Sistemi con Vincoli

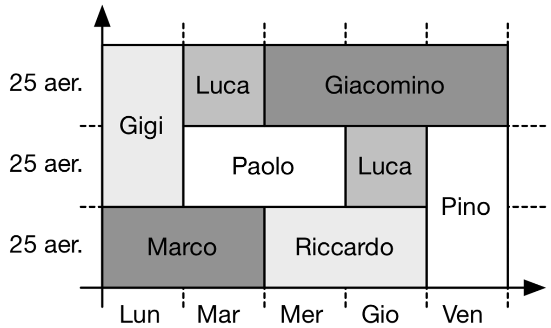
Prova d'Esame del 12 Febbraio 2016

Esercizio 1: La Premiata Ditta

Pierino è il boss indiscusso della scuola: ha raggiunto questa invidiabile posizione grazie ad un fiorente commercio di aereoplanini di carta dal design innovativo ad accattivante, che spopolano tra i bambini dalla prima alla quinta elementare. Per far fronte all'enorme mole di aereoplanini richiesti (50 al giorno!), il giovane imprenditore ricorre all'aiuto dei suoi amichetti, ad una consistente quantità di merendine per alimentare l'imponente meccanismo produttivo, e ad un sofisticato metodo di pianificazione basato su Programmazione a Vincoli.

In particolare, ogni settimana Pierino consultai suoi amici per raccogliere le loro disponibilità di massima: ogni amico fornisce la disponibilità ad effettuare uno o più turni, specificando per ognuno di essi il giorno di inizio, la durata, ed il numero di aereoplanini che è in grado di produrre in un giorno. Pierino ha cura che ogni giorno via sia la possibilità di produrre almeno 75 aereoplanini, quindi procede a selezionare alcuni dei turni, in modo da soddisfare la richiesta di 50 aereoplanini al giorno: si tratta di una operazione complessa, perché i giovani dipendenti sono golosi di merendine e Pierino ne dispone di una quantità giornaliera limitata.

Una data settimana, lo schema dei turni è il seguente:



Gigi e Pino hanno bisogno di 3 merendine al giorno, Luca e Marco di 2, Giacomino, Riccardo e Paolo di 1 merendina al giorno. Pierino può fornire 3 merendine ogni giorno. Inoltre Gigi e Marco non vanno d'accordo e non vogliono lavorare insieme, mentre i due turni di Luca vanno accettati o scartati in blocco. Riuscirà Pierino a soddisfare continuare la produzione nella settimana? Si modelli il problema e si mostri una possibile soluzione.

Suggerimento: si consideri il vincolo sul soddisfacimento della domanda per ultimo.

Soluzione:

Si può introdurre una variabile per ogni amichetto, per indicare se il suo turno viene selezionato

```
Gg, Lc, Pl, Mc, Gc, Rc, Pn in {0, 1}
```

Si noti che in questo modo i due turni di Luca sono trattati nel modo corretto.

I turni di Gigi e Marco sono sovrapposti ed i due non vogliono lavorare insieme. Pertanto, le due scelte sono in mutua esclusione:

```
Gg + Mc <= 1
```

Tutti i turni selezionati vanno adeguatamente supportati con merendine. Si tratta di un vincolo di risorsa:

```
cumulative([0, 0, 1, 1, 2, 2, 3, 4], // istanti di inizio
       [1, 2, 1, 2, 3, 2, 1, 1], // durate
       [3*Gg, 2*Mc, 2*Lc, 1*Pl, 1*Gc, 1*Rc, 3*Pn], // requisiti
       3) // capacità
```

Si noti che gli istanti di inizio sono fissati.

Poiché è garantito che i turni arrivino a produrre fino a 75 areoplanini al giorno, il vincolo sulla domanda può essere modellato come un cumulative "invertito". In pratica, ogni giorno bisogna scartare al più 25 areoplanini:

```
cumulative([0, 0, 1, 1, 2, 2, 3, 4], // istanti di inizio
    [1, 2, 1, 2, 3, 2, 1, 1], // durate
    [2*(1-Gg), (1-Mc), (1-Lc), (1-Pl), (1-Gc), (1-Rc), (1-Pn)], // requisiti
     1) // capacità
```

Utilizzando 25 areoplanini come unità di misura.

L'ultimo vincolo può essere anche modellato agevolmente utilizzando dei vincoli reificati. In particolare, possiamo usare:

```
LUN: Mc + 2*Gg ≥ 2
MAR: Mc + Lc + Pl ≥ 2
MER: Rc + Pl  Gc ≥ 2
GIO: Rc + Lc + Gc ≥ 2
VEN: 2*Pn + Gc ≥ 2
```

Sistemi con Vincoli

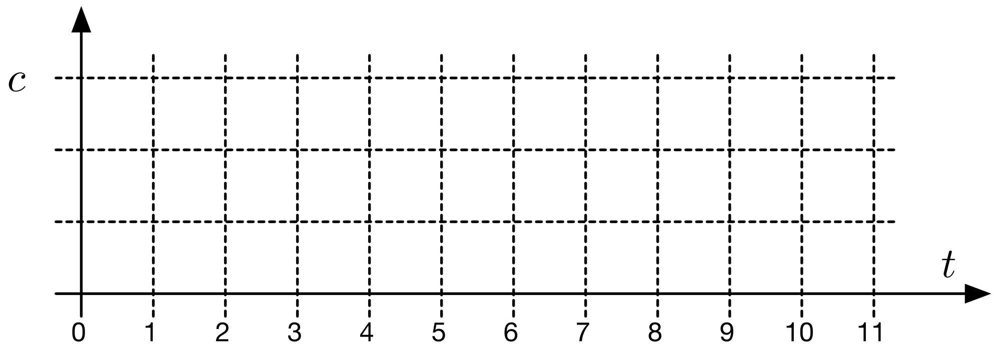
Prova d'Esame del 12 Febbraio 2016

Esercizio 2: Filtering con l'algoritmo "Timetable"

Si consideri il vincolo cumulative *cumulative(s, d, r, 3),* con:

#	$D(s_i)$	d_i	r_i
0	[0..0]	2	1
1	[0..1]	3	1
2	[3..4]	3	2
3	[2..4]	3	1
4	[0..8]	3	2

Si individuino le parti obbligatorie, quindi si mostri il profilo minimo di utilizzo della risorsa sulla griglia seguente. Quindi si esegua l'algoritmo di filtering "timetable" per s_4, indicando sulla griglia le posizioni visitate dal cursore ed il suo stato (C = checking ed S = seeking).

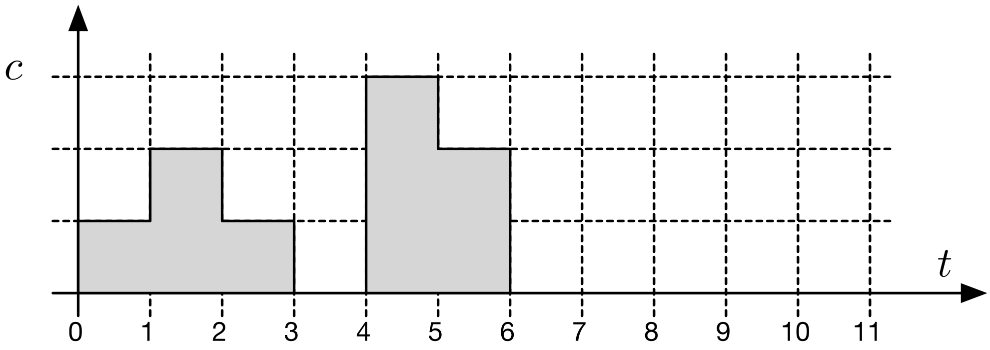


Soluzione:

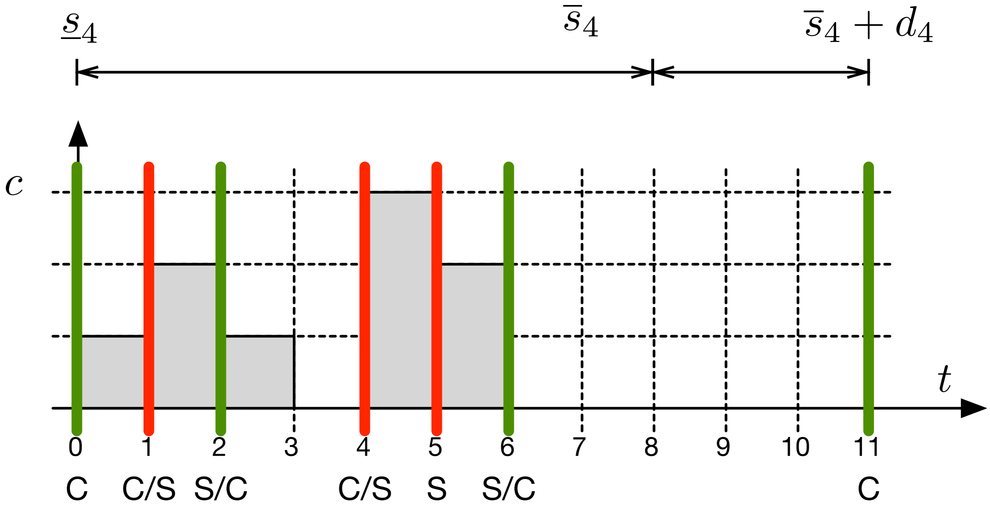
Parti obbligatorie:

#	$D(s_i)$	d_i	r_i	LST	EET
0	[0..0]	2	1	0	2
1	[0..1]	3	1	1	3
2	[3..4]	3	2	4	6
3	[2..4]	3	1	4	5
4	[0..8]	3	2	0	3

Profilo:


c
t
0
1
2
3
4
5
6
7
8
9
10
11

Cursore:


$\underline{s}_4$
$\overline{s}_4$
$\overline{s}_4 + d_4$
c
t
0
1
2
3
4
5
6
7
8
9
10
11
C
C/S
S/C
C/S
S
S/C
C

Sistemi con Vincoli

Prova d’Esame del 12 Febbraio 2016

Esercizio 3: propagazione di vincoli

Si consideri il seguente CSP:

$\boxed{0}\ x_1 = max(x_0, 0)$

$\boxed{1}\ x_0 + x_1 = x_2$

$\boxed{2}\ x_2 = 4\,(x_0 > 0)$

$x_0 \in \{-1..2\}, x_1 \in \{-1..1\}, x_2 \in \{0..4\}$

Si facciano le seguenti assunzioni:

- Sui vincoli “<=“ viene applicata la Bound Consistency
- Sulle espressioni “+”, “*” e “max” viene applicata la bound consistency (in particolare l’output dell’espressione ha come dominio un intervallo)
- Sui vincoli “=“ viene applicata la Generalized Arc Consistency

Si indichi l’attivazione dei vincoli determinata dall’algoritmo AC1.

- Si mostrino i domini ad ogni passo
- Si discuta il filtering effettuato ad ogni passo

Soluzione

1. Passo 1, Vincolo 0
    1. Riscrittura: {-1..1} = max({-1..2}, 0)
    2. Riscrittura: {-1..1} = {0..2}
    3. Filtering: x_1 non può valere “-1”, l’espressione “max(x_0, 0)” non può valere “2”. Di conseguenza, x_0 non può valere “2”
    4. Domini: x_0 in {-1..1}, x_1 in {0,1}, x_2 in {0..4}
2. Passo 2, Vincolo 1
    1. Riscrittura: {-1..1} + {0,1} = {0.4}
    2. Filtering: calcolo dei bound classico per i vincoli di somma
    3. Domini: x_0 in {-1..1}, x_1 in {0,1}, x_2 in {0..2}
3. Passo 3, Vincolo 2
    1. Riscrittura: {0..2} = 4 ({-1..1} > 0)
    2. Riscrittura: {0..2} = 4 {0,1}
    3. Riscrittura: {0..2} = {0..4}
    4. Filtering: l’espressione “4(x_0 > 0)” non può valere “4”, di conseguenza il metavincolo “x_0 > 0” deve essere falso. Da qui si deduce che x_0 non può valere “1”. Inoltre, poiché il metavincolo è falso, x_2 viene forzato a “0”
    5. Domini: x_0 in {-1,0}, x_1 in {0,1}, x_2 in {0}
4. Passo 4, Vincolo 0

   1. Riscrittura: {0,1} = max({-1,0}, 0)
   2. Riscrittura: {0,1} = {0}
   3. Filtering: x_1 non può valere “1”
   4. Domini: x_0 in {-1,0}, x_1 in {0}, x_2 in {0}
5. Passo 5, Vincolo 1
   1. Riscrittura: {-1,0} + {0} = {0}
   2. Filtering: x_0 non può valere “-1”
   3. Domini: x_0 in {0}, x_1 in {0}, x_2 in {0}
6. Passo 8, Vincolo 2
   1. Domini inalterati
7. Passo 7, Vincolo 0
   1. Domini inalterati
8. Passo 8, Vincolo 1
   1. Domini inalterati
9. Passo 9, Vincolo 2
   1. Domini inalterati

Sistemi con Vincoli

Prova d'Esame del 12 Febbraio 2016

Esercizio 3: progettazione

Si consideri il seguente CSP:

$$
\begin{aligned}
\min z = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} c_{i,j} \, (x_i = j) & \\
\sum_{i \in S_0} (x_i = j) \leq 1 & \qquad \forall j = 0..m-1 \\
x_i \neq x_h & \qquad \forall i, h \in S_1, i \neq h \\
y_0 = (x_0 = 0 \wedge x_1 = 1) & \\
y_1 = (x_0 = 1 \wedge x_1 = 2) & \\
y_2 = (x_0 = 0 \wedge x_1 = 2) & \\
y_0 + y_1 + y_2 = 1 & \\
x_i \in \{0..m-1\} & \qquad \forall i = 0..n-1 \\
y_0, y_1, y_2 \in \{0..1\} &
\end{aligned}
$$

Dove:

- $c_{i,j}$ sono dei coefficienti di costo
- $3 < n < m$
- $S_0 \cup S_1 = \{0..m\text{-}1\}$

È possibile sostituire alcuni dei vincoli utilizzando vincoli globali, o aggiungere dei vincoli globali ridondanti?

Soluzione

Le espressioni:

$$\sum_{j=0}^{n-1} c_{i,j} \, (x_i = j)$$

Possono essere sostituite con dei vincoli ELEMENT:

$[c_{i,j} | j = 0..m-1]_{x_i}$

I vincoli:

$y_0 = (x_0 = 0 \wedge x_1 = 1)$

$y_1 = (x_0 = 1 \wedge x_1 = 2)$

$y_2 = (x_0 = 0 \wedge x_1 = 2)$

$y_0 + y_1 + y_2 = 1$

Possono essere sostituiti con un vincolo TABLE:
$\textsc{table}([x_0, x_1], T)$

Dove la tabella T è data da:

x_0	x_1
0	1
1	2
0	2

Infine i vincoli:

$$\sum_{i \in S_0} (x_i = j) \leq 1 \qquad \forall j = 0..m-1$$

$$x_i \neq x_h \qquad \forall i, h \in S_1, i \neq h$$

Possono essere sostituiti da un singolo ALLDIFF, perché S_0 ed S_1 partizionano l'insieme {0..m-1} dei possibili valori
$\textsc{alldiff}(x)$

In alternativa si possono usare anche due ALDIFF, anche se la propagazione risulta un po' più debole in tale modo. Sempre in alternative agli ALDIFF, è possibile utilizzare dei vincoli GCC con tutte le cardinali massime impostate ad 1