

## AN INTEGER PROGRAMMING APPROACH TO REFERENCE STAFF SCHEDULING

JAMES C. BEAN

Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109, U.S.A.

and

MARGARET H. BEAN

Engineering/Transportation Library, The University of Michigan, Ann Arbor, MI 48109, U.S.A.

*(Received in revised form 25 March 1985)*

**Abstract**—We address the problem of scheduling the reference staff of a large academic library where available reference staffing and time flexibility are highly constrained. The problem is modeled as a multiple choice integer program and is solved using existing software. The computer model constructed a superior schedule to the current manual method and takes 1.2 seconds of computer time versus many hours by hand.

### 1. INTRODUCTION

In many large academic libraries, reference service is provided during the library's busiest hours. While reference schedules vary greatly from library to library, they generally consist of daytime, weekend, and evening hours. The number of hours that reference service can be provided is dependent upon the size of the reference staff. To ensure that a reference librarian is on duty during posted reference hours, and to plan for staff members' other commitments such as committee meetings or classes, it is frequently necessary to schedule the reference staff to cover these hours.

If the staff has ample time dedicated to reference, this scheduling problem is not particularly difficult. A good common sense algorithm executed by hand will deliver an adequate schedule. However, it is more likely that hours available for reference are highly constrained by other commitments. In this case, it can be difficult to find a feasible schedule, let alone one that is acceptable to all reference staff members.

This scheduling problem consists of two distinct parts: the managerial or negotiation process, and mathematically piecing together the puzzle of shifts. The process of constructing an acceptable schedule often involves several iterations through each of these phases. The preferences and requirements of the staff that arise in negotiation become the constraints on the mathematical model. Conversely, the model evaluates the implications of each such negotiation which may initiate further negotiation.

In this study, we show how to model the technical aspects of this scheduling problem as an integer program and find a solution using existing computer software. This model is a tool that can be used in the managerial process of staff scheduling.

### 2. PROBLEM STATEMENT

The motivating example is found in The University of Michigan Engineering/Transportation Library. The library serves 15 engineering departments with over 300 faculty and 4500 graduate and undergraduate students and consists of a main collection on the university's central campus and a branch library at a separate campus location three miles away. The same reference staff is responsible for staffing the reference desks at both libraries. The staff consists of three full-time librarians, one part-time librarian, and two library graduate student staff assistants (LGSSA's) from The University of

Michigan School of Library Science. Since the LGSSAs' class schedules change each semester, a new reference schedule is prepared every term. During the 1984 Winter term, the total staff time available for reference was 72.5 hours per week while the desired reference schedule required 67.5 hours of reference. Due to other responsibilities, the times that each could work the reference desk were constrained.

Schedules can be evaluated along two lines: hard constraints and soft constraints. A schedule must have certain characteristics. For example, no staff member can be assigned shifts that overlap in time at the two libraries. Though restrictions of this sort may seem obvious, it will be necessary to include them explicitly when describing the problem in a mathematical model. We will refer to these restrictions as *hard constraints* and require that the schedule meet them.

Other restrictions involve desired attributes of the schedule such as personal preferences and quality considerations. These restrictions will be referred to as *soft constraints* and are usually the result of preliminary negotiation with the staff. Soft constraints can be handled in either of two fashions. We can include them as constraints and require that the schedule meet them, or we can optimize these characteristics subject to the hard constraints. The approach taken here is the former, that is, to include soft constraints as requirements. If the problem has no feasible solution, the soft constraints can be slowly relaxed through further negotiation until a feasible schedule is found.

Following are lists of the hard and soft constraints appropriate to the motivating example. As is discussed in Section 5, many other constraints are possible.

*Hard constraints:*

- (1) Each staff member has a maximum number of reference hours.
- (2) Overlapping shifts at the two libraries cannot be assigned to the same person.
- (3) The LGSSA's cannot be assigned shifts that conflict with their class schedules.

*Soft constraints:*

- (1) No one should be assigned more than one shift per day at the hectic main library to prevent reference "burn-out."
- (2) No one should be assigned afternoon and evening reference during the same day at the main library.
- (3) The preferences of the senior librarians should be respected.
- (4) The hours assigned the two LGSSA's should be equitable.
- (5) Time should be assigned in blocks of two or three hours.

In order to describe the problem in a mathematical model and solve it using existing software we must decompose the problem. This consists of determining parts of the schedule that are better done by hand either before or after the computer solves the core of the schedule. Though the exact decomposition will vary by specific example, the characteristics to look for are represented in this example. Following is the decomposition used in attacking this specific example:

- (1) The Library Head works one reference shift per week and prefers a particular time slot. This time slot was assigned to her and removed from the problem.
- (2) The Sunday reference shifts are only related to other assignments by maximum reference hours. Further, the subset of the staff that works Sunday reference is known beforehand. Hence, these assignments can be made simply by hand and the hours subtracted from the reference time available for the staff involved.
- (3) Reference at the branch library is less hectic than the main library and is considered easy duty. Hence, these shifts are not related to any other shifts for quality considerations. The only constraints affecting these assignments require that they not conflict with overlapping assignments at the main library and class schedules. Further, tradition dictates that a certain subset of the staff rotate through these shifts. Hence, the appropriate hours are subtracted from available reference time before the computer model is solved.

This portion of the problem is decomposed prior to problem solution. However, the actual scheduling is done after the mathematical model is solved so that overlaps

with assignments in the main library can be avoided. These main library assignments are made during the computer run.

The shifts used in this example were the traditional shifts that the hand-constructed schedule attempted to use. They consisted, each weekday, of a three hour shift in the morning, a two hour shift around lunch, a three hour shift in the afternoon and a two and a half hour shift in the evening. After the decomposed sections were removed from the schedule, eighteen shifts were to be assigned over a week's time. The schedule would then repeat each week throughout the semester.

### 3. THE MODEL

Following is a mathematical description of the core scheduling problem.

Let  $x_{ij}$  be a decision variable that takes the value 1 if shift  $i$  is assigned to staff member  $j$  and 0 otherwise.

Let  $t_i$  be the length of shift  $i$  in hours and  $T_j$  be the maximum time assignable to staff member  $j$ .

Let  $D_k$  be the set of day shifts at the main library during day-of-the-week  $k$  and  $N_k$  be a set containing the afternoon and evening shifts at the main library during day-of-the-week  $k$ .

Let  $L$  be the number of staff members and  $M$  be the number of shifts to be assigned. We can now state the integer programming model as follows.

$$\text{find } x_{ij}, \quad i = 1, 2, \dots, M; \quad j = 1, 2, \dots, L$$

$$\text{subject to: } \sum_{i=1}^M t_i x_{ij} \leq T_j \quad \text{for } j = 1, 2, \dots, L \quad (1)$$

$$\sum_{i \in D_k} x_{ij} \leq 1 \quad \text{for } j = 1, 2, \dots, L; \quad k = 1, 2, \dots, 5 \quad (2)$$

$$\sum_{i \in N_k} x_{ij} \leq 1 \quad \text{for } j = 1, 2, \dots, L; \quad k = 1, 2, \dots, 5 \quad (3)$$

$$\sum_{j=1}^L x_{ij} = 1 \quad \text{for } i = 1, 2, \dots, M \quad (4)$$

$$x_{ij} \in \{0, 1\}.$$

Constraint set (1) requires, for each staff member, that the total hours assigned be not greater than the prescribed maximum. Constraint set (2) requires that, for each staff member, for each day of the week, that no more than one shift at the main library be assigned. Similarly, constraint set (3) eliminates the possibility of being assigned both the afternoon and evening shifts on the same day. Constraint set (4) states that each shift must be assigned to exactly one staff member.

For the example problem solved, this formulation has 80 variables and 73 constraints.

### 4. SOLUTION METHODOLOGY

The integer program above has a structure known as the multiple choice integer program. This problem has been discussed often in the recent literature (see Bean[1], Martin, Sweeney, and Doherty[2], Sweeney and Murphy[3], Martin[4]). This structure has been applied to a wide variety of problems such as catalogue space planning (Johnson, Zoltners, and Sinha[5]), sales resource allocation (Zoltners and Sinha[6]) and asset divestiture (Bean[7]).

This formulation has advantages over other possible techniques. The state space is so complex that dynamic programming is inefficient. The multiple choice integer program has advantages over general integer programming formulations because it

takes advantage of the special structure that this problem exhibits through constraint set (4). The branch-and-bound tree used for this problem has the characteristic that going forward from any node corresponds to fixing all of the variables associated with one shift; one at 1 and the others at 0. In a general binary tree only one variable would be fixed at a time. Conceptually, this approach reduces the problem from  $ML$  decisions to  $L$ . In the general tree there are  $2^{ML}$  branches, whereas the tree employed by the solution technique described below has only  $L^M$ . Hence, the number of potential solutions when the problem is solved with this algorithm increases exponentially in the number of shifts, but only polynomially in the size of the reference staff.

The solution technique used in this application is that described in [1]. This is a branch-and-bound approach using a branching technique especially designed for multiple choice problems. This basic procedure is then enhanced two ways. The variable reduction techniques of Martin *et al.* are used to eliminate variables from consideration *a priori*. Then, a Lagrangian version of the problem is solved that greatly speeds up the computation times.

The Lagrangian problem solved is not a Lagrangian relaxation, but a Lagrangian alteration of the costs. The feasible region is kept intact, but the general constraints are relaxed into the objective as well. For details see [1]. This Lagrangian alteration significantly improves the computational efficiency of the algorithm. For details on Lagrangian approaches to integer programming problem see [8].

## 5. RESULTS

This model was used to create a schedule for the academic term in progress, Winter 1984, so that it could be compared with the hand-constructed schedule currently implemented. (Comparison with other techniques would not have allowed comparison with an implemented schedule.) The comparison of the schedule, in lieu of costs, was made by the judgment of the reference staff. Iteratively, their comments were added to the model as constraints until the schedule was satisfactory to each member of the staff. The hand constructed schedule did not meet the requirements in several points of comparison. The major difference between the two schedules is that the desired two to three hour shifts are intact in the computer-generated schedule. In the hand-constructed schedule, shift fragmentation was necessary. Figure 1 shows graphically the hand-constructed schedule. The computer-generated schedule is shown in Figure 2. Note that the computer-generated schedule includes decomposed sections done by hand as explained in Section 2. Letters A-F in the figures represent the six reference staff members.

Of more importance to the Reference Head, whose task it is to create the schedule, is the fact that days of switching, penciling, and negotiation were replaced by 1.2 seconds of CPU time on the University of Michigan Amdahl 5860.

The code used is structured to solve general multiple choice integer programs. Hence, several hours of hand construction of the input deck was necessary. Were the code to be used on a regular basis, additional modules would be written to streamline this input process. Such streamlining was done for the asset divestiture problem described in [7] with great success. In a user-friendly format, the time required to set up and run this model would be negligible relative to the negotiation part of the overall staff scheduling process.

## 6. EXTENSIONS AND CONCLUSIONS

### 6.1 Extensions

The model presented here is by no means the limit of generality solvable by the code in [1]. As noted, the cost structure available in the code was not used at all. This leaves open the possibility of expanding the model to include values for each shift-staff assignment. A typical approach is to have each staff member weight the shifts according to personal preference. The objective could then be to maximize total preference, weighted by some seniority factor.

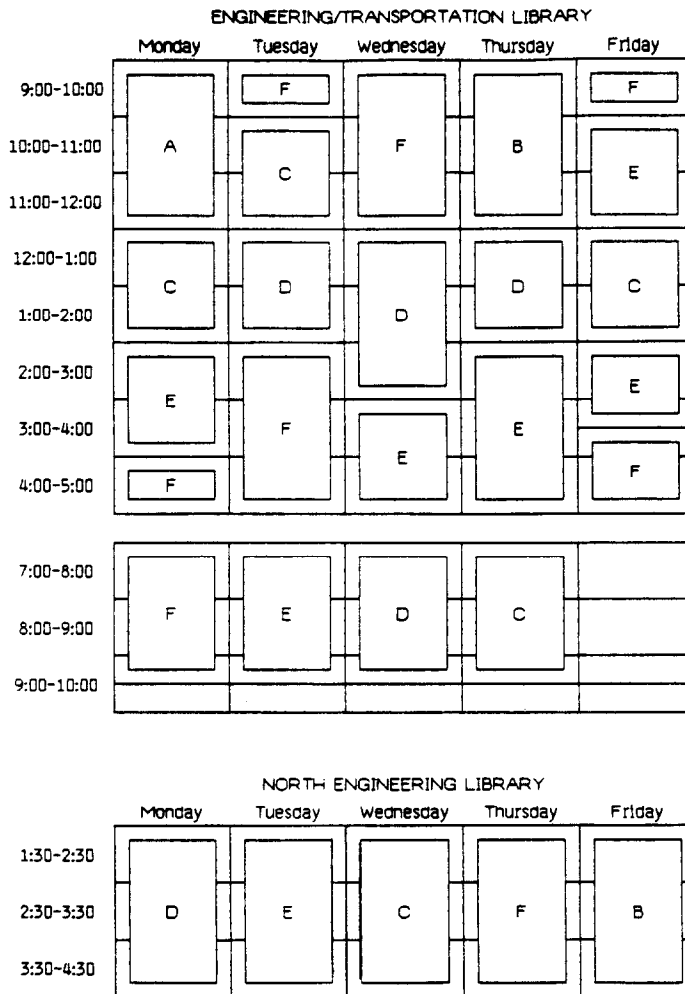


Fig. 1. Reference schedule. Hand constructed, actually implemented.

The model allows any constraint that can be formulated as a linear equality or inequality in the variables defined in Section 2. This flexibility is necessary since each specific application has different requirements.

This approach could also be used on other staff scheduling problems commonly found in libraries, e.g. circulation desk staffing. This particular problem is generally much larger than the one addressed in this study, and hence would require more decomposition and computer time to find a solution. However, there is nothing structurally to prevent such an analysis.

## 6.2 Conclusions

In this work, we have shown that reference staff scheduling in a large academic library can be modeled as a multiple choice integer program. After some judicious decomposition, the problem can be solved on a mainframe computer in seconds. The schedule generated by the computer had desirable characteristics that the hand-generated schedule did not.

The approach we have presented is not a substitute for the scheduling process currently used in most libraries. However, this mathematical approach can aid in the scheduling process by speeding up and improving the quality of the technical portion of the staff scheduling process. The model presented here can be altered to be appropriate for a wide range of applications. Further, the technology exists to solve the model quickly. Though the code used in this study was not tailored specifically for this application, such a development could easily be accomplished.

ENGINEERING/TRANSPORTATION LIBRARY					
	Monday	Tuesday	wednesday	Thursday	Friday
9:00-10:00					
10:00-11:00	A	F	F	B	C
11:00-12:00					
12:00-1:00	F	E	E	C	F
1:00-2:00					
2:00-3:00	E	C	D	D	E
3:00-4:00					
4:00-5:00					
7:00-8:00	C	D	E	F	
8:00-9:00					
9:00-10:00					

NORTH ENGINEERING LIBRARY					
	Monday	Tuesday	wednesday	Thursday	Friday
1:30-2:30					
2:30-3:30	B	F	C	E	D
3:30-4:30					

Fig. 2. Reference schedule. Computer constructed.

## REFERENCES

- [1] J. BEAN, A Lagrangian algorithm for the multiple choice integer program. *Operations Res.* **32**, 1185-1193 (1984).
- [2] K. MARTIN, D. SWEENEY, and M. DOHERTY, The reduced cost branch-and-bound algorithm for mixed integer programming. Working Paper, Graduate School of Business, University of Chicago (1981).
- [3] D. SWEENEY and R. MURPHY, Branch-and-bound methods for multi-item scheduling. *Operations Res.* **29**, 853-864 (1981).
- [4] K. MARTIN, Branching strategies for integer programming with special ordered sets. Presented at ORSA/TIMS National Meeting, Colorado Springs (1980).
- [5] M. JOHNSON, A. ZOLTNERS, and P. SINHA, An allocation model for catalogue space planning. *Management Science* **25**, 117-129 (1979).
- [6] A. ZOLTNERS and P. SINHA, Models for sales resource allocation. *Management Science* **26**, 242-260 (1981).
- [7] J. BEAN, Asset divestiture strategies with return on equity requirements. Department of Industrial and Operations Engineering, Technical Report 83-8. The University of Michigan, Ann Arbor, Michigan 48109 (1983).
- [8] M. FISHER, Lagrangian relaxation method for solving integer programming problems. *Management Science* **27**, 1-18 (1981).