ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Felici, C. Gentile

A POLYHEDRAL APPROACH FOR THE STAFF ROSTERING PROBLEM

R. 508 Settembre 1999

Giovanni Felici – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30 - 00185 Roma, Italy. Email : felici@iasi.rm.cnr.it.

Claudio Gentile – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30 - 00185 Roma, Italy. Email : gentile@iasi.rm.cnr.it.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica, CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

We consider a $0-1$ linear programming formulation for Staff Rostering Problems (SRP) that arise when the set of staff members is homogeneous, a 7-days-a-week service is to be provided, and some specific requirements on rest shifts have to be fulfilled. The objective function to be maximized expresses, in terms of weights on pairs of consecutive shifts, the global staff satisfaction. We consider some polytopes associated with SRP and study their structure. As a result we strengthen the original formulation and obtain very tight bounds at the root node of a Branch & Bound algorithm. Finally, some computational tests on real instances coming from the ground staff management of an airline company are presented.

Key words: Integer Programming, Polyhedral Methods, Rostering Problems.

1. Introduction

In this paper we consider a particular class of personnel management problems, referred to as Staff Rostering Problems (in the following, SRP).

Given a set of staff members P, a set of shifts S, and a minimum demand of each shift for all days of the week, one has to determine an assignment of the workers that satisfies the daily demand, respects the collective work agreement, and maximizes a measure of staff satisfaction. A staff member can cover any shift (i.e. the set of staff members is assumed to be homogenous), can be assigned only one shift per day, and has to be assigned a specific number of days off per week. The objective function takes into account staff satisfaction, expressed by the weighted sum of certain pairs of consecutive shifts.

The solution to the problem is represented by a *roster* design, that can be visualized by a table with a row for each staff member and seven columns indexed by the days of the week (see Figure 1). Note that a week starts on Monday and ends on Sunday. Each row of the table represents a sequence of shifts to be covered during a week (in the example, shifts are identified by their starting time). The sum of shift occurrences over a column respects the minimum demand of that shift for the corresponding day of the week.

Employee	MON	TUE	WED	THU	FRI	SAT	SUN
1	OFF	13.00	9.00	7.00	6.00	8.00	OFF
2	15.00	13.00	13.00	15.00	OFF	OFF	11.00
3	7.00	OFF	OFF	15.00	15.00	13.00	9.00
4	6.00	OFF	OFF	15.00	15.00	9.00	7.00
5	6.00	OFF	9.00	9.00	8.00	8.00	OFF
6	7.00	7.00	6.00	7.00	6.00	OFF	OFF
7	9.00	9.00	8.00	9.00	5.30	OFF	OFF
8	16.30	11.00	7.00	6.00	OFF	OFF	8.00
9	7.00	7.00	7.00	7.00	OFF	OFF	15.00
10	13.00	OFF	OFF	15.00	15.00	15.00	15.00
11	OFF	15.00	15.00	13.00	13.00	11.00	OFF
12	OFF	16.30	16.30	15.00	13.00	8.00	OFF
13	OFF	16.30	16.30	15.00	15.00	OFF	7.00

Figure 1: An example of roster table: a shift is assigned to each cell

Each staff member is assigned to a row of the *roster table* at the beginning of the planning period; at every new week, each member is assigned to the next row of the table, while the one in the last row is assigned to the first one. This circular assignment guarantees that, after $|P|$ weeks, the workload is equally distributed amongst all staff members.

The work agreement can impose different constraints on how to fill the cells of the roster table. Typically, shift workers are allowed two days off per week; some agreements allow to average this value over more than a week (e.g., one month), while others require each worker to have exactly two days off in every week. A rigid agreement may also specify the maximum number of work shifts that can be assigned to a worker in consecutive days and the number of days off assigned on weekends. Throughout the paper we refer to days off as particular shifts called *rest shifts*.

In this work, we focus on Staff Rostering Problems where rigid constraints are specified. In particular, we consider the following work agreement constraints: exactly two rest shifts in each week, no more than five consecutive work shifts, and at least one rest shift on Sunday every four weeks.

We make use of an integer programming formulation and propose a polyhedral approach for its solution that proved to be effective on large real world instances. The formulation of the problem is composed of three main classes of constraints, that we name:

- Structural Constraints;
- Work Agreement Constraints;
- Objective Function Related Constraints.

The *Structural Constraints* express the main relations between the integer variables; for example, they specify that every day a shift must be assigned to each staff member, or that a minimum number of staff members must be assigned to a given work shift. The *Work Agreement Constraints* are related to rules coming from the work agreement: for example, they include the constraints forbidding certain pairs of shifts to be scheduled in consecutive days, or constraints stating a bound on the number of work shifts that can be assigned in a given interval of the planning period. The objective function maximizes the staff satisfaction expressed by sequences of "good" shifts (e.g., rest shifts in consecutive days). As an approximation of these sequences, we maximize a weighted combination of the product of variables in two consecutive cells. We linearize the resulting model to apply linear integer programming methods, and introduce new binary variables linked to the previous variables by the *Objective Function Related Constraints.*

The paper is organised as follows: in Section 2 we review the previous work in the literature; in Section 3 we discuss an integer programming formulation of SRP; in sections 4 and 5 we present a particular relaxation of SRP and identify three classes of facet inducing inequalities, state their validity for the complete problem, and describe other classes of valid inequalities; in Section 6 we discuss a particular class of valid inequalities that are derived by solving certain subproblems of SRP; in Section 7 we discuss symmetry properties and branching rules; finally, in Section 8 we present some computational results for real instances provided by the ground staff management of the airline company Alitalia.

2. Literature review

Rostering problems have been extensively addressed in many research papers. They span several application fields, such as public services, hospitals, call centers and industry in general. Different techniques and methods have been proposed and analyzed in the related literature.

One of the most significant approaches to the problem is based on mathematical programming models and algorithms, where the problem is formalized as a set of linear constraints with a linear objective function. Due to the nature of the problem, the use of integer variables is required. Then, proper algorithms are applied in order to determine an optimal solution. This approach has been adopted in early works, where it proved to be successful only for simple problems ([9], [3]); mixed integer models are also considered in [4]. Also in this setting, several authors adopt a covering model, where the variables represent all the possible employee schedules and the objective is to find a subset of such schedules with optimal objective function value ([2], [12], and [10]). For large problems, it becomes nearly impossible to represent directly all possible schedules, and therefore column generation approaches have also been proposed in [7], where the schedules are generated by solving, at each iteration, a column generation subproblem.

Alternative approaches for rostering problems are based on network approaches. Here the problem constraints and the objective function are represented using the nodes and the arcs of a network, and solutions are determined by solving shortest path problems on such graphs ([1], [8]).

Although rostering problems may differ substantially from application to application, most of them belong to the $\mathcal{NP}$ complexity class and are thus difficult to solve (for example, see [2] for a proof of $\mathcal{NP}$-completeness of the cyclic staff scheduling problem). This fact has directed the effort of many researchers toward the use of heuristic strategies that provide good, but not necessarily optimal, solutions ([9], [11], and [6]). In [5] the problem of determining the minimum staff size is formulated and solved.

The work here presented relates to the first type of approach, that is, an exact mathematical model of the problem is adopted and special techniques are used to implement an algorithm capable of finding the optimal solution.

3. Integer Programming Formulation

We denote by P the set of staff members, by C the set of cells of the roster table, and by S the set of shifts, composed of a number of *work shifts* plus the *rest shift*. We associate with the seven days of the week the integers $\mathtt{1}, \mathtt{2}, \ldots, \mathtt{7}$, where $\mathtt{1}$ represents Monday, $\mathtt{2}$ represents Tuesday, etc., and $D = \{\mathtt{1}, \mathtt{2}, \ldots, \mathtt{7}\}$ the set of days in the week. The set C is used as a circular set indexed by the integers $0, \ldots, |C| - 1$, so a sum of indices belonging to C must always be considered *mod* $|C|$ (in the second week the staff member associated with the last row progresses to the first one and so on for the following weeks). To simplify the notation we define $C(d)$ as the set of cells corresponding to day d, and $d(c)$ as the day of the week associated with cell c of the roster table.

The main integer variables considered in SRP are binary assignment variables: for each cell $c \in C$ and for each shift $s \in S$

$$x_{cs} = \begin{cases} 1 & \text{if cell } c \text{ is assigned shift } s, \\ 0 & \text{otherwise.} \end{cases}$$

We subdivide the constraints of the problem in three classes: Structural Constraints, Work Agreement Constraints, and Objective Function Related Constraints.

Structural Constraints

Structural constraints are of two types: *assignment constraints* and *work shifts covering constraints.*

Assignment constraints impose that, for each staff member and for each day, exactly one shift has to be assigned. These constraints are formulated in terms of the cells of the roster table as follows:

$$\sum_{s \in S} x_{cs} = 1, \quad \text{for all } c \in C. \tag{1}$$

Work shifts covering constraints simply state that the number of staff members assigned to a given shift must be greater than or equal to the given demand for that shift on that day. To express these constraints, we define b_{sd} as the demand of shift s on day d, denote with $r \in S$ the rest shift, and write:

$$\sum_{c \in C(d)} x_{cs} \geq b_{sd}, \quad \text{for all } d \in D, \ s \in S \setminus \{r\}. \tag{2}$$

Work Agreement Constraints

Work agreement constraints are of two types: *forbidden sequences constraints* and *rest shift constraints.*

Forbidden sequences constraints are used to represent agreement restrictions on consecutive shifts. For example, two consecutive "night" shifts or two consecutive shifts that do not allow for a sufficiently long break may be forbidden.

To express forbidden sequences constraints, we consider the set of forbidden pairs $F \subset S \times S$, such that $(s,t) \in F$ if shift t cannot follow shift s. Thus we have:

$$x_{cs} + x_{c+1,t} \leq 1, \quad \text{for all } (s,t) \in F, \ c \in C. \tag{3}$$

The *rest shift constraints* considered in this paper are based on the following strict rules:

1. *weekly rest shifts*: there must be exactly two rest shifts in each week;

2. *Sunday rest shift*: there must be at least one rest shift on Sunday every month (four weeks).

3. *rest shifts interval*: there may be at most five consecutive days without a rest shift;

To express these constraints, we make use of certain subsets of the indices in C that identify the needed cells. Let $W(p)$ be the set of cell indices corresponding to row p of the roster table. The *weekly rest shift constraints* are then expressed by:

$$\sum_{c \in W(p)} x_{cr} = 2, \quad \text{for all } p \in P. \tag{4}$$

In order to express *Sunday rest shift* constraints, we define $S(p)$ as the set of cells associated with the four Sundays in rows p, $p+1$, $p+2$, $p+3$. Then we have $S(p) = \{7p-1, 7p+6, 7p+13, 7p+20\}$), and write

$$\sum_{c \in S(p)} x_{cr} \geq 1, \quad \text{for all } p \in P. \tag{5}$$

Rest shift interval constraints are expressed by:

$$\sum_{i=0}^{5} x_{c+i,r} \geq 1, \quad \text{for all } c \in C. \tag{6}$$

In sections 4 and 5 we will see that many classes of valid inequalities can be found on the rest shift constraints to strengthen the formulation.

Objective Function and Related Constraints

The objective function of SRP takes into account staff satisfaction, that is measured by the proximity of certain shifts. Its main component is the sequencing of rest shifts: rosters where the two weekly rest shifts are assigned to consecutive days are typically to be preferred. The same applies to other work shifts. Staff satisfaction is thus expressed in the x variables by the following quadratic objective function

$$\max \sum_{c\in C}\sum_{s\in S}\sum_{t\in S} w_{st}x_{cs}x_{c+1,t}$$

that is linearized to apply integer linear programming methods. A straightforward technique is to introduce additional integer variables associated with the product of pairs of x variables: $y_{cst} = 1$ if $x_{cs} = x_{c+1,t} = 1$, and $y_{cst} = 0$ otherwise. The objective function can then be expressed as a linear function on the y variables, that are related to the x variables by the following *objective function related constraints*:

$$\begin{array}{lcr}
y_{cst} \le x_{cs} & & \text{(7a)} \\
y_{cst} \le x_{c+1,t} & \text{for all } c \in C, s,t \in S & \text{(7b)} \\
y_{cst} \ge x_{cs} + x_{c+1,t} - 1 & & \text{(7c)}
\end{array}$$

The Complete IP Formulation of SRP

The complete integer programming formulation of the SRP is summarized below:

$$\max \quad \sum_{c\in C}\sum_{s\in S}\sum_{t\in S} w_{st}y_{cst}$$

$$\begin{array}{lcllr}
\sum_{s\in S} x_{cs} & = & 1 & c \in C & (1) \\
\sum_{c\in C(d)} x_{cs} & \ge & b_{sd} & d \in D,\ s \in S \setminus \{r\} & (2) \\
x_{cs} + x_{c+1,t} & \le & 1 & (s,t) \in F,\ c \in C & (3) \\
\sum_{c\in W(p)} x_{cr} & = & 2 & p \in P & (4) \\
\sum_{c\in S(p)} x_{cr} & \ge & 1 & p \in P & (5) \\
\sum_{i=0}^{5} x_{c+i,r} & \ge & 1 & c \in C & (6) \\
y_{cst} & \le & x_{cs} & c \in C,\ s,t \in S & (7a) \\
y_{cst} & \le & x_{c+1,t} & c \in C,\ s,t \in S & (7b) \\
x_{cs} + x_{c+1,t} - 1 & \le & y_{cst} & c \in C,\ s,t \in S & (7c) \\
x_{cs} \in \{0,1\} & & & c \in C,\ s \in S & \\
y_{cst} \in \{0,1\} & & & c \in C,\ s,t \in S &
\end{array}$$

The introduction of y variables makes it possible to reformulate certain constraints of the model, already expressed using only the x variables, in a more efficient fashion. In fact, combining the definition of the y variables with the assignment constraints (1), we can formulate the objective function related constraints (7) and the forbidden sequences constraints (3) in an alternative way.

Proposition 3.1. *Let $\bar{F}_s$ be the set of shifts that can be scheduled after shift s, and $\tilde{F}_t$ be the set of shifts that can be scheduled before shift t, i.e., $\bar{F}_s = \{t : (s,t) \notin F\}$, $\tilde{F}_t = \{s : (s,t) \notin F\}$. The following equalities are valid for SRP:*

$$\sum_{t \in \bar{F}_s} y_{cst} = x_{cs}, \quad \textit{for all } c \in C, s \in S \tag{8}$$

$$\sum_{s \in \tilde{F}_t} y_{c-1,st} = x_{ct}, \quad \textit{for all } c \in C, t \in S \tag{9}$$

Proof: Consider equality (8):

i) if $x_{cs} = 0$, then, as $y_{cst} \leq x_{cs}$ for constraints (7a), $y_{cst} = 0$ for all $t \in S$;

ii) if $x_{cs} = 1$, a shift $t \in \bar{F}_s$ has to be scheduled in cell $c+1$, then, as $y_{cst} \geq x_{cs} + x_{c+1,t} - 1 = 1$ for constraints (7c), $\sum_{t \in \bar{F}_s} y_{cst} = 1$.

The same argument applies to equality (9). □

Constraints (8) and (9) clearly dominate the objective function related constraints (7). Moreover, constraints (8), (9) and (1) dominate forbidden sequences constraints (3) as, for $(s,t) \in F$,

$$x_{cs} + x_{c+1,t} = x_{cs} + \sum_{s' \in \tilde{F}_t} y_{cs't} \leq x_{cs} + \sum_{s' \in \tilde{F}_t} x_{cs'} \leq \sum_{s' \in S} x_{cs'} = 1,$$

where the first equality is derived by constraints (9), the first inequality is derived by (7a), and the second inequality is valid as $s \notin \tilde{F}_t$; then, the last equality is given by the assignment constraints (1).

4. The Relaxed Rest Shift Subproblem

In this section we present a particular subproblem of SRP, called *Relaxed Rest Shift Subproblem* (RRSS), and derive three classes of facet inducing inequalities of its associated polytope $\mathcal{P}$. These inequalities are valid for SRP as well, and can be applied to strengthen the formulation given in Section 3.

The Relaxed Rest Shift Subproblem is defined by the rest shift constraints (4), (5), (6), and the constraints (7) relating the y variables and the x variables only for the rest shift:

$$\begin{array}{llll}
\sum_{c\in W(p)} x_{cr} & = 2 & p \in P & (4)\\
\sum_{c\in S(p)} x_{cr} & \geq 1 & p \in P & (5)\\
\sum_{i=0}^{5} x_{c+i,r} & \geq 1 & c \in C & (6)\\
y_{crr} & \leq x_{cr} & c \in C & (7'\mathrm{a})\\
y_{crr} & \leq x_{c+1,r} & c \in C & (7'\mathrm{b})\\
x_{cr} + x_{c+1,r} - 1 & \leq y_{crr} & c \in C & (7'\mathrm{c})\\
x_{cr} \in \{0,1\} & & c \in C & \\
y_{crr} \in \{0,1\} & & c \in C. &
\end{array}$$

It is clear from the above definition that RRSS is a relaxation of SRP. As a first result on the polytope $\mathcal{P}$, we find its dimension.

In the proofs of this section we frequently make use of solutions with special structure. An *ad hoc* notation is introduced to represent solutions in a compact and immediate way. As we often need to express if a day is assigned a work or a rest shift, we adopt the following symbols:

- an overlined number in D indicates a rest shift assigned on the corresponding day;
- an underlined number in D indicates a work shift assigned on the corresponding day.

For example, a week with rests only on Monday and Wednesday will be $\overline{1}\underline{2}\overline{3}\underline{4}\underline{5}\underline{6}\underline{7}$.

Proposition 4.1. *The dimension of the polytope $\mathcal{P}$ is equal to* $2|C| - |P|$.

Proof: As the number of variables is $2|C|$ and there are $|P|$ linearly independent equations given by constraints (4), the maximum dimension is $2|C| - |P|$.

Below we show that any equation $\alpha y + \beta x = \alpha_0$ satisfied by all points in $\mathcal{P}$ is a linear combination of constraints (4). We consider two cases, where pairs of solutions (y^1, x^1), $(y^2, x^2) \in \mathcal{P}$ with $y^1 = y^2 = 0$ are used to find relations on the coefficients in β by exploiting the equation $\beta x^1 = \beta x^2 = \alpha_0$.

First case: Given $\bar{p} \in P$, let (y^1, x^1), $(y^2, x^2) \in \mathcal{P}$ such that:

- x^1 has, starting from row $\bar{p}$, the pattern $\overline{1}\underline{2}\underline{3}\underline{4}\underline{5}\underline{6}\overline{7}\ \underline{1}\underline{2}\underline{3}\overline{4}\underline{5}\underline{6}\overline{7} \ldots \underline{1}\underline{2}\underline{3}\overline{4}\underline{5}\underline{6}\overline{7}\ \underline{1}\underline{2}\underline{3}\overline{4}\underline{5}\overline{6}\underline{7}$;
- x^2 has, starting from row $\bar{p}$, the pattern $\underline{1}\overline{2}\underline{3}\underline{4}\underline{5}\underline{6}\overline{7}\ \underline{1}\underline{2}\underline{3}\overline{4}\underline{5}\underline{6}\overline{7} \ldots \underline{1}\underline{2}\underline{3}\overline{4}\underline{5}\underline{6}\overline{7}\ \underline{1}\underline{2}\underline{3}\overline{4}\underline{5}\overline{6}\underline{7}$.

We denote with c the cell associated with Monday ($d(c) = 1$) in row $\bar{p}$. The above two solutions differ only in row $\bar{p}$, where $x^1_{cr} = 1$, $x^1_{c+1,r} = 0$, while $x^2_{cr} = 0$, $x^2_{c+1,r} = 1$. Thus $\beta_c = \beta_{c+1}$. The same argument applies when $d(c) \in \{2, 3, 4\}$; therefore, for each fixed row $\bar{p}$, the coefficients β_c have the same value for $c \in W(\bar{p})$ and $d(c) \in \{1, 2, 3, 4, 5\}$.

Second case: Here we conclude that the coefficients β_c are equal one another for all indices c belonging to a fixed row $\bar{p}$, considering the cells with $d(c) = \mathtt{6}$ or $d(c) = \mathtt{7}$. The two cases are very similar, and we omit for brevity the case $d(c) = \mathtt{7}$. For $d(c) = \mathtt{6}$, let (y^1, x^1), $(y^2, x^2) \in \mathcal{P}$ such that:

- x^1 has, starting from row $\bar{p}$, the pattern $\underline{\mathtt{1}}\,\overline{\mathtt{2}}\,\underline{\mathtt{3}}\,\underline{\mathtt{4}}\,\overline{\mathtt{5}}\,\underline{\mathtt{6}}\,\underline{\mathtt{7}}\;\; \underline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\overline{\mathtt{4}}\,\underline{\mathtt{5}}\,\underline{\mathtt{6}}\,\overline{\mathtt{7}} \ldots \underline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\overline{\mathtt{4}}\,\underline{\mathtt{5}}\,\overline{\mathtt{6}}\,\underline{\mathtt{7}}$;

- x^2 has, starting from row $\bar{p}$, the pattern $\underline{\mathtt{1}}\,\overline{\mathtt{2}}\,\underline{\mathtt{3}}\,\underline{\mathtt{4}}\,\underline{\mathtt{5}}\,\overline{\mathtt{6}}\,\underline{\mathtt{7}}\;\; \underline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\overline{\mathtt{4}}\,\underline{\mathtt{5}}\,\underline{\mathtt{6}}\,\overline{\mathtt{7}} \ldots \underline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\overline{\mathtt{4}}\,\underline{\mathtt{5}}\,\overline{\mathtt{6}}\,\underline{\mathtt{7}}$.

Note that the common rest shift on row $\bar{p}$ has been fixed on Tuesday, so the two solutions only differ on Friday $(c-1)$ and Saturday (c), thus $\beta_{c-1} = \beta_c$.

As all coefficients β_c are equal one another in a given row, they can be obtained as linear combinations of the left hand sides of equations (4). Then, if we subtract that combination to $\alpha y + \beta x = \alpha_0$, we obtain an equivalent equation $\alpha' y + \beta' x = \alpha'_0$ with $\alpha' = \alpha$ and $\beta' = 0$.

If we consider one of the four solutions used above, which have $y^1 = 0$ or $y^2 = 0$, we obtain that $\alpha'_0 = \alpha' y = 0$. Moreover, for each pair of consecutive cells it is easy to define a solution in $\mathcal{P}$ with a unique sequence of two consecutive rest shifts. For instance, choose $\bar{p} \in P$, $c \in W(\bar{p})$ such that $d(c) = \mathtt{2}$, and define a solution (y, x) with the following pattern (starting from row $\bar{p}$):

$$\underline{\mathtt{1}}\,\overline{\mathtt{2}}\,\overline{\mathtt{3}}\,\underline{\mathtt{4}}\,\underline{\mathtt{5}}\,\underline{\mathtt{6}}\,\underline{\mathtt{7}}\;\; \overline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\underline{\mathtt{4}}\,\underline{\mathtt{5}}\,\underline{\mathtt{6}}\,\overline{\mathtt{7}}\;\; \underline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\overline{\mathtt{4}}\,\underline{\mathtt{5}}\,\underline{\mathtt{6}}\,\overline{\mathtt{7}} \;\ldots\; \underline{\mathtt{1}}\,\underline{\mathtt{2}}\,\underline{\mathtt{3}}\,\overline{\mathtt{4}}\,\underline{\mathtt{5}}\,\overline{\mathtt{6}}\,\underline{\mathtt{7}};$$

As a consequence we derive that $\alpha'_c = \alpha' y = \alpha'_0 = 0$ for all $c \in C$. This completes the proof and, as any equation $\alpha y + \beta x = \alpha_0$ satisfied by all points in $\mathcal{P}$ is a linear combination of equations (4), then $dim(\mathcal{P}) = 2|C| - |P|$. □

We now describe three classes of valid inequalities and prove that they induce facets for $\mathcal{P}$, thus giving an insight of their theoretical effectiveness in strengthening the formulation of SRP. The proofs are based on the indirect method (see [13], Section 9.2.3), and differ only on few technical details.

Forbidding three consecutive rest shifts

Due to weekly rest shift constraints (4), it is not possible to assign three consecutive rest shifts in a week. We use this condition to strengthen the formulation with the following inequalities:

$$y_{crr} + y_{c+1,rr} \leq x_{c+1,r} \quad \text{for all } c \in C \text{ such that } d(c) \notin \{\mathtt{6}, \mathtt{7}\}. \tag{10}$$

Proposition 4.2. *The inequalities (10) are valid for the Relaxed Rest Shift Subproblem.*

Proof: The left hand side (hereafter, *lhs*) of (10) may be equal to 0, 1, or 2. If it is 0, then (10) is trivially satisfied. If it is 1, then either $y_{crr} = 1$ or $y_{c+1,rr} = 1$. In both cases we have $x_{c+1,r} = 1$ for constraints (7), so (10) is valid. If the *lhs* is 2, then there are three consecutive rest shifts and this is only possible starting from day $\mathtt{6}$ or day $\mathtt{7}$ (i.e., considering rest shifts of two distinct weeks). □

Theorem 4.3. *The inequalities (10) induce facets of $\mathcal{P}$.*

Proof: Assume $\alpha y + \beta x \leq \alpha_0$ to be an inequality inducing a facet of $\mathcal{P}$ that contains all the tight solutions for an inequality $y_{crr} + y_{c+1,rr} \leq x_{c+1,r}$ of type (10). We now prove the equivalence between $\alpha y + \beta x \leq \alpha_0$ and $y_{crr} + y_{c+1,rr} \leq x_{c+1,r}$ finding the values of the coefficients α, β, and α_0 of the first inequality using tight solutions for inequality (10).

First, note that weekly rest shift constraints (4) are equations for RRSS defined on disjoint subsets of the x variables. As all inequalities obtained adding a multiple of an equation are equivalent, we can suppose, without loss of generality, that the coefficients $\beta_{\bar{c}(p)} = 0$, where $\bar{c}(p)$ is a cell in $W(p)$ different from $c+1$, for each $p \in P$.

Second, let c_1, $c_2 \neq c+1$ be two cells; one can find two feasible solutions (y^1, x^1), (y^2, x^2) for RRSS such that the following conditions hold:

- $x^1_{c_1 r} = 1$, $x^1_{c_2 r} = 0$, and $x^2_{c_1 r} = 0$, $x^2_{c_2 r} = 1$;
- $y^i_{crr} = y^i_{c+1,rr} = x^i_{c+1,r} = 0$ for $i \in \{1,2\}$ and (y^1, x^1), (y^2, x^2) are equal for other indices.

As they are both tight for (10), then $\alpha y^1 + \beta x^1 = \alpha y^2 + \beta x^2 = \alpha_0$, and therefore $\beta_{c_1} = \beta_{c_2}$. Since c_1 or c_2 can be chosen equal to one of the indices $\bar{c}(p)$, then $\beta_{c_1} = \beta_{c_2} = 0$ for all c_1, $c_2 \neq c+1$. The solutions (y^1, x^1) and (y^2, x^2) can be chosen without sequences of consecutive rest shifts, that is, with all y variables equal to zero. So we have $\beta x^1 = \alpha_0$, and, as the coefficients in vector β are equal to zero for all cells different from $c+1$ and $x_{c+1,r} = 0$, we have $\alpha_0 = 0$.

Third, let $c' \neq c$, $c+1$, then there exists a tight feasible solution $(\bar{y}, \bar{x})$ such that $\bar{x}_{c+1,r} = 0$ and $\bar{y}_{c'rr} = 1$ is the unique y variable with value 1. Then $\alpha\bar{y} + \beta\bar{x} = \alpha_{c'} = \alpha_0 = 0$.

Finally, suppose that cell $c' = c$, i.e., $\bar{y}_{crr} = 1$ and $\bar{x}_{cr} = \bar{x}_{c+1,r} = 1$. Then $\alpha\bar{y} + \beta\bar{x} = \alpha_c + \beta_{c+1} = \alpha_0 = 0$, that is $\alpha_c = -\beta_{c+1} = \mu$. Similarly, considering $c' = c+1$, we get $\alpha_{c+1} = -\beta_{c+1} = \mu$. We have thus shown that the facet inducing inequality $\alpha y + \beta x \leq \alpha_0$ is equivalent to $\mu y_{crr} + \mu y_{c+1,rr} - \mu x_{c+1,r} \leq 0$, i.e., to an inequality of type (10). □

Strengthening weekly rest shift constraints

Here we describe a class of inequalities that considers the 7 cells of the roster table associated with a week and it is derived again using the weekly rest shift constraints (4).

Proposition 4.4. *For all $c \in C$ such that $d(c) = 1$ the inequalities*

$$x_{cr} + x_{c+1,r} + y_{c+2,rr} + y_{c+3,rr} + y_{c+4,rr} + y_{c+5,rr} \leq 1 + y_{crr} \tag{11}$$

are valid for the Relaxed Rest Shift Subproblem.

Proof: To show validity of (11), we note that it is obtained as a reinforcement of the following inequality:

$$y_{c+2,rr} + y_{c+3,rr} + y_{c+4,rr} + y_{c+5,rr} \leq 1. \tag{12}$$

Constraints (4) state that there are exactly two rest shifts in the same week. This trivially implies (12), as a value greater than 1 in its *lhs* would require at least 3 rest shifts in the same week.

Then we prove the proposition considering the following three cases that cover all the possible solutions:

i) $x_{cr} + x_{c+1,r} = 0$, that implies $y_{crr} = 0$. In this case, (11) reduces to (12);

ii) $x_{cr} + x_{c+1,r} = 1$; in this case the y variables in the *lhs* of (12) must have all value 0 because of (4), and thus (11) is satisfied;

iii) $x_{cr} + x_{c+1,r} = 2$. Also in this case the y variables in the *lhs* of (12) must have all value 0 because of (4); moreover, we have that $y_{crr} = 1$. Then the *rhs* of (11) takes value 2.

□

Theorem 4.5. *The inequalities (11) are facet inducing for the Relaxed Rest Shift Subproblem.*

Proof: As in the proof of Theorem 4.3 we consider an inequality $\alpha\ y\ +\ \beta\ x\ \leq\ \alpha_0$ inducing a facet of $\mathcal{P}$ that contains all tight solutions of inequality (11); moreover, we choose indices $\bar{c}(p)$ different from c and $c+1$, so the coefficients $\beta_{\bar{c}(p)}$ can again be assumed to be zero. Now we find the coefficients in all the other cases.

First, let $c_1, c_2 \notin \{c, c+1\}$ and consider two solutions for RRSS such that

- $x^1_{c_1r} = 1$, $x^1_{c_2r} = 0$, and $x^2_{c_1r} = 0$, $x^2_{c_2r} = 1$;
- $x^i_{cr} = 1$, $x^i_{c+1,r} = 0$ for $i \in \{1, 2\}$, and (y^1, x^1), (y^2, x^2) are equal for all other indices.

As they are both tight for (11), then $\alpha y^1 + \beta x^1 = \alpha y^2 + \beta x^2 = \alpha_0$, and therefore $\beta_{c_1} = \beta_{c_2}$. As in Theorem 4.3, c_1 or c_2 may be chosen equal to one of the indices $\bar{c}(p)$, so $\beta_{c_1} = \beta_{c_2} = 0$.

Moreover, the solutions (y^1, x^1) and (y^2, x^2) can be chosen with $y^1 = y^2 = 0$, so that $\beta x^1 = \alpha_0$. As all the coefficients in vector β are equal to zero for the cells different from c and $c+1$, $x_{cr} = 1$, and $x_{c+1,r} = 0$, we also have $\beta_c = \alpha_0 = \mu$. The same applies if (y^1, x^1) is chosen such that $x_{cr} = 0$ and $x_{c+1,r} = 1$, deriving that $\beta_{c+1} = \alpha_0 = \mu$.

Second, let $c' \notin \{c,\ c+1,\ \ldots,\ c+5\}$; we can choose a tight solution with $y_{c'rr} = 1$, $x_{cr} = 1$, and $x_{c+1,r} = 0$, and all other y variables equal to zero; thus $\alpha_{c'} + \beta_c = \alpha_0$, i.e., $\alpha_{c'} = 0$.

Third, let $c' = c+1$; we can choose a tight solution with $y_{c+1,rr} = 1$ and all other y variables equal to zero. Then $x_{c+1,r} = x_{c+2,r} = 1$, and thus $\alpha_{c+1} + \beta_{c+1} = \alpha_0$, i.e., $\alpha_{c+1} = 0$ since $\beta_{c+1} = \alpha_0 = \mu$.

Fourth, let $c' \in \{c+2,\ \ldots,\ c+5\}$; we can choose a tight solution with $y_{c'rr} = 1$, and thus, as the other variables in the considered inequality must be equal to zero and we have determined that the other coefficients are equal to zero, we obtain $\alpha_{c'} = \alpha_0 = \mu$.

Finally, consider a tight solution with $y_{crr} = 1$ and $x_{cr} = x_{c+1,r} = 1$, then $\alpha y + \beta x = \alpha_c + \beta_c + \beta_{c+1} = \alpha_0$. As we have seen that $\beta_c = \beta_{c+1} = \alpha_0 = \mu$, then $\alpha_c + 2\mu = \mu$, i.e. $\alpha_c = -\mu$, and therefore we have shown that the facet inducing inequality $\alpha y + \beta x \leq \alpha_0$ is proportional to $x_{cr} + x_{c+1,rr} + y_{c+2,rr} + y_{c+3,r} + x_{c+4,r} + y_{c+5,rr} \leq 1 + y_{crr}$. □

This class of inequalities can be easily generalized shifting the pair of x variables in (11) to days of the week different from Monday and Tuesday. We can summarize this class of valid inequalities in the following list:

$$\begin{array}{llllllll}
x_{cr} & + x_{c+1,r} & + y_{c+2,rr} & + y_{c+3,rr} & + y_{c+4,rr} & + y_{c+5,rr} & & \leq 1 + y_{crr} \\
 & x_{c+1,r} & + x_{c+2,r} & + y_{c+3,rr} & + y_{c+4,rr} & + y_{c+5,rr} & & \leq 1 + y_{c+1,rr} \\
y_{crr} & & + x_{c+2,r} & + x_{c+3,r} & + y_{c+4,rr} & + y_{c+5,rr} & & \leq 1 + y_{c+2,rr} \\
y_{crr} & + y_{c+1,rr} & & + x_{c+3,r} & + x_{c+4,r} & + y_{c+5,rr} & & \leq 1 + y_{c+3,rr} \\
y_{crr} & + y_{c+1,rr} & + y_{c+2,rr} & & + x_{c+4,r} & + x_{c+5,r} & & \leq 1 + y_{c+4,rr} \\
y_{crr} & + y_{c+1,rr} & + y_{c+2,rr} & + y_{c+3,rr} & & + x_{c+5,rr} & + x_{c+6,rr} & \leq 1 + y_{c+5,rr}.
\end{array} \tag{13}$$

Moreover this class can be completed with other two inequalities for each week regarding the beginning and the end of the week:

$$\begin{array}{l} x_{cr} + y_{c+1,rr} + y_{c+2,rr} + y_{c+3,rr} + y_{c+4,rr} + y_{c+5,rr} \leq 1 \\ y_{crr} + y_{c+1,rr} + y_{c+2,rr} + y_{c+3,rr} + y_{c+4,rr} + x_{c+6,r} \leq 1 \end{array} \qquad (14)$$

The proof of Proposition 4.4 and of Theorem 4.5 for the other inequalities in this class is very similar. In particular, in the proof of Theorem 4.5, one only has to pay attention to the coefficient of the y variable preceding the pair of x variables, applying the third step also in a backward sense. After the third step of the proof, it is easy to select a tight solution showing that this coefficient is equal to α_0 minus the coefficient of the first of the two x variables, which are equal.

Inequalities on the sequence Sunday-Monday

The sequence of two consecutive rest shifts on Sunday and Monday ($\overline{7}\,\overline{1}$) takes into account rest shifts of two weeks and can be used to strengthen other inequalities.

If a sequence $\overline{6}\,\overline{7}$ is chosen in a week, then a $\overline{7}$ must be chosen in the previous week, because:

i) two rest shifts have been chosen in that week, so from Monday to Friday five work shifts must be assigned;

ii) for constraints (6), at most five consecutive work shifts can be set.

Therefore the following inequalities are valid

$$y_{crr} \leq x_{c-6,r} \quad \text{for all } c \in C \text{ such that } d(c) = 6.$$

Considering two consecutive weeks, the sequences $\overline{7}\,\overline{1}$ at their join and $\overline{6}\,\overline{7}$ on the second week are incompatible, because there would be three rest shifts in the second week. Both sequences require a rest shift on Sunday in the first week, then the following inequalities are valid and strengthen the previous ones

$$y_{c-6,rr} + y_{crr} \leq x_{c-6,r} \quad \text{for all } c \in C \text{ such that } d(c) = 6. \qquad (15)$$

A similar argument can be applied if one chooses a sequence $\overline{1}\,\overline{2}$: in the following week there must be a rest shift on Monday, so the following inequalities are valid

$$y_{crr} \leq x_{c+7,r} \quad \text{for all } c \in C \text{ such that } d(c) = 1.$$

These relations can be strengthened using the sequence $\overline{7}\,\overline{1}$ at the join of the two weeks:

$$y_{crr} + y_{c+6,rr} \leq x_{c+7,r} \quad \text{for all } c \in C \text{ such that } d(c) = 1. \qquad (16)$$

Theorem 4.6. *The inequalities (15) and (16) are facet inducing for RRSS.*

Proof: We prove the theorem for an inequality of type (15)

$$y_{c-6,rr} + y_{crr} \leq x_{c-6,r}.$$

We recall that c corresponds to a Saturday and $c-6$ to the previous Sunday. We again consider a facet inducing inequality $\alpha y + \beta x \leq \alpha_0$ containing all tight solutions for (15), choose for each week p a cell $\bar{c}(p)$ different from $c-6$, and fix the coefficients $\beta_{\bar{c}(p)} = 0$.

First, let c_1, $c_2 \neq c-6$ be two cells and (y^1, x^1), (y^2, x^2) two tight solutions for (15) such that

- $x^1_{c_1r} = 1$, $x^1_{c_2r} = 0$, and $x^2_{c_1r} = 0$, $x^2_{c_2r} = 1$;
- $y^i_{c-6,rr} = y^i_{crr} = x^i_{c-6,r} = 0$ for $i \in \{1, 2\}$, and (y^1, x^1), (y^2, x^2) are equal for other indices.

Then $\alpha y^1 + \beta x^1 = \alpha y^2 + \beta x^2 = \alpha_0$, so $\beta_{c_1} = \beta_{c_2}$ and, as c_1 and c_2 can be chosen equal to $\bar{c}(p)$ for a $p \in P$, it follows $\beta_{c_1} = \beta_{c_2} = 0$. Moreover, in the previous solutions it is possible that $y^1 = y^2 = 0$, and therefore $\alpha_0 = \beta x^1 = 0$.

Second, let $c' \neq c$, $c - 6$; we can select a tight solution with $y_{c'rr} = 1$, $x_{c-6,r} = 0$, and all other y variables equal to zero, and then have $\alpha_{c'} = \alpha_0 = 0$.

Finally, consider a solution with $y_{c-6,rr} = x_{c-6,r} = 1$, then $\alpha_{c-6} + \beta_{c-6} = \alpha_0 = 0$, i.e., $\alpha_{c-6} = -\beta_{c-6}$. In a similar way we obtain $\alpha_c = -\beta_{c-6}$ (it is sufficient to consider a solution with $y_{crr} = x_{c-6,r} = 1$). Therefore, we have shown that the facet inducing inequality $\alpha y + \beta x \leq \alpha_0$ is equivalent to inequality (15). The proof for inequalities of type (16) is similar. □

5. Other classes of valid inequalities

Other classes of inequalities are derived considering different aspects of the problem such as:

- work shifts covering constraints;
- work shifts sequences;
- strong relations between rest shifts chosen in consecutive weeks.

While the following description is very general, in Section 8, discussing experimental results, we will make some considerations on the effectiveness of each class of inequalities.

Some sequences of rest shifts fixed for a week imply a particular rest shift for the next or the previous week. Because of Sunday rest shift constraints (5), we can deduce valid inequalities bounding the number of such sequences of rest shifts.

The number of rest shifts on each day is limited by work shifts covering constraints (2): the minimum number of staff members needed to cover the work shifts on day d is $\sum_{s \in S \setminus \{r\}} b_{sd}$. We define the maximum number of rest shifts that can be assigned on day d as:

$$MR(d) = |P| - \sum_{s \in S \setminus \{r\}} b_{sd}.$$

Saturday-Sunday sequences

If a sequence of rest shifts on Saturday and Sunday ($\overline{6}\,\overline{7}$) is chosen for a week, then in the previous week one must choose a rest shift on Sunday, otherwise a staff member would work more than five consecutive days (constraints (6)). Obviously, choosing too many sequences $\overline{6}\,\overline{7}$ would exhaust the rest shifts on Sundays, and then Sunday rest shift constraints (5) could not be satisfied for all staff members. The following inequality expresses the maximum number of sequences $\overline{6}\,\overline{7}$ that can be assigned:

$$\sum_{d(c)=6} y_{crr} \leq \left\lfloor \frac{4MR(7) - |P|}{3} \right\rfloor . \tag{17}$$

Proposition 5.1. *The inequality (17) is valid for SRP.*

Proof: As already mentioned, if we choose a sequence $\overline{6}\,\overline{7}$ in a week, we have to choose also a $\overline{7}$ in the previous week. Therefore we have satisfied constraints (5) for five rows of the roster table: two rows have a rest shift on Sunday and the previous three have a rest shift on Sunday in one of the following three weeks; for instance see the rows from 2 to 6 in Figure 1.

In general, if we have a sequence composed of $\overline{6}\,\overline{7}$ in i consecutive weeks, we use $i+1$ rest shifts on Sunday and satisfy constraints (5) for $i+4$ rows. Let k_i be the number of sequences composed of $\overline{6}\,\overline{7}$ in i consecutive weeks, then

- $MR(7) - \sum_{i \geq 1}(i+1)k_i$ is the residual number of rest shifts on Sundays;
- $|P| - \sum_{i \geq 1}(i+4)k_i$ is the number of staff members for which constraints (5) are not satisfied by the sequences $\overline{6}\,\overline{7}$.

The number of $\overline{7}$ still to be assigned is at least $(|P| - \sum_{i \geq 1} (i+4)\, k_i)/4$, as constraints (5) require at least one $\overline{7}$ every four weeks, and thus

$$MR(7) - \sum_{i \geq 1}(i+1)k_i \geq \frac{|P| - \sum_{i \geq 1}(i+4)k_i}{4}.$$

With simple computations we derive that

$$3 \sum_{i \geq 1} i\ k_i \leq 4MR(7) - |P|.$$

The total number of sequences $\overline{6}\,\overline{7}$ chosen is given by $\sum_{d(c)=6} y_{crr} = \sum_{i \geq 1} i\ k_i$, therefore

$$3 \sum_{d(c)=6} y_{crr} \leq 4MR(7) - |P|.$$

Now dividing by 3 and rounding the *rhs*, we obtain inequality (17). □

Monday-Tuesday sequences

When a sequence $\overline{1}\,\overline{2}$ is fixed for a week, then in the next week a $\overline{1}$ must be chosen due to constraints (4) and (6). In addition we have to satisfy constraints (5) on Sundays. Next inequality expresses the maximum number of sequences $\overline{1}\,\overline{2}$ that can be chosen.

Proposition 5.2. *The following inequality is valid for SRP:*

$$\sum_{d(c)=1} y_{crr} \leq \left\lfloor \frac{3MR(1)}{4} \right\rfloor. \tag{18}$$

Proof: Due to Sunday rest shift constraints (5), we can choose at most three consecutive sequences $\overline{1}\,\overline{2}$: the fourth week must be $\overline{1}\,\underline{2}\,\underline{3}\,\underline{4}\,\underline{5}\,\underline{6}\,\overline{7}$ because of constraints (6). Thus, every three sequences $\overline{1}\,\overline{2}$ we have four rest shifts on Monday and the proposition follows. □

Global bounds on Sunday-Monday sequences

The sequence of rest shifts on Sunday-Monday ($\overline{7}\,\overline{1}$) is very critical, as it considers rest shifts of two weeks and makes it possible to have sequences of more than two days of consecutive rest shifts.

If we consider three consecutive days inside a week, for instance Tuesday, Wednesday, and Thursday, it is obvious that the number of sequences $\overline{2}\,\overline{3}$ plus the number of sequences $\overline{3}\,\overline{4}$ is limited by the maximum number of rest shifts that can be assigned on Wednesday (i.e., $MR(3)$), as a rest shift on Wednesday cannot be used in both sequences, because that means assigning three rest shifts in a week.

So the inequality

$$\sum_{d(c)=2} y_{crr} + \sum_{d(c)=3} y_{crr} \leq MR(3),$$

is valid; note that the above inequality is dominated by the inequalities (10) forbidding three consecutive rest shifts.

The same argument cannot be applied to the sequences $\overline{6}\,\overline{7}\,\overline{1}$ or $\overline{7}\,\overline{1}\,\overline{2}$, as they consider rest shifts belonging to two different weeks. Nevertheless we will show that we can derive validity for inequalities of the same type of the one discussed above for the case of Tuesday-Wednesday-Thursday.

Theorem 5.3. *The following inequalities are valid for SRP:*

$$\sum_{d(c)=7} y_{crr} + \sum_{d(c)=1} y_{crr} \leq MR(1) \tag{19}$$

$$\sum_{d(c)=6} y_{crr} + \sum_{d(c)=7} y_{crr} \leq MR(7) \tag{20}$$

.

Proof: We prove validity of inequality (19). Validity of inequality (20) can be proved in the same way.

Let k_i be the number of sequences $\overline{1}\,\overline{2}$ repeated for $i \geq 1$ consecutive weeks. Note that for any of such sequences the number of $\overline{1}$ to be set is at least $i+1$, as after a sequence $\overline{1}\,\overline{2}$ there must be again $\overline{1}$.

A sequence $\overline{7}\,\overline{1}$ can be of two types: $\overline{7}\,\overline{1}\,\overline{2}$ or $\overline{7}\,\overline{1}\,\underline{2}$. The number of sequences $\overline{7}\,\overline{1}\,\overline{2}$ is less than or equal to the number of groups of sequences $\overline{1}\,\overline{2}$ in consecutive weeks $\sum_{i\geq 1} k_i$, while the number of sequences $\overline{7}\,\overline{1}\,\underline{2}$ is less than or equal to the number of rest shifts on Monday still available, i.e., $MR(1) - \sum_{i\geq 1}(i+1)k_i$. So we have that

$$\sum_{d(c)=7} y_{crr} \leq (\sum_{i\geq 1} k_i) + (MR(1) - \sum_{i\geq 1}(i+1)k_i) = MR(1) - \sum_{i\geq 1} ik_i.$$

From the definition of k_i is easy to see that the number of sequences of $\overline{1}\,\overline{2}$ is exactly $\sum_{i\geq 1} ik_i$, so we derive the inequality (19):

$$\sum_{d(c)=7} y_{crr} \leq MR(1) - \sum_{d(c)=1} y_{crr}.$$

□

Weekly relations on Sunday-Monday sequences

The inequalities stated in the following proposition derive straightforwardly from weekly rest shift constraints (4), which limit the number of rest shifts to exactly two for each week.

Proposition 5.4. *The following inequalities are valid for SRP:*

$$\sum_{h=0}^{4} y_{c+h,rr} + y_{c+6,rr} \leq 1 \qquad \textit{for all } c \in C \textit{ such that } d(c) = 1, \tag{21}$$

$$y_{crr} + \sum_{h=2}^{6} y_{c+h,rr} \leq 1 \qquad \textit{for all } c \in C \textit{ such that } d(c) = 7. \tag{22}$$

Proof: It is easy to see that if one the above inequalities is violated by an integer solution, then there are at least three rest shifts in a week. □

Minimum requirement on rest shifts

Let $mr(d)$ be the minimum number of rest shifts required on day d; this value depends both on covering constraints for work shifts and on the constraints for rest shifts. For instance $mr(7) \geq \lceil |P|/4 \rceil$.

Another lower bound on $mr(d)$ valid for all days of the week can be computed as follows. Note that the total number of shifts to assign is $7|P|$, the minimum number of work shifts to assign is $\sum_{s \in S \setminus \{r\}} \sum_{d \in D} b_{sd}$, the total number of rest shifts to assign is $2|P|$. Then

$$FW = 7|P| - \sum_{s \in S \setminus \{r\}} \sum_{d \in D} b_{sd} - 2|P|$$

is the number of work shifts that can be assigned freely. So the number of rest shifts that has to be assigned on a given day d_0 is at least $MR(d_0) - FW$. Indeed,

$$MR(d_0) - FW = 2|P| - \left(6|P| - \sum_{s \in S \setminus \{r\}} \sum_{d \in D \setminus \{d_0\}} b_{sd} \right),$$

where $2|P|$ is the total number of rest shifts to be assigned, and the term in parentheses is the maximum number of rest shifts that can be assigned on days different from d_0.

Therefore we consider $mr(d) = \max\{MR(d) - FW, 0\}$ for each day $d \neq 7$ and $mr(7) = \max\{MR(7) - FW, \lceil |P|/4 \rceil\}$.

If the $mr(d)$ rest shifts are not covered by consecutive rest shifts including day d, then another day of the week must be coupled with d to satisfy weekly rest shift constraints (4), thus limiting the number of consecutive rest shifts including the other days of the week.

We present two different cases and derive the corresponding inequalities. To simplify the exposition, we define new variables equal to the sum of sets of y variables: let $z_d = \sum_{c \in C(d)} y_{crr}$ be the sum of the sequence variables associated with the pair of days d and $d+1$.

Case 1: Sunday

The following inequality is clearly valid:

$$(MR(1) - z_1) + (MR(2) - z_1 - z_2) + (MR(3) - z_2 - z_3) +$$

$$+(MR(4) - z_3 - z_4) + (MR(5) - z_4 - z_5) \geq mr(7) - z_6.$$

Each term in parenthesis in the *lhs* is the maximum number of rest shifts that are not assigned to consecutive days and can be coupled with the rest shifts on Sunday not coupled with Saturday in the *rhs*. Now dividing the inequality by 2 and rounding we get,

$$z_1 + z_2 + z_3 + z_4 \leq \left\lfloor \frac{\sum_{1 \leq d \leq 5} MR(d) - mr(7)}{2} \right\rfloor + z_6.$$

Case 2: Sunday and Saturday together

Similarly the following inequality is valid:

$$(MR(1) - z_1) + (MR(2) - z_1 - z_2) + (MR(3) - z_2 - z_3) +$$

$$+(MR(4) - z_3 - z_4) + (MR(5) - z_4 - z_5) \geq mr(7) + mr(6) - z_6 - z_5.$$

And again dividing by 2 and rounding we get,

$$z_1 + z_2 + z_3 + z_4 \leq \left\lfloor \frac{\sum_{1 \leq d \leq 5} MR(d) - mr(7) - mr(6)}{2} \right\rfloor + z_6. \tag{23}$$

Finally, substituting $z_d = \sum_{c \in C(d)} y_{crr}$ we derive inequalities with the original variables.

Forbidding six consecutive work shifts

We present a class of inequalities based on work shifts that is very similar to the inequalities (10) forbidding three consecutive rest shifts.

While inequalities (10) are based on constraints (4) (two rest shifts every week) and variables on rest shifts, this new class considers constraints (6) (at most five consecutive working days) and variables on work shifts.

Proposition 5.5. *Given any initial cell c and six work shifts s_0, s_1, s_2, s_3, s_4, and s_5 (not necessarily different), the following inequality is valid for SRP:*

$$\sum_{h=0}^{4} y_{c+h, s_h s_{h+1}} \leq \sum_{h=1}^{4} x_{c+h, s_h} \tag{24}$$

Proof: The *lhs* of inequality (24) can be a value from 0 to 4. Actually, if all the y variables in the *lhs* have value 1 then we have six consecutive working days violating constraints (6). If the *lhs* is 0 the inequality is trivially satisfied. In the other cases, due to constraints (7) one can easily verify that the number of x variables equal to 1 in the *rhs* must be at least equal to the number of y variables equal to 1 in the *lhs*. □

6. Inequalities Based on SRP Subproblems

We identify some subsets of the SRP variables with a relevant role in the objective function and in the structure of the polyhedron (e.g., y variables associated with sequences of rest shifts). It is possible to define certain subproblems of SRP on a subset of shifts T and on a subset of shift sequences $L \subset T \times T$, whose optimal solution provide the right hand side of a rank inequality for the related variables. These inequalities turn to be very effective in the solution algorithm for the complete problem.

In real instances of SRP, rest shift sequences are assigned a weight w_{rr} that is significantly larger than weights associated with other shifts sequences; this fact derives from the translation of typical staff preferences into weights for the objective function. Thus we can solve SRP in two steps:

i) first solve the subproblem obtained considering only rest shifts and using all the related cuts;

ii) then solve the complete problem using the rank inequality coming from the above subproblem and all the other cuts.

We call the subproblem of point (i) the *Rest Shift Subproblem.* We define it adding new constraints to the Relaxed Rest Shift Subproblem (described in Section 4) that for each day of the week leave a number of staff members sufficient to cover the work shifts of that day. Thus, recalling from Section 5 that $MR(d) = |P| - \sum_{s \in S \setminus \{r\}} b_{sd}$, we write the following constraints:

$$\sum_{c \in C(d)} x_{cr} \leq MR(d) \quad \text{for all } d \in D.$$

In the objective function, we now maximize the sequences of rest shifts:

$$\max \sum_{c \in C} y_{crr},$$

where all weights w_{rr} are replaced by 1.

Therefore, the complete formulation of the Rest Shift Subproblem is the following:

$$\begin{array}{llll}
\mu = \max & \sum_{c \in C} y_{crr} & & \\
& \sum_{c \in C(d)} x_{cr} & \leq \; MR(d) & d \in D \\
& \sum_{c \in W(p)} x_{cr} & = \; 2 & p \in P \\
& \sum_{c \in S(p)} x_{cr} & \geq \; 1 & p \in P \\
& \sum_{i=0}^{5} x_{c+i,r} & \geq \; 1 & c \in C \\
& y_{crr} & \leq \; x_{cr} & c \in C \\
& y_{crr} & \leq \; x_{c+1,r} & c \in C \\
& x_{cr} + x_{c+1,r} - 1 & \leq \; y_{crr} & c \in C \\
& x_{cr} \in \{0,1\} & & c \in C \\
& y_{crr} \in \{0,1\} & & c \in C.
\end{array}$$

Proposition 6.1. *The following inequality is valid for SRP:*

$$\sum_{c \in C} y_{crr} \leq \mu. \tag{25}$$

Proof: It is easy to verify that the projection of any feasible solution of SRP on the space of the Rest Shift Subproblem is feasible for the latter problem as well. Thus, inequality (25) is valid for SRP. □

Although the Rest Shift Subproblem is difficult in general, its size is significantly smaller than the complete SRP. Moreover, many of the inequalities discussed in sections 4 and 5 are still valid for the Rest Shift Subproblem and contribute very effectively to determine its optimal solution in a Branch & Bound algorithm.

7. Branching rules

The SRP feasible solutions have a high degree of symmetry that on the one hand makes the problem difficult, as there are several different optimal solutions that must be visited by a myopic Branch & Bound algorithm to prove optimality, and on the other hand may be exploited to design a clever branching strategy. Breaking these symmetries is crucial for solving SRP to optimality.

We identify two types of symmetry,

- the *cyclic symmetry:* for each roster table there are $|P| - 1$ roster tables with the same objective function value which can be obtained by rotation of the table on the rows;
- the *Sunday symmetry:* in a roster table representing a feasible solution of SRP, consider the sequences of rows with the following structure: the first row follows a week with a rest on Sunday, and the last row ends with a rest on Sunday (a sequence can be composed of a single row). Any pair of such sequences can be swapped in the table without affecting the value of the objective function and the feasibility of the solution.

For an example of Sunday symmetry, consider the roster table in Figure 1. The sequence composed of rows 2, 3, 4, 5, and the one composed of rows 13 and 1 have the above described structure, and can thus be swapped resulting in another solution with the same number of rest sequences.

Another important point is related to the number of sequences $\bar{6}\,\bar{7}$ that can be assigned. We have seen that this number is limited by inequality (17) and in some cases reducing the *rhs* of the mentioned inequality may result in a strong reduction of the objective function allowing us to establish that inequality (17) is tight in the optimal integer solution.

We thus design two main branching strategies:

- the *Saturday-Sunday rule;*
- the *hole rule.*

Let $r = \lfloor (4MR(7) - |P|)/3 \rfloor$; the *Saturday-Sunday rule*, applied as the first branching rule, produces two branches: in the first branch we modify inequality (17) as equality

$$\sum_{d(c)=6} y_{crr} = r,$$

and in the second branch we reduce by one unit the *rhs*

$$\sum_{d(c)=6} y_{crr} \leq r - 1.$$

The *hole rule* tries to overcome the cyclic and the Sunday symmetries by considering possibly only one representative solution in each equivalence class. To describe the structure of the representative solutions we use sequences of rows where only the first row and the last row have Sunday rest. We then identify with h_i the number of such sequences with i rows without Sunday rest. Thus, h_0 is the number of consecutive weeks with Sunday rest, h_1 is the number of sequences of three rows where only the first and the last have Sunday rest, and so on. From constraint (5) it is easy to deduce that $h_i = 0$ when $i \geq 4$.

From the Sunday symmetry described above, it is clear that any swapping of the sequences above described results in a solution with the same objective value. Thus each configuration with the same values of h_0, h_1, h_2, h_3 belongs to the same equivalence class and it is sufficient to visit only one representative of each class in the branching tree.

To implement the hole rule, we fix the first Sunday of the roster table to be a Sunday rest (use of the cyclic symmetry); then we enumerate all the feasible configurations of h_0, h_1, h_2, h_3, define a subtree associated with each configuration by fixing Sunday rest variables in the problem, and solve the subtrees.

Using the cuts described in sections 4, 5, and the branching rules just exposed we have designed a Branch & Bound algorithm to solve both the Rest Shift Subproblem and the complete SRP.

8. Computational Results

The following tables report a synthesis of the results obtained on eight different SRP instances with dimensions ranging from 30 to 100 staff members. The data has been provided by the Alitalia ground staff management department.

The presented test problems are divided into two sets (set A and set B) differing only on the weight w_{st} of the objective function for $(s,t) \neq (r,r)$. In particular, the problems in set B have weights w_{st} different from zero only on pairs (s,t) with $s = t$, while the problems in set A have a more complex objective function with weights different from zero also on other pairs of shifts. Obviously, the dimensions of the two instances with the same number of staff members are the same and the same are the results on the Rest Shift Subproblems, where the weight coefficients for $(s,t) \neq (r,r)$ are not taken into account. These figures are then reported only once in the tables below.

Table 1 shows the number of integer variables on the eight problems and the optimal solutions for the LP relaxation of original formulation and of the formulation where the cuts have been added. All classes of inequalities described in the previous sections have polynomial cardinality, and are thus added straightforwardly to the initial formulation. The total number of rows for the two linear programs is also reported. We can see from the table that, for all considered problems, the value of the LP solution is reduced to more than one half if the proposed cuts are added.

Table 2 reports the contributions of each of the ten classes of valid inequalities proposed in this paper considering the problems in set A. For each class the number of cuts that are added to the formulation is also reported. We note that the contributions of the classes differ significantly, being much more effective for those classes that have been proven to be facet inducing for certain subproblems of the SRP. It is worth mentioning that the last class of the table, the *Rest Shift*

Staff		Initial Formulation		Formulation with Cuts	
Members	cols	rows	LP sol.	rows	LP sol.
30A	27720	5380	54270	7786	26176
30B			54320		26150
60A	55440	10690	113730	15496	51432
60B			113830		51290
75B	69300	13345	153610	19351	77182
75B			153750		71940
100A	92400	17770	204790	25776	97290
100B			205000		96930

Table 1: Reduction in the LP solution

Subproblem cut (25), shows a very high contribution because it actually conveys most of the structural information determined by some of the other classes: in fact, that cut is determined by solving the Rest Shift Subproblem reinforced by the applicable classes of inequalities among the nine other classes.

Finally, we show in Table 3 some statistics related to the integer solutions of the Rest Shift Subproblem and of SRP. The number of nodes and CPU times in the Branch & Bound tree are also reported. Optimal solutions are obtained with reasonable computation times for the Rest Shift Subproblem and for the complete problem. For all problems, we determine very quickly an integer feasible solution with a negligible gap, solving the complete problem with rest shifts fixed according to the optimal solution of the Rest Shift Subproblem (last column of the Table 3), and this solution is often proved to be optimal. We remark that the stand-alone CPLEX 6.5 has not been able to solve the Rest Ship Subproblem for instance 30A within an hour of computation time against 21 seconds of our method.

	STAFF MEMBERS							
	30A		60A		75A		100A	
Class of	n.of	LP	n.of	LP	n.of	LP	n.of	LP
Inequalities	cuts	sol.	cuts	sol.	cuts	sol.	cuts	sol.
(10)	150	32300	300	71770	375	108570	500	149190
(13), (14)	240	36290	480	77914	600	108570	800	148710
(15), (16)	60	48280	120	98710	150	132162	200	176207
(17)	1	48280	1	99700	1	153181	1	204790
(18)	1	52280	1	109740	1	153610	1	204790
(19), (20)	2	48280	2	98710	2	145610	2	196890
(21), (22)	60	37290	120	77760	150	108580	200	148730
(23)	1	45290	1	98700	1	144294	1	196800
(24)	1890	54126	3780	113352	4725	152888	6300	203832
(25)	1	26300	1	51770	1	77610	1	97830

Table 2: Contributions of the classes of inequalities for problems in set A

	Rest Shift Subproblem Optimal Solution			Complete Problem Heuristic Solution				
Staff Members	number of nodes	time in seconds(*)	solution value	number of nodes	time in seconds(*)	solution value	relaxed solution	gap
30A	71	21	25000	65	226	26160	26160	0.0%
30B				51	130	26070	26130	0.2301%
60A	753	129	49000	212	1513	51390	51432	0.0817%
60B				431	835	51290	51290	0.0%
75A	1438	1146	69000	339	2897	72110	72182	0.0998%
75B				222	1164	71940	71940	0.0%
100A	3942	4547	93000	313	3712	97150	97290	0.1441%
100B				377	3235	96930	96930	0.0%

(*) Using CPLEX 6.5 on Digital Alpha Workstation 500Mhz

Table 3: Optimal and Heuristic solutions for Rest Shift Subproblem and Complete Problem

Acknowledgements

We thank Beniamino Paoletti and Giancarlo Broggio (Alitalia) for their collaboration; we also thank Giorgio Gallo (University of Pisa), Paolo Nobili (University of Lecce), and Giovanni Rinaldi (IASI-CNR) for their help and support in writing this paper.

References

[1] N. Balakrishnan and R. Wong, "A Network Model for the Rotating Workforce Scheduling Problem," *Networks*, vol. 20, pp. 25–42, 1990.

[2] J. Bartholdi III, "A Guaranteed-Accuracy Round-off Algorithm for Cyclic Scheduling and Set Covering," *Operations Research*, vol. 29/3, pp. 501–510, 1981.

[3] J. Bartholdi III, J. Orlin, and H. Ratliff, "Cyclic Scheduling via Integer Programs with Circular Ones," *Operations Research*, vol. 28/5, pp. 1074–1085, 1980.

[4] N. Beaumont, "Scheduling staff using mixed integer programming," *European Journal of Operations Research*, vol. 98/3, pp. 473–484, 1997.

[5] R. Burns and G. Koop, "A Modular Approach to Optimal Multiple-Shift Manpower Scheduling," *Operations Research*, vol. 35/1, pp. 100–110, 1987.

[6] K. Dowsland, "Nurse scheduling with tabu search and strategic oscillation," *European Journal of Operations Research*, vol. 106/2-3, pp. 393–407, 1998.

[7] B. Jaumard, F. Semet, and T. Vovor, "A generalized linear programming model for nurse scheduling," *European Journal of Operations Research*, vol. 107/1, pp. 1–18, 1998.

[8] H. Millar and M. Kiragu, "Cyclic and non-cyclic scheduling of 12h shift nurses by network programming," *European Journal of Operations Research*, vol. 104, pp. 582–592, 1998.

[9] H. Miller, W. Pierskalla, and G. Rath, "Nurse Scheduling Using Mathematical Programming," *Operations Research*, vol. 24/5, pp. 857–870, 1976.

[10] J. Morris and M. Showalter, "Simple Approaches to Shift, Days-Off and Tour Scheduling Problems," *Management Science*, vol. 29/8, pp. 942–950, 1983.

[11] G. Romanin Jacur and D. Zaghini, "Tecniche euristiche per la soluzione del problema di turnazione del personale aeroportuale," in *Giornate di Lavoro AIRO*, pp. 395–410, 1998.

[12] D. Warner, "Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Programming Approach," *Operations Research*, vol. 24/5, pp. 842–856, 1976.

[13] L. Wolsey, *Integer Programming.* Wiley, New York, 1998.