Information Processing Letters 111 (2011) 251–255

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Dominating set based exact algorithms for 3-coloring

N.S. Narayanaswamy [a], C.R. Subramanian [b,*]

[a] Department of CSE, IIT Madras, Chennai, India
[b] The Institute of Mathematical Sciences, Chennai, India

ARTICLE INFO

Article history:
Received 6 February 2009
Received in revised form 28 October 2010
Accepted 17 November 2010
Available online 19 November 2010
Communicated by Wen-Lian Hsu

Keywords:
Graph algorithms
Design of algorithms
Graph coloring
3-coloring
Exact algorithms
Exponential time algorithms

ABSTRACT

We show that the 3-colorability problem can be solved in $O(1.296^n)$ time on any n-vertex graph with minimum degree at least 15. This algorithm is obtained by constructing a dominating set of the graph greedily, enumerating all possible 3-colorings of the dominating set, and then solving the resulting 2-list coloring instances in polynomial time. We also show that a 3-coloring can be obtained in $2^{o(n)}$ time for graphs having minimum degree at least $\omega(n)$ where $\omega(n)$ is any function which goes to ∞. We also show that if the lower bound on minimum degree is replaced by a constant (however large it may be), then neither a $2^{o(n)}$ time nor a $2^{o(m)}$ time algorithm is possible (m denotes the number of edges) for 3-colorability unless Exponential Time Hypothesis (ETH) fails. We also describe an algorithm which obtains a 4-coloring of a 3-colorable graph in $O(1.2535^n)$ time.

1. Introduction

A proper 3-coloring of a graph on n vertices is an assignment of 3 colors, $\{\mathcal{R}, \mathcal{G}, \mathcal{B}\}$, to the vertices such that no two adjacent vertices get the same color. Whenever it is clear from the context, we simply refer to it as a 3-coloring. A list 2-coloring of a graph, with each vertex having an associated list of two colors, is a color assignment to the vertices from their respective lists such that no two adjacent vertices get the same color. It is a well-known fact that list 2-colorability can be solved in polynomial time. Determining if a given graph $G = (V, E)$ admits a proper 3-coloring is a well-studied algorithmic problem in graph theory and combinatorics. While this problem is known to be NP-complete (see [12]) and hence is unlikely to admit polynomial time algorithms, the need to solve instances of this problem arising in practice still persists. While various approaches (like approximation algorithms or focusing on random instances) have been introduced and developed to circumvent the intractability, one still needs to compromise on the quality of the solution or on the solvability of an instance. Hence, the need to develop *exact* algorithms (those which solve the problem exactly as required) is still there, even though such algorithms may require exponential time in the worst-case. When specifying time bounds, we specify only the exponential factor (c^n for $c > 1$) involved and ignore the polynomial factors which can always be absorbed (for sufficiently large n) by marginally increasing c. Throughout, we use n to denote the number of vertices in the input graph.

The field of designing and analyzing exact algorithms for NP-hard problems is steadily growing. For 3-coloring graphs on n vertices, there have been several attempts. Lawler [17] describes a very simple algorithm (based on enumerating all maximal independent sets) and this approach can be shown to run in 1.4422^n time. Schiermeyer [18] describes a slightly more complicated 1.415^n time algorithm. Beigel and Eppstein [2,3] present a 1.3289^n time result significantly improving the previous results. Byskov [7] shows how to obtain a 4-, 5- or 6-coloring in time 1.7504^n, 2.1592^n and 2.3289^n, respectively, using polynomial space. He also shows how to obtain a 6- or 7-coloring in time 2.268^n and 2.4023^n, respectively, us-

* Corresponding author.
E-mail addresses: swamy@cse.iitm.ac.in (N.S. Narayanaswamy), crs@imsc.res.in (C.R. Subramanian).

0020-0190/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2010.11.009

ing exponential space. As for finding the chromatic number (denoted $\chi(G)$) exactly, Lawler [17] gave an algorithm that runs in 2.4423^n time. Building on his works, Eppstein [11] improved the bound to 2.415^n time. Later, Byskov [6] improved the bound further and showed that $\chi(G)$ can be computed exactly in 2.4023^n time. Björklund, Husfeldt [4] and Koivisto [16] showed that $\chi(G)$ can be computed in 2^n time. All these algorithms require exponential space in the worst case. Björklund, Husfeldt and Koivisto [4,16] also showed that with polynomial space $\chi(G)$ can be computed in 2.2461^n time. For graphs of bounded maximum degree, Björklund, Husfeldt, Kaski and Koivisto [5] obtained c^n-time algorithms, with $c < 2$, for determining $\chi(G)$.

1.1. Our contribution

Let $\delta(G)$ denote the minimum degree of a graph G. We consider exact algorithms for 3-coloring graphs from a graph class parameterized by an integer valued function $d = d(n)$. Let 3-COL(d) denote the corresponding decision question which is: Given a graph G for which $\delta(G) \geqslant d(n)$, is it 3-colorable? It can be easily seen that this question is NP-complete even if we restrict $d(n) = 3$ for every n and is also known [9] to be NP-complete even if we restrict G to be a 4-regular planar graph. In fact, as shown by Edwards in [10], it is NP-complete even if we assume that $d(n) \geqslant n^{1-\eta}$ where $\eta > 0$ is any small constant. In Section 2, we give an alternative proof of this fact. In Section 3, we use the degree lower bound to construct a *small* dominating set which is then used to design an exact algorithm to test if a given graph is 3-colorable. This is used to show that we can bring down the running time to any c^n ($c > 1$ is a constant) if we only assume that $\delta(G)$ is a suitably large constant (which depends on c). In fact, it follows that for all graphs with minimum degree δ at least 15, a 3-coloring (if it exists) can be found in time $O(1.297^n)$ which is better than the current best algorithms for 3-coloring. It also follows that a 3-coloring can be found in $2^{o(n)}$ time if the minimum degree increases (however slowly) with n. In Section 4, we look at an approach based on maximal independent sets and show that 3-colorable graphs can be 4-colored in $O(1.2535^n)$ time. Note that there is no polynomial time algorithm for 4-coloring a 3-colorable graph unless $P = NP$ (see [13]).

2. NP-completeness of 3-COL($d(n)$)

We present an alternate proof of the following theorem obtained in [10].

Theorem 2.1. *For every η, $0 < \eta \leqslant 1$, 3-COL($n^{1-\eta}$) is NP-complete.*

Proof. Given a graph G testing membership in 3-COL($n^{1-\eta}$) is clearly in NP: we check that the minimum degree is at least $n^{1-\eta}$, and then given a 3-coloring, it is straightforward to verify in polynomial time that it is indeed a proper 3-coloring. To prove that 3-COL($n^{1-\eta}$) is NP-complete, we present a polynomial time many-to-one reduction from 3-COL(3) to 3-COL($n^{1-\eta}$). Clearly, 3-COL(3) is just the 3-coloring problem restricted to graphs of minimum degree at least 3, and is NP-complete as mentioned before.

Given an instance of 3-COL(3), say $G = (V, E)$, our polynomial time reduction outputs a graph $H = (V_H, E_H)$ defined as follows. For each $u \in V$, consider a set I_u of n^a new vertices. The value of a will be chosen later. Define $V_H = \bigcup_{u \in V} I_u$. For each edge $\{u, v\} \in E$, let $E_{u,v} = \{\{u', v'\}\colon\ u' \in I_u, v' \in I_v\}$. Define $E_H = \bigcup_{\{u,v\} \in E} E_{u,v}$. That is, H is obtained from G by replacing each vertex u of G by an independent set I_u of size n^a and adding a complete bipartite graph between I_u and I_v whenever u and v are adjacent in G.

It is clear that $|V_H| = n^{1+a}$ and $\delta(H) \geqslant 3n^a$. We need to choose a so that $3n^a \geqslant n^{(1+a)(1-\eta)}$. It suffices to choose a so that $a \geqslant (1+a)(1-\eta)$. It can be verified easily that if $a = \frac{1}{\eta} - 1$, then $a \geqslant (1+a)(1-\eta)$. Also, it can be easily verified that H is 3-colorable if and only if G is 3-colorable. This proves that 3-COL($n^{1-\eta}$) is NP-complete. $\square$

3. Dominating sets and exact algorithms for 3-coloring

Enumerating all proper 3-colorings. We start by showing that in a graph G with r connected components, it is possible to enumerate all 3-colorings in time $3^r \times 2^{n-r}$. To do this, we show that in a connected graph G, we can enumerate all 3-colorings in time $3 \times 2^{n-1}$. The algorithm that achieves this is to consider a breadth-first traversal of G starting at a vertex v, and search for a 3-coloring layer-by-layer in the breadth-first traversal. The first layer consists of only vertex v, and it can be colored in 3-possible ways. After coloring v with one of the three colors, say $\mathcal{R}$, each vertex in the second layer, say L_2, can be colored only from the set $\{\mathcal{G}, \mathcal{B}\}$. All proper 2-colorings of this layer can be enumerated in time $2^{|L_2|}$. The search for a 3-coloring of G is based on this observation by searching the space of all partial 3-colorings recursively. In general, for $i \geqslant 1$, a 3-coloring of $\bigcup_{j \leqslant i} L_j$ results in a list 2-coloring instance on the vertices of L_{i+1}, and all proper colorings of this instance can be enumerated in time $2^{|L_{i+1}|}$. The total number of partial 3-colorings explored is at most $3 \times 2^{n-1}$, and any proper 3-coloring will be discovered by this recursive procedure. This argument also shows that the existence of a 3-coloring can be checked in time 2^n, by observing that starting with an assignment of any one of the 3 colors to a vertex in a connected component will lead to a 3-coloring, if it exists. The same argument also shows that counting the number of 3-colorings of a connected graph can be done in 2^n time. Applying this observation to each connected component of a graph G with r components, we obtain the following observations on enumerating and counting the number of 3-colorings of a graph. We state this as a theorem.

Theorem 3.1. *Let G be a graph with r connected components with each component on at most m vertices. Then,* (i) *It is possible to enumerate all proper* 3*-colorings of G in time $3^r \times 2^{n-r}$;* (ii) *It is possible to determine exactly the number of* 3*-colorings of G in time $r2^m \leqslant r2^{n-r}$;* (iii) *In particular, it is possible to check if G is* 3*-colorable in $r2^m$ time.*

Essentially, the same arguments lead to the following theorem also:

Theorem 3.2. *Let G be a connected graph with a given set S on k vertices such that $G - S$ has r connected components $C_1, \ldots, C_r$ on $n_1, \ldots, n_r$ vertices respectively. Let $m = \max_j n_j$. Then,* (i) *It is possible to determine exactly the number of* 3*-colorings of G in time $3^k \cdot (\sum_j 2^{n_j}) \leqslant r3^k 2^m$*; (iii) *In particular, it is possible to check if G is* 3*-colorable in $r3^k 2^m$ time.*

Dominating sets and testing membership in 3-COL(δ). The generic algorithm for testing if a given graph of minimum degree δ is 3-colorable is as follows:

3-coloring algorithm. Select a dominating set D. Exhaustively enumerate each proper 3-coloring of the subgraph induced by D and check if it can be extended to a proper 3-coloring of G.

Since D is a dominating set, each vertex in $V - D$ is adjacent to some vertex in D. Therefore, for each 3-coloring of D, there are at most two eligible colors for each vertex in $V - D$. In other words, each 3-coloring of D yields an instance of list coloring on the subgraph induced by $V - D$ in which each vertex has a list of at most two colors associated with it. It is well-known that one can test in polynomial time if there is a proper coloring for an instance of list 2-coloring. Therefore, the running time of the algorithm is at most $3^r 2^{|D|-r}$ (ignoring polynomial multiplicative factors) where r is the number of connected components of subgraph induced by D. In the case when D is a connected dominating set, then enumerating all 3-colorings of D can be done in $3 \times 2^{|D|}$ time by applying Theorem 3.2. We thus obtain the following theorem:

Theorem 3.3. *Let D be a dominating set of a graph G, and let the subgraph induced by D have r connected components. Then, a* 3*-coloring of G can be found, if it exists, in $3^r 2^{|D|-r}$ time.*

We now apply Theorem 3.3 to obtain two corollaries. For the first corollary, we use a classical result which states that in a graph G of minimum degree δ, it is possible to find in deterministic polynomial time a dominating set of size at most $n\frac{1+\ln(\delta+1)}{1+\delta}$ (see [1] for details). This yields the following corollary:

Corollary 3.1. *Let G be a graph with minimum degree $\delta \geqslant 3$. A* 3*-coloring of G can be found, if it exists, in time $(3^{\frac{1+\ln(\delta+1)}{1+\delta}})^n$. For $\delta \geqslant 15$ and $\delta \geqslant 50$, the running time is at most 1.29569^n and 1.1^n, respectively.*

Further, we get the following corollaries on testing 3-colorability.

Corollary 3.2. *Let G be a graph with minimum degree $\delta \geqslant \alpha n$ for some constant $\alpha > 0$. Then, a* 3*-coloring of G, if it exists, can be found in deterministic polynomial time.*

A simple counting argument establishes that any n-vertex graph G having more than $n^2/3$ edges is not 3-colorable. This implies that if $\delta(G) > 2n/3$, then G is not 3-colorable.

Corollary 3.3. *For any fixed $c > 1$, there exists a threshold $g(c)$ such that if G is any graph with minimum degree $\delta \geqslant g(c)$, then a* 3*-coloring of G* (*if it exists*) *can be found in c^n time.*

Corollary 3.4. *For any fixed $\omega = \omega(n)$ such that $\omega \to \infty$, the following holds*: *if G is any graph with minimum degree $\delta \geqslant \omega$, then a* 3*-coloring of G* (*if it exists*) *can be found in $2^{o(n)}$ time.*

Impagliazzo and Paturi proposed in [14] the following Exponential Time Hypothesis (ETH):

ETH. For every $k \geqslant 3$, there is no $2^{o(n)}$ time algorithm for solving k-SAT, where n denotes the number of variables in the input instance.

This is a stronger assumption than $P \neq NP$. But it is quite likely true also since no $2^{o(n)}$ time algorithm has been designed yet despite the efforts of researchers over several decades. Also, Impagliazzo, Paturi and Zane obtained in [15] the following theorem:

Theorem 3.4. *(See [15].) For every $k \geqslant 3$, k-SAT can be solved in $2^{o(n)}$ time if and only if it can be solved in $2^{o(m)}$ time, where n and m denote respectively the number of variables and clauses in the input instance.*

The standard polynomial time reduction (see Exercise 36.2 of [8]) from 3-SAT to 3-colorability reduces an instance F on n variables and m clauses to an instance G on $n' = 2n + 5m + 3$ vertices and $m' = 3n + 10m + 3$ edges. Using this reduction and applying Theorem 3.4, the following observation can be easily deduced.

Theorem 3.5. *Assuming ETH, there is neither a $2^{o(n)}$ time algorithm nor a $2^{o(m)}$ time algorithm for solving* 3*-colorability of an arbitrary graph on n vertices and m edges.*

Let $d \geqslant 0$ be any arbitrary but fixed integer. The polynomial time reduction of Theorem 2.1 transforms an arbitrary instance G (having no isolated vertices) on n vertices and m edges to an instance G' on $n' = nd$ vertices and $m' = md^2$ edges and having minimum degree at least d. This means that a $2^{o(n)}$ time algorithm for graphs with $\delta(G) \geqslant d$ implies a $2^{o(n)}$ time algorithm for arbitrary graphs which contradicts Theorem 3.5, assuming ETH. Thus, we get the following corollary:

Corollary 3.5. *Assuming ETH, for every* fixed *integer $d \geqslant 0$, there is neither a $2^{o(n)}$ time algorithm nor a $2^{o(m)}$ time algorithm for solving* 3*-colorability of an arbitrary graph on n vertices and m edges, with $\delta(G) \geqslant d$.*

Note. In view of Corollary 3.5, we notice that the lower bound ω (where $\omega = \omega(n) \to \infty$ as $n \to \infty$) on $\delta(G)$ mentioned in Corollary 3.4 is the best asymptotic lower bound one can hope for designing subexponential algorithms, provided ETH holds true.

4. On using maximal independent sets for coloring

Here we use a maximal independent set in the graph, which is a dominating set due to its maximality, to design an algorithm that outputs a coloring with at most 4 colors if the given graph is 3-colorable. However, as we will see, this algorithm cannot be used to recognize if the given graph is not 3-colorable. Note that if $P \neq NP$, then there is no polynomial time algorithm which always outputs a 4-coloring of a given 3-colorable graph [13].

Algorithm. Set $\alpha = 1/4.86537$. Select a maximal independent set I in polynomial time. If $|I|$ is *small*, that is, $|I| \leqslant \alpha|V|$, then enumerate all proper 3-colorings of I, and check if each coloring can be extended to a 3-coloring of G. If none of the colorings of I extends to a 3-coloring of G, then report that the graph is not 3-colorable. On the other hand, if $|I|$ is *large*, that is $|I| > \alpha n$, then test if the subgraph $G[V-I]$ is 3-colorable. If it is 3-colorable, then find a proper 3-coloring of $G[V-I]$ and output a 4-coloring of G in which the vertices of I get the fourth color. If $G[V-I]$ is not 3-colorable, then report that G is not 3-colorable.

Theorem 4.1. *The above algorithm outputs a* 4*-coloring of any given* 3*-colorable graph, in time at most* $3^{\frac{n}{4.86357}} \leqslant 1.2535^n$.

Proof. The time spent by the algorithm when I is small is $3^{|I|}$, and if I is large the time spent is at most $(1.3289)^{n-|I|}$ by using Eppstein's algorithm [11]. Both these terms are bounded by $3^{\alpha n}$. Therefore, the running time of the algorithm is at most $3^{\frac{n}{4.86357}}$. $\square$

Remark. We now show that it is possible to compute in $3^{n/4}$ time (given a 3-colorable graph G on n vertices) *either* a proper 3-coloring of G *or* a maximal independent set I that is 3-colored in every proper 3-coloring of G. Select a maximal independent set I in polynomial time. We first check if I is colored with a single color in some 3-coloring, and this is done by checking if the subgraph induced by $V-I$ is bipartite. Otherwise, if I is *small* (that is, $|I| \leqslant |V|/4$), then we enumerate all 3-colorings of I, and check, for each such coloring, if it can be extended to a proper 3-coloring of G. This takes $3^{|I|} \leqslant 3^{n/4}$ time. In the case when $|I|$ is large, we consider the subgraph induced by $V-I$, and enumerate each maximal independent I' of the subgraph induced by $V-I$ and test if there is a 3-coloring of G in which I' is assigned a single color. To do this, we color I', without loss of generality, with the color $\mathcal{R}$, and then color as many vertices of I as possible with the color $\mathcal{R}$, and then test if the remaining uncolored vertices form a bipartite graph. In case we find a proper coloring by this procedure, it is a proper 3-coloring of G. Further, the time complexity of this part of the algorithm is determined by the number of maximal independent sets in the induced subgraph $G[V-I]$. Since the number of maximal independent sets in $G[V-I]$ is at most $3^{\frac{n-|I|}{3}} \leqslant 3^{\frac{n}{4}}$, the running time of this step is at most $3^{\frac{n}{4}}$. In either case, the running time of the algorithm is at most $3^{\frac{n}{4}} \leqslant 1.3161^n$ and this is better than the current best known time complexity of 1.3289^n for 3-coloring. If we do not find a 3-coloring by this approach, then the following inference can be made about I: I is 3-colored in every 3-coloring of G. Further, each maximal independent set in $G[V-I]$ is colored with at least 2 colors in every 3-coloring of G.

5. Conclusions

We have shown that finding a 3-coloring can be realized more efficiently for graphs with large minimum degree. In fact, we have shown that any lower bound (which, as a function of n, goes to ∞) helps us to design a $2^{o(n)}$ time algorithm for 3-coloring. In addition, we also show that it is not possible (unless ETH fails) to strengthen the lower bound to some constant. This, in effect, shows that our results are "essentially" tight. A question related to 3-coloring is: Can we add edges to the graph (without destroying the 3-colorability) to increase its minimum degree. However, it is clearly true that the problem of finding even one edge which can be added without destroying the 3-colorability is at least as hard as 3-coloring itself. Another line of study is to see what structural handles we can get from a maximal independent set that is 3-colored in every 3-coloring.

Acknowledgements

The authors are thankful to Saket Saurabh for pointing out ETH and the references [14] and [15].

References

[1] N. Alon, J. Spencer, The Probabilistic Method, John Wiley and Sons, 2000.
[2] R. Beigel, D. Eppstein, 3-coloring in time $O(1.3446^n)$: a no-MIS algorithm, in: Proceedings of the 36th IEEE Symposium on Foundations of Computer Science, IEEE, 1995, pp. 444–453.
[3] R. Beigel, D. Eppstein, 3-coloring in time $O(1.3289^n)$, Journal of Algorithms 54 (2005) 168–204.
[4] A. Björklund, T. Husfeldt, Inclusion–exclusion algorithms for counting set partitions, in: Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS-2006), 2006, pp. 575–582.
[5] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Trimmed Moebius Inversion and graphs of bounded degree, in: Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science (STACS-2008), 2008, pp. 85–96.
[6] J.M. Byskov, Chromatic number in time $O(2.4023^n)$ using maximal independent sets, Tech. Rep. RS-02-45, Center for Basic Research in Computer Science (BRICS), December 2002.
[7] J.M. Byskov, Algorithms for k-coloring and finding maximal independent sets, in: Proceedings of the 14th ACM–SIAM Symposium on Discrete Algorithms, ACM, 2003, pp. 456–469.
[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, Cambridge University Press, 2001.
[9] D.P. Dailey, Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete, Discrete Mathematics 30 (1980) 289–293.
[10] K. Edwards, The complexity of coloring problems on dense graphs, Theoretical Computer Science 43 (1986) 337–343.
[11] D. Eppstein, Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction, in: Proceedings of the 12th ACM–SIAM Symposium on Discrete Algorithms, ACM, 2001, pp. 329–337.
[12] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.
[13] V. Guruswami, S. Khanna, On the hardness of 4-coloring a 3-colorable graph, SIAM Journal on Discrete Mathematics 18 (1) (2005) 30–40.
[14] R. Impagliazzo, R. Paturi, On the complexity of k-SAT, Journal of Computer and System Sciences 62 (2001) 367–375.

[15] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity? Journal of Computer and System Sciences 63 (2001) 512–530.

[16] M. Koivisto, An $O^*(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion–exclusion, in: Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS-2006), 2006, pp. 583–490.

[17] E.L. Lawler, A note on the complexity of the chromatic number problem, Information Processing Letters 5 (3) (1976) 66–67.

[18] I. Schiermeyer, Deciding 3-colorability in less than $O(1.415^n)$ steps, in: Proceedings of the 19th International Workshop on Graph-Theoretic Concepts in Computer Science, in: Lecture Notes in Comput. Sci., vol. 790, Springer-Verlag, 1994, pp. 177–182.