Available online at www.sciencedirect.com
SCIENCE DIRECT®
Operations Research Letters 32 (2004) 547–556
Operations Research Letters
www.elsevier.com/locate/dsw

Enumerating maximal independent sets with applications to graph colouring☆

Jesper Makholm Byskov [1]

BRICS[2], *Department of Computer Science, University of Aarhus, IT-parken, Aabogade 34, DK-8200 Aarhus N, Denmark*

Received 4 February 2004; accepted 3 March 2004

Abstract

We give tight upper bounds on the number of maximal independent sets of size k (and at least k and at most k) in graphs with n vertices. As an application of the proof, we construct improved algorithms for graph colouring and computing the chromatic number of a graph.

Keywords: Maximal independent set; Graph colouring; Chromatic number; Graph algorithms; Extremal graphs

1. Introduction

Algorithms for finding maximal independent sets (MISs) and MISs of size k (k-MISs) in graphs can be used in algorithms for colouring graphs. We show upper bounds on the number of MISs and k-MISs in undirected graphs with n vertices and give algorithms for finding them all. We also show how to use these algorithms to develop faster algorithms for deciding k-colourability of a graph and finding the chromatic number of a graph.

1.1. Maximal independent sets

Finding MISs in a graph is a well-studied problem arising naturally in many fields. For a survey, see Bomze et al. [1, Chapter 5.1].

The maximum number of MISs in a graph with n vertices is the same as the maximum number of maximal cliques in a graph with n vertices by looking at the complementary graph with edges between all non-neighbours in the original graph. Miller and Muller [11] were the first to show the so-called Moon and Moser bound, which was independently discovered by Moon and Moser [12], namely that a graph can have at most $3^{n/3}$ MISs and that there are graphs having this many, the so-called Moon and Moser graphs (see Theorem 1). The proofs in both papers are combinatorial, but there exists quite a few algorithms for finding all MISs in a graph, for example the algorithm by Tsukiyama et al. [14] which runs in time within a polynomial factor of the number of MISs in the graph.

☆ This paper is an extended version of a short paper appearing in Proceedings of the 14th Symposium on Discrete Algorithms (SODA) 2003 [3]. It also contains an algorithm for finding the chromatic number of a graph from a technical report [2].

[1] Part of this paper was written while visiting University of California, Irvine.

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

E-mail address: jespermn@brics.dk (J.M. Byskov).

0167-6377/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2004.03.002

For MISs of size at most k, Eppstein [7] has shown that no graph contains more than $3^{4k-n}4^{n-3k}$ of these and he has also given an algorithm for finding them all in time within a polynomial factor of this bound. For $k = n/3$, Eppstein's bound is the same as the bound of Moon and Moser, but for $k \leqslant n/3$ it is smaller. Croitoru [5] showed that in a graph with no MIS larger than k, the number of MISs is at most $\lfloor n/k \rfloor^{(\lfloor n/k \rfloor+1)k-n}(\lfloor n/k \rfloor + 1)^{n-\lfloor n/k \rfloor k}$. For $n/4 \leqslant k \leqslant n/3$ this expression is the same as the bound of Eppstein.

In this paper, we extend the proof of Moon and Moser to show that the bound of Croitoru is an upper bound on the number of k-MISs for all k. Similar to Moon and Moser, we show that the bound is tight and that the extremal graphs are unique. For MISs of size at most k the bound of Croitoru holds for $k \leqslant n/3$ and for $k \geqslant n/3$ the bound of Moon and Moser is a tight upper bound. For MISs of size at least k the bound of Croitoru holds for $k \geqslant n/3$ and the bound of Moon and Moser is tight for $k \leqslant n/3$. We also extend the algorithm of Eppstein to find all k-MISs, MISs of size at most k or MISs of size at least k in time within a polynomial factor of our bounds on the number of these.

Finally, we give an algorithmic proof that the number of MISs in a triangle-free graph is at most $2^{n/2}$, a result proven earlier by Hujter and Tuza [8] using a combinatorial argument.

1.2. Colouring

The first algorithms with non-trivial running times for colouring are by Lawler [9] who gave an $O(1.4423^n)$-time algorithm for 3-colouring, an $O^*(2^n)$-time algorithm for 4-colouring (for a definition of O^*, see Section 2) and an $O(2.4423^n)$-time algorithm for finding the chromatic number using space $O(\log n \cdot 2^n)$. The last algorithm builds on previous work of Christofides [4]. All three algorithms are described in Section 4.

Schiermeyer [13] gave an algorithm for 3-colouring running in time $O(1.398^n)$. The main idea is that a pair of vertices must either have different colours in which case an edge between them can be added or have the same colour in which case they can be contracted, meaning that one of them is removed and edges are added between the other and all the neighbours of the removed vertex. Using that idea to branch on vertices with degree at least four and a complicated case analysis for the remaining graphs, Schiermeyer gets his running time. He also gives algorithms for 4-, 5- and 6-colouring, but they do not have the running times that he claims [10].

The current best algorithm for 3-colouring is by Eppstein [6] and runs in time $O(1.3289^n)$. The algorithm first transforms the graph and then solves a generalised version of 3-colouring for the transformed graph. Both parts require a rather involved case analysis. In the same paper, Eppstein gives randomised algorithms for solving generalisations of 4- and 5-colouring in time $O(1.8072^n)$ and $O(2.2590^n)$, respectively. These are the previous best algorithms for those problems.

The algorithm by Lawler for finding the chromatic number has also been improved by Eppstein [7] who gave an algorithm running in time $O(2.4151^n)$ still using space $O(\log n \cdot 2^n)$. His algorithm is also described in Section 4.

In this paper, we give new algorithms for 4-, 5- and 6-colouring using polynomial space and running in time $O(1.7504^n)$, $O(2.1592^n)$ and $O(2.3289^n)$, respectively. We also give algorithms for finding all maximal 3-colourable subgraphs and for 6-colouring both running in time $O(2.2680^n)$, but using space $O(2^n)$. For finding all maximal k-colourable subgraphs for $k \geqslant 4$, we give algorithms running in time $O(2.4023^n)$ and space $O(2^n)$. We use these algorithms in a new algorithm for finding the chromatic number in time $O(2.4023^n)$ using space $O(2^n)$.

2. Preliminaries

All graphs in this paper are simple, undirected graphs $G = (V, E)$ and we let $n = |V|$. The subgraph induced by a subset of the vertices $S \subseteq V$ is denoted $G[S]$ and we let $N(v) = \{w \in V \mid (v, w) \in E\}$ denote the set of neighbours of v and $\bar{N}(v) = \{v\} \cup N(v)$. The graph K_n is the complete graph on n vertices.

We let $I(G)$ denote the set of all MISs in G, $I^{\leqslant k}(G)$ the subset of those that have size at most k and $I_v^{\leqslant k}(G)$ the subset of those that contain v. Similarly, we define $I^{=k}(G)$, $I_v^{=k}(G)$, $I^{\geqslant k}(G)$ and $I_v^{\geqslant k}(G)$.

All running times in this paper are exponential and we ignore polynomial factors. We use the notation

$O^*(c^n)$ to denote the class of functions that are within a polynomial factor of c^n. If we round up c we can omit the star, as a polynomial times an exponential function is slow growing than any exponential function with a larger exponent. We only state the space usage, which we count in bits, when it is not polynomial.

3. Maximal independent sets

Miller and Muller [11] and Moon and Moser [12] independently gave the following upper bound on the number of MISs in a graph:

Theorem 1(Miller and Muller [11], Moon and Moser [12]). *The maximum number of MISs in any graph is*

$$\begin{cases} 3^{n/3} & \text{if } n \equiv 0 \pmod 3, \\ 4 \cdot 3^{(n-4)/3} & \text{if } n \equiv 1 \pmod 3, \\ 2 \cdot 3^{(n-2)/3} & \text{if } n \equiv 2 \pmod 3, \end{cases} \tag{1}$$

and the extremal graphs consist of a union of either $n/3$ K_3s, one K_4 or two K_2s and $(n-4)/3$ K_3s, or one K_2 and $(n-2)/3$ K_3s.

Using a combinatorial argument that resembles Moon and Moser's proof of Theorem 1, we get

Theorem 2. *The maximum number of k-MISs in any graph is*

$$\lfloor n/k \rfloor^{(\lfloor n/k \rfloor + 1)k - n} (\lfloor n/k \rfloor + 1)^{n - \lfloor n/k \rfloor k} \tag{2}$$

and the extremal graphs consist of a union of $k - (n \bmod k)$ $K_{\lfloor n/k \rfloor}$s and $(n \bmod k)$ $K_{\lfloor n/k \rfloor + 1}$s.

Proof. For neighbours v and w in G we define $G_{v \to w}$ as the graph obtained from G by replacing v by a copy of w and keeping the edge between them, see Fig. 1

We want to relate the number of k-MISs in $G_{v \to w}$ to the number of k-MISs in G,

$$|I^{=k}(G_{v \to w})| = |I^{=k}(G)| + \cdots.$$

None of the k-MISs in either of the graphs contain both v and w, since these are neighbours in both graphs. A k-MIS containing w in one of the graphs is also independent in the other as all changes involve edges containing v and it is also maximal as v cannot be added, since it is a neighbour of w in both graphs; thus, the k-MISs containing w are the same in the two

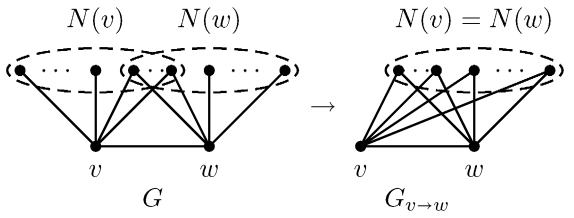

Fig. 1. Construction of $G_{v \to w}$.

graphs. The k-MISs in $G_{v \to w}$ containing v are exactly the same as those containing w, with v replacing w, so

$$|I^{=k}(G_{v \to w})| = |I^{=k}(G)| + |I_w^{=k}(G)| - |I_v^{=k}(G)| + \cdots.$$

All k-MISs in G that neither contain v nor w are also k-MISs in $G_{v \to w}$: they are independent since all changes involve only edges containing v and they are maximal since v cannot be added in $G_{v \to w}$ as it is a copy of w. $G_{v \to w}$ can have more k-MISs that neither contain v nor w than these; let the number of those be $f_{v \to w}^{=k}(G)$. Then

$$|I^{=k}(G_{v \to w})| = I^{=k}(G)| + |I_w^{=k}(G)| - |I_v^{=k}(G)| + f_{v \to w}^{=k}(G). \tag{3}$$

By the same argument,

$$|I^{=k}(G_{w \to v})| = |I^{=k}(G)| + |I_v^{=k}(G)| - |I_w^{=k}(G)| + f_{w \to v}^{=k}(G). \tag{4}$$

Suppose G has the maximum number of k-MISs among all graphs with n vertices and v and w are neighbours in G. Neither $G_{v \to w}$ nor $G_{w \to v}$ can have more k-MISs than G, so $|I_v^{=k}(G)| = |I_w^{=k}(G)|$ and $f_{v \to w}^{=k}(G) = f_{w \to v}^{=k}(G) = 0$ by (3) and (4). This implies that G can be replaced by $G_{v \to w}$ for neighbours v and w in G without changing $|I^{=k}(G)|$. Let $w \in V$ and $N(w) = \{v_1, v_2, \ldots, v_m\}$. By successively replacing G by $G_{v_i \to w}$ for $i = 1, 2, \ldots, m$, we end up with $\{w, v_1, v_2, \ldots, v_m\}$ being a connected component of the graph forming a clique. If we do so for all connected components, the resulting graph is a union of disconnected cliques. Let the number of vertices in these cliques be $i_1, i_2, \ldots, i_l$, where $i_1 + i_2 + \cdots + i_l = n$.

Then the number of k-MISs in G is

$$|I^{=k}(G)| = \begin{cases} i_1 \cdot i_2 \cdot \cdots \cdot i_l & \text{if } l = k, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

This expression is largest if none of the i_js differ by more than one, so (5) is maximal if $k - (n \bmod k) = (\lfloor n/k \rfloor + 1)k - n$ of the i_js are $\lfloor n/k \rfloor$ and $(n \bmod k) = n - \lfloor n/k \rfloor k$ of them are $\lfloor n/k \rfloor + 1$ giving $\lfloor n/k \rfloor^{(\lfloor n/k \rfloor + 1)k - n}(\lfloor n/k \rfloor + 1)^{n - \lfloor n/k \rfloor k}$. As we started out from a graph with the maximum number of k-MISs this is an upper bound on the number of k-MISs in any n-vertex graph and since we explicitly found the extremal graphs, this is also a lower bound obtained for the graphs consisting of a union of $k - (n \bmod k)$ $K_{\lfloor n/k \rfloor}$s and $(n \bmod k)$ $K_{\lfloor n/k \rfloor + 1}$s.

There are no other extremal graphs than those described above, by the following argument: look at the second-to-last graph G in the procedure above for changing an extremal graph into one consisting of a union of disconnected cliques. Now for some neighbours v, w in G, $G_{v \to w}$ is the last graph, so it is a union of disconnected cliques and so is $G[V \setminus v]$, since the changes only involve edges containing v. Now, v must have edges to all vertices in one of the cliques in G; otherwise, a k-MIS in $G_{v \to w}$ consisting of a vertex that is not the neighbour of v in G from each clique is not maximal in G since v can be added, so $f^{=k}_{v \to w}(G) > 0$, which contradicts that G is an extremal graph. Suppose v is fully connected to the clique C in G and $x \in C$. Now, $|I^{=k}_x(G)| = |I^{=k}_v(G)|$; otherwise, either $G_{x \to v}$ or $G_{v \to x}$ have more k-MISs than G, but this means that v cannot be connected to any other vertices in G, so $w \in C$ and $G = G_{v \to w}$; thus, the extremal graphs are exactly the ones described above. □

Theorem 3. *The maximum number of MISs of size at most k in any graph is equal to* (2) *for $k \leqslant n/3$ and equal to* (1) *for $k \geqslant n/3$ with equality for the same graphs as in Theorems* 1 *and* 2.

Proof. If we replace $I^{=k}$ by $I^{\leqslant k}$ and $f^{=k}_{v \to w}$ by $f^{\leqslant k}_{v \to w}$ in the proof of Theorem 2, we get that (5) with $l \leqslant k$ replacing $l = k$ is an upper bound on the number of MISs of size at most k. That expression is largest when the i_js are as small as possible, as long as they are at least three, so for $k \leqslant n/3$ the bound is equal to (2) and the extremal graphs are the same as in Theorem 2. For $k \geqslant n/3$ the expression is largest if all of the i_js are three. If $n \equiv 1 \pmod 3$ one of them should be a four and if $n \equiv 2 \pmod 3$ one of them should be a two. So we get the upper bound in (1) with the same extremal graphs as in Theorem 1. □

Theorem 4. *The maximum number of MISs of size at least k in any graph is equal to* (1) *for $k \leqslant n/3$ and equal to* (2) *for $k \geqslant n/3$ with equality for the same graphs as in Theorems* 1 *and* 2.

Proof. If we replace $I^{=k}$ by $I^{\geqslant k}$ and $f^{=k}_{v \to w}$ by $f^{\geqslant k}_{v \to w}$ in the proof of Theorem 2 we get that (5) with $l \geqslant k$ replacing $l = k$ is an upper bound on the number of MISs of size at least k. The maximum of that expression is equal to (1) for $k \leqslant n/3$ and (2) for $k \geqslant n/3$ with equality for the same graphs as in Theorems 1 and 2. □

Algorithm 1. Finding all k-MISs.

```
MIS(S, I, k):
    if |S| = k then
        check(I ∪ S)
    else if |S| > k > 0 then
        if the largest degree in S is ⩾ d then
            let v have the largest degree in S
            MIS(S \ N̄(v), I ∪ {v}, k − 1)
            MIS(S \ {v}, I, k)
        else
            let v have the smallest degree in S
            MIS(S \ N̄(v), I ∪ {v}, k − 1)
            for all w ∈ N(v) do
                MIS(S \ N̄(w), I ∪ {w}, k − 1)
```

We now give an algorithmic proof showing that we can find all k-MISs in a graph in time within a polynomial factor of our bound.

Theorem 5. *No graph contains more than*

$$d^{(d+1)k-n}(d+1)^{n-dk} \quad (6)$$

k-MISs for $d \in \mathbb{N} \setminus \{0\}$ and Algorithm 1 *finds them all in time within a polynomial factor of this bound.*

Setting $d = \lfloor n/k \rfloor$ we get (2). The bound (6) is only tight for $n/(d+1) \leqslant k \leqslant n/d$ by Theorem 2.

Proof. Algorithm 1 picks a vertex v and branches on including or excluding it. If I is a k-MIS that contains v

then $I\setminus\{v\}$ is a $(k-1)$-MIS in $G[V\setminus\bar{N}(v)]$ and if I does not contain v then I is a k-MIS in $G[V\setminus\{v\}]$ containing a neighbour of v. Finding a k-MIS in $G[V\setminus\bar{N}(v)]$ or $G[V\setminus\{v\}]$ is either done by making a recursive call on the graph or by trying to include each of the neighbours of v in turn. The parameter I is the independent set built so far, that must be extended to a MIS by adding at most k vertices from $G[S]$, *check*(I) outputs I if I is a MIS.

The number of leaves in the recursion tree of the algorithm running on a graph with n vertices is an upper bound on both the number of k-MISs and the running time of the algorithm (ignoring polynomial factors). Let $T(n,k)$ be the smallest function satisfying the recurrence relation

$$T(n,k)=\max\begin{cases} T(n-(d+1),k-1) \\ \quad +T(n-1,k), \\ T(n-1,k-1), \\ 2\cdot T(n-2,k-1), \\ \vdots \\ d\cdot T(n-d,k-1), \end{cases} \tag{7}$$

where $T(n,0)=T(k,k)=1$. Then $T(n,k)$ is an upper bound on the number of leaves in the recursion tree. Now, (6) is an upper bound on $T(n,k)$: if $n=k$, (6) equals $(d^d/(d+1)^{d-1})^k$ which is larger than one for $d\geqslant 1$ and if $k=0$, (6) equals $((d+1)/d)^n$ which is also larger than one. So by induction,

$$T(n-(d+1),k-1)+T(n-1,k) \leqslant (1+d)\cdot d^{(d+1)k-n}(d+1)^{n-dk-1}$$

which equals (6),

$$d\cdot T(n-d,k-1)\leqslant d\cdot d^{(d+1)k-n-1}(d+1)^{n-dk}$$

which also equals (6). Finally for $a<d$,

$$a\cdot T(n-a,k-1) \leqslant \frac{a\cdot(d+1)^{d-a}}{d^{d+1-a}}\cdot d^{(d+1)k-n}(d+1)^{n-dk}.$$

The last term is equal to (6) and the first to

$$\frac{(d+1)^{d+1-a}-(d+1-a)(d+1)^{d-a}}{((d+1)-1)^{d+1-a}}$$

and the numerator in this expression is the two first terms in the binomial formula for the denominator, so the whole expression is at most one; thus, (6) is an upper bound on $T(n,k)$. □

Theorem 6. *Eq.* (6) *is an upper bound on the number of MISs of size at most k in a graph for any $d\in\mathbb{N}, d\geqslant 3$ and Algorithm* 1 *modified so that it calls check(I) when $S=\emptyset$ in the first two lines and executes the* else*-part if only $k>0$ finds them all in time within a polynomial factor of* (6).

Proof. The only change in the recurrence relation is that $T(0,k)=1$ instead of $T(k,k)=1$. For $n=0$, (6) equals $(d^{d+1}/(d+1)^d)^k$, which is larger than one for $d\geqslant 3$, so for $d\geqslant 3$ we get the same running time. □

The algorithm in Theorem 6 with $d=3$ is essentially the algorithm of Eppstein [7].

Theorem 7. *Eq.* (6) *is an upper bound on the number of MISs of size at least k in a graph for $d=1,2$ and Algorithm* 1 *modified so that it calls check$(I\cup S)$ if either $|S|=k$ or $S=\emptyset$ and $k\leqslant 0$ in the first two lines and executes the* else*-part if $|S|>k$ finds them all in time within a polynomial factor of* (6).

Proof. Now, k can be negative. The recurrence relation is the same as in Theorem 6, except that $T(0,k)=1$ only for $k\leqslant 0$, which means that $(d^{d+1}/(d+1)^d)^k$ must be at least one for $k\leqslant 0$ which is true for $d=1,2$. □

Remark 8. The algorithms in Theorems 6 and 7 with $d=\lfloor n/k\rfloor$ run in time within a polynomial factor of the upper bounds for $k\leqslant n/3$, respectively, $k\geqslant n/3$. For other values of k, the bound is (1) and we can find all MISs in time within a polynomial factor of this bound by e.g. [14] or Remark 9.

Remark 9. If we modify the algorithm from Theorem 6 with $d=3$ to find all MISs (by removing k) the algorithm will run in time $O^*(3^{n/3})$.

Remark 10. If we want to find all MISs in a triangle-free graph we can use the algorithm from Remark 9 with a small adjustment: for a vertex v of degree two the two neighbours w_1 and w_2 cannot

themselves be neighbours as v, w_1 and w_2 would form a triangle. Then when branching on v, in the last recursive call we add w_2 and we can remove both its neighbours and w_1 as all MISs with w_1 have been found in the previous recursive call. We thus get a recurrence relation of the form $T(n) \geqslant 2 \cdot T(n-3) + T(n-4)$ and the algorithm has running time $O^*(2^{n/2})$ and finds at most $2^{n/2}$ MISs, the same as the bound of Hujter and Tuza [8].

4. Colouring

A k-colouring of a graph consists of a partition of the vertices into k disjoint-independent sets and we can assume that one of them is a MIS of size at least n/k. More general, we have

Lemma 11 (Madsen et al. [10]). *If $G[M]$ is a maximal k-colourable subgraph of G and $0 < k_1 < k$, there exists a maximal k_1-colourable subgraph $G[M']$ with $M' \subseteq M$ of G of size at least $|M| \cdot k_1/k$ such that $G[M \setminus M']$ is a maximal $(k-k_1)$-colourable subgraph of $G[V \setminus M']$.*

Proof. Pick among all k-colourings of $G[M]$ one with the largest set of vertices having only k_1 colours; they constitute a k_1-colourable subgraph $G[M']$ of G of size at least $|M| \cdot k_1/k$. By construction, $G[M']$ is maximal in $G[M]$. If any $v \in V \setminus M$ can be added to either $G[M']$ or $G[M \setminus M']$, $G[M \cup \{v\}]$ is k-colourable, but this contradicts the maximality of $G[M]$. So $G[M']$ is maximal in G and $G[M \setminus M']$ is maximal in $G[V \setminus M']$. □

We will use the following two techniques of Lawler [9] to check k-colourability of a graph:

Technique 1. *To check if a graph G is k-colourable, we check for all MISs I of size at least $\lceil n/k \rceil$ whether $G[V \setminus I]$ is $(k-1)$-colourable.*

The correctness follows from Lemma 11 with $M = V$ and $k_1 = 1$. Lawler used Technique 1 to get a running time of $O^*(3^{n/3})$ for checking 3-colourability, since 2-colourability can be checked in polynomial time. In our applications, we will use the following as an upper bound on the running time of Technique 1 (ignoring polynomial factors)

$$\sum_{j=\lceil n/k \rceil}^{n} |I^{=j}(G)| \cdot T_{k-1}(n-j),$$

where $T_{k-1}(n)$ is the running time for checking $(k-1)$-colourability of an n-vertex graph.

Technique 2. *To check if a graph is $2k$-colourable we check for all partitions of the vertex set whether both parts are k-colourable. This amounts to checking k-colourability of all subgraphs of the graph, so the running time is within a polynomial factor of*

$$\sum_{j=0}^{n} \binom{n}{j} \cdot T_k(j).$$

Lawler used Technique 2 to get a running time of $O^*(2^n)$ for 4-colouring, since 2-colourability can be checked in polynomial time.

For 3-colouring, the two techniques do not give us anything better than Eppstein's algorithm [6] which runs in time $O(1.3289^n)$, but for 4-, 5- and 6-colouring we can get something better than the previous best.

Theorem 12. *It can be checked in time $O(1.7504^n)$ whether a graph is 4-colourable.*

Proof. Using Technique 1, Theorem 5 with $d = 3$ and the 3-colouring algorithm of Eppstein, the running time is within a polynomial factor of

$$\sum_{k=\lceil n/4 \rceil}^{n} 3^{4k-n} 4^{n-3k} \cdot 1.3289^{n-k}$$

which is $O(4^{n/4} \cdot 1.3289^{3n/4}) = O(1.7504^n)$. □

Theorem 13. *It can be checked in time $O(2.1592^n)$ whether a graph is 5-colourable.*

Proof. Using Technique 1, Theorem 5 with $d = 4$ and the 4-colouring algorithm of Theorem 12, the running time is within a polynomial factor of

$$\sum_{k=\lceil n/5 \rceil}^{n} 4^{5k-n} 5^{n-4k} \cdot 1.7504^{n-k}$$

which is $O(5^{n/5} \cdot 1.7504^{4n/5}) = O(2.1592^n)$. □

Using Technique 1 for 6-colouring yields a running time of $O(2.5602^n)$. Using Technique 2 we get

Theorem 14. *It can be checked in time* $O(2.3289^n)$ *whether a graph is* 6*-colourable.*

Proof. The running time using Technique 2 is within a polynomial factor of

$$\sum_{i=0}^{n} \binom{n}{i} \cdot 1.3289^i$$

which is $O(2.3289^n)$. □

4.1. Using exponential space

To find the chromatic number of a graph, Christofides [4] noted that one can do something similar to Technique 1. The chromatic number of a graph G is one plus the minimum of the chromatic number of $G[V \setminus I]$ for all MISs I. In the algorithm of Christofides this is just done recursively, so the running time is $O^*(n!)$. Instead, Lawler [9] suggested to use dynamic programming and store the chromatic number of all subgraphs of G in a table X. If we index X with subsets of the vertices $X[S]$ in such a way that all subsets of S are before S in X we can simply run through X from $\emptyset$ to V and find the chromatic number of each subgraph. The running time of Lawler's algorithm is within a polynomial factor of

$$\sum_{S \subseteq V} |I(G[S])| \leqslant \sum_{i=0}^{n} \binom{n}{i} \cdot 3^{i/3}$$

which is $O((1+3^{1/3})^n) = O(2.4423^n)$. The algorithm uses space $O(\log n \cdot 2^n)$ to store X.

Eppstein [7] improved Lawler's algorithm by improving the way in which to fill the table X. First, X is initialised to hold the chromatic number of all subgraphs with chromatic number at most three by running Eppstein's 3-colouring algorithm on all subgraphs in time $O(2.3289^n)$ (same calculation as in the proof of Theorem 14). Then the algorithm runs through the table starting with the smaller subgraphs and 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: if $G[M]$ is a *maximal* k-colourable subgraph of G, then by Lemma 11 with $k_1 = k-1$ it has a maximal $(k-1)$-colourable subgraph $G[M']$ of size at least $|M'| \geqslant (k-1)/k \cdot |M|$, and $M \setminus M'$ is a MIS of $G[V \setminus M']$. The second part runs through X and for each subgraph $G[S]$ it finds all MISs I of $G[V \setminus S]$ of size at most $|S|/X[S]$ and updates the value of $X[S \cup I]$. Now, $X[S] = \chi(G[S])$ if $G[S]$ is a *maximal* k-colourable subgraph of G and since G is a maximal $\chi(G)$-colourable subgraph of itself, the algorithm correctly computes the chromatic number of G. Since the algorithm starts by finding all 3-colourable subgraphs, we only need to find all MISs of $G[V \setminus S]$ if $X[S] \geqslant 3$, so only MISs of size at most $|S|/3$ are considered. The running time is at most within a polynomial factor of

$$\sum_{S \subseteq V} |I^{\leqslant |S|/3}(G[V \setminus S])|.$$

Using (6) with $d = 3$ as a bound on the number of MISs of size at most $|S|/3$ (by Theorem 6) we get that this sum is at most

$$\sum_{S \subseteq V} 3^{4(|S|/3)-(n-|S|)} 4^{(n-|S|)-3(|S|/3)}$$

$$= \left(\frac{4}{3}\right)^n \sum_{i=0}^{n} \binom{n}{i} \cdot \left(\frac{3^{7/3}}{4^2}\right)^i$$

which is $O\left(\left(4/3 + 3^{4/3}/4\right)^n\right) = O(2.4151^n)$, so this is the running time of the algorithm. The space usage is $O(\log n \cdot 2^n)$.

4.2. k-colourable subgraphs

We want to use the idea of having a table for 6-colouring. In order to improve the running time, we need to mark all 3-colourable subgraphs in the table faster than $O(2.3289^n)$.

To find all maximal k-colourable subgraphs of a graph G, we use Lemma 11 and find all MISs I of G and for each I find all maximal $(k-1)$-colourable subgraphs of $G[V \setminus I]$. This will also find some k-colourable subgraphs that are not maximal. In [10], we show the following bound on the number of maximal bipartite subgraphs (MBSs):

Theorem 15 (Madsen et al. [10]). *No graph contains more than* $O(12^{n/4})$ *MBSs and we can find all MBSs of a graph in time* $O^*(12^{n/4}) = O(1.8613^n)$.

Remark 16. We do not know of any graphs matching the bound in Theorem 15. Currently, the best lower bound is obtained by a family of graphs containing

$105^{n/10} = 1.5926^n$ MBSs [10], so there is a rather large gap between this number and the upper bound of $O(1.8613^n)$ from Theorem 15.

For maximal 3-colourable subgraphs we have the following theorem:

Theorem 17. *We can find all (maximal) 3-colourable subgraphs of a graph in time* $O(2.2680^n)$ *using space* $O(2^n)$.

Proof. We use the above-mentioned algorithm and find all MISs I in G and for each find all MBSs in $G[V \setminus I]$. This will find all maximal 3-colourable subgraphs, but also some that are not maximal. We cannot check in polynomial time, whether a 3-colourable subgraph is maximal as we can for independent sets and bipartite subgraphs; instead, we mark all the 3-colourable subgraphs we find in a table X. We start by marking them as being maximal 3-colourable. Then we run through X from V to $\emptyset$ and for each subgraph $G[S]$ that is 3-colourable we mark $G[S \setminus \{v\}]$ as a non-maximal 3-colourable subgraph for all $v \in S$. Since each 3-colourable subgraph is either maximal or we can add another vertex to it and it remains 3-colourable this will correctly mark all subgraphs.

The running time of the first part is within a polynomial factor of

$$\sum_{I \in I(G)} |M(G[V \setminus I])| \leqslant \sum_{k=1}^{n} \sum_{I \in I^{=k}(G)} |M(G \setminus I)|,$$

where $M(G)$ denotes the number of MBSs in G. Splitting the first sum in two at $k = n/5$ and using the bound of $O(12^{n/4})$ from Theorem 15 on $M(G)$ and the bound from Theorem 5 with $d = 4$ and 5 we get that the sum is at most

$$\sum_{k=1}^{\lceil n/5 \rceil - 1} 5^{6k-n} 6^{n-5k} 12^{(n-k)/4} + \sum_{k=\lceil n/5 \rceil}^{n} 4^{5k-n} 5^{n-4k} 12^{(n-k)/4}$$

which is $O(60^{n/5}) = O(2.2680^n)$. The running time of the second part is $O^*(2^n)$. □

Corollary 18. *We can find all maximal 3-colourable subgraphs of a graph and possibly some that are not maximal in time* $O(2.2680^n)$ *using only polynomial space.*

Remark 19. We do not have any non-trivial, i.e., $o(2^n)$ upper bound on the number of maximal 3-colourable subgraphs in a graph.

If we mark all 3-colourable subgraphs in a table, we can decide if G is 6-colourable by Technique 2 in time $O^*(2^n)$; thus, we have the following theorem:

Theorem 20. *It can be checked in time* $O(2.2680^n)$ *whether a graph is 6-colourable using space* $O(2^n)$.

If we want to find all maximal 4-colourable subgraphs of a graph, we use Theorem 17 to mark all 3-colourable subgraphs in a table X in time $O(2.2680^n)$. We then find all MISs I of G and find all 3-colourable subgraphs $G[M]$ of $G[V \setminus I]$ and mark $G[I \cup M]$ as 4-colourable. Finding all 3-colourable subgraphs of $G[V \setminus I]$ can be done in time $O(2^{|V \setminus I|})$ by simply running through the entries in X corresponding to subgraphs of $G[V \setminus I]$ and check whether they are marked. We thus have

Theorem 21. *We can find all (maximal) 4-colourable subgraphs of a graph in time* $O(2.4023^n)$ *using space* $O(2^n)$.

Proof. The running time for finding all (maximal) 4-colourable subgraphs is within a polynomial factor of

$$\sum_{I \in I(G)} 2^{|V \setminus I|} \leqslant \sum_{k=1}^{n} |I^{=k}(G)| \cdot 2^{n-k}.$$

We split the sum in two at $k = n/5$ and use Theorem 5 with $d = 4$ and $d = 5$ to get an upper bound of

$$\sum_{k=1}^{\lceil n/5 \rceil - 1} 5^{6k-n} 6^{n-5k} 2^{n-k} + \sum_{k=\lceil n/5 \rceil}^{n} 4^{5k-n} 5^{n-4k} 2^{n-k}$$

which is $O(80^{n/5}) = O(2.4023^n)$. □

If we mark all 4-colourable subgraphs in a table, we can find all 5-colourable subgraphs in the same way and since the time for finding all 4-colourable subgraphs of $G[V \setminus I]$ is also $O(2^{|V \setminus I|})$ we get the same running time as for finding maximal 4-colourable subgraphs. Now, we can do the same thing for maximal

6-colourable subgraphs and in general for maximal k-colourable subgraphs. As we only need to remember the $(k-1)$-colourable subgraphs when finding the k-colourable ones we only need space $O(2^n)$. We have the following theorem:

Theorem 22. *We can find all (maximal) k-colourable subgraphs of a graph in time $O(2.4023^n)$ using space $O(2^n)$.*

4.3. Chromatic number

By applying Theorem 22 at most n times we can find the chromatic number of G, since it is the smallest k for which G is a k-colourable subgraph of itself. We thus have:

Theorem 23. *We can find the chromatic number of a graph in time $O(2.4023^n)$ using space $O(2^n)$.*

The algorithm of Eppstein [7] mentioned in Section 4.1 starts by precomputing all 3-colourable subgraphs. Above, we have shown how to find all 4-colourable subgraphs of a graph in time $O(2.4023^n)$. Having marked all 4-colourable subgraphs, the second part of Eppstein's algorithm will run in time $O(2.3814^n)$ by the same calculation as previously, except that we only need to consider MISs of size at most $|S|/4$ so we can use (6) with $d=4$ by Theorem 6. In total, we get a running time of $O(2.4023^n)$, the same as in Theorem 23. This uses space $O(\log n \cdot 2^n)$, as we need to store the chromatic number of each subgraph, but the polynomial in the running time is smaller than in Theorem 23.

5. Open problems

The bound of Moon and Moser on the number of MISs in a graph is tight. For MBSs, we have an upper bound of $O(1.8613^n)$, but currently the best lower bound is only 1.5926^n, so there is a gap to fill. For maximal k-colourable subgraphs we do not even have an $o(2^n)$ upper bound for $k \geqslant 3$.

The algorithm in [14] for finding all MISs in a graph runs in time within a polynomial factor of the number of MISs *in the graph*, whereas Algorithm 1 for finding all k-MISs in a graph runs in time within a polynomial factor of our upper bound. For MBSs we have an algorithm running in time $O(1.8613^n)$, which could be within a polynomial factor of the upper bound, but for maximal k-colourable subgraphs for $k \geqslant 3$ the running times are worse than $O(2^n)$ which is clearly not within a polynomial factor of the upper bound. So faster algorithms for finding all maximal k-colourable subgraphs, ideally running in time within a polynomial factor of the upper bound or even the number of maximal k-colourable subgraphs in the graph are desirable.

Such algorithms can lead to faster algorithms for colouring. We can, for instance, check if a graph is 4-colourable in time within a polynomial factor of the time for finding all MBSs, but also faster algorithms for colouring not using the ideas of finding all possible k-colourable subgraphs are worth pursuing. At the moment only k-colouring for $k \leqslant 6$ is faster than actually computing the chromatic number of the graph, but there is no evidence, that deciding 7-colourability should be as difficult as finding the actual chromatic number.

References

[1] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, in: D.-Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Vol. 4, Kluwer Academic Publishers, Boston, MA, 1999, pp. 1–74.

[2] J.M. Byskov, Chromatic number in time $O(2.4023^n)$ using maximal independent sets, Research Series RS-02-45, BRICS, Department of Computer Science, University of Aarhus, December 2002; URL http://www.brics.dk/RS/02/45/.

[3] J.M. Byskov, Algorithms for k-colouring and finding maximal independent sets, in: Proceedings of the 14th Symposium on Discrete Algorithms, SIAM, 2003, pp. 456–457.

[4] N. Christofides, An algorithm for the chromatic number of a graph, Comput. J. 14 (1) (1971) 38–39.

[5] C. Croitoru, On stables in graphs, in: Proceedings of the Third Colloqium on Operations Research (Cluj-Napoca, 1978), University "Babeş-Bolyai", Cluj-Napoca, 1979, pp. 55–60.

[6] D. Eppstein, Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction, in: Proceedings of the 12th Symposium on Discrete Algorithms, ACM and SIAM, 2001, pp. 329–337.

[7] D. Eppstein, Small maximal independent sets and faster exact graph coloring, J. Graph Algorithms & Applications 7 (2) (2003) 131–140.

[8] M. Hujter, Z. Tuza, The number of maximal independent sets in triangle-free graphs, SIAM J. Discrete Math. 6 (2) (1993) 284–288.
[9] E.L. Lawler, A note on the complexity of the chromatic number problem, Inf. Process. Lett. 5 (3) (1976) 66–67.
[10] B.A. Madsen, J.M. Nielsen, B. Skjernaa, On the number of maximal bipartite subgraphs of a graph, Research Series RS-02-17, BRICS, Department of Computer Science, University of Aarhus, April 2002; URL http://www.brics.dk/RS/02/17/.
[11] R.E. Miller, D.E. Muller, A problem of maximum consistent subsets, Research Report RC-240, IBM Research Center, March 1960.
[12] J.W. Moon, L. Moser, On cliques in graphs, Israel J. Math. 3 (1965) 23–28.
[13] I. Schiermeyer, Fast exact colouring algorithms, Tatra Mountains Math. Publ. 9 (1996) 15–30.
[14] S. Tsukiyama, M. Ide, H. Ariyoshi, I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6 (3) (1977) 505–517.