



Contexte

Job Scheduling

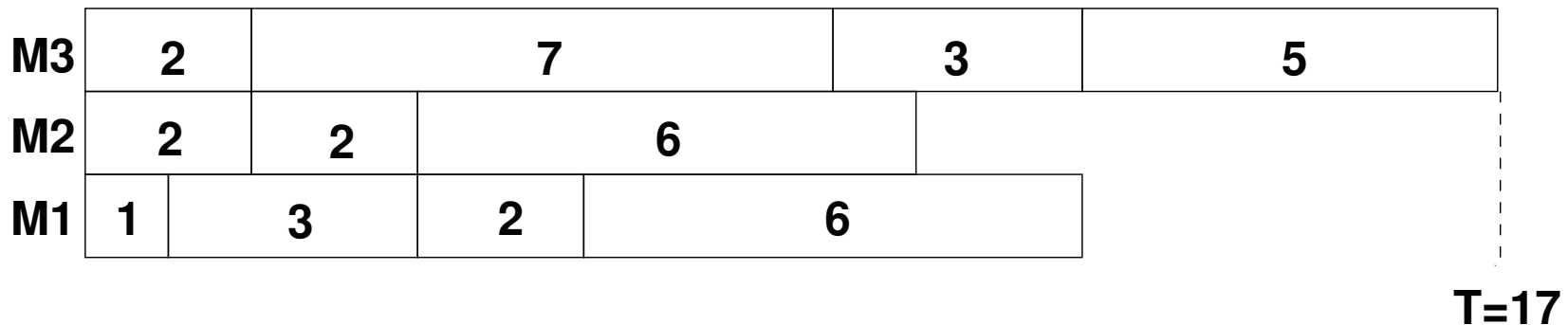
- n travaux (jobs)
- chacun ayant une durée d_i de réalisation
- jobs à distribuer sur m machines
- toutes les machines sont identiques
- **Objectif** : minimiser le temps (Min Makespan)

○
○○○
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Exemple

- $n = 11$ tâches
- $m = 3$ machines
- durées :

Tâche	1	2	3	4	5	6	7	8	9	10	11
Durée	2	7	1	3	2	6	2	3	6	2	5





Discussion sur le Min Makespan

- Problème “difficile”
- Terme officiel : NP-complet
- Signification : pour des grandes valeurs de n et m , pas d’algo à la fois
 - rapide (temps d’exécution “raisonnable”)
 - toujours optimal (makespan minimum trouvé)
- Rem : étude exhaustive $\Rightarrow \sim m^n$ possibilités
- Solutions : heuristiques ou approximations
- Ici : approximations (2 algos)



List Scheduling Algorithm (LSA)

- Numérotter les n tâches dans un ordre arbitraire

Pour i de 1 à n

 affecter i à la première machine disponible

FinPour

○
○○○
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

List Scheduling Algorithm (LSA) – Exemple

Tâche	1	2	3	4	5	6	7	8	9	10	11
Durée	2	7	1	3	2	6	2	3	6	2	5

M3	1	3	2	3	2						
M2	7					6					
M1	2	2	6				5				



List Scheduling Algorithm (LSA)

Propriété

LSA est un algorithme de 2-approximation

Signification : durée obtenue **toujours ≤ 2 durée optimale**



List Scheduling Algorithm (LSA) = 2-approximation

- T_{opt} = le temps optimal recherché
- T_{LSA} = le temps trouvé par LSA
- Bornes inférieures pour T_{opt} :
 - $T_{opt} \geq \max_{1 \leq i \leq n} d_i$ (maximum)
 - $T_{opt} \geq \frac{\sum_{i=1}^n d_i}{m}$ (moyenne)
 - en observant la machine qui travaille jusqu'à T_{LSA} , on a

$$T_{LSA} \leq \frac{\sum_{i=1}^n d_i}{m} + \max_{1 \leq i \leq n} d_i$$

Conclusion :

$$T_{LSA} \leq 2 \cdot T_{opt}$$



Largest Processing Time (LPT)

- Numéroté les n tâches dans un l'ordre **décroissant** des longueurs

Pour i de 1 à n

 affecter i à la première machine disponible

FinPour

○
○○○
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Largest Processing Time (LPT) - Exemple

Tâche	2	6	9	11	4	8	1	5	7	10	3
Durée	7	6	6	5	3	3	2	2	2	2	1

M3	6	3	2	2		
M2	6	5	2			
M1	7	3	2	1		