ORiON, Vol. 4, No. 2, pp. 92-102 *ISSN 0259-191X*

A UNIVERSITY COURSE TIMETABLING PROBLEM

J M PHALA
Academic Planning Office
University of Bophuthatswana (Unibo)
Private Bag X2046
Mmabatho
8681

ABSTRACT

This paper addresses a university course timetabling problem. The problem as experienced at Unibo is formulated in such a way that simple heuristics can be used to achieve the objective of finding either an 'optimal' or a feasible practical solution. The problem is divided into three phases, namely, grouping of courses into course groups, allocation of lecture rooms to courses within groups, and finally, scheduling of periods to course groups.

Heuristic algorithms are proposed to solve this problem. The computerised algorithms which have been integrated into Unibo's Student Management System are fully implemented [1].

1. INTRODUCTION

The problem we address in this paper is not new. "The timetabling problem has been of interest since late 1950's. Various researchers have dealt with the problem in different ways. The problem is of interest even now, largely due to its varied nature, complexity and last but not least its large size" [3].

Various approaches to the problem have been addressed in the literature. A survey of methods may be found in De Werra [4] and Junginger [5]. These methods include heuristic methods that simulate manual methods, exact mathematical algorithms, and heuristics that are derived and adapted from exact methods.

Because of the usual large size of the timetabling problem, and the fact that its variables are binary integer variables, it is a hindrance to solve it using mathematical programming techniques. ".. only simplified timetabling problems can be solved in this way, and the more realistic problems require heuristics." [4] Among such heuristic methods are node colouring methods, matchings in bipartite graphs, Lagrangian relaxation techniques, and quadratic assignment problem techniques.

We propose to formulate the problem as experienced at Unibo in such a way that the problem can be solved as a **multiphase** problem. Simple heuristic procedures are derived to achieve the objective of either finding an optimal solution, or finding a feasible practical solution to the problem.

Optimality in this problem is to some extent subjective, being an aid to a 'better' timetable in the sense of fair scheduling(to both the student and the lecturer), and efficient use of space. Therefore, given a properly formulated problem, we found heuristics good enough.

2. THE TIMETABLING PROBLEM

Tripathy [3] defines the timetabling problem as "the scheduling of a certain number of meetings, which are to be attended by a specific group of students and a teacher (or teachers), over a definite period of time, requiring certain resources (e.g. rooms, teaching aids, etc.) in conformity with the availability of certain other requirements."

We divide the problem as perceived and experienced at Unibo into three phases:

(i) Grouping of courses into course groups.
(ii) Allocation of rooms to courses within course groups.
(iii) Scheduling of periods to course groups.

2.1 GROUPING OF COURSES AND ROOM ALLOCATION

A course group is defined as a group of courses that may not be taken or registered for simultaneously by any student, and courses that are not being taught by the same lecturer, so that they can be scheduled in the same period for the same number of periods per week.

The grouping of courses reduces the size of the problem, and becomes necessary if the total mumber of periods needed to schedule to all courses exceeds the total number of periods available for scheduling. Otherwise, no grouping may be necessary at all. The groups are designed such that:

(i) The total number of courses of a specific size-group does not exceed the total number of available lecture rooms of that size.

(ii) The total number of courses in a group does not exceed the total number of available lecture rooms.

From the above definition of a group of courses, and the manner in which the groups are designed, we propose to combine phase (i) and phase (ii), thereby making the allocation of rooms to courses within groups feasible. Thus, the final grouping of courses is made dependent on the optimal allocation of rooms to courses within groups, subject to points (i) and (ii) above, and the condition that the courses must be clash-free both for the student and the lecturer.

The major part of the grouping phase is presently a manual exercise involving decisions based on the curricula for the different degree/diploma programmes and the allocation of courses to staff. This task is handled by the timetable committee [1], after which the allocation of rooms to courses within groups is then done by the computer centre, using a heuristic algorithm which is discussed in Section 2.1.2 below. This process may involve several runs of the algorithm, where the results of a run have to be referred back to the committee in the case of any infeasibility.

The advantage of this approach to course grouping and room allocation arises if one considers the fact that class sizes may fluctuate from time to time, and the unavailability of rooms to certain courses due to 'overflow' can be observed immediately, thus making it necessary to either rearrange certain courses between groups subject to points (i) and (ii) above, or even splitting certain courses of large size into course-sections subject to the availability of staff. This will also provide valuable input for overall space and personpower resource planning in the university.

An intuitive heuristic approach was developed to handle the room allocation problem with minimal effort. The formulation of the room allocation problem is given below, followed by the heuristic algorithm.

2.1.1 FORMULATION OF THE ROOM ALLOCATION PROBLEM

The room sizes at Unibo are distributed as in Table 1 below.

Table 1

Room Size	Number
40	4
50	4
80	4
120	4
180	4
240	1
320	1

All these rooms are equipped with basic equipment like white and black boards, OHP facilities, tape recording facilities, video equipment, and film and slide projection facilities. Thus the rooms are almost all equally equipped for lecture and tutorial purposes. As a result, we found it logical to maximize the utilization of seats in a room so as to facilitate a more efficient use of lecture room space. Similar approaches are, for example, advocated for by SABIN and WINTER [6], and LAPORTE and DESROCHES [7].

We define the following terms:

G = the total number of groups,

R = the total number of available rooms,

i = index denoting group,

$I = \{i : i=1,2,.....,G\}$,

NC_i = the total number of courses in group i,

j = index denoting course,

$J_i = \{j : j=1,2,.....,NC_i\}$,

NS_{ij} = total number of students in course j of group i.

k = index denoting room,

$K = \{k : k=1,2,.....,R\}$,

$RSIZE_k$ = size of room k.

We denote the decision variables by x_{ijk}, where

$$x_{ijk} = \begin{cases} 1 & \text{if course } j \text{ of group } i \text{ is allocated room } k, \\ 0 & \text{otherwise.} \end{cases}$$

The problem is to find the optimal allocation of rooms so as to maximize the utilization of seats:

$$\text{Maximize } \sum_{j \in J_i} \sum_{k \in K} c_{ijk} x_{ijk} \quad \text{for each } i \in I$$

(where c_{ijk} is the objective function coefficient related to the optimal allocation of room k to course j of group i:

$$c_{ijk} = \begin{cases} NS_{ij}/RSIZE_k & \text{if } RSIZE_k \geq NS_{ij}, \\ 0 & \text{otherwise.)} \end{cases}$$

subject to the following constraints:

(i) The number of students in a course is less than or equal to the size of the room:

$$\sum_{j \in J_i} NS_{ij} x_{ijk} \leq RSIZE_k, \quad k \in K \qquad (1)$$

(ii) A room is allocated to at most one course in a group:

$$\sum_{j \in J_i} x_{ijk} \leq 1, \quad k \in K \qquad (2)$$

(iii) Each course is allocated at most one room:

$$\sum_{k \in K} x_{ijk} \leq 1, \quad j \in J_i \qquad (3)$$

2.1.2 THE HEURISTIC ALGORITHM

Let $IASS_{ij}$ = a register for the room allocated to course j of group i.

$KCOUNT_k$ = a counter for the number of times room k was allocated to a course.

The steps in the algorithm are as follows (for each $i \in I$):

(1) Sort the NS_{ij}'s in ascending order of magnitude for each group, keeping record of the course index, j, of the nth smallest NS_{ij}. Let j_n denote this index.

(2) Sort the $RSIZE_k$'s in ascending order of magnitude, keeping record of the index, k, of the mth smallest $RSIZE_k$. Let k_m denote this index.

(3) **Initialization step:**

Set $KCOUNT_{ij} = 0$ and $IASS_{ij} = 0$ for all j, n = 1, and m=0.

(4) Set m = m+1.

(5) If $RSIZE_{k_m} \geq NS_{ij_n}$, continue. Otherwise, return to step (4).

(6) If $KCOUNT_{ik_m} = 1$, return to step (4).

(7) Set $IASS_{ij_n} = k_m$.

(8) If $n = NC_i$, stop.

(9) If m = R, stop.

(10) Set $KCOUNT_{ik_m} = KCOUNT_{ik_m} + 1$; n = n+1.

(11) Return to step (4).

2.2 SCHEDULING OF PERIODS TO COURSE GROUPS

Once the optimal grouping and allocation of rooms to courses has been achieved, the scheduling of periods to course groups will be the next phase. This phase is treated independently (by the computer centre), based on information supplied by the committee, since it does not affect the first two phases explicitly except for the fact that the groups are designed such that the total number of periods per week required to schedule to groups is at most equal to the total number of periods available for scheduling (See Section 2.1). A similar approach is suggested by GOSSELIN and TRUCHON [8], for example. The groups are thus implicitly designed in such a way that the number of periods needed for scheduling to all groups is 'minimized' subject to the availability of rooms, and the number of periods available for secheduling (See BARHAM and WESTWOOD [9] for a somewhat similar idea). The formulation of the scheduling of periods to course groups follows.

We define the following additional terms:

P = the total number of periods in a day,

P_i = the total number of periods per week to be scheduled to group i,

N_i = the maximum number of periods per day that can be scheduled to group i,

d = day d,

W = a week (= {d:d=1,2,3,4,5}),

p = index denoting period p,

L = the set of all periods per day (same for all days){p:p=1,2,...,P},

$$x_{ipd} = \begin{cases} 1 \text{ if course group i is scheduled in period p on day d,} \\ 0\text{, otherwise.} \end{cases}$$

The problem is to find an optimal schedule of periods to course groups:

$$\text{Maximize} \quad \sum_{i\varepsilon I} \sum_{p\varepsilon L} \sum_{d\varepsilon W} c_{ipd}\, x_{ipd}$$

subject to the following constraints:

(1) P_i periods must be scheduled to group i:

$$\sum_{d\varepsilon W} \sum_{p\varepsilon L} x_{ipd} = P_i \text{ for } i\varepsilon I \qquad (1)$$

(2) At most N_i periods must be scheduled to group i per day:

$$\sum_{p\varepsilon L} x_{ipd} \leq N_i \text{ for } i\varepsilon I,\ d\varepsilon W \qquad (2)$$

(3) At most one group is scheduled in period p on day d:

$$\sum_{i\varepsilon I} x_{ipd} \leq 1 \text{ for } d\varepsilon W,\ p\varepsilon L \qquad (3)$$

c_{ipd} is the objective function coefficient related to the desirability of scheduling courses in group i in period p on day d. For example, if it is desired to have group i scheduled in period p on day d, c_{ipd} is given a high value. In the case of any indifference, c_{ipd} is given the value 0, and in the case of total indifference, all the c_{ipd}'s are given the value 0, in which case only a feasible solution is desired [3,4]. In the context of this application, each lecturer responsible for a course/course section in a particular group is asked to rank the periods (for each day) from least preferred to most preferred (1 to P) and an average ranking is used for each group, period, and day combination. That is,

$$c_{ipd} = \frac{1}{NC_i} \sum_{j=1}^{NC_i} r_{ijpd},$$

where r_{ijpd} = rank for period p on day d for course/course section j in group i, and NC_i is as defined in Section 2.1.1.

In the case where only a feasible solution is desired (all $c_{ipd} = 0$), the problem is trivial. Furthermore, an extra constraint may be added in this case, in order to achieve a 'fair' schedule of periods(e.g. to avoid scheduling a particular group always in the first period in the morning, or the period just before or after lunch, etc.) by requiring that each group be scheduled at most once or twice or as the timetable committee might deem fit, per week in these periods:

$$\sum_{d \in W} ID_p \, x_{ipd} \leq f, \; i \in I, \; p \in L \qquad (4)$$

where

$$ID_p = \begin{cases} 1 & \text{if period } p \text{ is to be scheduled at most } f \text{ times per week to a group,} \\ 0, & \text{otherwise,} \end{cases}$$

and **f** is a constant determined by the timetable committee.

Because of the special and simple structure of this problem, we also designed a simple heuristic algorithm to solve it.

2.2.1 THE HEURISTIC ALGORITHM

Let $ISUMT_i$ = a counter for the number of periods scheduled to group i.

$ISUMD_{id}$ = a counter for the number of times group i was scheduled on day **d**.

$ISKED_{pd}$ = a register for the group scheduled in period p on day d.

$ICOUNT_{ip}$ = a counter for the number of times group i was scheduled in a specific period p(used only if constraints (4) are applicable).

$JCOUNT_{pd}$ = a counter for the number of times period p was scheduled to a group on day d.

The steps in the algorithm are as follows:

(1) Sort the c_{ipd}'s in descending order of magnitude, recording only the actual indexes i, p, d for the nth largest c_{ipd}. Let i(n), p(n), d(n) denote these indexes.

(2) **Initialization step:**
Set n=1, $ISKED_{pd}$=0 for all p and d, $ISUMT_i$=0 for all i, $ISUMD_{id}$= 0 for all i and d, $ICOUNT_{ip}$ =0 for all i and p, $JCOUNT_{pd}$=0 for all p and d.

(3) Set $ISKED_{p(n),d(n)}$= i(n).

(4) If n=5PG, stop.

(5) Set $ISUMT_{i(n)}$=$ISUMT_{i(n)}$+1, $ISUMD_{i(n),d(n)}$= $ISUMD_{i(n),d(n)}$+ 1, $ICOUNT_{i(n),p(n)}$= $ICOUNT_{i(n),p(n)}$+ 1, $JCOUNT_{p(n),d(n)}$= $JCOUNT_{p(n),d(n)}$+ 1.

(6) Set n=n+1.

(7) If $ISUMT_{i(n)}$= $P_{i(n)}$, return to step (6).

(8) If $ISUMD_{i(n),d(n)}$= $N_{i(n)}$, return to step (6).

(9) If $JCOUNT_{p(n),d(n)}$= 1, return to step (6).

(10) If $ID_{p(n)}$= 0, return to step (3).

(11) If $ICOUNT_{i(n),p(n)}$= f, return to step (6).

(12) Return to step (3).

3. IMPLEMENTATION RESULTS

The algorithms described above have been coded in FORTRAN77. The computerised timetable has been integrated into Unibo's Student Management System (SMS) which is menu-driven, running on a PRIME 550-2 from PRIME INFORMATION, and the system has been fully implemented. The timetable committee meets periodically towards the end of each semester to review the timetable, after which new information may be added to the system for the following semester's timetable which is made available at the end of the preceding semester.

The timetabling system as presently implemented has been running successfully since the second semester of 1986, being used solely for scheduling of formal lecture periods(which is where the actual problem was for a number of years). Ever since the system was introduced, students are able to get their timetables when they register, and classes are running smoothly without any interruptions due to timetabling problems.

On the other hand, because of the fact that courses are grouped into groups, it is almost impossible to satisfy the preferences of all lecturing staff when scheduling periods to course groups(reference to how c_{ipd} in Section 2.2 is obtained clarifies this), since the average ranking of a period is used. As a result of this, initially there were some dissatisfactions from some lecturing staff because they could not get the periods that they preferred, but after everybody had understood the mechanisms of the system, the system is running successfully with good corperation from all the staff. The system has been to the advantage of the students right from its inception, because the groups are designed on the basis of the different curricula.

The system presently does not schedule practical classes for science students. These are scheduled by the Science Cluster (consisting of four departments) once the final timetable is published.

4. CONCLUSION

Depending on the complexity of the timetabling problem, and taking account of the subjective nature of 'optimality' in this problem, we have thus far found heuristics to be good enough for solving this problem(which used to be a serious problem for some time) at Unibo.

The system is running successfully due to the fact that it has been possible to create, implement and operate it with the expertise and resources of a small team that represents both the people at policy level of university management, and those at the operational level [1]. Moreover, the system serves as a very important academic and physical planning tool.

ACKNOWLEDGEMENTS

The author wishes to thank Prof J S Wolvaardt and an anonymous referee for their judicious comments on an earlier version of this paper.

REFERENCES

[1] CLAY KISTLER, Chairman: **Unibo Timetable Committee.** The Timetable Committee consists of representatives of the different schools(faculties elsewhere), the computer centre, and the administration.

[2] TRIPATHY, A.,'A Lagrangian Relaxation Approach to Course Timetabling', **J. Opl Res. Soc.**, 31, 7(1980), 599-603.

[3] ------------,'School Timetabling - A case in Large Binary Integer Linear Programming', **Management Sci.**, 30, 12 (1984), 1473-1489.

[4] de WERRA, D.,'An Introduction to Timetabling', **European Journal of Operational Research,** 18(1985), 151-162.

[5] JUNGINGER, W.,'Timetabling in Germany - A survey', **Interfaces,** 16, 4(1986), 66-74.

[6] SABIN, G. C. W. and WINTER, G. K.,'The Impact of Automated Timetabling on Universities - A Case Study', **J. Opl Res. Soc.,** 37, 7(1986), 689-693.

[7] LAPORTE, G., and DESROCHES, S.,'The Problem of Assigning Students to Course Sections in a Large Engineering School', **Comput. & Ops. Res.**, 13, 4(1986), 387-394.

[8] BARHAM, A. M., and WESTWOOD, J. B.,'A Simple Heuristic to Facilitate Course Timetabling', **J. Opl Res. Soc.**, 29(1978), 1055-1060.

[9] GOSSELIN, K. and TRUCHON, M.,'Allocation of Classrooms by Linear Programming', **J. Opl Res. Soc.,** 37(1986), 561-569.