Etudes de problèmes d'ordonnancement
ValDom

MJ. HUGUET
homepages.laas.fr/huguet

1 Problème d'ordonnancement étudié : Job Shop

1.1 Définition du problème et modélisation A lire et comprendre

Le Job Shop est un problème d'ordonnancement dans lequel on dispose :

- d'un ensemble J de n Jobs (ou travaux) décomposé en une succession de tâches (ou activités ou opérations). Pour un job j, on note n_j le nombre de tâches.
- d'un ensemble R de m ressources disjonctives pour réaliser ces tâches.

Toute opération i d'un job j est notée (i, j). Avec $1 \leq i \leq n_j$ et $n_j \leq m$. Au maximum le problème comporte $n * m$ opérations. On note T l'ensemble des opérations.
On dit qu'une ressource est **disjonctive** si elle est disponible en un exemplaire unique et ne pouvant réaliser qu'une seule tâche à la fois. On supposera que les opérations d'un job donné utilise toutes des ressources différentes de R : ainsi pour chaque job il y a au maximum m opérations différentes.
Les contraintes sont les suivantes :

- toute opération (i, j) d'un job j nécessite une unique ressource donnée, notée k, pendant une durée $p_{(i,j)}$,
- chaque ressource k est disponible en un seul exemplaire,
- dans tout job j, l'opération (i, j) est successeur de l'opération $(i-1, j)$, $\forall i \in [2, n_j]$
- les jobs peuvent débuter à la date 0.

L'objectif est de minimiser la durée totale de l'ordonnancement aussi appelée *makespan*. La résolution de ce problème nécessite donc de déterminer la date de début de chaque opération.
Pour modéliser ce problème, on peut utiliser des variables entières représentant les dates de début $st_{(i,j)}$ des différentes opérations (i, j). On introduit dans le modèle deux opérations fictives, notées S et F représentant le début et la fin de l'ordonnancement. Ainsi, S précède chacune des premières opérations des jobs et F succède à chacune des dernières opérations des jobs.
On note T_k l'ensemble des opérations devant être effectuées par une ressource k.
Le modèle s'écrit alors :

$$Min(st_F) \tag{1}$$
$$st_{(i-1,j)} + p_{(i-1,j)} \leq st_{(i,j)},\ \forall j \in J, \forall i \in [2, n_j] \tag{2}$$
$$(st_{(i,j)} + p_{(i,j)} \leq st_{(i',j')})\ \vee\ (st_{(i',j')} + p_{(i',j')} \leq st_{(i,j)}),\ \forall (i,j),(i',j') \in T_k, \forall k \in R \tag{3}$$
$$st_{(i,j)} \in \mathbb{N}^+,\ \forall (i,j) \in T \tag{4}$$

L'objectif (1) correspond à la minimisation de la date de début de la dernière opération du problème, c'est à dire que l'on cherche à minimiser la date de fin de l'ordonnancement ou sa durée totale. Les contraintes (2) expriment les précédences entre les opérations d'un même job. Les contraintes (3) traduisent les contraintes de partage de ressource. Une ressource étant disponible en quantité unitaire, toute paire de tâche nécessitant une même ressource doit donc être séquencée sur cette ressource. Les contraintes (4) spécifient que les dates de début de chacune des opérations sont positives.

Note : les contraintes (3) peuvent être linéarisées par l'introduction d'une constante ayant une très grande valeur (notée M) et de variables booléennes $x^k_{(i,j)(i',j')}$ représentant le séquencement de deux opérations sur une ressource k. $x^k_{(i,j)(i',j')}$ vaut 1 si (i, j) précède (i', j') et 0 sinon.

$$st_{(i,j)} + p_{(i,j)} + M.(x^k_{(i,j)(i',j')} - 1) \leq st_{(i',j')} \tag{5}$$
$$st_{(i',j')} + p_{(i',j')} - M.x^k_{(i,j)(i',j')} \leq st_{(i,j)} \tag{6}$$

Illustration. **A comprendre sur cet exemple qui servira aussi en TP** Soit un problème de Job Shop composé de 2 jobs et de 3 ressources. Pour chaque opération de chaque job, le tableau ci-dessous donne la ressource nécessaire et la durée de réalisation sur cette ressource.

J_1	$r_1, 3$	$r_2, 3$	$r_3, 2$
J_2	$r_2, 2$	$r_1, 2$	$r_3, 4$

Les contraintes (2), (3) et (4) de ce problème peuvent être représentées par le graphe de la figure 1 sur lequel on a introduit les opérations fictives S et F. Dans ce graphe, un arc entre deux sommets $st_{(i,j)}$ et $st_{(i',j')}$, représentent la contrainte $st_{(i',j')} \geq st_{(i,j)} + l_{(i,j)}^{(i',j')} \Leftrightarrow st_{i',j')} - st_{(i,j)} \geq l_{(i,j)}^{(i,j)}$ où $l_{(i,j)}^{(i',j')}$ est la valeur de la distance entre $st_{(i,j)}$ et $st_{i',j')}$ (pour le problème étudié, cette distance vaut $p_{(i,j)}$). Les arcs représentés en trait plein représentent les contraintes (2) et (4) : ils traduisent les contraintes temporelles du problèmes et sont appelés arcs conjonctifs. Les arcs représentés en pointillé correspondent aux contraintes (3) : ils traduisent les contraintes de partage de ressources ; ils sont appelés arcs disjonctifs.
Pour obtenir une solution admissible à un problème de Job Shop, il faut orienter chacun des arcs disjonctifs sans créer de cycle de longueur positive dans le graphe.

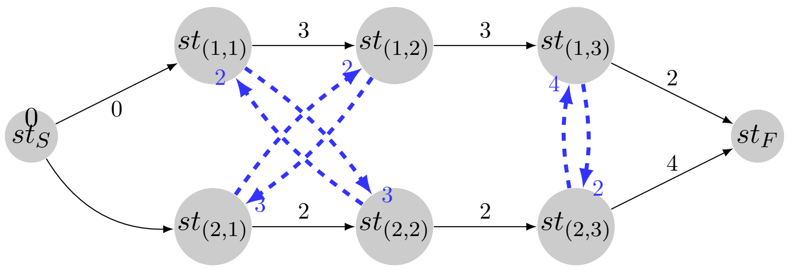

FIGURE 1 – Exemple de Job Shop

Le graphe de la figure 2 (sans cycle de longueur positive) représente une solution pour ce problème.

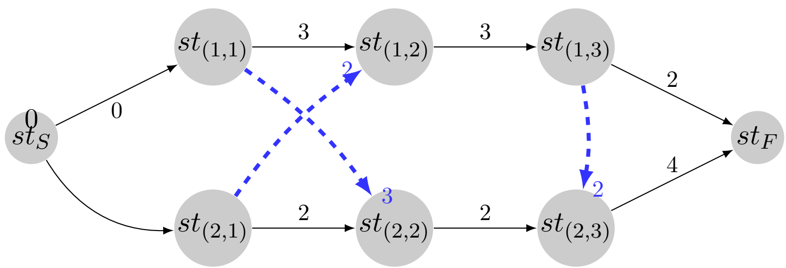

FIGURE 2 – Solution de l'exemple

Question. Quel est le coût de cette solution ? Comment est-il calculé ?
Indice : méthode PERT (durée minimale, dates au plus tôt, au plus tard, marges, ...).

Question. Une solution peut également se représenter sur un diagramme de Gantt. Tracez ce diagramme.

Ce problème "académique" a de nombreuses applications et variantes. Une étude récente [2] en présente une synthèse pour le secteur de l'industrie 4.0.

Vocabulaire. Les termes suivants sont équivalents : ressource=machine et opération=tâche=activité.

2 Exemple Optionnel

Soit un atelier réalisant 3 types de papier peint différents. Chaque type de papier est fabriqué sous forme d'un rouleau de papier continu qui passe sur plusieurs machines (chacune imprimant une couleur différente) L'ordre de passage (et la durée) sur les machines dépend du type de papier.

- Type 1 : Machine 1 (bleu, 45) puis Machine 3 (jaune, 15)
- Type 2 : Machine 2 (vert, 10), Machine 1 (bleu, 20) puis Machine 3 (jaune, 34)
- Type 3 : Machine 2 (vert, 17) puis Machine 3 (jaune, 28)

Une machine ne peut traiter qu'un seul type de papier à la fois et qu'un type de papier ne peut passer que sur une machine à la fois On souhaite réaliser ces 3 types de produits avec une durée minimale.

Question. Ce problème d'ordonnancement est un problème de Job-Shop. Donnez sa modélisation sous forme de graphe.

- proposez une borne inférieure de la durée totale
- proposez une solution (en fixant l'ordre de passage sur les machines sur le graphe)
- évaluez cette solution

3 Représentation d'une solution - Evaluation - Vérification

Cet exercice doit vous permettre de réfléchir :

- à une représentation des solutions ;
- à une procédure efficace d'évaluation de la fonction objectif d'une solution ;
- à une procédure efficace de vérification d'une solution

Plusieurs représentations d'une solution à un problème de Job Shop sont possibles. Pour le problème étudié (minimisation de la durée totale, on considère que les dates de début des opérations sont fixées au plus tôt. On consdière également que chaque job est composé de m opérations (il y a autant d'opérations que de ressources) et que les m opérations nécessitent toutes une ressource différente (ie. toutes les ressources seront nécessaires à un moment donné pour réaliser chaque job).

Optionnel - Pour avoir une synthèse de la variété de ces représentations consultez le document `Extrait-These_A-Caumond_2006`.

3.1 Représentation détaillée d'une solution

Une première représentation explicite d'une solution consiste à stocker la date de début de chaque opération de chaque job. Pour implémenter une telle représentation, nous allons utiliser une matrice de n lignes (la ligne i correspond au job i) et de m colonnes (la $j$eme colonne correspond à la $j$eme opération du job). Chaque case de la matrice fournit la date de début de l'opération correspondante.

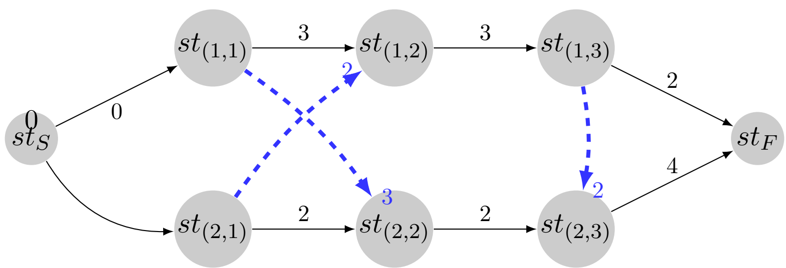

FIGURE 3 – Solution de l'exemple (avec numérotation des opérations)

Matrice représentant une solution :

num colonne	1	2	3
Job 1	0	3	6
Job 2	0	3	8

Question. Comment évaluer la durée totale à partir de cette matrice ?

Question. Comment vérifier que les contraintes de précédence et de ressources sont satisfaites ?

3.2 Représentation par ordre de passage sur les ressources

Une représentation plus synthétique consiste à donner pour chaque machine l'ordre dans lequel réaliser les différentes opérations.

r_1	$(1,1)$	$(2,2)$
r_2	$(2,1)$	$(1,2)$
r_3	$(1,3)$	$(2,3)$

Question. Comment passer de cette représentation par ordre sur les ressources à la représentation matricielle ?
Question. Testez votre méthode avec une autre solution. Par exemple :

r_1	$(2,2)$	$(1,1)$
r_2	$(2,1)$	$(1,2)$
r_3	$(1,3)$	$(2,3)$

3.3 Représentation par numéro de job.

Une autre représentation synthétique se base sur un ordre entre les opérations des jobs. Il serait possible de fournir un tableau contenant un ordre total de l'ensemble des opérations (une permutation sur l'ensemble des opérations). Par exemple $(1,1); (1,2); (2,1); (2,2); (1,3); (2,3)$. Cependant le nombre total de permutations serait plus important que le nombre de solutions intéressantes (pour l'objectif étudié). La variante considérée ici consiste à donner les ordres de passage des différents jobs (le même numéro de job est répété autant de fois qu'il y a d'opérations dans ce job).

Exemple :

num job	1	2	2	1	1	2

Question. Comment passer de cette représentation par ordre sur les ressources à la représentation matricielle ?
Question. Testez votre méthode avec une autre solution. Par exemple :

num job	1	1	2	2	1	2

En TP, la représentation détaillée d'une solution vous sera fournie. vous allez travailler avec la représentation synthétique basée sur l'ordre de passage par ressource. Vous devrez donc écrire un code pour passer de cette représentation synthétique à la représentation détaillée. Pour cela vous pourrez vous inspirez de la transformation de la représentation détaillée vares la représentation par numéro de job qui sera fournie.

4 Jeux de données

Les instances qui seront utilisées en TP sont disponibles sur le site :
`https://github.com/tamy0612/JSPLIB/tree/master/instances`.

L'exemple `ft06` est un exemple assez petit qui peut vous servir pour réfléchir à la représentation de solutions et à la conception de méthodes de résolution.

A faire. Prendre connaissance du format des instances de Job Shop.
A faire. Quelles caractéristiques pouvez-vous noter sur ces instances ?

5 Espace de recherche

La taille de l'espace de recherche dépend de la représentation choisie :

- avec la représentation par numéro de job, cette taille correspond au nombre de permutations du numéro de jobs avec répétition. Elle est égale à $\frac{(n \times m)!}{m!^n}$.
- avec la représentation par ordre de passage sur les ressources, cette taille correspond à toutes les permutations de tâches sur les ressources. Elle est égale à $(n!)^m$ où n est le nombre de jobs et m le nombre de machines (en supposant que les tâches d'un job passent par toutes les machines). Parmi ces permutations, certaines correspondent à des solutions non réalisables.
- avec la représentation détaillée par date de début de chaque tâche, cette taille dépend des valeurs possibles des dates de début et du nombre de tâches. Elle est égale à $(D_{max})^{n \times m}$ où D_{max} représente la durée maximale d'un ordonnancement (dans le pire cas, c'est la somme des durée de toutes les tâches). Parmi ces solutions, certaines ne sont pas réalisables.

Questions

- Calculer le nombre de solutions de l'espace de recherche associé à chaque représentation pour l'instance `ft06`
- En supposant que la génération et l'évaluation d'une solution prend une nano-seconde (pour chacune des représentations), quel serait le temps nécessaire pour explorer l'espace de recherche associé à chaque représentation ?
- Quelles conclusions en tirez-vous sur les représentations de solutions ?
- Quelles conclusions en tirez-vous sur les méthodes exhaustives ?

6 Méthodes de résolution

6.1 Heuristiques gloutonnes

Cet exercice doit vous permettre de réfléchir à des heuristiques gloutonnes (statique et/ou dynamique) pour le problème de Job Shop pour obtenir des solutions utilisant la représentation par ordre de passage sur les ressources et la représentation par numéro de job.

Dans la littérature des indicateurs tels que : début au plus tôt, fin au plus tard, durée opération/job, marge, charge sur les ressources, etc. sont souvent utilisés. Vous pouvez vous en inspirer.

Question. Consultez l'article [1] pour proposer une heuristique gloutonne pour chacune des représentations synthétiques. Prenez un exemple pour testez à la main vos heuristiques.

Question. Comment introduire de l'aléatoire dans vos heuristiques ?

Question. **sera traité en TP** Comment évaluer les résultats de vos heuristiques (quelles métriques peut-on utiliser) ?

6.2 Voisinages et méthode de descente

Consultez les documents (extraits de thèse) fournis sur ma page web pour proposer un/des voisinages afin de réaliser une méthode de descente pour chacune des représentations synthétiques. Vous aurez à le faire en TP.

Références

[1] Jacek Blazewicz, Wolfgang Domchke, and Erwin Pesch. The job shop scheduling problem : Conventional and new solution techniques. *European Journal of Operational Research*, 93(1) :1–33, Aug 1996.

[2] Jian Zhang, Guofu Ding, Yisheng Zou, Shengfeng Qin, and Jianlin Fu. Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing*, 30(4) :1809–1830, Apr 2019.