Small Maximal Independent Sets and Faster Exact Graph Coloring

David Eppstein

Univ. of California, Irvine
Dept. of Information and Computer Science

The Exact Graph Coloring Problem:

Given an undirected graph G

Determine the minimum number of colors
needed to color the vertices of G
so that no two adjacent vertices have the same color

We want worst-case analysis

No approximations
No unproven heuristics

Isn't it impossible to solve graph coloring exactly?

It seems to require exponential time [Garey and Johnson GT4]
but that's very different from impossible

So why study it?

With fast computers we can do exponential-time
computations of moderate and increasing size

Algorithmic improvements are even more important
than in polynomial-time arena

Graph coloring is useful
e.g. register allocation, parallel scheduling

Approximate coloring algorithms have poor approximation ratios

Interesting gap between theory and practice
worst-case bounds and empirical results differ in base of exponent

Register Allocation Application

Problem: compile high-level code to machine instructions
Need to associate code variables to machine registers

Even if code has few explicltly named variables,
compilers can add more as part of optimization

Two variables can share a register if not active at the same time

Solution:

Draw a graph, vertices = variables, edges = simultaneous activity
Color with k colors, k = number of machine registers

Fast enough exact algorithm might be usable at high levels of optimization

Previous work on exact coloring

Lawler, 1976: 3-coloring $O(1.4423^n)$
For each maximal independent set, test if complement bipartite

k-coloring (unbounded *k*) $O(2.4423^n)$
Dynamic programming

Schiermeyer, 1994: 3-coloring $O(1.415^n)$
Transform graph to increase degree until degree $= n - 1$

Beigel & Eppstein, 1995: 3-coloring $O(1.3446^n)$
Reduce to more general constraint satisfaction problem
Complicated case analysis to find good local reductions

Schöning, 1999: General constraint satisfaction algorithm
Random walk in space of value assignments
No improvement for coloring

Eppstein, 2001: 3-coloring $O(1.3289^n)$, 4-coloring $O(1.8072^n)$
More case analysis, simple randomized restriction

This paper: *k*-coloring (unbounded *k*) $O(2.4150^n)$

Lawler's algorithm

Dynamic programming:

For each subgraph induced by a subset of vertices
compute its chromatic number from previously computed information

```
for S in subsets of vertices of G:
    ncolors[S] = n
    for I in maximal independent subsets of S:
        ncolors[S] = min(ncolors[S],
                         ncolors[S-I] + 1)
```

Outer loop needs to be ordered from smaller to larger subsets
so `ncolors[S-I]` already computed when needed

Lawler's algorithm analysis

Key facts: n-vertex graph has $O(3^{n/3})$ maximal independent sets [Moon & Moser, 1965]
MIS's can be listed in time $O(3^{n/3})$ [Johnson, Yannakakis, & Papadimitriou 1988]

Worst case example: n/3 disconnected triangles

Time: sum $3^{|S|/3}$ = sum $\binom{n}{i} 3^{i/3} = O((1 + 3^{1/3})^n)$

Bottleneck: listing MIS's of every subset of vertices of G

Space: one number per subset, $O(2^n)$

First refinement:

When the loop visits subset `S`,
instead of computing its chromatic number from its subsets,
use its chromatic number to update its supersets

```
for S in subsets of vertices of G:
    for I in maximal independent subsets of G-S:
        ncolors[S+I] = min(ncolors[S+I],
                           ncolors[S] + 1)
```

Why is it safe to only consider maximal independent subsets of G-S?

We need only correctly compute `ncolors[S]` when `S` is maximal *k*-chromatic
but if `I` is not maximal, neither is `S+I`

Analysis

Same as original Lawler algorithm

Second refinement:

Only look at small maximal independent subsets

```
for S in subsets of vertices of G:

    limit = |S| / ncolors[S]

    for I in maximal independent subsets of G-S
    such that |I| ≤ limit:
        ncolors[S+I] = min(ncolors[S+I],
                           ncolors[S] + 1)
```

Why is it safe to ignore large maximal independent subsets of G-S?

If `X` is maximal *k*-chromatic, let `I` be its smallest color class
Then `S=X-I` is maximal $(k-1)$-chromatic
and `I` will be below the limit for `S`

So, the outer loop iteration for `S` will correctly set `ncolors[X] = k`

Small Maximal Independent Sets

To continue analysis, we need facts and algorithms
analogous to Moon-Moser and Johnson-Yannakakis-Papadimitriou

Theorem:

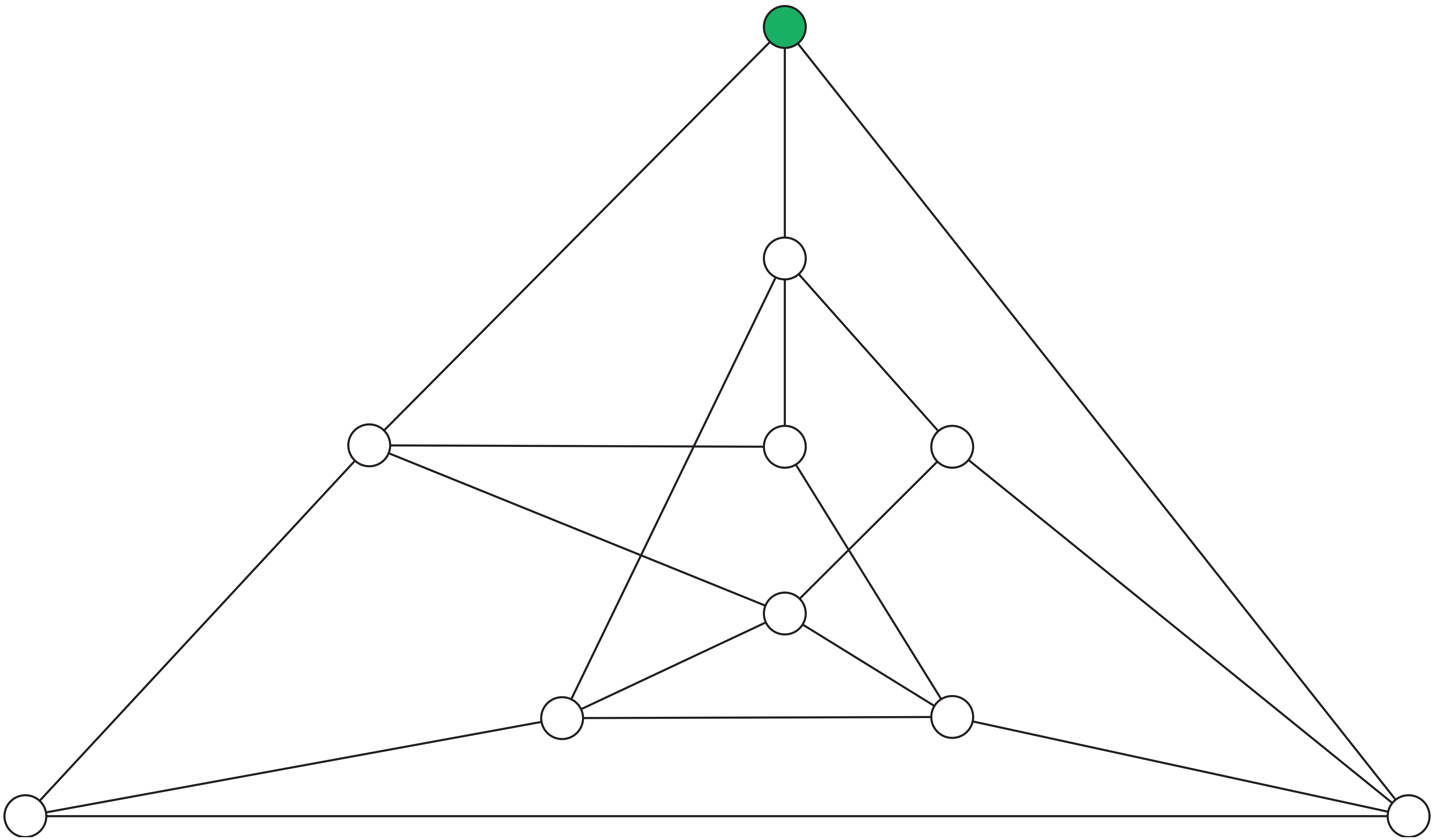
For any n-vertex graph G and limit L
there are at most $3^{4L-n}4^{n-3L}$ maximal independent sets I with $|I| \leq L$

All such sets can be listed in time $O(3^{4L-n}4^{n-3L})$

These bounds are tight when $n/4 \leq L \leq n/3$:
G = disjoint union of $4L - n$ triangles and $n - 3L$ K_4's

Proof idea:

Show set of MIS's = union of MIS sets of multiple smaller graphs
Combine smaller graph MIS counts to form recurrence

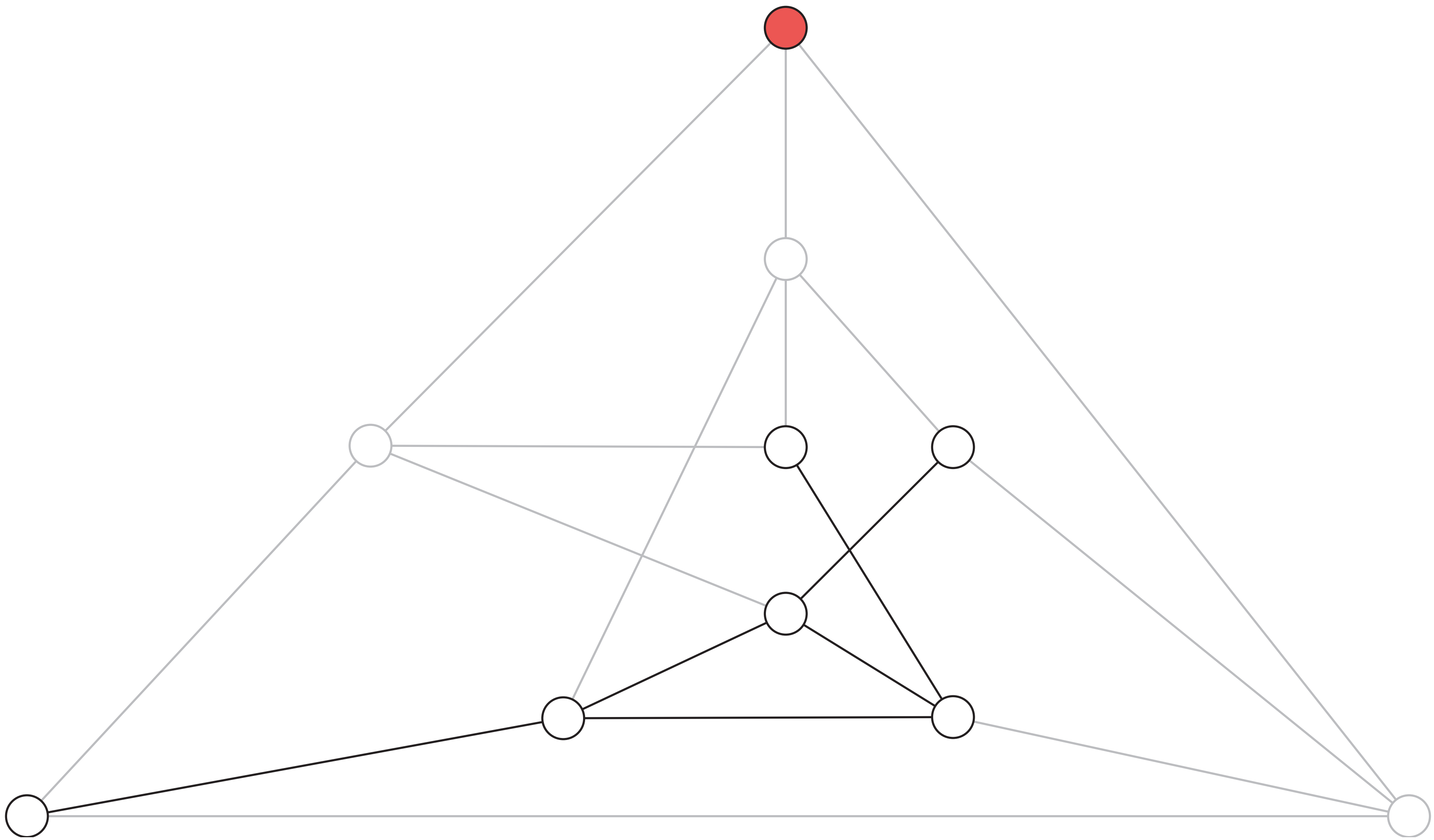


First case: vertex with degree ≥ three

If given vertex is part of MIS

Then rest of MIS is also an MIS of G − *neighbors(v)*

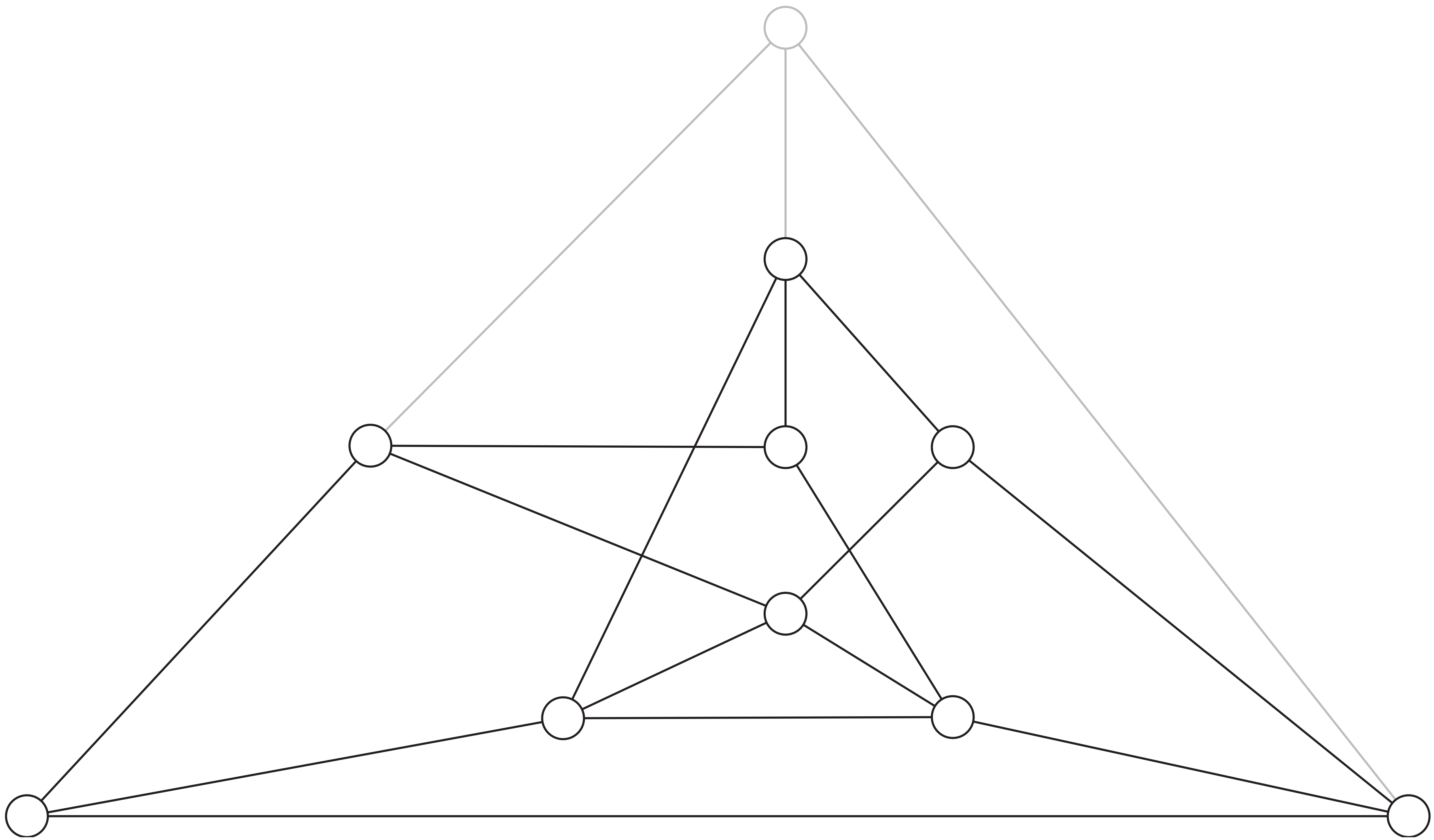
Subgraph has four fewer vertices, smaller bound on remaining MIS size



If given vertex is not part of MIS

Then it is also an MIS of G − v

Subgraph has one fewer vertex, same bound on MIS size



Not all MIS's of subgraph are MIS's of original graph
but overcounting doesn't hurt

Details of Case Analysis

If G contains v of degree ≥ 3:
split into MIS's containing v or not containing v
$\#MIS(G) \le \#MIS(n - 4, L - 1) + \#MIS(n - 1, L)$

If G contains v of degree = 1:
Every MIS contains either v or its neighbor
$\#MIS(G) \le 2\ \#MIS(n - 2, L - 1)$

If G contains v of degree = 0:
Every MIS contains v, $\#MIS(G) \le \#MIS(n - 1, L - 1)$

If G contains chain u-v-w-x all of degree = 2:
Each MIS contains u, contains v, or excludes u and contains w
$\#MIS(G) \le 2\ \#MIS(n - 3, L - 1) + \#MIS(n - 4, L - 1)$

Remaining case, G consists of disjoint triangles
has $3^{n/3}$ MIS's, all of size $n/3$

Prove by induction that each expression is at most $3^{4L - n}\, 4^{n - 3L}$

Easily turned into efficient recursive algorithm

Analysis of second refinement to coloring algorithm

Still not any better than Lawler

Problem:
If S has chromatic number at most 2 then limit = $|S|/2$
and small MIS bound only an improvement for $|S| \geq 2n/5$

Doesn't cover the the worst case sizes of sets $|S|$ for the algorithm

Final refinement:

Handle low-chromatic-number subsets specially

```
for S in subsets of vertices of G:

    if S is 3-colorable:
        compute ncolors[S] using 3-coloring alg

    if ncolors[S] ≥ 3:
        limit = |S| / ncolors[S]

        for I in maximal independent subsets of G-S
        such that |I| ≤ limit:
            ncolors[S+I] = min(ncolors[S+I],
                               ncolors[S] + 1)
```

Analysis

Each set `S` has `limit ≤ |S|/3`

So time to find small maximal independent sets of `G-S` is found by plugging `|G-S|` and `|S|/3` into small MIS formula:

$$\text{time to process } S = O(3^{4|S|/3 - |G-S|} 4^{|G-S| - 3|S|/3})$$

Sum over all `S` simplifies to $O((4/3 + 3^{4/3}/4)^n)$, approximately 2.415^n

Additional 3-coloring test per subset only adds $O(2.3289^n)$

Conclusions

Improvement to Lawler's exact coloring algorithm

Reduced base of exponent means
can solve problems larger by some constant factor

New algorithm still simple enough to possibly be useful
(BE95 gives simple $2^{n/2}$ alg for 3-coloring step, good enough here)

Space $O(2^n)$ may be a bigger problem than time

Ideas for possible further improvement

Reduce 4-coloring time below $2^{n/2}$
would allow algorithm to assume ncolors[S] ≥ 4

Can worst case number of small MIS's happen for many inner loop iterations?

Generalize small MIS bound to be tight for $L \leq n/4$
but doesn't affect worst case of current alg without further refinement