A new graph colouring algorithm

R. D. Dutton* and R. C. Brigham†

Many heuristic polynomial time algorithms exist to colour the vertices of connected undirected graphs having no loops or multiple edges. Associated with each such algorithm are graphs for which the number of colours required in the colouring produced by the algorithm is significantly greater than the minimum that are necessary. Thus it is desirable to have available a range of such algorithms. One, based on a theorem of Zykov, is described here. It has performed well on most graphs to which it has been applied, including some on which other algorithms fail. However, it has its own 'bad' graphs, and one such family is listed.

(Received June 1979)

We restrict ourselves to connected undirected graphs G which have no loops or multiple edges. A k-colouring of G is an assignment of one of k colours to each node of G so that connected nodes are assigned different colours. The chromatic number of G, $\chi(G)$, is the smallest k for which G is k-colourable. Graph colouring algorithms may be classified into (*a*) those which find a $\chi(G)$-colouring for any graph and (*b*) those which only guarantee a legitimate k-colouring for some k in the range $\chi(G) \leqslant k \leqslant n$, where n is the number of nodes of G. All known algorithms in the first class have execution times which are proportional to an exponential function of the number of nodes of G. There are, in fact, strong reasons for believing no polynomial time algorithm exists for obtaining a $\chi(G)$-colouring for arbitrary graphs (Gary and Johnson, 1976). Exponential algorithms are often impractical for even moderately sized graphs and, consequently, considerable effort has been devoted towards developing polynomial algorithms in the second class. There is a limit to the effectiveness of such algorithms. It has been shown that if there is no polynomial algorithm for the first class, then for every polynomial algorithm in the second class there exist graphs G for which the colouring obtained by the algorithm will be at least twice $\chi(G)$ (Gary and Johnson, 1976). In other words, if $A(G)$ is the number of colours which a polynomial algorithm A predicts is needed when applied to G, then there are graphs for which $A(G)/\chi(G) \geq 2$.

From this we conclude there probably is no universally 'good' polynomial time algorithm. It would therefore seem beneficial to have at our disposal several different algorithms along with some knowledge of the graphs for which each has problems. Among the simplest algorithms are the sequential colouring schemes and their various modifications (Matula, Marble and Isaacson, 1972). A more sophisticated approach is taken by Wood's algorithm (1969). Johnson (1974) and Mitchem (1976) have analysed these and designed classes of graphs for which each performs rather poorly. We present here yet another polynomial time colouring algorithm. We also construct a class of graphs for which $A(G)/\chi(G)$ becomes arbitrarily large.

Background

Basically the algorithm is a modification of an exponential time algorithm in Brualdi (1977) which was, in turn, suggested by a theorem due to Zykov. We first need the following definitions for arbitrary nonadjacent nodes x and y of a graph G.

Definition:
G/xy is the graph obtained from G by including the edge (x, y).

Definition:
$G{:}xy$ is the graph obtained from G by replacing x and y by a single node connected to all nodes adjacent to either x or y or both.

The latter operation is often called a merge, contraction, or identification of x and y. Zykov's result is the

Theorem
For arbitrary nonadjacent nodes x and y

$$\chi(G) = \min\{\chi(G/xy),\ \chi(G{:}xy)\}\ .$$

As a direct consequence of this theorem we have the

Lemma
For any graph G there exists at least one set of $n - \chi(G)$ pairs of nonadjacent nodes to which the merge operation may be successively applied to create a complete graph with $\chi(G)$ nodes.

The proof of this lemma is straightforward and will not be presented here. We offer as an example the 3-colourable graph in **Fig. 1**(*a*). Fig. 1(*b*) and 1(*c*) show the result of merging nodes 1 and 4 followed by 2 and 6. Fig. 1(*d*), 1(*e*) and 1(*f*) show a quite different result when merging is applied to the node pairs (3, 4), (2, 6) and (1, 5).

Intuitively, a good algorithm should select pairs of nodes for merging which will forestall the formation of a complete graph for as long as possible. This suggests, since every merge reduces

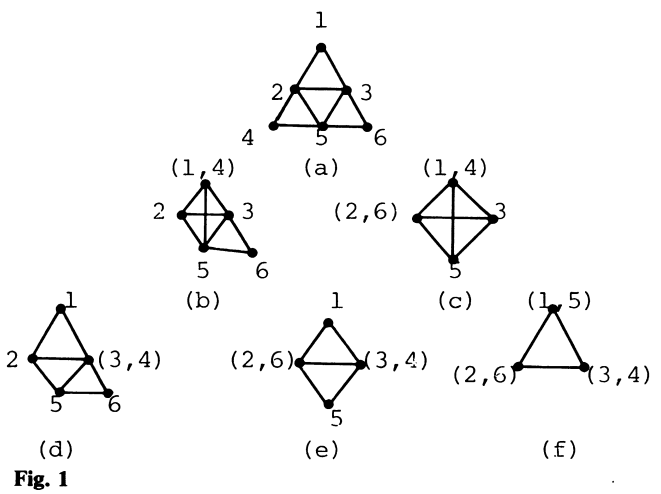

Fig. 1

*Department of Computer Science, University of Central Florida, Orlando, Florida 32816, USA.
†Department of Mathematics and Statistics, University of Central Florida, Orlando.

CCC-0010-4620/81/0024-0085 $01.00

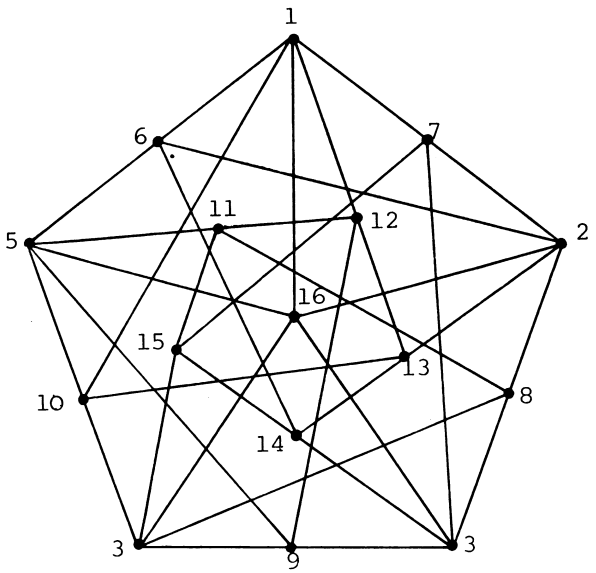

Fig. 2

the number of nodes by one, that we should select the pair which leaves as few edges as possible, i.e. removes as many edges as possible. This will tend to maximise the options available for the next merge. In a local sense this can be accomplished by merging the pair with the maximum number of common adjacent nodes.

The algorithm

The algorithm below assumes a graph G with an initial node set $\{v_1, v_2, \ldots, v_n\}$. Each iteration reduces the size of G by one node and at any point in time 'node' v_i represents not only the original node v_i but also all nodes which have been merged 'into' v_i, directly or indirectly, by previous iterations.

Step 1

For all nonadjacent nodes v_i and v_j compute c_{ij}, the number of common adjacent nodes.

Step 2

If no nonadjacent nodes remain, stop. Else determine the nonadjacent pair v_i and v_j for which $c_{ij} \geqslant c_{rs}$ for every other nonadjacent pair v_r and v_s.

Step 3

Merge v_i and v_j. Adjust the c_{rs} values for all affected nonadjacent pairs. Set $n \leftarrow n - 1$ and repeat from Step 2.

When the algorithm stops in Step 2 the current value of n is the estimate of the chromatic number of the graph and for $1 \leqslant i \leqslant n$ all nodes which were merged into node v_i may be assigned the i th colour. The time and space complexities are of order n^3 and n^2, respectively.

Several of the graphs found in Christofides (1975) were tested and produced minimal colourings. One of these, shown in **Fig. 2**, was stated to have a chromatic number 5.

Our algorithm identified the following four-colouring: {1, 2, 3, 4, 5}, {6, 11, 16}, {9, 13, 15}, {7, 8, 10, 12, 14}. Graphs which were specifically designed to thwart certain other colouring algorithms (Johnson, 1974) were also correctly coloured, e.g.

(1) $G_{m,2} = (V_m, E)$ where $V_m = \{a_i, b_i \text{ for } i = 1, \ldots m\}$ and $E = \{(a_i, b_j) \text{ for all } i \neq j\}$ and

(2) $G_{m,3}$ where $V_m = \{a_i, b_i, c_i, \text{ for } i = 1, \ldots, m\}$ and $E = \{(a_i, b_j), (a_i, c_j), (b_i, c_j) \text{ for all } i \neq j\}$

That the algorithm successfully coloured $G_{m,2}$ is not surprising since we can show it will perform correctly on any connected two-colourable (bipartite) graph. Informally, any connected bipartite graph with $n \geqslant 3$ has at least two nonadjacent nodes with one or more other nodes adjacent to both. The algorithm will select two such nodes, which will have the same colour in every two colouring of the graph, and merge them. Since they had the same colour in the original graph, the resulting graph will also be two-colourable, but with one less node. The algorithm will iterate until $n = 2$.

As expected, there are graphs for which the algorithm performs badly. Let $G_m = (V, E)$ have

$$V = \{a, b, x_i, y_i, z_{ij} \text{ for } i = 1, \ldots m \text{ and } j = 1, \ldots, m+1\}$$

and

$$E = \{(a, b)\} \cup \{(a, x_i), (b, y_i) \text{ for } i = 1, \ldots, m\} \cup \{(x_i, y_j) \text{ for } i \neq j\} \cup \{(x_i, z_{ij}), (y_i, z_{ij}) \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots m+1\}.$$

For $m \geqslant 2$ such graphs have $\chi(G_m) = 3$, but our algorithm estimates the chromatic number to be $m + 2$. These graphs also cause Wood's algorithm to perform badly, but are considerably less complex than those constructed by Mitchem and Johnson.

Conclusion

We have presented a colouring algorithm which can be included in the growing arsenal of such schemes. It performs well on a wide variety of graphs as well as minimally colouring any bipartite graph. We make no claims of superiority over any other colouring algorithm.

Acknowledgement

We wish to thank Mike Dotts for encoding and testing the algorithm.

References

Brualdi, R. A. (1977). *Introductory Combinatorics*, Elsevier North-Holland Inc., New York.

Christofides, N. (1975). *Graph Theory An Algorithmic Approach*, Academic Press, London.

Gary, M. R. and Johnson, D. S. (1976). The complexity of near-optimal graph coloring, *JACM*, Vol. 23 No. 1, pp. 43-49.

Johnson, D. S. (1974). Approximation algorithms for combinatorial problems, *J. Comptr. Syst. Sc.*, Vol. 9, pp. 256-278.

Matula, D. W., Marble, G. and Isaacson, J. (1972). Graph coloring algorithms, *Graph Theory and Computing* (R. Read, Ed.), Academic Press, New York, pp. 109-122.

Mitchem, J. (1976). On various algorithms for estimating the chromatic number of a graph, *The Computer Journal*, Vol. 19 No. 2, pp. 182-183.

Wood, D. C. (1969). A technique for coloring a graph applicable to large scale timetabling problems, *The Computer Journal*, Vol. 12 No. 4, pp. 317-319.