

# **Implementazione MATLAB e test di problemi di ottimizzazione lineare con obiettivi multipli e priorità**

## **Tesi di Laurea in Ingegneria Informatica**

**Candidato**

Gregorio Maria Manduzio

**Relatore**

Prof. Marco Cococcioni

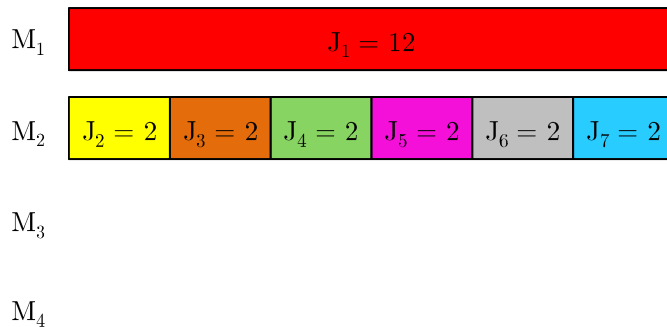


UNIVERSITÀ DI PISA

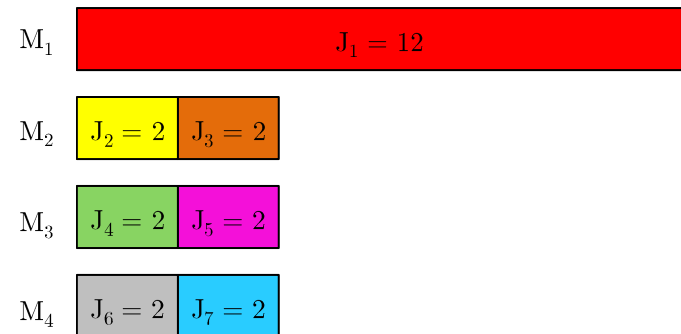
- Dati  $n$  lavori, ciascuno con il proprio tempo di completamento, da far svolgere ad  $m$  macchinari identici, a quanto ammonta il minimo tempo necessario per svolgerli tutti?
- Quale strumento utilizzare per risolvere tale problema?
- Come ottenere delle soluzioni bilanciate e più resistenti ad eventuali malfunzionamenti dei macchinari?

- La soluzione si trova risolvendo il problema del makespan scheduling mono-obiettivo.
- Un qualsiasi software in grado di risolvere problemi di Programmazione Lineare Interi, come CPLEX, Gurobi oppure MATLAB (utilizzato in questa tesi).
- Risolvendo nuovamente il problema del makespan scheduling ma in versione lessicografica.

$$\left\{ \begin{array}{ll} \min C_{\max} & \\ C_{\max} \geq C_i & M_i \in \mathcal{M} \\ C_i = \sum_{j=1}^n x_{i,j} \cdot p_j & M_i \in \mathcal{M} \\ \sum_{i=1}^m x_{i,j} = 1 & J_j \in \mathcal{J} \\ x_{i,j} \in \{0, 1\} & J_j \in \mathcal{J}, M_i \in \mathcal{M} \end{array} \right.$$



$$\left\{ \begin{array}{ll} \text{lexmin } C_1, \dots, C_m & \\ C_i \geq C_{i+1} & M_i \in \mathcal{M} \setminus M_m \\ \sum_{q=1}^{i-1} C_q + (m-i+1) \cdot C_i \geq \sum_{j=1}^n p_j & M_i \in \mathcal{M} \\ i \cdot C_i + \sum_{q=i+1}^m C_q \leq \sum_{j=1}^n p_j & M_i \in \mathcal{M} \\ C_i = \sum_{j=1}^n x_{i,j} \cdot p_j & M_i \in \mathcal{M} \\ \sum_{i=1}^m x_{i,j} = 1 & J_j \in \mathcal{J} \\ x_{i,j} \in \{0, 1\} & J_j \in \mathcal{J}, M_i \in \mathcal{M} \end{array} \right.$$



Sono stati prodotti gli script `makespan_singolo.m` e `makespan_lessicografico.m` (reperibili [qui](#)) per risolvere le due versioni del problema. Il secondo implementa l'algoritmo iterativo

- 1:  $v_1^* = \min\{C_1 : (\vec{x}, \vec{C}) \in \mathcal{S}\}$
- 2: **for**  $i = 2, \dots, m$  **do**
- 3:      $v_i^* = \min\{C_i : x \in \mathcal{S}, C_1 = v_1^*, \dots, C_{i-1} = v_{i-1}^*\}$
- 4: Restituisci il risultato ottenuto all'ultima iterazione

Se ne apprezzano le differenze fornendo in input tempi di completamento volutamente sbilanciati, ad esempio il problema nella slide precedente

```
>> makespan_singolo(4, [12 2 2 2 2 2 2])
```

MAKESPAN

```
C1 = 12
C2 = 6
C3 = 0
C4 = 6
```

ASSEGNAMENTO

|              | Job1 | Job2 | Job3 | Job4 | Job5 | Job6 | Job7 |
|--------------|------|------|------|------|------|------|------|
| Macchinario1 | 1    | 0    | 0    | 0    | 0    | 0    | 0    |
| Macchinario2 | 0    | 0    | 1    | 0    | 1    | 0    | 1    |
| Macchinario3 | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Macchinario4 | 0    | 1    | 0    | 1    | 0    | 1    | 0    |

SOLUZIONE

12

```
>> makespan_lessicografico(4, [12 2 2 2 2 2 2])
```

MAKESPAN

```
C1 = 12
C2 = 4
C3 = 4
C4 = 4
```

ASSEGNAMENTO

|              | Job1 | Job2 | Job3 | Job4 | Job5 | Job6 | Job7 |
|--------------|------|------|------|------|------|------|------|
| Macchinario1 | 1    | 0    | 0    | 0    | 0    | 0    | 0    |
| Macchinario2 | 0    | 0    | 0    | 0    | 1    | 0    | 1    |
| Macchinario3 | 0    | 0    | 1    | 0    | 0    | 1    | 0    |
| Macchinario4 | 0    | 1    | 0    | 1    | 0    | 0    | 0    |

SOLUZIONE

12

A questo link si trovano le soluzioni, ottenute con il software CPLEX, di alcune istanze di problema lessicografico. Prese le prime 10, si vuole misurare mediamente quanto si discostano da esse le soluzioni ottenute tramite lo script `makespan_lessicografico.m`, secondo la formula

$$\frac{(C_{max}^{MATLAB} - C_{min}^{MATLAB}) - (C_{max}^{CPLEX} - C_{min}^{CPLEX})}{C_{max}^{CPLEX} - C_{min}^{CPLEX}} \cdot 100$$

|            | $C_{max}^{MATLAB}$ | $C_{min}^{MATLAB}$ | $C_{max}^{CPLEX}$ | $C_{min}^{CPLEX}$ | Scostamento | Scostamento medio |
|------------|--------------------|--------------------|-------------------|-------------------|-------------|-------------------|
| Istanza 1  | 236.531            | 236.489            | 236.538           | 236.502           | 16,67%      | 2.73%             |
| Istanza 2  | 373.452            | 373.392            | 373.441           | 373.376           | -7,69%      |                   |
| Istanza 3  | 435.170            | 435.104            | 435.175           | 435.111           | 3,13%       |                   |
| Istanza 4  | 3.926.512          | 3.926.360          | 3.926.812         | 3.926.077         | -79,32%     |                   |
| Istanza 5  | 8.820.121          | 8.818.229          | 8.820.118         | 8.818.097         | -6,38%      |                   |
| Istanza 6  | 9.113.174          | 9.111.172          | 9.112.867         | 9.111.424         | 38,74%      |                   |
| Istanza 7  | 83.430.779         | 83.421.528         | 83.432.430        | 83.420.967        | -19,30%     |                   |
| Istanza 8  | 172.651.777        | 172.619.086        | 172.650.095       | 172.621.523       | 14,42%      |                   |
| Istanza 9  | 195.439.916        | 195.410.873        | 195.434.984       | 195.414.317       | 40,53%      |                   |
| Istanza 10 | 1.532.534.626      | 1.532.367.946      | 1.532.517.792     | 1.532.385.975     | 26,45%      |                   |