Volume 3, Number 2, 2006, pp. 35-48

Heuristic Algorithm for Workforce Scheduling Problems

Carlos Montoya
Research Assistant,
Department of Industrial Engineering,
Universidad de los Andes.

Gonzalo Mejía, PhD.
Assistant Professor,
Department of Industrial Engineering,
Universidad de los Andes.

Abstract

In this paper we present a heuristic approach for solving workforce scheduling problems. The primary goal is to minimize the number of required workers given a pre-established shift demand over a planning horizon. The proposed algorithm starts with an initial solution (initial number of workers and their shift assignment) and iteratively searches the state space, moving towards better solutions via a local search procedure. Local optima are avoided by guaranteeing that the algorithm never returns to a previously visited solution. The algorithm stops after a termination criterion is met. The solution provides a detailed schedule of each worker on each shift. A number of constraints such as minimum and maximum number of working hours, rest days, and maximum number of continuous working hours are considered. The algorithm was tested on a number of randomly generated problems of different sizes. A Mixed Integer Programming (MIP) formulation is proposed and used as a benchmark. Computational experiments show that the algorithm always found optimal or near-optimal solutions with significantly less computer effort.

Keywords: Workforce scheduling, shift scheduling, heuristics, shift assignment.

INTRODUCTION

The Workforce Scheduling Problem (WSP) is related with a number of topics which include shift design, days-off scheduling and workload assignment. Workforce scheduling is important to maximize profits without sacrificing the staff's efficiency. Related research (Best, 1991) shows that workforce schedules affect both health and social life of workers. Hence it is important to establish workforce schedules that both reduce costs for the organization and that guarantee adequate work conditions.

Volume 3, Number 2, 2006, pp. 35-48

This in order to decrease the possibilities of work related accidents or have a negative effect on the daily activities performance of the workers.

A number of applications of workforce scheduling problems are commonly found in every day life. These include scheduling of airline crews, hospital staff, call-center receptionists, public transportation, and industrial plants. This paper deals with the shift assignment problem. This scheduling problem consists of determining the optimal personnel assignment to a set of pre-determined shifts subject to legal and organizational constraints. Instances of this WSP are usually NP-Hard. In small cases, exact solutions are possible to find with mathematical programming. However, in real cases heuristic approaches are required to obtain near-optimal solutions in reasonable computing times.

In this paper we developed a local search heuristic algorithm to solve the problem of finding the minimum number of workers required to meet shift demands over a planning horizon. A number of constraints are considered which include the minimum number of working days in a week, the assignment of workers to non-consecutive shifts, etc. The resulting solution also shows the assignment of workers into the predefined shifts. We developed a procedure to test the proposed algorithm, which is the use of a Mixed Integer Programming (MIP) formulation.

WORKFORCE SCHEDULING DEFINITIONS

Shift. A shift is defined as the period of time in which a group of workers are on duty. The shift definition includes the starting and finishing times (Musliu, 2001).

Workforce. Workforce is defined as the required number of workers to satisfy a shift demand (Musliu, 2001).

Planning Horizon. The planning horizon (P) is considered the time period in which the workforce is assigned to different shifts (Musliu, 2001).

Cyclic schedules and non-Cyclic schedules. A cyclic schedule is such that the workforce is initially scheduled for a pre-defined planning horizon: In the second cycle, the first worker takes on the schedule of the second; the second one takes on the schedule of the third one, and so on. The length of the cycle will be equal to the number of workers. In the example shown in table 1 the length of the cycle is five. The cyclic schedule guarantees that all the employees work the same number of hours, but in this type of schedule it is more difficult to take into account worker needs and special requirements in the moment of establishing a workforce assignment. In non-cyclic schedules the workforce assignment is done every P periods of time according to worker and company needs.

Shift assignment. This is the assignment of workers to shifts previously established. Table 1 shows an assignment of 5 workers to 3 shifts during one week.

Table 1. Example of workers assignment to shifts (Taken from Musliu (2001))

Employee/day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	M	M	M	M	A		
2	A	A	A	A			
3	N	N	N	N	N		
4	A	A	A			N	N
5		M	M	M	M		

Shifts	
M	Morning
A	Afternoon
N	Night

Rotating Workforce Scheduling. It is the assignment of workers to shifts in a cyclic schedule (Musliu, 2001).

Feasible shifts. A shift is feasible when all workers can be assigned meeting all constraints. Typical constraints are the maximum consecutive workdays or the minimum/maximum daily hours a worker can be assigned, etc.

PREVIOUS WORK

The aforementioned problem can be classified as a Rotating Workforce Scheduling Problem (RWSP). This problem consists of finding the optimal assignment of shifts and work days for each worker. Researchers have taken different approaches to solve this problem. Balakrishman and Wong (1990) solved this problem by using network flow formulations. Lau considered (1996a) the shift assignment problem subject to personnel demand and shifts change constraints and proved that this problem is NP – complete. Later Gartner and Wahl (1998) proposed heuristics to solve for the manual construction of rotating workforce schedules. Brusco and Jacobs (1993) developed a Simulated Annealing approach to solve a cyclic staff scheduling problem. Hung (1993) proposed a simple algorithm to solve the problem of assigning workers into shifts, but considering that a worker comes to work 3 days per week. Later the same author (Hung, 1994) extended his previous work on this subject by considering that

Volume 3, Number 2, 2006, pp. 35-48

worker comes to work 4 days per week. The analysis seeks to minimize the workforce size subject to satisfying staffing requirements on weekdays and weekends, and work rules relating to shift changes, off-days, and off-weekends. Jackson, Dollard and Havens (1997) developed a procedure to solve the workforce scheduling problem considering employees with different skills and shifts with specific skills requirement. Aykin (1998) developed a branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows. Laporte (1999) considered developing rotating workforce schedules by hand showing how the constraints can be relaxed to obtain acceptable schedules. Musliu (2001) solved this problem using backtracking algorithms obtaining better results than those of Balakrishman and Wong (1990). Recently, Mejía and Montoya (2007) proposed a Petri Net based algorithm to minimize workforce requirements in systems with precedence and shift length constraints.

PROBLEM DESCRIPTION

In this paper we intend to find the minimum cost workforce assignment for a cyclic schedule. The input data for this model are the predefined shifts, the number of shifts, and the hours in which each shift takes place.

Variables

W: Total number of workers.

$w^+_{i,k}$: Excess of workers on the shift i of the day k.

$w^-_{i.k}$: Shortage of workers on the shift i of the day k.

Input data

c_w: Unit cost per worker(salary).

c_e: Cost of excess per worker and per shift.

c_s: Cost of shortages per worker and per shift.

W_0: Initial number of workers.

$W_{i,k}$: Personnel demand in the shift i of the day k.

s_k: Number of shifts in day k.

D_c: Maximum number of consecutive workdays a worker can be assigned.

M: Number of workdays that a worker must be assigned in one week.

N: Number of workdays in a week ($N \geq M$).

Assumptions

(i) The week has 7 working days (N=7).

(ii) Either shortages or excess of workers are allowed in a shift.

(iii) There exists a minimum and a maximum hours per day in which a worker can be assigned.

(iv) The total worker demand in each shift is known.
(v) Shift overlaps are not allowed.
(vi) The size of the workforce must satisfy at least the minimum of the daily personnel demands.
(vii) All workers must work exactly M days of the week, even if this results in excess of workers in some shifts.
(viii) The planning horizon (P) is one week

Objective function

$$(c_w \times W) + (\sum_{k=1}^{N} \sum_{i=1}^{s_k} c_e \times w^{+}{}_{i,k}) + (\sum_{k=1}^{N} \sum_{i=1}^{s_k} c_s \times w^{-}{}_{i,k})$$

Constraints
The following can be typical constraints to this problem:
(i) A worker must work at most M days in a week.
(ii) A worker can work at most d_c consecutive days.
(iii) A worker must rest at least a given number of hours between shifts.
(iv) A worker must work at most a given number hours in a day.
(v) A maximum number of shortage/excess of workers is defined.

PROPOSED ALGORITHM DESCRIPTION

The proposed algorithm starts with an initial solution, which consists of (i) an initial number of workers (ii) the assignment of such workers into shifts. Normally it is expected that the cost of shortages (c_s) is greater than the cost of per worker (c_w) and that the cost of excess (c_e) is greater than cero. In this case, if the assignment of workers results in no shortages or excess of workers, the solution is optimal and the algorithm stops; if not, a new worker is brought in and the workforce is re-assigned into shifts. Every time a worker is brought in, a new solution and value of the objective function are calculated and the best solution is stored. This procedure is repeated until a termination criterion is met.

Notice that a feasible solution with neither excess nor shortage of workers may not be the minimum cost solution because it depends on the values of c_w, c_e, and c_s. Thus, a zero shortage and zero excess of workers cannot typically used as a termination criterion. Usually, the algorithm terminates after a pre-defined number of iterations.

Initial Solution (S_0)

The initial solution (S_0) corresponds to the initial number of workers (W_0) and their initial distribution over the shifts and workdays. The procedure to distribute W_0 consists in assigning each employee progressively into each shift until completing the personnel demand for that shift. If it is not possible to satisfy demand due to shortages of available workers, the maximum number of available workers is assigned to that shift.

If all workers can be optimally assigned (i.e. no worker shortage or excess at all shifts) the algorithm stops. The objective function $C_{best} = C_{initial}$ is calculated and the current solution is stored $S_{best} = S_{0.}$

The only pattern to find the initial number of workers W_0 is given by the values of the personnel demand, but the costs (c_s, c_w, c_e) influence the value that should be assigned to W_0.

Typical results show that if $c_w > c_s$ it is very possible to obtain a solution with shortages. In this case W_0 is taken as the minimum worker requirements over the daily demands.

On the other hand, if $c_w < c_f$ it is very possible to obtain a solution with no shortages. If shortages are not allowed at any shift, the initial number of workers can be set either as the maximum or the average number of workers over the daily demand.

Depending on the value of c_e a solution with excess of workers might be obtained in some of the shifts.

In the case where c_e is a very large value in comparison to the other costs and if $c_w < c_s$ it might be viable to try with both the maximum or the average number of workers over the daily demand.

Iterative Procedure

1. Given an initial solution S_0, bring in a new worker w'.
2. Assign the new worker w' to those shifts with the greatest shortages.
3. If the worker w' completes M workdays a new solution (S_{new}) was found and go to step 7 ; if not go to step 4:
4. For the new worker w', select a feasible shift s_a with no shortages and with the minimum excess of workers. Check if another worker w_a already assigned to this shift, can be assigned to a feasible shift with shortages s_b.
5. If such a worker exists, the new worker w' is assigned to the s_a shift and worker w_a is assigned to the shift $s_{b.}$ If not, the new worker w' is assigned to s_a.
6. Go to step 3.
7. Calculate the objective function value C_{new} of the new solution S_{new}.
8. If $C_{new} \leq C_{best}$ then $S_{best} = S_{new}$, and $C_{best} = C_{new.}$
9. Go to step 1 until the termination criterion is met.

Termination criterion

The termination criterion consists of establishing a limit number of iterations (N_{max}). Several trials are needed to establish the N_{max} number, depending on the available CPU time and the desired solution quality.

The following example (example 1) shows the execution of the algorithm. Figure 1 (shown below) describes the proposed algorithm iterative procedure.

Step 1:

Initial solution (S_0): Let an initial solution with W = 5 workers assigned to meet the demand in three shifts per day, considering two work days and that a worker can work at most two consecutive days and can not be assigned to more than one shift per day.

First each worker is progressively assigned into shifts until meeting the personnel demand for each shift. For example, workers 1 and 2 are assigned to the Monday morning shift. Next workers 3 and 4 are assigned to the Monday afternoon shift and so forth. Notice that with five workers it is not possible to meet the demand on the Tuesday night shift. Table 2 shows the initial assignment of each worker into the morning (M), Afternoon (A) and night shifts (N) of the two work days (Monday and Tuesday) considered in this problem.

Table 2. Initial workforce assignment to shifts

Day	Shift	Worker demand	Assigned Workers			Shortage of workers per shift	Excess of workers per shift
Monday	M	2	1	2		0	0
Monday	A	2	3	4		0	0
Monday	N	1	5			0	0
Tuesday	M	3	1	2	3	0	0
Tuesday	A	1	4			0	0
Tuesday	N	2	5			1	0

Step 2:

Since this assignment implies a shortage of workers, a new worker is brought in (W=6) and she is assigned initially to the night shift of Tuesday which is the only shift with shortages. Table 3 shows the assignment of the workforce until this step.

Table 3. Temporary workforce assignment to shifts

Day	Shift	Personnel demand	Assigned Workers			Shortage of workers per shift	Excess of workers per shift
Monday	M	2	1	2		0	0
Monday	A	2	3	4		0	0
Monday	N	1	5			0	0
Tuesday	M	3	1	2	3	0	0
Tuesday	A	1	4			0	0
Tuesday	N	2	5	6		0	0

Step 3:

Since the new worker has to work two days and she has been assigned to only one workday, it is necessary to go to steps 4 and 5:

Steps 4 and 5:

In this step the new worker (worker 6) is assigned to a feasible shift with no shortages and with the minimum excess of workers. In this case all the shifts have no shortages and no excess of workers per shift. Considering the constraint that establishes that a worker can not be assigned to more than one shift per day, the feasible shifts in which worker 6 can be assigned are the morning, afternoon and night shifts of Monday. In this case the morning shift of Monday is chosen, because it is the earliest shift of the day. Table 4 shows the final assignment of the workforce.

Table 4. Final workforce assignment to shifts

Day	Shift	Personnel demand	Assigned Workers			Shortage of workers per shift	Excess of workers per shift
Monday	M	2	1	2	6	0	1
Monday	A	2	3	4		0	0
Monday	N	1	5			0	0
Tuesday	M	3	1	2	3	0	0
Tuesday	A	1	4			0	0
Tuesday	N	2	5	6		0	0

Step 6:

At this point all workers complete two workdays. Thus it is necessary to go to steps 7, 8 and 9 where the new value of the objective function is calculated (step 7). The actual best solution is updated (step 8). Step 9 checks whether the termination criterion (number of iterations) is met.

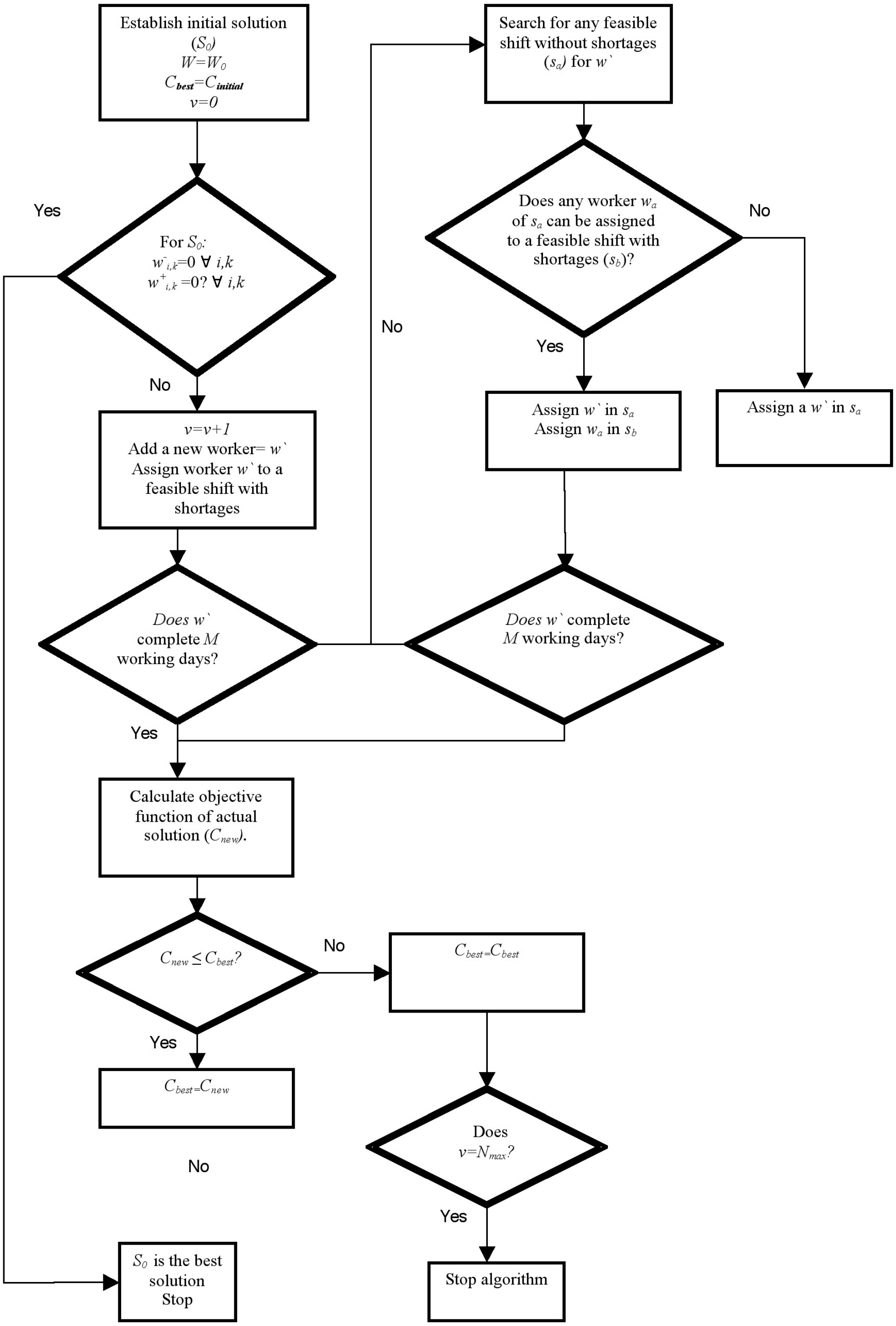

Figure. 1. Flow Diagram of the algorithm.

Volume 3, Number 2, 2006, pp. 35-48

To summarize: In the example 1 the initial solution includes shortages in the night shift of Tuesday, so an extra worker is added. She is assigned to this shift and also she is assigned in another shift to complete M working days, incurring in an excess of workers in the morning shift of Monday.

ALGORITHM PERFORMANCE

The proposed algorithm was implemented in VISUAL BASIC and run on a PC having 512Mb RAM.In order to study the performance of the proposed algorithm, we formulated a special instance of this problem as a Mixed Integer Program (MIP). In this instance, the number of shifts in every day of the week is the same and shortages are not allowed at any shift. This formulation takes the number of workers and the number of shifts per day as input data. This MIP formulation was written and run in GAMS®. We used a set of randomly generated test problems.

MIP Formulation

$$\text{Min} \sum_{j=1}^{W} \sum_{k=1}^{N} \sum_{i=1}^{T} x_{i,j,k}$$

s.t

$$[1] \quad \sum_{j=1}^{W} x_{i,j,k} \geq W_{i,k} \quad \forall i = 1,2,...T; \quad = 1,2,...N$$

$$[2] \quad \sum_{i=1}^{T} x_{i,k,j} \leq 1 \quad \forall k = 1,2,...N; \quad \forall j = 1,2,...W$$

$$[3] \quad \sum_{k=1}^{N} \sum_{i=1}^{T} x_{i,j,k} \leq M \quad \forall j = 1,2,......W$$

$$[4] \quad \sum_{i=1}^{T} x_{i,j,k} + \sum_{i=1}^{T} x_{i,j,k+1} + ... \sum_{i=1}^{T} x_{i,j\ k+D_c} \leq D_c \quad \forall k = 1,........N; \forall j = 1,......W$$

$$[5] \quad x_{T,j,k} + x_{1,j,k+1} \leq 1 \quad \forall k = 1,...N; \quad \forall j = 1,.......W$$

$$[6] \quad x_{i,j,k} \in \{0,1\} \quad \forall j = 1,2,...W; \quad \forall i = 1,2,...T; \quad \forall k = 1,2,...N$$

Where:
N: Number of workdays in a week.
$W_{i,k}$: Number of workers required in the shift i of the day k.
W: Total number of workers.
T: Number of shifts per day.
d_c: Maximum number of consecutive workdays a worker may be assigned.
M: Maximum number of workdays in a week.

$$x_{i,j,k} \begin{cases} 1 \text{ if the employee j is assigned to the shift i of the day k.} \\ 0 \text{ otherwise} \end{cases}$$

Explanation of the constraints

1. This set of constraints establishes that the personnel demand of the shift i of day k must be satisfied.
2. This set of constraints establishes that a worker may work at most one shift per day.
3. This set of constraints establishes that a worker may work at most M days per week.
4. This set of constraints establishes that a worker may work at most d_c consecutive days per week.
5. This set of constraints establishes that a worker may work at most M days per week.
6. This set of constraints establishes that a worker cannot be assigned in the last shift of one day k and in the first shift of the next day (k+1).

Establishing the optimal MIP solution

1. Set the total number of workers W as the best number of workers obtained with the algorithm allowing no shortages.
2. Find the optimal solution to the MIP formulation.
3. Set W = W - 1 and solve again the MIP problem. If the new problem is infeasible then the minimum number of workers was found with the algorithm; if a new optimal solution is found, the solution obtained with the algorithm was not optimal because the demand could have been met with fewer workers.

Results

We tested 10 different problems using both the mathematical formulation and the algorithm. The termination criterion for each problem was 100 iterations. These problems were randomly generated considering different ranges of personnel demand per shift, different maximum number of consecutive days and different number of shifts per day. Table 5 shows a description of the sizes of the tested problems:

Table 5.Description of the sizes of the tested problems

Problems	M	d_c	T	Range of demand per shift
Problem 1	5	4	3	30-80
Problem 2	5	3	3	30-80
Problem 3	5	4	3	115-170
Problem 4	5	3	12	30-45
Problem 5	5	3	12	65-85
Problem 6	5	3	8	115-150
Problem 7	5	3	3	295-445
Problem 8	5	5	5	18-25
Problem 9	5	5	6	10-20.
Problem 10	5	5	2	15-25

Table 6 shows the results obtained in all tested problems with both the proposed algorithm and the MIP solution:

Table 6 Proposed algorithm and the MIP solution results

	Proposed Algorithm Results			MIP Results	
Problems	W0	Best Solution(W)	CPU Time (sec)	Optimal Solution(W)	CPU Time (sec)
Problem 1	214	217	15	217	50
Problem 2	202	227	23	227	90
Problem 3	567	568	10	568	150
Problem 4	645	655	210	655	570
Problem 5	1269	1275	253	1275	2700
Problem 6	1492	1515	230	1515	2040
Problem 7	1506	1528	170	1528	310
Problem 8	143	144	45	144	20
Problem 9	110	111	40	111	15
Problem 10	70	70	20	70	10

CONCLUSIONS

The results show the validity of the proposed algorithm: In at least 10 tested problems we obtained the optimal solution. However, it is necessary to consider more scenarios than the ones considered in the MIP, so it was not possible to make a more profound and detailed analysis about the performance of the algorithm, but certainly we could establish that the proposed algorithm is able to obtain very good results making it a viable tool to implement.

The solutions obtained with the algorithm vary with the initial number of workers: However using the criteria and values suggested above to determine the initial number of workers, we always found the optimal solutions.

The algorithm shows good behavior in terms of execution times: In almost all the tested problems the execution times were significantly shorter than the ones obtained with the MIP solver. The greater differences in the execution times occur in the more complex and larger problems.

REFERENCES

Aykin,T. (1998). "A composite branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows". The Journal of the Operational Research Society, Vol 49 No 6,pp. 603–616.

Balakrishnan, N and Wong, R. (1990) "A network model for the rotating workforce scheduling problem". Networks, Vol 20, pp 25–42.

Best. (1991) "Guidelines for shift workers". Bulletin of European Time Studies No.3, European Foundation for the Improvement of Living and Working Conditions.

Brusco, J. and Jacobs, L. (1993). "A simulated annealing approach to the cyclic staff-scheduling problem. Naval Research Logistics, Vol 40, pp 69-84.

Gartner, J. and Wahl, S. (1998) "The significance of rota representation in the design of rotas". Scandinavian Journal of work, Environment and Health, Vol 24 No 3, pp 96-102.

Hung, R. (1993) "A three-day workweek multiple-shift scheduling model". Journal of Operational Research society of Japan, Vol 44, pp141-146.

Hung, R. (1994) "A multiple shift workforce scheduling model under 4 day workweek with weekday labor weekend demands". Journal of Operational Research society of Japan, Vol 45 No 1, pp 1088-1092.

Jackson, W.K. and Havens, W.S. and Dollard, H. (1997) "Staff scheduling: A simple approach that worked". Technical Report CMPT97-23, Intelligent Systems Lab, Centre for Systems Science, Simon Fraser University.

Laporte, G. (1999) "The art and Science of rotating schedules". Journal of Operational Research Society, Vol 50, pp 1011-1017.

Lau, H.C. (1996a). "Combinatorial approaches for hard problems in Manpower

Volume 3, Number 2, 2006, pp. 35-48

Scheduling". Journal of Operational Research society of Japan, Vol 39 No 1, pp 88-98.

Lau, H.C. (1996b) "On the complexity of manpower scheduling". Computers and Operations Research, Vol 23 No 1, pp 93-102.

Mejia, G. and Montoya, C (2007) "Resource assignment and scheduling using petri nets and the heuristic search". Proceedings of the ICPR 19 (International Conference on Production Research), Valparaiso, Chile.

Musliu, N. (2001) "Intelligent Search Methods for Workforce Scheduling: New ideas and practical applications". P.h.D Thesis, Fakultat Fur, Tecnische Universitat Wien, Vienna, Austria.

Biography

Carlos Montoya is currently an instructor at the industrial engineering department at the Universidad de los Andes in Bogotá, Colombia. He received a B.Sc degree in Industrial Engineering from the Universidad de Los Andes, Bogotá, Colombia (2005), and a M.Sc. degree in Industrial Engineering from the Universidad de Los Andes (2007). He has also worked as academic and research assistant between 2005 and 2007. His research interests include Production Scheduling, Workforce Scheduling and applications of AI in Production Systems. He has published papers on production systems in domestic and international conferences and currently he has two papers under review in international journals and one already accepted for publication.

Gonzalo Mejía is currently an assistant professor at the industrial engineering department at the Universidad de los Andes in Bogotá, Colombia. He received both his B.Sc and M.Sc degrees from the same university and his PhD from Lehigh University in 2003. His research is concentrated on manufacturing systems modeling and scheduling. In particular, he is focused on Petri net modeling approaches for hierarchical modeling of real-life manufacturing systems. He teaches both at the undergraduate and graduate level. He also has participated in the development of industrial scheduling systems. He is a member of IIE.