Ordonnancement sur machines parallèles

SIO - Laboratoire 2

Nicolas Crausaz & Maxime Scharwath

10.01.2023

Table des matières

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 réalisée, en disposant de m machines différentes (travail en parrallèle), cela en minimisant le retard moyen de l'éxécution des tâches.

Nous connaissons les **constantes** suivantes:

Pour chaque tâche $i = 1, ..., n$:

- 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 $i = 1, .., n$.

Définition des variables de décision

Pour trouver un ordonnancement de n tâches sur m machines, il va nous falloir répartir ces tâches de manière à ce qu'un tâche $i = 1, ..., n$ soit traitée sur une unique machine $k = 1, ..., m$. Pour cela il nous faut donc définir les variables de décision suivantes:

Une variable binaire:

$$U_{ik} = \begin{cases} 1 & \text{si la tâche i est executée sur la machine k} \\ 0 & \text{sinon} \end{cases}$$

ainsi qu'une seconde variable binaire:

$$x_{ik} = \begin{cases} \text{Date de début de la tâche i sur la machine k} & \text{si i est executé sur k} \\ 0 & \text{sinon} \end{cases}$$

Définition des variables auxiliaires

Afin de pouvoir connaître le retard de chaque tâche sur sa machine respective, nous définissons la variable T_i, correspondant au retard (tardiness) de la tâche i $i = 1, ..., n$ lors de son exécution:

$$T_i = \max(0, \sum_{k=1}^{m}(x_{ik} + p_i - d_i))$$

On introduit une variable auxiliaire binaire y_{ij}, pour chaque paire $\{i, j\}$, $i, j = 1, ..., n$ de tâches différentes sur une même machine et dont l'interprétation est:

$$y_{ij} = \begin{cases} 1 & \text{si la tâche est executé avant la tâche j sur la même machine} \\ 0 & \text{si i et j ne sont pas exécutés sur la même machine} \\ 0 & \text{sinon} \end{cases} \qquad i = 1, ..., n, j = 1, ...n$$

qui se linéarise de la manière suivante:

$$y_{ij} = \begin{cases} x_{ik} + p_i - x_{jk} <= M * (1 - y_{ij}) + M * (1 - U_{ik}) + M * (1 - U_{jk}) \\ x_{ik} + p_i - x_{jk} <= M * (1 - y_{ij}) + M * (U_{ik}) + M * (1 - U_{jk}) \\ x_{ik} + p_i - x_{jk} <= M * (1 - y_{ij}) + M * (1 - U_{ik}) + M * (1 - U_{jk}) \\ x_{ik} + p_i - x_{jk} <= M * (1 - y_{ij}) + M * U_{ik} + M * U_{jk} \\ x_{ik} + p_i - x_{jk} <= M * y_{ij} + M * (1 - U_{ik}) + M * (1 - U_{jk}) \\ x_{ik} + p_i - x_{jk} <= M * y_{ij} + M * (Uik) + M * (1 - U_{jk}) \\ x_{ik} + p_i - x_{jk} <= M * y_{ij} + M * (1 - U_{ik}) + M * U_{jk} \\ x_{ik} + p_i - x_{jk} <= M * y_{ij} + M * U_{ik} + M * U_{jk} \end{cases} \qquad \text{i,j = 1,...,n, k = 1,...m}$$

La constante M devant avoir une valeurs suffisamment grande, nous la définissons à:

$$M = \max(r_i) + \sum_{i=1}^{n} p_i$$

correspondant à la date de disponibilité de la tâche s'exécutant le plus tard additionné à la somme du temps d'exécution de toutes les tâches.

Définition de la fonction objectif

Nous recherchons un ordonnancement répartissant n tâches sur m machines différentes, qui minimum le **retards moyen** de l'exécution des tâches.

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

Définition des contraintes

— Une tâche n'est executé qu'une seule fois et sur une unique machine

sum Uik = 1, i 1 à n et l 1 à m

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

Xik >= ri * Uik, pour k = 1 à m

— La tâche ne s'exécute pas sur un autre machine que celle prévue

(xik <= M * Uik) ou mettre la grosse disjonction d'en haut ici aussi

T_i, x_i >= 0