



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**Scuola  
di Ingegneria**

Corso di Laurea Triennale  
in  
Ingegneria Informatica

**Progetto di Intelligenza Artificiale**  
Gianmarco Pastore

**CSP sulla creazione di un piano di studi  
modellato con Minizinc**

## Introduzione

In questo elaborato ho sviluppato il modello di un problema CSP, tramite il linguaggio MiniZinc. Il problema consiste nel creare tutti i piani di studio validi a partire da una serie di vincoli sui crediti, i periodi didattici e i settori disciplinari. Per realizzare il codice ho usato l'ide di MiniZinc nella sua versione per Linux Ubuntu.

## Dati utilizzati

I dati che ho utilizzato sono relativi a due corsi di laurea offerti dall'università di Firenze: Ingegneria Informatica e Filosofia. Ho recuperato sul sito [www.unifi.it](http://www.unifi.it) le informazioni sugli esami presenti nell'offerta formativa dei corsi, ossia crediti, periodi didattici, SSD. Per entrambi i CdL ho apportato delle modifiche alle informazioni per poterli utilizzare nel progetto, infatti con i dati originali molti vincoli del CSP sarebbero stati insoddisfacibili; ad esempio nel corso di Ingegneria Informatica la somma dei crediti relativi agli esami del primo anno, nella realtà, non arriva a 60. In generale ho aggiunto o rimosso alcuni esami e ho inserito dei dati inventati da me laddove non fossero stati specificati, come ad esempio SSD e periodo didattico di alcuni esami.

Questi dati si trovano all'interno dei datafile *Ingegneria.dzn* e *Filosofia.dzn*.

## Formulazione problema e modellazione in Minizinc

Ho scelto di utilizzare per questo problema un insieme di variabili booleane, una per ogni esame del corso, che assumono il valore *true* se il rispettivo esame è presente nel piano di studi e *false* altrimenti. Quindi se chiamiamo *nes* il numero totale di esami abbiamo:

$$X = \{X_1, \dots, X_{nes}\}$$
$$D_i = \{True, False\} \quad i = 1 \dots nes$$

dove  $X$  è l'insieme delle variabili e  $D_i$  è il dominio della variabile  $i$ -esima.

Questo tipo di variabile è facilmente realizzabile in MiniZinc con un *array of var bool*, che nel mio caso ho chiamato *pds* (piano di studi). Per indicizzare questo array ho scelto un enumerazione (struttura *enum* fornita da MiniZinc) di esami, inizializzata nei datafiles (.dzn) con i dati del CdL. Ho usato la stessa enum per indicizzare anche gli array dei cfu, dei periodi didattici e dei settori scientifico-disciplinari. Per indicare periodi didattici ho scelto di usare i numeri da 1 a 6 che indicano i 6 semestri dei tre anni previsti per i CdL, dove 1 è il primo semestre del primo anno, 2 il secondo semestre del primo anno, ..., 6 il secondo semestre del terzo anno; ogni esame è strettamente legato al proprio periodo didattico, ossia non è possibile svolgerlo in un anno o semestre diverso da quello indicato.

Ho sfruttato le enumerazioni anche per i settori scientifico-disciplinari degli esami. In particolare ho utilizzato una funzionalità aggiunta nelle ultime versioni di MiniZinc, ossia l'estensione dei tipi enumerati. Grazie ad essa ho potuto creare due diverse enumerazioni che si riferiscono rispettivamente ai SSD affini e caratterizzanti (ASSD e CSSD nel codice), che ho poi unito tramite dei "costruttori" creando un enumerazione finale SSD. Il grande vantaggio offerto dai tipi enumerati estesi consiste nella possibilità di verificare se un valore dell'enumerazione estesa appartiene effettivamente ad una delle enumerazioni sottostanti tramite i costruttori<sup>1</sup>. Questa funzionalità mi ha permesso di scrivere in maniera semplice i constraint visto che ogni esame ha un SSD ed è possibile controllare facilmente se si tratta di un CSSD o ASSD:

---

<sup>1</sup> spiegato sul manuale [www.minizinc.org/doc-2.5.3/en](http://www.minizinc.org/doc-2.5.3/en) al capitolo 2.2.4

```

constraint sum([cfu[exam] | exam in exams where SSD_of_exam[exam] in A(ASSD) /\
pds[exam]]) <=amax /\
    sum([cfu[exam] | exam in exams where SSD_of_exam[exam] in A(ASSD) /\
pds[exam]]) >=amin;

constraint sum([cfu[exam] | exam in exams where SSD_of_exam[exam] in C(CSSD) /\
pds[exam]] ) <=cmax /\
    sum([cfu[exam] | exam in exams where SSD_of_exam[exam] in C(CSSD) /\
pds[exam]] ) >=cmin;

```

Gli altri constraint che ho scritto sono relativi alle varie limitazioni sui cfu per anno, periodo didattico e prova finale.

Per quanto riguarda la ricerca della soluzione ho adoperato il comando *solve satisfy* di Minizinc a cui ho aggiunto l'annotazione *bool\_search* per indicare al solver di MiniZinc il tipo di ricerca da effettuare:

```

solve ::bool_search(pds, first_fail, indomain_min)
satisfy;

```

*first\_fail* è un'euristica che impone di scegliere come variabile quella con il dominio più piccolo mentre *indomain\_min* impone di scegliere come valore per la variabile il più piccolo possibile nel suo dominio <sup>2</sup>.

Il codice sorgente si trova tutto nel file *PDS.mzn*.

## Risultati e conclusioni

Innanzitutto il solver che ho utilizzato per la risoluzione del problema è Gecode 6.3.0, offerto da Minizinc. L'ho configurato in modo da restituire tutti i risultati trovati (sostituendo user defined behavior a default behavior), con optimization level O1 (default). Quando il programma viene eseguito viene chiesto all'utente di inserire i parametri che vengono adoperati per i vincoli sui crediti: amax, amin, cmax, cmin e il valore in cfu della prova finale. Da questi parametri dipende il numero delle soluzioni trovate e di conseguenza la soddisfacibilità del problema stesso. L'output restituito dal programma se impostiamo i parametri amax=100, amin=78, cmax=180, cmin=90, fp=6 è:

PIANO DI STUDI			
ANALISI_MATEMATICA_I	CFU: 9	SSD: A(MAT_05)	PD: 2
FISICA_I	CFU: 6	SSD: A(FIS_03)	PD: 1
FONDAMENTI_DI_INFORMATICA	CFU: 9	SSD: C(ING_INF_05)	PD: 1
PROGRAMMAZIONE	CFU: 6	SSD: C(ING_INF_05)	PD: 2
GEOMETRIA_E_ALGEBRA_LINEARE	CFU: 6	SSD: A(MAT_03)	PD: 1
CALCOLO_NUMERICO	CFU: 6	SSD: A(MAT_08)	PD: 2
LABORATORIO_DI_PROGRAMMAZIONE	CFU: 3	SSD: C(ING_INF_05)	PD: 2
TEORIA_DEI_CIRCUITI	CFU: 6	SSD: A(ING_IND_31)	PD: 2
VERIFICA_LINGUA_INGLESE	CFU: 3	SSD: A(INGL)	PD: 1
CHIMICA	CFU: 6	SSD: A(CHIM_07)	PD: 1
ALGORITMI_E_STRUTTURE_DATI	CFU: 6	SSD: C(ING_INF_05)	PD: 4
ANALISI_MATEMATICA_II_E_PROBABILITA	CFU: 6	SSD: A(MAT_05)	PD: 3
CALCOLATORI	CFU: 6	SSD: C(ING_INF_05)	PD: 3
FISICA_II	CFU: 6	SSD: A(FIS_03)	PD: 4
FONDAMENTI_DI_AUTOMATICA	CFU: 9	SSD: C(ING_INF_04)	PD: 4

<sup>2</sup> spiegato sul manuale [www.minizinc.org/doc-2.5.3/en](http://www.minizinc.org/doc-2.5.3/en) al capitolo 2.5.2

FONDAMENTI_DI_RETI_DI_TELECOMUNICAZIONI	CFU: 6	SSD: C(ING_INF_03)	PD: 4
FONDAMENTI_DI_RICERCA_OPERATIVA	CFU: 6	SSD: A(MAT_09)	PD: 3
FONDAMENTI_DI_SEGNALI_E_TRASMISSIONE	CFU: 6	SSD: C(ING_INF_03)	PD: 3
LABORATORIO_DI_ALGORITMI	CFU: 3	SSD: C(ING_INF_05)	PD: 4
MISURE_ELETTRICHE	CFU: 6	SSD: C(ING_INF_07)	PD: 3
BASI_DI_DATI	CFU: 6	SSD: C(ING_INF_05)	PD: 5
ELETTRONICA_DIGITALE	CFU: 6	SSD: C(ING_INF_01)	PD: 6
INFORMATICA_INDUSTRIALE	CFU: 6	SSD: C(ING_INF_05)	PD: 6
INFORMATICA_TEORICA	CFU: 6	SSD: C(ING_INF_05)	PD: 6
INGEGNERIA_DEL_SOFTWARE	CFU: 6	SSD: C(ING_INF_05)	PD: 5
MATEMATICA_DISCRETA_E_CODICI	CFU: 6	SSD: A(MAT_03)	PD: 5
METODI_MATEMATICI	CFU: 6	SSD: A(MAT_05)	PD: 5
PROGETTAZIONE_E_PRODUZIONE_MULTIMEDIALE	CFU: 6	SSD: C(ING_INF_05)	PD: 6
STATISTICA	CFU: 6	SSD: A(SECS_S_02)	PD: 5

CFU TOTALI:174

CFU ASSD: 78

CFU CSSD: 96

CFU semestre 1: 30

CFU semestre 2: 30

CFU semestre 3: 30

CFU semestre 4: 30

CFU semestre 5: 30

CFU semestre 6: 24

Esami non scelti:

SISTEMI_OPERATIVI	PD: 4	CFU: 6
-------------------	-------	--------

INTELLIGENZA_ARTIFICIALE	PD: 5	CFU: 6
--------------------------	-------	--------

LABORATORIO_DI_INFORMATICA	PD: 6	CFU: 3
----------------------------	-------	--------

SISTEMI_DISTRIBUITI	PD: 6	CFU: 6
---------------------	-------	--------

-----

=====

%%%mzn-stat: initTime=0.002085

%%%mzn-stat: solveTime=0.000907

%%%mzn-stat: solutions=45

%%%mzn-stat: variables=351

%%%mzn-stat: propagators=240

%%%mzn-stat: propagations=7990

%%%mzn-stat: nodes=127

%%%mzn-stat: failures=19

%%%mzn-stat: restarts=0

%%%mzn-stat: peakDepth=9

%%%mzn-stat-end

Finished in 166msec

In questi risultati è riportata una singola soluzione (l'ultima visualizzata), la quale mostra ogni esame inserito nel piano di studi con rispettivi cfu, SSD, e periodo didattico, i dati relativi ai cfu totali, per semestre e per SSD e infine le statistiche del solver Gecode 6.3.0. La correttezza della soluzione e il rispetto dei vincoli si può verificare facilmente osservando i valori riportati alla fine, coerenti con quelli dei singoli esami inseriti in questo particolare piano di studi<sup>3</sup>.

Con un breve calcolo si può verificare anche che le soluzioni totali in questo caso sono effettivamente 45, come evidenziato dalle statistiche di Gecode. I vincoli, in base ai parametri assegnati, permettono di effettuare 3 scelte per pd=4, 3 per pd=5 e 5 per pd=6, in totale 45.

Questi invece sono i risultati per il Cdl di Filosofia, inserendo in in input gli stessi parametri del caso precedente:

<sup>3</sup> I dati riportati sono quelli dell'output di MiniZinc, ma qui sono stati formattati per favorirne la leggibilità

# PIANO DI STUDI

FILOSOFIA_MORALE_I	CFU: 12	SSD: C(M_FIL_03)	PD: 2
FILOSOFIA_POLITICA	CFU: 12	SSD: A(SPS_01)	PD: 1
FILOSOFIA_TEORETICA_I	CFU: 12	SSD: C(M_FIL_01)	PD: 2
LOGICA_I	CFU: 12	SSD: C(M_FIL_02)	PD: 1
STORIA_DELLA_FILOSOFIA	CFU: 6	SSD: C(M_FIL_06)	PD: 1
STORIA_DELLA_FILOSOFIA_ANTICA	CFU: 6	SSD: C(M_FIL_07)	PD: 2
ANTROPOLOGIA_FILOSOFICA	CFU: 6	SSD: C(M_FIL_03)	PD: 3
ELEMENTI_DI_TEORIA_DELLA_CONOSCENZA	CFU: 6	SSD: C(M_FIL_01)	PD: 4
ESTETICA	CFU: 6	SSD: C(M_FIL_03)	PD: 3
ETICA_APPLICATA	CFU: 6	SSD: C(M_FIL_03)	PD: 4
FILOSOFIA_DELLA_LOGICA_E_DEL_LINGUAGGIO	CFU: 6	SSD: C(M_FIL_02)	PD: 4
FILOSOFIA_DELLA_MENTE	CFU: 6	SSD: C(M_FIL_01)	PD: 3
FILOSOFIA_DELLA_SCIENZA	CFU: 6	SSD: C(M_FIL_02)	PD: 3
FILOSOFIA_SOCIALE	CFU: 6	SSD: A(SPS_01)	PD: 3
STORIA_MODERNA	CFU: 12	SSD: A(M_STO_02)	PD: 4
ABILITA_INFORMATICHE	CFU: 6	SSD: A(INF)	PD: 5
CONOSCENZA_LINGUA_STRANIERA	CFU: 6	SSD: A(INGL)	PD: 5
GEOGRAFIA_POLITICA	CFU: 6	SSD: A(M_GGR_01)	PD: 6
ISTITUZIONI_DI_SOCIOLOGIA	CFU: 6	SSD: A(SPS_07)	PD: 6
LETTERATURA_FRANCESE	CFU: 6	SSD: A(L_LIN_03)	PD: 5
LETTERATURA_GRECA_I	CFU: 6	SSD: A(L_FIL_LET_02)	PD: 6
LETTERATURA_INGLESE	CFU: 6	SSD: A(L_LIN_10)	PD: 5
LETTERATURA_ITALIANA	CFU: 6	SSD: A(L_FIL_LET_10)	PD: 6
LETTERATURA_LATINA	CFU: 6	SSD: A(L_FIL_LET_04)	PD: 5

CFU TOTALI:174

CFU ASSD: 84

CFU CSSD: 90

CFU semestre 1: 30

CFU semestre 2: 30

CFU semestre 3: 30

CFU semestre 4: 30

CFU semestre 5: 30

CFU semestre 6: 24

Esami non scelti:

LOGICA_II	PD: 3	CFU: 6
STORIA_DELL_ESTETICA	PD: 3	CFU: 6
LETTERATURA_TEDESCA	PD: 6	CFU: 6
LETTERATURE_COMPARATE	PD: 5	CFU: 6
LINGUISTICA_GENERALE	PD: 6	CFU: 6
SERVIZI_DI_ORIENTAMENTO_AL_LAVORO	PD: 5	CFU: 1
TIROCINIO	PD: 6	CFU: 5

-----

=====

```

%%%mzn-stat: initTime=0.002615
%%%mzn-stat: solveTime=2.08031
%%%mzn-stat: solutions=1890
%%%mzn-stat: variables=327
%%%mzn-stat: propagators=365
%%%mzn-stat: propagations=627164
%%%mzn-stat: nodes=10373
%%%mzn-stat: failures=3297
%%%mzn-stat: restarts=0
%%%mzn-stat: peakDepth=16
%%%mzn-stat-end
Finished in 2s 550msec

```

Anche stavolta riporto solo l'ultima soluzione visualizzata, che risulta coerente con tutti i vincoli del CSP.

## **Siti e Materiale consultato**

- [1] Guida MiniZinc 2.5.3: [www.minizinc.org/doc-2.5.3/en/](http://www.minizinc.org/doc-2.5.3/en/)
- [2] Offerta formativa della facoltà di Ingegneria Informatica di Unifi
- [3] Offerta formativa della facoltà di Filosofia di Unifi