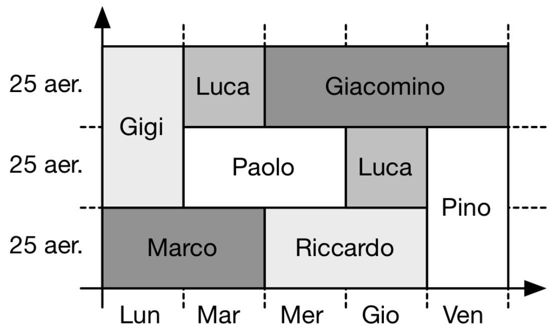
Sistemi con Vincoli

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 consulta i 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 esattamente 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 la produzione nella settimana? Si modelli il problema e si mostri una possibile soluzione.

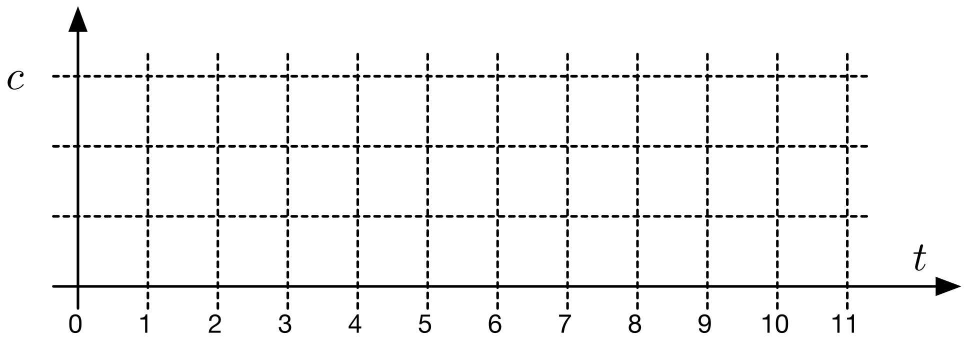
Suggerimento: si consideri il vincolo sul soddisfacimento della domanda per ultimo.

Esercizio 2: filtering con l'algoritmo "timetable"

Si consideri il vincolo *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. Si esegua poi 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).



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 e si discuta brevemente il filtering effettuato ad ogni passo

Esercizio 4: progettazione

Si consideri il seguente CSP:

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

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

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

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

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

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

$$y_0 + y_1 + y_2 = 1$$

$$x_i \in \{0..m-1\} \qquad \forall i = 0..n-1$$

$$y_0, y_1, y_2 \in \{0..1\}$$

Dove: *ci,j* sono dei coefficienti di costo, *3 < n < m* e *S0* ∪ *S1 = {0..m-1}*.

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

Esercizio 5: teoria

In quali casi utilizzata una strategia di ricerca randomizzata ed il meccanismo dei "restart" può portare vantaggi? Per quali ragioni?