BRICS

Basic Research in Computer Science

Chromatic Number in Time $O(2.4023^n)$ Using Maximal Independent Sets

Jesper Makholm Byskov

BRICS Report Series **RS-02-45**

ISSN 0909-0878 **December 2002**

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/02/45/`

Chromatic Number in Time $O(2.4023^n)$ Using Maximal Independent Sets

Jesper Makholm Byskov
BRICS*
Department of Computer Science
University of Aarhus, Denmark
jespermn@brics.dk

December, 2002

Abstract

In this paper we improve an algorithm by Eppstein (2001) for finding the chromatic number of a graph. We modify the algorithm slightly, and by using a bound on the number of maximal independent sets of size k from our recent paper (2003), we prove that the running time is $O(2.4023^n)$. Eppstein's algorithm runs in time $O(2.4150^n)$. The space usage for both algorithms is $O(2^n)$.

1 Introduction

1.1 Lawler's algorithm

Lawler [Law76] was the first to give a non-trivial algorithm for finding the chromatic number of a graph. He notes that in a colouring of a graph one of the colour classes can be assumed to be a maximal independent set (a MIS). So we can find the chromatic number $\chi(G)$ of a graph G by the following recursion:

$$\chi(G[S]) = \begin{cases} 1 + \min\{\chi(G[S \setminus I]) \mid I \in I(G[S])\} & \text{if } S \neq \emptyset, \\ 0 & \text{if } S = \emptyset, \end{cases}$$

*Basic Research in Computer Science (www.brics.dk), funded by the Danish National Research Foundation.

Algorithm 1 Finding the chromatic number (Lawler)

let X be an array indexed from 0 to $2^n - 1$
for $S = 0$ to $2^n - 1$ **do**
  for all MISs I of $G[S]$ **do**
    $X[S] = \min(X[S], X[S \setminus I] + 1)$
return X[V]

where $G[S]$ denotes the vertex-induced subgraph of $S \subseteq V$ and $I(G)$ denotes the set of all maximal independent sets of G. Lawler does not explicitly give an algorithm. He merely notes that one has to process all subsets of S before S is processed. If we index all subsets from 0 to $2^n - 1$ such that the bit-representation of each index is a bit vector denoting for each vertex whether it is in the set or not, Algorithm 1 will find the chromatic number of G. Using the bound $3^{n/3}$ on the number of maximal independent sets of a graph ([MM65]) and the fact that they can be found within a polynomial factor of this bound (see e.g. [TIAS77]), this has running time

$$O\left(\sum_{S \subseteq V} |I(G[S])|\right) = O\left(\sum_{i=0}^{n} \binom{n}{i} 3^{i/3}\right) = O\big((1 + 3^{1/3})^n\big),$$

which is $O(2.4423^n)$. The algorithm uses space $O(2^n)$ to store X.

1.2 Eppstein's algorithm

Eppstein [Epp01b] improves Lawler's algorithm. Lawler's algorithm computes $\chi(G[S])$ by looking at the values for subsets of S. Eppstein's algorithm also computes a table $X[S]$ for all $S \subseteq V$ (indexed as above), but every time it reaches a set S, it updates X for all supersets of S for which the value $X[S]$ could potentially give better values. More precisely, if $G[S]$ is a maximal k-colourable subgraph of G, it has a maximal $(k-1)$-colourable subgraph $G[S']$ of size at least $|S'| \geq (k-1)/k \cdot |S|$, and $S \setminus S'$ is a maximal independent set of $G[V \setminus S']$. Thus, when the chromatic number of $G[S']$ is computed, only the values of $G[S' \cup I]$ for all maximal independent sets I of size at most $|S'|/\chi(G[S'])$ in $G[V \setminus S']$ are updated. The algorithm is shown as Algorithm 2.

The first part of the algorithm checks 1- and 2-colourability in polynomial time and 3-colourability using the 3-colouring algorithm of [Epp01a], with running time $O(1.3289^n)$, of all subgraphs of G. The second part

Algorithm 2 Finding the chromatic number (Eppstein)

let X be an array indexed from 0 to $2^n - 1$
for $S = 0$ to $2^n - 1$ **do**
  if $\chi(S) \leq 3$ **then**
    $X[S] = \chi(S)$
  else
    $X[S] = \infty$
for $S = 0$ to $2^n - 1$ **do**
  if $3 \leq X[S] < \infty$ **then**
    for all MISs I of $G[V \setminus S]$ of size at most $|S|/X[S]$ **do**
      $X[S \cup I] = \min(X[S \cup I], X[S] + 1)$
return X[V]

runs through X and for each subgraph $G[S]$ it finds all maximal independent sets I of $G[V \setminus S]$ of size at most $|S|/X[S]$ and updates the value of $X[S \cup I]$.

It is clear that for any set $S \subseteq V$, $\chi(G[S]) \leq X[S]$ during the execution of the algorithm, since $X[S]$ is only updated when the algorithm actually finds a colouring with $X[S]$ colours. For every k, all maximal k-colourable subgraphs $G[M]$ have $X[M] = k$, since they have so for $k \leq 3$ after the first half of the algorithm has run, and thus by induction also for larger k, by the argument above. Since G is a maximal $\chi(G)$-colourable subgraph of itself, the algorithm correctly computes the chromatic number of G.

The running time of the the first part of the algorithm is:

$$\sum_{S \subseteq V} O(1.3289^{|S|}) = O\left(\sum_{i=0}^{n} \binom{n}{i} 1.3289^i\right),$$

which is $O(2.3289^n)$. The second part might be executed for almost all S, but since $X[S] \geq 3$, only maximal independent sets of size at most $|S|/3$ in $G[V \setminus S]$ are considered. Using a lemma, that states that a graph can have at most $3^{4k-n}4^{n-3k}$ maximal independent sets of size at most k and that they can be found within a polynomial factor of this bound, Eppstein gets that the running time of the second part is at most

$$\sum_{S \subseteq V} O(3^{4(|S|/3)-(n-|S|)} 4^{(n-|S|)-3(|S|/3)}) = O\left(\left(\frac{4}{3}\right)^n \sum_{i=0}^{n} \binom{n}{i} \left(\frac{3^{7/3}}{4^2}\right)^i\right)$$
$$= O\left(\left(\frac{4}{3} + \frac{3^{4/3}}{4}\right)^n\right),$$

which equals $O(2.4151^n)$, so this is the running time of the algorithm. The space usage is $O(2^n)$.

2 Results

We show how to improve the algorithm of Eppstein to run in time $O(2.4023^n)$. The first part of the algorithms are the same, namely marking all 3-colourable subgraphs of the graph in a bit vector. Then our algorithm finds all maximal independent sets I of the graph and for each checks 3-colourability of all subgraphs S of $G[V \setminus I]$, by looking in the bit vector. If they are 3-colourable, $G[S \cup I]$ is 4-colourable. This will find all maximal 4-colourable subgraphs (and maybe some that are not maximal). Using the bound $\lfloor \frac{n}{k} \rfloor^{(\lfloor n/k \rfloor + 1)k - n} (\lfloor \frac{n}{k} \rfloor + 1)^{n - \lfloor n/k \rfloor k}$ on the number of maximal independent sets of size k from our paper [Bys03], and the fact that they can be found within a polynomial factor of this bound, we get that the running time for finding all maximal 4-colourable subgraphs is:

$$\sum_{k=1}^{n} O(|I_k(G)| \cdot 2^{n-k}) = O\left(\sum_{k=1}^{\lfloor \frac{n}{5} \rfloor} 5^{6k-n} 6^{n-5k} 2^{n-k} + \sum_{k=\lfloor \frac{n}{5} \rfloor + 1}^{n} 4^{5k-n} 5^{n-4k} 2^{n-k}\right)$$
$$= O(80^{n/5}),$$

which is $O(2.4023^n)$, and where $I_k(G)$ denotes the set of all maximal independent sets of G of size k. Then our algorithm performs the second step of Eppstein's algorithm, but it only needs to consider maximal independent sets of size at most $|S|/4$ and using our improved bound on the number of these, we get that the running time is:

$$\sum_{S \subseteq V} O(4^{5(|S|/4)-(n-|S|)} 5^{(n-|S|)-4(|S|/4)}) = O\left(\left(\frac{5}{4}\right)^n \sum_{i=0}^{n} \binom{n}{i} \left(\frac{4^{9/4}}{5^2}\right)^i\right)$$
$$= O\left(\left(\frac{5}{4} + \frac{4^{5/4}}{5}\right)^n\right),$$

Algorithm 3 Finding the chromatic number (new algorithm)

```
let X be an array indexed from 0 to 2^n − 1
for S = 0 to 2^n − 1 do
  if χ(S) ≤ 3 then
    X[S] = χ(S)
  else
    X[S] = ∞
for all MISs I in G do
  for all subsets S of V \ I do
    if X[S] = 3 then
      X[S ∪ I] = min(X[S ∪ I], 4)
for S = 0 to 2^n − 1 do
  if 4 ≤ X[S] < ∞ then
    for all MISs of G[V \ S] of size at most |S|/X[S] do
      X[S ∪ I] = min(X[S ∪ I], X[S] + 1)
return X[V]
```

which is $O(2.3814^n)$. The algorithm is shown as Algorithm 3. It still uses space $O(2^n)$.

Remark. In [Bys03] we also show how to mark all 3-colourable subgraphs of a graph in a bit vector in time $O(2.2680^n)$, by finding all maximal independent sets and for each find all maximal bipartite subgraphs of the remaining graph (which can be done by finding maximal independent sets twice). This improves the running time of the first step and also simplifies the algorithm as it does not use the 3-colouring algorithm of Eppstein, but instead uses maximal independent sets.

References

[Bys03] J. M. Byskov. Algorithms for k-colouring and finding maximal independent sets. In *Proc. 14th Symp. Discrete Algorithms*. ACM and SIAM, Jan. 2003. To appear.

[Epp01a] D. Eppstein. Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction. In *Proc. 12th Symp. Discrete Algorithms*, pages 329–337. ACM and SIAM, Jan. 2001. URL `http://arXiv.org/abs/cs.DS/0009006`.

[Epp01b] D. Eppstein. Small maximal independent sets and faster exact graph coloring. In F. Dehne, J.-R. Sack and R. Tamassia, editors, *Proc. 7th Worksh. Algorithms and Data Structures*, volume 2125 of *Lecture Notes in Computer Science*, pages 462–470. Springer-Verlag, Aug. 2001. URL `http://arXiv.org/abs/cs.DS/0011009`.

[Law76] E. L. Lawler. A note on the complexity of the chromatic number problem. *Inf. Process. Lett.*, 5(3):66–67, Aug. 1976.

[MM65] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.

[TIAS77] S. Tsukiyama, M. Ide, H. Ariyoshi and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM J. Comput.*, 6(3):505–517, Sept. 1977.

Recent BRICS Report Series Publications

RS-02-45 Jesper Makholm Byskov. ***Chromatic Number in Time*** $O(2.4023^n)$ ***Using Maximal Independent Sets*****. December 2002. 6 pp.**

RS-02-44 Zoltán Ésik and Zoltán L. Németh. ***Higher Dimensional Automata*****. November 2002. 32 pp. A preliminary version appears under the title** ***Automata on Series-Parallel Biposets*** **in Kuich, Rozenberg and Salomaa, editors,** ***5th International Conference, Developments in Language Theory*****, DLT '01 Revised Papers, LNCS 2295, 2001, pages 217–227. This report supersedes the earlier BRICS report RS-01-24.**

RS-02-43 Mikkel Christiansen and Emmanuel Fleury. ***Using IDDs for Packet Filtering*****. October 2002. 25 pp.**

RS-02-42 Luca Aceto, Jens A. Hansen, Anna Ingólfsdóttir, Jacob Johnsen, and John Knudsen. ***Checking Consistency of Pedigree Information is NP-complete (Preliminary Report)*****. October 2002. 16 pp.**

RS-02-41 Stephen L. Bloom and Zoltán Ésik. ***Axiomatizing Omega and Omega-op Powers of Words*****. October 2002. 16 pp.**

RS-02-40 Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir. ***A Note on an Expressiveness Hierarchy for Multi-exit Iteration*****. September 2002. 8 pp.**

RS-02-39 Stephen L. Bloom and Zoltán Ésik. ***Some Remarks on Regular Words*****. September 2002. 27 pp.**

RS-02-38 Daniele Varacca. ***The Powerdomain of Indexed Valuations*****. September 2002. 54 pp. Short version appears in Plotkin, editor,** ***Seventeenth Annual IEEE Symposium on Logic in Computer Science*****, LICS '02 Proceedings, 2002, pages 299–308.**

RS-02-37 Mads Sig Ager, Olivier Danvy, and Mayer Goldberg. ***A Symmetric Approach to Compilation and Decompilation*****. August 2002. To appear in Neil Jones's Festschrift.**

RS-02-36 Daniel Damian and Olivier Danvy. ***CPS Transformation of Flow Information, Part II: Administrative Reductions*****. August 2002. 9 pp. To appear in the** ***Journal of Functional Programming*****. This report supersedes the earlier BRICS report RS-01-40.**