JOURNAL OF APPLIED MATHEMATICS AND DECISION SCIENCES, 6(3), 191–199

Optimum Workforce Scheduling Under the (14, 21) Days-Off Timetable

HESHAM K. ALFARES hesham@ccse.kfupm.edu.sa
System Engineering Department, King Fahd University of Petroleum & Minerals, P.O. Box 5067, Dhahran 31261, Saudi Arabia.

Abstract. An efficient optimum solution is presented for a real-life employee days-off scheduling problem with a three-week cycle. Over a given work cycle, each worker is given 14 successive workdays and 7 successive off days. This three-week days-off timetable is referred to as the (14, 21) schedule. Given different labor demands for each day of the week, the primary objective is to minimize the number of workers. The secondary objective is to reduce transportation cost by minimizing the number of active days-off patterns. The solution technique utilizes the dual LP solution to determine the minimum number of workers and feasible days-off assignments, without using linear or integer programming. The simple solution technique eliminates the need to use integer programming for this particular scheduling problem.

Keywords: Employee scheduling, staffing, optimization.

1. Introduction

Employee scheduling is a complicated multiple-objective problem, involving such diverse considerations as varying manning requirements, costs, availabilities, skills, and personal preferences. The efficient scheduling of the workforce can greatly reduce the labor cost, which is generally a significant proportion of total cost for most organizations. Days-off scheduling is a problem that arises in establishments that have seven-day workweeks, such as restaurants, power stations, and hospitals. The problem is to satisfy the continuous work requirements with employees who cannot work continuously. Therefore, the number of employees assigned to each days-off pattern must be determined, in order to satisfy all daily labor demands with the minimum number or cost of employees.

This paper is concerned with the (14, 21) days-off scheduling problem, which has a three-week or 21-day cycle. During a given cycle, each employee is assigned one work stretch of 14 consecutive workdays and a break of 7 consecutive days off. This is an actual work schedule used by a major oil company to schedule employees in remote areas. The main advantage of this schedule is the reduced cost of transportation to remote work locations. Under the (14, 21) schedule, each employee gets only one 7-day break instead of the usual three weekends off during the three-week cycle. The (14, 21) timetable is designated as a restricted work schedule; it is only applicable to employees in remote work locations.

The primary objective of the (14, 21) days-off scheduling problem is to minimize the workforce size, i.e., total number of employees assigned. In order to reduce transportation cost, we must add a secondary objective, which is that of minimizing the number of *active* days-off patterns (i.e., patterns to which some employees are actually assigned). All employees assigned to the same off days pattern will be transported together, usually by aircraft flights. Since the company owns its private aircraft fleet, transportation cost depends on the number of flights, not on the number of employees on each flight. Therefore, total transportation cost is proportional to the number of active days-off patterns.

Subsequent sections of this paper are organized as follows. First, recently published days-off scheduling approaches are surveyed, then integer programming models of the (14, 21) problem and its dual are presented. Subsequently, procedures for determining the minimum workforce size and assigning workers to days-off patterns are developed. Finally, conclusions are given.

2. Literature Survey

Workforce scheduling problems are traditionally classified into three types: shift scheduling, days-off scheduling, and tour scheduling. Shift (or time-of-day) scheduling determines each employee's work and break hours per day. Days-off (or days-of-week) scheduling determines each employee's workdays and off days per week or multiple-week work cycle. Tour scheduling combines the shift and days-off scheduling problems by determining each employee's daily work hours and weekly workdays. Nanda and Browne (1992) provide a thorough survey of literature on all three problem types. The focus in this paper is on days-off scheduling methodologies published since 1990.

A great deal of interest has been directed at the (5, 7) days-off problem, in which two consecutive days off are given per week. Alfares and Bailey (1997) establish a lower bound on the workforce size for the (5, 7) problem, which is used within an integrated project manpower and activity scheduling algorithm. Alfares (1998) develops an expression for the minimum workforce size, and includes it as a constraint in the linear programming (LP) relaxation of the integer programming model.

Hung and Emmons (1993), who analyzed compressed workweek scheduling, develop an optimal algorithm for the 3-4 workweek, assuming D workers are required everyday. Their multiple-shift, hierarchical-workforce model allows shift rotation and worker substitution. Alfares (1997) develops a single-shift optimum solution technique for 3-day workweeks. Burns *et al.* (1998) present a set-processing methodology for 3-day and 4-day workweeks with work stretch constraints. Billionnet (1999) uses integer programming to schedule a hierarchical workforce to satisfy varying labor demands over the week, giving each worker n off-days per week.

Hung (1991) presents a single-shift model for compressed workweeks, with the objective of minimizing the workforce size. This days-off scheduling model is based on two assumptions: (1) D workers are required on weekdays and E workers on weekends, where $D \geq E$, and (2) each worker must have A out of B weekends off. Under the same assumptions, Narasimhan (1996) considers multiple worker types, giving each worker two days off per week. Emmons and Burns (1991) also consider a workforce composed of n worker types, but assume a constant labor demand for all days of the week.

Nanda and Browne (1992, pp. 94-97) optimally solve the general (r, n) scheduling problem by the first period principle algorithm, giving each employee r consecutive work periods in a cycle of n periods. Although the first period principle algorithm can be adapted to the (14, 21) days-off scheduling problem, it has two significant disadvantages. First, it cannot guarantee that the minimum number of days-off patterns are used. Second, it involves six-steps with lots of calculations that include assigning employees day by day for several cycles. The solution presented in this paper is vastly simpler, and it produces the minimum number of active days-off patterns.

2.1. Mathematical Programming Models

The integer linear programming model shown below represents the (14, 21) days-off scheduling problem. Objective function (1) includes two prioritized goals. The primary goal is to minimize the total number of workers. The secondary goal is to minimize the number of active days-off patterns (i.e., the number of patterns to which some employees are actually assigned). The small value of its coefficient ε indicates the smaller weight of the secondary objective. Labor demand constraints (2) ensure that daily labor demands are satisfied for every day during the three-week cycle. As typical of most real-world situations, daily labor demands (r_1, ..., r_7) are allowed to vary from day to day of the week, but are constant from week to week. Logical constraints (3) are necessary to ensure that v_j is equal to 1 if x_j is positive, and v_j is equal to 0 if x_j is equal to 0.

$$MinimizeZ = \sum_{j=1}^{21} x_j + \varepsilon \sum_{j=1}^{21} v_j \tag{1}$$

subject to:

$$\sum_{j=1}^{21} a_{ij} x_j \geq r_i, \qquad i = 1, \ldots, 21 \tag{2}$$

$$Mv_j \geq x_j, \qquad j = 1, \ldots, 21 \tag{3}$$

$$x_j \geq 0 \quad \text{and integer}, \quad j = 1, \ldots, 21 \tag{4}$$

$$v_j = 0 \text{ or1}, \qquad j = 1, \ldots, 21 \tag{5}$$

where

a_{ij} = 1 if i is a workday for days-off pattern $j, 0$ otherwise. Matrix $A = \{a_{ij}\}$ is shown in Table 1

x_j = number of workers assigned to weekly days-off pattern j, i.e., number of workers whose first day off is$j$

ε = small constant ($\varepsilon \ll 1$)

v_j = logical variable: $v_j = 1$ if $x_j > 0$, and $v_j = 0$ if $x_j = 0$

M = large constant ($M \gg 1$)

r_i = minimum number of workers required on day i, where $r_k = r_{k+7} = r_{k+14}, k = 1, \ldots, 7$

The model formulated above is a non-trivial pure integer programming (IP) problem, with 42 constraints and 42 integer variables. Half of the variables (x_1, ..., x_{21}) are general integer, while the other half (v_1, ..., v_{21}) are binary. The size and pure-integer nature of this model make optimum solution by integer programming inefficient. Computational experiments have been carried out on a small number of initial test problems with different characteristics. Optimum solutions have been attempted using both Hyper Lindo® and Excel Solver® on a 450-MH Pentium II with 128MB of memory. In many cases, several hours were insufficient to obtain optimum integer programming solutions. Thus, an efficient optimum method will be presented next to solve this scheduling problem.

In order to develop this method, the first step taken is to simplify the above IP model. First, the secondary objective of minimizing the number of active days-off patterns will be temporarily ignored. Therefore, all the variables (v_1, ..., v_{21}) and constraints (3) and (5) relevant to this objective

Table 1. Days-off matrix $A = \{a_{ij}\}$ for the (14, 21) schedule

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
2	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
3	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
5	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
6	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

will be removed from the model. The secondary objective will be considered again at a later stage in the development of the solution. For now, however, the simplified model includes only constraints (2) and (4), which are imposed on the new objective function:

$$\text{Minimize } W = \sum_{j=1}^{21} x_j \tag{6}$$

where W is the workforce size, i.e., number of workers assigned to all 21 days-off patterns.

The dual of the LP relaxation of the simplified days-off scheduling model, defined by (2) and (6), with dual variables y_1, ..., y_{21}, is:

$$\text{Maximize } W = \sum_{i=1}^{21} r_i y_i \tag{7}$$

subject to

$$\sum_{i=1}^{21} a_{ij} y_i \leq 1, \quad j = 1, \dots, 21 \tag{8}$$

$$y_i \geq 0, \qquad i = 1, \dots, 21 \tag{9}$$

3. The Minimum Workforce Size

Given a one-week varying daily labor demands r_k, $k = 1, \dots, 7$, the minimum workforce size W can be determined using the dual model shown above, without integer programming. To solve the dual problem we allocate the unit resource - right hand side of constraints (8) - among the 21 dual variables in order to maximize the dual objective W. Based on a complete enumeration of all dual solutions, the optimum solution is obtained by allocating the unit right hand side of (8) to only three dual variables: y_m, y_{m+7}, and y_{m+14}, where $m = 1, \dots, 7$. Matrix A shown in Table 1 indicates that only two of these three variables are active or present in each constraint (8).

Consequently, we can assign a value of 1/2 to each of the three variables y_m, y_{m+7}, and y_{m+14}. Since $r_m = r_{m+7} = r_{m+14}$, substituting into (7) produces the workforce size as $W = 3r_m/2$. Given that the objective in (7) is maximization, we must choose m such that $r_m = r_{max} = max\{r_1, \dots, r_7\}$. Hence, the workforce size W is determined by multiplying the maximum daily labor demand r_{max} by 3/2. We must round up W in case it is not an integer; thus we obtain the following expression for the minimum workforce size:

$$W = \left\lceil \frac{3}{2}\, r_{max} \right\rceil \tag{10}$$

where $\lceil a \rceil$ is the smallest integer greater than or equal to a

4. Days-Off Assignments

The optimum dual solution developed above will be used to determine days-off assignments $x_1, \dots, x_{21}$. The principles of complementary slackness and basic primal-dual relationships will be used to obtain the solution of the primal (original) problem. At optimality, the primal and dual objective functions are equal. Thus for both primal and dual problems, the optimum value of the workforce size is the same, which is $W = \lceil 3r_{max}/2 \rceil$. Utilizing this information in the original model defined by (1) to (5), with the objective of minimizing the number of active days-off patterns, the optimum solution is given by:

$$\begin{aligned} x_k = x_{k+7} &= \lceil \frac{r_{\max}}{2} \rceil, \\ x_{k+14} &= \lfloor \frac{r_{\max}}{2} \rfloor, \quad k = 1, \ldots, 7 \end{aligned} \tag{11}$$

where $\lfloor a \rfloor$ is the largest integer less than or equal to a

Proof is provided in the Appendix that Equation system (11) represents a feasible and optimum solution to the (14, 21) days-off scheduling problem defined by equations (1) to (5). The value of k can be any integer from 1 to 7, which provides flexibility in choosing the active days-off assignments. For example, $r_{max} = 7$ in a certain work location. If we choose $k = 1$, then the solution is given as $x_1 = x_8 = 4$, and $x_{15} = 3$. The solution defined by (10) and (11) can be implemented by the following simple rules:

1. Assign a workforce of a size equal to three-halves the maximum daily manpower requirement (rounded up to the next integer).

2. Make one trip on the same day of the week to deliver workers to and from the work site.

3. Transport half the maximum daily manpower requirement (rounded up) in two of the trips and the remainder in the third trip of the three-week cycle.

5. Conclusions

An efficient optimum solution technique has been presented for scheduling employees using a real-life days-off schedule with a three-week cycle. The days-off timetable is referred to as the (14, 21) schedule. This schedule is mainly used for scheduling employees in remote work location, where transportation cost is high. Each employee is given one break consisting of seven consecutive days instead of the usual three weekends off during the three-week cycle. As typical of most real-life staffing situations, daily labor demands are assumed to vary between weekdays, but remain constant from week to week.

The primary objective of the solution technique is to minimize the workforce size. The secondary objective is to reduce transportation cost by minimizing the number of active days-off patterns, i.e., patterns to which some employees are assigned. The optimum solution by integer programming has been found to be inefficient. Therefore, an optimum, simple solution method has been developed. This method does not involve linear or integer programming, but utilizes the dual solution and primary-dual relationships. The solution method is simple enough to use manual calculations.

Acknowledgments

The author would like to express gratitude to King Fahd University of Petroleum & Minerals for supporting this research project. He also wishes to thank the O&IE Department at Saudi Aramco, at which this research was conducted, for the gracious hospitality. Thanks are also due to two anonymous referees for constructive comments on an earlier version of this manuscript.

References

1. H.K. Alfares. Optimum compressed workweek scheduling, *Proceedings of the 22nd International Conference on Computers & Industrial Engineering*, Cairo, Dec. 19-22, 13-16, 1997.
2. H.K. Alfares. An efficient two-phase algorithm for cyclic days-off scheduling, *Computers & Operations Research*, 25, 913-923, 1998.
3. H.K. Alfares, and J. E. Bailey. Integrated project task and manpower scheduling, *IIE Transactions*, 29, 711-718, 1997.
4. A. Billionnet. Integer programming to schedule a hierarchical workforce with variable demands, *European Journal of Operational Research*, 114, 105-114, 1999.
5. R.N. Burns, R. Narasimhan, and L.D. Smith. A set-processing algorithm for scheduling staff on 4-day or 3-day work weeks, *Naval research Logistics*, 45, 839-853, 1998.
6. H. Emmons, and R.N. Burns. Off-day scheduling with hierarchical worker categories, *Operations Research*, 39, 484-495,1991.
7. R. Hung. Single-shift workforce scheduling model under a compressed workweek, *OMEGA*, 19, 494-497, 1991.
8. R. Hung, H. Emmons. Multiple-shift workforce scheduling under the 3-4 compressed workweek with a hierarchical workforce, *IIE Transactions*, 25, 82-89, 1993.
9. R. Nanda, and J. Browne. *Introduction to Employee Scheduling*, Van Nostrand Reinhold, New York, 1992.
10. N. Narasimhan. An algorithm for single shift scheduling of hierarchical workforce, *European Journal of Operational Research*, 96, 113-121, 1996.

Appendix

Proof of feasibility and optimality of the solution defined by Equation (11)

The solution specified by (11) is feasible since, as seen from Table 1, two of the three variables (x_k, x_{k+7}, x_{k+14}) are always present in each constraint (2), or

$$a_{i,k} + a_{i,k+7} + a_{i,k+14} = 2, \quad i = 1, ..., 21, \quad k = 1, \ldots, 7 \tag{A.1}$$

Therefore, the left-hand side of each constraint (2) is at least equal to r_{max}, since

$$\begin{aligned} \sum_{j=1}^{21} a_{ij}x_j &= a_{i,k}x_k + a_{i,k+7}x_{k+7} + a_{i,k+14}x_{k+14} \\ &= a_{i,k}\left\lceil \frac{r_{\max}}{2} \right\rceil + a_{i,k+7}\left\lceil \frac{r_{\max}}{2} \right\rceil + a_{i,k+14}\left\lfloor \frac{r_{\max}}{2} \right\rfloor \\ &\geq \left\lceil \frac{r_{\max}}{2} \right\rceil + \left\lfloor \frac{r_{\max}}{2} \right\rfloor = r_{max} \end{aligned} \tag{A.2}$$

The solution defined by (11) is optimal in terms of both prioritized goals in (1). First, (11) gives the minimum workforce size W defined by (10), since

$$W = x_k + x_{k+7} + x_{k+7} = \left\lceil \frac{r_{\max}}{2} \right\rceil + \left\lceil \frac{r_{\max}}{2} \right\rceil + \left\lfloor \frac{r_{\max}}{2} \right\rfloor = \left\lceil \frac{3}{2} r_{max} \right\rceil \tag{A.3}$$

Second, (11) provides the minimum number of active days-off patterns, which is *three* (x_k, x_{k+7}, x_{k+14}). If we assign only two active days-off patterns x_m and x_n, the left-hand side of constraints (2) will be equal to $a_{i,m}x_m + a_{i,n}x_n$. As Table 1 shows, for any two columns m and n in matrix A, there will be at least one row i for which either$a_{i,m}$ or $a_{i,n}$ is equal to zero. Thus, the left-hand side value will be only x_m for some constraints (2) and only x_n for some other constraints (2). Since $x_m + x_n = W$, the maximum value of $min(x_m, x_n) = \lfloor W/2 \rfloor \cong \frac{3}{4} r_{max}$, which is less than r_{max}, making the solution infeasible.

Hence, with only two active days-off patterns, there will be some days in which insufficient workers are assigned to meet labor demands. Needless to say, this will be also the case if there is only one active days-off pattern. Thus, the minimum number of active days-off patterns required to produce a feasible solution is three.