graph, and an “optimal” timetable. If a colouring that only uses half the allocated exam periods is found then it is very likely to be the cause of numerous second order conflicts and therefore not a good timetable. We must, when searching for scheduling methods, assume that the problem is going to be heavily constrained, otherwise our procedures will not work when put up against a conflict-graph with significantly more conflicts. There are two obvious ways of overcoming this problem. Firstly, knowing how many exam periods are available, we may deliberately aim to use that number of colours. Alternatively, we may use the efficiently scheduled timetable as a basis for the final one, using some search mechanism to spread out the exams taking into account other constraints as it goes. This we consider to be the better of the two options and intend to use a genetic algorithm to optimise general criteria under the guidance of the timetabler. This will allow the best possible timetable to be found in a minimum of time.

## 6. References


[1] BURKE E.K., ELLIMAN D.G. and WEARE R.F. (1993), A University Timetabling System Based on Graph Colouring and Constraint Manipulation, *To appear in the Journal of Research on Computing in Education*.

[2] BURKE E.K., ELLIMAN D.G. and WEARE R.F. (1993), Automated Scheduling of University Exams, *Proceedings of I.E.E. Colloquium on “Resource Scheduling for Large Scale Planning Systems”*, Digest No. 1993/144.

[3] CARTER M.W. (1986) A Survey of Practical Applications of Examination Timetabling Algorithms *OR Practice* 34, 193-202.

[4] de WERRA D. (1985) An Introduction to Timetabling *Eur. Jrnl. Op. Res.* 19, 151-162.

[5] DUTTON R.D. and BRIGHAM R.C. (1981) A New Graph Coloring Algorithm *Comp. Jrnl.* 24, 85-86.

[6] KARP R.M. (1972) Reducibility among Combinatorial Problems in *Complexity of Computer Computations* Plenum Press, New York.

[7] LOFTI V. and CERVENY R. (1991) A Final-Exam-Scheduling Package *J.Opl. Res. Soc.* 42, 205-216.

[8] SABIN G.C.W. and WINTER G.K. (1986) The Impact of Automated Timetabling on Universities - A Case Study *Jrnl. Opl. Res. Soc.* 37, 689-693.

[9] 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 *Comp. Jrnl.* 10, 85-86.

$Z$ = Set of exams not yet scheduled and not available. (FUTURE)

3. Let *C(A,G)* be defined as above.

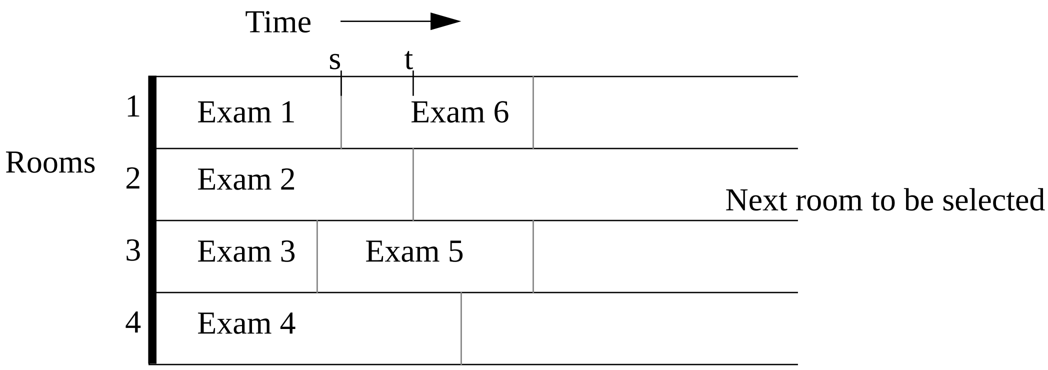
4. Let *R(A,B)* be the set of exams left over after the above room filling algorithm has been applied to the set of vertices *B*, where the set of vertices *A* are exams currently already assigned to a room.

5. Let *finish(x)* be the time at which currently scheduled exam *x* finishes.

**Algorithm**

1. $X = \emptyset, Y = C(X, G), time = 0, Z = V(G)$.
2. $X = X \cup (Y - R(X, Y)), Y = R(X, Y), Z = Z - Y$.
3. Find maximal $\xi \subseteq X$ s.t. for all $x, y \in \xi, finish(x) = finish(y)$ and for all $x \in \xi, z \in X, z \notin \xi, finish(x) < finish(z)$.
4. $time = finish(x)$ where $x \in \xi$.
5. $W = W \cup \xi, X = X - \xi$.
6. $Y = C(X \cup Y, G - W) \cup Y, Z = Z - Y$.
7. While $Y \cap Z \neq \emptyset$ repeat from 2 otherwise Halt.

**Example**




At time s, W={1,3}, X={2,4,5},Y={6 and other exams not conflicting with 2,4 or 5}.

At time t, W={1,2,3}, X={4,5,6}, Y={exams not conflicting with 4,5 or 6}.

At no time may any exam be in more than one of W,X,Y or Z

*Figure 5.*

## 5. Future Work

It is intended to continue extending the timetabling system described to include as many possible features as may be required for a generalised exam-timetabling system. This will include, as previously, stated a graphical user interface which will allow both interactive and batch scheduling of exams, allowing the user to make the definitive decision as to what the best timetable is. One of the biggest problems associated with timetabling is the disparity between an optimal solution of the associated model i.e. an optimal colouring of the conflictevent were to occur). Rooms on the other hand may be equipped for more than one of these possibilities. To schedule these exams, we fit each set of exams into a room that only takes one specialist type. If any exams are left over then for each room type that may take two sorts of exams, the left overs are scheduled in and so on, then with rooms that have three types of facility until there are either no exams or no rooms remaining. Any exams left over are then carried on to the next period as before. If, instead there are rooms still free, these may be added to the list of general classrooms for normal exams to use.

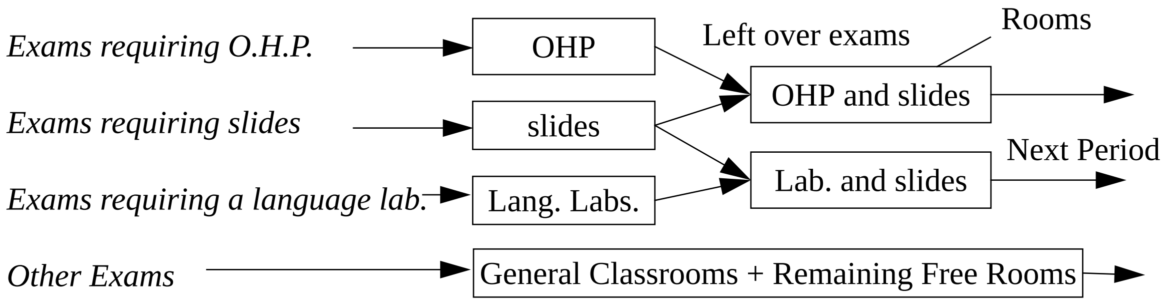



*Figure 4.*

**4.4 Variable Sized Periods**

Usually exams are scheduled in distinct time periods. These must each be as long as the longest exam scheduled in that period and usually for scheduling purposes are at least as long as the longest exam overall. If it were required to reduce the total length of time used to timetable all the exams that require scheduling, especially where accommodation is the constraining factor, we can use periods of variable size. Before, for each separate period, no two exams were allowed to student-conflict or staff-conflict. Now, for any particular time the same must apply.

The algorithms start in exactly the same way as normal, an A.M.I.S. first being found and then assigned to rooms leaving a list of unscheduled exams. Now, the algorithm will look for the first room to become free. The exam(s) that were previously filling that room are now ignored and extra exams are found using the A.M.I.S. algorithm that do not clash with any of the currently scheduled or left over exams. This room is then filled and the algorithm then proceeds to find the next free room and so on. The version described below will not take into account any desire for a gap between exams but any arbitrary time may be used.

**Definitions**

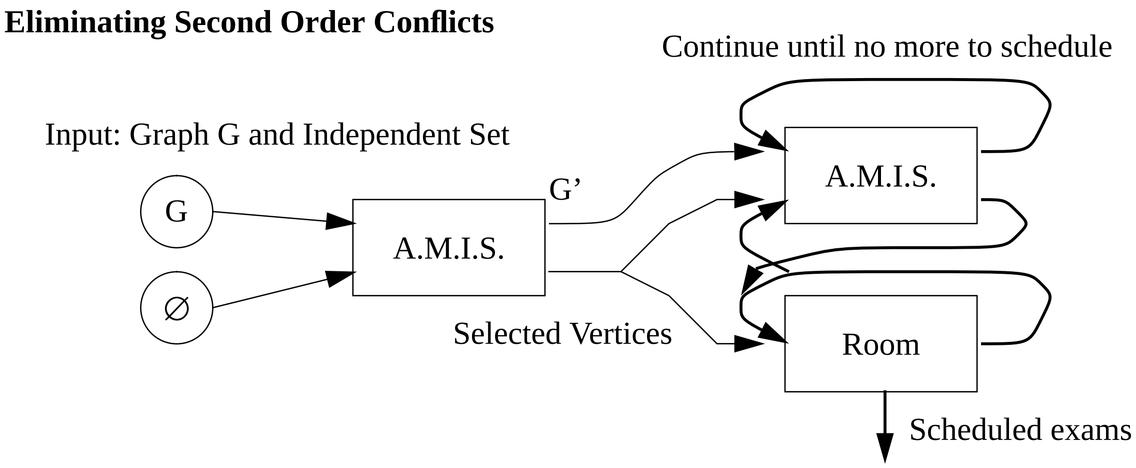
1. Let $G$ be a graph such that each vertex represents an exam to be scheduled and an arc between vertices represents a conflict.

2. Let $W$, $X$, $Y$, $Z$ form a partition of $G$.

$W$ = Set of exams already scheduled. (PAST)

$X$ = Set of exams currently scheduled. (PRESENT)

$Y$ = Set of exams available to be scheduled. (NEXT)




intervals up to about three hours. To minimise disturbance, instead of considering all the possible exams together, we take all the exams of similar length, trying to fit each set individually as before, but then take each room in turn, smallest first and choose the exam length that gives the greatest room usage. The exams of different lengths are then moved up to the next bigger room and so on in their own lists as specified in the room filling algorithm. This way, each room has exams only of one length. Of course gaps may occur, and in the most constrained of cases, these may be filled with other exams if necessary.

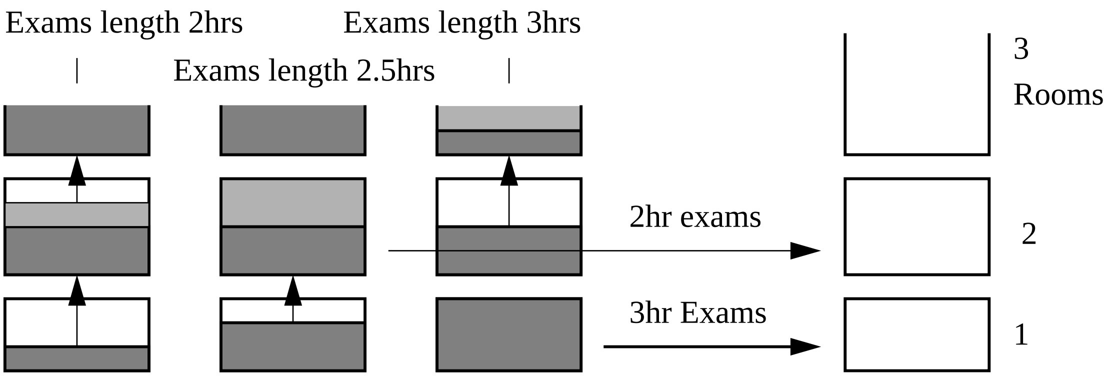



*Figure 3.*

### 4.3 Specialist Rooms

In general, exams consist of the examinee sitting at a desk, writing the answers to a set of questions as laid out on the exam paper. This process may take place in any classroom. Some exams, however, require the use of other specialist apparatus that may be disturbing to anyone else in the room. It is important then that such exams are not scheduled to an ordinary room.

An exam may require any one of a number of facilities. For example it might need a language or science lab or an over head projector or slides. For simplicity we assume that no exam will require more than one special facility (although it could easily be handled if such an unlikely

### 4.1 Eliminating Second Order Conflicts

In General, an nth order conflict would have n-1 periods between two exams. A first order conflict exists when two conflicting exams are scheduled in the same period. Similarly, a second order conflict is when a student has to sit exams in two consecutive exam periods. The method previously described already reduces this by using left-over exams which didn't fit into the rooms in one period as the start of the search for the next period. These exams will be a large non-conflicting set of currently unscheduled exams that do not conflict with any of the exams timetabled in the previous period. In the previous model however, other exams are added to create an A.M.I.S., so adjacent exams may still occur.

To eliminate second order conflicts we proceed as follows:

**Definitions**

1. Let G be a graph such that each vertex represents an exam to be scheduled and an arc between vertices represents a conflictand G be the set of vertices of G.

2. Let *C(A,G)* be the output of the above graph colouring algorithm, where A is a set of already coloured vertices in graph G, such that $A \cup C(A, G)$ is an A.M.I.S. of G and $A \cap C(A, G) = \emptyset$.

3. Let *R(B)* be the set of exams left over after the room filling algorithm described above has been applied to the set of vertices *A*.

4. Given two vertex sets *A* and *B*, we denote those vertices in *A* but not in *B* by *A-B*, similarly, define *G-A* to be the graph obtained by removing all veritces in A and associated edges from graph *G*.

**Algorithm**

1. $A_0 = \emptyset, A_1 = C(A_0, G), G = G - A_1$.
2. $A_2 = C(A_1, G), G = G - A_2$.
3. $B_2 = R(A_1)$.
4. $i = 3$.
5. $A_i = C(B_{i-1}, G), G = G - A_i$.
6. $B_i = R(A_{i-1} \cup B_{i-1})$
7. $i = i + 1$.
8. While $V(G) \neq \emptyset$ and $A_i \cup B_i \neq \emptyset$ repeat from 5 otherwise Halt.

.

*Figure 2.*

### 4.2 Minimising Disturbance

It is desirable during an exam that there is minimal disturbance to those people sitting the exams. One major cause of disturbance is when two exams are scheduled in the same room at once and one finishes before the other. The room filling algorithm mentioned previously works by filling up the smallest rooms first, then continuing on with the next bigger room until there are none left.

In minimising disturbance we are also trying to maintain the high room usage level as before. University exams tend to only come in a few possible lengths, one hour exams then half hour two exams may conflict so that students or staff members have to be in two different places at once. Also we have constraints on accommodation. Although for exam scheduling, we may allow more than one exam to be in a room at once, each room still has an upper bound for the number of students that may be assigned to it. Finally, the set of exam conditions must also be honoured. For example, an exam may require the use of a language laboratory or other similar facility.

## 3. A General Method Using Graph Colouring and Knapsack filling

The problem of splitting exams up into different groups has been shown to be equivalent to that of colouring a graph[9]. Each exam is represented by a different vertex with an edge between vertices where the two respective exams conflict in some way. Colouring the graph is then the process of allocating different colours to each vertex so that no two adjacent (conflicting) vertices have the same colour. Each colour is the equivalent of one period in the timetable. Therefore, ignoring the constraints imposed by the availability of rooms, the smallest exam period possible is the same as the minimum number of colours needed to colour the graph (the chromatic number). This problem has been proved to be NP-complete [6].

Similarly, the problem of fitting exams into rooms is equivalent to the knapsack filling problem with a set of exams to be fitted into a set of rooms. The aim here would be to fit as many exams as is possible into each room to maximise use of limited accommodation. This too has been shown to be NP-complete[6].

The methods used, described more fully in [1], start with a graph colouring algorithm which, initially takes a vertex of maximal degree, recursively chooses other vertices with the most common adjacent vertices, then it merges them together to produce an almost maximal independent set (A.M.I.S.). This is a set of near maximal size of vertices that may all be coloured the same colour. The set is then passed to the room filling algorithm which will attempt to fit as many exams in the rooms as is possible. Any left over are then used as the basis for the next period's search. This reduces the number of consecutive exams that students may have to take.

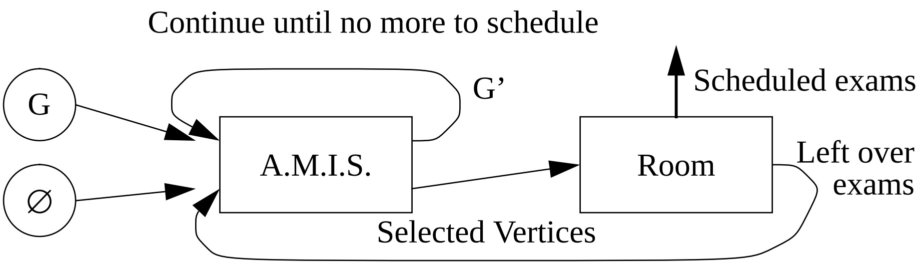



*Figure 1.*

## 4. Extensions To The Method

With exam scheduling, although possibly not as much as is the case with course scheduling, the concept of a "good" timetable is still very much a subjective idea. We intend to overcome this problem by the use of a graphical interface which may be used interactively either during the construction phase or to improve the final result. Other desirable features may however be listed. For example, not having two exams in a row, trying to use larger rooms more, having the largest exams earlier in the timetable or reducing the amount of disturbance to candidates when another exam leaves half way through. Some of these are considered in more detail below.

# Extensions To A University Exam Timetabling System

E.K.Burke, D.G.Elliman, R.Weare

University of Nottingham
University Park, Nottingham, NG7 2RD.
e-mail: ekb@uk.ac.nottingham.cs

*ABSTRACT*

The problem of constructing a generalised automated timetabling system is not only very difficult but involves a large number of variables and conflicting constraints. Many of the particular difficulties are associated with not knowing what constitutes a good timetable.

We present here, firstly a brief description of an already developed powerful method of assigning exams to periods and rooms so that a set of minimal but necessary constraints are satisfied. Secondly we consider how other desirable features may be incorporated into the same model with the least possible loss of flexibility or optimality. We also present a method of adapting the existing system to use different length periods in order to further reduce the total exam period. With universities still growing in size and complexity, this may well soon be a necessary feature of a standard timetabling system.

## 1.Introduction

In 1992, the University of Nottingham introduced a new modular degree course structure. The idea behind this is to provide the students with greater choice and flexibility in the range of subjects they can study to make up their degree. Whereas before, a student may have taken a few subsidiary options in the first year and then been largely confined to their own department, now they may take courses from any of a wide range of topics throughout their three or four years. This will necessarily place an extra burden on the timetabling of both courses and examinations. When considering a course scheduling problem it is desirable that no two courses student-conflict. However, such compromises must often occur in order to obtain a solution and the student is forced to chose between courses. In this paper we deal with the problem of exam scheduling where no exams may student-conflict under any circumstances.

At Nottingham, there are nearly ten thousand students, each of whom may be on any one of two hundred different degree courses. Each semester, approximately one thousand exams must be scheduled into an eleven day exam period. This means approximately one hundred exams must be scheduled per day. With modularisation and the large number of courses, we are now presented with a very large and increasingly difficult problem requiring ever more powerful and flexible scheduling tools.

## 2.Exam Scheduling

Exam scheduling is the process of assigning exam entities, each of which represents a set of students, at least one invigilator, (some institutional rules state there must be at least two invigilators), and a set of conditions under which the exam must take place, to a particular slot in the timetable and to a particular room. As previously stated, we have the condition that no