

12 giugno 2019 - laboratorio

# Constraint Programming

Prof. Marco Gavanelli

18 settembre 2018

## Descrizione problema

Una linea di produzione deve produrre dei trattori.

Nel file `production.pl` viene dato un insieme di ordini; ogni ordine è rappresentato da un fatto

$ordine(ID, Deadline)$
------------------------

dove

- $ID$  rappresenta l'identificativo dell'ordine (un numero intero)
- $Deadline$  è la scadenza entro cui l'ordine andrebbe evaso, in una certa unità di tempo (es. in giorni). Se l'ordine viene evaso entro la scadenza, dà luogo ad un guadagno, altrimenti non dà luogo ad alcun guadagno.

Produrre un ordine impiega una unità di tempo e consuma tutte le risorse disponibili. Si può cominciare a produrre a partire dall'istante 0.

Ci sono vincoli di precedenza fra gli ordini; le precedenze sono riportate in fatti

$precedenza(ID1, ID2)$
------------------------

un tale fatto significa che l'ordine con identificativo  $ID1$  deve essere terminato prima che si inizi a produrre l'ordine  $ID2$ .

Tutti gli ordini devono essere terminati entro l'istante di tempo riportato nel fatto

$endtime/1.$
--------------

Qualora un ordine venga evaso entro la sua scadenza, si ottiene un guadagno unitario; gli ordini che non vengono terminati entro la scadenza non danno luogo ad alcun guadagno.

Si desidera massimizzare il guadagno totale.

## CLP (punti 12)

Si risolva il problema usando ECLiPSe.

## ASP (8 punti)

Si risolva il problema in Answer Set Programming.

## MiniZinc (punti 5)

Si risolva il problema usando MiniZinc. Come dati di input, si usino quelli del file `dati.mzn`, che possono essere copiati sul file da consegnare.

Nel file vengono dati:

- il numero di ordini *Nordini*
- l'array *deadline*; *deadline[i]* rappresenta la deadline per l'ordine *i*
- il tempo totale *EndTime* entro cui tutti gli ordini devono essere evasi
- il numero di relazioni di precedenza *Nprecedenze*
- un array *p* che contiene le precedenze. Per ogni indice *i* compreso fra 1 e *Nprecedenze*, l'ordine *p[i,1]* deve essere evaso prima di *p[i,2]*