

El Farissi Belal  
Bodo Maxence  
588R



# Projet Distanciel Optimisation pour la gestion

Le problème Min Makespan

# Min Makespan - Exercice

On se donne l'instance I de Min-Makespan suivante :

- $m=5$  machines
- $n= 14$  tâches
- Durées des tâches :

Tâche	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Durée	4	5	3	6	7	6	4	5	7	8	3	7	8	1

## Exercice 1

Temps	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
M1	1				7				11			14	/					
M2	2					8					13							
M3	3			6					12							/		
M4	4					9							/					
M5	5							10							/			

Le temps  $T_{LSA}(I)$  est égale à 18

## Exercice 2

Temps	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
M1	10								8					11		
M2	13								1				7			
M3	9							4						14	/	
M4	5							6						/		
M5	12							2					3			

Le temps  $T_{LPT}(I)$  est égale à 16

### Exercice 3

Temps	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	10								9						
M2	13								5						
M3	12							3			11			14	/
M4	4						2				1				
M5	6						8				7				

Preuve que le  $T_{OPT}(I) \leq 15$ , il ne sera pas possible de descendre à un temps inférieur à 15 avec seulement 5 machines.

### Exercice 4

Comme le montre la solution trouvée à l'exercice 3,  $T_{OPT} \geq 15$  on peut donc en déduire que  $T_{OPT} = 15$

### Exercice 5

Tâche	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Durée	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	6

Durée	1	2	3	4	5	6	7	8	9	10	11	12
M1	1	7	13	19	25							
M3	2	8	14	20	/							
M3	3	9	15	21	/							
M4	4	10	16	22	/							
M5	5	11	17	23	/							
M6	6	12	18	24	/							

$T_{LSA}(I_6) = 12$

## Exercice 6

$$T_{LSA}(l_{2p}) = 4p$$

En effet, on a 3 tâches distinctes,

La première de longueur 1 est égale à  $4p$  tâches répartie en  $2p$  machine

La deuxième de longueur 2 est égale à  $2p(p-1)$  tâches réparties encore une fois en  $2p$  machines

Enfin la longueur de la dernière tâches est toujours égale à  $2p$

Donc :

$$= [4p \cdot 1 + (2p(p-1)) \cdot 2] / 2p + 1 \cdot 2p$$

$$= [4p + (2p^2 - 2p) \cdot 2] / 2p + 2p$$

$$= [4p + 4p^2 - 4p] / 2p + 2p$$

$$= 4p^2 / 2p + 2p$$

$$= 4p$$

## Exercice 7

Durée	1	2	3	4	5	6	7
M1	25	25	25	25	25	25	7
M3	13	13	18	18	23	23	8
M3	14	14	19	19	24	24	9
M4	15	15	20	20	1	4	10
M5	16	16	21	21	2	5	11
M6	17	17	22	22	3	6	12

$$T_{LPT}(l_6) = 7$$

## Exercice 8

$$T_{LPT}(l_{LPT}) = 2p+1$$

La tâche de longueur  $2p$  utilise toujours 1 machine sur  $2p$  temps. (en rouge)

Les  $2p(p-1)$  tâches de longueur 2 utilisent tout le temps  $2p-1$  machines puis  $p-1$  machines et s'étalent sur  $2p$  temps. (en jaune)

Les  $4p$  tâches de longueur 1 utilisent  $p$  machines puis  $2p$  machines sur 3 temps. (en bleu)

Du coups on en déduit que la durée est toujours à  $2p+1$

## Exercice 9

$$\frac{(4p1)+(2p(p-1)*2)+(1*2p)}{2p} = 2p+1$$

La valeur de  $T_{LPT}(I_{2p})$  est égale à  $T_{LPT}$  comme on peut le voir cela correspond toujours à la valeur maximale car le tableau est toujours entièrement rempli.

## Exercice 10

$$\frac{T_{LSA}(I_{2p})}{T_{OPT}(I_{2P})} = \frac{4p}{2p+1} = \frac{4}{2+\frac{1}{p}}$$

## Exercice 11

$\frac{4}{2+\frac{1}{p}}$  tends vers 2 quand p tends vers  $+\infty$

## Exercice 12

L'instance choisie est composée de 9 tâches et 5 machines

Tâche	1	2	3	4	5	6	7	8	9
Durée	7	7	6	6	5	5	4	4	4

	Temps	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LPT	m1	1	1	1	1	1	1	1	7	7	7	7	9	9	9	9
	m2	2	2	2	2	2	2	2	8	8	8	8				
	m3	3	3	3	3	3	3	5	5	5	5	5				
	m4	4	4	4	4	4	4	6	6	6	6	6				
OPT	Temps	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	m1	1	1	1	1	1	1	1	6	6	6	6	6			
	m2	2	2	2	2	2	2	2	5	5	5	5	5			
	m3	3	3	3	3	3	3	4	4	4	4	4	4			
	m4	7	7	7	7	8	8	8	8	9	9	9	9			

$T_{LPT} = 15$        $T_{OPT} = 12$

Donc ratio  $T_{LPT} / T_{OPT} = 15/12 = 1.25$  c'est la valeur la plus près de 1.5 que nous avons trouvées. Pour cela, nous nous sommes dit qu'il fallait des valeurs tournant autour d'une valeur choisie (ici 5) après plusieurs essais nous avons trouvé ce ratio qui est le plus près trouvé.

# Min Makespan - Projet de Programmation

## 2.1 Code

Nous avons décidé de coder ce programme en C++ en effet c'est le langage sur lequel nous sommes le plus à l'aise.

Ce code est composé de ces différentes fonctions:

- un main qui sert juste d'interface et permet à l'utilisateur de choisir entre le mode l2p et random
- Une fonction model2p qui traite le cas l2p, il suffit seulement de rentrer le paramètre p, par la suite le programme se chargera de calculer le nombre de machine et le nombre de tâche nécessaire ainsi que la longueur de ces dernières. Puis par la suite l2p calculera LSA LPT et RMA grâce aux fonctions correspondantes.
- Une fonction modeRand dans laquelle il faudra rentrer 5 arguments, le nombre de machine (m), le nombre de tâche (n), le nombre d'instance (k), la longueur minimale des tâches (dmin) et la durée maximale des tâches (dmax) l'algorithme se charge donc de créer k instance avec des longueurs de tâches différentes à chaque fois, puis calcule le ratio moyen LSA, le ratio moyen LPT et le ratio moyen RMA.
- une fonction LSA qui va se charger de ranger les tâches dans les machines selon l'algorithme LSA et va retourner la valeur de cet algorithme.
- Une fonction LPT qui va d'abord trier les tâches par tailles décroissantes puis appliquer l'algorithme LSA dessus et retourner la valeur de cet algorithme.
- Une fonction RMA qui va distribuer aléatoirement les tâches aux machines puis retourner le résultat.
- Et une fonction triDecroissant qui va tout simplement trier le tableau de tâche de manière décroissante.

## 2.2 Questions

1) Mode  $I_{2p}$

p	ratioLSA	p	ratioLSA	p	ratioLSA	p	ratioLPT	p	ratioLPT	p	ratioLPT
1	1	15	1.875	70	1.971	1	0.75	15	0.969	70	0.993
2	1.33	20	1.904	80	1.975	2	0.83	20	0.976	80	0.994
3	1.5	25	1.923	90	1.978	3	0.875	25	0.981	90	0.994
4	1.6	30	1.935	100	1.980	4	0.9	30	0.984	100	0.995
5	1.667	35	1.944	120	1.983	5	0.917	35	0.986	120	0.996
6	1.714	40	1.951	140	1.986	6	0.929	40	0.988	140	0.996
7	1.75	45	1.956	160	1.988	7	0.937	45	0.989	160	0.997
8	1.778	50	1.961	180	1.989	8	0.944	50	0.990	180	0.997
9	1.8	55	1.964	200	1.990	9	0.95	55	0.991	200	0.997
10	1.818	60	1.967	300	1.993	10	0.954	60	0.991	300	0.998

On peut voir qu'avec le paramètre  $p$  grandissant, la précision du ratio est de plus en plus précise pour LSA et LPT.

Le ratio LSA se rapproche progressivement de 2 comme prouvé plus haut.

Le ratio LPT lui se rapproche progressivement de 1, ce qui correspond bien au ratio

$T_{LPT} / T_{LSA}$

p	ratioRMA	p	ratioRMA	p	ratioRMA
1	1	15	1.625	70	1.951
2	1	20	1.904	80	2.068
3	1.375	25	1.846	90	1.826
4	1.5	30	1.887	100	1.881
5	1.917	35	2.056	120	2.004
6	1.714	40	2.171	140	1.840
7	2.312	45	2.196	160	1.953
8	1.444	50	1.990	180	1.834
9	1.8	55	1.982	200	1.836
10	1.909	60	1.803	300	2.035

Quant au ratioRMA, dû au côté aléatoire de RMA il est assez disparate mais on peut quand même voir qu'il se rapproche de 2 avec p grandissant.



2) Mode  $I_R$

n	m	k	dmin	dmax	ratio moyen LSA	ratio moyen LPT	ratio moyen RMA
22	4	5	5	14	1.090	1.034	1.559
26	5	3	7	7	1.135	1.135	1.703
8	8	3	5	10	1	1	2.352
2	4	3	10	5	1	1	1.233
26	4	6	10	11	1.065	1.060	1.442
2	4	3	5	15	1	1	1.185
14	8	2	8	8	1.067	1.067	2.133
9	9	4	1	13	1	1	1.957
2	8	5	8	11	1	1	1
24	9	5	8	8	1.091	1.091	1.963
15	6	2	3	13	1.130	1.078	2.343
15	9	5	5	8	1.241	1.138	2.514
13	4	3	10	12	1.176	1.185	1.314
28	3	6	4	12	1.048	1.027	1.425
19	5	4	9	14	1.069	1.035	1.645
20	5	3	4	12	1.111	0.989	1.404
2	6	6	7	14	1	1	1
14	9	6	10	15	1.311	1.194	2.400

30	2	3	5	10	1.006	0.991	1.099
3	8	2	6	12	1	1	1
17	8	6	3	8	1.259	1.114	2.137
7	7	3	1	15	1	1	1.662
25	8	2	1	6	1.199	1	2.192
28	5	5	2	5	1.070	1.009	1.381
12	9	5	5	9	1.429	1.227	2.413
14	5	3	5	8	1.105	1.035	1.622
11	9	5	10	10	1.538	1.535	2.461
29	10	6	3	7	1.137	1.022	2.05
24	10	5	4	13	1.242	1.076	1.997
3	2	3	1	14	1.098	1.039	1.231

Pour trouver cet exemple, il a été lancé 30 fois le programme  $lr$  sur des valeurs aléatoires. On peut voir que le ratio moyen LSA comme celui de LPT et RMA est assez disparate et compliqué à analyser.

Pour conclure ce projet, nous avons pu voir que l'algorithme le plus optimal reste LPT qui se rapproche le plus souvent de OPT, il est toujours inférieur à  $1.5OPT$  comme vue dans le cours.