ESCO 2016. 5th European Seminar on Computing (Pilsen, Czech Republic, June 5-10, 2016)

Analyzing constraints influence in the design of rotating crew schedules.

R. M. Falcón[1] (rafalgan@us.es), E. Barrena[2], D. Canca[1], G. Laporte[3]
[1] University of Seville, [2] University of Granada, [3] HEC Montreal

D	D	D	D	D	R	R
E	E	R	R	N	N	N
R	R	E	E	E	E	E
R	R	R	R	R	D	D
N	N	N	N	R	R	R

Abstract. The rotating crew scheduling problem can be modeled as an optimization problem in which there are several constraints regarding the shifts of the workers and one commonly used objective function is to balance the number of hours per week for every worker. Often, managers prefer to be presented with an array of solutions from which they can make a selection, and there are constraints which reduce the feasibility space more than others. Our aim is to provide the set of all solutions rather than a unique optimal solution as well as to analyze the effect of the constraints in this solution set. For that aim, we follow an algebraic computational approach that solves both days off and shift assignment problems related to the design of rotating schedules. 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 constraints imposed to them.

Introduction.

- **Shift**: Team of employees who work for a specific period of time:

  Day (D), Evening (E), Night (N), Rest (R).
- **s**: Number of shifts (including Rest).
- **Schedule**: Timetable which shows the distribution of shifts.
- **Scheduling**: Process of creating a schedule by **minimizing costs** and **maximizing employee satisfaction**.
  - It has to be carefully designed in services **available 24/7**.
- **Rotating Schedule**: $n \times 7$ schedule whose (i, j) entry corresponds to the shift or rest period initially assigned to the i^{th} team, the j^{th} day of the first week of work.
- **Workload matrix**: $s \times 7$ array whose (i, j) entry determines how many times the i^{th} shift has to be assigned the j^{th} day of the week.

$$W = (w_{ij}) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

Several **constraints** are necessary to preserve equal opportunities among workers and to prevent health risks [Laporte, 1999, 2004].

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).

D	D	D	D	D	R	R
E	E	R	R	N	N	N
R	R	E	E	E	E	E
R	R	R	R	R	D	D
N	N	N	N	R	R	R

→

1	1	1	1	1	4	4
2	2	4	4	3	3	3
4	4	2	2	2	2	2
4	4	4	4	4	1	1
3	3	3	3	4	4	4

Methods to design rotating schedules:

- Manual approach.
- Integer programming.
- Heuristic procedures.
- Network flows.

Advantages:

- Find the optimal rotating schedule by minimizing costs and maximizing employee satisfaction.

Disadvantages:

- Do not determine all the possible rotating schedules.
- Do not 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.

Boolean polynomials related to rotating schedules.

- RS_W: Set of rotating schedules of s shift works and t team works, which have $W = (w_{ij})$ as workload matrix.
- $[s] = \{1, \ldots, s\}$: Set of shift works of RS_W in **forward rotation order**. The s^{th} shift correspond to rest periods).
- $RS_W \equiv$ Set of $t \times 7$ arrays $R = (r_{ij})$ on $[s]$ such that, given $i \in [s]$ and $j \in [7]$, the j^{th} column of R contains w_{ij} times the symbol i.

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix} \rightarrow R \equiv \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 4 & 4 \\ \hline 2 & 2 & 4 & 4 & 3 & 3 & 3 \\ \hline 4 & 4 & 2 & 2 & 2 & 2 & 2 \\ \hline 4 & 4 & 4 & 4 & 4 & 1 & 1 \\ \hline 3 & 3 & 3 & 3 & 4 & 4 & 4 \\ \hline \end{array}$$

It is interesting to **impose some of the entries** of the rotating schedule.

C.2 ⇒ Full weekends off well spaced.

$$\begin{cases} W = (w_{ij}) \\ E = (e_{ij}) \end{cases} \rightarrow \quad R = (r_{ij}) \in RS_{W,E}$$

Variables of the Polynomial Method:

$$x_{ijk} = \begin{cases} 1, & \text{if } r_{ij} = k, \\ 0, & \text{otherwise.} \end{cases}$$

$$i \in [t] \quad j \in [7] \quad k \in [s]$$

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

$$\begin{aligned} I_{W,E} = \; & \langle 1 - x_{ije_{ij}} : i \in [t], j \in [7], e_{ij} \in [s] \rangle + \\ & \langle x_{ijk} : i \in [t], j \in [7], e_{ij} \in [s], k \in [s] \setminus \{e_{ij}\} \rangle + \\ & \langle x_{ijk} \cdot (1 - x_{ijk}) : i \in [t], j \in [7], k \in [s], e_{ij} = 0 \rangle + \\ & \langle 1 - \textstyle\sum_{k \in [s]} x_{ijk} : i \in [t], j \in [7], e_{ij} = 0 \rangle + \\ & \langle x_{ijk} : i \in [t], j \in [7], k \in [s], w_{kj} = 0 \rangle + \\ & \langle w_{kj} - \textstyle\sum_{i \in [t]} x_{ijk} : j \in [7], k \in [s], w_{kj} \neq 0 \rangle. \end{aligned}$$

Moreover,

$$|RS_{W,E}| = \dim_{\mathbb{Q}}(\mathbb{Q}[x_{111}, \ldots, x_{t7s}]/I_{W,E})$$

.

Implementation of the method in Singular. Examples. [Laporte, 1999]

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

According to Constraints C.1 and C.2, we impose:

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

Constraints C.3	C.4	C.5	C.6	$\lvert\mathcal{RS}_{W_1,E_1}\rvert$	r.t. (sec)	$\lvert\mathcal{RS}_{W_2,E_2}\rvert$	r.t. (sec)
				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	360	93
	x	x		36	1	48	7
	x		x	9	1	108	27
		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.

References.

- N. Alon, Combinatorial Nullstellensatz, Recent trends in combinatorics (Mátraháza, 1995). Combin. Probab. Comput. 8 (1999) no. 1–2, 7–29.
- D. A. Cox, J. B. Little and D. O'Shea, Using Algebraic Geometry, Springer-Verlag, New York, 1998.
- W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann, Singular 4-0-2. A computer algebra system for polynomial computations, 2016. http://www.singular.uni-kl.de.
- R. M. Falcón, E. Barrena, D. Canca and G. Laporte, http://personal.us.es/raufalgan/LS/crew.lib.
- R. M. Falcón, E. Barrena, D. Canca and G. Laporte, Counting and enumerating feasible rotating schedules by means of Gröbner bases, Math. Comput. Simul. 125 (2016), 139-151.
- G. Laporte, The art and science of designing rotating schedules, Journal of the Operational Research Society 50 (1999) no. 10, 1011–1017.
- G. Laporte and G. Pesant, A general multi-shift scheduling system, Journal of the Operational Research Society 55 (2004) no. 11, 1208–1217.