

# Relazione elaborato AI

Nocentini Federico

8 febbraio 2019

## 1 Introduzione

In questa relazione andremo a spiegare il lavoro svolto nella modellizzazione di un CSP attraverso l'ambiente sviluppo MiniZinc. Lo scenario é il seguente: determinare i turni di servizio di un insieme di  $n$  infermieri che lavorano in un ipotetico reparto ospedaliero. Ciascun infermiere deve garantire almeno  $k$  ore settimanali per non più di  $h$  ore al giorno, comprendendo nelle  $k$  ore un turno notturno di 8 ore. Chi effettua il turno notturno ha diritto al riposo per i due giorni successivi. In ogni momento del giorno devono essere presenti in reparto almeno  $D$  infermieri ed in ogni momento della notte ne devono essere presenti almeno  $N$ .

## 2 Interpretazione

Il modello é stato pensato per un orario settimanale in cui il numero di turni diurni nell'arco delle 16h del giorno é definito nel seguente modo:

$$nTurni = \begin{cases} 2 & \text{se } h \geq 8 \\ 3 & \text{se } h < 8 \end{cases};$$

Questo tipo di definizione é dovuta al fatto che se le ore lavorative massime in un giorno sono  $< 8$  per coprire le 16h ci vogliono 3 turni altrimenti ne bastano 2. Il numero di turni notturni é 1 e il numero di turni di riposo é 2. Quindi se ci sono 2 turni diurni saranno: MATTINA e SERA, se i turni diurni sono 3 saranno: MATTINA, POMERIGGIO e SERA. Il modello genera in output una tabella che rappresenta i turni di servizio.

## 3 Modellizzazione

I dati sono:

- Numero di infermieri:  $n$ ;
- Numero minimo infermieri per turno diurno:  $D$ ;
- Numero minimo infermieri per turno notturno:  $N$ ;
- Numero minimo di ore settimanali:  $k$ ;
- Numero massimo di ore per giorno:  $h$ ;
- Numero di giorni:  $nGiorni=7$ ;
- Numero di turni in cui sono divise le 16h del giorno:  $nTurni$ ;

Le variabili sono memorizzate in una matrice  $T \in M(nx7)$  e hanno un dominio che dipende dal numero di turni:

$$D = \begin{cases} \{1, 2, 3, 4\} & \text{se } nTurni = 2 \text{ (NOTTE, RIPOSO, MATTINA, SERA)} \\ \{1, 2, 3, 4, 5\} & \text{se } nTurni = 3 \text{ (NOTTE, RIPOSO, MATTINA, SERA, POMERIGGIO)} \end{cases}$$

Per definire i vincoli ho bisogno di definire:

- Un intervallo di valori  $E$  che dipende dalla cardinalità del dominio delle variabili:

$$E = \begin{cases} [3, 4] & \text{se } nTurni = 2 \\ [3, 4, 5] & \text{se } nTurni = 3 \end{cases};$$

Che serve per ciclare su i tipi di turno nell'arco delle 16h diurne che possono essere MATTINA e SERA oppure MATTINA,SERA e POMERIGGIO.

Il solver deve costruire la matrice dei turni rispettando i seguenti vincoli:

1.  $h * 4 + 8 \geq k$   
Per garantire almeno  $k$  ore settimanali e non sfiorare le  $h$  ore per giorno.
2.  $(16/nTurni) * 4 + 8 \geq k$   
Per garantire almeno  $k$  ore settimanali.
3.  $\forall k \in [1, n](\sum_{i=1}^7 1 \{T_{k,i} = 1\} = 1)$   
In una settimana un infermiere deve fare solo un turno notturno.
4.  $\forall k \in [1, n](\sum_{i=1}^7 1 \{T_{k,i} = 2\} = 2)$   
In una settimana un infermiere deve fare 2 giorni di riposo.
5.  $\forall j \in E(\forall k \in [1, n](\sum_{i=1}^7 1 \{T_{k,i} = j\} \geq 1))$   
In una settimana un infermiere deve fare 4 turni diurni. Ogni tipo di turno deve essere effettuato almeno una volta
6.  $\forall k \in [1, n]\forall i \in [1, 5](T_{k,i} = 1 \Rightarrow T_{k,i+1} = 2)$   
Dopo un turno notturno primo giorno di riposo.(Per i primi 5 giorni della settimana)
7.  $\forall k \in [1, n]\forall i \in [1, 5](T_{k,i} = 1 \Rightarrow T_{k,i+2} = 2)$   
Dopo un turno notturno secondo giorno di riposo.(Per i primi 5 giorni della settimana)
8.  $\forall k \in [1, n](T_{k,7} = 1 \Rightarrow T_{k,1} = 2)$
9.  $\forall k \in [1, n](T_{k,7} = 1 \Rightarrow T_{k,2} = 2)$
10.  $\forall k \in [1, n](T_{k,6} = 1 \Rightarrow T_{k,7} = 2)$
11.  $\forall k \in [1, n](T_{k,6} = 1 \Rightarrow T_{k,1} = 2)$   
Dopo un turno di notte 2 giorni di riposo.(Per sabato e domenica)
12.  $\forall i \in [1, 7](\sum_{k=1}^n 1 \{T_{k,i} = 1\} \geq N)$   
Per ogni giorno ci devono essere almeno  $N$  infermieri che fanno il turno di notte.
13.  $\forall i \in [1, 7](\sum_{k=1}^n 1 \{T_{k,i} = 2\} \geq N * 2)$   
Per ogni giorno ci devono essere almeno  $N*2$  infermieri che riposano, questo é dovuto al fatto che dopo una notte 2 giorni di riposo.
14.  $\forall j \in E(\forall i \in [1, 7](\sum_{k=1}^n 1 \{T_{k,i} = j\} \geq D))$   
Per ogni tipo di turno diurno ci devono essere almeno  $D$  infermieri che lavorano.

## 4 Implementazione

La matrice T sopra definita é stata implementata come un array 2d e chiamata *tabella*. Ai valori del dominio delle variabili sono stati attribuiti dei nomi per rendere piú semplice la lettura e l'implementazione:

- notte=1;
- riposo=2;
- mattina=3.
- sera=4.
- pomeriggio=5.

Per l'implementazione sono stati usati:

- INFERMIERI: set di interi che mi rappresenta gli infermieri;
- GIORNI: set di interi che mi rappresenta i giorni;
- EFFETTIVI: rappresenta l'intervallo E, é stato implementato come un set di interi;

Per l'implementazione dei vincoli sopra citati in Minizinc sono stati usati i seguenti comandi:

- *bool2int(x)* che é definito nel seguente modo:  $bool2int(x) = \begin{cases} 1 & \text{se } x \text{ TRUE} \\ 0 & \text{se } x \text{ FALSE} \end{cases}$ ;
- *forall(1..n)* che serve per ciclare;
- *sum(1..k)* che serve per fare le sommatorie;

L' implementazione dei vincoli, in ordine rispetto alla formulazione matematica, é la seguente:

1. *constraint*  $h * 4 + 8 \geq k$ ;
2. *constraint*  $(16/nturni) * 4 + 8 \geq k$ ;
3. *constraint* forall(k in INFERMIERI)(sum(i in GIORNI)(bool2int(tabella[k,i]==notte))=1);
4. *constraint* forall(k in INFERMIERI)(sum(i in GIORNI)(bool2int(tabella[k,i]==riposo))=2);
5. *constraint* forall(j in EFFETTIVI)(forall(k in INFERMIERI)(sum(i in GIORNI)(bool2int(tabella[k,i]==j))≥1));
6. *constraint* forall(k in INFERMIERI)(forall(i in GIORNI diff nGiorni-1..nGiorni)(tabella[k,i]==notte ⇒ tabella[k,i+1]==riposo));
7. *constraint* forall(k in INFERMIERI)(forall(i in GIORNI diff nGiorni-1..nGiorni)(tabella[k,i]==notte ⇒ tabella[k,i+2]==riposo));
8. *constraint* forall(k in INFERMIERI)((tabella[k,7]==notte⇒tabella[k,1]==riposo));
9. *constraint* forall(k in INFERMIERI)((tabella[k,7]==notte⇒tabella[k,2]==riposo));
10. *constraint* forall(k in INFERMIERI)((tabella[k,6]==notte⇒tabella[k,7]==riposo));
11. *constraint* forall(k in INFERMIERI)((tabella[k,6]==notte⇒tabella[k,1]==riposo));

12. *constraint* forall(i in GIORNI)(sum(k in INFERMIERI)(bool2int(tabella[k,i]==notte))>=N);
13. *constraint* forall(i in GIORNI)(sum(k in INFERMIERI)(bool2int(tabella[k,i]==riposo))>=N\*2);
14. *constraint* forall(j in EFFETTIVI)(forall(i in GIORNI)(sum(k in INFERMIERI)(bool2int(tabella[k,i]==j))>=D)));

## 5 Output

Per rendere la tabella degli orari leggibile si é utilizzato il comando *fix* che invece di stampare i numeri nella tabella stampa delle stringhe memorizzate in un array, in particolare invece di stampare il numero *i* stampa l'*i*-esima stringa dell'array. Per esempio se in una determinata cella della tabella si trova il numero 2 allora *fix* ci sostituisce RIPOSO. Nella tabella le colonne rappresentano i giorni, le righe rappresentano gli infermieri. Si può scegliere il tipo di rappresentazione della tabella commentando rispettivamente la riga 21 o la riga 22.