

E.A.5.9 (Meetings)

1.1 Modellazione

Dati i parametri N, K, M, T, S siano

- $\mathcal{N} = \{1, \dots, N\}$ l'insieme di identificatori dei manager
- $R = \min(\lfloor \frac{N}{K} \rfloor, S)$ il numero massimo di stanze utilizzabili in un turno
- $\mathcal{R} = \{1, \dots, R\}$
- $\mathcal{K} = \{1, \dots, k\}$ l'insieme di posti in una stanza
- $\mathcal{T} = \{1, \dots, T\}$ l'insieme di turni possibili

E sia (X, D, C) l'istanza parametrica di CSP t.c.

$$X = \{X_{t,r,k} \mid t \in \mathcal{T} \wedge r \in \mathcal{R} \wedge k \in \mathcal{K}\} \cup \{Y_t \mid t \in \mathcal{T}\} \quad (1)$$

- $X_{t,r,k}$ è l'id del manager al posto k nella stanza r al turno t
- Y_t è il numero di stanze usate al t -esimo turno

Con i rispettivi domini

$$D = \{D_{X_{t,r,k}} \mid D_{X_{t,r,k}} = \mathcal{N}\} \cup \{D_{Y_t} = \mathcal{R}\} \quad (2)$$

Ed l'insieme di vincoli

$$C = C_M \cup C_{\text{alldiff}} \cup C_{\text{symmetry}} \cup C_{\text{pairs}} \quad (3)$$

La somma delle stanze usate nei vari turni corrisponde al numero totale di incontri

$$C_M = \left\{ \langle \{Y_t \mid t \in \mathcal{T}\}, \sum_{t \in \mathcal{T}} Y_t = M \rangle \right\} \quad (4)$$

(alldifferent) In un dato turno un manager non può stare in due stanze in due posti diversi

$$C = \{\text{alldifferent}(X_{t,r,k}) \mid t \in \mathcal{T}\} \quad (5)$$

(symmetry breaking) In una determinata stanza, l'ordine dei manager non conta, quindi se ne può fissare uno

$$C_{\text{symmetry}} = \left\{ \begin{aligned} &\langle \{X_{t,r,k}, X_{t,r,k+1}\}, X_{t,r,k} < X_{t,r,k+1} \rangle \mid \\ &t \in \mathcal{T} \wedge r \in \mathcal{R} \wedge k \in \mathcal{K} \end{aligned} \right\} \quad (6)$$

Per ogni coppia di manager, ci deve essere un turno e una stanza in cui questi due si incontrano (è un vincolo su tutte le variabili, un po' bruttino... ma fa il suo)

$$\begin{aligned}
C_{\text{pairs}} = \{ & \\
& \langle \{ X_{t,r,k}, X_{t,r,l}, Y_t \mid t \in \mathcal{T} \wedge r \in \mathcal{R} \wedge k, l \in \mathcal{K} \}, \\
& \forall i, j \\
& (i, j \in \mathcal{N} \wedge i < j) \rightarrow \\
& \quad \exists t, r, k, l \\
& \quad t \in \mathcal{T} \wedge r \in \mathcal{R} \wedge k, l \in \mathcal{K} \wedge \\
& \quad k < l \wedge r \leq Y_t \wedge X_{t,r,k} = i \wedge X_{t,r,l} = j \\
& \rangle \mid t \in \mathcal{T} \wedge r \in \mathcal{R} \wedge k, l \in \mathcal{K} \wedge k < l \\
& \}
\end{aligned} \tag{7}$$

1.2 Istanziamento

Istanziamento lasciata al lettore...

1.3 Codifica MiniZinc

```
include "globals.mzn";

int: N = 6;
int: K = 3;
int: M = 8;
int: T = 5;
int: S = 6;

int: R = min(N div K, S);

array[1..T, 1..R, 1..K] of var 1..N: X;
array[1..T] of var 0..R: Y;

constraint sum(t in 1..T)(Y[t]) = M;

constraint forall(t in 1..T)(
    alldifferent([X[t, r, k] | r in 1..R, k in 1..K])
);

constraint forall(t in 1..T, r in 1..R, k in 1..K - 1)(
    X[t, r, k] < X[t, r, k + 1]
);

constraint forall(i in 1..N - 1, j in i + 1..N)(
    exists(t in 1..T, r in 1..R, k in 1..K - 1, l in k + 1..K)(
        X[t, r, k] = i /\ X[t, r, l] = j /\ r ≤ Y[t]
    )
);

output[
    "turn" ++ show(t) ++ ": | " ++ concat(
        [concat([show_int(-2, X[t, r, k]) ++ " " | k in 1..K]) ++ "| " | r in 1..R]
    ) ++ "\n" | t in 1..T
];

output [
    show(Y[t]) ++ " " | t in 1..T
];
```