PUC

Series: Monographs in Computer Science
and Computer Applications

Nº 1/72

FINDING THE OPTIMAL COLORINGS OF A GRAPH

by

Antonio Luz Furtado
Clésio S. dos Santos
Sergio Ivan Roschke
João Carlos P. Bauer

Computer Science Department - Rio Datacenter

Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

FINDING THE OPTIMAL COLORINGS OF A GRAPH

Antonio Luz Furtado
Associate Professor

Computer Science Department
PUC/RJ

and

Clésio S. dos Santos
Sergio Ivan Roschke
João Carlos P. Bauer

Faculty Members of The
Computer Science Department
PUC/RJ

This paper was submitted for publication elsewhere

Series Editor: Prof. A. L. Furtado February/1972

ABSTRACT

The problem of finding the chromatic number of a graph and exhibiting one or all optimal colorings has several practical applications.

It is equivalent to partitioning a set of objects, some of which are pairwise "incompatible", into the minimum number of cells, so that no two incompatible objects are assigned to the same cell.

Situations where this applies are production scheduling, construction of examination timetables, storage of goods, etc.

Heuristic procedures for the solution of the described problem have been developed by Berge [1] , Welsh and Powell [2], and Wood [3].

More recently Christofides [4] has presented a deterministic algorithm that is based on the concept of maximal internally stable sets.

This paper also employs this concept to suggest three different approaches:

a. a simple algorithm with relatively small storage requirements;

b. an integer linear programming formulation;

c. a branch and bound algorithm, that keeps storage requirements at a reasonable level and aims at efficiency by minimizing the number of steps.

We have used as sub-algorithm an efficient method by Bron-Kerbosch [5] originally developed for determining the cliques of a graph, but that can be easily adapted to obtain the maximal internally stable sets.

A number of examples were run on an IBM /360 model 40.

1. THEORETICAL FOUNDATIONS

Definition 1

The chromatic number r of a finite undirected graph is the minimum number of colors that should be used to color its nodes, so that no two adjacent nodes have the same color.

Definition 2

An optimal coloring is any assignment of r colors to the nodes of the graph in accordance with the above requirement.

Definition 3

A graph is uniquely colorable if it admits exactly one optimal coloring, up to interchange of colors.

Definition 4

An internally stable set of nodes is any subset of nodes of a graph such that no two nodes of the subset are adjacent.

Definition 5

A maximal internally stable set(MISS) is an internally stable set that is not properly contained in another internally stable set.

Definition 6

A complete sub-graph is a sub-graph whose nodes are all adjacent to every other node in the sub-graph.

Definition 7

A maximal complete sub-graph, also called a clique, is a complete sub-graph that is not properly contained in another complete sub-graph.

Definition 8

An intersection graph is a graph whose nodes represent sets, and two nodes are linked by an edge whenever the two corresponding sets have at least one element in common.

Let:

$G = (N, E)$ - a finite undirected graph of order n

r - the chromatic number of G

m - the number of MISS of G

$C = \{C_1, C_2, \ldots, C_r\}$ - an optimal coloring, where C_i denotes a subset of N assigned color i.

$M_{(t)} = \{M_1, M_2, \ldots, M_t\}$, $1 \leq t \leq m$ - a set of MISS, and

$L_{(t)}[M_i] = \bigcup(M_1, M_2, \ldots, M_{i-1}, M_{i+1}, \ldots, M_t)$

Lemma 1

To every C there corresponds at least one $M_{(r)}$.

Proof: For every C_i in C, $C_i \subseteq M_j$ holds for at least one M_j, since each color is an internally stable set and therefore is contained in at least one MISS. So $M_{(r)}$ can be constructed by extending each C_i to some corresponding M_j.
To see that all elements of $M_{(r)}$ are distinct, note that if the same M_j is obtained from C_i and C_k then the internally stable set $C' = C_i \cup C_k \subset M_j$ can replace C_i and C_k, and therefore C was not optimal.

From lemma 1 we conclude the existence of one or more sets $M_{(r)}$ for every optimal coloring.

However we wish to work in the opposite direction, to generate the optimal colorings from the sets of MISS. The next two lemmas show that N is not covered with $s < r$ MISS.

Lemma 2

Let s be the minimum number so that $M_{(s)}$ covers N. Then each element M_j of $M_{(s)}$ has at least one node that is not covered in $L_{(s)}[M_j]$.

Proof: If $M_j - L_{(s)}[M_j]$= empty

then $M_j \subset L_{(s)}[M_j]$ and $M_{(s)}$ is not minimal.

Lemma 3

If $M_{(s)}$ covers N then $s \geq r$.

Proof: Construct M' from each element of $M_{(s)}$ by taking

$M'_j = M_j - L_{(s)}[M_j]$.

From lemma 2 we see that the cardinality of M' is s.

Construct M" by assigning each node in $M_j \cap L_{(s)}[M_j]$ to exactly one among the M'_k whose corresponding M_k include the node.

By construction the cardinality of M" is still s. Since M" covers every node of N exactly once we conclude that it is a coloring and thus its cardinality s cannot be less than r.

Theorem 1

All optimal colorings of G can be obtained by taking each $M_{(r)}$ in turn, assigning the nodes that are only covered in one M_j to C_j, and making all possible assignments of the remaining nodes to each C_k whose corresponding M_k covers the node.

Note that the theorem does not exclude that the same optimal coloring can be generated by more than one $M_{(r)}$.

Corollary 1

m is an upper bound for r .

Corollary 2

A graph is uniquely colorable iff it admits only one $M_{(r)}$ and the MISS in $M_{(r)}$ are disjoint.

Corollary 3

Let $M_{(t)} \subset M_{(r)}$, $1 \leq t < r$; $\tilde{M}_i \notin M_{(t)}$;

$\tilde{M}_i - M_{(t)} \subset \tilde{M}_j - M_{(t)}$ for one or more $\tilde{M}_j$ such that

$\tilde{M}_j \notin M_{(t)}$ and $\tilde{M}_j - M_{(t)} \neq$ empty.

Then $\tilde{M}_i$ cannot belong to $M_{(r)}$.

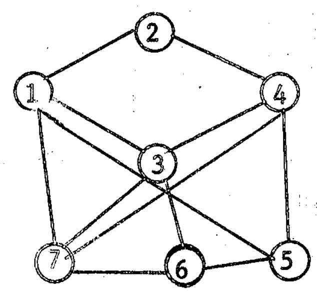
Proof: $M_{(r)}$ would certainly include one or more of the $\tilde{M}_j$ and if it also contained $\tilde{M}_i$ then $\tilde{M}_i \subset L_{(r)}$ $[\tilde{M}_i]$,an absurdity.

Corollary 4

If all nodes of a MISS M_j are covered in some other MISS and every one of these contains at least one node of M_j then there can be no $M_{(r)}$ including M_j .

Christofides' graph [4] given below is an example of the stuation described in corollary 4. It admits four MISS:

$M_1 = \{1,4,6\}$, $M_2 = \{2,3,5\}$, $M_3 = \{2,5,7\}$, $M_4 = \{2,6\}$; clearly M_4 fulfills the above conditions and can be said to be dominated by the other MISS.



Theorem 2

The chromatic number of a graph is equal to the largest chromatic number of its connected components.

Proof: Number arbitrarily the colors of each connected component.

Now let $C^* = \{C^*_1, C^*_2, \ldots, C^*_r\}$ be an optimal coloring of the component with largest chromatic number.

The nodes with colors $i = 1,2,\ldots,r_p$ for $1 \leq r_p \leq r$ of each component can certainly be added to the corresponding C^*_i, which finally yields one

$C = \{C_1, C_2, \ldots, C_r\}$.

Corollary 5

If a graph has cutnodes then an optimal coloring can be obtained from the optimal colorings of its components, each component being taken together with its cutnodes.

Proof: The only restriction to the scheme in the proof of theorem 2 is that a color C_i will include a given cutnode iff, among the colors contributing to C_i, all that come from components containing the cutnode do include the cutnode.

The decompositions provided by theorem 2 and its corollary may be convenient in view of storage and execution time considerations, as we shall see in section 5.

Fact 1 - The cliques of a graph are the MISS of its complement with respect to the complete graph of the same order, and conversely the MISS of a graph are the cliques of its complement (see [6] , page 36).

This allows us to use algorithms, originally developed for finding cliques, to the problem of finding MISS, either by suitably adapting the algorithms or by applying them to the complement of the given graph. Algorithms originally designed for finding MISS are given in [7] and [8] .

Fact 2 - The family of MISS of G, together with the relation M_i R M_j iff $M_i \cap M_j \neq$ empty define an intersection graph, which is the counterpart of the clique graph presented in [9] (page 20).

Fact 2 provides a way to represent the relations between the MISS of a graph, which is interesting from a theoretical viewpoint and may eventually disclose relevant characteristics in some particular cases (see section 5).

2. A SIMPLE ALGORITHM

This algorithm determines the chromatic number r and all $M_{(r)}$ of a graph. It assumes that the MISS of the graph have been previously obtained.

Let M, an n by m matrix, represent the MISS of G, with:

$m_{ij} = 1$ if node i is covered by MISS j , and

$m_{ij} = 0$ otherwise

Let X be a vector of m elements and B a vector of n elements.

step 1 - Set k to false.

step 2 - Set p to 1.

step 3 - Generate $K_{(p)} = \{K_1, K_2, \ldots, K_q\}$, $q = \binom{m}{p}$, the set of all combinations of the numbers 1 through m taking p numbers at a time.

step 4 - Set j to 1 .

step 5 - For every $1 \le i \le m$ set x_i equal to 1 if $i \in K_j$.

step 6 - Compute $B = M.X$ according to the usual rules for matrix by vector multiplication.

step 7 - For every $1 \le i \le n$ test if some $b_i = 0$; if so, go to step 10.

step 8 - Set k to true.

step 9 - Print X .

step 10 - Set j to $j + 1$; if $j \le q$ go to step 5.

step 11 - If k is true stop.

step 12 - Set p to $p + 1$ and go to step 3.

Indeed $b_u \ge 1$ iff $m_{uv} = 1$ (node u is covered in MISS v) and $x_v = 1$ (MISS v is included in the given combination) for one or more $1 \le v \le m$. If every $b_u \ge 1$ for $1 \le u \le n$ then all N is covered.

We look for the minimum value of $1 \le p \le m$ for which all N is covered. When this happens we know that $p = r$ (the chromatic number), and print X , noting that $x_i = 1$ means that MISS i is included.

The algorithm proceeds up to the end of iteration $p = r$, thus trying all combinations in $K_{(r)}$ and printing those that cover all N.

We see that the algorithm yields all $M_{(r)}$. It is not difficult to use theorem 1 to generate all optimal colorings from the $M_{(r)}$.

A simpler form of the algorithm consists of replacing multiplication by union, computing B by:

$$B = \bigcup_{v \in K_j} M_v$$

where M_v is a column of matrix M. Then we look for a combination K_j giving $B = 1_n$ where 1_n is a vector of n ones.

3. AN INTEGER LINEAR PROGRAMMING FORMULATION

This algorithm determines the chromatic number r and one $M_{(r)}$ of a graph, assuming that the MISS were previously obtained.

The matrix multiplication scheme in section 2 suggests the following formulation for the problem:

Let:

$$\sum_{j=1}^{m} m_{ij} \cdot x_j \geq 1 \quad , 1 \leq i \leq n$$

$$0 \leq x_j \leq 1 \quad , 1 \leq j \leq m$$

be a set of constraints, and

$$\sum_{j=1}^{m} x_j = \text{minimum}$$

the objective function.

The value of the objective function, which simply counts the number of MISS in a solution, will be the chromatic number. The subscripts of the nonzero elements of X will indicate the MISS that are in the solution. Normally the linear programming algorithms will stop as soon as one optimal solution is reached (meaning that only one $M_{(r)}$ is obtained).

Matrix M does not always satisfy the conditions (see [10], page 124) that guarantee, when using the simplex method, that an optimum solution with integer values for X will be reached.

However integer linear programming algorithms are available, and more specifically algorithms for the so-called set covering problems [11], of which this problem is a sub-class.

4. A BRANCH AND BOUND ALGORITHM

This algorithm determines the chromatic number r and one optimal coloring of a graph, assuming that its MISS were previously obtained.

We shall call $S^{k,\ell} = (M,V)$ a state where:

M is a set of MISS;

V is a set whose elements V_i are the number of MISS that cover node i, in the state.

State $S^{1,1}$ is formed in the preliminary phase when the MISS are determined (e.g. by the Bron-Kerbosch algorithm).

step 1 - Set k and z to 1.

step 2 - Set ℓ and w to 1.

step 3 - Search for a node i with minimum $V_i^{k,\ell}$.

step 4 - Form T with the elements of $M^{k,\ell}$ covering i, and $\bar{T}$ with the elements of $M^{k,\ell}$ not covering i; let the cardinality of T be p and the cardinality of $\bar{T}$ be q.

step 5 - Set v to 1.

step 6 - Form M' with:

a. $\bar{T}_j - T_v$, for all $1 \le j \le q$, if $\bar{T}_j \cap T_v \ne$ empty.

b. $T_j - T_v$, for all $v+1 \le j \le p$.

In other words, M' will contain all the MISS having nodes in common with the selected MISS T_v , such nodes in common being removed from them; note from the range of j in item b that already selected MISS from set T are not considered.

step 7 - Form $M^{k+1,w}$, with $\bar{T}_j$, for all $1 \le j \le q$, if $\bar{T}_j \cap T_v$ = empty.

MISS having no nodes in common with T_v are immediately placed in the new, state $M^{k+1,w}$. Some MISS from M' may be added to $M^{k+1,w}$,depending on the outcome of steps 8 and 9.

step 8 - Consider $M'' = M' \cup M^{k+1,w}$; let the cardinality of M' be s and the cardinality of M'' be t.

step 9 - Complete $M^{k+1,w}$ with M'_j , for all $1 \le j \le s$, if $M'_j \not\subset M''_x$ for all $1 \le x \le t$.

The symbol '$\not\subset$' means 'is not properly contained'; so the case $M'_j = M''_x$, which of course occurs for all M'_j , does not cause their rejection.

$M'_j \subset M''_x$ means that all nodes in M'_j are also covered in M''_x, which has at least one node more.

step 10 - If the cardinality of $M^{k+1,w}$ is 1, stop.

step 11 - Compute $V^{k+1,w}$ from $M^{k+1,w}$.

step 12 - Set v to $v+1$ and w to $w+1$.
If $v \leq p$ go to step 6 .

step 13 - Set ℓ to $\ell + 1$. If $\ell \leq z$ go to step 3.

step 14 - Set z to $w-1$.

step 15 - Set k to $k+1$ and go to step 2.

As in all branch and bound algorithms we do not want to try all possibilities, which means here to try all combinations of the m MISS $k = 1,2,\ldots,r$ MISS at a time.

The bound conditions are:

a. do not select a MISS T_v if the remaining MISS in the state do not enable us to cover one or more nodes not yet covered; this justifies steps 3 and 4;

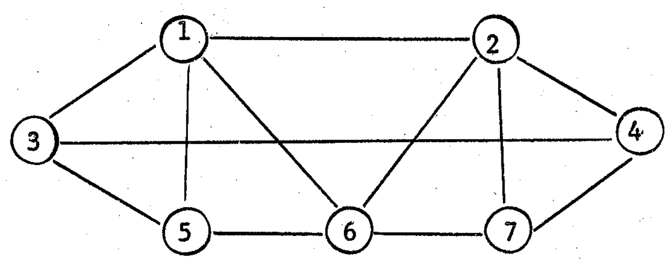
b. do not generate more than once the same combination of MISS, only altering their order of appearance; this is the reason for the range of j in item b of step 6;

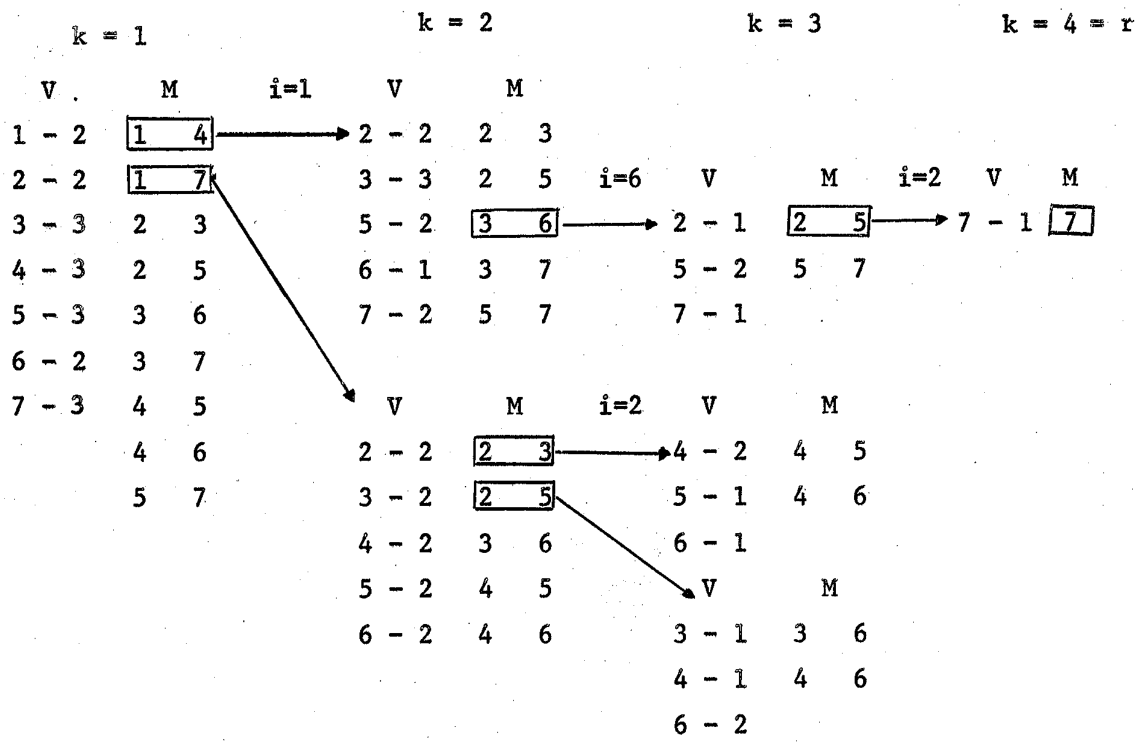
c. do not further consider for a sequence of MISS a MISS M_j such that $M_j - T_v \subset M_x - T_v$; see corollary 3 of section 1 and steps 6 through 9; it may be interesting to remark that $M^{k+1,w}$ is the set of MISS of the sub-graph obtained by removing the nodes in T_v from the set of nodes of the original graph (or containing sub-graph, if $k \neq 1$) , and that the M_j thus rejected are non-maximal internally stable sets of the sub-graph.

The algorithm stops (see step 10) with card $(M^{k+1,w}) = 1$ because, from bound condition a, this means that all the nodes not yet covered can be covered by the single remaining MISS. On the other hand if card $(M^{k+1,w}) > 1$ at least two more MISS have yet to be selected ; to see this recall that in every state all MISS are reduced to the nodes that were not covered so far, and that they are all maximal (MISS of a sub-graph).

A heuristic criterion that sometimes speeds up the process is to select the T_v in the decreasing order of their cardinality. In other words, from the MISS that cover node i choose first the MISS that covers the largest number of nodes.

Let us consider now an example. Let G be the graph:





We note that:

a. as opposed to the algorithms in sections 2 and 3, this algorithm produces one actual optimal coloring, instead of an $M_{(r)}$; in the example the optimal coloring is

C_1 = {1, 4} ,{3, 6} , {2, 5} ,{7} ;

b. if we go on with iteration $k = r$ some although not all other optimal colorings may eventually be produced; in the example these would be:

C_2 = {1, 7}, {2, 3} , {4, 5} , {6}

C_3 = {1, 7}, {2, 5} , {3, 6} , {4}

5. CONCLUDING REMARKS

A comparison of what the three algorithms do suggest the following mixed strategy;

a. use the algorithm of section 3 or the algorithm of section 4 to obtain the chromatic number r and one $M_{(r)}$ or one optimal coloring; if this is all that is needed in a given application, stop;

b. use the algorithm of section 2, but only to generate the combinations of m, r at a time (since r has been determined in a); this gives all $M_{(r)}$; again one may stop if this suffices, which may well be the case since the set of all $M_{(r)}$ indicates all possible optimal colorings;

c. if all possible optimal colorings in explicit form are required, do all possible assignments described in theorem 1 of section 1 .

The worst problem with algorithms dealing with MISS is that the number of MISS may grow exponentially with n, as it has been shown for cliques [12].

For example, in a graph with 39 nodes each three of which form a triangle, there are 3^{13} MISS. Fortunately we can resort to the partitioning (section 1, theorem 2) into 13 connected components; this leaves us with 3 x 13 MISS, noting besides that we shall only handle 3 of them at each time. If testing for graph equality (or , much worse, for graph isomorphism) could be done at a reasonable cost we could reduce this to the total of 3 MISS.

Corollary 5 provides a slightly more involved partitioning scheme.

A careful management of core storage, by using list structures and by freeing all parts of them that are no longer needed, is also mandatory for moderately large and for large graphs.

It is hoped that the representation of the MISS of a graph by means of an intersection graph will help to investigate more deeply the coloring problem. The intersection graph of the MISS from the graph of section 4 is:

$M_1 = 1 \; 4$

$M_2 = 1 \; 7$

$M_3 = 2 \; 3$

$M_4 = 2 \; 5$

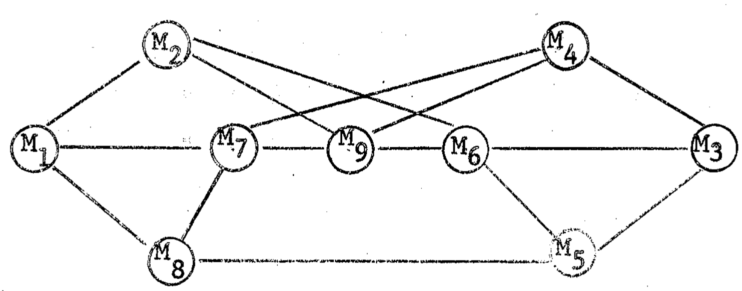
$M_5 = 3 \; 6$

$M_6 = 3 \; 7$

$M_7 = 4 \; 5$

$M_8 = 4 \; 6$

$M_9 = 5 \; 7$



It may be interesting to remark that one of the MISS of this new graph is formed by the nodes M_1, M_5, M_4, and another MISS by M_1, M_5, M_9; on the other hand M_1, M_5, M_4, M_9 were the MISS of the original graph that were used in the optimal coloring that we obtained through the algorithm from section 4 .

In case the MISS in an $M_{(r)}$ are all disjoint then:

a. $M_{(r)}$ is already an optimal coloring;

b. $M_{(r)}$ is one of the MISS of the intersection graph of the MISS from the original graph.

A second monograph to appear in this series will present the information structures and the PL/I programs to implement the algorithms of sections 2 and 4, as well as some measurements of core requirements and execution times.

6. REFERENCES

[1] Berge, C. - " The theory of Graphs" - Methuen, 1962.

[2] Welsh, D. J. A. and Powell, M. B. - "An upper bound to the chromatic number of a graph and its application to time-tabling problems"- The Computer Journal, vol. 10, p. 85, 1967.

[3] Wood, D. C. - "A technique for coloring a graph applicable to large scale timetabling problems" - The Computer Journal, vol 12, p. 317 , 1969.

[4] Christofides, N. - "An algorithm for the chromatic number of a graph" The Computer Journal, vol 14, p. 38, 1971.

[5] Bron, C. and Kerbosch, J. - "Finding all cliques of an undirected graph" - private communication.

[6] Berge, C. - "Théorie des graphes et ses applications" - Dunod, 1967.

[7] Berdnarek, A. and Taulbee, C. - "On maximal chains" - Revue Roumaine de Mathématiques Pures et Appliquées, vol 11, p. 23.

[8] Maghout, K. - "Applications de l'algèbre de Boole à la théorie des graphes et aux programmes linéaires et quadratiques" - Cahiers du Centre d'Etudes de Recherche Opérationnelle" - vol 5, p. 193.

[9] Harary, F. - "Graph theory" - Addison - Wesley, 1969.

[10] Hu, T.C. - "Integer programming and network flows" - Addison - Wesley, 1969.

[11] Lemke, C. E., Salkin, H.M. and Spielberg, K. - "Set covering by single - branch enumeration with linear programming subproblems" Opns. Res. 19, 998 - 1022, 1971.

[12] Moon, J.W. and Moser,L. - "On cliques in Graphs" - Israel J. Math, 3, p. 23, 1965.

Note: References 1,2,3,7 and 8 were taken from [4].