

Distanciel de Complexité des algorithmes

Le problème Min Makespan

Année 2017/2018

Carol COUILLEROT - Pierre CAILLAUD

Exercice: Algorithme LPT

1

M1	d2	d8	d5	d3
M2	d6	d11	d7	
M3	d9	d4	d1	d10

2

Sur cet exemple, l'application de LPT donne le résultat optimal : $T_{LPT}(I) = 1 * T_{opt}(I)$.
donc le ratio d'approximation est de **1**.

3

Soit $t_{deb}(j)$ le temps auquel débute la tâche j.

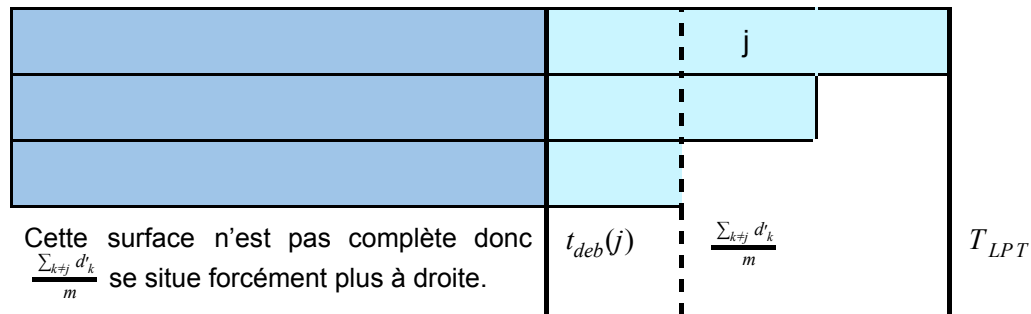
Montrons d'abord que $t_{deb}(j) \leq \frac{\sum_{k \neq j} d'_k}{m}$:

Comme on affecte toujours à la première machine libre, on a forcément une surface pleine avant $t_{deb}(j)$.

Dans le cas où $t_{deb}(j) = \frac{\sum_{k \neq j} d'_k}{m}$, on a :

M1		j
M2		
M3		
	$\frac{\sum_{k \neq j} d'_k}{m}$	$t_{deb}(j)$
		T_{LPT}

Dans tous les autres cas, on a par exemple:



On a montré que $t_{deb}(j) \leq \frac{\sum_{k \neq j} d'_k}{m}$

or, $\forall I, T_{LPT}(I) = t_{deb}(j) + d'_j$

donc $T_{LPT}(I) \leq \frac{\sum_{k \neq j} d'_k}{m} + d'_j$

4

Dans le cas où $n \leq m$, chaque machine se voit affecté au plus une tâche.

Dans ce cas $T_{LPT} = d'_{max}$ où d'_{max} est le temps de la tâche la plus longue.

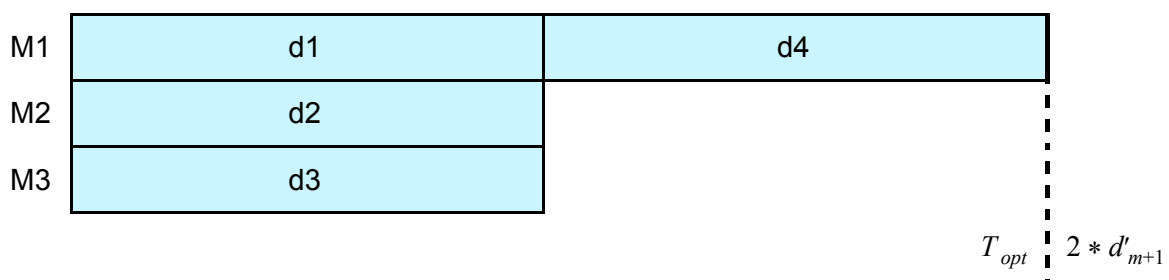
Comme les tâches sont indivisibles, T_{LPT} est forcément optimal.

5

Dans le cas où $n \geq m + 1$, chaque machine se voit affecté au moins une tâche et M1 au moins deux tâches.

Illustrons le cas où la valeur de $2 * d'_{m+1}$ est la plus grande possible. Pour obtenir une valeur de $2 * d'_{m+1}$ maximale, il nous faut une valeur de d'_{m+1} qui soit maximale. Les tâches étant réparties par ordre décroissant de leur durée, cela implique que $d'_{max} = d'_1$. Il nous faut donc illustrer le cas où $d'_1 = d'_{m+1}$.

Supposons donc le cas où $m=3$ et $n = 4$ avec $d_1 = d_2 = d_3 = d_4 = 3$



Dans ce cas, on a $T_{opt} = 2 * d'_{m+1}$

Pour que $2 * d'_{m+1}$ soit supérieur à T_{opt} , il faudrait que d'_{m+1} soit supérieur à d'_1 ce qui est impossible puisque les tâches sont affectées par ordre décroissant.

donc $2 * d'_{m+1} \leq T_{opt}(I)$

6

Dans le cas où $n \leq m$, d'après la réponse **4** on est dans le cas (a)

Sinon, dans le cas où $n \geq m + 1$, on a :

Soit:

$T_{opt} = 2 * d'_{m+1}$ et dans ce cas $j = m + 1$ comme illustré plus haut.

Soit:

$T_{opt}(I) > 2 * d'_{m+1}$ et dans ce cas on a $j > m + 1$

7

Le cas (a) étant nécessairement optimal, on ne se place que dans le cadre du cas (b).

On a donc d'une part :

$j > m + 1$ et les tâches sont affectées dans l'ordre décroissant de leur durée,

donc $d'_j \leq d'_{m+1}$.

Or, d'après la réponse **5**, $2 * d'_{m+1} \leq T_{opt}(I) \Leftrightarrow d'_{m+1} \leq \frac{1}{2} T_{opt}(I)$

On a donc

$d'_j \leq d'_{m+1} \leq \frac{1}{2} T_{opt}(I)$

A

D'autre part,

On sait que $T_{LPT}(I) = t_{deb}(j) + d'_j \leq \frac{\sum_{k \neq j} d'_k}{m} + d'_j \leq T_{opt}(I) + d'_j$

donc

$T_{LPT}(I) \leq T_{opt}(I) + d'_j$

B

On déduit donc de **A** et **B** que :

$$T_{LPT}(I) \leq T_{opt}(I) + \frac{1}{2} T_{opt}(I)$$

$$\Leftrightarrow T_{LPT}(I) \leq \frac{3}{2} T_{opt}(I)$$

8

On peut en conclure que, quelle que soit l'instance du problème Min Makespan, la complexité en temps pour le résoudre avec LPT est au pire 1,5 fois supérieure au temps optimal.

Rapport de projet de programmation

Structures choisies et implémentation des algos LSA et LPT

Pour représenter une instance du problème nous avons choisis de créer une classe Instance. Cette classe contient l'ensemble des propriétés d'une instance ainsi que l'ensemble des méthodes de résolution (LSA, LPT et MyAlgo). L'interface utilisateur a été implémentée directement dans la fonction main.

Une instance est donc composée des attributs suivant:

- nbM : Entier représentant le nombre de machines
- n : Entier représentant le nombre de tâches
- d : Tableau de n entiers contenant les durées de toutes les tâches.
- m : Tableau de nbM entiers contenant le temps de fonctionnement de chaque machine

ainsi que des variables borne_inf_max, borne_inf_moy, res_LSA, res_LPT, res_Perso, dont les noms sont suffisamment explicites pour ne pas les décrire.

Une fois l'instance créée à partir des fonctions createFromString(), createFromFile() ou createRandom(), on détermine ses bornes avec la fonction calculBornes().

L'exécution de l'algorithme LSA sur notre instance est très simple, on parcourt l'ensemble de nos tâches et, pour chacune d'elle on affecte sa durée à m[premiereMachineDispo()]. La fonction premiereMachineDispo() renvoyant l'indice de la machine dont le temps de fonctionnement est minimal. Enfin, on récupère le résultat de LSA qui correspond à la durée de fonctionnement maximum de nos machines.

L'exécution de LPT est similaire, on veille simplement à trier les tâches d par ordre décroissant avant de les affecter aux machines.

Enfin, la fonction result() nous permet de récupérer l'ensemble de ces valeurs sous la forme d'une chaîne de caractères.

En ce qui concerne les résultats de types I_g , ils sont calculés dans la fonction main sur un ensemble d'instance aléatoires. Les résultats sont ensuite écrits directement dans un fichier dont le nom aura été au préalable spécifié par l'utilisateur.

Algorithme MyAlgo

Notre algorithme, comme l'algorithme LPT, commence par trier les tâches par ordre décroissant de leurs durées.

La partie entière de la moyenne (somme des tâches)/(nombre de machines) est ensuite calculée.

A partir de là une machine est sélectionnée, la liste triée des tâches est parcourue.

Si la durée de la tâche courante plus les durées des tâches déjà ajoutées à cette machine est inférieur ou égale à la moyenne, alors elle est ajoutée.

Une autre machine est ensuite sélectionnée et ainsi de suite.

Si à la fin il reste des tâches non affectées elles sont ajoutées aux machines dont la somme des durées des tâches affectées est minimale.

La complexité est $O(n*m)$ avec n le nombre de tâches et m le nombre de machines, puisqu'en effet pour chaque machine on parcourt au pire l'ensemble des tâches.

Les résultats sont globalement proches de l'algorithme LPT et parfois meilleurs, notamment sur des instances avec des durées sur un intervalle dense (peu de valeurs différentes en comparaison du nombre de tâches).

Prenons l'instance suivante :

3 machines, 10 tâches : 10, 10, 10, 9, 8, 8, 7, 6, 5, 5

Algorithme LPT

Machines	Tâches				Somme
M1	10	9	5	5	29
M2	10	8	7		25
M3	10	8	6		24

MyAlgo

calcul de la moyenne $(10+10+10+9+8+8+7+6+5+5)/3 = 26$

Machines	Tâches				Somme
M1	10	10	6		26
M2	10	9	7		26
M3	8	8	5	5	26