Orario Scolastico

Filippo Botti e Lorenzo Riccardi

10 marzo 2022

Sommario

Your abstract.

1 Assegnamento

In una scuola si deve pianificare l'orario settimanale. Ogni materia richiede un certo numero di ore. Inoltre il docente di una materia presenta una richiesta di un giorno libero, l'indisponibilità in certe ore e inoltre vorrebbe avere un basso numero di buchi nel proprio orario. E però prevedibile che alcune di queste richieste non possano essere soddisfatte. Si vuole pianificare l'orario in modo da soddisfare il maggior numero possibile delle richieste dei docenti. Si formuli il modello matematico per questo problema, lo si scriva in AMPL e si definiscano i dati di una particolare istanza, risolvendola. Si faccia inoltre un'analisi di cosa succede se si modificano alcuni dei dati dell'istanza. Pur costruendo un modello generale, per semplicità nelle istanze testate si consideri un numero limitato di materie e di ore giornaliere

2 Modello Matematico

In questa sezione andremo a descrivere il modello matematico pensato per risolvere il problema.

2.1 Insiemi

Per prima cosa occorre definire gli insiemi che caratterizzano il problema:

- Professori
- Classi
- Materie
- Ore
- Giorni
- Giorni Liberi $\subset$ (Professori $\times$ Giorni)
- Lezioni $\subset$ (Giorni $\times$ Ore)
- Cattedre $\subset$ (Materie $\times$ Professori)
- Ore Libere $\subset$ (Professori $\times$ Lezioni)

2.2 Parametri

Successivamente occorre dichiarare alcuni parametri che ci serviranno per la definizione del problema:

- $\forall$ m in Materie: $ore\_per\_materia[m] \geq 0$, interi
- M
- ore_al_giorno, intero

2.3 Variabili

Dati gli insiemi e i parametri sopra descritti, il problema consiste nel trovare una rappresentazione possibile dell'orario scolastico e cercare di soddisfare le richieste dell'assegnamento. Per questo motivo definiamo le variabili:

- $x[c,m,p,g,h]$ binarie, le quali rappresentano le lezioni. Infatti avranno valore 1 nel caso in cui nella classe c la materia m è insegnata dal professore p nel giorno g all'ora h, 0 negli altri casi.
- $gl[p,g]$ anch'esse binarie, che rappresentano i giorni lavorativi di ogni professore. Saranno dunque pari ad 1 se il professore p ha almeno una lezione nel giorno g, 0 negli altri casi.

2.4 Vincoli

Possiamo ora modellare il problema andando a definire i vincoli che esso deve rispettare. Per ogni vincolo vedremo la definizione simbolica e una spiegazione che ne convalida l'esistenza.

Ore giornaliere

$\forall$ *c in Classi,* $\forall$ *g in Giorni*:
$\sum_{m \in Mat} \sum_{p:(m,p) \in Cat} \sum_{h:(g,h) \in Lezioni} x[c,m,p,g,h] = ore\_al\_giorno$

Per ogni classe e per ogni giorno la somma di tutte le ore di lezione deve essere pari al numero di ore giornaliere definito dall'orario.

Ore in contemporanea per classe

$\forall$ *c in Classi,* $\forall$ *g in Giorni,* $\forall$ *h in Ore* | *(g,h)* $\in$ *Lezioni*:
$\sum_{m \in Mat} \sum_{p:(m,p) \in Cat} x[c,m,p,g,h] = 1$

Per ogni classe, per ogni giorno e per ogni ora la somma di tutte le ore di lezione deve essere pari ad 1, ovvero non posso avere due lezioni alla stessa ora dello stesso giorno in una classe.

Ore in contemporanea per professore

$\forall$ *p in Prof,* $\forall$ *g in Giorni,* $\forall$ *h in Ore* | *(g,h)* $\in$ *Lezioni*:
$\sum_{c \in Classi} \sum_{m:(m,p) \in Cat} x[c,m,p,g,h] <= 1$

Per ogni professore, per ogni giorno e per ogni ora la somma di tutte le ore di lezione deve essere minore o uguale ad 1, ovvero non posso avere due lezioni alla stessa ora dello stesso giorno tenute dallo stesso professore.

Ore massime per professore in una classe

$\forall$ *p in Prof,* $\forall$ *g in Giorni,* $\forall$ *c in Classi*:
$\sum_{m:(m,p) \in Cat} \sum_{h:(g,h) \in Lezioni} x[c,m,p,g,h] <= 3$

Per ogni professore, per ogni giorno e per ogni classe la somma delle ore di lezioni tenute dal professore nella classe selezionata deve essere minore o uguale a tre, ovvero ogni classe non può avere per più di tre ore lo stesso professore durante la stessa giornata.

Ore massime per materia in una classe

$\forall$ *m in Materie,* $\forall$ *g in Giorni,* $\forall$ *c in Classi*:
$\sum_{p:(m,p) \in Cat} \sum_{h:(g,h) \in Lezioni} x[c,m,p,g,h] <= 2$

Per ogni materia, per ogni giorno e per ogni classe la somma delle ore di lezione della materia nella classe selezionata deve essere minore o uguale a due, ovvero ogni classe non può avere più di dure ore della stessa materia durante la stessa giornata.

Ore materia a settimana

$\forall$ *m in Materie,* $\forall$ *c in Classi*:
$\sum_{p:(m,p)\in Cat} \sum_{(g,h)\in Lezioni} x[c,m,p,g,h] = ore\_materia[m]$

Per ogni classe e per ogni materia la somma delle ore totali a settimana della materia deve essere uguale al valore stabilito per la materia.

Definizione giorni lavorativi

$\forall$ *p in Prof,* $\forall$ *g in Giorni*:
$\sum_{p:(m,p)\in Cat} \sum_{m:(m,p)\in Cat} \sum_{h:(g,h)\in Lezioni} x[c,m,p,g,h] >= gl[p,g]$

$\forall$ *(m,p) in Cat,* $\forall$ *(g,h) in Lezioni,* $\forall$ *c in Classi*:
$gl[p,g] >= x[c,m,p,g,h]$

Stiamo definendo le variabili relative ai giorni lavorativi. La prima espressione stabilisce che $gl[p,h]$ deve essere minore o uguale alla somma delle lezioni tenute dal professore nel giorno selezionato, quindi se il professore non lavora nel giorno scelto $gl[p,g]$ dovrà essere uguale (essendo binaria) a zero.
La seconda espressione stabilisce invece che $gl[p,g]$ debba essere maggiore o uguale a ogni variabile $x[c,m,p,g,h]$, in particolare se un professore avrà almeno una lezione nel giorno selezionato $gl[p,g]$ dovrà essere uguale (sempre perché binaria) ad uno.

Giorni lavorativi

$\forall$ *p in Prof*:
$\sum_{ginGiorni} gl[p,g] <= 5$

Data una settimana lavorativa di sei giorni, vogliamo che i professori abbiano almeno un giorno libero, di conseguenza la somma dei giorni lavorativi per ogni professore deve essere minore o uguale a cinque.

Singolo professore per materia

$\forall$ *(m,p) in Cat,* $\forall$ *(g,h) in Lezioni,* $\forall$ *c in Classi*:
$\sum_{pr\neq p:(m,pr)\in Cat} \sum_{(gi,hi)\in Lezioni} x[c,m,pr,gi,hi] <= (1 - x[c,m,p,g,h]) * M$

Per ogni classe, materia, professore, giorno e ora la somma delle ore di lezione di un professore, diverso da quello iniziale, per la stessa materia deve essere minore o uguale a zero, se il professore iniziale insegna la materia nella classe selezionata, mentre deve essere inferiore a $BigM$ negli altri casi.

Ore non consecutive dei professori

$\forall$ *(m,p) in Cat,* $\forall$ *g in Giorni,* $\forall$ *c in Classi,* $\forall$ *h in Ore | (h+1) in Ore, (g,h) in Lezioni*:
$\sum_{j>=h+1:j\in Lezioni} \sum_{(mat,p)\in Cat} x[c,mat,p,g,j] <= (1 - x[c,m,p,g,h]) * M + x[c,m,p,g,h+1] * M$

Per ogni classe, materia, professore, giorno e ora la somma delle ore successive a quella selezionata deve essere minore o uguale a zero se il professore insegna nella classe all'ora selezionata e non in quella successiva, deve essere inferiore a $BigM$ negli altri casi. Sostanzialmente questo vincolo indica che se un professore non può avere lezioni separate durante la stessa giornata nella stessa classe.

2.5 Obiettivi

Andiamo ora a definire gli obiettivi del problema.

Minimizzare i giorni lavorativi

$\forall$ *p in Prof*:
$\sum_{g \in Giorni} gl[p, g]$

L'idea è quella di minimizzare i giorni lavorativi dei professori, questo per compattare le lezioni dei professori con meno ore.

Minimizzare le ore di lavoro nelle ore libere richieste

$\forall$ *p in Prof*:
$\sum_{c \in Classi} \sum_{(g,h) \in Lezioni:(p,g,h) \in Ore\_Libere} \sum_{m:(m,p) \in Cat} gl[p, g]$

Si vuole minimizzare le ore di lavoro dei professori nelle ore per le quali i professori hanno fatto richiesta di indisponibilità.

Minimizzare i giorni di lavoro nei giorni liberi richiesti

$\forall$ *p in Prof*:
$\sum_{g \in Giorni:(p,g) \in Giorni\_Liberi} gl[p, g]$

Stesso ragionamento dell'obiettivo precedente, ma riferito ai giorni per i quali i professori hanno fatto richiesta di indisponibilità.

Minimizzare le ore buche

$\forall$ *p in Prof,* $\forall$ *g in Giorni,* $\forall$ *h in Ore* | *(h+1) in Ore*:
$\sum_{m:(m,p) \in Cat} \sum_{c \in Classi} sum_{j>h:(g,j) \in Lezioni}(x[c, m, p, g, j] - x[c, m, p, g, h+1] * M$

L'obiettivo in questione vuole minimizzare le ore buche tra una lezione ed un'altra per ogni professore durante lo stesso giorno. Per ogni professore, per ogni giorno e per ogni ora la somma delle ore di lezione successive a quella selezionata meno *BigM* se il professore ha lezione nell'ora immediatamente successiva a quella selezionata deve essere minimizzata. Idealmente la somma sopra descritta deve essere pari a zero, ovvero un professore non deve avere ore buche tra una lezione ed un'altra.

3 Modello AMPL

In questa sezione andremo a trattare la conversione in linguaggio AMPL del modello matematico appena formulato, dapprima da un punto di vista strutturale, quindi definendo solo il problema e, successivamente, da un punto di vista implementativo, definiendo dunque alcune istanze del problema e commentando i loro risultati.

3.1 Insiemi

In questa sezione andremo a definire gli insiemi necessari all'elaborazione del progetto. Ricordando che la definizione di un insieme si effettua attraverso la keyword *SET* e che le keywords *whitin* e *cross* stanno rispettivamente per sottoinsieme e prodotto cartesiano.
Otteniamo dunque le seguenti definizioni:

```
set PROFESSORI;
set CLASSI;
set MATERIE;
set ORE;
set GIORNI;
set GIORNI_LIBERI within PROFESSORI cross GIORNI;
set LEZIONI within GIORNI cross ORE;
```

```
set CATTEDRE within MATERIE cross PROFESSORI;
set ORE_LIBERE within PROFESSORI cross LEZIONI;
```

3.2 Parametri

Passiamo ora alla definizione dei parametri, usando la keyword *param* per dichiararli e la keyword *integer* per stabilire la loro natura di numeri interi:

```
param ore_per_materia{MATERIE} >= 0, integer;
param M := 100;
param ore_al_giorno integer, default 5;
```

Come vediamo in questa sezione compare un parametro, il parametro M che non viene utilizzato nel modello matematico. Questo parametro viene infatti utilizzato quando, nei vincoli o negli obiettivi, compare una condizione *if-else*. Infatti, non avendo a disposizione un comando tale da poter espimere questa condizione logica, dovremo utilizzare la cosiddetta *Big M Notation*, la quale verrà spiegata più avanti nell'apposita sezione.

3.3 Variabili

In questa sezione andremo a definire il cuore del problema. Come già ampiamente spiegato la risoluzione del problema si basa sull'utilizzo di due tipologie di variabili binarie, definite attraverso la keyword *binary*:

```
var x{c in CLASSI, (m,p) in CATTEDRE, (g,h) in LEZIONI} binary;
var gl{p in PROFESSORI, g in GIORNI} binary;
```

3.4 Vincoli

Definiti gli insiemi, i parametri e le variabili con cui lavorare non ci resta che definire i vincoli che il nostro problema DEVE rispettare. Essendo questa sezione la parte cruciale del problema sarebbe riduttivo elencarli come fatto per insiemi, parametri e variabili, è dunque ragionevole un breve commento per ognuno di essi.

Giorni lavorativi

```
subject to giorno_lavorativo{(p,g) in PROFESSORI cross GIORNI}:
        gl[p,g] <= sum{m in MATERIE, h in ORE, c in CLASSI : (m,p) in CATTEDRE}
        x[c,m,p,g,h];

subject to giorno_lavorativo2{(m,p) in CATTEDRE, (g,h) in LEZIONI, c in CLASSI}:
        gl[p,g] >= x[c,m,p,g,h];
```

Questi due vincoli vengono sfruttati per definire le variabili binarie relative ai giorni lavorativi: infatti dal primo vincolo *giorno lavorativo* definiamo che ogni variabile gl[p,g] associata ad un professore e ad un giorno deve essere inferiore alla somma delle ore lavorative del professore nel giorno stabilito. Questo per fare in modo che, se nel giorno stabilito il professore non avesse alcuna lezione, allora la variabile gl[p,g] dovrà essere minore o uguale a zero, cioè zero e quindi rappresenta il fatto che il professore non lavori in quel determinato giorno.

Analogamente il secondo vincolo *giorno lavorativo2*, determina che la variabile gl[p,g] debba essere maggiore o uguale a ciascuna delle ore di lezione che ha nel rispettivo giorno, perciò ne basta una sola a determinare il fatto che gl[p,g] sia uguale ad uno, che quindi rappresenta il fatto che il professore lavori in quel determinato giorno.

Ore giornaliere

```
subject to ore_giornaliere{c in CLASSI, g in GIORNI} :
        sum{(m,p) in CATTEDRE, h in ORE: (g,h) in LEZIONI}
        x[c,m,p,g,h] = ore_al_giorno;
```

Questo vincolo rappresenta il numero delle ore giornaliere di lezione: per ogni classe e per ogni giorno, infatti, la somma di tutte le lezioni, di tutti i professori deve essere uguale a al parametro che definisce il numero di ore giornaliere, che se non specificato tra i dati vale cinque.

Ore in contemporanea per classe

```
subject to ore_in_contemporanea_classe{c in CLASSI, (g,h) in LEZIONI} :
        sum{(m,p) in CATTEDRE} x[c,m,p,g,h] = 1;
```

Questo vincolo rappresenta il fatto che una classe debba avere una ed una sola lezione per ogni ora, dunque la sommatoria, per ogni classe e per ogni ora di lezione del numero di lezioni deve essere pari ad uno.

Ore in contemporanea per professore

```
subject to ore_in_contemporanea_prof{p in PROFESSORI, (g,h) in LEZIONI} :
        sum{c in CLASSI, m in MATERIE :
         (m,p) in CATTEDRE} x[c,m,p,g,h] <= 1;
```

Questo vincolo è sostanzialmente uguale come logica al vincolo precedente, ma riguarda i professori. L'unica differenza è che un professore non è detto che debba per forza avere una lezione per ogni ora, dunque non sarà più un vincolo di uguaglianza, ma sarà un vincolo di minore, nel caso in cui non abbia lezione, o uguale, nel caso in cui abbia lezione, ad uno.

Ore massime di un professore in una classe

```
subject to ore_massime_professori{c in CLASSI, g in GIORNI, p in PROFESSORI} :
        sum{h in ORE, m in MATERIE : (m,p) in CATTEDRE}
        x[c,m,p,g,h] <= 3;
```

Questo vincolo indica il fatto che un professore non possa insegnare per più di tre ore nella stessa classe, nello stesso giorno. Infatti per ogni classe, giorno e professore, la somma delle lezioni deve essere minore o uguale a tre.

Ore massime di una materia in una classe

```
subject to ore_massime_materia{c in CLASSI, g in GIORNI, m in MATERIE} :
        sum{h in ORE, p in PROFESSORI : (m,p) in CATTEDRE } x[c,m,p,g,h] <= 2;
```

Analogamente al vincolo precedente, il numero di ore di una stessa materia, per ogni classe, nella stessa giornata non deve soprassare le due ore.

Ore di una materia a settimana

```
subject to ore_materia{c in CLASSI, m in MATERIE} :
        sum{(g,h) in LEZIONI, p in PROFESSORI : (m,p) in CATTEDRE }
         x[c,m,p,g,h] = ore_per_materia[m];
```

Questo vincolo indica il fatto che, per ogni materia il numero totale delle ore a settimana debba essere uguale al parametro definito per rispettare l'orario scolastico. Ovviamente il vincolo è esteso ad ogni classe, infatti, per ogni classe e per ogni materia, la somma totale delle ore relative alla materia stabilita deve essere uguale al numero delle ore a settimana per la relativa materia.

Singolo prof per materia

```
subject to singolo_prof_per_materia{c in CLASSI, g in GIORNI, m in MATERIE,
                p in PROFESSORI, h in ORE: (m,p) in CATTEDRE}:
                sum{gi in GIORNI, pr in PROFESSORI,
                hi in ORE: pr!=p && (m,pr) in CATTEDRE}
                x[c,m,pr,gi,hi] <= (1-x[c,m,p,g,h])*M;
```

Questo vincolo stabilisce il fatto che per ogni classe, il numero di professori che insegnano la stessa materia debba essere uguale a uno, ovvero non posso avere due professori diversi per una stessa materia, nella stessa classe. Infatti, per ogni classe, giorno, materia, professore e ore, la somma delle ore dei professori che insegnano la stessa materia, ma che non sono il professore preso in considerazione inizialmente deve essere minore o uguale a zero, se il professore primario p insegna la materia in quella classe, mentre deve essere minore o uguale a M costante elevata in tutti gli altri casi. E' qui che vediamo dunque l'utilizzo della *Big M Notation*: se il professore insegna la materia in quella determinata classe allora la somma di tutte le lezione degli altri professori che insegnano la stessa materia in quella classe dovrà essere minore o uguale a (1-1)*M ovvero zero; nel caso invece in cui il professore non insegnasse la materia nella classe la somma dovrà essere minore o uguale a (1-0)*M ovvero M, ed essendo M una costante appositamente scelta molto elevata sarà dunque ridondante perchè il numero di ore, qualsiasi esso sia, non supererà mai, per come è stata definita, M.

Ore non consecutive per classe

```
subject to ore_non_consecutive{c in CLASSI, g in GIORNI,
                m in MATERIE, p in PROFESSORI,
                h in ORE: h+1 in ORE &&  (m,p) in CATTEDRE} :
                sum{j in h+1..5, mat in MATERIE : (mat,p) in CATTEDRE}
                x[c,mat,p,g,j] <= (1-x[c,m,p,g,h])*M + x[c,m,p,g,h+1]*M;
```

Questo vincolo rappresenta il fatto che un professore non possa avere due ore in una classe che siano non consecutive, indipendentemente dalla materia che insegna. Analogamente a quanto descritto in precedenza si utilizza la *Big M Notation* per scrivere il vincolo: per ogni classe, giorno, materia e professore la somma delle ore successive a quella presa inizialmente in considerazione, del professore preso in considerazione deve essere minore o uguale a zero nel caso in cui nell'ora successiva a quella considerata il professore non abbia lezione nella classe selezionata; in caso contrario la somma deve essere inferiore ad M costante elevata e quindi vincolo ridondante.

Giorni lavorativi

```
subject to giorni_lavorativi{p in PROFESSORI}:
        sum{g in GIORNI} gl[p,g] <=5;
```

Questo vincolo impone il fatto che i professori lavorino al massimo cinque giorni sui sei disponibili.

3.5 Obiettivi

In quest'ultima parte andremo a definire gli obiettivi del problema, analizzando come fatto per i vincoli, obiettivo per obiettivo.

Giorni lavorativi

```
minimize obj_giorni_lavorativi{p in PROFESSORI}:
        sum{g in GIORNI} gl[p,g];
```

Questo obiettivo vuole andare a minimizzare i giorni lavorativi per ogni professore, la logica è la stessa sulla quale si basa il vincolo relativo ai giorni lavorativi massimi dei professori, ma in questo caso vogliamo minimizzare il più possibile questa somma.

Giorni lavorativi in giorni liberi

```
minimize lezioni_in_giorni_liberi{p in PROFESSORI}:
        sum{g in GIORNI:
                (p,g) in GIORNI_LIBERI} gl[p,g];
```

Riprendendo la definizione del problema, vediamo come vi sia la possibilità per ogni professore di inserire giorni liberi. Il sistema deve quindi cercare di rispettare il più possibile la volontà dei professori, evitando di inserire lezioni dei professori in questi giorni. Per questo motivo, per ogni professore, la somma dei giorni lavorativi, considerando solo i giorni in cui è stata fatta richiesta di indisponibilità deve essere minima.

Ore buche

```
minimize ore_di_fila{g in GIORNI, p in PROFESSORI,h in ORE:
                    (g,h) in LEZIONI && h+1 in ORE}:
                    sum{m in MATERIE,c in CLASSI,j in ORE: (g,j) in LEZIONI
                    && j>h && (m,p) in CATTEDRE}
                    (x[c,m,p,g,j] - x[c,m,p,g,h+1] * M);
```

Questo vincolo è forse il più utile e quello che più si avvicina all'idea di orario accettabile. Si sta cercando di eliminare il più possibile le ore buche tra una lezione e l'altra per ogni professore. Per ogni giorno, per ogni professore se l'ora successiva a quella presa in considerazione è buca, allora devo minimizzare la somma di tutte le ore successive; se invece l'ora successiva a quella presa in considerazione non è buca, la somma delle ore successive deve essere minore di M costante molto elevata.

Ore libere dei professori

```
minimize ore_libere{p in PROFESSORI} :
        sum{(g,h) in LEZIONI,c in CLASSI, m in MATERIE :
                (m,p) in CATTEDRE &&
                (p,g,h) in ORE_LIBERE} x[c,m,p,g,h];
```

Questo obiettivo stabilisce che il sistema deve cercare di non assegnare lezioni ad un professore nelle ore per le quali ha fatto richiesta di non lavorare. Di conseguenza, per ogni professore, il numero delle ore di lezione nelle ore libere dovrà essere minimizzato.