MONASH UNIVERSITY
FACULTY OF BUSINESS & ECONOMICS

SCHEDULING STAFF USING MIXED INTEGER PROGRAMMING

Nicholas Beaumont

Working Paper 20/97
March 1997

Abstract

This paper describes the solution of a problem of scheduling a workforce so as to meet demand which varies markedly with the time of day and moderately with the day of week. The main objectives were determining how many staff to emply and the times at which shifts should start.

The problem was expressed as a large MIP problem initially presenting computational difficulties. The difficulties vanished when the formulation was modified and when a package allowing use of reduce and (especially) special ordered sets became available.

The client commissioned the study primarily to benchmark its existing schedule by comparing it with a theoretical optimum. The optimal schedule and comparison are very sensitive to technical and cost coefficients which are not preciesely known.

Keywords: Scheduling, Staff Rostering, Integer Programming.

1 Introduction

The Problem The client for which this research was done employed staff who drove to and serviced customers. The demand varied markedly with day of week and time of day, and had to be met 24 hours each day and 365 days each year. Queuing was undesirable, albeit inevitable, but its cost had not been quantified by the organisation. A rule of thumb was that customers should have to wait more than two hours only in exceptional circumstances.

The identity of the organisation and the kind of work done is confidential but analogous activities are:

- Repairpeople travelling between and servicing faulty lifts,
- Police travelling between and dealing with incidents and
- Taxis serving the public.

The workforce comprised the organisation's own employees (paid by the shift and subject to various scheduling restrictions) and self employed contractors who were available on call and paid a fixed amount per call. The problem required finding the right number of employees, the times at which they should start work and how much of the demand should be met by contractors.

The essence was to balance the number of workers on duty at any time with the demand at that time. A complicating factor is that, initially, management did not want to have to administer more than about 5 to 8 different starting times in any one day. This restriction was later relaxed to '12 to 15'.

Past Work Scheduling problems and their treatments are very varied. The problem of designing a staffing schedule or roster (sometimes known as a tour) subject to a particular st of constraints is solved in this paper by expressing it as a mixed integer program and solving it. Examples of this technique's use are given by Williams[14]. Early examples of the use of linear programming in rostering problems are given by Baker & Magazine[1] and Bartholdi, Orlin & Ratcliff [2]. The former considers the maximum workstretch (number of contiguous days on duty) constraint and gives limited recognition of demand varying with the day of the week. We were unable to find other examples of the the design of rosters in which the use of combinatorial optimization is important. Experience suggests that expressing scheduling problems as MIP's leads to large and intractable problems.

The solution of a problem with some similarities is given by Townsend[13]. An aspect of this problem was there being several different duties which had to be distributed fairly amongst crews. The rules governing the pattern of days on and days off were simpler than those described in this paper.

Set covering techniques have been used in aircrew scheduling by, for example, Ryan[12]. This technique uses rosters as an input, one aspect of the technique is allocating rosters to staff. The problem solved in this paper was finding a feasible roster. Bechtold, Brusco & Michael[3] exemplify the approach of generating numerous feasible rosters and finding the few which maximise desirable criteria. Easton & Rossin's[8] approach is to take a set of feasible rosters and use a heuristic to try and find improvements to them.

Other contributors include Hung[9, 10, 11] who considers problems complicated by factors such as a non-homogenous labour force (one kind of worker can replace another, but not vice versa). These papers tend to emphasise problems in which the same number of days are worked each week. Some attention is given to the important practical point of the days worked each week being contiguous.

2 Data

The following data are required. The cost data has been disguised without changing relative values.

Demand The organisation had recorded the time and date of each of the approximately 1 million telephone calls for service between 1st October 1989 and 30th September 1990. It is probable that the number of calls underestimated demand as some people getting a busy tone may have baulked. Computer programs were written to:

- Estimate any seasonal variation in demand. Table 1 shows that, although there was considerable variation amongst the months, there was (to the surprise of the organisation) little variation amongst seasons.
- Calculate the average demand in each of the 504 20-minute periods of the week, the reasons for the choice of a 20-minute period are discussed in section 3 below. Figure 1 displays the demand, averaged over the year, in each 20-minute period of the week.

Cycles An input to the problem described in this paper is a set of **cycles**. A cycle is a cyclic pattern of days worked and days off followed by an individual worker and may be of any length. The organisation has used 1 week (the same days are worked each week) 20 week and 52 week cycles.

The problem of obtaining cycles (the **cycle design problem**) which reflected labour agreements and optimised a weighted sum of objectives could be expressed as an MIP which is discussed in a separate paper[4]. One objective is to have the total number of people on duty on a given day of the week roughly proportional to the mean demand on that day. Constraints are exemplified by there being a maximum allowed number of contiguous days on duty and minimum number of contiguous days off duty. A 52 week cycle is followed by a small multiple (q) of 52 employees. Of $52q$ employees, q would be working each week of the cycle.

The single 52 week cycle now being used by the organisation was used as input. The standard cost of a week's work (CE) included the cost of annual leave. Allowance was made for staff taking five weeks annual leave.

Cost of Staff The relevant costs of an employee's 'eight-hour' shift are assumed fixed and known. It was suggested that the organisation consider employing drivers on four and twelve hour shifts. Twelve hour shifts were eliminated because Australian overtime regimes implied that a twelve-hour shift would be about 20% more expensive than contiguous eight and four-hour shifts.

Customers may be served by either permanent staff (paid by the week) or contractors who are paid an amount per job which varies with the time. The fee for contractors' 'night' jobs is higher than that for a 'day' job. At present, 'day' is between 7:00 am and 7:00 pm on weekdays and 'night' is any other time.

Server Capacity A critical input is the average maximum number of calls that an employee can handle per hour. The organisation's estimate was 3.0. This figure is probably less during times of high traffic congestion or at night, when the few employees on duty have to drive long distances between jobs.

A nominally eight-hour shift lasts 8 hours 20 minutes and includes a 40-minute meal-break. Given that staff must not work more than five hours without a meal-break, this must start no earlier than 160 minutes nor finish later than 340 minutes after the start of the shift. It is assumed that meal-breaks are uniformly distributed between these limits. It follows that if x employees start at (say) midday then, before 2:40pm and after 5.40pm x are available, but that between those times only $(180-40)/180x = 0.778x$ are available. This is probably unrealistic, one would hope that where possible, employees would eat when the demand was low. The serving capacity of an average employee for each 20 minute period of a shift was input.

Cost of Waiting A difficult to ascertain figure is the hourly cost of keeping a customer waiting. This might be expected to vary with the time of day.

Other data are described the next section.

3 Formulation

3.1 Parameters and Indices

P the number of periods into which a 24-hour day was divided, normally 72.

NC The number of cycles input.

$i \in I = \{1, 2, ..., 7P\}$ is a subscript for the $7 \times P$ periods in a week, starting from 00:01am –00.20am Sunday.

$j \in J = \{1, 2, ..., P\}$ is a subscript for the periods in a day, counted from 00:01am.

$c \in C = \{1, 2, ..., NC\}$ is a subscript for the NC cycles

NS_c $c \in C$ The number of shifts in each cycle.

$s \in S_c = \{1, 2, ..., NS_c\}$ $c \in C$ is a subscript for the NS_c shifts of each cycle.

3.2 Constants

L_c $c \in C$ The length of each cycle in days.

B_{cds} $c \in C$ $d \in D$ $s \in S_c$ A flag which is 1 if day d of the week is present in shift s of cycle c, 0 otherwise. This is derived from the input cycles described above.

D_i $i \in I$ The mean demand in period i of the week. A factor by which all demands could be multiplied was made available to the client but is not discussed here.

K The maximum number of different times of one day at which shifts which can be started, typically 5 to 15.

M The maximum number of workers available.

M_j $j \in J$ The maximum number of workers who can simultaneously start work in period j. As the organisation could not cope with a large number of workers starting work in one 20 minute, this was set at 20. period.

R_i $i \in I$ The relative efficiency of an employee or contractor in period i of the week. This has a mean of 1.0 but might be lower, because of traffic congestion, at peak periods.

$A8_k$ $k = 1, \ldots, \lceil 25P/72 \rceil$ The expected mean number of jobs an 8-hour employee can complete in period k of a shift in normal conditions ($\lceil x \rceil$ is the smallest integer $\geq x$). A normal shift lasts 8 hours and 20 minutes. Typically some dead time is used getting into position and returning to base. For some periods $A8_k$ will be about 0.78 because some employees will be having meal breaks (see section 2). The net effectiveness of a worker will be $R_i A8_k$ $i \in I$ $k = 1, \ldots, \lceil 25P/72 \rceil$. Analogous data is used for other length shifts.

$A4_k$ $k = 1, \ldots, \lceil P/6 \rceil$ The expected mean number of jobs an employee can complete in each period of a 4 hour shift in normal conditions.

CC_i The cost of a contractor dealing with a job during period i. All costs are expressed in Australian dollars (\$A1 $\approx$ \$US0.81).

CE_c $c \in C$ The weekly cost of mounting cycle $c \in C$.

$C4_i$ $i \in I$ The cost of mounting a four-hour shift starting in period i.

CW_i $i \in I$ The cost per hour of a customer waiting in period i (a typical value was \$12/hour).

CT_i $i \in I$ The mean cost incurred by a vehicle travelling between successive jobs in period i.

$D = \sum_{i \in I} D_i$ The total mean weekly demand.

$LEAVEFACT$ A factor expressing the fact that workers on e.g. a 47 week shift have an efficiency of 47/52 because they are on leave for 5 weeks of the year.

3.3 Variables

δ_j $j \in J$ A binary variable which is 1 if any shift of any cycle starts in period j, otherwise 0.

x_{cjs} $c \in C$ $j \in J$ $s \in S_c$ The number of 'eight-hour' workers on each shift s of cycle c starting work at the beginning of period j.

The employees working a particular shift (s) of a particular cycle (c) must all start work at the same time i.e., for each $s \in S_c, c \in C$ at most one member of the set x_{cjs} $j \in J$ can be non-zero i.e. for each $s \in S_c, c \in C$ the x_{cjs} $j \in J$ comprise a Special Ordered Set Type 1 (SOS1).

y_i $i \in I$ The number of four-hour shifts starting at the beginning of period i.

z_i $i \in I$ The number of jobs done by contractors in period i.

w_i $i \in I$ The maximum number of jobs that can be done in period i.

v_{cjs} $c \in C$ $j \in J$ $s \in S_c$ The total number of 8 hour workers, weighted by their cover factor ($A8_k$), from shift s of cycle c at work in period j.

q_c $c \in C$ The common, integer, number of 8 hour workers employed in each shift of cycle c.

r_i $i \in I$ The total number of 8 hour workers, weighted by their cover factor ($A8_k$), at work in period i.

u_i $i \in I$ The number of customers who are waiting for service at the beginning of period i.

T total weekly cost of the service operation.

T_p total cost of employees and contractors.

T_v total vehicle travelling cost.

T_w total customer waiting cost.

γ_{cjs} $c \in C$ $j \in J$ $s \in S_c$ These binary variables are used only to express the fact that for each $s \in S_c, c \in C$ the x_{cjs} $j \in J$ comprise a Special Ordered Set Type 1 (SOS1). They are not used in the model's final implementation.

All variables are non-negative.

3.4 The Objective Function

The total payment to the employees and contractors in a week (T_p) is the sum of all the payments to employees on eight-hour and four-hour shifts and contractors. Thus:

$$T_p = \sum_{c \in C} L_c CE_c q_c + \sum_{i \in I} C4_i y_i + \sum_{i \in I} CC_i z_i \tag{1}$$

The total cost of the organisation's vehicles travelling between jobs is:

$$T_v = \sum_{i \in I} CT_i(D_i - z_i) = \sum_{i \in I} CT_i D_i - \sum_{i \in I} CT_i z_i. \tag{2}$$

The term $\sum_{i \in I} CT_i D_i$ is constant and is omitted from the objective function. The cost of waiting is:

$$T_w = (24/P) \sum_{i \in I} CW_i u_i. \tag{3}$$

The total cost to be minimised is therefore:

$$T = T_p + T_v + T_w. \tag{4}$$

3.5 Constraints

Constraints (5) and (6) express the requirement for shifts to start at at most K different times of the day.

$$x_{cjs} \leq M_j\delta_j \quad c \in C \; j \in J \; s \in S_c \tag{5}$$

$$\sum_{j\in J}\delta_j \leq K. \tag{6}$$

All staff working a particular shift s of a particular cycle c must start work in the same 20 minute period. Constraints (7) and (8) express the fact that for each $s \in S_c, c \in C$ the x_{cjs} $j \in J$ comprise a Special Ordered Set Type 1 (SOS1), i.e. that for each $s \in S_c, c \in C$ at most one of the x_{cjs} $j \in J$ can be non-zero. In CPLEX 4.1[7] this can be expressed directly, the γ_{cjs} $c \in C$ $j \in J$ $s \in S_c$ are not input.

The SOS1's are relaxed by dropping the requirement that at most one member be non- zero. Equivalently, the constraint γ_{cjs} $c \in C$ $j \in J$ $s \in S_c$ binary is replaced by γ_{cjs} $c \in C$ $j \in J$ $s \in S_c \leq 1$. This is tantamount to omitting Constraints (7) and (8).

$$x_{cjs} \leq M_j\gamma_{cjs} \quad c \in C \; j \in J \; s \in S_c \tag{7}$$

$$\sum_{j\in J}\gamma_{cjs} \leq 1 \quad c \in C \; s \in S_c. \tag{8}$$

The number of people working in each shift of cycle c is common. There is no intrinsic reason for this and in future constraint (9) may be relaxed.

$$q_c - \sum_{j\in J} x_{cjs} = 0 \quad c \in C \; s \in S_c. \tag{9}$$

The maximum number of jobs w_i that can be done in period i depends on the total number of service vehicles on the road during period i including 8-hour shifts, 4-hour shifts and contractors. Since an 8-hour shift lasts $\lceil 25P/72 \rceil$ periods, and a 4-hour shift lasts $\lceil P/6 \rceil$ periods, w_i is a function of:

$$x_{cjs} \quad j = (i-k-1)|(7\times P)+1, k = 1,2,\ldots,\lceil 25P/72 \rceil \quad c \in C \; s \in S_c \tag{10}$$

$$y_j \quad j = (i-k-1)|(7\times P)+1, k = 1,2,\ldots,\lceil P/6 \rceil \quad \text{and} \tag{11}$$

$$z_i \quad i \in I. \tag{12}$$

$a|b$ signifies the remainder on dividing a by b. It is now necessary to relate x_{cjs}, v_{csi} and r_i:

$$v_{cjs} = \sum_{k=j-24}^{j} A8_k x_{csk} \quad c \in C \; s \in S_c \tag{13}$$

$$r_i = \sum_{c \in C} \sum_{s=1}^{NS_c} B_{cis} v_{cjs} \quad i = j, j+P, j+2P, \ldots, j+6P \; j \in J. \tag{14}$$

In practice, v_{cjs} was eliminated between equations (13) and (14) to reduce the problem's size. The subscripts in constraints (10), (11), (13) and (14) must be interpreted with care, all subtraction of indicies in 14 must be made modulo $7 \times P$ with resulting zeros being replaced by $7 \times P$. This is because 8-hour shifts starting late Saturday (high subscript) will finish early on Sunday (low subscript) of the next week.

Constraint (15) expresses w_i, the service capacity available in any interval $i \in I$ as a function of the eight-hour and four hours shifts starting in earlier relevant periods, and z_i the number of contractors used. Constraint (16), in conjunction with the objective function expresses the fact that u_i, the number waiting for service at the end of period i, is $\max\{0, u_{i-1} + D_i - w_i\}$.

$$w_i = R_i r_i + R_i \sum_{k=i-11}^{i} A4_k y_k + z_i \quad i \in I \tag{15}$$

$$w_i + u_i - u_{i-1} \geq D_i \quad i \in I \tag{16}$$

There will normally be a maximum number (M) of workers available.

$$\sum_{c \in C} L_c q_c \leq M \tag{17}$$

The following bounds are superficially redundant but their presence hastens solution:

$$x_{cjs} \leq M_j \quad c \in C \; j \in J \; s \in S_c. \tag{18}$$

The model was augmented with a number of bounds, for example, lower and upper limits on the proportion of the total demand reserved for contractors (0% and 100% were used) and limits on the number of contractors available in each period of the week (50) were input. A number of variables (e.g.

$\sum_{i \in I} y_i$) had upper bounds and could be eliminated by setting upper bounds to zero. Some extra equality constraints gave useful subtotals such as the total waiting time, total number of jobs done by contractors, total number of staff starting at particular times and components of the total cost.

The combination of (1) through (9) and (13) through (18) and the non-negativity restrictions forms the model, a mixed integer program with thousands of constraints and variables. The model includes P binary variables (δ_j $j \in J$), NC integer variables (q_c $c \in C$) and $\sum_{c \in C} L_c$ Special Ordered Set Type 1 's (the size of a typical problem as modified is given in section 4.2 below).

It is now possible to discuss the reason for choosing a 20-minute time period. The shorter the period is, the more accurate are the answers. However, the number of constraints, binary variables and total variables is each roughly inversely proportional to the period's length. A smaller period would have generated a bigger matrix and taken appreciably longer to solve.

4 Numerical Solution

In its original form the problem was difficult to solve. It was evidently ill-conditioned and had to be perturbed. A desktop package with SOS1 capabilities failed to solve the problem. The package CPLEX 2.0[6], when run on a CRAY computer performed somewhat better primarily because it had a perturbation option. Because it did not have a special ordered set facility many extra variables (γ_{cjs} $c \in C$ $j \in J$ $s \in S_c$) and constraints ((7) and (8)) had to be input. This bigger problem was computationally intractable and failed to find MIP solutions.

A simplified version of the problem could be solved routinely using the package CPLEX 4.1[7] on a CRAY computer model J916. This process is explained in the next two sections.

4.1 Problem Simplification

The problem was simplified in various ways:

- The number of four-hour shifts was set to zero and only one cycle was input.

- It was observed that, when the RHS of (6) was set to 72, only 22 of the $\delta_j \; j \in J$ were non-zero in the solution of the relaxed problem. When an integer solution was sought, only nine of the δ's (a subset of the 22) were non-zero. As the 22 were well spread, the matrix was regenerated with the index set J comprising these 22 values instead of the original $\{1, 2, \ldots 71, 72\}$.

  In a formulation with $P = 72$, and a single 52 week cycle the model had 3201 constraints, 2900 variables (22 binary, 2 integer), 36 SOS1's (each comprising 22 elements) and 98998 non-zero matrix elements.

- Computational problems, especially the need to perturb problems, were eased somewhat by examining (5) which represented $792(22 \times 36)$ constraints and 1584 coefficients. It was decided to replace (5) by

$$\sum_{s \in S_c} x_{cjs} \leq M_j L_c \delta_j \quad c \in C \, j \in J, \tag{19}$$

  yielding 22 constraints and 814 coefficients. This is contrary to the advice given by H. P. Williams[14] — that one should not aggregate constraints involving integer variables, but it worked well here, presumably because the reduction in problem size outweighed the effects of a worse relaxation.

4.2 Hardware and Software

- The package CPLEX 4.1 allowed SOS1's . The problem size was reduced because variables ($\gamma_{cjs} \; c \in C \; j \in J \; s \in S_c$ and constraints (7) and (8)) could be omitted.

- The package had aggregation and reduction facilities, the former (which did 1936 substitutions) can eliminate equality constraints at the cost of increasing matrix density, the latter eliminated 68 rows and 69 columns of the relaxed problem. The problem thus reduced had 1355 rows, 1845 columns and 94322 non-zeros.

- The way in which CPLEX 4.1 selects an entity to arbitrate and the direction of arbitration during the solution of an MIP worked well but were neither exhaustively documented nor perfectly clear[7][p 67].

Overriding the package's default strategy worsened performance. However, specifying priorities (q_c $c \in C$ highest, δ_j $j \in J$, then x_{cjs} $c \in C$ $j \in J$ $s \in S_c$) did help.

- The package was run on CRAY model J916 computer with results and times summarised in table 2. A commercial user would have been charged $A33 for each 1000 seconds of use.

5 Results

Experience showed that, for variants of the current problem, a first integer solution was found quickly but that subsequent solutions, if found at all, required much more computer time (see lines 1 and 2 of table 2). After some experimentation, only one solution was sought for each problem.

Management's first priority was benchmarking current practice. It had been speculated that it would be cheaper to discard permanent employees (paid by the shift whether working or not) and use only contractors (paid by the job). Managers were particularly interested in the total weekly cost, the number of people who ought to be employed on each shift, the distribution of shift starting times, the percentage of jobs done by contractors and the service level (reflected in a the amount of queuing time).

Solution of the relaxed problem required 22.21 seconds and 2332 iterations (235 in phase I); it had an optimal value of $219670. MIP solutions (summarised in table 2) were obtained for various values (22 and 10–1) of the RHS of (6). The first entry in the third column [22(12)] means that, although shifts could start in up to 22 different periods of the day, only 12 periods appeared in the optimal solution. Lines 3-12 give analogous results for RHS of 10–1. The run times (the runs started from the solution of the relaxed problem) are small except when the number of allowable start times is three or less. The optimal value of most of the problems depicted in table 2 is less than 1% more than that of the relaxed problem.

Line 12 incidentally summarises what wil happen if there are no employees, all work being contracted out —the total cost increases by 25% over the base case depicted in line 1. In lines 13 and 14 the number of shifts is set to 1 and 3 (implying 52 and 156 permanent employees) respectively with a very slight increase in total costs. Lines 15 and 16 demonstrate that the solution is sensitive to the mean number of jobs an employ can do per hour. A 10% increase in employee efficiency unsurprisingly diverted work from contractors to employees with a total cost saving of 5%. A 10% decrease in employee efficiency diverted work to contractors from employees with a cost increase of 3%.

6 Comparison with Present Practice

We were able to compare our recommendation with the scheduling now being used. The policy now used is summarised in table 4.

The extant policy was simulated by fixing appropriate variables at appropriate values. In particular, the δ_j $j \in J$ was fixed to 0 or 1 and a set of variables was introduced to store the total number of shifts starting at a particular time. These variables were fixed to the values of the current policy. The organisation now uses half hour intervals, inconsistent with the 20-minute interval used in the model. It was assumed that the number starting at 7.30am were split between 7.20am and 7.40am.

The results from this run and the optimal solution from table 2 for seven starting times are summarised in table 5. Practice is 17% more expensive than what is recommended, we note that changing the number of exployees from 156 to 104 would reduce this discrepancy to 6%.

We are not certain that the current policy is as good as table 5 implies or that the data provided is correct. Anecdotal evidence suggests that at least half of the clients have to queue for at least one hour. A small proportion of waiting times are attributable to customers' calls taking a few minutes to be passed to operators. The cost of queuing supplied by the organisation was rather low ($12/hour), using a higher value drove waiting times to nearly zero.

The Relative Cost of Contractors and Staff There was considerable debate over the relevant costs of the organisation's own staff, in this paper a figure preferred by the most influential department of the client was used, rather than the 30% lower official figure. An employee had to work at 88% or 62% of nominal capacity to be cheaper than day-rate or night-rate contractors respectively.

7 Implementation

Finding a satisfactory solution was not enough. It is desirable for the client's staff to be able to change the input data and run the system with minimal outside help.

We originally envisaged writing a suite of programs which would allow the user to read, vet and store sets of input data. It now seems preferable to store the data in a spreadsheet (a familiar tool). It would be possible to embellish the spreadsheet with data validation procedures. Writing a program which would read the binary file output by the MIP package and write the results into a file which can be read by a spreadsheet package was contemplated. Once the data is in a spreadsheet, it is easy to present, manipulate, summarise and graphically present the answers.

The system is presently running on a CRAY computer, the MPS input file being generated by a Fortran 90 program which reads data files output from the spreadsheet in which the source data is stored. For the foreseeable future a consultant will be required to amend and run the model, the client cannot justify employing a person with the requisite special skills. Purchasing a version of CPLEX 4.1 which will run on a powerful PC is a yet to be explored option. Taking the whole procedure in house would simplify the transfer of files needed to run the system.

The benefits we expect the organisation to get from the system are:

A benchmark The organisation could compare its actual operating costs (so far as they can be ascertained) with a theoretical minimum cost.

Modelling We can test the effect of, for example, changing a starting time on staff utilisation and customer waiting times. There is particular interest in studying proposals arising in negotiations with the unions. Some people in the organisation held views which the model (or elementary considerations) invalidated. One such assumption was that it would be better to have no employees and rely wholly on contractors.

Better cost estimates The author's experience is that organisations often do not get direct benefit from actual use of OR models. They get considerable benefit from being prompted to better ascertain costs and parameters. This prompts them to scrutinise and change their operations, often thereby invalidating the assumptions on which the model is based.

Cost Savings The MIP model includes all costs, including queuing costs and indicates modest savings. We used only one 52 week cycle. Greater savings might come from using more than one cycle or allowing variation of the number of people working in each week of a cycle.

Acknowledgement The author would like to thank Mr Jim Youngman of the client organisation for introducing him to this problem and his help, especially in obtaining data and software: and Mr Chen Qifu (now at Wuhan Iron and Steel University, Wuhan, 430081, Peoples Republic of China) who was the co-author of an earlier version of this paper.

References

[1] Baker, K. R. & Magazine, 'Workforce scheduling with cyclic demands and day-off constraints', *Management Science* 24/2, 1977, pp 161–170.

[2] Bartholdi, J. J., Orlin, J. B. & Ratcliff, H. D., 'Cyclic scheduling via integer programs with circular ones', *Operational Research* 28 (1980) 1075–1085.

[3] Bechtold, Stephen E, Brusco & Michael J., 'Working set generation methods for labour tour scheduling', *European Journal of Operational Research* 74 (1994) 540–551.

[4] Beaumont, N. B., *Using mixed integer program to design employee rosters*, working paper, Department of Business Management, Monash University, 1996.

[5] Bianco, L., Bielli, M., Mingozzi, A., Ricciardelli, S. & Spadoni, M., 'A heuristic procedure for the crew rostering problem', *European Journal of Operational Research* 58 (1992) 272–283.

[6] CPLEX Optimization Inc, *Using the CPLEX callable library and CPLEX mixed integer optimizer (version 2.0)* CPLEX Optimization Inc, USA, 1993

[7] CPLEX Optimization, Inc, *Using the CPLEX callable library and CPLEX mixed integer optimizer (version 4.1)* CPLEX Optimization, Inc, USA, 1995

[8] Easton, F. F. & Rossin, D. F., '.Equivalent alternate solutions for the tour scheduling problem', *Decision Sciences* 22 (1991) 985–996

[9] Hung, R., 'Single-shift off-day scheduling of a hierarchical workforce with variable demands', *European Journal of Operational Research* 78 (1994) 49–57.

[10] Hung, R., 'A multiple-shift workforce scheduling model under the 4-day workweek and weekend labour demands', *Journal of the Operational Research Society* 45 (1994) 1088–1092.

[11] Hung, R., 'Multiple-shift workforce scheduling under the 3-4 workweek with different weekday and weekend labour requirements' *Management Science* 40 (1994) 280–284.

[12] Ryan, D. M., 'The solution of massive generalized set partitioning problems in aircrew rostering', *Journal of the Operational Research Society* 43 (1992) 459–467.

[13] Townsend, W., 'An Approach to Bus-Crew Roster Design in London Regional Transport', *Journal of the Operational Research Society* 39 (1988) 543–550.

[14] Williams, H. P., *Model Building in Mathematical Programming* , John Wiley and Sons, 1993.

Month	Total demand	Average No of calls/day	Rank
1	79503	2565	11
2	85968	3070	1
3	81920	2643	7
4	77863	2595	9
5	87510	2823	4
6	85815	2861	2
7	81921	2642	8
8	80066	2583	10
9	70593	2353	12
10	85317	2752	6
11	85677	2856	3
12	86603	2794	5

Table 1: Summary of Each Month's Demand

Problen no.	Run time (sec)	Nodes explored	Maximum periods	Shifts	Total number of contractors	Cost of contractors	Total waiting time [a]	Waiting cost	Staff cost	Cost of staff travel	Objective ($A)
1	49.95	32	22 (12) [b]	2	8732	116221	259	1036	94000	9078	220335
2[c]	852.97	934	22(12)	2	8724	116155	242	969	94000	9090	220214
3	54.37	36	10 (9)	2	8738	116246	369	1477	94000	9069	220792
4	55.9	74	9	2	8741	116316.5	369	1476	94000	9064.5	220857
5	57.38	77	8(7)	2	8730	116428	372	1487	94000	9081	220996
6	49.37	53	7	2	8807	117031.5	637	2549	94000	8965.5	222546
7	42.88	43	6(5)	2	8803	118419.5	396	1585	94000	8971.5	222976
8	46.75	39	5(4)	2	8896	118324	1047	4188	94000	8832	225344
9	48.37	40	4	2	8843	117645.5	1046	4184	94000	8911.5	224741
10	1267.59	165	3	1	11749	171497.5	419	1676	47000	4552.5	224726
11	1120.7	101	2	1	11744	171488	627	2508	47000	4560	225556
12	923.41	39	1	0	14784	224296	12904	51616	0	0	275912
13[d]	36.98	38	22 (9)	1	11751	169086.5	600	2398	47000	4549.5	223034
14[e]	58.01	54	22(14)	3	5778	67272	327	1308	141000	13509	223089
15[f]	55.89	132	22(15)	3	4904	55042	361	1445	141000	14820	212307
16[g]	55.23	48	22(4)	1	12048	178350	1023	4094	47000	4104	229444

[a] 20 min periods

[b] This notation means that, although 22 time slots were available, only 12 were used.

[c] This is the same problem as 1 but this is the best solution found in 1000 seconds.

[d] Number of shifts was fixed at 1.

[e] Number of shifts was fixed at 3.

[f] Efficiency raised by 10%.

[g] Efficiency reduced by 10%.

Table 1: Results of various runs

Number of starting times	Shifts	Starting times and number of shifts
22 (12)	2	16/8[a] 23/18 24/4 40/8 41/4 43/6 44/2 47/6 50/4 52/4 54/6 60/2
10 (9)	2	7/2 16/8 24/20 40/8 41/8 44/4 47/12 54/8 56/2
9	2	7/2 16/8 24/20 40/10 41/8 47/14 54/8 56/2
8(7)	2	7/2 16/8 24/20 40/18 47/16 54/6 56/2
7	2	7/6 24/20 40/20 47/16 54/8 56/2
6(5)	2	23/20 24/20 47/20 54/10 56/2
5(4)	2	24/20 41/20 47/20 54/12
4	2	24/20 40/20 47/20 54/12
3	1	22/16 23/6 41/14
2	1	23/20 40/16
1	0	nil

[a] 16 is the start period (period 1 is 00:01am-00:20am) and 8 is the number of people starting work at that time

Table 3: Relation between the number of starting times and shifts

Starting times	Number of staff starting at this time
6.01am	9
7.01am	18
7.31am	21
8.01am	12
14.31pm	9
15.01pm	12
16.01pm	27

Table 4: Present Policy

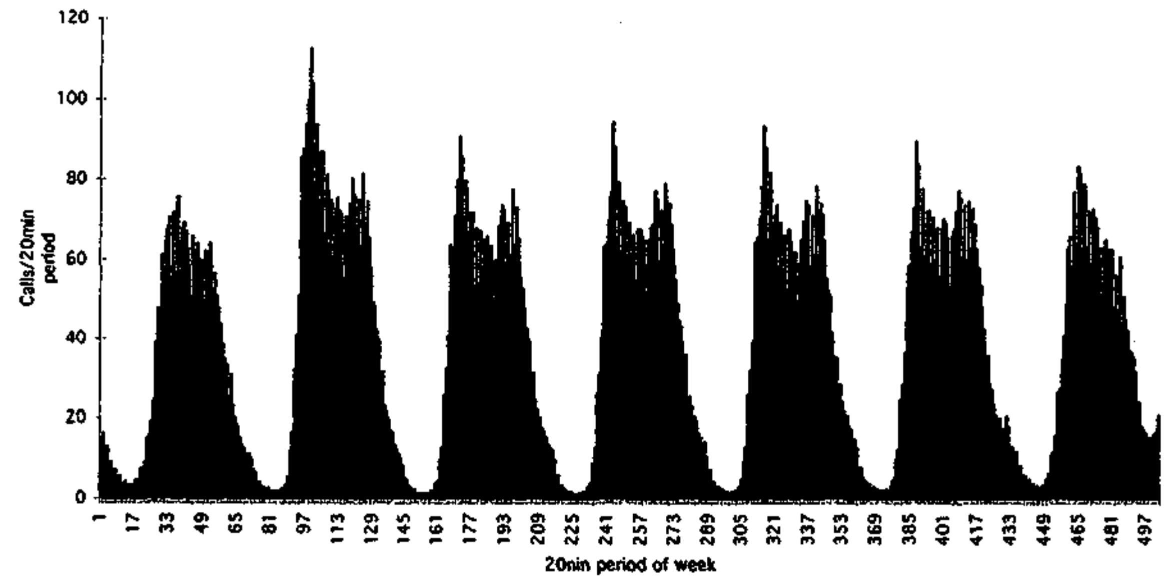

Figure 1: The Demand in Each 20-Minute Period of the Week

	Optimal Policy with 7 Starting Times Allowed	Current Policy with 7 Starting Times Allowed
Total number queueing	637	1,841
Total cost of queueing	$2549	$7,362
Cost of staff	$94,000	$141,000
Cost of travel between jobs	$8,966	$20,698
Cost of contractors	$117031	$92,427
Jobs done by contractors	8807	5,914
Total weekly cost	$222,546	$261,487

Table 5: Comparison of Present and Proposed Policies