A technique for colouring a graph applicable to large scale timetabling problems

By D. C. Wood*

The connection is explained between scheduling problems and colouring the vertices of a graph such that no two adjacent vertices are the same colour, and the minimum number of colours are used. A method of colouring a graph suitable for large scale timetabling problems is described.

(Received November 1968)

1. Introduction

Many scheduling problems are concerned with the arrangement of events, each requiring the same duration of time, subject to the conditions that certain events may or may not take place concurrently. It is well known that such problems can be expressed in terms of graphs. The events are represented by the vertices of the graph, and a pair of vertices are joined by an undirected edge if and only if the corresponding events cannot take place at the same time. Scheduling the events subject to the constraints is therefore equivalent to colouring the corresponding graph such that no two adjacent vertices are the same colour. The determination of the minimum number of intervals of time needed for the schedule is therefore the same as finding the minimum number of colours required for the graph. This is known as the chromatic number of the graph, and its determination for an arbitrary graph is an unsolved problem. A more general account of the theory of graphs is contained for example in Berge (1962).

In the case of examination timetables, the events to be scheduled are the examinations, each requiring a period. The conditions can be represented by a conflict matrix $C = \{c_{ij}\}$, where $c_{ij} = 1$ if subjects i and j cannot take place concurrently because some candidate takes both subjects, and $c_{ij} = 0$ otherwise. The objective is the determination of the minimum number of periods required to accommodate as many as 500 examination papers.

2. Colouring by ordering vertices

The method widely used for this type of problem is to arrange the vertices in decreasing order of their degree, that is, the number of edges having the vertex as their endpoint. The first vertex starts the first colouring group. The vertices are inspected in order: any vertex which is not connected to a member of the first group is added to that group. The second group is started with the first vertex not yet coloured, and again the vertices are inspected in order: any vertex which is uncoloured and is not connected to a member of the second group is added to that group. The process continues until all the vertices have been coloured. This algorithm has been formally defined, for example, by Welsh and Powell (1967).

It is fairly clear that any vertex of degree less than the number of groups can always be added to one of the existing groups, since it cannot be connected to all of them. Therefore once the number of groups exceeds the degree of the first uncoloured vertex in the ordered list, the remaining vertices can always be coloured without introducing any further groups.

Whilst the above method is simple and usually produces acceptable results, it cannot in general be guaranteed to give the minimum number of groups. However, if, for example, a subset of n vertices can be located, every pair of which is connected, it can be proved that n is a lower bound. With small sets of data arising from timetabling problems, involving say 20 vertices, the lower and upper bound have been found to be fairly close, and by a process of trial and error the exact chromatic number can be determined.

However, when there are as many as 500 vertices, the lower and upper bounds can differ considerably, and one can only seek an improved method for colouring the graph.

3. Example

Consideration of the simple example in **Fig. 1** illustrates how the number of groups may be unnecessarily increased by the way the vertices are numbered.

Since all the vertices are of degree 2, they will be arranged in the order

$$1,\ 2,\ 3,\ 4,\ 5,\ 6.$$

Colouring the vertices in the above manner produces the following groups

$$(1,\ 2),\quad (3,\ 4),\quad (5,\ 6).$$

However, the vertices may equally well have been numbered differently and consequently arranged in the order

$$1,\ 5,\ 4,\ 2,\ 6,\ 3$$

in which case only two groups are obtained

$$(1,\ 4,\ 6),\quad (5,\ 2,\ 3).$$

It can be seen that once vertices 1 and 2 are grouped together, the number of groups is increased. How can such a step be avoided?

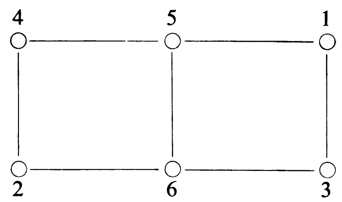

Fig. 1

* *Department of Applied Mathematics and Computer Science*, *University of Virginia*, *Charlottesville*, *Virginia* 22903, *U.S.A.*
Present address: *Mitre Corporation*, *McLean*, *Virginia* 22101, *U.S.A.*

4. Colouring by forming similarity matrix

The problem is analogous to the classification of objects by attributes, where a collection of objects has to be divided into clusters. A technique used in taxonomy is to form a similarity matrix, to indicate those pairs of objects of greatest similarity which should be put into the same group. In the same way a similarity matrix $S = \{s_{ij}\}$ is formed to determine which vertices should be the same colour. The similarity matrix found most effective is defined as

$$s_{ij} = 0 \text{ if } c_{ij} = 1$$

$$s_{ij} = \sum_k (c_{ik} \,\&\, c_{jk}) \text{ if } c_{ij} = 0$$

where & is the logical operation 'and' (**Table 1**), and k is summed over all vertices other than i and j. In other words, if i and j are not connected, the similarity is the number of other vertices k which are connected to both i and j.

Table 1
and

	0	1
0	0	0
1	0	1

The similarity matrix is scanned to find the greatest similarity. The first colouring group is started by a pair of vertices with the maximum similarity. Each pair of vertices with this similarity is then coloured according to the following algorithm. The similarity level is reduced by one, and again each pair of vertices with this similarity is coloured. The similarity matrix is scanned repeatedly, reducing the similarity level by one each time, until all vertices have been coloured. Vertices of degree less than the number of groups are left uncoloured, since they can always be added to one of the existing groups.

5. Colouring algorithm

The following procedure is adopted to colour a pair of vertices i, j, according to whether both, one or neither of the vertices are already coloured.

(*a*) Both i and j are coloured.

1. Go to next pair.

(*b*) One vertex, say i, is in colouring group G, and the other vertex, j, is uncoloured.

1. If the degree of j is less than the number of groups, then j can always be coloured and is ignored; go to next pair.
2. Try to add j to group G, i.e. if $c_{jk} = 0$ for each vertex k in group G, then add j to G; go to next pair.
3. Go to next pair if j cannot be added to group G.

(*c*) Neither i nor j is coloured.

1. If the degree of both i and j is less than the number of groups, they are ignored.
2. Find the first group G to which i and j can be added, i.e. $c_{ik} = 0$ and $c_{jk} = 0$ for each vertex k in group G.
3. If i and j cannot be added to an existing group, they become the first members of a new group.

6. Example

The conflict matrix and similarity matrix obtained from Fig. 1 are shown in **Table 2**. From the similarity matrix, it is evident that (1, 4, 6) and (2, 3, 5) have the greatest similarities. The above algorithm combines the vertices in this way, requiring two groups. Such a colouring is independent of the numbering of the vertices.

Table 2

Conflict Matrix

C_{ij}	1	2	3	4	5	6
1	0	0	1	0	1	0
2	0	0	0	1	0	1
3	1	0	0	0	0	1
4	0	1	0	0	1	0
5	1	0	0	1	0	0
6	0	1	1	0	0	0

Similarity Matrix

S_{ij}	1	2	3	4	5	6
1	0	0	0	1	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	0
4	1	0	0	0	0	1
5	0	1	1	0	0	0
6	1	0	0	1	0	0

7. Implementation

Since the store requirements for the conflict matrix for large sets of data are prohibitive, and the matrix consists mostly of zeros, a list is stored for each subject of those subjects with which it clashes. It would also be pointless to store the full similarity matrix since the elements are required in descending order of magnitude, and the majority of them are zero. Each similarity level is therefore stored as a chain list of the pairs of subjects with that similarity, thus avoiding the need to scan the matrix repeatedly.

With large examples, the lists for small similarities, e.g. less than five, would be enormous and of little value, so they are not recorded. Any subject still uncoloured when the similarity is reduced to this level can easily be coloured singly.

8. Application and conclusion

Welsh and Powell (1967) propose as an upper bound for the chromatic number

$$\max_i \; min\,(i, d_i + 1)$$

where d_i is the degree of the vertex i. The following three estimates of the chromatic number have been computed for various practical examples and are given in **Table 3**.

Table 3

Comparison of estimates of chromatic number

NUMBER OF SUBJECTS	α	β	δ
71	10	10	30
116	20	20	50
204	10	10	21
380	9	9	15
498	15	13	35
569	19	18	48

1. α = number of colours obtained by ordering the vertices.
2. β = number of colours used in the similarity matrix method.
3. δ = upper bound proposed by Welsh and Powell.

Although the numerical examples indicate the similarity matrix method is better than ordering the vertices in only two of the cases, exhaustive searching has verified that the chromatic number is obtained in every case. In all the examples, the upper bound of the chromatic number given by Welsh and Powell is of little value. The first two examples in Table 3 pertain to high school course requests and the last four to university examinations.

A more comprehensive comparison of the two methods has been made using randomly generated data. The random number generator is used to produce conflict matrices with a given probability of two subjects conflicting. Each matrix is coloured by both methods, and the percentage of matrices for which the similarity method is better (i.e. requires fewer groups), the same or worse is given in **Table 4**. Matrices of order 20, 50 and 100 are used with probabilities of conflicts of 0·25, 0·5 and 0·75. The range of the estimates of the chromatic numbers for each type of matrix is included in the table.

Two conclusions are apparent from Table 4.

1. The usual method of ordering the vertices does not give the best result in approximately 25% of cases, since the similarity matrix method obtains a better solution.
2. With small matrices, of order 20, there is little to choose between the methods. The difference between the estimates of the chromatic number rarely exceeded one. With large matrices, of order 100, the methods differ considerably. Ordering the vertices is better when the probability of a conflict is low (i.e. the conflict matrix is sparse); the similarity matrix method is better when the probability of conflicts is high.

Table 4

Comparison of estimates of chromatic number of random matrices obtained by similarity method and ordering vertices

NUMBER OF VERTICES	PROBABILITY OF CONFLICT	% FOR WHICH SIMILARITY METHOD IS			RANGE OF CHROMATIC NUMBER
		BETTER	SAME	WORSE	
20	0.25	21	61	18	3–5
20	0.5	26	54	20	5–8
20	0.75	21	67	12	8–10
50	0.25	8	66	26	6–8
50	0.5	30	52	18	10–13
50	0.75	48	31	21	16–20
100	0.25	7	20	73	10–13
100	0.5	13	50	37	18–22
100	0.75	72	20	8	28–33

References

BERGE, C. (1962). *The Theory of Graphs and Its Applications*, London: Methuen.

WELSH, D. J. A., and POWELL, M. B. (1967). An upper bound for the chromatic number of a graph and its application to timetabling problems, *The Computer Journal*, Vol. 10, p. 85.

WOOD, D. C. (1968). A system for computing university examination timetables, *The Computer Journal*, Vol. 11, p. 41.