

# Corsi e docenti in Minizinc

*Massimo Ghinassi*

18 febbraio 2013

## 1 Introduzione

Il modello realizzato implementa un semplice problema di soddisfacimento di vincoli per l'assegnazione di corsi a docenti in un contesto accademico. Una volta lanciato il terminale si procede scrivendo il comando **minizinc** seguito dal file **Curricola.mzn** e dal file **Dati.dzn** desiderato. Se si vogliono far comparire le statistiche, dopo **minizinc** basta digitare **-s** e **-a** se vogliamo visualizzare tutte le possibili soluzioni.

Il modello realizzato è sufficientemente generico per poter risolvere qualsiasi problema.

## 2 Come funziona il modello Minizinc

Il modello sviluppato per il progetto assegnato è molto semplice, sono stati creati due file **.dzn** distinti che contengono i dati di interesse al modello sviluppato, nel file dei dati troviamo il numero dei docenti e il numero dei corsi presi in considerazione della specifica istanza e due array con i relativi nomi dei corsi e dei docenti utili per una migliore visualizzazione dell'output.

Per specificare l'anno accademico e le ore settimanali di ogni corso sono stati creati due array (**ac**, **hc**) uno con i valori 1,2,3 per specificare a quale anno accademico si riferisce un corso e il numero di ore settimanali per ciascun corso nell'altro.

In particolare abbiamo una matrice **insegna** in cui viene specificato se un docente può insegnare uno specifico corso, **insegna[i,j]** contiene 1 se il docente **i** può insegnare il corso **j**, 0 altrimenti.

Nel file **.mzn** è presente il modello usato per risolvere il nostro problema, dove vengo specificati i vincoli richiesti, in particolare viene controllato che un docente non può insegnare più di un corso nello stesso anno curricolare, i docenti possono insegnare solo corsi di cui sono competenti e che il carico complessivo di ciascun docente non sia superiore di due ore a quello di un altro docente. Alla fine abbiamo l'output in cui vengo stampati i corsi e i docenti per ogni anno curricolare, in questa fase viene utilizzato la funzione *fix* per fissare i dati inseriti nell'array orario.

## 2.1 Confronto sui Tempi di esecuzione

La migliore strategia di selezione delle variabili è il **first-fail**, che consente di selezionare la variabile con meno valori possibili nel dominio, per la scelta del valore da assegnare alle variabili invece viene adottata una politica **least constrained value** (LCV) che permette di lasciare più spazio alle variabili vicine.

Minizinc offre diverse strategie di ricerca, secondo una sintassi di questo tipo:

**int\_search(variables, varchoice, constrainchoice, strategy)**

Dove:

1. **variables** è un array di var int.
2. **varchoice** può essere: input-order, first-fail, smallest.
3. **constrainchoice** può essere: indomain-min, indomain-median, indomain-random, indomain-split.
4. **strategy** è sempre complete.

I problemi testati sono Dati.dzn e Dati\_2.dzn, per le due istanze del problema si sono ottenuti i seguenti risultati, espressi in punti di scelta (choice points):

Strategia	choice points
first-fail e indomain-min	3
first-fail e indomain-median	12
first-fail e indomain-random	3
first-fail e indomain-split	4
input-order e indomain-min	3
input-order e indomain-median	25
input-order e indomain-random	3
input-order e indomain-split	4
smallest e indomain-min	3
smallest e indomain-median	21
smallest e indomain-random	4
smallest e indomain-split	5

In Tabella 2 invece riportiamo i tempi ottenuti per il secondo set di dati.

Strategia	choice points
first-fail e indomain-min	20
first-fail e indomain-median	124
first-fail e indomain-random	123
first-fail e indomain-split	26
input-order e indomain-min	20
input-order e indomain-median	298
input-order e indomain-random	130
input-order e indomain-split	26
smallest e indomain-min	20
smallest e indomain-median	43
smallest e indomain-random	21
smallest e indomain-split	28

Analizzando i risultati ottenuti si nota che utilizzando come **constrain-choice** indomain-min o indomain-split otteniamo risultati molto simili e con piccoli valori di punti di decisione per qualsiasi scelta di varchoice, mentre se scegliamo indomain-media o indomain-random le cose peggiorano molto nel set di dati più grandi.

### 3 Conclusioni

In conclusione la miglior scelta per risolvere il problema risulta essere first-fail o smallest abbinati a indomain-min o indomain-split, ma ciò può dipendere molto dalla struttura del problema quindi può variare caso per caso.