Counting and enumerating feasible rotating schedules by means of Gröbner bases

Raúl Falcón[1*], Eva Barrena[2,3*], David Canca[4*], Gilbert Laporte[2,3*]

[1] *School of Building Engineering, Avenida de Reina Mercedes 4 A, 41012, Seville, Spain*
[2] *Interuniversity Research Center on Network Enterprise, Logistics and Transportation (CIRRELT)*
[3] *HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7*
[4] *School of Engineering, University of Seville, Avenida de los Descubrimientos s/n, 41092, Seville, Spain*

Abstract

This paper deals with the problem of designing and analyzing rotating schedules with an algebraic computational approach. Specifically, we determine a set of Boolean polynomials whose zeros can be uniquely identified with the set of rotating schedules related to a given workload matrix subject to standard constraints. These polynomials constitute zero-dimensional radical ideals, whose reduced Gröbner bases can be computed to count and even enumerate the set of rotating schedules that satisfy the desired set of constraints. Thereby, it enables to analyze the influence of each constraint in the same.

Keywords: Rotating schedule, Boolean ideal, Gröbner basis.
2000 MSC: 90B35, 13F20, 13P10

1. Introduction

The workforce scheduling problem consists of assigning employees to shifts or days-off for a certain period of time. This can be done following a rotating (or cyclic) approach, in which case the schedule is repeated periodically over an infinite horizon or following a non-cyclic approach for a finite planning horizon.

*Corresponding author. Phone: + 34 954 550158; Fax: +34 954 556683
Email addresses: rafalgan@us.es (Raúl Falcón[1]), eva.barrena@cirrelt.ca (Eva Barrena[2,3]), dco@us.es (David Canca[4]), gilbert.laporte@cirrelt.ca (Gilbert Laporte[2,3])

In a rotating schedule, all employees have the same schedule but perform different shifts with a certain time offset. Such schedules are common to several industries and public sector organizations where work is carried out 24 hours a day, seven days a week. In such contexts, individual preferences of the employees are not taken into account, and the typical objective is to find schedules that guarantee equity between workers. The assignment of shifts per week to t distinct work teams yields a schedule of t rows and 7 columns. Specifically, the (i, j) entry of the schedule corresponds to the shift or rest period that is initially assigned to the i^{th} team, the j^{th} day of the first week. Once the week finishes, each team moves down to the following row of the schedule (or to the first row in case of being the last team) to perform the shift assignment of the new week. In the case of non-cyclic workforce schedules, individual preferences of employees can be taken into account and typically, the objective is the design of schedules fulfilling as much as possible the workers' preferences. This is often the case of nurse scheduling. Both approaches, rotating and non-cyclic schedules include constraints such as the minimum number of employees required for each shift and shift order preferences (for instance, no morning shifts immediately after night shifts) as well as weekend constraints (at least one day off during the weekend every three or four weeks).

Rotating and non-cyclic workforce scheduling problems are NP-complete problems (Lau [31]), and consequently hard to solve. Lau [31] analyzes the complexity of the changing shift assignment problem (CSAP), a rotating scheduling problem similar to the one considered in this paper where shift change constraints are imposed by a shift change Boolean matrix. The author considers the decision problem associated to the CSAP and investigates whether a feasible schedule exists, given a workload matrix, a set of teams, a set of shift types and a shift change matrix. The author demonstrates the complexity of the rotating scheduling problem by a polynomial many-one reduction from the 3SAT problem. In fact, the author concludes that the inclusion of order type constraints between different shifts (including days off) gives rise to NP-hard problems. Due to the high number of constraints that must be satisfied in practical appli-

cations, including typical constraints affecting order between shifts, designing computerized workforce schedules has been a research challenge during the last three or four decades ([6]).

Different approaches have been used to solve problems of workforce scheduling. The work of Tien and Kamiyama [36] contains an early survey on algorithms used for workforce scheduling. For a general view we refer the interested reader to the recent survey of Van den Bergh et al. [37], where a comprehensive list of hard and soft constraints encountered in these problems is reported. Manual approaches, integer programming, heuristic procedures, constraint programming and network flow models have been used in order to obtain rotating schedules [1]. McMillan [17] have integrated different techniques from management science and artificial intelligence to solve general shift scheduling problems. Balakrishnan and Wong [3] used a network flow formulation in order to solve the rotating workforce scheduling problem. The constraints were incorporated in the network, except for the staff-covering constraints which were treated as side constraints. A similar approach was used by Lau [30] who modeled the problem as a fixed-charge network and showed that a feasible schedule can be obtained by finding disjoint paths in the network. Several other ad hoc algorithms for rotating workforce schedules with different workweek lengths have been proposed [21, 22]. Laporte et al. [27] presented an effective ILP based algorithm for the construction of rotating schedules. Laporte [26] considers the design of rotating workforce schedules by hand and shows how relaxing several constraints can yield reasonable solutions. Laporte and Pesant [28] proposed a constraint programming algorithm that can handle a larger variety of constraints than previous methods. They first provide a classification of the main constraints classes governing the design of rotating schedules. The algorithm can easily produce several solutions within reasonable computing times.

Traditionally, the focus of previous researchers has been the design of rotating schedules with the objective of minimizing costs and maximizing employee satisfaction, and consequently, the different available methods do not generally produce all possible rotating schedules satisfying certain conditions, but only

those that are deemed to be of good quality. However, as discussed by Laporte and Pesant [28], the problem is to some extent fuzzy in the sense that optimality is not easily defined through a formula, but human judgement is required in practice to make a choice from a set of candidate solutions meeting predefined constraints. These authors propose a constraint programming algorithm capable of producing a set of high quality solutions to be presented to the decision makers. Several authors also emphasize the need to provide several solutions, pointing out the need for human judgement to make a choice from a set of feasible solutions. Different optimal solutions meeting predefined constraints are thus calculated by varying the criteria taken as objective function. Musliu et al. [33] generate a large number of plausible schedules to be evaluated with multiple criteria. The main feature of their framework is the possibility to generate high-quality schedules through human interaction. Castillo et al. [9] also generate different solutions, each optimizing a different criterion. Generating all solutions is also useful in view of the fact that new criteria are often discovered a posteriori and experience suggests that managers prefer to be presented with an array of solutions from which they can make a selection. Moreover, knowing the number of solutions helps analyze the influence of the constraints in the resulting solution set.

Following this observation, this paper focuses on the analysis of the number of solutions depending on the constraint types taken into consideration rather than on obtaining a single optimal solution. We are interested in generating all solutions satisfying certain constraints and in understanding the influence of each constraint type on the set of feasible solutions. Our methodology will therefore produce valuable information that can be used in an early phase of working conditions negotiation.

The literature on the analysis of constraint influence and the number of solutions of a given instance is scarce and problem specific. Pesant [34] exploits the problem structure and derives polynomial time evaluations of the number of solutions of individual constraints. These may be combined to approximate the total number of solutions or can be used to guide search heuristics. Pesant and

Quimper [35] propose and evaluate algorithms to compute solution densities of variable-value pairs in knapsack constraints.

The alternative approach that we propose in this paper is to make use of the combinatorial structure of rotating schedules in order to count and enumerate the solutions satisfying specific subsets of a predefined set of constraints. Like in combinatorial analysis, Gröbner bases enable us to count the number of solutions without actually enumerating them. Knowing the number of solutions is useful since this provides an assessment of how restrictive some constraints are. If there are too many conflicting constraints, then the instance may have no solution. Gröbner bases provide this information whereas it may take longer for other enumerative algorithms to prove infeasibilities. Specifically, we observe that the assignment of shifts per week to t distinct work teams in a rotating schedule can be represented by the entries of a $t \times 7$ array. One can thus observe the similarity between rotating schedules and Latin squares. A Latin square of order n is an $n \times n$ array in which each cell contains one element chosen from a set of n symbols (in our case, shift types), such that each symbol occurs precisely once in each row and column. A Latin square can then be considered as a very special case of a rotating schedule, since the latter allows to repeat shifts in the same week and in the same day. As such, we generalize the ideas developed in [13, 14] in order to determine explicitly the rotating schedules satisfying certain constraints. Accordingly, we use the polynomial method of Alon [2] and Bernasconi et al. [4] and we deal with the counting and enumeration of rotating scheduling as a combinatorial structure that generalize the concept of Latin square. Depending on the number of times shifts must be assigned each day, we impose certain conditions that can be modeled by polynomials whose variables represent the entries of the array. This facilitates the use of the polynomial method, which solves counting and enumeration problems in combinatorics by computing the reduced Gröbner basis of a zero-dimensional ideal uniquely related to a given combinatorial object. An analysis of the use of Gröbner basis techniques to solve discrete combinatorial problems with constraints has been recently proposed by Jefferson et al. [23].

The remainder of this paper is organized as follows. In Section 2, we indicate some preliminary concepts and results on commutative algebra. In Section 3, we enumerate the standard constraints that we are going to deal with. In Section 4, we identify the rotating schedules of a given workload matrix and satisfying a certain set of constraints, with the set of zeros of a Boolean ideal, which can be explicitly determined by computing the corresponding reduced Gröbner basis. Since the computation time required to obtain such a basis is extremely sensitive to the number of variables, we also show how to reduce it by means of generation by columns. The proposed methods are then implemented in three procedures in the open computer algebra system for polynomial computations SINGULAR [12], which are used in Section 5 to study the influence of several important constraint types in the design of rotating schedules related to part-time employers. Finally, we focus on the analysis of the well-known real case of the Edmonton Police Department [8], for which we prove its infeasibility with respect to the constraints exposed in Section 3 and we expose some alternative solutions to those that appear in the literature.

2. Preliminaries.

In order to analyze the constraints of the problem, we interpret them as a set of polynomials that we reduce to its Gröbner basis, from which we can extract fundamental information. For the sake of completeness, we first introduce some basic concepts of commutative algebra (see [10, 11] for more details). Let $R = k[\mathbf{x}] = k[x_1, \ldots, x_n]$ be a polynomial ring in n variables over a field k. A *total order* $\leq$ on R is a binary relation among the polynomials of R such that, given three polynomials $p, q, r \in R$, it is verified that

1. if $p \leq q$ and $q \leq p$, then $p = q$;
2. if $p \leq q$ and $q \leq r$, then $p \leq r$;
3. $p \leq q$ or $q \leq p$.

A *term order* $\prec$ on R is a total order on the set of monomials $\mathbf{x}^a = x_1^{a_1} \cdot \ldots \cdot x_n^{a_n}$ in R such that

1. given $a, b \in \mathbb{N}^n$ verifying that $\mathbf{x}^a \prec \mathbf{x}^b$, it is fulfilled that $\mathbf{x}^{a+c} \prec \mathbf{x}^{b+c}$, for all $c \in \mathbb{N}^n$;
2. $1 \prec \mathbf{x}^a$, for all $a \in \mathbb{N}^n \setminus \{\mathbf{0}\}$.

The largest monomial of a polynomial of R with respect to a given term order $<$ is called its *initial monomial*. A subset I of R is called an *ideal* of R if

1. $0 \in I$;
2. given two polynomials $p, q \in I$, it is verified that $p + q \in I$;
3. given two polynomials $p \in I$ and $q \in R$, it is verified that $p \cdot q \in I$.

The *variety* $V(I)$ related to an ideal I of R is defined as the set of *zeros* of its polynomials, that is to say,

$$V(I) = \{(a_1, \ldots, a_n) \in k^n \colon\ p(a_1, \ldots, a_n) = 0,\ \text{for all } p \in I\}. \tag{1}$$

Two polynomials $p, q \in R$ are *congruent modulo* an ideal I of R if $p - q \in I$. It is an equivalence relation. The *quotient* R/I is then defined as the set of equivalence classes of R with respect to this relation. The ideal generated by a finite set of polynomials $p_1, \ldots, p_m \in R$ is defined as

$$\langle p_1, \ldots, p_m \rangle = \{p \colon\ p = \sum_{i=1}^{m} q_i \cdot p_i,\ \text{ where } q_i \in R,\ \text{for all } i \in \{1, \ldots, m\}\}. \tag{2}$$

Given a term order $\prec$ and an ideal I of R, the ideal generated by the initial monomials of all the non-zero elements of I is called its *initial ideal* $I_\prec$. Any monomial of R not contained in $I_\prec$ is called a *standard monomial* of I with respect to $\prec$. If the variety $V(I)$ is finite, the ideal I is said to be *zero-dimensional*. In such a case, the quotient R/I is a finite-dimensional vector space whose dimension coincides with the number of standard monomials of I. Moreover, this

dimension is always greater than or equal to the number of points of $V(I)$. The equality is achieved if I is *radical*, that is, if any polynomial p belongs to I whenever there exists a natural $n \in \mathbb{N}$ such that $p^n \in I$.

The dimension of R/I and the points of $V(I)$ can be completely determined by means of Gröbner bases. A *Gröbner basis* of I with respect to $\prec$ is any generating set G of I such that the initial monomials of its elements generate the initial ideal $I_\prec$. It is said to be *reduced* if all its polynomials are monic and no monomial of a polynomial of G can be generated by the initial monomials of the other polynomials of the basis. This reduced basis is unique and can be decomposed into finitely many disjoint subsets, each of them being the zeros of a triangular system of equations, whose factorization and subsequent resolution are easier than the system related to the generators of the original ideal I [20, 29, 32]. The most general algorithm to obtain the reduced Gröbner basis of an ideal is the multivariate division algorithm for polynomials of Buchberger [7], which can be used over any field. Further, the algorithms F_4 and F_5 of Faugère [15, 16] and the algorithm *slimGB* of Brickenstein [5] are more efficient over the rational field or a finite field.

3. Problem description. Rotating Schedules

Several constraints must be taken into account in order to preserve equal opportunities among workers and to prevent health risks like stress, sleep disorder or digestive upsets. They are normally classified as hard and soft constraints, depending on whether they must be obligatorily fulfilled or whether they are preferable but not necessary. Often, the hard constraints compose the feasibility space and the soft ones are penalized in the objective function. In our case, we analyze the feasibility space resulting from the subsets of constraints taken into consideration, and therefore whenever we consider a subset of constraints, each of them will be taken as hard. Analogously to [27], we consider the following constraints:

C.1) Schedules should contain as many full weekends off as possible. That is

to say, given two rotating schedules that have the same number of Saturdays and Sundays off, the schedule with the greater number of complete weekends off is preferable. This is done by imposing that the number of weekends off must be equal to the minimum of the number of Saturdays and Sundays off.

C.2) Weekends off should be well spaced out in the cycle. We ensure that this is fulfilled by considering patterns in which the weekends off are as much well spaced as possible. That is to say, given two rotating schedules with the same number of weekends off, we consider the schedule with the greater number of rotations among periods of consecutive weekends off and periods of consecutive weekends with at least one working day. If both schedules have the same number of such rotations, we consider the one with the smallest mean deviation in the number of weeks that compose the periods of weekends off. Thus, for instance, the following three patterns of six-week rotating schedules, whose Saturdays off and Sundays off are represented by the symbol X, are ordered from worst to best according to what we have just stated. So, from the beginning we would impose the last pattern.

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Due to the cyclical structure of a rotating schedule, observe that any other choice based on a cyclic rotation of the weeks would yield the same original pattern.

C.3) A shift change can only occur after at least one day off.

C.4) The number of consecutive work days must not exceed 6 days and must not be less than 2.

C.5) The number of consecutive rest days must not exceed 6 days and must not be less than 2.

C.6) A shift change without at least 24 hours of rest is not allowed. Observe that C.3 implies C.6, since a day off suppose at least 24 rest hours.

In the following section, we use the combinatorial structure of any rotating schedule to model Constraints C.1–C.6 as a system of *Boolean polynomials*, that is to say, polynomials on a quotient ring $\mathbb{Z}/2\mathbb{Z}[x_1, \ldots, x_m]/\langle x_1^2 - 1, \ldots, x_m^2 - 1 \rangle$, whose zeros correspond to the set of rotating schedules that satisfy such conditions.

4. Boolean polynomials related to rotating schedules

In order to design a rotating schedule, it is necessary to know in advance its related *workload matrix*, that is, the number of shifts of each type that have to be assigned each day of the week. Given $s-1$ distinct shifts works, let the workload matrix $W = (w_{ij})$ be an $s \times 7$ array with all column sums equal to the number of work teams t. Each element w_{ij} indicates the number of work teams required for shift i on the j^{th} day if $i < s$ and the number of work teams having such a day as rest day, otherwise. The days are ordered from Monday (first column) to Sunday (last column). Thus, for instance, the next array is a workload matrix related to a rotating schedule with three distinct work shifts (Day, Evening and Night) and five work teams, where Mondays and Tuesdays are rest days for all the work teams.

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 5 & 5 & 3 & 1 & 1 & 2 & 2 \end{pmatrix}$$

Shift works are usually denoted in a rotating schedule by their initials: Day (D), Evening (E) and Night (N). Besides, rest days are usually denoted by X. A possible rotating schedule related to the previous workload matrix is then

$$\begin{pmatrix} \text{X} & \text{X} & \text{X} & \text{X} & \text{X} & \text{N} & \text{N} \\ \text{X} & \text{X} & \text{X} & \text{N} & \text{N} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} & \text{N} & \text{N} & \text{N} & \text{N} \\ \text{X} & \text{X} & \text{E} & \text{E} & \text{E} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{D} & \text{D} & \text{D} & \text{D} & \text{D} \end{pmatrix}$$

For our purposes, however, since we are interested in defining Boolean polynomials that can be used to design rotating schedules, we represent the $s-1$ work shifts by the numbers $1,\ldots,s-1$, in forward rotation order and the days off by the number s. The set $\{1,\ldots,s\}$ is denoted as $[s]$. The previous rotating schedule is then rewritten with the elements of the set [4]: Day (1), Evening (2), Night (3) and rest day (4).

$$\begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 3 & 3 \\ 4 & 4 & 4 & 3 & 3 & 4 & 4 \\ 4 & 4 & 4 & 3 & 3 & 3 & 3 \\ 4 & 4 & 2 & 2 & 2 & 4 & 4 \\ 4 & 4 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Let RS_W denote the set of rotating schedules of $s-1$ work shifts and t work teams, which have W as workload matrix. It can be identified with the set of $t\times 7$ arrays $R=(r_{ij})$ with elements taken from the set $[s]$ such that the frequency vector of the symbols that appear in each column of R is given by the corresponding column of W, that is, given $i\in[s]$ and $j\in[7]$, the j^{th} column of R contains w_{ij} occurrences the symbol i. The aim of the problem is to construct the subset RS_W satisfying Constraints C.1–C.6. Concretely, we first determine the relations between the entries of $R\in\text{RS}_W$ that are given by those of W and the conditions imposed by the constraints. These relations yield a zero-dimensional ideal whose solutions can be identified with the elements of RS_W and that can be obtained by using Gröbner bases. Thus, for instance, Constraint C.1 implies that any rotating schedule of RS_W should have $f_W=\min\{w_{s6},w_{s7}\}$ full weekends off. Constraint C.2 is of special interest since it enables to impose some of the entries of our future rotating schedule. This idea can be generalized for any entry of the matrix R. Concretely, those entries marked with the symbol s corresponding to the f_W full weekends off could be distributed by hand in advance, in a well-spaced way in the cycle. Indeed, this is the usual way to

proceed for designing rotating schedules [28]. Let $S_{t,s}$ be the set of $t \times 7$ arrays with entries in the set $[s] \cup \{0\}$. Given $E = (e_{ij}) \in S_{t,s}$, we say that $R = (r_{ij}) \in \mathrm{RS}_W$ contains E if $r_{ij} = e_{ij}$ for all $i \in [t]$ and $j \in [7]$ such that $e_{ij} \neq 0$. If $\mathrm{RS}_{W,E}$ denotes the subset of rotating schedules of RS_W containing E, then $RS_W = \bigcup_{E \in S_{t,s}} RS_{W,E}$. We show in Theorem 1 how each set $RS_{W,E}$ can be identified with the set of zeros of an ideal that is zero-dimensional and radical. Its reduced Gröbner basis can then be computed to explicitly determine the cardinality of $\mathrm{RS}_{W,E}$.

Theorem 1. *The set* $\mathrm{RS}_{W,E}$ *can be identified with the set of zeros of the following zero-dimensional ideal of* $\mathbb{Q}[x_{111}, \ldots, x_{t7s}]$.

$$I_{W,E} = \langle\, 1 - x_{ije_{ij}} :\ i \in [t], j \in [7], e_{ij} \in [s] \,\rangle + \langle\, x_{ijk} :\ i \in [t], j \in [7], e_{ij} \in [s], k \in [s] \setminus \{e_{ij}\} \,\rangle +$$

$$\langle\, x_{ijk} \cdot (1 - x_{ijk}) :\ i \in [t], j \in [7], k \in [s], e_{ij} = 0 \,\rangle + \langle\, 1 - \sum_{k \in [s]} x_{ijk} :\ i \in [t], j \in [7], e_{ij} = 0 \,\rangle +$$

$$\langle\, w_{kj} - \sum_{i \in [t]} x_{ijk} :\ j \in [7], k \in [s], w_{kj} \neq 0 \,\rangle.$$

Moreover, $|\mathrm{RS}_{W,E}| = \dim_{\mathbb{Q}}(\mathbb{Q}[x_{111}, \ldots, x_{t7s}]/I_{W,E})$.

Proof. Any rotating schedule $R = (r_{ij}) \in \mathrm{RS}_{W,E}$ can be uniquely identified with a zero $(x_{111}, \ldots, x_{t7s})$, where $x_{ijk} = 1$ if $r_{ij} = k$ and 0, otherwise. The finiteness of RS_W implies $I_{W,E}$ to be zero-dimensional. Besides, since $I_{W,E} \cap \mathbb{Q}[x_{ijk}] = \langle\, x_{ijk} \cdot (1 - x_{ijk}) \,\rangle \subseteq I_{W,E}$ for all $i \in [t]$, $j \in [7]$ and $k \in [s]$, Proposition 2.7 of [10] assures $I_{W,E}$ to be radical and thus, Theorem 2.10 of [10] implies that $|\mathrm{RS}_{W,E}| = |V(I_{W,E})| = \dim_{\mathbb{Q}}(\mathbb{Q}[x_{111}, ..., x_{t7s}]/I_{W,E})$. □

The ideal $I_{W,E}$ of Theorem 1 is embedded in a polynomial ring over the rational field $\mathbb{Q}$. In order to reduce the computational time necessary to determine the reduced Gröbner basis of such an ideal, it is interesting to study the possibility of replacing $\mathbb{Q}$ by a finite field. In fact, since the variables x_{ijk} of the ideal $I_{W,E}$ can only take the values zero and one, it is convenient to study the feasibility of working with the field $\mathbb{Z}/2\mathbb{Z}$. The resulting ideal would be then of Boolean type, whose suitability in the computation of reduced Gröbner bases related to solving counting problems in combinatorics was already exposed by

Bernasconi et al. in [4]. In our case, there exist two kinds of polynomials in the ideal $I_{W,E}$ that have to be slightly modified in order to work with the base field $\mathbb{Z}/2\mathbb{Z}$:

a) The polynomials of the form $1 - \sum_{k\in[s]} x_{ijk}$, where $i \in [t], j \in [7]$ and $e_{ij} = 0$. In $\mathbb{Z}/2\mathbb{Z}$, we could find a zero of the ideal $I_{W,E}$ containing more than one one in the assignment of 0/1-values to the s-tuple of variables $(x_{1jk}, \ldots, x_{sjk})$. Since this is not possible in our construction, we add the following monomials to the ideal $I_{W,E}$:

$$x_{ijk} \cdot x_{ijl}, \text{ for all } l \in \{j+1, \ldots, s\}.$$

b) The polynomials of the form $w_{kj} - \sum_{i\in[t]} x_{ijk}$, where $j \in [7], k \in [s]$ and $w_{kj} \neq 0$. In $\mathbb{Z}/2\mathbb{Z}$, the value of w_{kj} would be replaced by $w_{kj} \mod 2$ and hence, the set of zeros of the ideal $I_{W,E}$ would be modified. To recover the same set of zeros, we substitute each one of the mentioned polynomials in the ideal $I_{W,E}$ by

$$1 - \sum_{1\leq i_1<\ldots<i_{w_{kj}}\leq t} x_{i_1jk} \ldots x_{i_{w_{kj}}jk}.$$

Besides, in order to avoid a problem similar to that indicated in the previous assertion, we also add the following monomials:

$$x_{i_1jk} \cdot \ldots \cdot x_{i_{w_{kj}+1}jk}, \text{ for all } 1 \leq i_1 < \ldots < i_{w_{kj}+1} \leq t.$$

The next result is a consequence of the previous reasoning.

Theorem 2. *The set* $\mathrm{RS}_{W,E}$ *can be identified with the set of zeros of the following zero-dimensional ideal of* $\mathbb{Z}/2\mathbb{Z}[x_{111}, \ldots, x_{t7s}]$.

$$I'_{W,E} = \langle\, 1 - x_{ije_{ij}} :\ i \in [t], j \in [7], e_{ij} \in [s] \,\rangle + \langle\, x_{ijk} :\ i \in [t], j \in [7], e_{ij} \in [s], k \in [s] \setminus \{e_{ij}\} \,\rangle +$$

$$\langle\, x_{ijk} \cdot (1 - x_{ijk}) :\ i \in [t], j \in [7], k \in [s], e_{ij} = 0 \,\rangle + \langle\, 1 - \sum_{k\in[s]} x_{ijk} :\ i \in [t], j \in [7], e_{ij} = 0 \,\rangle +$$

$$\langle\, x_{ijk} \cdot x_{ijl} :\ i \in [t], j \in [7], k, l \in [s], e_{ij} = 0 \,\rangle +$$

$$\langle\, 1 - \sum_{1\leq i_1<\ldots<i_{w_{kj}}\leq t} x_{i_1jk} \ldots x_{i_{w_{kj}}jk} :\ j \in [7], k \in [s], w_{kj} \neq 0 \,\rangle +$$

$$\langle\, x_{i_1 jk} \cdot \ldots \cdot x_{i_{w_{kj}+1} jk} \colon\ 1 \leq i_1 < \ldots < i_{w_{kj}+1} \leq t, j \in [7], k \in [s], w_{kj} \neq 0 \,\rangle.$$

Moreover, $|\mathrm{RS}_{W,E}| = \dim_{\mathbb{Z}/2\mathbb{Z}}(\mathbb{Z}/2\mathbb{Z}[x_{111}, \ldots, x_{t7s}]/I'_{W,E})$. □

Constraints C.3−C.6 can also be imposed on our rotating schedules once we translate them to polynomials of $\mathbb{Q}[x_{111}, \ldots, x_{t7s}]$ or $\mathbb{Z}/2\mathbb{Z}[x_{111}, \ldots, x_{t7s}]$, which can be incorporated to the ideal $I_{W,E}$ of Theorem 1 or to the ideal $I'_{W,E}$ of Theorem 2, respectively. In fact, these polynomials do not depend on the base field. To see this, let us study each constraint separately.

Constraint C.3 implies that if two consecutive cells (i_1, j_1) and (i_2, j_2) of our rotating schedule contain two numbers $k, l < s$, then $k = l$. Such a condition can be translated to Boolean polynomials by imposing the monomial $x_{i_1 j_1 k} \cdot x_{i_2 j_2 l}$ to be zero whenever $k \neq l$. Thus, if $x_{i_1 j_1 k} = 1$, then $x_{i_2 j_2 l} = 0$ and hence, it is not possible to have a shift work change without at least one day off.

C.3) For all $i \leq t$, $j \leq 7$ and $k, l < s$ such that $k \neq l$, we add the monomial

$$x_{ijk} \cdot x_{((i+\lfloor \frac{j}{7} \rfloor - 1 \mod t)+1)((j \mod 7)+1)l} \tag{3}$$

In order to obtain the lower bound of two work days of Constraint C.4, we impose that given three consecutive cells, (i_1, j_1), (i_2, j_2) and (i_3, j_3), in our rotating schedule, if the first and third cells correspond to rest days, then the second one is also associated to a rest day. To this end, it is sufficient to impose the polynomial $x_{i_1 j_1 s} \cdot (x_{i_2 j_2 s} - 1) \cdot x_{i_3 j_3 s}$ to be zero. Hence, if $x_{i_1 j_1 s} = x_{i_3 j_3 s} = 1$ (that is, if the first and third days are rest days), then $x_{i_2 j_2 s} = 1$ (that is, the second day is also a rest day).

C.4.1) For all $i \leq t$ and $j \leq 7$, we add the polynomial

$$\begin{aligned} x_{ijs} \cdot (x_{((i+\lfloor \frac{j}{7} \rfloor - 1 \mod t)+1)((j \mod 7)+1)s} - 1) \cdot \\ x_{((i+\lfloor \frac{j+1}{7} \rfloor - 1 \mod t)+1)((j+1 \mod 7)+1)s} \end{aligned} \tag{4}$$

Analogously, for an upper bound of six work days, it is necessary that, given seven consecutive cells, $(i_1, j_1), \ldots, (i_7, j_7)$, of our rotating schedule, at least one

of such cells is associated to a rest day. To obtain it, we impose the polynomial $\prod_{d=1}^{7}(x_{i_d j_d s} - 1)$ to be zero. Since at least one of the seven variables must be distinct from zero, there exists at least one rest day every seven days.

C.4.2) For all $i \leq t$ and $j \leq 7$, we add the polynomial

$$\prod_{d=1}^{7}(x_{((i+\lfloor \frac{j}{d} \rfloor - 1 \mod t)+1)ds} - 1) \tag{5}$$

Observe that the first index of the variable x only changes when $d \leq j$. In that case, it moves to the next work team.

The polynomials related to Constraint C.5 are imposed analogously to those of Constraint C.4.

C.5.1) For all $i \leq t$ and $j \leq 7$, we add the polynomial

$$\begin{gathered}(x_{ijs} - 1) \cdot x_{((i+\lfloor \frac{j}{7} \rfloor - 1 \mod t)+1)((j \mod 7)+1)s} \cdot \\ (x_{((i+\lfloor \frac{j+1}{7} \rfloor - 1 \mod t)+1)((j+1 \mod 7)+1)s} - 1)\end{gathered} \tag{6}$$

C.5.2) For all $i \leq t$ and $j \leq 7$, we add the monomial:

$$\prod_{d=1}^{7} x_{((i+\lfloor \frac{j}{d} \rfloor - 1 \mod t)+1)ds} \tag{7}$$

Finally, Constraint C.6 implies that, if two consecutive cells (i_1, j_1) and (i_2, j_2) of our rotating schedule contain, respectively, two numbers $k, l < s$, then $k < l$. This condition can be translated to Boolean polynomials by imposing the binomial $x_{i_1 j_1 k} \cdot x_{i_2 j_2 l}$ to be zero whenever $k > l$.

C.6) For all $i \leq t$, $j \leq 7$ and $k, l < s$ such that $k > l$, we add the monomial

$$x_{ijk} \cdot x_{((i+\lfloor \frac{j}{7} \rfloor - 1 \mod t)+1)((j \mod 7)+1)l} \tag{8}$$

It is important to remark at this point that the computation of the reduced Gröbner basis of a zero-dimensional ideal is extremely sensitive to the number of variables [18, 19, 24, 25]. Specifically, Lackshman [24, 25] proved that the

complexity of such a computation is $O(d^n)$, where d is the maximal degree of the polynomials of the ideal and n is the number of variables. In our case, the degree of the polynomials is given by Theorems 1 and 2. Our algorithm is thus extremely sensitive to the number of variables, and it is convenient to reduce this number whenever possible. A first attempt for that is to eliminate those variables related to non-zero entries of the matrix E, because the assignment of the 0/1 values to those variables is uniquely determined. Specifically, if $e_{ij} \in [s]$, then both ideals $I_{W,E}$ and $I'_{W,E}$ contain the polynomial $1 - x_{ije_{ij}}$ and the set of monomials $\{x_{ijk} \colon k \in [s] \setminus \{j\}\}$. Hence, the variable $x_{ije_{ij}}$ can be substituted by 1 whenever it appears in a polynomial of $I_{W,E}$ or $I'_{W,E}$. Analogously, we can eliminate directly any monomial of such ideals that contain a variable x_{ijk}, with $k \in [s] \setminus \{e_{ij}\}$.

A more specific method to reduce the number of variables is to combine the polynomial method just described with that of generation by columns [28]. This last method consists of determining all the shifts of one day, before obtaining those of the following day. The number of variables necessary in such a case is considerably reduced, because it is not necessary to consider the subscript related to the day. Depending on the day, we need to consider different sets of multivariate polynomials related to Constraints C.1-C.6. Similarly to Theorems 1 and 2, these polynomials determine an ideal whose zeros are uniquely related to the rotating schedules of the set $\mathrm{RS}_{W,E}$. Due to their extensive length, these polynomials have been made available online on `http://personal.us.es/raufalgan/CrewGB.html`. Further, the following remarks are considered.

1) Firstly, it is useful to determine day-to-day the set $S_{W,E}$ of all the possible distributions of rest shifts that can be found in a rotating schedule of the set $\mathrm{RS}_{W,E}$. Any such distribution can be uniquely identified with a matrix $S = (s_{ij})$ whose entries are in the set $\{0, s\}$, where each symbol s represents the exact position of a rest shift. Observe that Constraints C.3 and C.6 do not have any influence in these distributions and hence,

we can omit them at this stage. Specifically, for a fixed day $d \in [7]$, we can consider the set of Boolean variables $\{x_i \colon i \in [t]\}$, such that $x_i = 1$ if there exists a rest shift the day d of the i^{th} week, and 0 otherwise. The following two polynomials should then be added to the considered ideal:

$$x_i - 1, \text{ if } e_{id} = s, \qquad \text{and} \qquad \sum_{i=1}^{t} x_i - w_{sd} \tag{9}$$

2) In those cases in which Constraints C.3 and C.6 are not under consideration, in order to calculate the cardinality of $\mathrm{RS}_{W,E}$, it is sufficient to consider any distribution $S = (s_{ij}) \in S_{W,E}$. If $E \cup S$ denotes the $t \times 7$ matrix whose entries combine those of E with the non-zero entries of S, then the following is verified:

$$|\mathrm{RS}_{W,E}| = |S_{W,E}| \cdot |\mathrm{RS}_{W,E \cup S}| \tag{10}$$

3) Given $S \in S_{W,E}$, if Constraint C.3 is imposed, then any block of consecutive work days contained between two consecutive blocks of rest days in S must be assigned to the same work shift. The variables related to these work days can be identified and hence, the number of variables of Theorems 1 and 2 is reduced.

5. Implementation of the method

We have considered all the results of the previous section to implement in SINGULAR three procedures, *rotating*, *rotating2* and *ColGen*, included in the library *scheduling.lib*, which is available online on `http://www.personal.us.es/raufalgan/LS/scheduling.lib`.

The three procedures determine explicitly the subset of rotating schedules of the set $\mathrm{RS}_{W,E}$ that satisfy some of the Constraints C.1−C.6. Specifically, the procedure *rotating* makes use of the ideal $I_{W,E}$ of Theorem 1 and Polynomials (3)−(8). The procedure *rotating2* also makes use of these polynomials, but it is based on the ideal $I'_{W,E}$ of Theorem 2 and on the reduction of variables indicated

in the second half of Section 4. Finally, the procedure *ColGen* combines both polynomial and generation by columns methods.

To test these procedures, we have considered the following four and five weeks workload matrices W_1 and W_2 used by Laporte [26]. These matrices correspond to part-time employees for which the initial workload matrix contains zero entries distributed throughout the week.

$$W_1 = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 4 & 4 & 2 & 0 & 0 & 1 & 1 \end{pmatrix} \qquad W_2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 5 & 5 & 3 & 1 & 1 & 2 & 2 \end{pmatrix}$$

According to Constraints C.1 and C.2, we have imposed that the corresponding rotating schedules must contain the following two respective arrays:

$$E_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad E_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Due to the cyclical structure of a rotating schedule, observe that our particular choice of the second weekend of $E1$ as weekend off does not have any influence on the subsequent computation and it can indeed be any other week. Analogously, what is important in the choice of weekends off in E_2 is that they are well spaced, but any other choice based on a cyclic rotation would yield the same result in the subsequent computations.

Once we have included the previous arrays as input of the procedures *rotating*, *rotating2* and *ColGen*, we show in Table 1 the number of rotating schedules $|\mathrm{RS}_{W_i,E_i}|$, $i = 1, 2$, related to such arrays, according to Constraints C.3−C.6. For each case, we also indicate the running times in seconds related to the procedures *rotating*, *rotating2* and *ColGen*, in a system with an *Intel Core i7-2600, 3.4 GHz* and *Ubuntu*. Specifically, we indicate the running time necessary to count the number of possible rotating schedules and, between parentheses, we also indicate the running time necessary to enumerate them. The algorithm which we have used to determine the reduced Gröbner bases is *slimGB* [5], which is efficiently implemented in SINGULAR.

Constraints				$\lvert RS_{W_1,E_1}\rvert$	Running time in seconds			$\lvert RS_{W_2,E_2}\rvert$	Running time in seconds		
C.3	C.4	C.5	C.6		*rotating*	*rotating2*	*ColGen*		*rotating*	*rotating2*	*ColGen*
				15,552	0 (439)	0 (20)	0 (14)	648,000	1 (*)	0 (6,672)	1 (661)
x				3	0 (1)	0 (0)	0 (0)	360	63 (119)	1 (8)	1 (2)
	x			15,552	0 (479)	0 (20)	0 (14)	171,072	3,031 (15,515)	18 (1,249)	1 (180)
		x		15,552	0 (720)	0 (20)	0 (14)	145,152	650 (11,904)	8 (947)	1 (157)
			x	81	0 (5)	0 (0)	1 (1)	13,824	5,121 (9,606)	20 (1,345)	38 (60)
x	x			3	0 (1)	0 (0)	0 (0)	42	5 (10)	0 (1)	0 (1)
x		x		3	0 (1)	0 (0)	0 (0)	62	11 (26)	1 (1)	1 (1)
x			x	3	0 (1)	0 (0)	0 (0)	360	93 (150)	1 (8)	1 (2)
	x	x		15,552	0 (697)	0 (20)	0 (14)	46,656	5 (3,449)	0 (168)	1(50)
	x		x	81	0 (5)	0 (0)	1 (1)	3,060	5,442 (7,611)	10 (353)	10 (15)
		x	x	81	0 (5)	0 (0)	1 (1)	1,848	1,249 (1,727)	5 (115)	9 (12)
x	x	x		3	0 (1)	0 (0)	0 (0)	10	1 (2)	1 (1)	0 (0)
x	x		x	3	0 (1)	0 (0)	0 (0)	42	3 (7)	1 (1)	0 (1)
x		x	x	3	0 (1)	0 (0)	0 (0)	62	6 (14)	1 (1)	1 (1)
	x	x	x	81	0 (5)	0 (0)	1 (1)	698	35 (258)	1 (13)	3 (4)
x	x	x	x	3	1 (1)	0 (0)	0 (0)	10	1 (3)	1 (1)	0 (0)

Table 1: Distribution of rotating schedules according to the type of constraints. * indicates a case for which the computer system runs out of memory.

The three methods explicitly determine those rotating schedules of RS_{W_1,E_1} and RS_{W_2,E_2} satisfying all the constraints (see last row of Table 1). Specifically, the set RS_{W_1,E_1} contains the following three rotating schedules:

$$\begin{pmatrix} 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3\\ 4&4&4&1&1&1&1 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&1&1&1&1\\ 4&4&4&3&3&3&3 \end{pmatrix}, \quad \begin{pmatrix} 4&4&1&1&1&1&1\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3\\ 4&4&4&3&3&3&3 \end{pmatrix}.$$

Analogously, the set RS_{W_2,E_2} contains the following ten rotating schedules:

$$\begin{pmatrix} 4&4&4&4&4&3&3\\ 4&4&4&3&3&4&4\\ 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&1&1&1&1&1 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&4&4&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3\\ 4&4&4&3&3&4&4\\ 4&4&1&1&1&1&1 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&3&3&3&3\\ 4&4&1&1&4&4&4\\ 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&4&1&1&1 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&4&1&1&1\\ 4&4&1&1&4&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&4&1&1&1\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3\\ 4&4&1&1&4&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix},$$

$$\begin{pmatrix} 4&4&4&4&4&3&3\\ 4&4&2&2&2&4&4\\ 4&4&1&1&1&1&1\\ 4&4&4&3&3&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&4&4&1&1\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3\\ 4&4&1&1&1&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&4&4&3&3\\ 4&4&4&3&3&4&4\\ 4&4&1&1&1&1&1\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&4&4&1&1\\ 4&4&1&1&1&4&4\\ 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix}, \quad \begin{pmatrix} 4&4&4&4&1&1&1\\ 4&4&1&1&4&4&4\\ 4&4&4&3&3&3&3\\ 4&4&2&2&2&4&4\\ 4&4&4&3&3&3&3 \end{pmatrix}.$$

The number of possible rotating schedules in Table 1 also provides information about the influence of each constraint on the final schedule. Observe, for

instance, how Constraints C.4 and C.5 do not have any influence on the design of a rotating schedule of workload matrix W_1, that is, they do not reduce the number of solutions when they are considered alone nor in combination with other constraints. However, the same constraints influence the design of rotating schedules of workload matrix W_2. It can also be observed that, since C.3 implies C.6, the latter has no influence in the number of solutions when both are considered together.

We analyze the effectiveness of the method by depicting the behavior of the computational time as the number of work teams increases. To this end, we have repeated the procedure just described, but we now analyze the following series of workload matrices:

$$W_i = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ i+3 & i+3 & i+1 & i-1 & i-1 & i & i \end{pmatrix},$$

for $i \in \mathbb{N}$. Recall that the number of work teams is equal to the column sum, that is, $i + 3$, and observe that the workload matrices for $i = 1$ and $i = 2$ correspond to the previously studied matrices W_1 and W_2. Figure 1 shows the computational time required by the procedure *ColGen* to count the number of rotating schedules related to each workload matrix W_i. The x-axis determines the workload matrix, and the y-axis the computational time. We show different curves depending on the set of constraints that has been considered. Since Constraints C.1 and C.2 are imposed a priori when creating the schedule pattern, we distinguish six distinct cases depending on which of the Constraints C.3–C.6 are considered: none of the four constraints C.3–C.6 is imposed ($\emptyset$); each constraint is separately imposed or all the four constraints are imposed together. We interpolate the results obtained for the discrete values of $i \in \mathbb{N}$ and thus obtain six curves. The exponential shape of these curves shows the already mentioned sensitivity of Gröbner bases to the number of variables. It can also be observed that the computational time improves for every workload matrix when all constraints are considered together, which fits with the results shown in Table

1 for W_1 and W_2, where it can be seen that the computational times diminishes as new constraints are added to the constraints set taken into consideration.

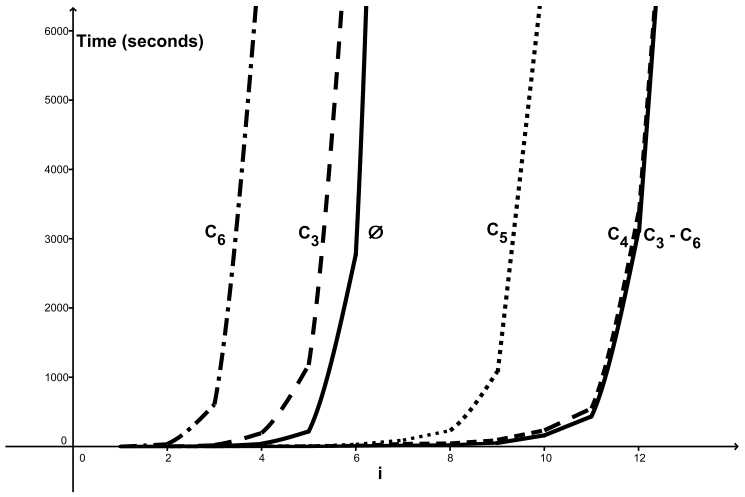

Figure 1: *ColGen* computational time for counting solutions of workload matrices W_i as the number of work teams $(i+3)$ increases.

As we have seen, our method can find the number of solutions and enumerate them for feasible cases. Unlike heuristic methods, it also identifies the infeasible cases. Algebraically, infeasibility means for example that the Gröbner basis of the corresponding ideal $I'_{W,E}$ in Theorem 2 is $\langle\, 1 \,\rangle$, or equivalently, the dimension of the quotient ring $\mathbb{Z}/2\mathbb{Z}[x_{111}, \ldots, x_{t7s}]/I'_{W,E}$ is 0.

In order to illustrate the effectiveness of our method for infeasible cases, we have applied it to the well-known real problem of Edmonton Police Department, introduced by Butler [8] and studied by Balakrishnan and Wong [3] and Laporte and Pesant [28]. This is a rotating workforce example with a 9-week cycle and 3 shift types that includes additional constraints in shift change patterns. Its related workload matrix is

$$W = \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 3 & 3 & 3 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 2 & 2 & 2 & 3 \end{pmatrix}$$

The method proposed by Balakrishnan and Wong needed 73.54 seconds to find a first solution of the problem and that of Laporte and Pesant needed 3.78 seconds. None of the proposed solutions in the literature satisfies all Constraints C.1–C.6. By using the procedure *ColGen*, we obtain in 347 seconds that there does not exist any distribution of rest shifts according to the workload matrix W and to Constraints C.1, C.2, C.4 and C.5 and hence, we can ensure that the problem is indeed infeasible under such constraints.

Besides, in order to compare the performance of our model with that of other models, we consider the proposal of Balakrishnan and Wong [3] and fix the same rest shifts as these authors do in their proposed solution. In 1.87 seconds (equal to 2.54% of the 73.54 seconds that they needed), we obtain not only the same solution as them, but also the following alternative solution, whose night shifts are concentrated in two blocks instead of three, and the opposite for the morning shifts:

$$\begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 4 \\ 4 & 1 & 1 & 1 & 1 & 1 & 4 \\ 4 & 4 & 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 4 & 4 & 1 & 1 & 1 \\ 1 & 4 & 4 & 2 & 2 & 2 & 2 \\ 2 & 2 & 4 & 4 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 4 & 4 & 1 \\ 1 & 1 & 1 & 1 & 4 & 4 & 4 \end{pmatrix}$$

Analogously, Laporte and Pesant [28] propose three solutions for this problem excluding Constraints C.3 and C.5. We fix the same rest days as them and prove in less than two seconds for each one of the three cases that there is no solution that satisfies such constraints.

6. Conclusions

We have shown how the polynomial method can be used in order to count and enumerate all possible rotating schedules that satisfy a given set of constraints, and thus to analyze their influence on the existence of such schedules. We have also seen that the computation time required for counting all feasible rotating schedules is small but that depending on the constraints considered, the

computational cost necessary to count and to enumerate them can be excessive even for small instances. We overcome this problem by improving the polynomial structure of the method in order to accelerate the counting process and by constructing the schedule using generation by columns in combination with Gröbner bases in order to enumerate the solutions within less time. This combination enables us to solve cases that were intractable by just using Gröbner bases due to memory and storage problems. It yields the same results for the tractable cases, normally within less computational time. Thereby, we provide a methodology that determines the exact number of rotating schedules for a given workload matrix and constraints, and also enumerates them when the problem is feasible. The decision maker can then choose the more convenient rotating schedules among all the feasible ones, and see the influence of each constraint in the resulting number of rotating schedules. In cases where the problem is infeasible, such as the Edmonton Police Department, our method enables us to detect infeasibility within a short time, to solve the same constraints relaxations as in the literature within 2.54% of the computation time, as well as to analyze how constraint relaxation influences the number and quality of feasible solutions.

Acknowledgements

This work was partly supported by the Excellence program of the Andalusian Government under grant P09-TEP-5022, and by the Natural Sciences and Engineering Research Council of Canada under grant 39682-10. This support is gratefully acknowledged. Thanks are due to the Editors and to the referees for their valuable comments.

References

[1] H. K. Alfares, Survey, categorization and comparison of recent tour scheduling literature, Ann. Oper. Res. 127 (2004) 145–175.

[2] N. Alon, Combinatorial Nullstellensatz, Combin. Probab. Comput. 8 (1999) 7–29.

[3] N. Balakrishnan, R. T. Wong, A network model for the rotating workforce scheduling problem, Netw. 20 (1990) 25–42.

[4] A. Bernasconi, B. Codenotti, V. Crespi, G. Resta, Computing Gröbner bases in the Boolean setting with applications to counting, in: G. F. Italiano, S. Orlando (Eds), Proceedings of the Workshop on Algorithm Engineering (WAE'97), University of Venice, Venice, 1997, pp. 209–218.

[5] M. Brickenstein, Slimgb: Gröbner bases with slim polynomials, Rev. Mat. Complut. 23 (2010) 453–466.

[6] P. Brucker, R. Qu, E. Burke, Personnel scheduling: Models and complexity, European J. Oper. Res. 210 (3) (2011) 467–473.

[7] B. Buchberger, Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, Ph. D. Thesis. University of Innsbruck. English translation 2006: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal, J. Symbolic Comput. 41 (2006) 475–511.

[8] B. Butler, Computerized manpower scheduling, Master thesis, University of Alberta, Canada, 1978.

[9] I. Castillo, T. Joro, Y. Y. Li, Workforce scheduling with multiple objectives, European J. Oper. Res. 196 (2009) 162–170.

[10] D. A. Cox, J. B. Little, D. O'Shea, Using Algebraic Geometry, Springer Verlag, New York, 1998.

[11] D. A. Cox, J. B. Little, D. O'Shea, Ideals, Varieties and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra, Springer Verlag, New York, 2007.

[12] W. Decker, G.-M. Greuel, G. Pfister, H. Schönemann, Singular 3-1-6. A computer algebra system for polynomial computations, 2014. URL http://www.singular.uni-kl.de.

[13] R. M. Falcón, The set of autotopisms of partial Latin squares, Discrete Math. 313 (2013) 1150–1161.

[14] R. M. Falcón, J. Martín-Morales, Gröbner bases and the number of Latin squares related to autotopisms of order ≤ 7, J. Symbolic Comput. 42 (2007) 1142–1154.

[15] J. C. Faugère, A new efficient algorithm for computing Gröbner bases (F_4), J. Pure Appl. Algebra 139 (1999) 61–88.

[16] J. C. Faugère, A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5), in: T. Mora (Ed) Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation (ISAAC'02), ACM, New York, NY, 2002, 75–83.

[17] F. Glover, C. McMillan, The general employee scheduling problem: An integration of MS and AI, Comput. Oper. Res. 13 (1986) 563–573.

[18] A. Hashemi, Nullstellensätze for zero-dimensional Gröbner bases, Comput. Complexity 18 (2009), 155–168.

[19] A. Hashemi, D. Lazard, Sharper complexity bounds for zero-dimensional Gröbner bases and polynomial system solving, Internat. J. Algebra Comput. 21 (2011), 703–713.

[20] D. Hillebrand, Triangulierung nulldimensionaler Ideale - Implementierung und Vergleich zweier Algorithmen, Master Thesis, Universitaet Dortmund, Fachbereich Mathematik, 1999.

[21] R. Hung, A three-day workweek multiple-shift scheduling model, J. Oper. Res. Soc. 44 (1993) 141–146.

[22] R. Hung, A multiple-shift workforce scheduling model under 4-day workweek with weekday and weekend labour demands, J. Oper. Res. Soc. 45(9) (1994) 1088–1092.

[23] C. Jefferson, P. Jeavons, M. J. Green, M. R. C. van Dongen, Representing and solving finite-domain constraint problems using systems of polynomials, Ann. Math. Artif. Intell. 67 (2013) 359–382.

[24] Y. N. Lackshman, On the complexity of computing a Gröbner basis for the radical of a zero dimensional ideal, in: H. Ortiz (Ed), Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing (STOC'90), ACM, New York, NY, 1990, pp. 555–563.

[25] Y. N. Lackshman, A single exponential bound on the complexity of computing Gröbner bases of zero-dimensional ideals, in: T. Mora, C. Traverso (Eds), Effective Methods in Algebraic Geometry, Progr. Math. 94, Birkhäuser Boston, Boston, MA, 1991, pp. 227–234.

[26] G. Laporte, The art and science of designing rotating schedules, J. Oper. Res. Soc. 50 (1999) 1011–1017.

[27] G. Laporte, Y. Nobert, J. Biron, Rotating schedules, European J. Oper. Res. 4 (1980), 24–30.

[28] G. Laporte, G. Pesant, A general multi-shift scheduling system, J. Oper. Res. Soc. 55 (2004) 1208–1217.

[29] D. Lazard, Solving zero-dimensional algebraic systems, J. Symb. Comp. 13 (1992) 117–132.

[30] H. C. Lau, Combinatorial approaches for hard problems in manpower scheduling, J. Oper. Res. Soc. Japan 39 (1996) 88–89.

[31] H. C. Lau, On the complexity of manpower scheduling, Comput. Oper. Res. 23 (1996) 93–102.

[32] H. M. Möller, On decomposing systems of polynomial equations with finitely many solutions, Appl. Algebra Eng. Commun. Comput. 4 (1993) 217–230.

[33] N. Musliu, J. Gärtner, W. Slany, Efficient generation of rotating workforce schedules, Discrete Appl. Math. 118 (2002) 85–98.

[34] G. Pesant, Counting solutions of CSPs: A structural approach, in: L. P. Kaelbling, A. Saffiotti (Eds), Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufmann, San Francisco, CA, 2005, pp. 260–265.

[35] G. Pesant, C.-G. Quimper, Counting solutions of knapsack constraints, in: L. Perron, M. A. Trick (Eds), Proceedings of the 5th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Springer, Berlin, 2008, pp. 203–217.

[36] J. M. Tien, A. Kamiyama, On manpower scheduling algorithms, SIAM Review 24 (1982) 275–287.

[37] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, L. De Boeck, Personnel scheduling: A literature review, European J. Oper. Res. 226 (2013) 367–385.