

Distanciel

Le problème MIN MAKESPAN

Octobre-Novembre 2022

Ce “projet distanciel” se compose de deux parties : un exercice et un projet de programmation. L’ensemble est à réaliser en binômes. Pour la partie “projet de programmation”, deux séances de TP d’1h20 y seront consacrées. Tout le reste se fera en distanciel. Ne pas hésiter à utiliser le forum Madoc “Échanges et questions autour du projet Min Makespan” pour échanger entre vous, et aussi pour me poser vos questions.

Un compte-rendu est à rédiger, qui contiendra donc deux parties. La Partie 1 contient vos réponses aux questions de l’exercice ; la Partie 2 contient un rapport de type “rapport de projet de programmation”. La Partie 2 ne doit pas dépasser 10 pages. Le format de fichier attendu pour ce compte-rendu (Parties 1 et 2) est le PDF.

L’ensemble des éléments de ce “projet distanciel” est à rendre sous la forme d’une archive NOM1-NOM2.zip. La décompression de cette archive doit produire un répertoire NOM1-NOM2 contenant tous les éléments de votre travail (réponses aux questions de l’exercice, rapport de projet, sources du projet, exécutable du projet, etc.) et *un fichier texte contenant les instructions détaillées de compilation et d’exécution du projet*.

Cette archive est à **déposer sur Madoc**, au plus tard le **Samedi 26 Novembre 2022 à 20h22**. Un malus sera appliqué à chaque fois qu’une des consignes données ci-dessus n’aura pas été respectée. Notamment, en cas de retard, le malus sera le suivant : -1 point par heure de retard (toute heure entamée étant comptée comme complète).

1 MIN MAKESPAN – Exercice

Le but de cet exercice est d’étudier le problème MIN MAKESPAN, dans lequel les durées des tâches sont toujours des entiers – voir pour cela les transparents du CM3 “Ordonnancement”, disponibles sur Madoc.

On rappelle notamment que dans les transparents du CM3, deux algorithmes sont présentés pour résoudre (de manière approchée) le problème MIN MAKESPAN : ces algorithmes s’appellent LSA et LPT.

On se donne l’instance I de MIN-MAKESPAN suivante : $m = 5$ et $n = 15$ tâches, dont les durées sont données ci-dessous.

Tâche	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Durée	6	5	4	7	5	4	3	6	7	2	7	8	10	5	9

1. Indiquer sur un quadrillage (comme dans les transparents du CM3 “Ordonnancement”) le résultat de l’algorithme LSA sur l’instance I , et clairement indiquer le temps $T_{LSA}(I)$ obtenu.
2. Indiquer sur un quadrillage le résultat de l’algorithme LPT sur l’instance I , et clairement indiquer le temps $T_{LPT}(I)$ obtenu.

On appelle $T_{opt}(I)$ le temps optimal pour l’instance I .

3. Démontrer que $T_{opt}(I) \leq 18$. Pour cela, donner une solution qui convient, toujours sous la forme d’un quadrillage.
4. Démontrer que $T_{opt}(I) \geq 18$.¹

1. Indice : des bornes inférieures pour T_{opt} vous sont données dans les transparents du CM3.

Pour une instance I de MIN MAKESPAN, on appelle d_{\max} la durée de la tâche la plus longue.

5. Fournir une instance I de MIN MAKESPAN à $m \geq 4$ machines et $n \geq 10$ tâches pour laquelle on a $T_{LPT}(I) > T_{LSA}(I)$. Justifier.
6. Donner une instance I' de MIN MAKESPAN à $m \geq 4$ machines et $n \geq 10$ tâches pour laquelle on a $T_{opt}(I') = d_{\max}$. Justifier.
7. Donner une instance I'' de MIN MAKESPAN à $m \geq 4$ machines et $n \geq 10$ tâches pour laquelle on a $T_{opt}(I'') = d_{\max}$ et $T_{LPT}(I'') > T_{opt}(I'')$. Justifier.

2 MIN MAKESPAN – Projet de Programmation

Dans ce projet, la programmation se fera dans le langage de votre choix. *Assurez-vous cependant que vos programmes compilent et fonctionnent sous Linux sur les machines du CIE.*

Le but de ce projet est d'étudier deux problèmes : le MIN MAKESPAN et le problème SEMI-FAST MIN MAKESPAN.

Le problème SEMI-FAST MIN MAKESPAN est une variante de MIN MAKESPAN, dans laquelle (1) le nombre de machines est pair, (2) les durées des tâches sont toutes paires, et (3) les machines numérotées de 1 à $\frac{m}{2}$ réalisent la tâche **deux fois plus vite** que les machines numérotées de $\frac{m}{2} + 1$ à m . Par exemple, sur une machine de numéro $1 \leq i \leq \frac{m}{2}$, une tâche de durée 10 sera réalisée en un temps de $\frac{10}{2} = 5$, alors qu'elle prendra un temps de 10 sur une machine de numéro $\frac{m}{2} + 1 \leq j \leq m$.

Pour ces deux problèmes, on vous demande d'implémenter trois algorithmes, de les tester et de fournir les résultats de ces tests à l'utilisateur. Les trois algorithmes à implémenter pour chaque problème sont décrits ci-dessous :

1. l'algorithme *List Scheduling Algorithm* (LSA) ;
2. l'algorithme *Largest Processing Time* (LPT) ;
3. un algorithme, que nous appellerons *Random Machine Assignment* (RMA) qui consiste, pour chaque tâche prise dans l'ordre fourni, à décider *au hasard* quelle machine va l'exécuter.

Remarque : pour ce projet, on ne s'intéresse qu'au *temps total de réalisation* calculé par les algorithmes. La réalisation précise (c'est-à-dire, quelles tâches affecter à quelle machine et dans quel ordre) n'est pas demandée. Ainsi, pour implémenter ces algorithmes, on peut juste manipuler deux tableaux d'entiers :

- un tableau $D[]$ qui va contenir les durées des tâches à affecter aux machines ;
- un tableau $M[]$, initialisé à 0, qui va représenter chacune des m machines (par exemple, $M[2]$ représente la machine numéro 2). Chaque case i de M contiendra, à tout moment de l'algorithme considéré, la *durée cumulée* des tâches affectées à la machine i .

Par exemple, si à un moment dans l'algorithme la machine 8 doit exécuter une tâche de longueur 12, alors on l'écrira simplement de la manière suivante :

- dans le cas du MIN MAKESPAN : $M[8] \leftarrow M[8] + 12$
- dans le cas du SEMI-FAST MIN MAKESPAN :
 $M[8] \leftarrow M[8] + 6$ si $m \geq 16$
 $M[8] \leftarrow M[8] + 12$ sinon

2.1 Travail demandé

Proposer un programme facile d'utilisation qui, par l'intermédiaire d'un menu, permet à l'utilisateur (1) de renseigner le problème de son choix, (2) de renseigner les 5 paramètres permettant de générer des instances et (3) de lire à l'écran les résultats de cette instance sur les trois algorithmes évoqués ci-dessus.

Instances d'entrée. L'utilisateur fournit 5 entiers m, n, k, d_{\min} et d_{\max} (attention : dans le cas du SEMI-FAST MIN MAKESPAN, il faudra s'assurer que m est pair). Il vous faut alors générer k instances différentes : pour chacune de ces k instances, on a m machines et n tâches ; les durées des tâches sont des entiers générés *au hasard*, avec la contrainte que (1) la durée d de chaque tâche vérifie $d_{\min} \leq d \leq d_{\max}$ et (2) uniquement dans le cas du SEMI-FAST MIN MAKESPAN, d est un entier pair.

Production des résultats. Pour chaque ensemble d’instances, les trois algorithmes LSA, LPT et RMA seront exécutés. Les résultats seront écrits à l’écran de la façon suivante :

```
ratio moyen LSA =
ratio moyen LPT =
ratio moyen RMA =
```

La valeur de `ratio moyen LSA` est la moyenne des ratios LSA sur les k exemples générés, sachant que le `ratio LSA` d’une instance se calcule de la façon suivante :

- on prend la valeur la plus grande entre `Borne inférieure` ‘‘maximum’’ et `Borne inférieure` ‘‘moyenne’’ : appelons cette valeur B ;
- `ratio LSA` est le résultat de la division de T_{LSA} (le résultat obtenu par l’algorithme LSA) par B .

Rem : ce ratio n’est donc pas forcément un entier.

La valeur de `ratio moyen LPT` est la moyenne des ratios LPT sur les k exemples générés, sachant que le `ratio LPT` se calcule de façon similaire à `ratio LSA`, à ceci près que dans le (b), c’est le résultat l’algorithme LPT (et non de LSA) que l’on divise par B .

La valeur de `ratio moyen RMA` est la moyenne des ratios RMA sur les k exemples générés, sachant que le `ratio RMA` se calcule de façon similaire au `ratio LSA`, à ceci près que dans le (b), c’est le résultat l’algorithme RMA (et non de LSA) que l’on divise par B .

2.2 Questions

- Pour chacun des deux problèmes MIN MAKESPAN et SEMI-FAST MIN MAKESPAN, exécuter des tests sur une quarantaine d’exemples différents, en faisant varier ‘‘intelligemment’’ les paramètres d’entrée, et remplir le tableau ci-dessous.²
- Discuter et expliquer les résultats obtenus.

k	m	n	d_{\min}	d_{\max}	ratio moyen LSA	ratio moyen LPT	ratio moyen RMA
...

2. Si le temps d’exécution est ‘‘trop long’’ ou qu’il y a dépassement de mémoire, indiquer ‘‘PR’’ (Pas de Réponse)