Designing rotating schedules by using Gröbner bases

Raúl Falcón[1], David Canca[1], Eva Barrena[2]

[1] University of Seville (Spain)

[2] CIRRELT and HEC Montréal (Canada)

{rafalgan, dco}@us.es, eva.barrena-algara@hec.ca

Abstract

In the current paper, we deal with the problem of designing rotating schedules from 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 and with the different constraints which are usually imposed to them. These polynomials constitute zero-dimensional radical ideals, whose reduced Gröbner bases can be computed to determine explicitly the set of rotating schedules which satisfy each constraint and hence, making possible to analyze their influence in the final pattern. Finally, we use this polynomial method to classify and characterize the set of rotating schedules related to a given number of shifts and work teams.

Keywords

Rotating schedule, Boolean ideal, Gröbner basis.

1 Introduction

Crew rostering is the last relevant step within the tactical phase of railway planning. Once the distinct shifts are designed to cover all programmed services, it becomes necessary to proceed with the individual assignment of the personal. The high complexity of this last task is mainly due to the differences which exist among shifts (compare the most common: day, evening and night shifts) from a quantitative as well as from a qualitative point of view. In addition, the individual acquired rights of the personnel have to be taken into consideration. Shift works have special relevance in those facilities which provide a service which is available at any time and day of the week. Due to the mentioned significant differences among shifts, labor schedules in these jobs have to be carefully designed. A scheduling pattern which is highly recommended for shift works is that of rotating schedules, where the assignment of shifts per week to n distinct work teams is explicitly exposed in a schedule of n rows and 7 columns. Specifically, the (i, j) entry of the schedule corresponds to the shift or rest period which 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 know the shift assignment of the new week.

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 which have to be assigned each day of the week. Besides, several constraints have to be taken into account to preserve equal opportunities among workers and to prevent health risks like stress, sleep disorder or digestive upsets. In the current paper, we consider the following six constraints exposed by Laporte [8, 9]:

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

C.2) Weekends off should be well spaced out in the cycle.

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) In consecutive days, forward rotations (day, evening, night) are generally preferred to backward rotations (day, night, evening).

There exist distinct methods and techniques in the literature to design rotating schedules [1] like manual approach, integer programming, heuristic procedures or network flows. Since the main goal of designing rotating schedules is minimizing costs and maximizing employee satisfaction, these methods do not determine in general all the possible rotating schedules verifying certain conditions, but only those which are on the path of finding the optimal model. However, it would be interesting to analyze the influence of each kind of constraint on the set of feasible solutions, that is, to deal with the number of rotating schedules which are eliminated or incorporated every time that we add or remove a specific condition. As a possible alternative, the combinatorial structure of any rotating schedule facilitates the use of the polynomial method established by Alon [2] and Bernasconi et al. [4], which solves enumeration and counting problems in Combinatorics by computing the reduced Gröbner basis of a zero-dimensional ideal uniquely related to a given combinatorial object. In this regard, see, for instance, the surveys of De Loera et al. [10, 11] on possible applications in graph theory. Indeed, graph theory has already been used in the scheduling problem [7].

The current paper is organized as follows. In Section 2, we identify the rotating schedules of a given workload matrix and satisfying Constraints C.1-C.6, with the set of zeros of a Boolean ideal, which can be explicitly determined by computing the corresponding reduced Gröbner basis. Such a computation has been implemented in a procedure in SINGULAR [6], which is used in Section 3 to study the influence of Constraints C.3-C.6 in the design of rotating schedules related to part time employers. Finally, since Gröbner bases are extremely sensitive to the number of variables, we show in Section 4 how the previous method can be improved by considering column generation.

2 Boolean polynomials related to rotating schedules.

Given two positive integers $s, t \in \mathbb{N}$, let $W = (w_{ij})$ be a $s \times 7$ array with all column sums equal to t and let RS_W denote the set of rotating schedules of s shift works (including that corresponding to rest days) and t team works, which have W as workload matrix. That is, w_{ij} indicates the number of team works which have to have the i^{th} shift the j^{th} day. 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.

Hereafter, $[s] = \{1, \ldots, s\}$ is assumed to represent the set of shift works of RS_W in forward rotation order (thus, for instance, $1, 2$ and 3 can represent, respectively, day, evening and night shifts), where the last symbol s corresponds to a rest day. In particular, the set RS_W can be identified with that of $t \times 7$ arrays $R = (r_{ij})$ based on $[s]$ such that the frequency vector of the symbols which 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} times the symbol i.

In practice, it is also interesting to have the possibility of imposing some of the entries of our future rotating schedule. Thus, for instance, according to Constraint C.2, the symbols 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, it is the usual way to proceed for designing rotating schedules [9]. In this regard, let $E = (e_{ij})$ be a $t \times 7$ array with entries in the set $[s] \cup \{0\}$, where $e_{ij} \in [s]$ if the entry (i, j) is imposed to our rotating schedule, or zero, otherwise. We say that $R = (r_{ij}) \in \mathrm{RS}_W$ contains E if $r_{ij} = e_{ij}$, for all $i \in [t]$ and $j \in [7]$. Let $\mathrm{RS}_{W,E}$ denote the subset of rotating schedules of RS_W containing E. The next result shows how this set can be identified with that of zeros of a Boolean ideal which is zero-dimensional and radical. Its reduced Gröbner basis can be then computed to determine explicitly the cardinality of $\mathrm{RS}_{W,E}$.

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

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

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

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

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 $I_{W,E} \cap \mathbb{Q}[x_{ijk}] = \langle\, x_{ijk} \cdot (1 - x_{ijk})\,\rangle \subseteq I_{W,s,t}$ for all $i \in [t]$, $j \in [7]$ and $k \in [s]$, Proposition 2.7 of [5] assures $I_{W,E}$ to be radical and thus, Theorem 2.10 of [5] implies that $|\mathcal{R}_{W,E}| = |V(I_{W,E})| = \dim_{\mathbb{Q}}(\mathbb{Q}[x_{111}, ..., x_{t7s}]/I_{W,E})$. □

Constraints C.3 to C.6 can be imposed to our rotating schedules if we translate them in terms of Boolean polynomials of $\mathbb{Q}[x_{111}, \ldots, x_{t7s}]$, which can be incorporated to the ideal $I_{W,E}$.

C.3) For all $k \in [s-1]$ and $l \in [s-1] \setminus \{k\}$, we add:

$$\begin{cases} x_{ijk} \cdot x_{i(j+1)l}, \text{ for all } i \in [t], j \in [6], \\ x_{i7k} \cdot x_{(i+1)1l}, \text{ for all } i \in [t-1], \\ x_{t7k} \cdot x_{11l}. \end{cases}$$

C.4) For a lower bound of 2 work days, we add, for each $k \in [s-1]$:

$$\begin{cases} (x_{ijk} - 1) \cdot x_{i(j+1)k} \cdot (x_{i(j+2)k} - 1), \text{ for all } i \in [t], j \in [5], \\ (x_{i6k} - 1) \cdot x_{i7k} \cdot (x_{(i+1)1k} - 1), \text{ for all } i \in [t-1], \\ (x_{i7k} - 1) \cdot x_{(i+1)1k} \cdot (x_{(i+1)2k} - 1), \text{ for all } i \in [t-1], \\ (x_{t6k} - 1) \cdot x_{t7k} \cdot (x_{11k} - 1), \\ (x_{t7k} - 1) \cdot x_{11k} \cdot (x_{12k} - 1). \end{cases}$$

For an upper bound of 6 work days, we add:

$$\begin{cases} \prod_{j=d}^{7} x_{ijk} \cdot \prod_{j=1}^{d-1} x_{(i+1)jk}, \text{ for all } i \in [t-1], d \in [7], k \in [s-1], \\ \prod_{j=d}^{7} x_{tjk} \cdot \prod_{j=1}^{d-1} x_{1jk}, \text{ for all } d \in [7], k \in [s-1]. \end{cases}$$

C.5) Similarly to Constraint C.4, we add:

$$\begin{cases} (x_{ijs} - 1) \cdot x_{i(j+1)s} \cdot (x_{i(j+2)s} - 1), \text{ for all } i \in [t], j \in [5], \\ (x_{i6s} - 1) \cdot x_{i7s} \cdot (x_{(i+1)1s} - 1), \text{ for all } i \in [t-1], \\ (x_{i7s} - 1) \cdot x_{(i+1)1s} \cdot (x_{(i+1)2s} - 1), \text{ for all } i \in [t-1] \\ (x_{t6s} - 1) \cdot x_{t7s} \cdot (x_{11s} - 1), \\ (x_{t7s} - 1) \cdot x_{11s} \cdot (x_{12s} - 1). \end{cases}$$

$$\begin{cases} \prod_{j=d}^{7} x_{ijs} \cdot \prod_{j=1}^{d-1} x_{(i+1)js}, \text{ for all } i \in [t-1], d \in [7], \\ \prod_{j=d}^{7} x_{tjs} \cdot \prod_{j=1}^{d-1} x_{1js}, \text{ for all } d \in [7]. \end{cases}$$

C.6) For all $k \in \{2, \ldots, s-1\}$, $l \in [k-1]$, we add:

$$\begin{cases} x_{ijk} \cdot x_{i(j+1)l}, \text{ for all } i \in [t], j \in [6], \\ x_{i7k} \cdot x_{(i+1)1l}, \text{ for all } i \in [t-1], \\ x_{t7k} \cdot x_{11l}. \end{cases}$$

3 Implementation of the method.

We have considered all the Boolean polynomials of the previous section in order to implement in Singular the procedure *rotating* [3], which determines explicitly the subset of rotating schedules of $\mathrm{RS}_{W,E}$, which satisfy some of the Constraints C.1-C.6. It is worth highlighting the effectiveness of this procedure in case of considering rotating schedules related to part time employees for which the initial workload matrix contains zero entries distributed throughout the week. To test it, we have considered the following two workload matrices used by Laporte in [8].

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

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

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

We show in Table 1 the number of rotating schedules related to the previous arrays, according to the constraints C.3-C.6 which can be imposed. In each case, we also indicate the running time (*r.t.*) in seconds which has been necessary in a system with an *Intel Core i7-2600, 3.4 GHz* and *Ubuntu*. The computational cost of those cases marked by an asterisk has turned out to be excessive for the processing capability of the mentioned computer system.

Constraints C.3	C.4	C.5	C.6	$\lvert\mathcal{RS}_{W_1,E_1}\rvert$	r.t.	$\lvert\mathcal{RS}_{W_2,E_2}\rvert$	r.t.
				15,552	0	648,000	0
x				3	0	360	97
	x			36	0	216	8
		x		15,552	0	145,152	650
			x	81	0	*	*
x	x			3	1	42	4
x		x		3	1	62	14
x			x	3	1	71	93
	x	x		36	1	48	7
	x		x	9	1	360	13
		x	x	81	1	*	*
x	x	x		3	1	10	3
x	x		x	3	1	42	6
x		x	x	3	1	62	15
	x	x	x	9	1	30	8
x	x	x	x	3	1	10	5

Table 1: Distribution of rotating schedules according to the type of constraints.

The three rotating schedules related to W_1 and E_1 which satisfies all the constraints are:

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

The ten rotating schedules related to W_2 and E_2 which satisfy all the constraints are:

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

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

The number of possible rotating schedules in Table 1 may also give us information about the influence of each constraint on the final schedule. Thus, for instance, we can observe how Constraint C.5 does not have any influence on the design of a rotating schedule of workload matrix W_1, i.e., it does not diminishes the number of solutions when it is considered alone neither in combination with other constraints. However, it can be observed that it has influence on the design of rotating schedules of workload matrix W_2.

4 Final remarks and further work.

In the current paper, we have shown how the polynomial method can be used in order to determine explicitly all the possible rotating schedules which satisfy a given set of constraints and to analyze their influence on the existence of such schedules. Besides, we have just seen in Table 1 that, depending on the constraints in which we are interested, the computational cost which is necessary to obtain a rotating schedule can be excessive even for small orders. A possible alternative to be considered as further work is to construct such a schedule by using the *column generation* method [9], which consists of determining all the shifts of one day, before of obtaining those of the following day. The number of variables which is necessary to use in such a case is considerably reduced and hence, the computational cost is improved.

References

[1] H. K. Alfares, Survey, Categorization, and Comparison of Recent Tour Scheduling Literature, Annals of Operations Research 127 (2004) no. 1-4, 145–175.

[2] N. Alon, Combinatorial Nullstellensatz, Recent trends in combinatorics (Mátraháza, 1995). Combin. Probab. Comput. 8 (1999) no. 1–2, 7–29.

[3] E. Barrena, D. Canca and R. M. Falcón, `http://personal.us.es/raufalgan/LS/crew.lib`.

[4] A. Bernasconi, B. Codenotti, V. Crespi and G. Resta, Computing Groebner Bases in the Boolean Setting with Applications to Counting, 1st Workshop on Algorithm Engineering (WAE). Venice, Italy, 1997, pp. 209–218.

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

[6] W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann, Singular 3-1-6. A computer algebra system for polynomial computations, 2013. `http://www.singular.uni-kl.de`.

[7] M. Gamache, A. Hertz and J. O. Ouellet, A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding, Computers & OR 34 (2007) no. 8, 2384–2395.

[8] G. Laporte, The art and science of designing rotating schedules, Journal of the Operational Research Society 50 (1999) no. 10, 1011–1017.

[9] G. Laporte and G. Pesant, A general multi-shift scheduling system, Journal of the Operational Research Society 55 (2004) no. 11, 1208–1217.

[10] J. A. Loera, J. Lee, S. Margulies and S. Onn, Expressing Combinatorial Problems by Systems of Polynomial Equations and Hilberts Nullstellensatz, Combinatorics, Probability and Computing 18 (2009) 551–582.

[11] J. A. De Loera, C. J. Hillar, P. N. Malkin and M. Omar, Recognizing Graph Theoretic Properties with Polynomial Ideals, Electron. J. Combin. 17 (2010) no. 1, Research Paper 114, 26 pp.