M1 Informatique
X1II030

Université de Nantes
2019/2020

Distanciel
Le problème MIN MAKESPAN

Septembre 2019 – Janvier 2020

Ce "projet distanciel" se compose de deux parties : un exercice (de type TD), et un mini-projet de programmation. Ce projet est à réaliser en **binômes** (un seul monôme autorisé par groupe de TD, si celui-ci possède un nombre impair d'étudiants). Ne pas hésiter à utiliser le forum Madoc "Échanges et questions autour du projet Min Makespan" pour échanger entre vous, et aussi poser vos questions à l'enseignant.

Un compte-rendu est à rédiger, qui contiendra donc deux parties : d'une part, vos réponses aux questions de l'exercice ; d'autre part un rapport de type "rapport de projet de programmation" (ce dernier ne doit pas dépasser 10 pages). Le format de fichier attendu pour ce compte-rendu 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 l'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, jeux d'essai) et *un fichier texte contenant les instructions détaillées de compilation et d'exécution du projet*.

Cette archive est à **déposer sous Madoc**, au plus tard le **Vendredi 17 Janvier 2020 à 20h20**. Un malus sera appliqué lorsqu'une ou plusieurs des consignes données ci-dessus n'auront pas été respectées.

1 MIN MAKESPAN – Exercice

Soit le problème de minimisation suivant, appelé MIN-MAKESPAN (voir aussi Exercice 3.2 de la feuille TD3, disponible sur Madoc, et les transparents 34 à 42 disponibles dans la partie "Distanciel" de Madoc) :

> MIN-MAKESPAN
> *Instance* : n tâches de durées $d_1, d_2 \dots d_n$; m machines identiques $M_1, M_2 \dots M_m$, chaque machine ne pouvant réaliser qu'une tâche à la fois
> *Solution* : Une affectation des n tâches aux m machines
> *Mesure* : le temps de terminaison T de l'ensemble des tâches (aussi appelé *makespan*)

On propose un nouvel algorithme pour résoudre MIN MAKESPAN, que nous appellerons LPT (pour LARGEST PROCESSING TIME) :

(a) ordonner les tâches suivant l'ordre décroissant de leur durée, et les renommer $d'_1, d'_2 \dots d'_n$ de telle façon que $d'_1 \geq d'_2 \geq \dots \geq d'_n$;
(b) affecter les tâches aux machines comme dans l'algorithme LSA (voir Exercice 3.2 de la feuille TD3) en se basant sur ce nouvel ordre.

1. Indiquer ce que donne l'algorithme LPT sur l'exemple de l'Exercice 3.2 de la feuille TD3, sous la forme d'un dessin similaire à la Figure 1 de l'Exercice 3.2.
2. Quel est le ratio d'approximation obtenu par LPT sur cet exemple ?

Pour toute instance I de MIN-MAKESPAN, on note $T_{LPT}(I)$ le temps obtenu par l'algorithme LPT sur I et $T_{opt}(I)$ le temps optimal recherché pour I.

3. Montrer que pour toute instance I, $T_{LPT}(I) \leq \frac{\sum_{k \neq j} d'_k}{m} + d'_j$, où j est le numéro de la tâche qui se termine en dernier.
4. Supposons pour commencer que $n \leq m$. En déduire dans ce cas que LPT est toujours optimal.

Supposons maintenant que $n \geq m + 1$.

5. Montrer que $T_{opt}(I) \geq 2d'_{m+1}$.
6. Appelons j le numéro de la tâche qui se termine en dernier quand LPT est appliqué. Montrer que l'on est nécessairement dans un des deux cas suivants : (a) $T_{LPT}(I) = T_{opt}(I)$ ou (b) $j \geq m + 1$.
7. En vous appuyant sur les questions précédentes, montrer que pour toute instance I, $T_{LPT}(I) \leq r \cdot T_{opt}(I)$, où r est une constante dont vous donnerez la valeur.
8. Conclure quant au ratio d'approximation de l'algorithme LPT.

On s'intéresse à la classe d'instances suivante, que l'on appellera I_m : m est le nombre de machines, et le nombre de tâches à accomplir est égal à $2m + 1$. Les durées des tâches sont, dans cet ordre, les suivantes :

- 3 tâches de durée m
- 2 tâches de durée $m + 1$
- 2 tâches de durée $m + 2$
- 2 tâches de durée $m + 3$

  $\vdots$
- 2 tâches de durée $m + m - 2$
- 2 tâches de durée $m + m - 1$

9. Donner, en la justifiant, la valeur de $T_{opt}(I_m)$. Illustrer votre réponse sur l'instance I_5 (donc $m = 5$).
10. Donner, en la justifiant, la valeur de $T_{LPT}(I_m)$. Illustrer votre réponse sur l'instance I_5.
11. Quelle est la valeur du ratio $\frac{T_{LPT}(I_m)}{T_{opt}(I_m)}$ lorsque m devient grand ? En déduire une borne inférieure sur le ratio d'approximation de l'algorithme LPT.
12. Donner, en la justifiant, la valeur de $T_{LSA}(I_m)$. Illustrer votre réponse sur l'instance I_5.
13. Quelle est la valeur du ratio $\frac{T_{LSA}(I_m)}{T_{opt}(I_m)}$ lorsque m devient grand ?

2 MIN MAKESPAN – Projet de programmation

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

2.1 Présentation Générale

Le but de ce projet est d'étudier le problème MIN MAKESPAN (voir feuille de TD3, et voir aussi les transparents 34 à 42 disponibles dans la partie "Distanciel" de Madoc). Plus précisément, on vous demande d'implémenter plusieurs algorithmes répondant au problème, de les tester et de fournir les résultats de ces tests à l'utlisateur. Les trois algorithmes à implémenter sont :

1. l'algorithme List Scheduling Algorithm (LSA) ;
2. l'algorithme Largest Processing Time (LPT) ;
3. un algorithme (polynomial) de votre choix (MYALGO), que vous devez donc inventer, et que vous expliquerez en détail (en argumentant vos choix) dans le rapport.

Remarque : pour ce projet, on ne s'intéresse qu'au *temps total de réalisation* (c'est-à-dire le *makespan*) 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 2 doit exécuter une tâche de longueur 7, alors on l'écrira simplement de la manière suivante : `M[2] <-- M[2]+7`.

2.2 Travail demandé

Proposer un programme convivial (=*user friendly* en anglais) qui, par l'intermédiaire d'un menu, permet à l'utilisateur (1) de rentrer la/les instance/s de son choix et (2) de voir les résultats obtenus par les trois algorithmes évoqués ci-dessus sur cette/ces instance/s.

Instances d'entrée. L'utilisateur doit pouvoir choisir entre quatre modes de saisie des instances d'entrée :

1. (**Depuis un fichier**) L'instance est chargée à partir d'un fichier, présent dans le répertoire courant, et dont le nom est donné par l'utilisateur. Ce fichier ne contient qu'une ligne, qui est une chaîne (sans espaces) comme ci-dessous :

   $$m : n : d_1 : d_2 : d_3 \dots d_n$$

   Cela signifie que l'instance de MIN MAKESPAN est une instance à m machines et n tâches, dont les durées sont $d_1, d_2, d_3 \dots d_n$.
2. (**Au clavier**) L'instance est fournie au clavier par l'utilisateur, sous la forme d'une chaîne (sans espaces) comme ci-dessous :

   $$m : n : d_1 : d_2 : d_3 \dots d_n$$

   et dont la signification est la même que dans le cas précédent.
3. (**Génération d'une instance de type** I_m) L'utilisateur entre un entier m au clavier, qui correspond au nombre de machines, et l'instance I_m (décrite dans la Partie 1, juste après la Question 8) est générée.
4. (**Génération aléatoire de plusieurs instances**) L'utilisateur fournit 5 entiers $m, n, k, \min$ et $\max$. Il faut alors générer k instances distinctes : pour chacune de ces k instances, on a m machines, n tâches, et les durées des tâches sont générées de manière *aléatoire*, à la condition que chaque durée d_i, $1 \leq i \leq n$, vérifie $\min \leq d_i \leq \max$.

On appellera ces quatre modes de création d'instance I_f (fichier), I_c (clavier), I_m et I_R (Random).

Production des résultats. Pour chaque mode de création d'instances (I_f, I_c, I_m et I_R), les trois algorithmes LSA, LPT et MYALGO seront exécutés. Selon le mode choisi, les résultats seront fournis de façons différentes.

1. (**Modes I_f, I_c et I_m**) Les résultats seront affichés à l'écran de la façon suivante.

```
Borne inférieure ''maximum'' =
Borne inférieure ''moyenne'' =
Résultat LSA =
ratio LSA =
Résultat LPT =
ratio LPT =
Résultat MyAlgo =
ratio MyAlgo =
```

   Pour davantage d'informations sur les notions de `Borne inférieure ''maximum''` et de `Borne inférieure ''moyenne''`, voir le transparent 40 du fichier explicatif qui accompagne la partie "Distanciel" de Madoc.

   La valeur `Résultat LSA` est le temps T_{LSA} calculé sur l'instance considérée.

   La valeur `ratio LSA` se calcule de la façon suivante :

   (a) on prend la valeur la plus grande entre `Borne inférieure ''maximum''` et `Borne inférieure ''moyenne''` : appelons cette valeur $\max_b$;

   (b) `ratio LSA` est le résultat de la division de `Résultat LSA` par $\max_b$ (donc ce n'est pas forcément un entier).

   Pour les résultats concernant LPT (respectivement MYALGO), il suffit d'adapter les deux calculs ci-dessus en remplaçant LSA par LPT (respectivement MYALGO).

2. (**Mode I_R**) Dans ce mode, on n'affichera à l'écran que les valeurs moyennes des ratios. Plus précisément, seules les trois valeurs suivantes seront affichées :

```
ratio moyen LSA =
ratio moyen LPT =
ratio moyen MyAlgo =
```

La valeur de `ratio moyen LSA` (respectivement de `ratio moyen LPT`, de `ratio moyen MyAlgo`) est la moyenne des ratios LSA (respectivement des ratios LPT, des ratios MyAlgo), tels que définis ci-dessus, sur les k exemples générés.

Complexité en temps et Discussion des résultats. Dans le compte-rendu, en 2 pages maximum, discuter des complexités en temps (et au pire) de :

- la génération des instances (mode I_m et I_R) ;
- la copie en mémoire des instances (modes I_f et I_c) ;
- chacun des trois algorithmes implémentés.

Réaliser votre propre campagne de tests sur les 3 algorithmes et les 4 types d'instance considérés, et discuter les résultats obtenus dans le compte-rendu (2 pages maximum). On attend une description brève (mais précise) de la campagne de tests effectués, ainsi que des éléments de discussion/réflexion sur les points suivants (liste non exhaustive) :

- comparaison entre le ratio LSA pour I_m (quand m est grand) et le ratio trouvé à la Question 13 de la Partie 1 ;
- comparaison entre le ratio LPT pour I_m (quand m est grand) et le ratio trouvé à la Question 11 de la Partie 1 ;
- comparaison des performances des trois algorithmes, notamment dans le mode I_R ;
- Classes d'instance pour lesquelles LSA est meilleur que les deux autres algorithmes ;
- classes d'instance pour lesquelles LPT est meilleur que les deux autres algorithmes ;
- classes d'instance pour lesquelles MYALGO est meilleur que les deux autres algorithmes ;
- etc.