Some Balanced Colouring Algorithms For Examination Timetabling

GHAZALI BIN SULONG
Fakulti Sains Komputer & Sistem Maklumat,
Universiti Teknologi Malaysia,
Jalan Semarak, 54100
Kuala Lumpur.

ABSTRACT

This paper describes some balanced colouring algorithms designed to construct examination schedules in such a way that : (1) all examination take place within a minimum number of days; (2) students are never scheduled to take two examinations at the same time; (3) the number of courses are scheduled into each period are approximately equal. These algorithms were tested on a large range of random graphs.

Keywords: Balanced colouring, scheduling, random graphs, graph colouring.

INTRODUCTION

Examination timetabling is becoming an important operation management problem in universities. The complexity of the problem depends largely on the amount of freedom students have in the choice of courses. In general, greater freedom of choice increases the difficulty of producing an examination schedule to fit into a limited time interval (period) without creating timetable conflicts for some students. As well as minimizing the number of periods and avoiding conflicts, timetable must be built to respect room capacities, and to space out examinations as evenly as possible for most students. There may also be other criteria or restrictions have to be taken into account and sometimes they are more important than the minimization of the number of periods [1, 2].

Here we will consider one particular constraint, namely that of so-called 'balanced scheduling (balanced colouring)'. Loosely speaking in its simple form a balanced scheduling, is one in which the numbers of courses are scheduled into each period are approximately equal.

The problem may be formulated in terms of graph colouring [3] as follows: Let there be n courses to be scheduled into k periods. Let the n courses to be vertices $V_1, V_2, .., V_n$ of a graph G = (V, U) and let the occurence of a student registered in courses V_i and V_j be represented by an edge joining the vertex pair (V_i, V_j). Then the scheduling of the n courses into k periods is analogous to partitioning the vertices of the graph into no more than k disjoint sets $S_1, S_2,..,S_k$ such that in any given set, there is no edge joining any pair of vertices.

Let

$$\Delta = \max_{i \leq k} |S_i| - \min_{i \leq k} |S_i|$$

In many circumstances we would try to find a colouring for which the quantity Δ is as small as possible.

We suppose throughout this paper that the vertices of G are ordered by some priority rule and are in the order $V_1, V_2, .., V_n$.

METHODS OF SOLUTION

Balanced Method 1

A balanced colouring algorithm was proposed by Brelaz et. al. [4] using a vertex-by-vertex sequential colouring algorithm. The method proceeds as folows:

Step 1 V_1 is assigned colour 1.

Step 2 Each of the remaining vertices is sequantially coloured so that a vertex V_i (i = 2,3,..,n) is coloured with a colour j which has been used the smallest number of times among the feasible colours, or with a new colour if necessary.

Computational experiments have been carried out by Brelaz et. al. on 55 random graphs having from 5 to 100 vertices with this method incorporated into Welsh & Powell's vertex ordering [5] and Matula's vertex ordering [6] respectively. The results revealed that with these methods the quantity Δ decreased considerably in comparison with the corresponding unbalanced methods.

Balanced Method 2

The above balanced colouring technique can also be incorporated into the sequential colouring algorithm based on the saturation degree of a vertex described by Brelaz et. al. The method proceeds as follows: Let V_1 be the vertex which has the highest degree.

Step 1 Vertex V_1 is assigned colour 1.

Step 2 Calculate the saturation degree of all the remaining incoloured vertices.

Step 3 Choose a vertex which has the highest saturation degree. If there is more than one such vertex, choose the one with the vertex degree.

Step 4 Colour the chosen vertex V_i with the colour j which has been used the smallest number of times among the feasible colours.

Step 5 If all the vertices are coloured, stop. Otherwise go to step 2.

Since the usefulness of the above balanced methods are limited to vertex-by-vertex sequential colouring methods, we now modify the balanced colouring technique to incorporate it into the colour-by-colour method by Dunstan [7]. The method proceed as follows:

Balanced Method 3

Initially, colour all the vertices of G using Dunstan's method with Williams's vertex ordering [8]; let k be the number of colours needed for colouring the graph. Let m_j denote the number of vertices coloured with colour j.

Step 1 Set u = {n/k}; this is an upper bound for the number of vertices that can be coloured with a single colour. ({x} is the smallest integer ≥ x).

Step 2 Vertex V_1 is assigned colour 1. Set $m_1 = 1$, and j = 1.

Step 3 Each of the remaining uncoloured vertices V_i (i = 2. 3...,n) is sequentially coloured with colour j if $m_j < u$, provided V_i is not adjacent to any vertex already given that colour. Set $m_j = m_j + 1$.

Step 4 When all the uncoloured vertices have been examined, those vertices coloured with colour j are deleted, together with all incident edges. The remaining vertices are reordered by the same rule, and go to step 5.

Step 5 If all the vertices are coloured, stop.

Otherwise set j = j + 1, $m_j = 0$, and return top step 3.

Since the number of colours used may exceed k, the value of u may not, in the end, represent the ideal number for a well-balanced colouring.

COMPUTATIONAL RESULTS AND DISCUSSION

All the balanced colouring algorithms were tested on a large range of random graphs having from 25 to 300 vertices in which each pair of the vertices is joined by an edge with a probability ranging in steps of 0.1 from 0.1 to 0.9. For each value of the probability six graphs were taken. The value of Δ and the number of colours were then recorded on every occasion. The results were then averaged and were tabulated in Table 1. The results from the related unbalanced methods were also recorded. Due to lack of space we cannot present all the results. We give those for 100, 200 and 300 vertices, these being the most numerous and hence the most reliable statistically. The pattern for larger graphs was almost identical; the differences were correspondingly greater. All the algorithms were coded in FORTRAN77.

The random graphs were computer generated utilizing pseudo random numbers from a Uniform distribution between 0 and 1. For each possible edge, if the number generated was less than the desired probability, the egde was included.

The following notation is used in Table I below.

- n = the number of vertices of the graph.
- p = the probability that each pair of vertices of the graph is joined by an edge. This value is called as a density of the graph.
- B1 = balanced method 1 with Welsh and Powell's vertex ordering technique.
- B2 = balanced method 2.
- B3 = balanced method 3 with Williams's vertex ordering technique.
- UB1 = the ordinary vertex-by-vertex sequential colouring algorithm with Welsh and Powell's vertex ordering technique.
- UB2 = unbalanced method by Brelaz.
- UB3 = unbalanced method by Dunstan with Williams's vertex ordering techique.
- k = the number of colours used by the above methods.

Each of the unbalanced methods UBi (i = 1, 2, 3) is individually associated with the balanced methods Bi (i = 1, 2, 3).

It is much easier discuss the result in table I by classifying the random graph according to their density in three families : i) lower density: p = 0.1,0.2,0.3 and 0.4, ii) medium density: p = 0.5, 0.6 and 0.7, iii) high density: p = 0.8 and 0.9.

We notice that with balanced methods, the value of Δ has considerably decreased as is shown in Table I, especially for the random graphs with low and medium densities. Of these methods, for the graphs with low density the values of Δ obtained by method B1 were the smallest, followed by methods B2 and B3. For medium density graphs; the values of Δ are in ascending order for method B3, B1 and B2 respectively. Finally for high density graphs; methods B3 produced the smallest values of Δ, followed by method B1 and B2.

Furthermore, the differences between the value of Δ obtained by the balanced methods and the value obtained by the corresponding unbalanced methods decreases as the density of the graphs increases, especially for the high density graphs. This is due to the fact as the density of the graphs increases, most of the vertices have high degree and so are difficult to colour. The only way out is by colouring the vertices with a new colour, thus inceasing the number of colours used.

The number of colours required by all balanced methods inreased in comparison with the related unbalanced methods.

For most of the methods the requirements of trying to minimize both Δ and the number of colours used are conflicting; typically reducing one leads to an increase in the other. It would be interesting to combine these two quantities into a single measure to reflect both aspects of the solution. An obvious type of measure is

n = 100

Table I

The average values of Δ

p	B1	B2	B3	UB1	UB2	UB3
1.	1.00	1.50	3.50	18.00	18.83	14.50
	(7.50)	(7.00)	(6.33)	(6.50)	(6.17)	(5.67)
	[8.50]	[8.50]	[9.83]	[24.50]	[25.00]	[20.17]
2.	1.83	2.50	8.33	12.83	13.00	11.17
	(10.67)	(10.67)	(9.67)	(9.50)	(9.83)	(8.83)
	[12.50]	[13.17]	[18.00]	[22.33]	[22.83]	[20.00]
3.	2.50	3.83	5.17	8.83	9.83	5.83
	(15.33)	(14.17)	(12.67)	(13.00)	(13.33)	(11.83)
	[17.83]	[18.00]	[17.84]	[21.83]	[23.13]	[17.66]
4.	2.83	4.00	3.50	6.67	7.33	4.33
	(18.33)	(17.83)	(15.67)	(16.17)	(16.50)	(14.83)
	[21.16]	[21.83]	[19.17]	[22.84]	[23.83]	[19.16]
5.	2.67	3.33	2.17	5.83	6.33	4.17
	(22.00)	(22.67)	(18.50)	(20.67)	(20.33)	(18.50)
	[24.67]	[26.00]	[20.67]	[26.50]	[26.66]	[22.67]
6.	2.50	3.33	2.83	4.67	5.33	3.50
	(25.67)	(27.00)	(22.83)	(23.67)	(24.67)	(22.17)
	[28.17]	[30.33]	[25.66]	[28.34]	[30.00]	[25.67]
7.	2.17	2.83	1.33	3.67	4.00	3.67
	(30.00)	(31.00)	(27.50)	(28.17)	(29.17)	(26.67)
	[32.17]	[33.83]	[28.83]	[31.84]	[33.17]	[30.34]
8.	2.33	2.33	1.33	3.67	3.67	2.50
	(36.00)	(37.67)	(34.50)	(34.83)	(36.33)	(32.83)
	[38.33]	[40.00]	[35.83]	[38.50]	[40.00]	[35.33]
9.	1.83	2.17	1.00	2.50	2.50	2.17
	(46.17)	(45.83)	(45.17)	(43.83)	(44.50)	(42.00)
	[48.00]	[48.00]	[46.17]	[46.33]	[47.00]	[44.17]

Notes:

1) The average number of colours is given in parentheses.
2) The average values of (k + Δ) is given in '[ ]'.

n = 200

Table I (continuation)

The average values of Δ

p	B1	B2	B3	UB1	UB2	UB3
.1	2.83 (11.50) [14.33]	2.50 (10.83) [13.33]	20.00 (10.00) [30.00]	24.50 (9.67) [34.17]	28.67 (10.00) [38.67]	24.50 (9.00) [33.50]
2.	1.83 (17.83) [19.66]	4.67 (18.00) [22.67]	8.50 (15.00) [23.50]	15.33 (15.00) [30.33]	16.67 (15.83) [32.50]	9.83 (14.00) [23.83]
3.	4.00 (25.00) [29.00]	5.00 (24.67) [29.67]	5.83 (20.50) [26.33]	10.67 (21.50) [32.17]	11.83 (21.23) [33.63]	9.00 (11.83) [29.17]
4.	3.33 (30.67) [34.00]	4.33 (30.67) [35.00]	4.17 (26.33) [30.50]	8.67 (27.00) [35.67]	9.00 (27.67) [36.67]	7.00 (25.50) [32.50]
5.	4.00 (37.50) [41.50]	4.33 (37.67) [42.00]	4.17 (32.17) [36.34]	6.33 (33.67) [40.00]	7.67 (34.83) [41.50]	5.17 (31.50) [36.67]
6.	3.83 (45.50) [49.33]	4.00 (45.17) [49.17]	3.17 (39.33) [42.50]	5.83 (41.33) [47.16]	5.83 (42.00) [47.83]	4.00 (38.33) [42.33]
7.	3.00 (52.83) [55.83]	3.17 (54.33) [57.50]	2.00 (46.50) [48.50]	4.83 (50.00) [54.83]	4.50 (50.50) [55.00]	3.17 (47.17) [50.34]
8.	2.67 (65.00) [67.67]	3.00 (64.83) [67.83]	1.83 (58.00) [59.83]	4.00 (61.50) [65.50]	4.17 (63.00) [67.17]	3.00 (58.50) [61.50]
9.	2.17 (83.17) [85.34]	2.33 (83.33) [85.66]	1.33 (77.00) [78.33]	3.00 (78.67) [81.67]	3.00 (79.00) [82.00]	2.17 (75.67) [77.84]

Notes:

1) The average number of colours is given in parentheses.
2) The average values of (k + Δ) is given in '[ ]'.

n = 300

Table I (continuation)

The average values of Δ

p	B1	B2	B3	UB1	UB2	UB3
1.	1.83 (14.50) [16.33]	4.00 (14.17) [18.17]	7.33 (12.17) [19.50]	27.50 (12.50) [40.00]	31.50 (12.67) [44.17]	21.33 (11.17) [32.50]
2.	4.00 (23.83) [27.83]	4.33 (24.00) [28.33]	11.33 (19.83) [31.16]	18.33 (20.67) [38.00]	18.33 (21.33) [39.66]	18.33 (18.67) [27.00]
3.	4.53 (33.33) [37.86]	6.33 (34.00) [40.33]	7.17 (27.00) [34.17]	12.33 (28.83) [41.16]	13.67 (29.83) [43.50]	9.83 (26.83) [36.66]
4.	4.33 (43.17) [47.50]	5.33 (43.17) [48.50]	3.50 (35.67) [39.17]	9.33 (37.17) [46.50]	10.33 (38.50) [48.83]	7.33 (34.33) [41.66]
5.	4.33 (52.33) [56.66]	4.33 (52.17) [56.50]	5.17 (44.17) [49.34]	7.17 (46.17) [53.34]	8.17 (46.50) [54.67]	4.67 (43.67) [48.34]
6.	3.83 (63.17) [67.00]	4.67 (63.33) [68.00]	2.83 (55.17) [58.00]	6.50 (56.17) [62.67]	6.50 (58.00) [64.50]	3.67 (53.83) [57.50]
7.	3.00 (75.50) [78.50]	3.17 (78.33) [81.50]	2.83 (67.50) [70.33]	4.83 (69.17) [74.00]	5.17 (70.67) [75.84]	4.50 (66.50) [71.00]
8.	2.67 (91.67) [94.34]	3.33 (94.67) [98.00]	2.17 (83.33) [85.50]	4.00 (85.33) [89.33]	4.00 (86.50) [90.50]	2.67 (82.00) [84.67]
9.	2.83 (117.50) [120.33]	3.00 (118.33) [121.33]	1.50 (108.50) [110.00]	3.00 (111.00) [114.00]	3.00 (111.83) [114.83]	2.33 (107.00) [109.33]

Notes:

1) The average number of colours is given in parentheses.
2) The average values of (k + Δ) is given in '[ ]'.

where k is the number of colours used and B is some weighting factor deemed suitable in given circumstances. To illustrate this B was taken to be 1, so that k + Δ was calculated. The average value of this, for graphs of a given type, is shown in '[ ]' in table I.

It is clear that when combining these two aspects in this way, balanced method 3 has the smalllest values compared to others (except for the graphs with $p = 0.1$ and 0.2, where the values obtained by balanced method 1 were smaller than that of balanced method 3). Hence, if the constraints of the measure of imbalance Δ and the number of colours are imposed, the proposed balanced method 3 is superoir to others.

A theoretical result on balanced colouring was given by Hajnal and Szemeredi [9]. They have stated that a graph $G = (V, U)$ with vertices of degree $d(V_1), d(V_2), .., d(V_n)$ has $\Delta \leq 1$ if $k > \max\{d(V_i)\} + 1$, where $i = 1, 2, .., n$ and k is the minimum number of colours needed for colouring the graph. Loosely speaking, in order to obtain a colouring which has $\Delta \leq 1$ one has to colour the graph using at least $\max\{d(V_i)\} + 2$ colours. This is clearly true as is shown in Table I where the values of Δ decreased as the number of colours increased. Hence, obtaining a colouring which requires a minimum number of colours and with $\Delta \leq 1$ is an intractable problem.

REFERENCES

1. Brelaz, D. Nicolier, Y. and Werra, D. de, Compactness and balancing in scheduling, *Zeitschrift fur Operations Research*, 21, 65-73, 1977.
2. Dunstan, F.D.J., Sequential colourings of graphs, *Proc. of the 5th. British combinatorial conference*, Aberdeen, 151-158, 1976.
3. Ghazali bin Sulong, A new graph colouring algorithm for large scale scheduling problems, *Malaysian Journal of Computer Science*, 5, 24-36, 1989.
4. Hajnal, A. and Szemeredi, E., *Proof of a conjecture of Erdos. Combinatorial theory and its applications*, II Erdos, P. Renyi, A., and Sos, V.T. (ed.), North-Holland, Amsterdam, 601-623, 1970.
5. Laporte, G. and Desroches, S., Examination timetabling by computer, *Computers and Operations Research*, 11, 351-360, 1984.
6. Matula, D.M. and Beck, L.L., Smallest-last ordering and clustering and graph colouring algorithms. *Journal of the ACM*, 30, 417-427, 1983.
7. Williams, M.R., The colouring of very large graphs, *Proc. of the Calgary International Conference on combinatorial structures & their applications*, New York, 477-478, 1970.
8. White, G.M. and Chan, P.W., Towards the construction of opitimal examination schedules, *Canadian Journal of Operational Research and Information Processing*, 17, 219-229, 1979.
9. Welsh, D.J.A and Powell, M.B., An upper bound for the chromatic number of a graph and its application to timetabling problems, *The Computer Journal*, 10, 85-86, 1967.