
Ordonnancement sur machines parallèles

SIO - Laboratoire 2

Nicolas Crausaz & Maxime Scharwath

21.12.2022

Table des matières

Modélisation mathématique	3
Définition des variables de décision	3
Définition de la fonction objectif	3
Définition des contraintes	3

Modélisation mathématique

Le contexte de ce travail est d'effectuer la modélisation d'un problème d'ordonnancement, consistant à trouver un plan d'ordonnancement permettant de répartir n tâches, devant toutes être réalisées, en disposant de m machines différentes (travail en parallèle), cela en minimisant le retard moyen de l'exécution des tâches.

Nous connaissons les constantes suivantes:

Pour chaque tâche $i = 1, \dots, n$:

- Sa date de disponibilité (date de début au plus tôt, release date) r_i
- Sa date d'échéance (date de fin au plus tard, due date) d_i
- Son temps d'exécution (durée de réalisation, processing time) p_i

On supposera, sans perte de généralité, que la plus petite date de disponibilité est égale à 0 et que les données sont cohérentes et vérifient, en particulier, $r_i \geq 0$ et $p_i \geq 0$ pour chaque tâche $i = 1, \dots, n$.

Définition des variables de décision

Nous définissons les variables de décision suivantes:

x_i : date de début de l'exécution de la tâche i , $i = 1, \dots, n$.

L'exécution de chaque tâche i ne peut commencer avant sa date de disponibilité r_i :

$x_i \geq r_i$, pour tout i

On définit le retard (tardiness) T_i de la tâche i par - $T_i = \max_{i=1, \dots, n} (0, x_i + p_i - d_i)$

- e_{ij} : indique si la tâche i s'exécute sur la machine j , $i = 1, \dots, n$, $j = 1, \dots, m$

$$e_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est exécuté sur la machine } j \\ 0 & \text{sinon.} \end{cases}$$

Définition de la fonction objectif

$$\text{Minimiser } z = \frac{1}{n} \sum_{i=1}^n T_i$$

Définition des contraintes

- Une tâche n'est exécuté qu'une seule fois et sur une unique machine

$$\sum_{i=1}^n e_{ij} = 1 \quad j = 1, \dots, m$$

L'exécution de chaque tâche ne peut commencer avant sa date de disponibilité.

- La tâche suivante doit être exécuté après la date de fin + le retard de la tâche précédente si les tâches i et j sont sur la même machine

pour chaque paire $\{i, j\}$ de tâches différentes Si elle sont sur la même machine , soit la tâche i termine son exécution avant que la tâche j ne débute la sienne soit c'est l'inverse

Non négativité de T_i, X_i