OR Spektrum (2001) 23: 283–294

OR Spektrum

Efficient optimization of cyclic labor days-off scheduling*

Effizientes Optimierungsverfahren zur zyklischen Einsatzplanung

Hesham K. Alfares

Systems Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia (e-mail: hesham@ccse.kfupm.edu.sa)

Received: November 2, 1999 / Accepted: May 30, 2000

Abstract. An efficient method is presented for the optimum solution of the cyclic manpower days-off scheduling problem. This method, which is based on linear programming, allows unequal costs to be considered for different days-off patterns. First, the solution of the dual linear programming model is used to determine the minimum workforce size. Then, a procedure based on the dual solution is introduced to assign the workforce to days-off patterns in order to minimize the total labor cost. The new method offers an alternative to specialized linear or integer programming software, since it requires only few and simple calculations.

Zusammenfassung. Vorgestellt wird ein effizientes Optimierungsverfahren zur zyklischen Personaleinsatzplanung unter Beachtung freier Arbeitstage. Das auf der linearen Programmierung aufbauende Verfahren berücksichtigt unterschiedliche Kosten für die verschiedenen Wochenarbeitspläne. Zunächst wird auf der Grundlage des dualen LP-Modells die minimale Personalstärke bestimmt. Darauf aufbauend wird ein Lösungsverfahren vorgeschlagen, um die Arbeitskräfte auf die verschiedenen Wochenarbeitspläne aufzuteilen bzw. ihnen unterschiedliche freie Tage zuzuordnen. Die Zielsetzung besteht darin, die gesamten Lohnkosten zu minimieren. Das neue Lösungsverfahren stellt eine Alternative zum Einsatz spezieller linearer bzw. ganzzahliger Optimierungssoftware dar, da es nur wenige und einfache Rechenschritte benötigt.

* The author wishes to express gratitude to King Fahd university of Petroleum and Minerals for supporting this research effort, and to two anonymous referees for their constructive comments on an earlier version of this paper.

Key words: Manpower scheduling – Optimization – Mathematical programming – Labor planning – Personnel management – Staffing

Schlüsselwörter: Personaleinsatzplanung – Optimierung – Mathematische Programmierung – Arbeitsplanung – Personalmanagement

Introduction

Manpower days-off scheduling is a practical problem that arises in organizations that operate seven days a week, such as airlines, hospitals, and police stations. Since workers must be given weekly breaks, they must be assigned to specific days-off patterns. The objective is to determine the number of workers assigned to each pattern, in order to satisfy daily labor demands at the minimum cost or with the minimum workforce size.

The most commonly used days-off patterns include two consecutive days off per week. There are 7 such patterns for any given week; because of the cyclic nature of the problem, Sunday and Monday are considered as a pair of consecutive off days. Since each pattern includes 5 workdays per week, the problem is usually referred to as the (5, 7) days-off scheduling problem. The (5, 7) problem is relatively small, with 7 constraints (daily labor demands) and 7 variables (days-off assignments).

In this age of advanced computing technology, the (5, 7) problem can be solved very quickly on the personal computer, using any integer programming (IP) software. To justify the need for a new solution method, two reasons are given. First, in the case of solving the (5, 7) problem as a subproblem within a larger program, there would be no need for specialized IP subroutines, which are not generally available. Second, there might be a need to solve a large number of these problems. For example, to simultaneously schedule tasks and labor for a small project, Alfares and Bailey [2] had to solve thousands of days-off scheduling problems. In such cases, the computational efficiency of the new method will be a significant advantage.

This paper is organized as follows. First, a review of relevant literature is given. Then, the integer programming models of the (5, 7) problem and its dual are presented. Subsequently, the procedure for determining the minimum workforce size is described. This is followed by a detailed development of the proposed method of assigning workers to days-off patterns. Next, an example is solved, and then conclusions and recommendations are given. Finally, a description of the algorithm steps is given in the Appendix.

Literature review

Labor scheduling problems are traditionally classified into three types: (1) shift, or working time of the day, scheduling, (2) days-off, or working days of the week, scheduling, and (3) tour scheduling, which combines the first two types. Tien and Kamiyama [20], Bedworth and Bailey [9], and Nanda and Browne [17] provide comprehensive surveys of literature on all these types. The scope of this review is limited to the days-off scheduling problem.

Dantzig's [14] set covering integer programming model is adapted by Baker [4] to represent the days-off scheduling problem. Monroe [15] uses a simple trial-and-error algorithm to maximize consecutive pairs of days off. Rothstein [18] utilizes mathematical programming to formulate and solve the same problem. Chen [13] uses a three-stage manual procedure to obtain the solution. Burns [11] develops techniques to minimize the workforce size, giving each worker four off days in a two-week period. Burns and Carter [12] determine the schedule with the minimum workforce size to satisfy three conditions for each employee: (1) five workdays per week, (2) A out of every B weekends off, and (3) no more than six consecutive workdays.

Several approaches have been developed for the (5, 7) problem, in which each worker must work five days per week, but only consecutive pairs of off days are allowed. Tiberwala et al. [19] presents a three-step procedure in which the number of iterations equals the number of workers required. Browne and Tiberwala [10] simplify the three steps involved, but do not reduce the number of iterations. Baker [3] develops a two-phase algorithm, which starts by calculating the lower bound on a workforce size, and then uses trial and error on a special tableau to determine days-off assignments.

Several techniques based on continuous linear programming (LP) relaxation of the integer-valued (5, 7) problem have been developed [5, 6, 7]. Bartholdi III and Ratliff [7] solve the problem as a finite set of network flow matching problems. For row-circular constraint matrices, Bartholdi III et al. [6] use a simple rounding technique to obtain the optimum integer solution from the solution of the LP relaxation. For problems in which the matrix is not row circular, an LP round-off heuristic is developed by Bartholdi III [5].

Morris and Showalter [16] describe an iterative, three-step cutting plane procedure for optimally solving the (5, 7) problem. Another iterative, manual procedure utilizing three simple rules is presented by Bechtold and Showalter [8]. The objective of both procedures is to minimize the workforce size. Vohra [21] develops an expression for the minimum workforce size of the (5, 7) problem in terms of daily labor demands. Alfares [1] relates Vohra's [21] expression to Baker's [3] lower bound, and performs a computational comparison of integer programming and LP-based methods.

Integer programming model

The (5, 7) problem assigns workers to the 7 days-off patterns with 2 successive days off per week, so that daily labor demands are satisfied with the minimum number or cost of workers. With the objective of minimizing the number of workers, the (5, 7) days-off scheduling problem can be represented as an integer linear programming model, as follows:

$$\text{Minimize} W = \sum_{j=1}^{7} x_j \tag{1}$$

subject to

$$\left(\sum_{j=1}^{7} x_j\right) - x_{i-1 \bmod 7} - x_i \geq r_i, i = 1, 2, ..., 7 \tag{2}$$

$$x_j \geq 0 \text{ and integer}, \quad j = 1, 2, ..., 7 \tag{3}$$

where

W = workforce size, i.e., total number of workers assigned
r_i = number of workers required on day i, $i = 1, 2, ..., 7$
x_j = number of workers assigned to weekly days-off pattern j, i.e., off on days j and $j + 1$ mod 7, where day 1 = Monday and day 7 = Sunday

Bartholdi III and Ratliff [7] modify the above formulation to obtain a "complementary model" with a sparser matrix. Since $\sum_{j=1}^{7} x_j$ is equal to W, (2) is written as follows:

$$x_{i-1 mod 7} + x_i \leq b_i \ , \qquad i = 1, 2, ..., 7 \tag{4}$$

where

$$b_i = W - r_i \ , \qquad i = 1, 2, ..., 7 \tag{5}$$

The dual of the LP relaxation of the days-off scheduling model, with dual variables y_i, $i = 1, 2, ..., 7$, is given by:

$$\text{maximize } W = \sum_{i=1}^{7} r_i y_i \tag{6}$$

subject to

$$\left(\sum_{i=1}^{7} y_i\right) - y_j - y_{j+1 \bmod 7} \leq 1 \ , \quad j = 1, 2, ..., 7 \tag{7}$$

$$y_i \geq 0 \ , \qquad i = 1, 2, ..., 7$$

Determining minimum workforce size

Given seven daily labor demands, the minimum workforce W can be easily obtained without using integer programming. Vohra [21] uses the dual formulation above to show that the minimum W is given by:

$$W = \sum_{j=1}^{7} x_j = \max\left\{r_{\max}, \left\lceil \frac{1}{5}\sum_{i=1}^{7} r_i \right\rceil, \left\lceil \frac{R_{\max}}{3} \right\rceil\right\} \tag{8}$$

where

$r_{\max} = \max\{r_1, r_2, \ldots, r_7\}$
$R_{\max} = \max\{R_1, R_2, \ldots, R_7\}$
$\lceil a \rceil =$ smallest integer $\geq a$

Table 1. Sets of subscripts defined by (10) and their complements

i	S_i	$\overline{S_i}$
1	1,2,4,6	3,5,7
2	2,3,5,7	1,4,6
3	3,4,6,1	2,5,7
4	4,5,7,2	1,3,6
5	5,6,1,3	2,4,7
6	6,7,2,4	1,3,5
7	7,1,3,5	2,4,6

and

$$R_i = \sum_{j \in S_i} r_j \ , \qquad i = 1, 2, \ldots, 7 \tag{9}$$

$$S_i = \text{ set of 4 subscripts } \subset \{1, 2, 3, 4, 5, 6, 7\} \ , \qquad i = 1, 2, \ldots, 7 \tag{10}$$

Sets S_1 to S_7 are shown in Table 1.

After determining the value of W by Equation (8), the values of x_j can be found by using linear programming. Bartholdi III et al. [6] proposes replacing the integrality constraint (3) by the following workforce size constraint:

$$\sum_{j=1}^{7} x_j = W \tag{11}$$

The resulting continuous linear program is then solved to obtain the integer values of x_j. According to Bartholdi III et al. [6], the solution is guaranteed to be integer. A simple yet optimum solution technique which does not involve either linear or integer programming will be developed next, using linear programming duality concepts.

Assigning workers to days-off patterns

In the first stage of the algorithm, the minimum workforce size W is determined by (8). The objective now is to determine the number of workers assigned to each pattern: $x_1, \ldots, x_7$. Basic primal-dual relationships will be used to exploit information from the dual solution for obtaining the solution of the primal (original) days-off scheduling problem. The solution will depend on which argument of the right-hand side of (8) is maximum. Thus excluding ties, there are three possible cases.

Case 1. $r_{\max}$ is maximum

If $r_i = r_{\max}$ is the maximum argument in (8), then $W = r_i$. Substituting into (6) we have:

$\sum_{i=1}^{7} r_i y_i = r_i$. Therefore $y_i = 1, \quad \cup y_{j \neq i} = 0,$ and $\sum_{i=1}^{7} y_i = 1$.

Substituting into (7) we get:

$$1 - y_j - y_{j+1 \bmod 7} \leq 1 \ , \quad j = 1, 2, \ldots, 7$$

Since all $y_{j \neq i}$ are equal to zero except y_i which is equal to 1, the above constraints are satisfied as equations excepts if $j = i-1$ or $j = i$. Therefore, only one dual variable ($y_i = 1$) is basic and only two dual constraints ($i - 1$ and i) are inequalities. Thus the primal problem has one equation i and two variables equal to zero: x_{i-1} and x_i. Including this information in system (4), we have many alternatives for assigning workers to days off patterns to ensure feasibility. For example, the following rules assign as many workers as possible to x_6 then to x_5 and x_7, keeping in mind that the total assignment cannot exceed W. These rules are also applicable in the case of ties for the maximum demand value $r_i = r_{\max}$.

$$\begin{aligned}
x_6 &= \min\{b_6, b_7\} \\
x_5 &= \min\{b_5, b_6 - x_6, W - x_6\} \\
x_7 &= \min\{b_1, b_7 - x_6, W - x_5 - x_6\} \\
x_4 &= \min\{b_4, b_5 - x_5, W - x_5 - x_6 - x_7\} \\
x_3 &= \min\{b_3, b_4 - x_4, W - x_4 - x_5 - x_6 - x_7\} \\
x_2 &= \min\{b_2, b_3 - x_3, W - x_3 - x_4 - x_5 - x_6 - x_7\} \\
x_1 &= \min\{b_1 - x_7, b_2 - x_2, W - x_2 - x_3 - x_4 - x_5 - x_6 - x_7\}
\end{aligned} \tag{12}$$

Case 2. $\left\lceil \frac{1}{5} \sum_{i=1}^{7} r_i \right\rceil$ *is maximum*

If $\lceil \Sigma r_i/5 \rceil$ is the maximum argument in (8), then $W = \lceil \Sigma r_i/5 \rceil$. Substituting into (6) we have:

$$\sum_{i=1}^{7} r_i y_i = \lceil \Sigma r_i/5 \rceil \ . \quad \text{Therefore,} \quad y_i = 1/5 \ , \quad i = 1, 2, \ldots, 7 \ ,$$

$$\text{and} \quad \sum_{i=1}^{7} y_i = 7/5 \ .$$

Substituting into (7) we get:

$$7/5 - 1/5 - 1/5 = 1 \ , \quad j = 1, 2, \ldots, 7$$

In this case, all dual variables are basic ($y_i = 0.2$, $i = 1, 2, \ldots, 7$), and all dual constraints are equations. Therefore all primal variables are also basic and all primal constraints are equations. Constraint system (4) is transformed into the following set of equations:

$$x_{i-1} + x_i = b_i \ , \quad i = 1, 2, \ldots, 7 \tag{13}$$

The solution of the 7×7 linear system of equations is given by:

$$x_i = W - \sum_{j \in \overline{S}_i} b_j \ , \quad i = 1, 2, \ldots, 7 \tag{14}$$

where

$\overline{S_i} =$ the complement of S_i , shown in Table 1

It is more convenient to use equation (14) only once to find x_1. Using (13), the remaining variables can then be calculated more easily as follows:

$$x_1 = W - b_3 - b_5 - b_7 \tag{15}$$

$$x_i = b_i - x_{i-1} \ , \quad i = 2, 3, \ldots , 7 \tag{16}$$

Case 3. $\left\lceil \frac{R_{\max}}{3} \right\rceil$ *is maximum*

Let $R_i = R_{\max}$. In this case, $W = \lceil R_i/3 \rceil$, four dual variables ($y_j = 1/3, j \in S_i$) are basic, and one dual constraint (i) is an inequality. Thus the primal problem has 4 equations ($j \in S_i$) and one variable equal to zero: x_i. Including this information in system (4), and incorporating modularity conditions, we obtain:

$$\begin{aligned} x_{i-1} &= b_i \\ x_i &= 0 \\ x_{i+1} &= b_{i+1} \end{aligned} \tag{17}$$

and

$$\begin{aligned} x_{i+2} &\le b_{i+2} - b_{i+1} \\ x_{i+2} + x_{i+3} &= b_{i+3} \\ x_{i+3} + x_{i+4} &\le b_{i+4} \\ x_{i+4} + x_{i+5} &= b_{i+5} \\ x_{i+5} &\le b_{i+6} - b_i \end{aligned} \tag{18}$$

System (17) specifies the assignments for 3 patterns only. If all the patterns are treated equally, a feasible assignment for the 4 remaining patterns can be easily found by:

$$\begin{aligned} x_{i+2} &= \min \{b_{i+2} - b_{i+1}, b_{i+3}\} \\ x_{i+3} &= b_{i+3} - x_{i+2} \\ x_{i+4} &= \min \{b_{i+4} - x_{i+3}, b_{i+5}\} \\ x_{i+5} &= b_{i+5} - x_{i+4} \end{aligned} \tag{19}$$

If ties exist for the maximum value $R_i = R_{\max}$, any of the systems corresponding to the applicable values of the index i can be chosen arbitrarily.

Minimizing total labor cost

The costs of different days-off patterns are related to the number of overtime-paid weekend workdays. Days-off pattern 6 is the cheapest with no weekend workdays, followed by patterns 5 and 7 with one weekend workday each. The remaining patterns have the highest cost with two weekend workdays each. Let us assume

that a regular-week workday costs $\$A$ per worker, and a weekend workday costs $\$A(1 + B)$ per worker (A, $B \geq 0$). In this case, the *relative* weekly costs of the seven days-off patterns are given by:

$$c_1, \ldots, c_7 = \{5 + 2B, 5 + 2B, 5 + 2B, 5 + 2B, 5 + B, 5, 5 + B\} \tag{20}$$

To incorporate these different costs, the right-hand side vector of the dual constraints (7) changes to the transposed cost vector $\{c_1, \ldots, c_7\}^T$, modifying the 3 dual solutions obtained with uniform costs. However, as long as $B \leq 5$, the change is slight and does not affect the values of workforce size W determined by (8). This means that for this cost structure, the minimum *cost* in all 3 cases is always obtained with the minimum *number* of workers. If the varying costs of different days-off patterns were taken into consideration, the objective would be to minimize the total *cost* of workers, or,

$$\text{minimize } Z = \sum_{j=1}^{7} c_j x_j \tag{21}$$

This objective would be accomplished by assigning as many workers as possible, out of W, to the cheapest patterns: x_6 then to x_5 and x_7. This has already been done for Case 1, when $W = r_{\max}$. In Case 2, when $W = \lceil \Sigma r_i/5 \rceil$, days-off assignments ($x_1, \ldots, x_7$) are unique values obtained by solving a set of linear equations, and are not affected by pattern costs. In Case 3, when $W = \lceil R_{\max}/3 \rceil$, the above general solution for any $R_i = R_{\max}$, given by (17) and (19), may not be applicable. Therefore, individual cases for each i, $i = 1, 2, \ldots, 7$, must be considered

Table 2. Values of days-off assignments, $x_1, \ldots, x_7$, for all possible values of W

No	W	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	$\lceil R_1/3 \rceil$	0	b_2	$b_4 - x_4$	$\min\{b_4, b_5 - x_5\}$	$b_6 - x_6$	$\min\{b_6, b_7 - b_1\}$	b_1
2	$\lceil R_2/3 \rceil$	b_2	0	b_3	$b_5 - x_5$	$\min\{b_5, b_6 - x_6\}$	$\min\{b_7, R_2 - R_6\}$	$b_7 - x_6$
3	$\lceil R_3/3 \rceil$	$b_1 - x_7$	b_3	0	b_4	$b_6 - x_6$	$\min\{b_6, R_3 - R_6\}$	$\min\{b_1, b_7 - x_6\}$
4	$\lceil R_4/3 \rceil$	$\min\{b_1, b_2\}$	$b_2 - x_1$	b_4	0	b_5	$\min\{b_7, b_6 - b_5\}$	$b_7 - x_6$
5	$\lceil R_5/3 \rceil$	$b_1 - x_7$	$\min\{b_2, b_3\}$	$b_3 - x_2$	b_5	0	b_6	$\min\{b_1, b_7 - b_6\}$
6	$\lceil R_6/3 \rceil$	$b_2 - x_1$	$\min\{b_2, R_6 - R_2\}$	$\min\{b_4, b_3 - x_2\}$	$b_4 - x_3$	b_6	0	b_7
7	$\lceil R_7/3 \rceil$	b_1	$b_3 - x_3$	$\min\{b_3, b_4 - x_4\}$	$b_5 - x_5$	$\min\{b_5, b_6 - b_7\}$	b_7	0
8	$\lceil \Sigma r_i/5 \rceil$	$W - b_3 - b_5 - b_7$	$b_2 - x_1$	$b_3 - x_2$	$b_4 - x_3$	$b_5 - x_4$	$b_6 - x_5$	$b_7 - x_6$
9	$r_{\max}$	$\min\{b_1 - x_7, b_2 - x_2, W - \sum_{i=2}^{7} x_i\}$	$\min\{b_2, b_3 - x_3, W - \sum_{i=3}^{7} x_i\}$	$\min\{b_3, b_4 - x_4, W - \sum_{i=4}^{7} x_i\}$	$\min\{b_4, b_5 - x_5, W - \sum_{i=5}^{7} x_i\}$	$\min\{b_5, b_6 - x_6, W - x_6\}$	$\min\{b_6, b_7\}$	$\min\{b_1, b_7 - x_6, W - x_5 - x_6\}$

separately. To illustrate the procedure, the case where R_2 is maximum is discussed in detail below. Results for the remaining cases, using a similar methodology, are summarized in Table 2.

Sample derivation: R_2 is maximum

Based on the dual solution, variable x_2 is equal to zero and constraints $S_2 = (2, 3, 5, 7)$ are equations. Using $i = 2$, system (17) provides $x_1 = b_2$ and $x_3 = b_3$, while system (18) gives:

$$x_4 \leq b_4 - b_3 \tag{22.1}$$
$$x_4 + x_5 = b_5 \tag{22.2}$$
$$x_5 + x_6 \leq b_6 \tag{22.3}$$
$$x_6 + x_7 = b_7 \tag{22.4}$$
$$x_7 \leq b_1 - b_2 \tag{22.5}$$

To maximize x_6, subject to (22.3) and (22.4), we set:

$$x_6 = \min\{b_6 - x_5, b_7\}$$

However, from subtracting (22.1) from (22.2) we get:

$$x_5 \geq b_5 - b_4 + b_3$$

thus

$$\begin{aligned} x_6 &\leq b_6 - x_5 \leq b_6 + b_4 - b_5 - b_3 \\ &= W - r_6 + W - r_4 - (W - r_5) - (W - r_3) = (r_3 + r_5) - (r_4 + r_6) \\ &= (r_2 + r_3 + r_5 + r_7) - (r_2 + r_4 + r_6 + r_7) = R_2 - R_6 \end{aligned}$$

Finally, days-off assignments are given by:

$$x_6 = \min\{R_2 - R_6, b_7\}$$
$$x_7 = b_7 - x_6$$
$$x_5 = \min\{b_5, b_6 - x_6\}$$
$$x_4 = b_5 - x_5$$

It is important to note that the workforce size is still the minimum given by (8):

$$\begin{aligned} W &= x_1 + x_3 + (x_4 + x_5) + (x_6 + x_7) \\ &= b_2 + b_3 + b_5 + b_7 \\ &= (W - r_2) + (W - r_3) + (W - r_5) + (W - r_7) \\ &= 4W - (r_2 + r_3 + r_5 + r_7) = 4W - R_2 \end{aligned}$$

Thus

$$W = R_2/3$$

An example

Based on the preceding development, the algorithm steps are summarized in the Appendix. An example is solved below to illustrate the calculations involved in these steps. The resulting solution matches the one obtained by integer programming.

Given seven daily labor requirements for a given week,

$r_1, \ldots, r_7 = 20, 1, 10, 19, 7, 19, 13$

Calculations are carried out using the following table.

i	1	2	3	4	5	6	7	Sum
r_i	20	1	10	19	7	19	13	89
R_1	20	1	–	19	–	19	–	59
R_2	–	1	10	–	7	–	13	31
R_3	20	–	10	19	–	19	–	68
R_4	–	1	–	19	7	–	13	40
R_5	20	–	10	–	7	19	–	56
R_6	–	1	–	19	–	19	13	52
R_7	20	–	10	–	7	–	13	50

$r_{\max} = r_1 = 20$

$\lceil \Sigma r_i/5 \rceil = \lceil 89/5 \rceil = \lceil 17.8 \rceil = 18$

$\lceil R_{\max}/3 \rceil = \lceil R_3/3 \rceil = \lceil 68/3 \rceil = \lceil 22.67 \rceil = 23$

Using (8),

$W = \max\{20, 18, 23\} = 23 = \lceil R_3/3 \rceil$

Since $R_3/3 = 22.67$ is not an integer, we must make it integer by incrementing one of labor requirements corresponding to set S_3 (r_1, r_3, r_4, or r_6) by 1. Based on the criteria specified in step 2(a) of the algorithm, we choose to increase r_3, thus $r'_3 = 11$.

Using Equation (5), we obtain:

$b_1, \ldots, b_7 = 3, 22, 12, 4, 16, 4, 10$

By incrementing r_3 by 1, R_2, R_3, R_5, and R_7 are also increased by the same amount. Since $W = \lceil R_3/3 \rceil$, we use system (3) in Table 2 to find days-off assignments as follows:

$x_2 = 12$

$x_3 = 0$

$x_4 = 4$

$x_6 = \min\{4, 69 - 52\} = \min\{4, 17\} = 4$

$x_5 = 4 - 4 = 0$

$x_7 = \min\{3, 10 - 4\} = \min\{3, 6\} = 3$

$x_1 = 3 - 3 = 0$

Conclusions

This paper has presented a new, efficient optimization algorithm for the cyclic (5, 7) days-off scheduling problem. This LP-based algorithm can be used to minimize either the total number or the total cost of workers assigned. A solved example has been provided to illustrate the simple calculation steps. Avoiding the need to use mathematical programming provides computational efficiency and ease of use. Moreover, the algorithm is easy to program, removing the need for specialized integer programming software. This advantage is especially significant if the algorithm is coded as a module within a larger program.

The algorithm presented in this paper applies only to the (5, 7) days-off scheduling problem because of the special structure of the constraint matrix. However, a similar methodology could possibly be used for solving some other labor scheduling problems in which the dual solution is easily obtained. In particular, a similar approach could be generalized to solve the general (k, n) cyclic staffing problem.

Appendix. Algorithm steps

1. Determine the minimum workforce size W using equation (8)
   $W = \max\{r_{\max}, \lceil \Sigma r_i/5 \rceil, \lceil R_{\max}/3 \rceil\}$. In the case of ties go to step 3
2. (a) If $\max\{r_{\max}, \lceil \Sigma r_i/5 \rceil, \lceil R_{\max}/3 \rceil\} = \lceil R_{\max}/3 \rceil$, then:
   - if $R_{\max}/3$ is not integer, increment $R_i = R_{\max}$ by $(3W - R_{\max})$ to make it a multiple of 3; among the four daily labor demands r_j, $j \in S_i$, that can be increased, avoid whenever possible: (1) weekend, i.e., r_6 and r_7; and (2) the maximum labor demand $r_{\max}$.
   - calculate b_1, b_2, ..., b_7 using equation (5), then apply system No. i, $i = 1, \ldots, 7$, in Table 2 to find $x_1, x_2, \ldots, x_7$. If there are ties for maximum value $R_i = R_{\max}$, use any of the systems No. i corresponding to the applicable values of the index i arbitrarily.

   (b) If $\max\{r_{\max}, \lceil \Sigma r_i/5 \rceil, \lceil R_{\max}/3 \rceil\} = \lceil \Sigma r_i/5 \rceil$, then:
   - if $\Sigma r_i/5$ is not integer, increment Σr_i by $(5W - \Sigma r_i)$ in order to make it a multiple of 5; among all seven daily labor demands $r_1, \ldots, r_7$, choose the ones to be increased according to the criteria given above in step 2(a).
   - calculate b_1, b_2, ..., b_7 using equation (5), then apply system No. 8 in Table 2 to find x_1, x_2, ..., x_7.

   (c) If $\max\{r_{\max}, \lceil \Sigma r_i/5 \rceil, \lceil R_{\max}/3 \rceil\} = r_{\max}$, then:
   - calculate b_1, b_2, ..., b_7 using equation (5), then apply system No. 9 in Table 2 to find x_1, x_2, ..., x_7.
3. In the case of ties for $\max\{r_{\max}, \lceil \Sigma r_i/5 \rceil, \lceil R_{\max}/3 \rceil\}$, choose the argument that needs the minimum total increment and go to the corresponding step: 2.(a) for $\lceil R_{\max}/3 \rceil$, 2.(b) for $\lceil \Sigma r_i/5 \rceil$, and 2.(c) for $r_{\max}$. While $r_{\max}$ does not need incrementing, the increment for $\lceil \Sigma r_i/5 \rceil$ is $(5W - \Sigma r_i)$, and the increment for $\lceil R_{\max}/3 \rceil$ is $(3W - R_{\max})$.

References

1. Alfares HK (1998) An efficient two-phase algorithm for manpower days-off scheduling. Computers & Operations Research 25: 913–923
2. Alfares HK, Bailey JE (1997) Integrated project task and manpower scheduling. IIE Transactions 29: 711–718
3. Baker KR (1974) Scheduling a full-time workforce to meet cyclic staffing requirements. Management Science 20: 1561–1568
4. Baker KR (1976) Workforce allocation in cyclical scheduling problems: A survey. Operational Research Quarterly 27: 155–167
5. Bartholdi III JJ (1981) A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. Operations Research 29: 501–510
6. Bartholdi III JJ, Orlin JB, Ratliff HD (1980) Cyclic scheduling via integer programs with circular ones. Operations Research 28: 1074–1085
7. Bartholdi III JJ, Ratliff HD (1978) Unnetworks, with applications to idle time scheduling. Management Science 24: 850–858
8. Bechtold SE, Showalter MJ (1985) Simple manpower scheduling methods for managers. Production and Inventory Management 26: 116–133
9. Bedworth DD, Bailey JE (1987) Integrated production control systems: Management, analysis, design, 2nd edn, pp 387–420. Wiley, New York
10. Browne JE, Tiberwala RK (1975) Manpower scheduling. AIIE Transactions 7: 22–23
11. Burns RN (1978) Manpower scheduling with variable demands and alternative weekends off. INFOR 16: 101–111
12. Burns RN, Carter MW (1985) Workforce size and single-shift schedules with variable demands. Management Science 31: 599–607
13. Chen D-S (1978) A simple algorithm for a workforce scheduling model. AIIE Transactions 10: 244–251
14. Dantzig GB (1954) A comment on Edie's traffic delays at toll booths. Operations Research 2: 339–341
15. Monroe G (1970) Scheduling manpower for service operations. Industrial Engineering 2: 10–17
16. Morris GM, Showalter MJ (1983) Simple approaches to shift, days-off and tour scheduling problems. Management Science 29: 942–950
17. Nanda R, Browne J (1992) Introduction to employee scheduling. Van Nostrand Reinhold, New York
18. Rothstein M (1972) Scheduling manpower by mathematical programming. Industrial Engineering 4: 29–33
19. Tiberwala R, Philippe D, Browne J (1972) Optimal scheduling of two consecutive idle periods. Management Science 19: 71–75
20. Tien JM, Kamiyama A (1982) On manpower scheduling algorithms. SIAM Review 24: 275–287
21. Vohra RV (1987) The cost of consecutivity in the (5, 7) cyclic staffing problem. IIE Transactions 29: 296–299