The 7th Balkan Conference on Operational Research
“BACOR 05”
Constanta, May 2005, Romania

A HEURISTIC ALGORITHM FOR THE LIST COLORING OF A RANDOM GRAPH

MAYA SATRATZEMI
University of Macedonia, Dept. of Applied Informatics

CONSTANTIN TSOUROS
Aristotle University, Faculty of Technology

Abstract
Let G = (V, E) a graph and L (v_i) a set of colors associated to every node $v_i \in V$.
A list coloring of G is an assignment of a color c (v_i) $\in$ L (v_i) to every node of V so that no two adjacent nodes are assigned the same color.
Significant theoretical research into this special case of the classic coloring problem started roughly the last decade.
In the corresponding literature, algorithms are referred to for a particular class of graphs, e.g. trees. In this paper a heuristic algorithm is formulated in order to achieve a list coloring to a given random graph G with the smallest possible number of colors, whenever G has such a coloring.
A small numerical example of the presented algorithm is given and the paper is incorporated with a relative computational experiment.

Keywords*: Heuristic, Algorithm, List coloring, NP-complete*

1. INTRODUCTION

Graph theory is a convenient mathematical tool, which can be used to model, as visual representations, problems that arise in various scientific fields as well as in real life practical situations.

One of the most outstanding concepts in graph theory is the notion of graph coloring. The *coloring* of a graph $G = (V, E)$, see [1,2] is an assignment of colors to its nodes so that no two adjacent nodes have the same color. A coloring of G with k colors is a k-coloring. A subset S $\subseteq$ V is *independent* if every pair of nodes $x, y \in S$ is nonadjacent. The nodes that are assigned the same color are independent and they form a *color class*. The smallest k for which G has a k-coloring is the *chromatic number* $\chi(G)$. The k-*coloring* of an arbitrary graph G is *NP*-complete, see [3], while the determination of the chromatic number of G is *NP- hard*.

Various significant applications can be modeled and studied by applying some generalizations of the ordinary graph coloring described above.

Let L (v_i) be a set of colors associated to every node $v_i \in V$. A *list coloring* of *G* is an assignment of a color $c(v_i) \in L(v_i)$ to every node of *V* so that no two adjacent nodes are assigned the same color. Significant theoretical research into this special case of the ordinary coloring problem started roughly in the last decade, see [4].

Here, a heuristic algorithm that we call *LC* is developed in order to *list-color* a given random graph. Procedure *LC* is heuristic in the sense that it seeks to *list-color* G with the smallest possible number of colors. The next Section is devoted to a formal and algorithmic presentation of procedure *LC* while Section 3 gives a systematic application of *LC* on a small numerical example. A computational experiment relative to algorithm *LC*, together with the conclusions are the content of the final Section.

2. PROCEDURE *LC*

By G_W =($W \subseteq V$, $E_w \subseteq E$) we denote the subgraph of $G = (V, E)$ that is defined by $W \subseteq V$ and $E_w = \{ (x,y) \in E$ and $x,y \in W \}$. The degree of a node $v \in W$ is expressed by $\deg_w(v)$. Consider the graph $G = (V, E)$ with $n = |V|$ and the sets of admissible colors L_i so as to color node $v_i \in V$. In order to facilitate the computation and the algorithmic process a positive integer is associated to every color, thus, in the sequel a color is expressed by a natural number. Let $LL = \bigcup_{i=1}^{n} L_i$ and $Q_i = \{q_{i_1}, q_{i_2}, \ldots, q_{i_{r_i}}\}$ where q_{i_j} is such that $i \in L(q_{i_j})$, in other words set Q_i contains the nodes that can be colored with color *i*. In general terms the reasoning of *LC* is as follows.

At the beginning of the algorithmic process, we pose $G = G_W$ and set *LL* comprises all the admissible colors to be used. In an instant of the execution the colors already used are eliminated from *LL*. Also color *i* is removed from *LL* in the case where a color has been assigned to all the elements of Q_i.

α) The elements of every Q_i for which $Q_i \neq \varnothing$ are positioned in such a way that $\deg(q_{i_j}) \leq \deg(q_{i_{j+1}})$, j=1,2,… $q_{i_{r_i}}$ and subsequently we form heuristically an independent set $S_i \subseteq Q_i$ of the subgraph G_W with the largest possible number of nodes. Thus, for every such Q_i corresponds an independent set S_i.

Set S_m is selected so that $S_m = \max \{ |S_i|, i \in LL \}$ and color m is assigned to all the nodes of S_m.

The working subgraph G_W and sets Q_i are reduced by removing the nodes of S_m from their content, also set LL is updated by extracting color m and colors i for which the corresponding set Q_i ended up in the empty set. The procedure is repeated from point α until one of the two following conditions hold.

- When all nodes of G have been colored ($G_W = \varnothing$).
- The occurrence of condition $LL = \varnothing$ means that no solution was found.

The above thoughts are depicted in the systematic formal presentation of algorithm LC that follows.

```
{ ncol : number of colored nodes }
{   m  :  color to be assigned       }
{  $C_z$ :   color class                  }
{ KEEP: comprises the intermediate generated independent sets}
{   $S$     :  independent set with the greater cardinality in the current iteration}

   { Read data,initial conditions }
     Read $G = (V, E)$, $n = |V|$, $k$, $L_i$, $i$=1, 2, …, $n$.
     Set $G_W(V_W) \leftarrow G$ ,  $ncol \leftarrow z \leftarrow 0$
      Do
          For every $i \in LL$ create the elements of
             $Q_i = \{q_{i_1}, q_{i_2}, ...., q_{i_{r_i}}\}$ where $q_j$ such that $i \in L(q_j)$.
             The elements of $Q_i$ are ordered so that
             $\deg_{G_W}(q_{i_j}) \le \deg_{G_W}(q_{i_{j+1}})$, where $j = 1, 2, ...., q_{i_{r_i - 1}}$

    { Form independent set }
             set $mc \leftarrow 0$, $S = \varnothing$, $LT \leftarrow LL$..
             while $LT \neq \varnothing$
                    select $i \in LT$.
                    Set $k \leftarrow 1$, $KEEP \leftarrow \{q_{i_1}\}$
                    for $h$=2 to $|Q_i|$
                       if $q_{i_h} \notin \Gamma(KEEP)$ then
                      set k←k+1, $KEEP \leftarrow KEEP \cup \{ q_{i_h} \}$
                     endif
                    endfor
                       if $k > mc$ then
                      $mc \leftarrow k$, $S \leftarrow KEEP$, $m \leftarrow i$
                    endif
                    Set $LT \leftarrow LT - \{i\}$
          endwhile
  { *Color assignment, successful termination test* }
                    Set $z \leftarrow z+1$, $x_z \leftarrow m$, $nc_z \leftarrow |S|$, $C_z \leftarrow S$
                    {Assign color $m$ to the nodes of set $C_z$}
                    Set $ncol \leftarrow ncol+|S|$, $LL \leftarrow LL-\{m\}$
                    if $ncol = n$ then
                        write “number of colors =”,$z$
                        for $i$=1 to $z$
                            write $x_i$, $C_j$, j=1, 2, …, $nc_z$
                        endfor
                      Stop
                    endif

    { Update $LL$, $Q_i$, $i \in LL$, $G_w$ }
          Set $LT \leftarrow LL$,  $W = W - \{ S \}$
          while $LT \neq \emptyset$
                    select $i \in LT$
                    for $j$=1 to $|Q_i|$
                        set $Q_i \leftarrow Q_i - \{ S \}$
                    endfor
                    Set $LT \leftarrow LT-\{i\}$
                  if $Q_i = \emptyset$ then set $LL \leftarrow LL-\{i\}$
          endwhile
  { *unsuccessful termination test* }if  $LL = \emptyset$ then
          write “no solution found”
           Stop
         endif
     Loop
```

Obviously the time complexity of LC is $O(n^3)$.

3. NUMERICAL EXAMPLE

Consider the graph $G=(V,E)$, with $n = |V| = 6$ of Figure 1.
Without loss of the generality, the lists of admissible colors, in order to color the nodes of G have the same cardinality k = 3. Also the colors in the lists were selected from the range [1,n]. Therefore the starting number of rows of array Q is equal to n. On the right side of the page, array Q is shown initially as it was formed as a first step by removing the rows that:

- Correspond to the color used in the previous stage.
  (see row 2 and 1 in iterations 2 and 3 respectively)
- All its nodes have been colored in a preceding phase.
  (see row 5 in iteration 3)

After the removal of the rows according to the above description, the remaining elements of the active rows of array Q are ordered as mentioned in Section 2.

The number of colors used in the final solution, see Figure 4, corresponds to an optimal one, since the chromatic number of the graph of Figure 1 is $x(G) = 3$

Iteration 1

i	Q	Q	I.D
1	3 5	5 3	5 3
2	1 3 4 6	6 1 4 3	6 1 4 *
3	1 2 5 6	6 1 5 2	6 1
4	2 5 6	6 5 2	6 2
5	1 3 4	1 4 3	1 4
6	2 4	4 2	4

m = 2 , S_2 = { 6 , 1 , 4 }

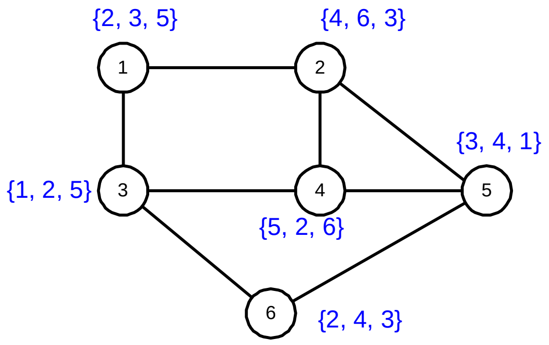

Figure 1

Iteration 2

1	3 5	5 3	3 5 *
3	5 2	2	2
4	5 2	2	2
5	3	3	3
6	2	2	2

m = 1 , S_1 = { 3 , 5 }

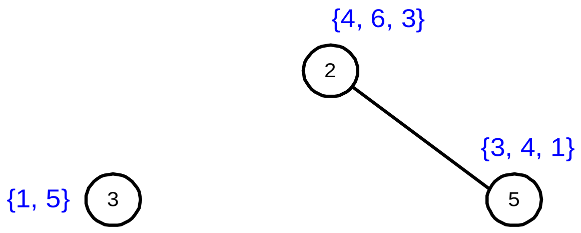

Figure 2

Iteration 3

3	2	2	2 *
4	2	2	2
6	2	2	2

m = 3 , S_3 = { 2 }

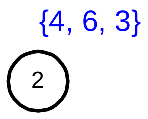

Figure 3

Final solution

$S_2 = \{ 6,1,4 \}$
$S_1 = \{ 3,5 \}$
$S_3 = \{ 2 \}$

$S_2 \bigcup S_1 \bigcup S_3 = V$

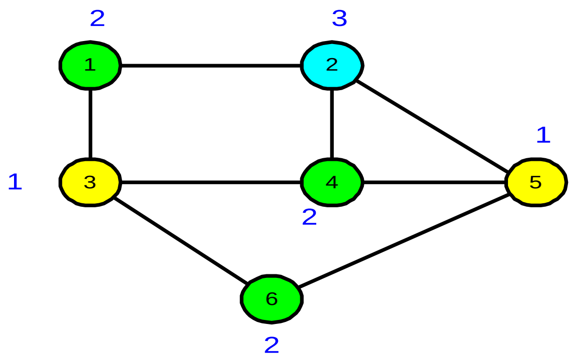

Figure 4

4. CONCLUSIONS-COMPUTATIONAL EXPERIMENT

A heuristic algorithm called *LC* is given for the list coloring concept which is a generalization of the classic coloring problem. The list coloring has significant practical applications to real life problems such as job scheduling, traffic phasing, allocation of broadcast channels, register assignment, etc, see [4,5,6,7].

Algorithm *LC* was coded in Quick-basic and executed on a Pentium IV with a 3.2 GHz processor. The CPU computer running time was negligible, for example, the graphs with 150 nodes required at most one second of execution time for all graph densities. All tested graphs and the node color lists were randomly generated with the use of the uniform distribution available in the Quick-basic version 4.5. The experiment was performed on graphs with 50,100 and 150 nodes and with densities 0.2,0.4,0.6 and 0.8. The graph density considered here is the value of the ratio of the number of edges of *G*, say e, over the number of edges of the complete graph, namely, $d=2e/(n(n-1))$. For all the nodes of the tested graphs we considered that the corresponding node lists have the same cardinality k and we experimented for values of k=2,3,4. The randomly generated colors in the node lists are in the range $[1,n]$,$[1, n/2]$ $[1, n/4]$ and the corresponding results are shown in Tables 1,2 and 3 respectively. From the tables below we can make the expected observation.

For greater value of list cardinality, a smaller number of colors (in general) are required to list color a graph. Also for smaller graph densities, a smaller number of colors is needed to list color a graph.

An interesting remark on the table results is the fact that fewer colors (in general) are needed whenever the range of the colors in the lists becomes tighter.

The above remark does not stand in two cases, see, Tables 2 and 3, specifically in the cases n=150, d= 0.8, k= 4 and n = 150, d=0.8, k=2. However, this is not an unexpected fact for *NP-complete* problems.

k=2					k=3					k=4				
n/d	0.2	0.4	0.6	0.8	*n/d*	0.2	0.4	0.6	0.8	*n/d*	0.2	0.4	0.6	0.8
50	23	27	35	37	50	19	24	28	35	50	16	23	25	34
100	52	60	69	78	100	41	47	64	68	100	34	40	51	60
150	75	85	103	116	150	57	76	87	106	150	51	64	72	98

Table 1. color range [1,n]

k=2					k=3					k=4				
n/d	0.2	0.4	0.6	0.8	*n/d*	0.2	0.4	0.6	0.8	*n/d*	0.2	0.4	0.6	0.8
50	20	26	31	38	50	20	22	25	34	50	14	18	26	34
100	41	54	65	76	100	30	43	55	64	100	29	35	49	59
150	65	82	97	115	150	53	66	76	104	150	41	56	72	90

Table 2. color range [1..n/2]

k=2					k=3					k=4				
n/d	0.2	0.4	0.6	0.8	*n/d*	0.2	0.4	0.6	0.8	*n/d*	0.2	0.4	0.6	0.8
50	15	26	30	35	50	14	19	26	33	50	12	20	25	30
100	40	54	61	74	100	27	43	50	63	100	25	30	43	58
150	59	72	96	117	150	43	63	76	101	150	34	54	70	93

Table 3. color range [1..n/4]

BIBLIOGRAPHY

[1] Christofides N.,(1975)*"Graph Theory-An Algorithmic Approach"*, Academic Press;

[2] Harary F.,*"Graph Theory"*, (1969) Addison-Wesley, Reading MA;

[3] Garey M.R. and Johnson D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979;

[4] Zsolt Tuza.,(1997) Discussiones Mathematicae , Graph Theory, pp 161-228;

[5] Thomas Zeitlhofer,Bernhard Wess(2003)., "List-coloring of interval graphs with application to register assignment for heterogeneous register-set architectures", Signal Processing v 83, Issue 7 pp 1411-1425;

[6] Kalfakakou R., Nikolakopoulou G., Savvidou E and Tsouros M., (2003) "*Graph Radiocoloring Concepts*" YUJOR, V 13**,** pp 207-215;

[7] Satratzemi M.,Tsouros M., (2004)"*A Heuristic Algorithm for the Set T-Coloring Problem*", 1st International Conference on Information & Communication Technologies: from theory to Applications-ICTTA'04.