UNIVERSITÀ
DI PARMA

Modello matematico e risoluzione AMPL per il problema "packing di rettangoli"

Corso "Ricerca Operativa" a cura del Professore Locatelli Marco

Realizzato da Giuseppe Ricciardi, Signorelli Antonio

Università degli studi di Parma, febbraio 2023

Contents

1 Introduzione: il problema del packing di rettangoli

È dato un contenitore rettangolare la cui base ha la lunghezza fissata B, mentre possiamo scegliere la lunghezza H della sua altezza.
Abbiamo un certo numero N di rettangoli con lati che misurano l e h e che possono essere orientati verticalmente oppure orizzontalmente. Dobbiamo stabilire l'altezza minima del contenitore in modo da disporre i rettangolini nel contenitore senza sovrapposizioni.
Si tratta di un problema di minimizzazione geometrica bidimensionale, per la risoluzione del problema le due dimensioni del contenitore sono state utilizzate come assi di riferimento: la lunghezza rappresenta l'asse delle x mentre l'altezza l'asse delle y.

2 Il modello matematico

2.1 Dati del problema

I dati "noti a priori" del problema sono:

- La base del contenitore, indicata con B.
- Il numero di rettangoli, indicato con N.
- La lunghezza dei rettangoli da inserire nel contenitore, indicata con l.
- L'altezza dei rettangoli da inserire nel contenitore, indicata con h.

2.2 Variabili del problema

Per quanto riguarda le variabili utilizzate:

- L'altezza del contenitore, indicata con H. È una *variabile continua*.
- x_i rappresenta la posizione orizzontale del vertice in basso a sinistra del rettangolo i-esimo. È una *variabile continua*.
- y_i rappresenta la posizione verticale del vertice in basso a sinistra del rettangolo i-esimo. È una *variabile continua*.
- $rotazione_i$, è una *variabile binaria*: vale 1 se il rettangolo i-esimo viene ruotato di 90° (scambiando l'altezza con la lunghezza), 0 altrimenti.
- $lunghezza_i$, è una *variabile continua*: assume valore pari ad l se il rettangolo i-esimo non è stato ruotato, altrimenti assume valore h.
- $larghezza_i$, è una *variabile continua*: assume valore pari ad h se il rettangolo i-esimo non è stato ruotato, altrimenti assume valore l.

- *controllo_orizzontale*$_{i,j}$ (abbreviato in $c\_o_{i,j}$), è una *variabile binaria*: è usata per l'attivazione del vincolo di controllo della sovrapposizione orizzontale tra due rettangoli. Vale 1 se è presente della sovrapposizione orizzontale, 0 altrimenti.

- *controllo_verticale*$_{i,j}$ (abbreviato in $c\_v_{i,j}$), è una *variabile binaria*: è usata per l'attivazione del vincolo di controllo della sovrapposizione verticale tra due rettangoli, vale 1 se è presente della sovrapposizione verticale oppure 0 nel caso contrario.

2.3 Vincoli del problema

Dalla formulazione del problema vengono ricavati i seguenti vincoli:

1. la somma della coordinata del vertice in basso a sinistra x_i sommato alla sua larghezza w_i, corrispondente al vertice in basso a destra del rettangolo, non deve superare la larghezza del contenitore B.

2. la somma della coordinata del vertice in basso a sinistra y_i sommato alla sua altezza h_i, corrispondente al vertice in alto a sinistra del rettangolo, non deve superare l'altezza del contenitore H.

3. La variabile $lunghezza_i$ deve corrispondere al giusto valore in base alla rotazione.

4. Stesso discorso del punto precedente viene applicato per la variabile $altezza_i$.

5. Per la non sovrapposizione ci sono 4 casi da considerare:

   - il rettangolo i si trova a sinistra rispetto al rettangolo j: $x_i + l_i \leq x_j$ oppure ("OR")
   - il rettangolo i si trova a destra rispetto al rettangolo j: $x_j + l_j \leq x_i$ oppure ("OR")
   - il rettangolo i si trova in basso rispetto al rettangolo j: $y_i + h_i \leq y_j$ oppure ("OR")
   - il rettangolo i si trova in alto rispetto al rettangolo j: $y_j + h_j \leq y_i$

   Bisogna però tenere in conto della rotazione dei rettangoli, per questo motivo sono state introdotte le variabili $lunghezza_i$ e $larghezza_i$.

Per modellare i vincoli nel punto 5 sono state utilizzate le variabili binarie $c\_o_{i,j}$ e $c\_v_{i,j}$, la lunghezza del contenitore B e la sua altezza H come limiti superiori espliciti.

2.4 Obiettivo del problema

L'obiettivo del problema consiste nel minimizzare il valore dell'altezza H del contenitore, ovvero:

$$\min(H)$$

2.5 Formulazione del problema

Il modello matematico segue l'ordine dei vincoli elencati nella sezione 2.3

$$\min \quad H \tag{1}$$

soggetto a

$$x_i + lunghezza_i \leq W \quad \forall i \in N \tag{2}$$

$$y_i + altezza_i \leq H \quad \forall i \in N \tag{3}$$

$$lunghezza_i = l_i(1 - rotazione_i) + h_i * rotazione_i \quad \forall i \in N \tag{4}$$

$$altezza_i = h_i(1 - rotazione_i) + l_i * rotazione_i \quad \forall i \in N \tag{5}$$

$$x_i + lunghezza_i \leq x_j + B * c\_o_{i,j} \quad \forall i, j \in N, i \leq j \tag{6}$$

$$x_j + lunghezza_j \leq x_i + B * c\_o_{j,i} \quad \forall i, j \in N, i \leq j \tag{7}$$

$$y_i + altezza_i \leq y_j + H * c\_v_{i,j} \quad \forall i, j \in N, i \leq j \tag{8}$$

$$y_j + altezza_j \leq y_i + H * c\_v_{j,i} \quad \forall i, j \in N, i \leq j \tag{9}$$

$$c\_o_{i,j} + c\_o_{j,i} + c\_v_{i,j} + c\_v_{j,i} \leq 3 \quad \forall i, j \in N i \leq j \tag{10}$$

$$c\_o_{i,j}, c\_v_{i,j}, rotazione_i \in \{0, 1\} \quad \forall i, j \in N, i \neq j \tag{11}$$

$$x_i, y_i, l_i, h_i, lunghezza_i, altezza_i \in \mathbb{R} \quad \forall i \in N \tag{12}$$

Le equazioni dalla 6 alle 10 modellano il vincolo di non sovrapposizione dei rettangoli (punto 5, sezione 2.3).
La scelta di utilizzare l'insieme dei rettangoli per cui $i \leq j$ è stata presa per risparmiare memoria: se i rettangoli i e j non si sovrappongono $\rightarrow$ i rettangoli j ed i non si sovrappongono.

3 Implementazione del modello mediante AMPL

Il risolutore utilizzato per l'algoritmo AMPL è *gurobi*. Il progetto è composto da 3 file:

- *packrettangoli.mod* contiene la rappresentazione del modello matematico, accuratamente commentato.

- *packrettangoli.dat* fornisce al risolutore dei dati da poter utilizzare.

- *packrettangoli.run* include i comandi per automatizzare l'esecuzione del risolutore e produce sul terminale una stampa del valore ottimo del problema e la posizione dei rettangoli (coordinate x ed y del vertice in basso a sinistra ed il valore della variabile binaria *rotazione*).

4 Risultati

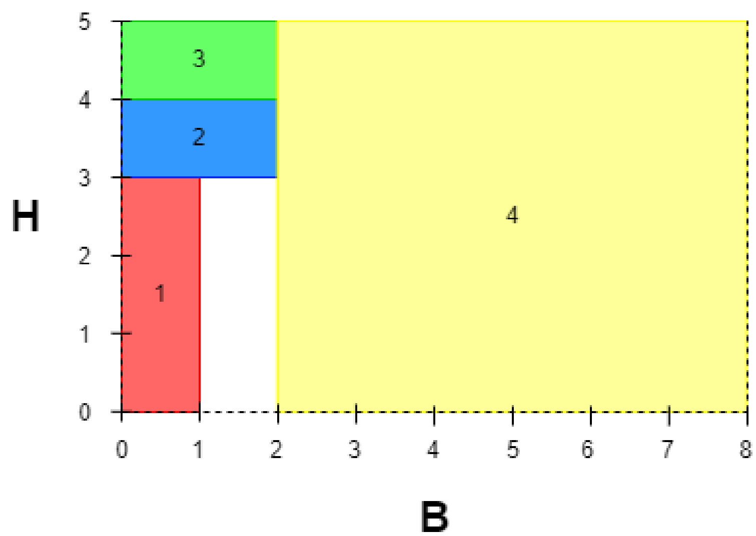
Di seguito alcune rappresentazioni grafiche dei risultati prodotti dall'algoritmo proposto:

INPUT

```
set RETTANGOLI := 1 2 3 4 ;
param B := 8; #base contenitore
param l := #lunghezze rettangoli
1 3
2 1
3 2
4 5;
param h := #altezze rettangoli
1 1
2 2
3 1
4 6;
```

OUTPUT

```
Minimum height: 5
Rectangle coordinates:
Rect 1: x=0, y=0, ruotato=1
Rect 2: x=0, y=4, ruotato=1
Rect 3: x=0, y=3, ruotato=0
Rect 4: x=2, y=0, ruotato=1
```

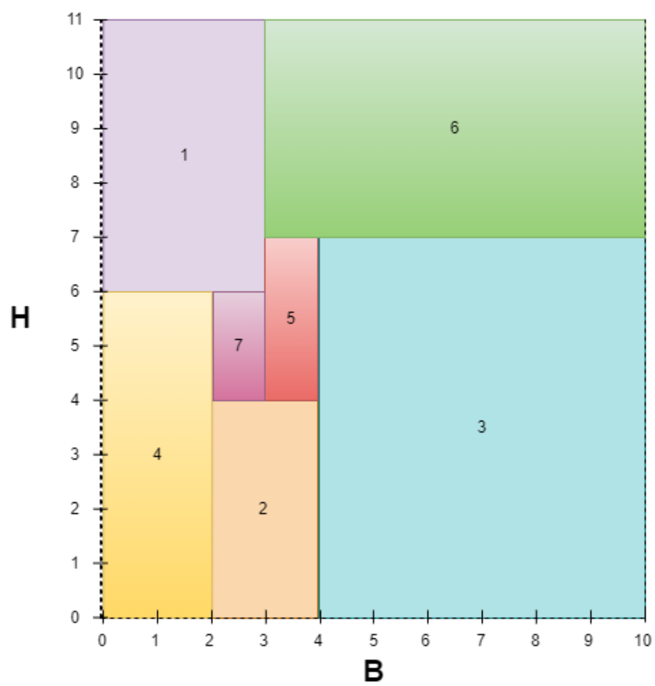


INPUT

```
set RETTANGOLI := 1 2 3 4 5 6 7 ;
param B := 10; #base contenitore
param l := #lunghezze rettangoli
1 3
2 4
3 7
4 6
5 3
6 4
7 2;
param h := #altezze rettangoli
1 5
2 2
3 6
4 2
5 1
6 7
7 1;
```

OUTPUT

```
Minimum height: 11
Rectangle coordinates:
Rect 1: x=0, y=6, ruotato=0
Rect 2: x=2, y=0, ruotato=1
Rect 3: x=4, y=0, ruotato=1
Rect 4: x=0, y=0, ruotato=1
Rect 5: x=3, y=4, ruotato=1
Rect 6: x=3, y=7, ruotato=1
Rect 7: x=2, y=4, ruotato=1
```

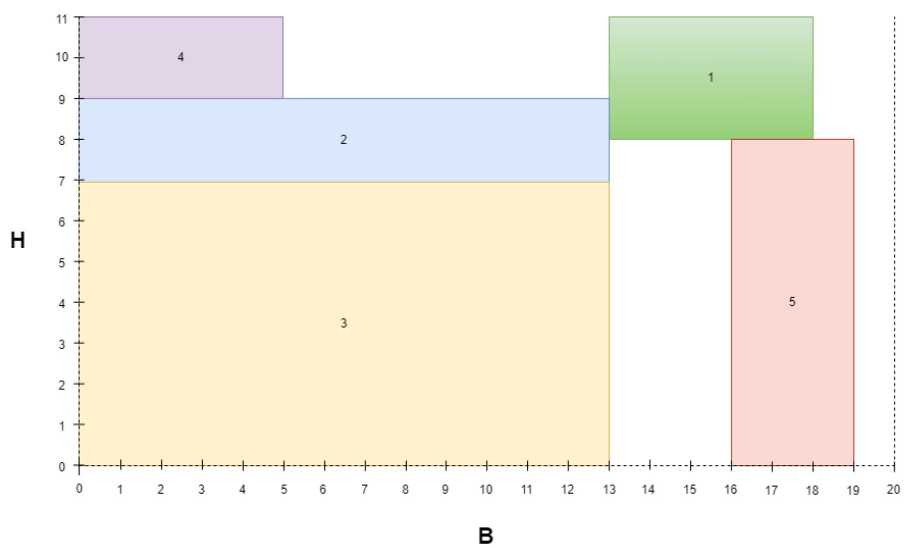


INPUT

```
set RETTANGOLI := 1 2 3 4 5 ;
param B := 20; #base contenitore
param l := #lunghezze rettangoli
1 3
2 13
3 7
4 5
5 8;
param h := #altezze rettangoli
1 5
2 2
3 16
4 2
5 4;
```

OUTPUT

```
Minimum height: 11
Rectangle coordinates:
Rect 1: x=13, y=8, ruotato=1
Rect 2: x=0, y=7, ruotato=0
Rect 3: x=0, y=0, ruotato=1
Rect 4: x=0, y=9, ruotato=0
Rect 5: x=16, y=0, ruotato=1
ampl:
```



INPUT

```
set RETTANGOLI := 1 2 3;
param B := 10; #base contenitore
param l := #lunghezze rettangoli
1 3
2 4
3 7;
param h := #altezze rettangoli
1 4
2 2
3 6;
```

OUTPUT

```
Minimum height: 7
Rectangles coordinates:
Rect 1: x=1, y=2, ruotato=0
Rect 2: x=0, y=0, ruotato=0
Rect 3: x=4, y=0, ruotato=1
```