IMA Journal of Mathematics Applied in Business and Industry (2000) **11**, 269–283

Dual-based optimization of cyclic four-day workweek scheduling

HESHAM K. ALFARES†

System Engineering Department, King Fahd University of Petroleum & Minerals, PO Box 2003, Dhahran 31261, Saudi Arabia

An optimization method is presented for the cyclic labour days-off scheduling problem, in which workers are given three consecutive days off per week. This method does not include linear or integer programming, and it does not assume that the costs of different days-off work patterns are equal. The dual problem is first solved to determine the minimum workforce size. Then, the dual solution is used to determine days-off assignments that minimize the total labour cost. By requiring only simple manual calculations, the new method eliminates the need for linear or integer programming software.

Keywords: workforce scheduling; optimization; integer programming; labour planning; staffing.

Introduction

Effective scheduling can reduce the cost of labour, which is usually the most expensive resource of any organization. Days-off scheduling is a practical problem for organizations that operate seven days a week, such as airlines, hospitals and police stations. Because workers must be given weekly breaks, they must be assigned to specific days-off work patterns. The objective is to determine the number of workers assigned to each pattern, in order to satisfy daily labour demands at minimum cost or with minimum workforce size.

The most common type of days-off work pattern includes two consecutive days off per week. As each pattern includes five workdays per week, the problem is usually referred to as the (5, 7) days-off scheduling problem. Recently, there has been a lot of interest in compressed workweek schedules. For example, Browne & Nanda (1987); Gould (1988); Hung (1993, 1994) and Hung & Emmons (1993) analysed three-day and four-day workweeks. Moreover, Kogi & Thurman (1993) reported a general international trend towards irregular schedules and shorter work hours.

Organizations are demanding more productivity and efficiency of the workforce, while employees are requesting higher flexibility and longer leisure and family time. Therefore, four-day workweeks are becoming increasingly common in many companies. Lankford (1998), for example, described a real-life application of a four-day workweek schedule at Hewlett Packard (HP). The Analytical Central Call Management (CCM) group at HP recently implemented a compressed workweek, consisting of four ten-hour days (4×10) per week. This implementation is attributed to HPs appreciation of the workforce needs

† E-mail: hesham@ccse.kfupm.edu.sa

and the competitive challenges of the global economy, which compel businesses to use work time more efficiently.

This paper extends the work done by Alfares (1998) for the five-day workweek (5, 7) problem, to the four-day workweek or the (4, 7) problem. A simple, yet optimum, solution method is presented to obtain either the minimum number or the minimum cost of the workforce, using the easily obtainable dual solution and primal–dual relations. It is assumed that each employee is given four workdays and three consecutive days off per week. According to Nanda & Browne (1992, p. 28), workers do not like fragmented days off; thus, the three days off are required to be consecutive, satisfying employee preferences and certain labour rules. While this may lead to a slightly larger workforce, it improves employee morale and productivity. Moreover, giving workers more than one weekly break can result in disruption of workflow and loss of continuity.

The new method offers a number of advantages over integer programming (IP). First, while IP requires specialized training and the availability of certain software packages, the new method can be implemented manually. This advantage is important for small businesses that have no computers. Second, the new method is more computationally efficient than IP, which usually involves several iterations. This is a significant benefit if a large number of days-off problems must be solved. Finally, unlike IP, the new method is easy to program in any language as a module within a larger program. Several practical applications, such as the simultaneous scheduling of tasks and labour for projects (Alfares & Bailey, 1997) or job shops (Aardal & Ari, 1987), require the optimum solution of days-off problems within larger, multifunctional programs. The new method is the only practical choice in such cases, because IP is not easy to code, and IP subroutines are generally not available in most programming languages.

The remainder of this paper is organized as follows. First, a review of relevant literature is given. Then, IP models of the problem and its dual are presented. Subsequently, the procedures for determining the minimum workforce size and assigning workers to days-off patterns are described. A numerical example is solved next. Finally, conclusions are given.

Survey of literature

Labour scheduling problems are traditionally classified into three types: (1) shift, or time-of-day, scheduling, (2) days-off, or days-of-week, scheduling, and (3) tour scheduling, which combines the first two types. Comprehensive surveys of the literature on all these types are provided by Baker (1976), Tien & Kamiyama (1982), and Nanda & Browne (1992). The scope of this review is limited to the days-off scheduling problem, with particular attention to compressed workweek scheduling.

Baker (1976) developed a set covering IP model to specifically represent the days-off scheduling problem. By assuming that each worker must have two days off per week, which were not necessarily consecutive, Monroe (1970) used a simple trial-and-error algorithm to maximize consecutive regular days off (RDOs). Rothstein (1972) utilized linear programming (LP) to formulate and solve the same problem. Later, Chen (1978) used a three-stage manual procedure to obtain the solution.

Several approaches have been developed for the (5, 7) problem, in which only consecutive pairs of days off are allowed. Tiberwala *et al.* (1972) developed a three-step procedure in which the number of iterations equals the number of workers required. Browne & Tiberwala (1975) simplified the three steps involved, but did not reduce the number of iterations. Baker (1974) developed a two-phase algorithm, which started by calculating the lower bound on workforce size, and then used trial and error to determine days-off assignments.

Morris & Showalter (1983) described an iterative, three-step cutting plane procedure to optimally solve the (5, 7) days-off problem. Another iterative, manual procedure utilizing three simple rules was developed by Bechtold & Showalter (1985). The objective of both procedures is to minimize the workforce size. Vohra (1987) developed an expression for the minimum workforce size for the (5, 7) problem. Burns & Koop (1987) optimally scheduled a multiple-shift workforce with two days off per week and n out of m weekends off.

Browne & Nanda (1987) evaluated the scheduling efficiency of four-day workweeks at transportation facilities. Gould (1988) discussed an actual schedule in which two teams alternated work and rest periods of four days each within an eight-day cycle. Hung & Emmons (1993) optimized the 3–4 workweek problem, where in a cycle of two weeks, each worker works three days in one week and four days in the other. Multiple-shift models developed by Hung (1993, 1994) for three-day and four-day workweeks, are similar to Burns & Koop's (1987). These models assume that, (1) D workers are required in each weekday and E workers in each weekend day, and (2) the objective is to minimize the workforce size. In comparison, the model presented below does not assume labour demands to be constant for weekdays or weekend days, nor does it assume that the costs of days-off patterns are equal; thus it can minimize workforce size or cost.

Integer programming models

The (4, 7) labour 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

$$\left(\sum_{j=1}^{7} x_j\right) - x_{i-2} - x_{i-1} - x_i \geqslant r_i, \quad i = 1, 2, \ldots, 7 \tag{2}$$

or

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \geqslant \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \end{bmatrix} \tag{3}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, 7 \tag{4}$$

where W is the workforce size, i.e. the total number of workers assigned to days-off patterns, x_j is the number of workers assigned to weekly days-off pattern j, i.e. the number of workers off on days j, $j+1$ and $j+2$ (as the problem has a weekly cycle, all subscripts are mod 7), and r_i is the number of workers required on day i ($i = 1, 2, \ldots, 7$).

Because $\sum_{j=1}^{7} x_j$ is equal to W, equation (2) can be written as

$$x_{i-2} + x_{i-1} + x_i \leqslant b_i, \qquad i = 1, 2, \ldots, 7 \tag{5}$$

where

$$\begin{aligned} b_i &= W - r_i, \qquad i = 1, 2, \ldots, 7 \\ &= \text{maximum number of workers off on day } i. \end{aligned} \tag{6}$$

For the (4, 7) days-off scheduling model specified by expressions (1), (2) and (4), the dual model with dual variables y_i, $i = 1, 2, \ldots, 7$, is given by

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

subject to

$$\begin{array}{lllllllll} & & & y_4 & + y_5 & + y_6 & + y_7 & \leqslant 1 \\ y_1 & & & & + y_5 & + y_6 & + y_7 & \leqslant 1 \\ y_1 & + y_2 & & & & + y_6 & + y_7 & \leqslant 1 \\ y_1 & + y_2 & + y_3 & & & & + y_7 & \leqslant 1 \\ y_1 & + y_2 & + y_3 & + y_4 & & & & \leqslant 1 \\ & y_2 & + y_3 & + y_4 & + y_5 & & & \leqslant 1 \\ & & y_3 & + y_4 & + y_5 & + y_6 & & \leqslant 1 \end{array} \tag{8}$$

$$y_i \geqslant 0, \qquad i = 1, 2, \ldots, 7. \tag{9}$$

Determining the minimum workforce size

Given seven daily labour demands $r_1, \ldots, r_7$, the minimum workforce size W can be easily obtained without IP using the dual model shown above. An optimum solution to the above dual problem corresponds to a feasible solution to the primal (original) days-off scheduling problem. Moreover, the value of the *optimal* objective function W is the same for the two problems. To solve the dual problem we allocate the unit resource (right hand side of equation (8) equal to unity) among the dual variables in order to maximise the dual objective W, which is a linear combination of labour demands. There are five possible solutions, depending on the given labour demands.

(1) The most obvious solution is to allocate the whole unit resource to the maximum demand, i.e. if $r_k = \max\{r_1, \ldots, r_7\}$, set $y_k = 1$ and all other dual variables to zero. In this case $W = r_k = r_{\max}$. This solution is better than or equal to the average that would result from assigning a value of $1/m$ to any m dual variables, where $m = 2, \ldots, 7$. However, the structure of the problem makes it possible to assign

TABLE 1
Sets of subscripts defined by equations (14)–(16)

i	s_i	s_i^*	t_i	u_i
1	1, 2, 3, 5, 6	4, 7	1, 2, 5	1, 3, 5
2	2, 3, 4, 6, 7	5, 1	2, 3, 6	2, 4, 6
3	1, 3, 4, 5, 7	6, 2	3, 4, 7	3, 5, 7
4	1, 2, 4, 5, 6	7, 3	1, 4, 5	1, 4, 6
5	2, 3, 5, 6, 7	1, 4	2, 5, 6	2, 5, 7
6	1, 3, 4, 6, 7	2, 5	3, 6, 7	1, 3, 6
7	1, 2, 4, 5, 7	3, 6	1, 4, 7	2, 4, 7

a value of $1/m$ to *more* than m dual variables. There are four cases where this is possible, which are discussed next.

(2) Because each constraint in equation (8) contains only four dual variables, it is possible to divide the unit right-hand-side of each constraint among those four variables. Thus it is feasible to assign a value of $1/4$ to all *seven* dual variables. In this case the workforce size $W = \frac{1}{4}\sum_{d=1}^{7} r_d$ or $W = \frac{7}{4}\bar{r}$, where $\bar{r}$ is the average labour demand.

(3) Another possibility exists for seven sets of five dual variables, whose subscripts, denoted by s_i, are shown in Table 1. Because $s_i = \mod 7\{i, i+1, i+2, i+4, i+5\}$ for $i = 1, \ldots, 7$, at least two of any three *adjacent* variables must belong to a set s_i. Therefore, at least two of the three *adjacent* variables missing from each constraint in equation (8) must belong to a set s_i.

For example, if $i = 3$ then $s_3 = \mod 7\{3, 3+1, 3+2, 3+4, 3+5\} = \{1, 3, 4, 5, 7\}$. From this set, y_1 and y_3 are absent from constraint 1, y_3 and y_4 are absent from constraint 2, y_3, y_4 and y_5 are absent from constraint 3, and so on. As no more than three variables from each set are present in any constraint, it is possible to assign a value of $1/3$ to all *five* variables in a given set. In this case, we would choose the set having the maximum total demand; then $W = \frac{1}{3}\max\{\sum_{j\in s_i} r_j\}$ or $W = \frac{5}{3}\overline{r_i}$, where $\overline{r_i}$ is the average demand among the five days belonging to set s_i.

(4) A similar situation pertains to seven sets of three dual variables, whose subscripts, denoted by t_i, are shown in Table 1. Because $t_i = \mod 7\{i, i+1, i+4\}$ for $i = 1, \ldots, 7$, at least one of any three *adjacent* variables must belong to a set t_i. As no more than two variables from each set are present in any constraint, a value of $1/2$ can be assigned to all *three* variables in a given set. Choosing the set with the maximum total demand; $W = \frac{1}{2}\max\{\sum_{j\in t_i} r_j\}$ or $W = \frac{3}{2}\overline{r_i}$, where $\overline{r_i}$ is the average demand among the three days belonging to set t_i.

(5) The final allocation applies to seven sets of three dual variables, whose subscripts, denoted by u_i, are shown in Table 1. Because $u_i = \mod 7\{i, i+2, i+4\}$ for $i = 1, \ldots, 7$, at least one of any three *adjacent* variables must belong to a set u_i. Therefore, it is possible to assign a value of $1/2$ to all *three* variables in a given set. In this case, $W = \frac{1}{2}\max\{\sum_{j\in u_i} r_j\}$ or $W = \frac{3}{2}\overline{r_i}$, where $\overline{r_i}$ is the average demand among the three days belonging to set u_i.

It is also possible to assign a value of 1/4 to six dual variables, or a value of 1/3 to four variables. However, these allocations are obviously dominated by solutions (2) and (3) above, respectively. To determine the workforce size, we choose the maximum value of W obtained from the five above solutions, and must also round up W in case it is not an integer. Therefore, we obtain the following expression for the minimum W:

$$W = \max\left\{ r_{\max}, \left\lceil \frac{1}{4}\sum_{i=1}^{7} r_i \right\rceil, \left\lceil \frac{S_{\max}}{3} \right\rceil, \left\lceil \frac{T_{\max}}{2} \right\rceil, \left\lceil \frac{U_{\max}}{2} \right\rceil \right\} \tag{10}$$

where $r_{\max} = \max\{r_1, r_2, \ldots, r_7\}$, $S_{\max} = \max\{S_1, S_2, \ldots, S_7\}$, $T_{\max} = \max\{T_1, T_2, \ldots, T_7\}$, $U_{\max} = \max\{U_1, U_2, \ldots, U_7\}$, and $\lceil a \rceil$ = smallest integer $\geqslant a$, i.e. a rounded up to the nearest integer, and

$$S_i = \sum_{j \in s_i} r_j, \qquad i = 1, 2, \ldots, 7 \tag{11}$$

$$T_i = \sum_{j \in t_i} r_j, \qquad i = 1, 2, \ldots, 7 \tag{12}$$

$$U_i = \sum_{j \in u_i} r_j, \qquad i = 1, 2, \ldots, 7 \tag{13}$$

$$s_i = \text{modular } 7\{i, i+1, i+2, i+4, i+5\}, \qquad i = 1, 2, \ldots, 7 \tag{14}$$

$$t_i = \text{modular } 7\{i, i+1, i+4\}, \qquad i = 1, 2, \ldots, 7 \tag{15}$$

$$u_i = \text{modular } 7\{i, i+2, i+4\}, \qquad i = 1, 2, \ldots, 7. \tag{16}$$

Sets s_i, t_i, and u_i, $i = 1, \ldots, 7$, are shown in Table 1.

Each argument in equation (10) has an intuitive interpretation that can be logically explained. First, the workforce size cannot be smaller than number of workers required on any given day; thus $W \geqslant r_{\max}$. Second, with each of the W workers assigned four workdays, the total person-days assigned is equal to $4W$. As this must be greater than the total person-days required, $\sum r$; then $W \geqslant \sum r/4$. Third, if we, for example, sum s_1 rows (1, 2, 3, 5 and 6) in system (3), we obtain: $3\left(\sum_{j=1}^{7} x_j\right) - x_1 \geqslant S_1$; thus $3W \geqslant S_1$, or $W \geqslant S_1/3$.

Similar logic can be used explain the remaining arguments and to show that $W \geqslant S_i/3$, $W \geqslant T_i/2$, and $W \geqslant U_i/2$, where $i = 1, 2, \ldots, 7$.

Assigning workers to days-off patterns

Minimum cost assignment

Having determined $W = \sum x$ by equation (10), the next step is to allocate the W workers among the seven days-off patterns. In other words, we need to determine the values of $x_1, \ldots, x_7$. In this section a solution method is developed to obtain the optimum values of these variables. Considering different costs for days-off patterns, the objective is to assign workers to work patterns in order to minimize total cost. Naturally, the cost of each days-off pattern is related to the number of overtime-paid weekend workdays. The seven columns of the matrix in constraint (3) represent days-off patterns, while the seven rows represent days of the week, the last two rows representing the weekend. Assuming each worker is paid 1·0 unit per regular workday and 1·5 units (50% premium) per weekend workday, the costs of the seven days-off patterns are given as

$$c_1, \ldots, c_7 = 5, 5, 5, 4{\cdot}5, 4, 4, 4{\cdot}5 \tag{17}$$

where c_j is the weekly cost of days-off work pattern j per worker.

Introducing these costs into the dual model, the right hand side of constraint (8) changes to the transposed cost vector $(5, 5, 5, 4{\cdot}5, 4, 4, 4{\cdot}5)^T$, or normalized vector $(1, 1, 1, 0{\cdot}9, 0{\cdot}8, 0{\cdot}8, 0{\cdot}9)^T$. Naturally, the basic dual variables and their values, and also the slacks of the dual constraints, do change slightly. However, in all cases, the value of W obtained from equation (10) is not affected. This means that for the cost structure defined by (17), the minimum *cost* is always obtained with the minimum *number* of workers.

The following example will be discussed next to illustrate the previous point. If the maximum among the arguments of equation (10) is $T_1/2$, there are four possible dual solutions, depending on the actual values of $r_1, \ldots, r_7$. Making no assumptions about the value of W, but utilizing primal–dual relationships, four different primal solutions are obtained. In all four cases, shown in Table 2, the basic primal variables add to one value for W:

$$\begin{aligned} W &= \sum_{j=1}^{7} x_j = b_1 + b_2 + b_5 \\ &= (W - r_1) + (W - r_2) + (W - r_5) \\ &= 3W - (r_1 + r_2 + r_5) \\ &= 3W - T_1. \end{aligned}$$

Thus $W = T_1/2$. The minimum cost of the W workers is obtained by assigning as many workers as possible, out of W, to the cheapest days-off patterns: x_5 and x_6, then to x_4 and x_7. The value of W determined by equation (10) is obtained from the optimum solution of the dual problem. Two basic primal–dual relationships will be used for obtaining the solution of the primal (original) days-off scheduling problem. First, a basic dual variable corresponds to a primal equation; second, a dual inequality corresponds to a non-basic (zero) primal variable. The solution will depend on which argument of the right-hand side of equation (10) is maximum; thus there are five possible cases.

Case 1 $r_{\max}$ is maximum

Let $r_i = r_{\max}$. In this case, $W = r_i$, only one dual variable ($y_i = 1$) is basic, and three dual constraints ($i - 2$, $i - 1$ and i) are inequalities. Thus the primal problem has one equation (i) and three variables equal to zero: x_{i-2}, x_{i-1}, and x_i. As $W = r_i$ means $b_i = 0$, this information is embedded in system (5), which is used to assign as many workers as possible to the cheapest patterns: x_5 and x_6, then to x_4 and x_7. There are many optimum alternatives for assigning workers to these patterns. The following rules are simply chosen to ensure

TABLE 2

The four dual and corresponding primal solutions when $T_1/2$ is maximum

No.	Values of basic dual variables	Dual inequalities	Values of basic primal variables
1	$y_1 = y_2 = y_5 = 0{\cdot}4$, $y_6 = 0{\cdot}2$	1, 4, 7	$x_2 = b_2, x_3 = b_5 - x_5$, $x_5 = b_6 - x_6, x_6 = b_1$
2	$y_1 = y_5 = 0{\cdot}5, y_2 = 0{\cdot}3$	1, 3, 4, 7	$x_2 = b_2$, $x_5 = b_5, x_6 = b_1$
3	$y_1 = y_5 = 0{\cdot}45, y_2 = 0{\cdot}35$, $y_7 = 0{\cdot}1$	1, 3, 7	$x_2 = b_2, x_4 = b_5 - x_5$, $x_5 = b_7 - x_6, x_6 = b_1$
4	$y_1 = y_2 = y_5 = 0{\cdot}4$, $y_6 = y_7 = 0{\cdot}1$	1, 7	$x_2 = b_2, x_3 = b_5 - x_4 - x_5$, $x_4 = b_6 - b_7$, $x_5 = b_7 - x_6, x_6 = b_1$

feasibility.

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

Case 2 $\lceil \frac{1}{4} \sum_{i=1}^{7} r_i \rceil$ *is maximum*

In this case, $W = \lceil \Sigma r/4 \rceil$, all dual variables are basic ($y_i = 1/4, i = 1, 2, \ldots, 7$), and all dual constraints are equations. Therefore all primal variables are also basic and all primal constraints are equations. Constraint system (5) is transformed into the following set of equations:

$$x_{i-2} + x_{i-1} + x_i = b_i, \qquad i = 1, 2, \ldots, 7. \tag{18}$$

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

$$x_i = W - \sum_{j \in s_i*} b_j, \qquad i = 1, 2, \ldots, 7 \tag{19}$$

where s_i* is the complement of s_i, shown in Table 1.

Case 3 $\lceil \frac{S_{max}}{3} \rceil$ *is maximum*

Let $S_i = S_{max}$. In this case, $W = \lceil S_i/3 \rceil$, five dual variables are basic ($y_j = 1/3, j \in s_i = i, i+1, i+2, i+4, i+5$), and one dual constraint (i) is an inequality. Thus the primal problem has five equations ($j \in s_i$) and one variable equal to zero: x_i. Ignoring inequalities, system (5) is a 5×6 system of five equations in six unknowns. Because we know that the sum of these unknowns is equal to W, we can add the following equation to obtain a 6×6 system:

$$x_{i+1} + x_{i+2} + x_{i+3} + x_{i+4} + x_{i+5} + x_{i+6} = W. \tag{20}$$

The solution of this system, specifying days-off assignments, is given by

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

The last equation above indicates that the solution is not unique for x_{i+3} and x_{i+4}. The specific solution will depend on the costs of the two patterns. The whole value of the right hand side ($b_{i+5} - x_{i+5}$) is assigned to the cheaper pattern, and a value of zero is assigned to the other one, with ties broken arbitrarily. Specific solutions for each case of i, $i = 1, \ldots, 7$, are shown in Table 3.

Case 4 $\lceil \frac{T_{\max}}{2} \rceil$ *is maximum*

Let $T_i = T_{\max}$. In this case, $W = \lceil T_i/2 \rceil$, three dual variables are basic ($y_j = 1/2$, $j \in t_i = i, i+1, i+4$), and two dual constraints ($i-1, i$) are inequalities. Thus the primal problem has three equations ($j \in t_i$) and two variables equal to zero: x_i, and $x_{i-1} = x_{i+6}$. Including this information in system (5), the first two equations give

$$\begin{aligned} x_{i+5} &= b_i \\ x_{i+1} &= b_{i+1}. \end{aligned}$$

Using these values in the remaining constraints, with three unknowns but only one equation, many alternative feasible solutions exist for this system. The optimum solution can be found for each case of i, ($i = 1, \ldots, 7$) by assigning as much as possible to the cheapest days-off patterns. Specific days-off assignments for each i, $i = 1, \ldots, 7$, are shown in Table 3.

Case 5 $\lceil \frac{U_{\max}}{2} \rceil$ *is maximum*

Let $U_i = U_{\max}$. In this case, $W = \lceil U_i/2 \rceil$, three dual variables are basic ($y_j = 1/2$, $j \in u_i = i, i+2, i+4$), and two dual constraints ($i, i+2$) are inequalities. Thus the primal problem has three equations ($j \in u_i$) and two variables equal to zero: x_i, x_{i+2}. Including this information in system (5), we obtain $x_{i+1} = b_{i+2}$. Using this value in the remaining constraints, with four unknowns but only two equations, many alternative feasible solutions exist for this system. The optimum solution can be found for each case of i, ($i = 1, \ldots, 7$) by assigning as much as possible to the cheapest days-off patterns. Individual days-off solutions for each i, $i = 1, \ldots, 7$, are shown in Table 3.

TABLE 3

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

W	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$r_{\max}$	$W - \sum_{j=2}^{7} x_j\}$	$\min\{b_2 - x_7, b_3 - x_3, b_4 - x_3 - x_4, W - \sum_{j=3}^{7} x_j\}$	$\min\{b_3, b_4 - x_4, b_5 - x_4 - x_5, W - \sum_{j=4}^{7} x_j\}$	$\min\{b_4, b_5 - x_5, b_6 - x_5 - x_6, W - x_5 - x_6\}$	$\min\{b_5, b_6, b_7\}$	$\min\{b_1, b_6 - x_5, b_7 - x_5, W - x_5\}$	$\min\{b_2, b_1 - x_6, b_7 - x_5 - x_6, W - \sum_{j=4}^{6} x_j\}$
$\Sigma r/4$	$W - b_4 - b_7$	$W - b_1 - b_5$	$W - b_2 - b_6$	$W - b_3 - b_7$	$W - b_1 - b_4$	$W - b_2 - b_5$	$W - b_3 - b_6$
$S_1/3$	0	$W - b_1 - b_5$	$b_3 - x_2$	0	$b_6 - x_6$	$b_1 - x_7$	$b_2 - x_2$
$S_2/3$	$b_3 - x_3$	0	$W - b_2 - b_6$	$b_4 - x_3$	$b_7 - x_7$	0	$b_2 - x_1$
$S_3/3$	$b_3 - x_2$	$b_4 - x_4$	0	$W - b_3 - b_7$	$b_5 - x_4$	$b_1 - x_1$	0
$S_4/3$	0	$b_4 - x_3$	$b_5 - x_5$	0	$W - b_1 - b_4$	$b_6 - x_5$	$b_2 - x_2$
$S_5/3$	$b_3 - x_3$	0	$b_5 - x_4$	$b_6 - x_6$	0	$W - b_2 - b_5$	$b_7 - x_6$
$S_6/3$	$b_1 - x_7$	$b_4 - x_4$	0	$b_6 - x_5$	$b_7 - x_7$	0	$W - b_3 - b_6$
$S_7/3$	$W - b_4 - b_7$	$b_2 - x_1$	0	$b_5 - x_5$	$b_7 - x_6$	$b_1 - x_1$	0
$T_1/3$	0	b_2	$b_5 - x_4 - x_5$	$\min\{b_4 - b_2, b_5 - x_5, b_6 - b_1 - x_5\}$	$\min\{b_5, b_6 - b_1, b_7 - b_1\}$	b_1	0
$T_2/3$	0	0	b_3	$b_6 - x_5 - x_6$	$\min\{b_6, b_5 - b_3, b_7 - b_2\}$	$\min\{b_1 - b_2, b_6 - x_5, b_7 - b_2 - x_5\}$	b_2
$T_3/3$	b_3	0	0	b_4	$\min\{b_7, b_5 - b_4, b_6 - b_4\}$	$\min\{b_1 - b_3, b_7 - x_5, b_6 - b_4 - x_5\}$	$b_7 - x_5 - x_6$

TABLE 3
Continued . . .

W	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$T_4/3$	$b_1 - x_6 - x_7$	b_4	0	0	b_5	$\min\{b_1, b_6 - b_5, b_7 - b_5\}$	$\min\{b_1 - x_6, b_2 - b_4, b_7 - b_5 - x_6\}$
$T_5/3$	$\min\{b_2 - x_7, b_3 - b_5, b_1 - b_6 - x_7\}$	$b_2 - x_1 - x_7$	b_5	0	0	b_6	$\min\{b_2, b_1 - b_6, b_7 - b_6\}$
$T_6/3$	$\min\{b_3, b_1 - b_7, b_2 - b_7\}$	$\min\{b_3 - x_1, b_4 - b_6, b_2 - b_7 - x_1\}$	$b_3 - x_1 - x_2$	b_6	0	0	b_7
$T_7/3$	b_1	$b_4 - x_3 - x_4$	$\min\{b_3 - b_1, b_4 - x_4, b_5 - b_7 - x_4\}$	$\min\{b_4, b_5 - b_7, b_6 - b_7\}$	b_7	0	0
$U_1/2$	0	b_3	0	$b_5 - x_5$	$\min\{b_5, b_7 - b_1\}$	$\min\{b_1, b_6 - b_5\}$	$b_1 - x_6$
$U_2/2$	$b_2 - x_7$	0	b_4	0	$\min\{b_6, b_5 - b_4\}$	$b_6 - x_5$	$\min\{b_2, b_7 - b_6\}$
$U_3/2$	$\min\{b_3, b_1 - b_7\}$	$b_3 - x_1$	0	b_5	0	$\min\{b_7, b_6 - b_5\}$	$b_7 - x_6$
$U_4/2$	$b_1 - x_7$	$\min\{b_4, b_2 - b_1\}$	$b_4 - x_2$	0	b_6	0	$\min\{b_1, b_7 - b_6\}$
$U_5/2$	$\min\{b_2, b_1 - b_7\}$	$b_2 - x_1$	$b_5 - x_4$	$\min\{b_5, b_6 - b_7\}$	0	b_7	0
$U_6/2$	0	$\min\{b_3, b_2 - b_1\}$	$b_3 - x_2$	$b_6 - x_5$	$\min\{b_6, b_7 - b_1\}$	0	b_1
$U_7/2$	b_2	0	$b_4 - x_4$	$\min\{b_4, b_6 - b_7\}$	$\min\{b_7, b_5 - b_4\}$	$b_7 - x_5$	0

Steps of the algorithm

1. Determine the minimum workforce size W using equation (10).
2. (a) If $\max\{r_{\max}, \Sigma r/4, S_{\max}/3, T_{\max}/2, U_{\max}/2\} = r_{\max}$, then:
   - determine $b_1, \ldots, b_7$ using (6), then apply $r_{\max}$ row in Table 3 to find $x_1, \ldots, x_7$.

   (b) If $\max\{r_{\max}, \Sigma r/4, S_{\max}/3, T_{\max}/2, U_{\max}/2\} = \Sigma r/4$, then:
   - if $\Sigma r/4$ is not integer, increment Σr by $(4W - \Sigma r)$ to make it a multiple of four; among all seven daily labour demands r_i, avoid whenever possible increasing:
     (i) weekend demands, r_6 or r_7, and (ii) maximum labour demand, $r_{\max}$.
   - determine $b_1, \ldots, b_7$ by (6), then apply $\Sigma r/4$ row in Table 3 to find $x_1, \ldots, x_7$.

   (c) If $\max\{r_{\max}, \Sigma r/4, S_{\max}/3, T_{\max}/2, U_{\max}/2\} = S_{\max}/3$, then:
   - if $S_{\max}/3$ is not integer, increment $S_i = S_{\max}$ by $(3W - S_i)$ to make it a multiple of three. Among the five daily labour demands that can be increased $r_j, j \in s_i$, choose the ones to be increased according to the criteria given in step 2(b).
   - determine $b_1, \ldots, b_7$ by (6), then apply $S_i/3$ row in Table 3 to find $x_1, \ldots, x_7$.

   (d) If $\max\{r_{\max}, \Sigma r/4, S_{\max}/3, T_{\max}/2, U_{\max}/2\} = T_{\max}/2$, then:
   - if $T_{\max}/2$ is not integer, increment $T_i = T_{\max}$ by unity to make it a multiple of two. Among the three daily labour demands that can be increased $r_j, j \in t_i$, choose the one to be increased according to the criteria given in step 2(b).
   - determine $b_1, \ldots, b_7$ by (6), then apply $T_i/2$ row in Table 3 to find $x_1, \ldots, x_7$.

   (e) If $\max\{r_{\max}, \Sigma r/4, S_{\max}/3, T_{\max}/2, U_{\max}/2\} = U_{\max}/2$, then:
   - if $U_{\max}/2$ is not integer, increment $U_i = U_{\max}$ by unity to make it a multiple of two. Among the three daily labour demands that can be increased $r_j, j \in u_i$, choose the one to be increased according to the criteria given in step 2(b).
   - determine $b_1, \ldots, b_7$ by (6), then apply $U_i/2$ row in Table 3 to find $x_1, \ldots, x_7$.
3. In the case of ties, apply any system arbitrarily.

A numerical example

The following example is used to illustrate the simple calculations required for implementing the algorithm, where the tableau serves to clarify the computations of S_i, T_i, and U_i. Given the following daily labour demands for a work week:

$$r_1, r_2, \ldots, r_7 = 9, 7, 2, 6, 8, 7, 3,$$

calculations are carried out on the following tableau.

i	1	2	3	4	5	6	7	Total
r_i	9	7	2	6	8	7	3	42
S_1	9	7	2		8	7		33
S_2		7	2	6		7	3	25
S_3	9		2	6	8		3	28
S_4	9	7		6	8	7		37
S_5		7	2		8	7	3	27
S_6	9		2	6		7	3	27
S_7	9	7		6	8		3	33
T_1	9	7			8			24
T_2		7	2			7		16
T_3			2	6			3	11
T_4	9			6	8			23
T_5		7			8	7		22
T_6			2			7	3	12
T_7	9			6			3	18
U_1	9		2		8			19
U_2		7		6		7		20
U_3			2		8		3	13
U_4	9			6		7		22
U_5		7			8		3	18
U_6	9		2			7		18
U_7		7		6			3	16

From the tableau

$$
\begin{aligned}
r_{\max} &= 9 \\
\Sigma r/4 &= 42/4 = 10{\cdot}5 \\
S_{\max}/3 &= S_4/3 = 37/3 = 12{\cdot}33 \\
T_{\max}/2 &= T_1/2 = 24/2 = 12 \\
U_{\max}/2 &= U_4/2 = 22/2 = 11.
\end{aligned}
$$

Using equation (10), the workforce size is

$$W = \lceil S_4/3 \rceil = \lceil 12{\cdot}33 \rceil = 13.$$

Because $S_4/3 = 12{\cdot}33$ is not an integer, we must add two to S_4 in order to make it divisible by three. The set S_4 contains demands for days 1, 2, 4, 5 and 6. Using the criteria specified in step 2(a) of the algorithm, we increment the demand for day 4 by two (thus $r_4 = 8$), and then use equation (6): $b_i = 13 - r_i$, to obtain

$$b_1, b_2, \ldots, b_7 = 4, 6, 11, 5, 5, 6, 11.$$

Using the S_4 row in Table 3, we obtain the following days-off assignments:

$$\begin{aligned}
x_1 &= 0\\
x_4 &= 0\\
x_5 &= W - b_1 - b_4 = 13 - 4 - 5 = 4\\
x_6 &= b_6 - x_5 = 6 - 4 = 2\\
x_3 &= b_5 - x_5 = 5 - 4 = 1\\
x_2 &= b_4 - x_3 = 5 - 1 = 4\\
x_7 &= b_2 - x_2 = 6 - 4 = 2.
\end{aligned}$$

Conclusions

A new, efficient optimization algorithm for the cyclic (4, 7) labour days-off scheduling problem has been presented. The algorithm is based on the solution of the dual linear programming model, but does not involve linear or integer programming. Because the costs of different days-off work patterns are not assumed to be equal, the algorithm can be used to minimize either the total number or the total cost of workers assigned.

Eliminating the need for linear or integer programming provides both efficiency and ease of use. Moreover, the simplicity of the algorithm makes it easy to program or even implement manually, removing the need for specialized training and software. A similar approach could be used for solving other scheduling problems in which the dual solution is easily obtainable.

References

AARDAL, K. & ARI, A. 1987 Decomposition principles applied to the dynamic production and workforce scheduling problem. *Eng. Costs Prod. Econ.*, **12**, 39–49.

ALFARES, H. K. 1998 An efficient two-phase algorithm for manpower days-off scheduling. *Comput. Oper. Res.*, **25**, 913–923.

ALFARES, H. K. & BAILEY, J. E. 1997 Integrated project task and manpower scheduling. *IIE Trans.*, **29**, 711–718.

BAKER, K. R. 1974 Scheduling a full-time workforce to meet cyclic staffing requirements. *Manage. Sci.*, **20**, 1561–1568.

BAKER, K. R. 1976 Workforce allocation in cyclical scheduling problems: a survey. *Oper. Res. Q.*, **27**, 155–167.

BECHTOLD, S. E. & SHOWALTER, M. J. 1985 Simple manpower scheduling methods for managers. *Prod. Inventory Manage.*, **26**, 116–133.

BROWNE, J. & NANDA, R. 1987 Scheduling efficiency of the four-day week at transportation facilities. *Proceedings of The Institute of Transportation Engineers 57th Annual Meeting*, August 16–20 New York: pp. 58–62.

BROWNE, J. E. & TIBERWALA, R. K. 1975 Manpower scheduling. *AIIE Trans.*, **7**, 22–23.

BURNS, R. N. & KOOP, G. J. 1987 A modular approach to optimal multiple-shift manpower scheduling. *Oper. Res.*, **35**, 100–110.

CHEN, D-S 1978 A simple algorithm for a workforce scheduling model. *AIIE Trans.*, **10**, 244–251.

GOULD, C. H. 1988 Rolling fours: novel work schedule. *J. Construct. Eng. Manage.*, **114**, 577–593.

HUNG, R. 1993 A three-day workweek multiple-shift scheduling model. *J. Oper. Res. Soc.*, **44**, 141–146.

HUNG, R. 1994 A multiple-shift workforce scheduling model under the 4-day workweek with weekday and weekend labour demands. *J. Oper. Res. Soc.*, **45**, 1088–1092.

HUNG, R. & EMMONS, H. 1993 Multiple-shift workforce scheduling under the 3–4 compressed workweek with a hierarchical workforce. *IIE Trans.*, **25**, 82–89.

KOGI, K. & THURMAN, J. E. 1993 Trends in approaches to night and shiftwork in new international standards. *Ergonomics*, **36**, 3–13.

LANKFORD, W. 1998 Changing schedules: a case for alternative work schedules. *Career Develop. Int.*, **3**, 161–163.

MONROE, G. 1970 Scheduling manpower for service operations. *Ind. Eng.*, **2**, 10–17.

MORRIS, G. M. & SHOWALTER, M. J. 1983 Simple approaches to shift, days-off and tour scheduling problems. *Manage. Sci.*, **29**, 942–950.

NANDA, R. & BROWNE, J. 1992 *Introduction to Employee Scheduling*. New York: Van Nostrand Reinhold.

ROTHSTEIN, M. 1972 Scheduling manpower by mathematical programming. *Ind. Eng.*, **4**, 29–33.

TIBERWALA, R., PHILIPPE, D., & BROWNE, J. 1972 Optimal scheduling of two consecutive idle periods. *Manage. Sci.*, **19**, 71–75.

TIEN, J. M. & KAMIYAMA, A. 1982 On manpower scheduling algorithms. *SIAM Rev.*, **24**, 275–287.

VOHRA, R. V. 1987 The cost of consecutivity in the (5, 7) cyclic staffing problem. *IIE Trans.*, **29**, 942–950.