

CAHIER DU LAMSADE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
(Université Paris-Dauphine)
Unité de Recherche Associée au CNRS n° 825

THE SHIFT SCHEDULING PROBLEM: DIFFERENT FORMULATIONS AND SOLUTION METHODS

CAHIER N° 146
juillet 1997

Eric JACQUET-LAGRÈZE ¹
Denis MONTAUT ¹
Ariane PARTOUCHE ²

received: April 1997.

¹ Eurodécision – 9, rue de la Porte de Buc – 78000 Versailles, france; e-mail: ejl@eurodecision.fr

² LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16;
e-mail: partouche@lamsade.dauphine.fr

Contents

Résumé-Abstract	i
1 Introduction	1
2 Description of the Shift Scheduling Problem	3
2.1 Definitions	3
2.2 Assumptions for the experimental study	3
3 Testing an implicit formulation of the SSP	4
3.1 The implicit model	4
3.1.1 Introduction to the model	4
3.1.2 Data	4
3.1.3 Variables	5
3.1.4 Objective	5
3.1.5 Constraints	5
3.2 The ILP approach	6
3.3 The CLP approach	7
3.4 Experimental results	8
3.4.1 Test on a very small problem with 10 periods	8
3.4.2 Test on a small problem with 48 periods	9
4 Testing two resolutions of the set covering formulation	9
4.1 The model SC	9
4.2 The ILP approach	10
4.3 The CLP approach	10
4.4 Experimental results	11
4.4.1 Set-covering model tested on a very small problem with 10 periods	11
4.4.2 Set-covering model tested on a small problem with 48 periods	11
4.4.3 Comparison of the 4 approaches	11
5 Improvement of the set covering formulation solved by ILP for large scale applications	12
5.1 Introducing flexibility	12
5.2 The heuristic method proposed	13
5.3 Quality of solutions relative to the level of flexibility	14
6 Conclusion	16
References	16

La Construction de Vacations : Différentes Formulations et Méthodes de Résolution

Résumé

Le problème de construction de vacations (SSP) consiste à déterminer un ensemble de vacations permettant de couvrir une courbe de charge étendue et variable tout en minimisant les coûts. La formulation du SSP comme un problème de couverture d'ensembles a été largement étudiée et une variété d'heuristiques de résolution ont été proposées. Après un tour d'horizon de la littérature, nous proposons une modélisation implicite du problème que nous résolvons d'une part par la Programmation Linéaire en Nombres Entiers, d'autre part à l'aide de la Programmation Par Contraintes. Nous testons également ces deux méthodes de résolution sur la formulation par couverture d'ensembles et comparons les quatre approches pour finalement conclure à la nette supériorité de la PLNE avec le modèle classique.

Nous proposons alors une heuristique quasi-optimale basée sur la PLNE qui permet un fort niveau de flexibilité et nous mesurons de manière expérimentale les effets d'une meilleure flexibilité sur la productivité.

Mots clés : vacations, programmation linéaire en nombres entiers, programmation par contraintes, flexibilité.

The Shift Scheduling Problem: Different Formulations and Solution Methods

Abstract

The shift scheduling problem (SSP) consists of determining the best set of shifts to cover an extended and variable requirement curve, minimizing the cost. The formulation of the SSP as a set-covering problem has been much studied and many heuristic approaches have been proposed. After a literature review, we suggest an implicit model of the SSP. We solve it both with Integer Linear Programming and with Constraint Logic Programming. We also test those two resolution methods on the set-covering formulation and compare the four approaches to finally conclude that ILP on the classical set-covering model gives much better results.

Then we propose an almost optimal heuristic based on ILP, that allows a high level of flexibility and we experimentally measure the benefit of more flexibility on productivity.

Key-words: shifts, shift scheduling, integer linear programming, constraint logic programming.

1 Introduction

Manpower scheduling problems occur in organizations for which the operating day exceeds the staff working day or the operating week is longer than the legal work week and the demand varies over each time period. Examples include the scheduling of nurses in hospitals, operators in telephone companies and ground crews in airports. To provide continuous service and to meet the requirements for all activity periods, employees work on multiple-shifts: the problem is then to schedule working time and rest time for every employee.

Research interest in manpower scheduling problems focuses on three kinds of problems: *the shift scheduling problem*, *the days-off scheduling problem* and *the tour scheduling problem*. The first problem consists of finding the best set of allowable shifts to cover the staff requirement for each time period. The planning horizon is usually a day. The second and third problems aim at constructing schedules for several consecutive weeks, indicating if each day is a working day or not for the days-off problem and determining the working hours for the tour scheduling problem. This paper deals with the Shift Scheduling Problem (SSP).

As early as 1954, Dantzig formulated the SSP as a set-covering problem. He solved it without the integrity constraints and noticed that, even if the optimal solution might be fractional, it hardly was. He also noted that the insertion of breaks causes a large increase in the number of decision variables. Moreover, Bartholdi (1981) showed that this problem is NP-complex. Although solving the relaxed problem is relatively easy, finding an optimal integer solution can become very difficult for realistic problems.

Many approaches for solving the SSP have been proposed since Dantzig. As shown by Segal (1974), the problem ignoring breaks can be modeled as a network-flow problem, for which the variables are guaranteed to be integral as long as the requirements are. But this formulation is far from most real world situations and researchers have worked on problems including breaks. Segal himself considered a scheduling problem for telephone operators in which the shifts may include one lunch break and up to two relief periods, depending on their length. He solved separately the shift assignment by his network-flow formulation and the break placement by a heuristic approach.

Another kind of approach is to consider the relaxed linear program and round the solution by a more or less sophisticated heuristic. Keith's heuristic (1979) for instance, starts with a rounded solution of the relaxed LP, allowing understaffing, and adds an agent on the shift which maximizes the understaffing reduction, until no understaffing persists. The last step consists of dropping an agent from a shift if it reduces overstaffing while simultaneously creating no understaffing.

Other heuristic approaches to this problem, mainly heuristics based on linear programming but also some construction heuristics, can be found in Henderson and Berry (1976, 1977); Morris and Showalter (1983); Bailey and Field

(1985); Bechtold and Showalter (1987)... Bechtold *et al.* (1991) conducted a comparative analysis of several heuristics and showed the superiority of the heuristics of Keith (1979) and Morris and Showalter (1983). Morris and Showalter also suggest different ways to round the relaxed LP solution.

The different approaches found in the literature usually suppose no understaffing. Jacquet-Lagrèze and Meziani (1988) worked on allowable understaffing. Looking for a balance between cost and service quality, they pointed out the bicriteria aspect of the problem. Easton and Rossin (1991) suggested equivalent alternate solutions to integrate several criteria.

In his thesis, Thompson (1988) compared a procedure that starts with a break-less LP formulation and insert the meal breaks in a second step to a LP formulation which includes only a single break position. As the first approach gave very poor results, Thompson concluded that more flexibility in the allowable break options is essential for the minimisation of the labor amount scheduled.

Bechtold and Jacobs (1990) pointed out that the explicit representation of the shifts in Dantzig's model implies a dramatic increase of the number of alternate shifts when a higher flexibility is desired (starting time, shift length, assignment of breaks). For this reason, they present an implicit formulation of the breaks to match explicitly represented shifts. Their model, tested with four different labor requirement patterns and ten shift-length combinations proved to be better than the Dantzig's model in terms of execution time and memory size requirements. Thompson (1995) improved this latter model by implicitly matching meal breaks to implicitly represented shifts. His variables represent first the number of shifts from the same shift pattern that begin (or end) at any period, and second, the number of breaks for each shift pattern, beginning at each period. He considers thus about 15,000 alternate shifts and succeeds solving the problem. Recently, Aykin (1996) proposed another implicit approach to treat the problem with multiple break windows. All the possible shifts are enumerated explicitly without breaks, and for each shift and break window, he defines a set of variables, each one representing the number of shifts that begin their break at each possible start time of the break window. Compared to the explicit formulation, this approach can save a large number of variables when shifts involve several breaks as opposed to one. Note, however, that due to the problem size, all these models are solved by heuristic procedures.

For several years the authors of this paper have been working on the SSP for major French companies including Air France and Aéroports de Paris. The emergence of Constraint Logic Programming (CLP) methods (see Esquirol *et al.* (1995); Ilog (1995)) provided an opportunity to test an implicit formulation of the problem, avoiding the difficulty of dealing with the huge number of alternate shifts that are generated in the classical approach. *Section 2* describes the SSP. *Section 3* relates our experience with the implicit formulation for which we compare two resolution methods: CLP and Integer

Linear Programming (ILP). Both methods lead to very disappointing results when compared to the classical approach as it is shown in *Section 4*. *Section 5* presents research work done on the classical set covering formulation in order to solve large scale industrial problems whose size increases with the demand of companies for more flexibility and efficiency. First, we propose a heuristic yielding an almost optimal integer solution. Second, we conduct an experimental study to evaluate the quality of the solutions as flexibility increases.

2 Description of the Shift Scheduling Problem

2.1 Definitions

The planning period is the smallest time interval for which the staff requirement is defined. This is often simply called period.

A shift describes the presence of one or more employees on a day, indicating the work and break periods. It is characterized by a beginning time, an ending time, up to three break times, all corresponding to the beginning of a planning period. Each shift has a certain **cost**.

The shift length is the number of planning periods (often expressed as a number of hours) between the shift beginning and ending times, including the break lengths.

The length step is the minimum difference in possible shift lengths. It is a multiple of the planning period.

The beginning window is the time interval during which a shift should begin.

The beginning step is the minimum time interval between two consecutive shift beginnings. It is a multiple of the planning period.

A break is a succession of planning periods during which the employee does not cover the requirement.

The break window is the time interval during which the break can be assigned.

The break step is the minimum time interval between two consecutive alternate breaks. It is a multiple of the planning period.

2.2 Assumptions for the experimental study

The following assumptions correspond to the experimental study conducted at Aéroports de Paris.

1. The planning period is 10 minutes (requirement is given for each 10 minutes period).
2. The shift length is limited by a minimum length and a maximum length.
3. All breaks are of the same length, an integer multiple of the planning period (meal-breaks only, up to two per shift).
4. The break windows are the same for all shifts. There is mainly one break window for each shift but some very long shifts cover two break windows and some short ones do not cover any.
5. The shift cost is a linear function of the shift length.
6. Understaffing is not allowed.

3 Testing an implicit formulation of the SSP

3.1 The implicit model

3.1.1 Introduction to the model

Our implicit formulation of the problem aims at finding a set of individual shifts, assuming two main decision variables for each employee: beginning and ending times of his shift. In order to model the breaks, we need for each shift one additional variable per break (since we assume that all breaks are of the same length). Typically, for a situation involving 100 employees and allowing a high level of flexibility scheduling (10 minutes for all steps), number of main decision variables rises here to only 300 against about 100,000 in the classical set covering approach (*see section 4*). However, to test the formulation, we started this work with a simplified break-less version.

Both the CLP and ILP resolution methods require additional technical variables. First, the number of shifts in the solution is not known. We thus have to consider a maximum number of shifts and the possibility for the model to generate a smaller number (thanks to binary *variables* b_j). Second, we must introduce binary variables for the requirement covering constraints (*variables* a_{ij} , which are data in a classical set covering formulation).

3.1.2 Data

n	=	number of planning periods i ,
m	=	maximum number of shifts j ,
w_i	=	requirements for period i ,
w_{max}	=	maximum requirement,
d_{min}	=	minimum shift length,
d_{max}	=	maximum shift length,
e_{max}	=	maximum total surplus for the entire requirement curve.

3.1.3 Variables

$$\begin{aligned}
t_j &= \text{first period of shift } j \text{ (beginning period),} \\
f_j &= \text{first period after the end of shift } j \text{ (not included in the} \\
&\quad \text{shift),} \\
d_j &= \text{shift length } j \text{ (} d_j = f_j - t_j \text{),} \\
b_j &= \begin{cases} 1 & \text{if shift } j \text{ is selected} \\ 0 & \text{if not,} \end{cases} \\
a_{ij} &= \begin{cases} 1 & \text{if period } i \text{ is covered by shift } j \\ 0 & \text{if not,} \end{cases} \\
e_i &= \text{surplus in period } i,
\end{aligned}$$

3.1.4 Objective

We want to minimize the overall work time:

$$\min F_0 = \sum_{j=1}^m d_j b_j \quad (1)$$

Note that for any solution which covers the requirements, we have:

$$F_0 = \sum_{i=1}^n w_i + \sum_{i=1}^n e_i \quad (2)$$

Therefore, it is equivalent to minimize F_0 or the overall surplus F_1 :

$$\min F_1 = \sum_{i=1}^n e_i \quad (3)$$

3.1.5 Constraints

Constraints for variables definition:

Shift length:

$$d_{min} \leq f_j - t_j \leq d_{max} \quad \forall j \quad (4)$$

Covering indicator definition a_{ij} :

$$\begin{cases} \text{if } b_j = 1 \text{ and } (t_j \leq i \text{ and } i < f_j) \text{ then} \\ \text{else} \end{cases} \quad \begin{cases} a_{ij} = 1 & \forall i, j, \\ a_{ij} = 0 & \forall i, j \end{cases} \quad (5)$$

Requirement covering constraints:

$$\sum_{j=1}^m a_{ij} - e_i = w_i \quad \forall i \quad (6)$$

3.2 The ILP approach

The use of LP leads us to model the problem in a specific way. We assume here that if a shift is not selected, its length is null.

- The criterion minimized F_0 then becomes $\sum_{j=1}^m d_j$;
- The formulation of the shift length constraint must allow null length for non selected shifts ($b_j = 0$). Thus (4) becomes:

$$d_{min}b_j \leq f_j - t_j \leq d_{max}, \quad \forall j \quad (7)$$

Note that it is not necessary to introduce b_j in the right hand side of the constraint since optimizing F_0 will give automatically null length for non selected shifts.

- The definition of the covering indicator a_{ij} is slightly different compared to constraint (5). For a selected shift j ($b_j = 1$), $a_{ij} = 1$ if shift j includes the planning period i and is *useful* to cover the requirement w_i . On the opposite, $a_{ij} = 0$ if shift j does not include period i or creates surplus in this period, i.e.,

$$a_{ij} = \begin{cases} 1 & \text{if } i \in [t_j, f_j[\text{ and shift } j \text{ is useful in period } i, \\ 0 & \text{otherwise.} \end{cases}$$

This is expressed by:

1. the constraint:

$$t_j - T(1 - a_{ij}) \leq i \leq f_j + T(1 - a_{ij}) - 1 \quad (8)$$

where T is a constant large enough. In the linear model with $T = n$, this constraint becomes:

$$t_j + na_{ij} \leq i + n, \quad \forall i, j \quad (8.1)$$

and

$$f_j - na_{ij} \geq i - n + 1, \quad \forall i, j \quad (8.2)$$

(8.1) makes $a_{ij} = 0$ for $i < t_j$ and (8.2) makes $a_{ij} = 0$ for $i \geq f_j$.

2. When i is in the interval $[t_i, f_i[$, the requirement covering constraints (6) force a_{ij} to be equal to 1 if it is necessary.
3. Moreover, to make sure that all $a_{ij} = 0$ for a non-selected shift, we add:

$$\sum_{i=1}^n a_{ij} \leq d_{max}b_j \quad \forall j \quad (9)$$

3.3 The CLP approach

The model as defined in section 3.1 can be directly applied as CLP offers many modeling facilities. With Ilog Solver, constraint (5) will be simply formulated as follow:

$$a_{ij} = (b_j = 1) \text{ and } (t_j \leq i) \text{ and } (i < f_j) \quad (10)$$

If the model as described in section 3.1 is logically sufficient in order to describe the whole problem to a CLP solver, it is necessary to add some constraints which help to get better performance by reducing the combinatorial aspect. Such constraints are:

- Elimination of symmetrical equivalent solutions by forcing the solver to create all shifts by increasing beginning time:

$$t_j \geq t_{j-1} \quad (11)$$

- Bounds restriction: the number of shifts is at least equal to the maximum requirement

$$\sum_{j=1}^m b_j \geq w_{max} \quad (12)$$

- Help for shift generation on the first ascending part of the requirement curve: if $w_i > w_{i-1}$ and i is in this part of the curve, then we need at least $w_i - w_{i-1}$ shifts to start at period i . Otherwise, we can say that, for each i , the number of shifts to start should be less than w_i . To formulate this, we define some new variables:

$$s_{ij} = \begin{cases} 1 & \text{if shift } j \text{ starts at period } i, \\ 0 & \text{if not.} \end{cases}$$

This starting shift indicator is defined as follow:

$$s_{ij} = (b_j = 1) \text{ and } (t_j = i), \quad \forall i, j \quad (13)$$

and the constraint is:

$$\sum_{j=1}^m s_{ij} \leq \begin{cases} w_i - w_{i-1} & \text{for each } i \text{ on the first ascending part of the curve,} \\ w_i & \text{for the others } i. \end{cases} \quad (14)$$

Since the CLP approach does not optimize a function, we have to introduce a bound e_{max} and the following constraint:

$$\sum_{i=1}^n e_i < e_{max} \quad (15)$$

The bound will be iteratively decreased until the problem appears to be infeasible.

We stress the importance of the tuning strategy in order to achieve acceptable performance. It is necessary to finely control the order of the valuation of the variables. The most efficient method consists of giving values first to variables which should have a causal effect on others. In the present case, we gave values to the variables in the following order: b_j, t_j, f_j .

3.4 Experimental results

3.4.1 Test on a very small problem with 10 periods

Data:

- break-less shifts may start at each period,
- shift length from 4 to 6 periods,
- table of requirement

period	1	2	3	4	5	6	7	8	9	10
requirement	1	2	4	3	5	3	1	2	2	1

- the total requirement is 24

Results:

- an optimal solution given by both approaches:

number of shift	begin/end times	length
1	from 1 to 5	5
1	from 2 to 5	4
2	from 3 to 6	4
1	from 5 to 9	5
1	from 7 to 10	4

- the total length is 26, i.e., a surplus of 2
- execution times on a P486 100Mhz:
 - LP 220 seconds
 - CLP 4 seconds

3.4.2 Test on a small problem with 48 periods

Three tests were performed with different bounds on the shift lengths.

The ILP approach was not able to produce any feasible solution within 20 minutes. The CLP found, for the 3 tests, a first solution in about 7 seconds but these solutions have no reason to be optimal ones since the resolution was stopped when the first solution was found. On the 3 tests, this first solution happened to be optimal one time. For the two other ones, they were far from the optimum (see table 1). Moreover, the CLP showed no ability to prove optimality or non-optimality in a reasonable time.

Both approaches were thus very disappointing even if CLP showed a capacity to generate feasible solutions quickly. Without going further with more examples, these first results were so bad compared to the classical approach that we concluded that the implicit formulation could not compete with the classical one.

4 Testing two resolutions of the set covering formulation

In this section, we first introduce the classical set-covering formulation and then compare the performances of ILP and CLP approaches on its resolution. Finally, we give an overall comparison of the two formulations (implicit and set-covering) and the two resolution approaches (ILP and CLP).

4.1 The model SC

The set-covering approach consists of two steps:

1. generate a set of potential shifts;
2. select a subset of shifts which covers the staff requirement.

Note that in this approach, more than one employee can be assigned to any potential shift. Therefore, the decision variables represent the number of employees assigned to each shift.

(P)

$$\min \sum_{j \in \mathcal{J}} c_j x_j \quad (16)$$

s.t.:

$$\sum_{j \in \mathcal{J}} a_{ij} x_j \geq w_i, \quad \forall i \in \mathcal{I}, \quad (17)$$

$$x_j \geq 0, \text{ integer}, \quad \forall j \in \mathcal{J} \quad (18)$$

where

- n = number of planning periods,
- $\mathcal{I}$ = set of planning periods ($i = 1, 2, \dots, n$),
- m = number of feasible shifts,
- $\mathcal{J}$ = set of feasible shifts ($j = 1, 2, \dots, m$),
- x_j = number of employees assigned to shift j ,
- c_j = cost of one employee assigned to shift j ,
- $a_{ij} = \begin{cases} 1 & \text{if period } i \text{ is covered by shift } j \\ 0 & \text{if not,} \end{cases}$
- w_i = requirements for period i .

4.2 The ILP approach

This formulation of the SSP can be solved using a standard MIP package. For small problems, we obtain easily an optimal integer solution. Furthermore, in many larger problems, no branch and bound is required since the relaxed problem solution is integer. This can be explained by the almost unimodularity of the matrix. However, when the relaxed solution is fractional, it may be difficult to solve large scale problems without introducing some heuristic procedures. Such an heuristic is proposed in section 5.

Because of the usual difficulties met by branch and bound on large scale problems and to complete our comparison, we compared again the performances of this classical resolution method and the newer resolution approach which is the CLP, on the same formulation of the SSP. Some comparative results are given in section 4.4.

4.3 The CLP approach

We present here the CLP resolution of the set-covering formulation. For the needs of the method, the model is slightly different than the previous one. Note that, in a first time, we did not integrate the breaks in this approach.

The shifts library is represented as an object with three vectors: vector of beginning times, vector of ending times, vector of shift lengths. Variables x_j represent, as in the LP formulation, the number of employees assigned to shift j . The domain of allowable values incorporates the integrity constraints. Unlike the LP formulation, a_{ij} is here a variable:

$$a_{ij} = \begin{cases} x_j & \text{if planning period } i \text{ is covered by shift } j, \\ 0 & \text{if not.} \end{cases}$$

We also define a vector of variables A_i with: $A_i = \sum_{j=1}^m a_{ij}$.
The constraint (17) becomes then:

$$A_i \geq w_i \quad \forall i \in \mathcal{I}. \tag{19}$$

We define e_i , the surplus as the difference between A_i and w_i . We add some helpful constraints for the purpose of aiding the solution process by constraint propagation:

$$e_i \leq e_{max} \quad \forall i \in \mathcal{I} \quad (20)$$

$$\sum_{i=1}^n e_i \leq E_{max} \quad (21)$$

The last constraint corresponds to the objective function being minimised. The optimisation in CLP is performed by an iterative procedure. The procedure first obtains feasible solution and then add a new constraint to bound the objective function according to the last value found. Then it tries to find a new solution and so on.

4.4 Experimental results

4.4.1 Set-covering model tested on a very small problem with 10 periods

Data: correspond to 3.4.1. There is 18 alternate shifts.

Results:

- both approaches give a surplus of 2
- execution times on a P486 100Mhz:

ILP <1 second

CLP 41 seconds (12 iterations to find the optimal solution)

4.4.2 Set-covering model tested on a small problem with 48 periods

Three tests were performed with different bounds on the shift lengths, yielding to about 400 alternate shifts.

For the 3 tests, the ILP finds an optimal solution in less than 1 second. The CLP is much less efficient. For two tests out of three, a first solution was found within 1 second but no better one can be found within 20 minutes.

4.4.3 Comparison of the 4 approaches

Table 1 shows clearly that, among the 4 considered approaches, only the set-covering model, solved by ILP seems efficient for the SSP. The following section aims then at improving this approach to allow, at the same time, a high level of scheduling flexibility and a quasi-optimal resolution of the problem.

	SC ILP		SC CLP		Implicit ILP		Implicit CLP	
	CPU	obj	CPU	obj	CPU	obj	CPU	obj
test 1(10 per.)	<1"	2	41"	2	220"	2	4"	2
test 2(48 per.)								
2.1	<1"	171	(1)		(1)		7"	171 (2)
2.2	<1"	211	(1)		(1)		7"	225 (2)
2.3	<1"	447	(1)		(1)		6"	474 (2)

run time in seconds on a PC486 100Mhz

(1) no results were obtained within 20 minutes (2) first solution found

Table 1: Execution time and objective values compared

5 Improvement of the set covering formulation solved by ILP for large scale applications

5.1 Introducing flexibility

The first stage, before solving the SSP as a set-covering problem, is to build the library of shifts. For this, we first define all allowable shift patterns according to the initial assumptions described in section 2.2. Then, for each shift pattern, we generate exhaustively the allowable shifts. The size of this library is function of many parameters and especially the three steps defined in section 2: the length step, the beginning step, the break step.

Table 2 compares library sizes corresponding to different sets of steps, with or without breaks. We notice here that the break assignment, following rules 2 and 3 given in 2.2 and parameters given below, increases the number of feasible shifts about 35 times. The parameters used to build these libraries are those used at Aéroports De Paris:

- The minimum shift length is 4 hours, the maximum length is 12 hours.
- The beginning window starts at 4:30 and ends so that no shift ends on the next day (discontinuous problem).
- Break length is 40 minutes, break step is 10 minutes and break windows are (11:00-3:30) for lunch and (18:00-20:00) for dinner.

Note also that the planning period is here 10 minutes.

The sharper the steps are, the larger the library is and the better we may cover the requirements as it will be shown in table 4. However, the planning period is a lower bound for the sharpness of the steps. Indeed, the library generated with all the steps corresponding to one planning period (B10-10) is exhaustive and would assure the optimality of a solution found by an exact method*.

*Experimentations described further will not include the B10-10 library since such a

Name	Length step in minutes	Begin step in minutes	Number of Shifts		
			no break(a)	1-2 breaks(b)	b/a
B60-60	60	60	113	3678	32.54
B60-30	60	30	199	6879	34.56
B60-20	60	20	303	10,686	35.26
B60-10	60	10	574	20,079	34.98
B30-30	30	30	346	12,516	36.17
B30-20	30	20	511	18,421	36.04
B30-10	30	10	992	36,561	36.85
B10-10	10	10	3136	108,183	34.49

Table 2: Size of some shift libraries

5.2 The heuristic method proposed

Because of the integrity constraints, this formulation cannot be solved by exact methods when the shift library becomes large. For this reason, we propose an efficient heuristic based on the relaxed LP solution and branch and bound procedure.

In a first step, we solve the relaxation of problem (P) which we denote as (P'). After solving the (P') problem with different shift libraries, we found approximately 30% of the solutions to be integer. For the other solutions, approximately half of the non-zero variables were integer. We suppose then being not far from the integer optimal solution.

The second step consists of fixing all the variables whose value is zero and limiting the domain of the others (integer or not) to ± 2 . Then we start a branch and bound procedure. The algorithm is stopped when the first integer solution is reached.

We tested this heuristic with different libraries on seven 24 hours requirements corresponding to the weekly ground crew requirements at Aéroports de Paris. For two libraries, a relatively small one and a medium one, the results are presented in Table 3. Although the minimum objective value found for (P') is a lower bound of the integer optimum objective value, we compare it to the solution found by our heuristic (H). We indicate also the number of nodes in the branch and bound procedure and the execution time on a Unix Station (IBM Risc6000, 25T).

The average difference in the objective function value (obj) between (P') and (H) is around 0.3%, which is extremely small considering the initial imprecision. The heuristic solution is said to be "nearly-optimal". We also notice that the execution time, with a medium library containing 10,000 shifts, is always around the minute.

large library requires to implement a column generation approach, which was not in the scope of this paper.

Test 1 : Library B60-60 (3678 shifts)

Day	(P')	(H)		total CPU(sec)	Difference $\frac{H-P'}{P'}(\%)$
	obj	obj	nodes		
Monday	643.3	648.3	5	25	0.78
Tuesday	669.4	669.4	0	29	0
Wednesday	654	657	5	56	0.46
Thursday	834.8	837.1	4	21	0.27
Friday	636.3	638.3	3	35	0.31
Saturday	651.6	651.6	0	17	0
Sunday	743.3	744.7	2	31	0.19
Average					0.29

Test 2 : Library B60-20 (10686 shifts)

Day	(P')	(H)		total CPU(sec)	Difference $\frac{H-P'}{P'}(\%)$
	obj	obj	nodes		
Monday	596	597	1	35	0.17
Thuesday	609	614	3	69	0.82
Wednesday	612.8	616.1	4	43	0.54
Thursday	782.9	783.6	2	52	0.08
Friday	584.9	584.9	0	53	0
Saturay	527.2	527.9	3	67	0.13
Sunday	707.5	710.5	3	50	0.42
Average					0.31

(P')= relaxed LP; (H) = integer solution given by heuristic

Table 3: The heuristic efficiency tested with two different libraries

5.3 Quality of solutions relative to the level of flexibility

The large number of shift alternatives increases the complexity but improves the labor utilisation. We study here, using our heuristic, the improvement of the solution relative to the library size. We define two ratios of productivity P1 and P2:

R : Requirement, i.e., total number of hours required.

ProdH : Productive Hours, i.e., number of hours spent working, exclusive of the breaks.

PaidH : Paid Hours, i.e., including breaks.

Ratio P1 = R / ProdH (%).

Ratio P2 = R / PaidH (%).

Library name	# of Shifts	R	ProdH	PaidH	P1 %	P2 %	CPU sec
B60-60	3678	494	576	634	85.8	77.9	20
B60-30	6879	494	549	620	90.0	79.7	38
B60-20	10686	494	532	596	92.9	82.9	63
B60-10	20079	494	521	591	94.9	83.7	124
B30-30	12516	494	545	609	90.7	81.1	48
B30-20	18421	494	529	592	93.4	83.4	106
B30-10	36561	494	513	581	96.4	86.0	199

Note that $BX-Y$ corresponds to a length step of X minutes and begin step of Y minutes.

Table 4: Solutions quality compared to different level of flexibility

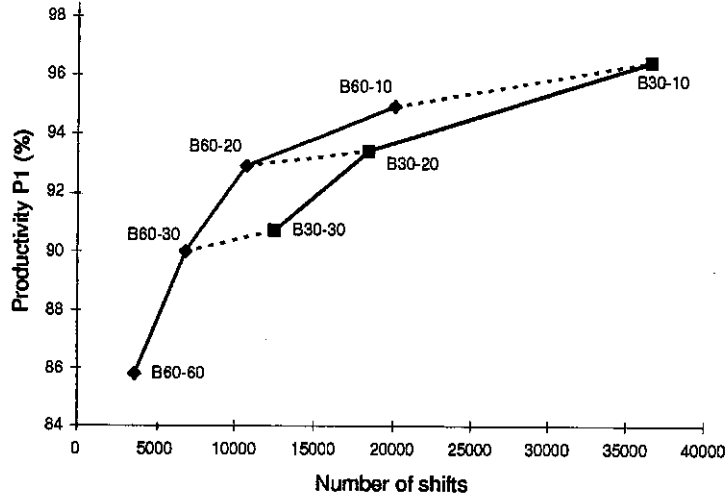


Figure 1: Productivity as a function of library sizes

Table 4 shows that, as the flexibility increases, it becomes much easier to meet the requirements generating less and less surplus. With the B30-10 library, the productivity level is already over 96%. Obviously, the B10-10 library would possibly provide a better solution but with a very small improvement of productivity compared to the increase of the problem size (108,000 variables instead of 36,000). Looking at the requirement and covering curves, we could also observe that, when the number of feasible shifts allows it, even the peaks of the staff requirement are covered without generating much surplus around.

Figure 1 shows the productivity evolution (P1 is represented) while length step and begin step are becoming sharper. This figure leads us to two observations. First, and that is not very surprising, the improvement rate is very strong when flexibility is low and become weaker as flexibility increases. Second, and this is a more specific result of our experimental study, we can

notice that **precision on the begin step is more efficient than precision on the length step**. That is shown by the fact that the increase rate is stronger for the plain curve than for the dotted one. Consider for instance results obtained with B60-30. We clearly see that, increasing the size of the library, it is better to move to B60-20 (more precision on begin step) than to B30-30 (more precision on the length step). In this particular case, we can even notice that B60-20 dominates B30-30 with regard to the two criteria, productivity and number of shifts. If we consider now moving from B60-20, it is also better, in terms of productivity, to sharpen the begin step (B60-10) than the length one (B30-20). Even if this time, there is no dominated solution since B60-10 is a bit larger than B30-20, the improvement brought by B60-10 compared to B60-20 is only about 0.5% whereas the number of shifts is almost double. Of course, execution time increases also with flexibility but 3-4 minutes is still a reasonable time for getting such excellent results.

6 Conclusion

In the first part of this paper, we have compared two models and two resolution approaches to solve the shift scheduling problem. Even if CLP resolution may have been improved by introducing additional constraints like cumulative constraints, various symmetry elimination or cut techniques, and may have benefited from a more complex instantiation strategy, the gap in results of each approach shows the preponderance of a set-covering formulation using ILP for this kind of problem.

The ILP success can be explained by the compact representation provided by the set-covering model and by the quasi-unimodular structure of the matrix. Moreover, constraints like covering constraints or an objective function which is a weighted sum of all variables are difficult to integrate in a CLP resolution since a bound on a sum do not immediately prune the domain of the variables concerned. Nevertheless, in other studies, CLP had proved to be an interesting way to solve very constrained problems (see Smith *et al.* (1995); Partouche (1996)). In further work, it would be interesting to display the general characteristics that make a problem more adapted to ILP or CLP. Both methods may also be associated to solve efficiently complex problems (see Jacquet-Lagrèze and Partouche (1997)).

In the second part, this paper shows the stakes in terms of productivity of increasing significantly the number of potential shifts, whenever flexibility is admitted in the organisation. Thanks to the computers power, it is now possible to solve almost optimally large scale problems with a standard LP package. Even when optimality is not reached, the potentiel gain on productivity (from 85% with 3,600 shifts to 96% with 36,600 shifts) is extremely convincing.

References

- Aykin, T. (1996). Optimal shift scheduling with multiple break windows. *Management Science*, **42**(4), 591–602.
- Bailey, J. and Field, J. (1985). Personnel scheduling with flexshift models. *Journal of Operations Management*, **5**(3), 327–338.
- Bartholdi, J. J. (1981). A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research*, **29**, 501–510.
- Bechtold, S. E. and Jacobs, L. W. (1990). Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, **36**(11), 1339–1351.
- Bechtold, S. E. and Showalter, M. J. (1987). A methodology for labor scheduling in a service operating system. *Decision Sciences*, **18**(1), 89–107.
- Bechtold, S. E., Brusco, M. J., and Schowalter, M. J. (1991). A comparative evaluation of labor tour scheduling methods. *Decision Sciences*, **22**, 683–699.
- Easton, F. F. and Rossin, D. F. (1991). Equivalent alternate solutions for the tour scheduling problem. *Decision Sciences*, **22**, 985–1001.
- Esquirol, P., Lopez, P., Fargier, H., and Schiex, T. (1995). Constraint programming. *Jorbel*, **35**(2), 5–36.
- Henderson, W. B. and Berry, W. L. (1976). Heuristic methods for telephone operator shift scheduling: An experimental analysis. *Management Science*, **22**, 1372–1380.
- Henderson, W. B. and Berry, W. L. (1977). Determining optimal shift schedules for telephone traffic exchange operators. *Decision Sciences*, **8**, 239–255.
- Ilog, editor (1995). *Ilog Solver and Ilog Schedule : first international users' conference*, Abbaye des Vaux de Cernay, France.
- Jacquet-Lagrèze, E. and Meziani, R. (1988). Linear programming and interactivity: a manpower scheduling DSS. In *IFORS 88*, volume Operational Research'87, pages 176–189.
- Jacquet-Lagrèze, E. and Partouche, A. (1997). A hybrid method for a staff scheduling problem. In Ilog, editor, *Third Ilog International Users Meeting*, Paris. to be published.
- Keith, E. (1979). Operator scheduling. *AIIE Transactions*, **11**, 37–41.

- Morris, J. G. and Showalter, M. J. (1983). Simple approaches to shift, days-off and tour scheduling problems. *Management Science*, **29**(8), 942–950.
- Partouche, A. (1996). Comparaison de deux approches sur un problème de planification d’horaires : la construction de cycles de travail. Cahier 142, Lamsade, Université Paris-Dauphine.
- Segal, M. (1974). The operator-scheduling problem: A network-flow approach. *Operations Research*, **22**, 808–823.
- Smith, B. M., Brailsford, S. C., Hubbard, P. M., and Williams, H. P. (1995). The progressive party problem: Integer linear programming and constraint programming compared. In Ilog, editor, *Ilog Solver and Ilog Schedule : first international users’ conference*, Abbaye des Vaux de Cernay France.
- Thompson, G. M. (1988). *A Comparaison of Techniques for Scheduling Non-Homogeneous Employees in a Service Environment Subject to Non-Cyclical Demand*. Ph.D. thesis, Florida State University.
- Thompson, G. M. (1995). Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*, **41**(4), 595–607.