A catalog of regular rules in personnel scheduling problems

Guillaume Ghienne, Odile Bellenguez, Guillaume Massonnet, María I. Restrepo

1 Definitions and notations

We assume that time is divided in discrete consecutive *intervals* $\mathcal{I}$ and that Σ is a finite and non empty set of *status*. A status describes an assignment (e.g., working activity, rest break) for a given interval $i \in \mathcal{I}$. We therefore call *schedule* any sequence of status, no matter its size. In particular, ϵ denotes an empty schedule and given a status $s \in \Sigma$ and an integer $i \in \mathbb{N}$ we note s^i the schedule consisting of i successive status s. Each status $sin\Sigma$ might have a "cost", denoted as c_s. This cost might represent, for instance, the number of working periods of the status. In the following we explicitly refer to two Personnel Scheduling Problems (PSP): the Nurse Scheduling Problem (NSP) and the Multi-Activity Shift Scheduling Problem (MASSP). For each of these problems, Table 1 presents an example of sets $\mathcal{I}$ and Σ that we will reuse throughout this document.

	$\mathcal{I}$	Σ
NSP	days	$\{$*Early (E), Late (L), Night (N), Day-off (O)*$\}$
MASSP	15-min intervals	$\{$*Activity 1 (A_1), Activity 2 (A_2) , Rest (R), Break (B), Lunch (L)*$\}$

Table 1: Example of interpretation of our formalism for different PSP.

A *regular rule* is a set of schedules (possibly infinite) that can be represented by a regular language over the alphabet Σ. It implicitly models a working regulation usually expressed as a verbal expression (e.g., "a *Night* shift cannot immediately precede an *Early* shift" or "a working day cannot contain more than five assignments of *Activity 1*"). This document proposes a list of generic rules commonly present in the PSP literature, and a way of modeling them with our framework composed of four standard parametric rules (`cardinality`, `stretch`, `pattern`, `knapsack`) as well as four operations (`side`, `windows`, `periodic`, `mask`) and some usual operations on regular language as union, intersection or complementation.

2 One-level rules

We can express some rules using intervals $\mathcal{I}$ as time basis (e.g., constraining a number of *Night* shifts assignments in an NSP or bounding a number of consecutive intervals with the same activity in a MASSP). In the following we call these rules *one-level rules* in contrast to multi-level rules that we will define in the next section. We can identify two categories of one-level rules. The *horizontal rules* are expressed over the entire schedule whereas *local rules* constrain only some sub-sequences of a schedule (e.g., blocks of consecutive identical status). Let $\Sigma = \{s_1, \ldots, s_n\}$ a set of $n \geq 2 \in \mathbb{N}$ status and let $\mathcal{T} = \{\Sigma_1, \ldots, \Sigma_m\}$ be a partition of Σ of size $m \in \{2, \ldots, n\}$. For all $i \in \{1, \ldots, m\}$, we call Σ_i the type of any status in Σ_i and we can assume without loss of generality that $s_i \in \Sigma_i$. Let $\mathcal{P} = \Sigma'_1, \ldots, \Sigma'_{m'}$ with $m' \geq 2 \in \mathbb{N}$ be a sequence of elements in $\mathcal{T}$ such that for all $i \in \{1, \ldots, m'-1\}, \Sigma_i \neq \Sigma_{i+1}$. Finally, let $c_i \in \mathbb{R}^+$ be a assignment cost associated with any assignment in Σ_i for all $i \in \{1, \ldots, m\}$. To not overload the notations, we do not make the difference between a status $s \in \Sigma$ and the corresponding singleton $\{s\}$.

2.1 Horizontal rules

Cardinality: for all $i \in \{1, \ldots, m\}$, there must be at least $l_i \in \mathbb{N}$ and at most $u_i \geq l_i \in \mathbb{N} \cup \{\infty\}$ occurrences of a status in Σ_i,

$$\texttt{cardinality}(\langle \Sigma_1, \ldots, \Sigma_m \rangle, \langle l_1, \ldots, l_m \rangle, \langle u_1, \ldots, u_m \rangle).$$

Example (NSP). $\texttt{cardinality}(\langle N, O, \{E, L\}\rangle, \langle 0, 2, 0\rangle, \langle 2, 3, \infty\rangle)$ this rule ensures that a schedule must have at most 2 “number of *Night* shifts and between 2 or 3 days-off”.

Example (MASSP). $\texttt{cardinality}(\langle\{A_1, A_2\}, \{R, B, L\}\rangle, \langle 12, 0\rangle, \langle 24, \infty\rangle)$ this rule ensures that the “total working time lies between 3 and 6 hours”.

Cardinality (periodic partial variant): this is a cardinality rule that is only applied to the subsequence build by keeping only the ith assignment of a schedule if there is $(q, r) \in \mathbb{N} \times \{0, \dots, k-1\}$ such that $i = kq + r$ and $b_r = 1$ where $k \in \mathbb{N}^*$ and $b_0 \dots b_{k-1} \in \{0, 1\}^k$,

$$\texttt{mask}(\Pi^{\mathcal{C}}, b_0 \dots b_{k-1})$$

where $\Pi^{\mathcal{C}}$ is a cardinality rule.

Example (NSP). $\texttt{mask}(\Pi^{\mathcal{C}}, 0000011)$ only applies the rule $\Pi^{\mathcal{C}}$ to weekend days (assuming schedules start on a Monday).

Knapsack: this rule guarantees that the sum of the *status* assignment costs is at least $l \in \mathbb{R}^+$ and at most $u \geq l \in \mathbb{R} \cup \{\infty\}$,

$$\texttt{knapsack}(\langle\Sigma_1, \dots, \Sigma_m\rangle, \langle c_1, \dots, c_m\rangle, l, u).$$

Example (NSP). “The total working time in a schedule must fall between 28 and 40 hours”. This rule is represented by $\texttt{knapsack}(\langle\{E, L\}, N, O\rangle, \langle 7.5, 10, 0\rangle, 28, 40)$, where *Night* shifts are 10 hours long and *Early* and *Late* shifts last 7.5 hours.

Pattern: in this rule assignment types must appear in the order defined by $\mathcal{P}$ and the ith stretch of assignments of the same type must have a size of at least $l_i \in \mathbb{N}^*$ and at most $u_i \geq l_i \in \mathbb{N} \cup \{\infty\}$ for all $i \in \{1, \dots, m'\}$,

$$\texttt{pattern}(\langle\Sigma_1, \dots, \Sigma_m\rangle, [\Sigma'_1 \dots \Sigma'_{m'}], \langle l_1, \dots, l_{m'}\rangle, \langle u_1, \dots, u_{m'}\rangle)).$$

Example (MASSP). $\texttt{pattern}(\langle R, \{A_1, A_2, B, L\}\rangle, [R\{A_1, A_2, B, L\}R], \langle 16, 1, 16\rangle, \langle\infty, \infty, \infty\rangle))$ is a representation of the rule “At least four hours of *Rest* activities have to be placed at the beginning and at the end of the schedule, there must be at least one other activity during the day”.

Pattern (0-stretch variant): as a pattern rules, assignment types must appear in the order defined by $\mathcal{P}$ and the ith stretch of assignments of the same type must have a size of at least $l_i \in \mathbb{N}^*$ and at most $u_i \geq l_i \in \mathbb{N} \cup \{\infty\}$ for all $i \in \{1, \dots, m'\}$, except for one stretch $j \in \{1, \dots, m'\}$ for which $l_j = 0$.

$$\Pi^{\mathcal{P}} \cup \Pi^{\mathcal{P}_j}$$

where $\Pi^{\mathcal{P}}$ is the corresponding pattern rule assuming $l_j = 1$ and $\Pi^{\mathcal{P}_j}$ is the pattern rule build as follow: if $j = 1$ or $j = m'$ or $\Sigma'_{j-1} \neq \Sigma'_{j-1}$ then simply remove j from $\{1, \dots, m'\}$ in the pattern definition, otherwise, remove j and $j-1$ from $\{1, \dots, m'\}$ and replace l_{j+1} (resp. u_{j+1}) by $l_{j-1} + l_{j+1}$ (resp. $u_{j-1} + u_{j+1}$). Such a variant is easy to generalize to a set of $m'' \leq m' \in \mathbb{N}^*$ that may not appear in the pattern by doing the union of the $2^{m''}$ resulting pattern rules.

2.2 Local rules

Stretch: for all $i \in \{1, \dots, m\}$, the consecutive assignments with the same type Σ_i must appear in stretches of length between $l_i \in \mathbb{N}^*$ and $u_i \geq l_i \in \mathbb{N} \cup \{\infty\}$,

$$\texttt{stretch}(\langle\Sigma_1, \dots, \Sigma_m\rangle, \langle l_1, \dots, l_m\rangle, \langle u_1, \dots, u_m\rangle).$$

Example (NSP). The rule “Days-off has to appear in stretches with size at least 2, and a valid schedule cannot have more than 3 consecutive *Nights*” is represented by $\texttt{stretch}(\langle N, O, \{E, L\}\rangle, \langle 1, 2, 1\rangle, \langle 3, \infty, \infty\rangle)$.

Stretch (∞-side variant): this rule relaxes the minimum length of stretches of type Σ_i for $i \in \{1, \dots, m\}$ at the beginning and the end of a schedule,

$$\texttt{side}(\Pi^{u_i=\infty}, s_i^{l_i}, s_i^{l_i}) \cap \Pi^{l_i=1}$$

where $\Pi^{u_i=\infty}$ (resp. $\Pi^{l_i=1}$) is the rule obtained by replacing u_i by ∞ (resp. l_i by 1). This rule is easily adaptable to several types of assignments and to consider the relaxation for only the end or only the beginning of a schedule.

Example (NSP). "Working shifts have to be in stretches of at least 4 assuming an infinite number of working shifts were allocated at the end of the previous planning". This rule can be modeled through the ∞-side variant, allowing the schedule to start with any number of working shifts.

Forbidden pattern: no subsequence of size $k \in \mathbb{N}^*$ can respect a pattern rule $\Pi^{\mathcal{P}}$,

$$\texttt{windows}(\Pi^{\mathcal{P}}, k, 0, 0).$$

More complex variants can be formalized by doing the union of different pattern rules and/or by intersecting such rules for different values of k.

Example (NSP). The rule "A *Night* shift cannot be followed by an *Early* shift" can be modeled using $\Pi^{\mathcal{P}}\texttt{pattern}(\langle N, E, \{O, L\}\rangle, [NE], \langle 1, 1\rangle, \langle\infty, \infty\rangle)$ and $k = 2$.

3 Multi-level rules

Multi-level rules use intervals $\mathcal{I}$ as a first level time basis and then express some regulations on a larger time basis. For example, in a NSP, some regulations may constrain the number of weeks (i.e., second level) containing a certain number of days-off (i.e., first level). This last example would use a periodic basis of seven days and we call such rules *periodic rules*. On the contrary, *sliding rules* consider a rolling basis as a second level. For example, a MASSP may ensure that every sub-sequence of 10 consecutive symbols (i.e., second level) does not contain more than a certain number of a given activity (i.e., first level).

These rules usually use an horizontal rule, hereafter denoted by Π as a first level and use a k-intervals basis with $k \geq 2 \in \mathbb{N}$ on which they express the second level rule.

3.1 Periodic rules

Periodic rule: the rule Π is applied periodically on a k interval basis,

$$\texttt{periodic}(\Pi, \Pi^{\mathcal{B}}, k)$$

where $\Pi^{\mathcal{B}}$ denotes an automaton whose language is a set of words that contain only the symbol 1 (e.g., $\texttt{cardinality}(\langle 0, 1\rangle, \langle 0, 0\rangle, \langle 0, \infty\rangle))$.

Example (NSP). The "Rule Π must be respected every week" is modeled with $k = 7$.

Periodic rule (cardinality variant) a rule Π is applied at least $l \in \mathbb{N}$ and at most $u \in \mathbb{N}^*$ periods on a k interval basis,

$$\texttt{periodic}(\Pi, \Pi^{\mathcal{B}}, k)$$

where $\Pi^{\mathcal{B}}$ is the cardinality rule $\texttt{cardinality}(\langle 0, 1\rangle, \langle 0, l\rangle, \langle\infty, u\rangle)$. Such a rule can easily be extended to also constrain the number of periods where Π is not respected.

Example (MASSP). With $(l, u, k) = (1, \infty, 48)$ allows to guarantee that "at least one half a day has to respect the rule Π".

Periodic rule (stretch variant) a rule Π is applied to consecutive periods of k intervals by blocks of at least $l \in \mathbb{N}^*$ and at most $u \geq l \in \mathbb{N} \cup \{\infty\}$,

$$\texttt{periodic}(\Pi, \Pi^{\mathcal{B}}, k)$$

where $\Pi^{\mathcal{B}}$ is the stretch rule $\texttt{stretch}(\langle 0,1\rangle, \langle 0,l\rangle, \langle \infty, u\rangle)$. Such a rule can easily be extended to also constrain the number of consecutive periods where Π is not respected.

Example (NSP). "If a week respect the rule Π, so must do the two next ones" is modeled with $k = 7$, $l = 3, u = \infty$.

Periodic rule (slided-cardinality variant) a rule Π is respected at least $l \in \mathbb{N}$ and at most $u \geq l \in \mathbb{N} \cup \{\infty\}$ periods of k intervals for each sequence of $k' \geq 2 \in \mathbb{N}$ consecutive periods,

$$\texttt{periodic}(\Pi, \texttt{windows}(\overline{\Pi^{\mathcal{B}}}, k', 0, 0), k)$$

where $\Pi^{\mathcal{B}}$ is the cardinality rule $\texttt{cardinality}(\langle 0,1\rangle, \langle 0,l\rangle, \langle \infty, u\rangle)$.

Example (MASSP). "Over five rolling days, at least two days must satisfy the rule Π" this rules uses $k = 96$, $k' = 5$, $l = 2$ and $u = \infty$.

Periodic rule (partial variant) each period of k intervals, when keeping only the ith assignment of this period if there is $(q,r) \in \mathbb{N} \times \{0,\ldots,k-1\}$ such that $i = kq + r$ and $b_r = 1$ where $b_0 \ldots b_{k-1} \in \{0,1\}^k$, must respect the rule Π,

$$\texttt{mask}(\texttt{periodic}(\Pi, \Pi^{\mathcal{B}}, \sum_{i=0}^{k-1} b_i), b_0 \ldots b_{k-1})$$

where $\Pi^{\mathcal{B}}$ denotes an automaton whose language is a set of words that contain only the symbol 1 (e.g., $\texttt{cardinality}(\langle 0,1\rangle, \langle 0,0\rangle, \langle 0,\infty\rangle)$). Such rule can be enriched by using any of binary automaton instead of $\Pi^{\mathcal{B}}$ (e.g., the ones presented before in this section).

Example (NSP). "Every weekend must respect the rule Π" is build with $k = 7$ and the binary sequence 0000011.

3.2 Sliding rules

Sliding rule the rule Π is applied on every sub-sequence of k consecutive status,

$$\texttt{windows}(\overline{\Pi}, k, 0, 0).$$

Example (MASSP). "Over each rolling hour, the rule Π must be respected" is a sliding rule with $k = 4$.

Sliding rule (cardinality variant) the rule Π must be respected between $l \in \mathbb{N}$ and $u \geq l \in \mathbb{N} \cup \{\infty\}$ times over all sub-sequences of k consecutive assignments,

$$\texttt{windows}(\Pi, k, l, u).$$

Example (NSP). "5 consecutive days without a day-off is authorized at most twice" use a cardinality rule $\Pi = \texttt{cardinality}(\langle O, \Sigma \backslash \{O\}\rangle, \langle 1,0\rangle, \langle \infty,\infty\rangle)$, $k = 5$, $l = 0$ and $u = 2$.

4 Logical rules

They allow to enforce logical links between a set of rules. For instance, "if Π is respected, then Π' is not mandatory". The classical operations on regular languages (i.e., intersection, union and complementation) are usually sufficient to model these rules. We proposes two examples implying links between two rules Π and Π'.

Exclusive or rule either Π is respected or Π' is respected but not both,

$$(\Pi \cup \Pi') \cap (\overline{\Pi \cap \Pi'})$$

which can be generalized to a finite set of rules of size three or more.

Conditional rule if Π is respected then Π' must also be respected,

$$(\Pi \cap \Pi') \cup \overline{\Pi}$$

which can be generalized to a finite set of rules of size three or more.

Example (MASSP). "If there are more than 16 intervals of working activities, then there must be at least 4 intervals assigned to a *Break*". This is an example of a conditional rule.