \leq 1 + \frac{1}{4k+2}$, the subgraph H' contains at least a fraction $1 - \frac{1}{2k+1}$ of all the vertices. Therefore $i(H)$ is at least $\frac{1}{2}(1 - \frac{1}{2k+1}) = \frac{k}{2k+1}$, which completes the proof. ∎

Finally, we can prove the following limitation result.

Corollary 2 *For every positive integer k, if $i(\mathcal{H}) < \frac{k}{2k+1}$, then $R(\mathcal{H}, K_t) = \Omega((t/\log t)^{1+1/(4k+2)})$.*

Proof. By theorem 4 above, every H in $\mathcal{H}$ either has an odd cycle of length $2k-1$ or less, or satisfies $\rho(H) \geq 1 + 1/(4k+2)$. The first case implies that $\rho'(H) \geq (2k-2)/(2k-3) = 1 + 1/(2k-3)$, thus in either case $\rho'(H) \geq 1 + 1/(4k+2)$. ∎

This result implies that for a graph with independence ratio $\frac{k}{2k+1}$, no subgraph-removal algorithm can guarantee an independent set larger than $\mathcal{O}(n^{1-1/(4k+3)})$. Recall that our cycle-based algorithm will find an independent set of size $\Omega(n^{1-1/k})$ and thus the cycle-based algorithm is close to optimal. The above result also implies that for approximately solving the vertex cover problem, no subgraph-removal algorithm can achieve a performance guarantee better than $2 - \theta(\log\log n/\log n)$, the performance guarantee obtained by the algorithm of Monien and Speckenmeyer.

5 Discussion

The central open question is what the best possible approximation guarantees for the independent set and graph coloring problems are. The results of Berger and Rompel, and Blum show that there do exist more intelligent methods than mere subgraph exclusion. Their results, however, rely on special properties of k-colorable graphs, thus it is still not clear if better methodologies exist for the independent set and vertex cover problems.

One direction may be in approximating the number $\alpha(G)$ (or $cl(G)$) instead of actually constructing an independent set (clique). One such candidate is the ϑ function introduced in a seminal paper by Lovász [16], which has been shown to be polynomial time computable [12]. Its value always lies between the clique and the chromatic number of the graph, but it is open as to how far it can stray from each measure. Is it perhaps the case that ϑ approximates both the clique and the chromatic number within a factor of $\sqrt{n}$?

References

[1] M. Ajtai, J. Komlós, and E. Szemerédi. A note on Ramsey numbers. *J. Combin. Theory A*, 29:354–360, 1980.

[2] R. Bar-Yehuda and S. Even. A $2 - \frac{\log\log n}{2\log n}$ performance ratio for the weighted vertex cover problem. Technical Report #260, Technion, Haifa, January 1983.

[3] B. Berger and J. Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 1990. to appear.

[4] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. In *Proc. Symp. Theoret. Aspects of Comp. Sci.*, pages 256–268. Springer-Verlag Lecture Notes in Comp. Sci. # 349, 1989.

[5] A. Blum. Some tools for approximate 3-coloring. Unpublished manuscript, 1989.

[6] A. Blum. An $\tilde{O}(n^{.4})$ approximation algorithm for 3-coloring. In *Proc. ACM Symp. Theory of Comp.*, pages 535–542, 1989.

[7] V. Chvátal. Determining the stability number of a graph. *SIAM J. Comput.*, 6(4), Dec. 1977.

[8] P. Erdős. Some remarks on chromatic graphs. *Colloq. Math.*, 16:253–256, 1967.

[9] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

[10] T. Gallai. Kritische graphen I. *Publ. Math. Inst. Hungar. Acad. Sci.*, 8:165–192, 1963. (See Bollóbas, B. *Extremal Graph Theory*. Academic Press, 1978).

[11] M. R. Garey and D. S. Johnson. *Computers and Intractibility: A Guide to the Theory of NP-completeness.* Freeman, 1979.

[12] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1988.

[13] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9:256–278, 1974.

[14] D. S. Johnson. Worst case behaviour of graph coloring algorithms. In *Proc. 5th Southeastern Conf. on Combinatorics, Graph Theory, and Computing*, pages 513–527. Utilitas Mathematicum, 1974.

[15] N. Linial and V. Vazirani. Graph products and chromatic numbers. In *Proc. IEEE Found. of Comp. Sci.*, pages 124–128, 1989.

[16] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans Info. Theory*, IT-25(1):1–7, Jan. 1979.

[17] B. Monien and E. Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Inf.*, 22:115–123, 1985.

[18] J. B. Shearer. A note on the independence number of triangle-free graphs. *Preprint*, 1982. (See Bollóbas, B. *Random Graphs*. Academic Press, 1985).

[19] J. Spencer. *Ten Lectures on the Probabilistic Method*, volume 52. SIAM, 1987.

[20] A. Wigderson. Improving the performance guarantee for approximate graph coloring. *J. ACM*, 30(4):729–735, 1983.