Air Traffic Controller Shift Scheduling by Reduction to CSP, SAT and SAT-related Problems*

Mirko Stojadinović

Faculty of Mathematics
University of Belgrade
mirkos@matf.bg.ac.rs

1 Encodings of the Problem

Time slots can be of any fixed specified length but we assume the length of time slot is 1 hour. Shift schedules are generated for a period of one month and for each month, a new shift schedule needs to be generated. There are many reasons for this: expected monthly traffic intensity changes, different controllers use vacation days in different months, etc. We assume that there are only two types of positions: a tower and a terminal, and that there are no night shifts. These assumptions are not a limitation since the encodings can be easily extended to support more types of positions and night shifts.

Let us assume that the days are $1, \dots, n_d$, the controllers are $1, \dots, n_c$ and the shifts are $1, \dots, n_s$. In order to make the encodings compact and more efficient, we assume that working shifts are $1, \dots, n_s - 2$, that $res = n_s - 1$ is rest shift and $vac = n_s$ is vacation shift. Time slots take values $0, \dots, 23$ and for each shift s, the first (s_f) and the last (s_l) working hour of that shift are fixed.

We experimented with different encodings and constraints. In the following 3 subsections, descriptions of 3 encodings which showed good results are described. All 3 encodings are *arc consistent* [2], but other forms of consistencies are not guarantied (e.g. node or path consistency). This means that in the beginning of the solving, propagation cannot deduce anything by using binary constraints, but can use constraints of other arities for deduction. Only the values of variables that determine controllers shifts for each day and controllers positions for each time slot are used when making tabular schedule for employees. Other variables are auxiliary and they are used to improve the readability and the efficiency of the encodings. The fact that vacation shifts and vacation working hours are fixed for the period is used to make the encodings more compact. Due to space limit, the descriptions of some constraints are omitted[1]. In subsection **??** we describe how optimization instances are solved.

* This work was partially supported by the Serbian Ministry of Science grant 174021.

[1] The omitted variable relationships and the constraints of the encodings are available online from: `http://jason.matf.bg.ac.rs/~mirkos/Atco.html`

1.1 The first encoding

Linear arithmetic constraints and global constraint *count* [1] are imposed on integer variables. The *count* constraint requires that the number of occurrences of the value of the expression e in the set of expressions $e_1, \ldots, e_k$ is in some arithmetic relation ($=, \neq, \leq, <, \geq, >$) with the expression n. E.g., $count(\{x_1, x_2, x_3, x_4\},\ 5) > 3$ (where $e = 5$, $e_i = x_i$, the relation is $>$, and $n = 3$) specifies that the value 5 occurs more than 3 times in the set of variables $\{x_1, x_2, x_3, x_4\}$.

Integer variables.

- $dc_{d,c}$: on the day d the controller[2] c can be assigned any from the possible shifts $1, \ldots, n_s$. Note that the fact that the working shifts are $1, \ldots, n_s - 2$ allows us to state that the controller c is working in a facility on the day d by imposing constraint $dc_{d,c} \leq n_s - 2$.
- $dhc_{d,h,c}$: in the hour h of the day d the controller c can be assigned different tasks: c can be on position on tower ($TOW = 0$) or terminal ($TER = 1$), c can have a break hour in a facility ($B = 2$), a vacation hour ($V = 3$) or a rest time ($R = 4$). Note that this allows us to state that c is having working hour in the facility in the hour h of the day d by imposing constraint $dhc_{d,h,c} \leq B$.
- $h_{d,c}$: on the day d the controller c is counted a certain number of working hours $(0, \ldots, 12)$.

Variable relationships.

- If the controller c takes the working shift s on the day d, then c works on working hours of that shift and rests during other hours of the day. So, for any $j \in \{s_f, \ldots, s_l\}$: $dc_{d,c} = s \rightarrow dhc_{d,j,c} \leq B$. For any $j \in \{0, \ldots, 23\} \setminus \{s_f, \ldots, s_l\}$: $dc_{d,c} = s \rightarrow dhc_{d,j,c} = R$.
- If the controller c takes rest shift on the day d, then c has rest time during all hours of the day. So, for any $j \in \{0, \ldots, 23\}$: $dc_{d,c} = res \rightarrow dhc_{d,j,c} = R$.
- If the controller c works in a facility in the hour h of the day d, then c takes a working shift on that day: $dhc_{d,h,c} \leq B \rightarrow dc_{d,c} \leq n_s - 2$.
- If the controller c takes the non-rest shift s with working hours $s_f, \ldots, s_l$ on the day d, then this implies c is working $s_l - s_f + 1$ hours on that day: $dc_{d,c} = s \rightarrow h_{d,c} = s_l - s_f + 1$. In case of the rest shift: $dc_{d,c} = res \rightarrow h_{d,c} = 0$.

Hard constraints.

- *Assigning shifts:* On the day d the controller c takes one of the possible shifts (already imposed as variable $dc_{d,c}$ takes one from the values $1, \ldots, n_s$).
- *Vacation shifts:* If the controller c is allowed to take vacation shift vac on the day d, then $dc_{d,c} = vac$. This also implies a certain vacation hours and rest time during other hours of that day. So, for any $j \in \{vac_f, \ldots, vac_l\}$: $dhc_{d,j,c} = V$. For any $j \in \{0, \ldots, 23\} \setminus \{vac_f, \ldots, vac_l\}$: $dhc_{d,j,c} = R$. If the controller c does not take vacation shift on the day d, then $dc_{d,c} \neq vac$ and for any $j \in \{0, \ldots, 23\}$: $dhc_{d,j,c} \neq V$.

---

[2] Most of the constraints have to be true for all controllers, but we use some fixed controller c in the descriptions. Similarly for days, hours and shifts.

- *Consecutive working shifts:* The controller c must take at least one rest shift in cws days in a row starting from the day d, whenever c does not take vacation shift on any of these days: $count(\{dc_{d,c}, \ldots, dc_{d+cws,c}\},\ res) \geq 1$.
- *Consecutive rest shifts:* The controller c must not take rest shifts more than crs days in a row starting from the day d: $count(\{dc_{d,c}, \ldots, dc_{d+crs,c}\},\ res) \leq y$.
- *Heads of the shifts:* On the day d and a facility working hour h one of controllers that have licence to be the head of the working shift has to be in the facility. Let $c_1, \ldots, c_p$ be all of these controllers that do not have vacation hour in this time slot. One from them does not take rest hour: $count(\{dhc_{d,h,c_1}, \ldots, dhc_{d,h,c_p}\},\ R) \neq p$.
- *Minimum rest shifts:* The controller c must take at least the minimum number m of rest shifts per month: $count(\{dc_{1,c}, \ldots, dc_{n_d,c}\},\ res) \geq m$.
- *Rest time between shifts:* If the controller c takes a shift s_i on the day d, then there are some shifts that c must not take on day $d+1$, as c needs a specified number of rest hours. For each such shift s_j the following constraint is imposed: $dc_{d,c} = s_i \rightarrow dc_{d+1,c} \neq s_j$.
- *Two day working hour constraints:* On the day d and the following day the controller c must not in sum work more than some specified number of hours q (e.g. $q = 22$): $h_{d,c} + h_{d+1,c} \leq q$.
- *Maximum working hours:* The controller c must not work more than the specified max working hours in a month: $\sum_{d=1}^{n_d} h_{d,c} \leq max$.
- *Minimum working hours:* The controller c has to work at least the specified min working hours in a month in order to get full wage: $\sum_{d=1}^{n_d} h_{d,c} \geq min$.
- *One position per time slot:* The controller c in the hour h of the day d can work on at most one place: c can be on position on tower or on position on terminal or have break hour in a facility or have a working hour in the vacation shift. No constraints need to be imposed as variable $dhc_{d,h,c}$ can take exactly one from the values $0, \ldots, 4$.
- *Positions filled:* If in the hour h of the day d at least k controllers are needed for tower position, then the following constraint is imposed: $count(\{dhc_{d,h,1}, \ldots, dhc_{d,h,n_c}\},\ TOW) \geq k$. Analogously for terminal position[3].
- *Consecutive time slots on position:* On the day d starting from the hour h the controller c is on position not more than the specified number m of hours in a row: $dhc_{d,h,c} \geq B \vee \ldots \vee dhc_{d,h+m,c} \geq B$.
- *Licences:* If the controller c does not have a licence to be on position on tower, then for each day d and hour h: $dhc_{d,h,c} \neq TOW$. Analogously for terminal position.

Soft constraints. Each wish of a controller is expressed as a constraint that is true iff the wish is not satisfied. Each of these constraints is made equivalent to a fresh integer variable with the domain $\{0, 1\}$. If all these variables take value 0, then all wishes are satisfied. We denote each fresh variable with $x_{c,i}$, where for each fixed controller c index i takes different values.

[3] Actually, several *count* constraints are replaced by one *global cardinality* constraint [1] in order to obtain stronger filtering, but we skip presenting the details.

- *Shift preferences:* If the controller c prefers working shifts $s_1, \ldots, s_z$, then each shift s different from these shifts, rest or vacation shift is considered undesirable on any day d: $x_{c,i} \leftrightarrow dc_{d,c} = s$. E.g., if the month has 30 days and if the smallest unused non-negative index of the variables associated with the controller c is j, then 30 variables $x_{c,j}, \ldots, x_{c,j+29}$ are introduced.
- *Minimize consecutive working shifts:* The controller c prefers to take consecutive working shifts as rarely as possible. For each day d: $x_{c,i} \leftrightarrow dc_{d,c} \leq n_s - 2 \wedge dc_{d+1,c} \leq n_s - 2$.
- *Maximize consecutive rest shifts:* The controller c does not prefer isolated rest shifts. For each three consecutive days $d, d+1, d+2$, when c does not take vacation shift on any of these days: $x_{c,i} \leftrightarrow dc_{d,c} \leq n_s - 2 \wedge dc_{d+1,c} = res \wedge dc_{d+2,c} \leq n_s - 2$.

1.2 The second encoding

As the syntax of some solvers does not allow the usage of global constraints, we adapt the first encoding not to use these constraints. *Integer variables*, *variable relationships* and *soft constraints* are the same as in the first encoding.

Hard constraints. Most of the hard constraints are the same as in the first encoding, so we only describe the constraints which are encoded differently. New variables and constraints specifying their relationships are introduced for if-then-else expressions, so this encoding can be of much greater size than the first one.

- *Consecutive working shifts:* The controller c must take at least one rest shift in cws days in a row starting from the day d, whenever c does not take vacation shift on any of these days: $dc_{d,c} = res \vee \ldots \vee dc_{d+cws,c} = res$.
- *Consecutive rest shifts:* The controller c must not take rest shifts more than crs days in a row starting from the day d: $dc_{d,c} \neq res \vee \ldots \vee dc_{d+crs,c} \neq res$.
- *Heads of the shifts:* On the day d and the facility working hour h one of controllers $c_1, \ldots, c_p$ that have licence to be the head of the working shift has to be in the facility: $dhc_{d,h,c_1} \leq B \vee \ldots \vee dhc_{d,h,c_p} \leq B$.
- *Minimum rest shifts:* The controller c must take at least a minimum number of rest shifts m per month: $\sum_{d=1}^{n_d}(\text{if } (dc_{d,c} = res) \text{ then } 1 \text{ else } 0) \geq m$.
- *Positions filled:* If in the hour h of the day d at least k controllers are needed for tower position, then the following constraint is imposed: $\sum_{c=1}^{n_c}(\text{if } (dhc_{d,h,c} = TOW) \text{ then } 1 \text{ else } 0) \geq k$. Analogously for terminal position.

1.3 The third encoding

If $l_1, \ldots, l_n$ are Boolean literals, then the formula $l_1 + \ldots + l_n \ \# \ k, \quad k \in \mathbb{N}, \quad \# \in \{\leq, <, \geq, >, =\}$ is called *Boolean cardinality constraint* (BCC) [3]. In our presentation of the constraints we use equivalences, implications and clauses as often as possible in order to improve the readability of the paper, but the third encoding actually uses BCCs only. Each equivalence can be converted to 2 implications (from left to right and vice versa). The implication $a \to b$ can be directly translated to a clause $\neg a \vee b$ and more complicated implications can be translated to clauses by using De Morgans laws and distributivity rules[4]. Note that each clause $l_1 \vee \ldots \vee l_n$ is actually BCC $l_1 + \ldots + l_n \geq 1$.

Propositional variables.

- $dcs_{d,c,s}$: on the day d the controller c takes the shift s.
- $dc_{d,c}$: on the day d the controller c takes a working shift.
- $dhc_{d,h,c}$: the hour h of the day d for the controller c is a working hour (in a facility or on vacation). If false, c is having rest hour.
- $tow_{d,h,c}/ter_{d,h,c}/pos_{d,h,c}$: in the hour h of the day d the controller c is on position on tower/on position on terminal/on any position.
- $b_{d,h,c}/v_{d,h,c}$: in the hour h of the day d the controller c has break hour in a facility/has vacation working hour.
- $tow_{d,h,c}$: in the hour h of the day d the controller c is on position on tower.
- $ter_{d,h,c}$: in the hour h of the day d the controller c is on position on terminal.
- $pos_{d,h,c}$: in the hour h of the day d the controller c is on position (on tower or terminal).
- $b_{d,h,c}$: in the hour h of the day d the controller c has break hour in a facility.
- $v_{d,h,c}$: in the hour h of the day d the controller c takes vacation working hour.

Variable relationships.

- If the controller c takes the working shift s on the day d, then c works on working hours of that shift and does not work during other hours of the day. So, for any $j \in \{s_f, \ldots, s_l\}$: $dcs_{d,c,s} \to dhc_{d,j,c}$. For any $j \in \{0, \ldots, 23\} \setminus \{s_f, \ldots, s_l\}$: $dcs_{d,c,s} \to \neg dhc_{d,j,c}$.
- If the controller c takes rest shift on the day d, then c does not work in any hour of that day. So, for any $j \in \{0, \ldots, 23\}$: $dcs_{d,c,res} \to \neg dhc_{d,j,c}$.
- If the controller c takes vacation shift on the day d, then this implies certain working hours in vacation shift and non-working time during other hours of that day. So, for any $j \in \{vac_f, \ldots, vac_l\}$: $dcs_{d,c,vac} \to v_{d,j,c}$. For any $j \in \{0, \ldots, 23\} \setminus \{vac_f, \ldots, vac_l\}$: $dcs_{d,c,vac} \to \neg dhc_{d,j,c}$.
- The controller c works in the hour h of the day d iff c is on position on tower or on position on terminal or has break hour in a facility or has vacation working hour: $dhc_{d,h,c} \leftrightarrow tow_{d,h,c} \vee ter_{d,h,c} \vee b_{d,h,c} \vee v_{d,h,c}$.
- The controller c is on position in the hour h of the day d iff c is on position on tower or on terminal: $pos_{d,h,c} \leftrightarrow tow_{d,h,c} \vee ter_{d,h,c}$.
- The controller c takes working shift on the day d iff c takes neither vacation nor rest shift on that day: $dc_{d,c} \leftrightarrow \neg dcs_{d,c,vac} \wedge \neg dcs_{d,c,res}$,

[4] There is no risk of exponential blow-up as implications in this encoding have small number of literals.

Hard constraints.

- *Assigning shifts:* On the day d the controller c takes one of the possible shifts (any of working shifts, rest or vacation shift): $dcs_{d,c,1} + \ldots + dcs_{d,c,n_s} = 1$.
- *Vacation shifts:* If the controller c is allowed to take vacation shift on the day d, then $dcs_{d,c,vac} = 1$. This also implies a certain vacation hours and rest time during other hours of that day. So, for any $j \in \{vac_f, \ldots, vac_l\}$: $v_{d,j,c} = 1$. For any $j \in \{0, \ldots, 23\} \setminus \{vac_f, \ldots, vac_l\}$: $dhc_{d,j,c} = 0$. If the controller c does not take vacation shift on the day d, then $dcs_{d,c,vac} = 0$ and for any $j \in \{0, \ldots, 23\}$: $v_{d,j,c} = 0$.
- *Consecutive working shifts:* The controller c must not work in a facility more than cws days in a row starting from the day d, whenever c does not take vacation shift on any of these days: $\neg dc_{d,c} \vee \ldots \vee \neg dc_{d+cws,c}$.
- *Consecutive rest shifts:* The controller c must not take rest shifts more than crs consecutive days starting from the day d: $\neg dcs_{d,c,res} \vee \ldots \vee \neg dcs_{d+crs,c,res}$.
- *Heads of the shifts:* On the day d and the facility working hour h one of controllers that have licence to be the head of the working shift has to be in the facility. Let $c_1, \ldots, c_p$ be all of these controllers that do not have vacation hour in this time slot. One from them has to be in the facility: $tow_{d,h,c_1} \vee ter_{d,h,c_1} \vee b_{d,h,c_1} \vee \ldots \vee tow_{d,h,c_p} \vee ter_{d,h,c_p} \vee b_{d,h,c_p}$.
- *Minimum rest shifts:* The controller c must take at least the specified number m of rest shifts per month: $dcs_{1,c,res} + \ldots + dcs_{n_d,c,res} \geq m$
- *Rest time between shifts:* If the controller c takes a shift s_i on the day d, then there are some shifts c must not work on day $d+1$, as c needs a specified number of rest hours. For each such shift s_j the following constraint is imposed: $dcs_{d,c,s_i} \rightarrow \neg dcs_{d+1,c,s_j}$.
- *Two day working hour constraints:* If the controller c takes the shift s_i on the day d, then there are some shifts that c must not take on day $d+1$, as c must not work more than a specified number of working hours in a 2 day period. For each such shift s_j the following constraint is imposed: $dcs_{d,c,s_i} \rightarrow \neg dcs_{d+1,c,s_j}$.
- *Maximum working hours:* The controller c must not work more than the specified max working hours in a month: $\sum_{d=1}^{n_d} \sum_{h=0}^{23} dhc_{d,h,c} \leq max$.
- *Minimum working hours:* The controller c has to work at least the specified min working hours in a month in order to get full wage: $\sum_{d=1}^{n_d} \sum_{h=0}^{23} dhc_{d,h,c} \geq min$.
- *One position per time slot:* The controller c in the hour h of the day d can work on at most one place: c can be on position on tower or on position on terminal or have break hour in a facility or have a working hour in the vacation shift: $tow_{d,h,c} + ter_{d,h,c} + b_{d,h,c} + v_{d,h,c} \leq 1$.
- *Positions filled:* If in the hour h of the day d at least k controllers are needed for tower position, then the following constraint is imposed: $\sum_{c=1}^{n_c} tow_{d,h,c} \geq k$. Analogously for terminal position.
- *Consecutive time slots on position:* On the day d starting from the hour h the controller c is on position not more than the specified number m of hours in a row: $\neg pos_{d,h,c} \vee \ldots \vee \neg pos_{d,h+m,c}$.

– *Licences:* If the controller c does not have a licence to be on position on tower, then for each day d and hour h: $\neg tow_{d,h,c}$. Analogously for terminal position.

Soft constraints. Fresh Boolean variables are introduced in the same way as integer variables with the domain $\{0,1\}$ in the first encoding.

– *Shift preferences:* If the controller c prefers working shifts $s_1, \ldots, s_z$, then any other shift s which is different from these shifts, rest or vacation shift is considered undesirable on any day d: $x_{c,i} \leftrightarrow dcs_{d,c,s} = 1$.
– *Minimize consecutive working shifts:* The controller c prefers to take consecutive working shifts as rarely as possible. For each day d: $x_{c,i} \leftrightarrow dc_{d,c} \wedge dc_{d+1,c}$.
– *Maximize consecutive rest shifts:* The controller c dose not prefer isolated rest shifts. For each three consecutive days $d, d+1, d+2$, when c does not take vacation shift on any of these days: $x_{c,i} \leftrightarrow dc_{d,c} \wedge dcs_{d+1,c,res} \wedge dc_{d+2,c}$.

References

1. Nicolas Beldiceanu, Mats Carlsson, and Jean-Xavier Rampon. Global constraint catalog. Technical report, SICS, 2005.
2. Christian Bessiere. Arc-consistency and arc-consistency again. *Artificial intelligence*, 65(1):179–190, 1994.
3. Carsten Sinz. Towards an optimal cnf encoding of boolean cardinality constraints. In Peter van Beek, editor, *CP*, volume 3709 of *Lecture Notes in Computer Science*, pages 827–831. Springer, 2005.