

# Relazione elaborato AI

Nocentini Federico

6 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 viene specificato se il turno di un determinato infermiere in un determinato giorno é: GIORNO, NOTTE o RIPOSO. Nei 7 giorni settimanali sono previsti 4 giorni in cui si lavora di giorno, 1 in cui si lavora di notte e 2 in cui si riposa. Per rendere lo scenario più realistico 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 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  $D = \{1, 2, 3\}$  che mi rappresenta rispettivamente NOTTE, GIORNO e RIPOSO. Il solver deve costruire la tabella dei turni rispettando i seguenti vincoli:

- $h * 4 + 8 \geq k$ ;  
Per garantire almeno  $k$  ore settimanali e non sforare le  $h$  ore per giorno.

- $\forall k \in [1, n](\sum_{i=1, t[k, i]=1}^7 t[k, i] = 1 * 1)$   
In una settimana un infermiere deve fare solo un turno notturno.
- $\forall k \in [1, n](\sum_{i=1, t[k, i]=3}^7 t[k, i] = 2 * 3)$   
In una settimana un infermiere deve fare 2 giorni di riposo.
- $\forall k \in [1, n](\sum_{i=1, t[k, i]=2}^7 t[k, i] = 4 * 2)$   
In una settimana un infermiere deve fare 4 turni diurni.
- $\forall k \in [1, n]\forall i \in [1, 5](t[k, i] = 1 \Rightarrow t[k, i + 1] = 3)$   
Dopo un turno notturno primo giorno di riposo.(Per i primi 5 giorni della settimana)
- $\forall k \in [1, n]\forall i \in [1, 5](t[k, i] = 1 \Rightarrow t[k, i + 2] = 3)$   
Dopo un turno notturno secondo giorno di riposo.(Per i primi 5 giorni della settimana)
- $\forall k \in [1, n](t[k, 7] = 1 \Rightarrow t[k, 1] = 3)$
- $\forall k \in [1, n](t[k, 7] = 1 \Rightarrow t[k, 2] = 3)$
- $\forall k \in [1, n](t[k, 6] = 1 \Rightarrow t[k, 7] = 3)$
- $\forall k \in [1, n](t[k, 6] = 1 \Rightarrow t[k, 1] = 3)$   
Dopo un turno di notte 2 giorni di riposo.(Per sabato e domenica)
- $\forall i \in [1, 7](\sum_{k=1, t[k, i]=1}^n t[k, i] \geq N * 1)$   
Per ogni giorno ci devono essere almeno N infermieri che fanno il turno di notte.
- $\forall i \in [1, 7](\sum_{k=1, t[k, i]=3}^n t[k, i] \geq N * 2 * 3)$   
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.
- $\forall i \in [1, 7](\sum_{k=1, t[k, i]=2}^n t[k, i] \geq D * nTurni * 2)$   
Per ogni giorno ci devono essere almeno D\*nTurni infermieri che fanno il turno diurno.

## 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;
- giorno=2;
- riposo=3.

Per l'implementazione sono stati usati:

- INFERMIERI: set di interi che mi rappresenta gli infermieri;
- GIORNI: set di interi che mi rappresenta i giorni;

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 é la seguente:

- $constraint \ h * 4 + 8 \geq k$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)(sum(i \text{ in } GIORNI)(bool2int(tabella[k,i]==notte))=1)$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)(sum(i \text{ in } GIORNI)(bool2int(tabella[k,i]==riposo))=2)$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)(sum(i \text{ in } GIORNI)(bool2int(tabella[k,i]==giorno))=4)$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)(forall(i \text{ in } GIORNI \text{ diff } nGiorni-1..nGiorni)(tabella[k,i]=notte \Rightarrow tabella[k,i+1]=riposo))$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)(forall(i \text{ in } GIORNI \text{ diff } nGiorni-1..nGiorni)(tabella[k,i]=notte \Rightarrow tabella[k,i+2]=riposo))$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)((tabella[k,7]=notte \Rightarrow tabella[k,1]=riposo))$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)((tabella[k,7]=notte \Rightarrow tabella[k,2]=riposo))$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)((tabella[k,6]=notte \Rightarrow tabella[k,7]=riposo))$ ;
- $constraint \ forall(k \text{ in } INFERMIERI)((tabella[k,6]=notte \Rightarrow tabella[k,1]=riposo))$ ;
- $constraint \ forall(i \text{ in } GIORNI)(sum(k \text{ in } INFERMIERI)(bool2int(tabella[k,i]==notte)) \geq N)$ ;
- $constraint \ forall(i \text{ in } GIORNI)(sum(k \text{ in } INFERMIERI)(bool2int(tabella[k,i]==riposo)) \geq N*2)$ ;
- $constraint \ forall(i \text{ in } GIORNI)(sum(k \text{ in } INFERMIERI)(bool2int(tabella[k,i]==giorno)) \geq D*nTurni)$ ;

## 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 GIORNO.