Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 46 (2014) 89–96

www.elsevier.com/locate/endm

An Approximate Algorithm for the Chromatic Number of Graphs

Guillermo de Ita Luna [1,2]

Facultad de Ciencias de la Computación, BUAP, Puebla, México

J. Raymundo Marcial-Romero [3]

Facultad de Ingeniería, UAEMex, Toluca, México

Yolanda Moyao [4]

Facultad de Ciencias de la Computación, BUAP, Puebla, México

Abstract

We have designed a novel polynomial-time approximate algorithm for the graph vertex colouring problem. Contrary to the common top-down strategy for solving the colouring graph problem, we propose a bottom-up algorithm for colouring graphs. Given an input graph G, we establish an upper bound to approximate the colouring of the input grap given by $\lceil \delta(G)/2 \rceil + 2$ where $\delta(G)$ is the average degree of G.

Keywords: Graph Coloring, Approximate Algorithm, Chromatic Number.

[1] Work supported by SNI-Conacyt-México

[2] Email:`deita@cs.buap.mx`

[3] Email:`jrmarcialr@uaemex.mx`

[4] Email:`ymoyao@cs.buap.mx`

http://dx.doi.org/10.1016/j.endm.2014.08.013

1 Introduction

Graph vertex colouring problem is an active field of research, with many interesting subproblems [4,5,6] and applications in areas like scheduling, frequency allocation, planning, etc [2].

The graph colouring problem consists in colouring properly the vertices of a graph with the smallest possible number of colours, so that no two adjacent vertices receive the same colour. If a colouring with k colours exists, then the graph is said to be k-colourable. The chromatic number of a graph G, denoted as $\chi(G)$, represents the minimum number of colours for proper colouring G.

The chromatic number $\chi(G)$ is polynomial computable when $\chi(G) \leq 2$, but when $\chi(G) \geq 3$ the problem becomes NP-complete, even for graphs G with degree $\Delta(G) \geq 3$. As a consequence, there are many unanswered questions related to the colouring of a graph [5].

Following the line of exact algorithms and using maximal independent sets to compute the chromatic number, Beigel et. al. [1] established an $O(2.4151^n)$ time algorithm . Subsequently, Byskov [2] provided an $O(2.4023^n)$ time algorithm. Both algorithms have a top-down strategy using a combination of improved upper bounds on the number of maximal independent sets (MIS) of size at most k, in a dynamic programming approach.

We have designed a novel bottom-up heuristic for colouring graphs. Given an input graph G, first we remove even cycles and acyclic subgraphs since they can be 2-colourable. Taken the resulting graph, said G_1, the procedure iterates building in each iteration a MIS K_i and discarding it from G_i, forming so a subgraph $G_{i+1} = (G_i - K_i)$. The procedure iterates until a polynomial-time 2-colourable subgraph is reached.

The knowledge of lower bounds for the independence number of the graph ($\alpha(G)$) has been a relevant measure to determine combinatorial properties of a graph. In this paper, we show that $\alpha(G)$ is not the unique useful measure for computing $\chi(G)$. If G is a connected graph and K is a MIS of G, we establish the first lower bound on the maximum number of edges incident to the nodes of K, and we show how that lower bound establishes a new upper bound.

We build a MIS K_i for each subgraph G_i satisfying that the number of edges of G_i incident to nodes of K_i, is at least the number of current nodes minus 1, i.e. $|E_{G_i}(K_i)| \leq |V(G_i)| - 1$. That lower bound for $|E_{G_i}(K_i)|$ allows us to design an iterative procedure such that, if each remained subgraph $G_{i+1} = (G_i - K_i)$ is connected, then our procedure establishes an average number of $\lceil \delta(G)/2 \rceil + 2$ colours as the chromatic number of G, where $\delta(G)$ is the average degree of G.

2 Preliminaries

Let $G = (V, E)$ be an undirected simple graph (i.e. finite, loop-less and without multiple edges) with vertex set V and set of edges E. $E(G)$ and $V(G)$ emphasize that these are the edges and vertex sets of a particular graph G. Two vertices v and w are called *adjacent* if there is an edge $\{v, w\} \in E$, joining them. The *neighbourhood* of $x \in V$ is $N(x) = \{y \in V : \{x, y\} \in E\}$ and its *closed neighbourhood* is $N(x) \cup \{x\}$ which is denoted by $N[x]$.

We denote the cardinality of a set A, by $|A|$. Given a graph $G = (V, E)$, the degree of a vertex $x \in V$, denoted by $\delta(x)$, is $|N(x)|$. If A is a set of vertices from a graph G, $N(A)$ is the set of neighbour vertices from any vertex of A, that is, $N(A) = \cup_{x \in A} N(x)$, while $N[A] = N(A) \cup A$.

The maximum degree of G or just the degree of G is $\Delta(G) = max\{\delta(x) : x \in V\}$, while we denote with $\delta_{min}(G) = min\{\delta(x) : x \in V\}$ and with $\delta(G) = (2 \cdot |E|)/|V|$ the average degree of the graph.

Given a subset of vertices $S \subseteq V(G)$ the subgraph of G denoted by $G|S$ has vertex set S and a set of edges $E(G|S) = \{\{u, v\} \in E : u, v \in S\}$. $G|S$ is called the *subgraph of G induced by S*. We write $G - S$ to denote the graph $G|(V - S)$. The subgraph induced by $N(v)$ is denoted as $H(v) = G|N(v)$ which has to $N(v)$ as the set of nodes and all edges upon them.

Given a subgraph $H \subseteq G$ and for a vertex $x \in V(H)$, let $E_H(x) = \{\{x, u\} \in E(G) : u \in H\}$, and let $\delta_H(x)$ be the cardinality of $E_H(x)$, if $H = G$ then $\delta_G(x) = \delta(x)$. $N_H(x)$ denotes the set of nodes from H adjacent to x. For any subgraph $H \subseteq G$, $\delta_G(H) = \sum_{x \in H} \delta_G(x)$. If H is an independent set of G then $\delta_G(H)$ is the number of edges of G incident to any node of H.

A path from a vertex v to a vertex w in a graph is a sequence of edges: $v_0v_1, v_1v_2, \ldots, v_{n-1}v_n$ such that $v = v_0$, $v_n = w$, v_k is adjacent to v_{k+1} and the length of the path is n. A simple path is a path such that $v_0, v_1, \ldots, v_{n-1}, v_n$ are all distinct. A cycle is just a nonempty path such that the first and last vertices are identical, and a simple cycle is a cycle in which no vertex is repeated, except the first and last vertices.

A k-cycle is a cycle of length k, that is, a k-cycle has k edges. A cycle of odd length is called an odd cycle, while a cycle of even length is called an even cycle. A graph G is acyclic if it has not cycles.

A connected component of G is a maximal induced subgraph of G, that is, a connected subgraph which is not a proper subgraph of any other connected subgraph of G. Note that, in a connected component, for every pair of its vertices x, y, there is a path from x to y. If an acyclic graph is also connected, then it is called a free tree. Let G be a connected graph, a node $v \in V(G)$ is called a no articulation point if $G \backslash v$ is a connected graph. A subset $S \subset V(G)$ is called a no articulation set if $G \setminus S$ is a connected graph.

A colouring of a graph $G = (V, E)$ is an assignment of colours to its vertices. A colouring is *proper* if adjacent vertices always have different colours. A k-colouring of G is a mapping from V into the set $\{1, 2, \ldots, k\}$ of k "colours". The chromatic number of G denoted by $\chi(G)$ is the minimum value k such that G has a proper k-colouring. If $\chi(G) = k$, G is then said to be k-chromatic. The value $\chi(G)$ is polynomial computable when $\chi(G) \leq 2$, but when $\chi(G) \geq 3$, the problem becomes NP-complete, even for graphs G with degree $\Delta(G) \geq 3$.

Given a graph $G = (V, E)$, $S \subseteq V$ is an independent set in G if for whatever two vertices v_1, v_2 in S, $\{v_1, v_2\} \notin E$. Let $I(G)$ be the set of all independent sets of G. An independent set $S \in I(G)$ is *maximal*, abbreviated as MIS, if it is not a subset of any larger independent set and, it is *maximum* if it has the largest size among all independent sets in $I(G)$. The *independence number* $\alpha(G)$ is the cardinality of the maximum independent set of G.

Let $G = (V, E)$ be a graph, G is a *bipartite graph* if V can be partitioned into two subsets U_1 and U_2, called *partite sets*, such that every edge of G joins a vertex of U_1 and a vertex of U_2. If G is a k-chromatic graph, then it is possible to partition V into k independent sets $V_1, V_2, ..., V_k$, called *colour classes*, but it is not possible to partition V into $k-1$ independent sets.

3 An Approximate Algorithm for $\chi(G)$

Given an input connected graph $G = (V, E)$, let $n = |V|, m = |E|$ be the number of nodes and edges, respectively. A depth-first search (dfs) on G is applied starting the search with the node $v \in V$ of minimum degree, and selecting among different potential nodes to visit the node with minimum degree first and with minimum value in its label as a second criterion.

While the $dfs(G)$ is computed, a set I_B, which consists of nodes not part of odd cycle from G, can be computed in polynomial time on the size $(n+m)$ of G. We show that I_B is a bipartite subgraph of G, and then I_B can be coloured at the end of the colouring process by the two last colours used for the last bipartite subgraph from G (subprocedure 2-colouring).

If $\delta(G) = (2m/n) \leq 2$ then G has not intersected cycles and it can be coloured in linear time on the number of nodes. Otherwise, if $\delta(G)$ is close to n, e.g. $\delta(G) \geq n - 4$, the complement graph of G, denoted as $\overline{G}$, shows the different colour classes of G.

Let G_0 be the initial graph which satisfies $2 < \delta(G_0) < n-3$ and each node of G_0 is part of of odd basic cycles. Let $G_{i+1} = (G_i - K_i)$ be the remaining subgraph after the i-iteration of our procedure. Let us denote as δ_i to $\delta(G_i)$ the average degree of G_i, $n_i = |V(G_i)|$ and $m_i = |E(G_i)|$. In each iteration, the procedure builds a MIS K_{i+1} in the remaining subgraph G_{i+1}.

We show that our procedure builds a MIS K_i of G_i satisfying that if G_i is a connected graph, then K_i is a maximal independent set of G_i such that $\sum_{x \in K_i} \delta(x) \geq |V(G_i)| - 1$.

Theorem 1 *Let G be a connected graph, there exists a maximal independent set K of G such that $\sum_{x \in K} \delta(x) \geq |V(G)| - 1$*

Proof. The proof proceeds by induction on the number of nodes n of the graph. □

Thus, let G be a connected graph such that $|E(G)| \geq |V(G)| + 1$. The Algorithm 1, called $Build\_MIS(G)$, builds a MIS which satisfies Theorem 1.

Algorithm 1 BUILD_MIS(G)

Require: G a non directed graph
Ensure: K_0 is a MIS such that $\delta_G(K_0) \geq |V(G)| - 1$
  while $|E(G_i)| > |V(G_i)|$ **do** {Contraction process}
    choose a no articulation node $x \in G_i$
    push x to a stack V and remove x from G_i
    $G_{i+1} = G_i - \{x\}$
  end while
  Builds K_0 a MIS such that $\delta(K_0) \geq |V(G_{i+1})| - 1$
  repeat {Extending the MIS K_0}
    pop x from stack V
    if $N_{G_{j-1}}(x) \cap K_0 = \emptyset$ **then**
      $K_0 = K_0 \cup \{x\}$
    end if
  until stack is empty
  Returns K_0 {At this point $\delta_G(K_0) \geq |V(G)| - 1$}

We describe the general strategy of our proposal for colouring G, called *Seek_Chromatic_Number(G)* (Algorithm 2).
Firstly, in each main iteration of the loop in Algorithm 2, G_i is tested to be polynomial-time 2-colourable and in this case, the procedure finishes and a polynomial-time 2-colouring procedure is executed.
Secondly, a MIS K_i such that $\delta_{G_i}(K_i) \geq |V(G_i)| - 1$ is formed.
Thirdly, we colour the nodes in K_i with the current colour, and let $G_{i+1} = G_i - \{K_i\}$, and the process is repeated with G_{i+1}.

Algorithm 2 Seek_Chromatic_Number(G)

Require: G a non directed graph
Ensure: An approximate value for $\chi(G)$

```
k = 3; G=dfs(G)
I_B = {u ∈ V(G) : u is not part of any odd cycle of G
G = G − I_B
if G is Polynomial_3Colourable then
  Returns χ(G) is 3
end if
while is_bipartite(G)==false do {While there are odd cycles in G}
  T = Build_MIS(G)
  G = G − T
  k = k + 1{Updating for the next MIS}
end while
G = G ∪ I_B{returns the first bipartite component }
Call 2-colouring(G){At the end, the remaining graph is 2-colourable}
Returns χ(G) is k + 2
```

4 Complexity Analysis

Given a connected initial graph G, let $G_0 = (G - I_B)$ be the input graph without its intial bipartite component (I_B), $G_0 = (V, E)$ with $n = |V|$ and $m = |E|$. Let us assume that $m = t \cdot n, t > 1$, and that G_0 has intersected odd cycles, hence $m > n$.

Let T_i be the MIS formed in the iteration i of the loop of algorithm 2. Let $G_{i+1} = G_i - T_i$, $n_{i+1} = |V_{i+1}|, m_{i+1} = |E_{i+1}|$ and let $\delta_i = \frac{2m_i}{n_i}$ be the average degree of each subgraph G_i. In each iteration, the number of nodes and edges are updated as: $n_{i+1} = n_i - |T_i|$ and $m_{i+1} = m_i - |E_{G_i}(T_i)|$, since in each iteration the nodes in T_i are deleted as well as its incident edges: $E_{G_i}(T_i)$.

In each iteration algorithm 1 builds a MIS T_i of the current graph G_i such that $\sum_{x \in T_i} \delta_{G_i}(x) \geq (n_i - 1)$ under the assumptions that G_i is connected and $m_i \geq n_i$.

In the first iteration it holds: $\sum_{x \in T_1} \delta_G(x) \geq (n-1)$. In the second iteration $\sum_{x \in T_2} \delta_{G-T_1}(x) \geq n_1 - 1$ which is equivalent to $\sum_{x \in T_2} \delta_G(x) - |T_1| \geq (n - |T_1|) - 1$ since each node in T_2 was originally adjacent to some node in T_1, T_1 is the first MIS of G and $n_1 = n - |T_1|$. Thus $\sum_{x \in T_2} \delta_G(x) \geq n - 1$.

The same analysis holds for the third iterations $\sum_{x \in T_3} \delta_{G-(T_1 \cup T_2)}(x) \geq n_2 - 1$ which is equivalent to $\sum_{x \in T_3} \delta_G(x) - (|T_1| + |T_2|) \geq (n - |T_1| - |T_2|) - 1$, since each node in T_3 was originally adjacent to some node in T_1 and some node in T_2, $n_2 = n - |T_1| - |T_2|$. Thus, $\sum_{x \in T_3} \delta_G(x) \geq n - 1$.

The main cycle in algorithm 2 ends when the graph G_k is a bipartite graphs (2-coloring graphs). Thus, in the iteration k, it holds $\sum_{x \in T_k} \delta_G(x) \geq n - 1$. So,

$$\sum_{x \in \cup_{i=1}^{k} T_i} \delta_G(x) = \sum_{x \in V} \delta_G(x) = 2m \geq k \cdot (n-1)$$

since $T_i, i = 1, \ldots, k$ is a partition of V and the sum of the degree of the nodes of a connected graph is the double of the number of edges.

The last inequality establishes an order of $k \leq (2m)/(n-1)$ iterations for the while in algorithm 2. Then, $[(2m)/(n-1)]+1$ colours are enough (because of the two colours used in the last iteration) to colour the initial graph G, that is $\delta(1 + [1/(n-1)]) + 1$, or $\left\lceil \delta \cdot \frac{n}{n-1} \right\rceil + 1$ colours, $\delta = 2m/n$ being the average degree of the initial graph G_0.

Thus, if each G_{i+1} generated by our heuristic is connected, then $\lceil \delta(G_0) \rceil + 2$ colours are enough for colouring the input graph G, where $\delta(G_0)$ is the average degree of the input graph G without its first bipartite component.

Notice that the main purpose to consider $G_0 = (G - I_B)$ for starting the iterative procedure *Seek_Chromatic_Number* is to reduce the possibilities that G_i will be a disconnected subgraph, $i = 1, \ldots, k$. But, if during an iteration of our procedure G_i is disconnected, then $\chi(G_i) = max\{\chi(H_1), \ldots, \chi(H_t)\}$, where $H_i, \ldots, H_t$ are the different connected components from G_i and the number of colours for colouring G could not be upper bounded by $\lceil \delta(G_0) \rceil + 2$.

One of the most expensive time task included in *Build_MIS* is to recognize articulation points (or cut vertices) on the current subgraph, this task is done in time $O(m + n)$, and assuming $m > n$ (which are the cases when *Build_MIS* is executed) the total time for recognizing articulation points is $O(m) = O(2m)$.

The number of iterations of the step 1 of *Build_MIS* (which coincides with the number of iterations in the step 3) is at most $\lceil n/3 \rceil$ because at most $\lceil n/3 \rceil$ nodes can be removed from the original graph in order to form an acyclic graph. And to determine the articulation points in the step 1 is of order $O(m)$. And the second step (to build the initial MIS) requires at most time $O(n)$. Then, *Build_MIS* has a time complexity in the worst case of $O(n^2) = \lceil n/3 \rceil \cdot n$.

The most expensive step, with respect to the time complexity, of algorithm 2 is the "while" whose body has a time complexity of $O(n^2)$ because it consists of performing *Build_MIS*. The number of iterations in algorithm 2 is proportional to $\delta(G) \cdot \frac{n}{n-1}$, then in the worst case the total time of our procedure will be $n^2 \cdot \frac{2m}{n} \cdot \frac{n}{n-1} = 2\frac{n^2 \cdot m}{n-1} \approx 2 \cdot m \cdot n$. Thus, an upper bound for the time complexity of our procedure is $O(m \cdot n)$, which is a polynomial value on the size of the input graph G.

5 Conclusions

We have presented a novel polynomial-time algorithm for determining the chromatic number of a graph $\chi(G)$. Given an input connected graph G, our heuristic discards a first bipartite component of G, denoted by I_B, formed by the nodes which are no part of odd cycle in G since those nodes can be coloured at the end of the process with the first two basic colours. Let $G_0 = G - I_B$ be the remaining subgraph. Our proposal is based on selecting, in an iterative manner, a MIS K_i from the current subgraph G_i such that $\delta_{G_i}(K_i) \geq |V(G_i)| - 1$. That lower bound on the number of edges in the current graph G_i, with an endpoint in any node of K_i, allow us to design an iterative procedure such that if each remained subgraph $G_{i+1} = (G_i - K_i)$ is connected, then we obtain an upper bound to colour a graph; given that $\lceil \delta(G_0) \rceil + 2$ colours are enough to colour the input graph G, where $\delta(G_0)$ is the average degree of the initial subgraph without its first bipartite component.

On the other hand, if any G_i is disconnected then $\chi(G_i) = max\{\chi(H_1), \ldots, \chi(H_t)\}$ where $H_i, \ldots, H_t$ are the different connected components from G_i.

References

[1] Beigel R., and D. Eppstein, *3-colouring in time $O(1.3289^n)$*, Journal of Algorithms **54** (2) (2005) 168–204.

[2] Byskov J.M., "Exact Algorithms for graph colouring and exact satisifiability". Phd thesis, University of Aarbus, Denmark, 2005.

[3] De Ita Guillermo, and Javier A. Castillo *Recognizing 3-colorings cycle-patterns on graphs*, Pattern Recognition Letters **34** (4) (2013), 433–438.

[4] Golumbic M.C., "Algorithmic Graph Theory and Perfect Graphs". 2nd edn. North Holland, 2004.

[5] Kocay William, and Donald L. Kreher, " Graphs, Algorithms, and Optimization". Chapman & Hall/CRC, 2005.

[6] Wilson R. A., "Graphs, Colourings and the Four-colour Theorem". Oxford University Press, 2002.