



# Le problème du MIN MAKESPAN

---

**Placé Louka**

**UFR Sciences et Techniques de Nantes**

**L3 Info - MIAGE**

**2022 - 2023**

## Sommaire

### I. MIN MAKESPAN - Exercice

a) Question 1	..p2
b) Question 2	..p2
c) Question 3	..p3
d) Question 4	..p3
e) Question 5	..p3
f) Question 6	..p4
g) Question 7	..p4

### II. MIN MAKESPAN - Projet de programmation

a) Introduction	..p5
b) Explication des fonctions	..p6
c) Explication de l'algorithme principale	p10
d) Test et résultats des algorithmes	..p12

## I. MIN MAKESPAN - Exercice

Instance I de Min Makespan :

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  $TL_{SA}(I)$  obtenu.

Algorithme LSA sur l'instance I :

M5	5 (tâche E)	6 (tâche H)	5 (tâche N)	
M4	7 (tâche O)	2 (tâche J)	10 (tâche M)	
M3	4 (tâche C)	4 (tâche F)	8 (tâche L)	
M2	5 (tâche B)	3 (tâche G)	7 (tâche K)	
M1	6 (tâche A)	7 (tâche I)	9 (tâche O)	

$$M1 : 6+7+9=22 > M4 : 7+2+10=19 > M3 : 4+4+8=16 > M2 : 5+3+7=15$$

$$TL_{SA}(I) = A+I+O=6+7+9=22$$

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.

Instance I de Min Makespan rangé dans l'ordre décroissant pour l'algorithme du LPT :

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

$$M > O > L > D, I, K > A, H > B, E, N > C, F > G > J$$

Algorithme LPT sur l'instance I :

M5	7 (tâche I)	6 (tâche A)	5 (tâche N)	
M4	7 (tâche O)	7 (tâche K)	3 (tâche G)	
M3	8 (tâche L)	6 (tâche H)	4 (tâche F)	
M2	9 (tâche O)	5 (tâche O)	4 (tâche C)	
M1	10 (tâche M)	5 (tâche E)	2 (tâche J)	

Les chemins les plus longs :

$$T_{LPT}(I) = \begin{cases} M2 : O+B+C = 9+5+4 = 18 \\ M3 : L+H+F = 8+6+4 = 18 \\ M5 : I+A+N = 7+6+5 = 18 \end{cases}$$

Le temps optimal pour l'instance I est 18,  $T_{opt}(I)$  est égale à 18.

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

L'algorithme LPT sur l'instance I de la réponse à la question 2. répond à celle-ci.

On a bien  $T_{opt}(I) = 18$  donc  $T_{opt}(I) \leq 18$ .

4. Démontrer que  $T_{opt}(I) \geq 18$ .

On sait que  $T_{opt} \geq$  la plus longue tâche, dite borne inférieur maximum et

$T_{opt} \geq$  la moyenne des longueurs des tâches, dite borne inférieur moyenne

$T_{opt} \geq 10 \rightarrow$  borne inférieur maximum

$A+B+C+D+E+F+G+H+I+J+K+L+M+N+O = 6+5+4+7+5+4+3+6+7+2+7+8+10+5+9 = 88$

$\frac{88}{5} = 17.6 \rightarrow$  borne inférieur moyenne. Les tâches sont des entiers donc 18.

On choisit la borne la plus contraignante donc 18. Donc on obtient bien  $T_{opt}(I) \geq 18$ .

5. Fournir une instance I de MIN MAKESPAN avec  $m \geq 4$  machines et  $n \geq 10$  tâches pour laquelle on a  $T_{LPT}(I) > T_{LSA}(I)$ . Justifier.

Instance I de MIM MAKESPAN à  $m \geq 4$  machines et  $n \geq 10$  tâches pour laquelle on a  $T_{LPT}(I) > T_{LSA}(I)$  :

A	B	C	D	E	F	G	H	I	J
1	3	2	5	8	3	5	4	4	1

M4	5 (tâche D)				4 (tâche I)				
M3	2 (tâche C)		3 (tâche F)			4 (tâche H)			
M2	3 (tâche B)			5 (tâche G)					1 (tâche J)
M1	1 (tâche A)	8 (tâche E)							

$$T_{LSA}(I) = D + I = C + F + H = B + G + J = A + E = 5+4 = 2+3+4 = 3+5+1 = 1+8 = 9$$

Instance I de MIM MAKESPAN rangé dans l'ordre décroissant avec  $m \geq 4$  machines et  $n \geq 10$  tâches pour laquelle on a  $T_{LPT}(I) > T_{LSA}(I)$  :

E	D	G	H	I	F	B	C	A	J
8	5	5	4	4	3	3	2	1	1

M4	4 (tâche H)	4 (tâche I)	
M3	5 (tâche G)	3 (tâche B)	1 (tâche J)
M2	5 (tâche D)	3 (tâche F)	1 (tâche A)
M1	8 (tâche E)	2 (tâche C)	

$$T_{LPT}(I) = E + C = 8 + 2 = 10$$

On a donc bien  $T_{LPT}(I) > T_{LSA}(I)$ .

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.

Instance  $I'$  de Min Maskespan avec  $m \geq 4$  machines et  $n \geq 10$  tâches pour laquelle on a  $T_{opt}(I') = d_{max}$  :

A	B	C	D	E	F	G	H	I	J
9	1	2	2	1	2	2	1	2	2

M4	2 (tâche D)		1 (tâche H)	2 (tâche I)			
M3	2 (tâche C)		2 (tâche G)				
M2	1 (tâche B)	1 (tâche E)	2 (tâche F)		2 (tâche J)		
M1	9 (tâche A)						

Borne inférieur maximum = 9 =  $d_{max}$

$$\text{Borne inférieur moyenne} = \frac{9+1+2+2+1+2+2+1+2+2}{4} = \frac{24}{4} = 6$$

On a la Borne inférieur maximum > Borne inférieur moyenne, donc on prend la plus contraignante.

On a donc bien  $T_{opt}(I') = d_{max}$ .

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.

Instance  $I''$  de Min Maskespan avec  $m \geq 4$  machines et  $n \geq 10$  tâches pour laquelle on a  $T_{opt}(I'') = d_{max}$  et  $T_{LPT}(I'') > T_{opt}(I'')$  :

A	B	C	D	E	F	G	H	I	J
20	16	14	14	5	1	1	1	1	1

M4	14 (tâche d)	1 (tâche G)	1 (tâche I)						
M3	14 (tâche c)	1 (tâche F)	1 (tâche H)	1 (tâche J)					
M2	16 (tâche B)	5 (tâche E)							
M1	20 (tâche A)								

On a donc  $T_{LPT}(I'') = 20$

La borne inférieur maximum = 20 =  $d_{max}$

La borne inférieur moyenne =  $\frac{20+16+14+14+5+1+1+1+1+1}{4} = \frac{74}{4} = 18.25$  (  $\rightarrow$  19 car les tâches ne peuvent être que des entiers)

Borne inférieur maximum > Borne inférieur moyenne, donc  $T_{opt}(I'') = 20$

On a donc bien  $T_{opt}(I'') = d_{max} = 20$  et  $T_{LPT}(I'') > T_{opt}(I'')$ .

## II. MIN MAKESPAN - Projet de programmation

### a) Introduction

### Contexte

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  $m/2$  réalisent la tâche deux fois plus vite que les machines numérotées de  $m/2 + 1$  à  $m$ . Par exemple, sur une machine de numéro  $1 \leq i \leq m/2$ , une tâche de durée 10 sera réalisée en un temps de  $10/2 = 5$ , alors qu'elle prendra un temps de 10 sur une machine de numéro  $m/2 + 1 \leq j \leq m$ .

Pour ces deux problèmes, on va implémenter trois algorithmes, les tester puis fournir les résultats à l'utilisateur :

- 1/ l'algorithme List Scheduling Algorithm (LSA);
- 2/ l'algorithme Largest Processing Time (LPT);
- 3/ l'algorithme 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 effectuer 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  $i$ .

## 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.

### b) Explication des fonctions

La fonction *saisir\_entre*(*D*,*dmin*,*dmax*) prend en paramètre deux entiers, un minimum et un maximum appelé *dmin* et *dmax* et la fonction génère aléatoirement pour chaque *n* nombre de tâche un entier compris entre le *dmin* et le *dmax* qui va correspondre à la durée de la tâche. Cette fonction est utile dans le cas du Min Makespan.

```
def saisir_entre(dmin,dmax):
```

La fonction *saisir\_entre\_semifast*(*dmin*,*dmax*) prend en paramètre deux entiers, un minimum et un maximum appelé *dmin* et *dmax* et la fonction génère aléatoirement pour chaque *n* nombre de tâche un **entier pair** compris entre le *dmin* et le *dmax* qui va correspondre à la durée de la tâche. Cette fonction est utile dans le cas du Semi-Fast Min Makespan.

```
def saisir_entre_semifast(dmin,dmax):
```

- La fonction  $LSA(D, M, m)$  où  $D$  est un tableau (des tâches),  $M$  est un tableau (des machines) et  $m$  est le nombre de machines. Cette fonction sert à attribuer les tâches de la même manière que l'algorithme du LSA. C'est-à-dire qu'elle va remplir le tableau  $M$  dans l'ordre croissant des numéros de machines en leur attribuant des tâches, puis il prendra la machine qui aura fini d'effectuer sa tâche le plus tôt pour lui attribuer une tâche toujours dans l'ordre croissant des tâches.

Les calculs de la borne inférieure maximum et moyenne sont également effectués par cette fonction, le choix de la borne pour le calcul du ratio LSA se fait en fonction de la borne la plus contraignante. La borne inférieure maximum est égale à la tâche la plus longue et la borne inférieure moyenne est la somme des tâches divisé par le nombre de machine  $m$  puis arrondi à l'entier supérieur (car une tâche ne peut être qu'un entier). Le ratio LSA noté  $Tlsa$  est récupéré dans une variable global qui va permettre le calcul du ratio moyen LSA en fonction du nombre d'instances.

```
#=====
def LSA(D,M,m):
```

- La fonction  $LPT(D, M, m)$  où  $D$  est un tableau trié dans l'ordre décroissant (des tâches),  $M$  est un tableau (des machines) et  $m$  est le nombre de machines. Cette fonction sert à attribuer les tâches de la même manière que l'algorithme du LPT. C'est-à-dire qu'elle va remplir le tableau  $M$  dans l'ordre croissant des numéros de machines en leur attribuant des tâches, puis il prendra la machine qui aura fini d'effectuer sa tâche le plus tôt pour lui attribuer une tâche toujours dans l'ordre croissant des tâches.

Les calculs de la borne inférieure maximum et moyenne sont également effectués par cette fonction, le choix de la borne pour le calcul du ratio LPT se fait en fonction de la borne la plus contraignante. La borne inférieure maximum est égale à la tâche la plus longue et la borne inférieure moyenne est la somme des tâches divisé par le nombre de machine  $m$  puis arrondi à l'entier supérieur (car une tâche ne peut être qu'un entier). Le ratio LPT noté  $Tlpt$  est récupéré dans une variable global qui va permettre le calcul du ratio moyen LPT en fonction du nombre d'instances.

```
def LPT(D,M,m):
```



- La fonction  $RMA(D, M, m)$  où  $D$  est un tableau (des tâches),  $M$  est un tableau (des machines) et  $m$  est le nombre de machines. Cette fonction sert à attribuer les tâches de la même manière que l'algorithme du RMA. C'est-à-dire qu'elle va remplir le tableau  $M$  de machines aléatoirement en leur attribuant des tâches.

Les calculs de la borne inférieure maximum et moyenne sont également effectués par cette fonction, le choix de la borne pour le calcul du ratio RMA se fait en fonction de la borne la plus contraignante. La borne inférieure maximum est égale à la tâche la plus longue et la borne inférieure moyenne est la somme des tâches divisé par le nombre de machine  $m$  puis arrondi à l'entier supérieur (car une tâche ne peut être qu'un entier). Le ratio RMA noté  $Trma$  est récupéré dans une variable global qui va permettre le calcul du ratio moyen RMA en fonction du nombre d'instances.

```
def RMA(D, M, m):
```

Il existe les mêmes fonctions  $LSA$ ,  $LPT$  et  $RMA$  mais ajustées afin de répondre au problème du Semi-Fast Min Makespan, elles se nomment respectivement  $LSAsemi$ ,  $LPTsemi$  et  $RMAsemi$ . Ces fonctions ne diffèrent pas grandement des fonctions pour le Min Makespan, en effet il suffit d'y rajouter quelques conditions. Au niveau des fonctions qui ne gèrent que l'affectation des tâches aux machines, il faut prendre en compte un paramètre très important, les machines numérotées de 1 à  $m/2$  réalisent la tâche deux fois plus vite que les machines numérotées de  $m/2+1$  à  $m$ . Il y a d'autres conditions telles que le nombre de machines doit être pair ainsi que les durées des tâches sont des entiers pairs mais cela est géré quand l'utilisateur rentre ses valeurs de son choix.

De plus, le calcul des bornes inférieure minimum et moyenne se voit modifier. En effet le calcul de la borne inférieure maximum est égal à la durée de la tâche la plus longue divisé par 2 et la borne inférieure moyenne est égal à la somme des durées des tâches divisé par le nombre de machines multiplié par  $\frac{2}{3}$ .

- La fonction  $LSAsemi(D, M, m)$  où  $D$  est un tableau (des tâches),  $M$  est un tableau (des machines) et  $m$  est le nombre de machines. Cette fonction sert à attribuer les tâches de la même manière que l'algorithme du LSA. Seulement, elle répond aux conditions du Semi-Fast Makespan évoquées ci-dessus.

```
def LSAsemi(D, M, m):
```

- La fonction  $LPTsemi(D, M, m)$  où  $D$  est un tableau (des tâches),  $M$  est un tableau (des machines) et  $m$  est le nombre de machines. Cette fonction sert à attribuer les tâches de la même manière que l'algorithme du LPT. Seulement, elle répond aux conditions du Semi-Fast Makespan évoquées ci-dessus.

```
def LPTsemi(D, M, m):
```

- La fonction  $RMAsemi(D, M, m)$  où  $D$  est un tableau (des tâches),  $M$  est un tableau (des machines) et  $m$  est le nombre de machines. Cette fonction sert à attribuer les tâches de la même manière que l'algorithme du RMA. Seulement, elle répond aux conditions du Semi-Fast Makespan évoquées ci-dessus.

```
def RMAsemi(D, M, m):
```

### c) Explication du programme principal

- On initialise les valeurs des moyennes des ratios à 0 pour tous les algorithmes. De même pour la variable *choix* qu'on met à 0.  
La variable *choix* servira plus tard dans le choix du Min Makespan ou Semi-Fast Makespan par l'utilisateur.

```
#=====
#=====Programme Principal
#=====

#Initialise les variables
MoyenneRatioLSA = 0
MoyenneRatioLPT = 0
MoyenneRatioRMA = 0

MoyenneRatioLSAsemi = 0
MoyenneRatioLPTsemi = 0
MoyenneRatioRMAsemi = 0

choix = 0
```

- Tout d'abord l'utilisateur sera invité à indiquer le choix de son problème :
  - 1 pour le problème du Min Makespan
  - 2 Pour le le problème du Semi-Fast Min Makespan

Cette opération va se boucler tant que l'utilisateur ne rentre pas soit 1 ou 2.

```
# Condition vérifié si l'user tape 1 ou 2
while choix != 1 and choix !=2 :
    choix = int(input('Tapez 1 pour le problème du Min-Makespan et 2 pour le Semi-Fast'))
```

- En fonction du choix de l'utilisateur, le programme du Min-Makespan (1) ou celui du Semi-Fast Min Makespan (2) va s'exécuter. Ici c'est le cas du Min Makespan.  
Tout d'abord les variables *dmin* et *dmax* sont initialisées à 0 et 1. Puis l'utilisateur doit entrer son nombre *m* de machine(s), *n* de tâche(s), *k* d'instance(s) tout en ayant comme conditions que cela soit strictement supérieur à 0, sinon le programme re-demande à l'utilisateur d'entrer une valeur tant qu'elle correspond à la condition.  
Les tableaux D des tâches et M des machines sont déclarés en fonction du nombre de tâches et de machines et initialisé à 0. Des copies M1 et M2 sont faites du tableau M pour en avoir un par algorithme.

Les valeurs  $dmin$  et  $dmax$  sont à renseigner avec la condition que  $dmin$  soit supérieur ou égale 1 et  $dmax$  soit strictement supérieur à  $dmin$ .

```
if choix ==1 : #Si tape 1 (alors min-makespan)
```

- Pour chaque instance demandée par l'utilisateur, on va d'abord faire appel à la fonction *saisir\_entre*( $dmin, dmax$ ) qui va remplir notre tableau de tâches  $D$ . On va copier ce tableau  $D$  en  $D1$  et  $D2$ . Ensuite, on fait appel à la fonction *LSA*( $D, M, m$ ) pour calculer le ratio LSA nommé *Tlsa* qu'on récupère pour en faire une somme dans *MoyenneRatioLSA* afin d'avoir plus tard la moyenne de ce ratio LSA sur  $k$  instance.

De même pour calculer le ratio LPT, seulement le tableau  $D1$  utilisé dans la fonction *LPT*( $D1, M1, m$ ) possède les mêmes valeurs des tâches que  $D$  seulement trié dans l'ordre décroissant (***D1.sort(reverse = True)***).

Puis également pour calculer le ratio RMA, on fait appel à la fonction *RMA*( $D2, M2, m$ ) ici  $D2$  est le même tableau que  $D$ .

```
#boucle pour chaque k instance
for i in range(k):
```

```
    LSA(D,M,m) #LSA
    MoyenneRatioLSA = MoyenneRatioLSA + Tlsa
```

```
    LPT(D1,M1,m) #LPT
    MoyenneRatioLPT = MoyenneRatioLPT + Tlpt
```

```
    RMA(D2,M2,m) #RMA
    MoyenneRatioRMA = MoyenneRatioRMA + Trma
```

- À Chaque fin de boucle  $k$ , les tableaux  $M, M1, M2$  sont remis à 0 afin d'être opérationnel pour le prochain tour de boucle

Enfin, le calcul de la moyenne des ratios du Min Makespan se fait par la somme calculée par tous les ratios de chaque algorithme divisé par  $k$  instance(s).

Puis on affiche ces valeurs.

```
print('====Les Moyennes des ratios =====')
```

```
MoyenneRatioLSA = MoyenneRatioLSA/k
print("La moyenne des ratios LSA :",MoyenneRatioLSA)
```

```
MoyenneRatioLPT = MoyenneRatioLPT/k
print("La moyenne des ratios LPT :",MoyenneRatioLPT)
```

```
MoyenneRatioRMA = MoyenneRatioRMA/k
print("La moyenne des ratios RMA :",MoyenneRatioRMA)
```

- Dans le cas où l'utilisateur choisit le choix du Semi-Fast Min Makespan (2), les algorithmes LSA, LPT et RMA s'effectueront avec les fonctions *LSAsemi*, *LPTsemi* et

*RMAsemi.* À la différence du Min Maskespan, ici on s'assurera que l'utilisateur entre un nombre de machine qui soit pair et que les durées des tâches sont toutes paires.

Enfin, le calcul de la moyenne des ratios du Semi-Fast Min Makespan se fait par la somme calculée par tous les ratios de chaque algorithme divisé par  $k$  instance(s). Puis on affiche ces valeurs.

#### d) Test et résultats des algorithmes

##### Résultats pour Min Makespan :

	k	m	n	dm in	dma x	Ratio Moyen LSA	Ratio Moyen LPT	Ratio Moyen RMA
1	1	100	450	1	999	1.314586191129401	1.0278920896204846	2.3676268861454046
2	32	100	450	1	999	1.2628568360211399	1.0224342212658377	2.6919865737206434
3	128	100	450	1	999	1.257641310421218	1.023289678526727	2.656949065286725
4	1024	100	450	1	999	1.2574657195302283	1.0240627990729654	2.637666306748929
5	4096	100	450	1	999	1.257094885091767	1.0240663295725876	2.644476440137439
6	8192	100	450	1	999	1.256722126804552	1.023948212691004	2.645663268797885
7	8192	100	450	1000	99999	1.2553294043331602	1.0256744933298525	2.6309744630975067
8	8192	100	1000	1000	99999	1.1152041733803153	1.0037657076487418	2.033618921971691
9	8192	10	100	1000	99999	1.0826606343654888	1.0051076131106949	1.5959676006076051
10	8192	10	100	1	9	1.0675663589361215	1.0	1.5676735633720136
11	8192	10	100	1	20	1.0765298416816604	1.0013798844151756	1.5902859940755594
12	32768	100	450	1	999	1.256877809936217	1.0241637165807014	2.6424728903346772
13	2048	3000	6150	1000	9999	PR	PR	PR
14	2048	300	615	100	999	1.5619970519061588	1.078868975395764	3.9522601816133296
15	2048	2	615	100	999	1.001013620194746	1.000293485474646	1.035066883518259
16	2048	2	615	5000	10000	1.000827263132405	1.0010839394940083	1.0328811229900827
17	2048	50	615	100	999	1.0805478008918448	1.0105226099381355	1.7835505308730923
18	2048	340	615	100	999	1.6293809489870839	1.0260392742316504	4.209895176043174
19	2048	340	615	5000	10000	1.3786451220747453	1.0774495116794007	3.869114316909492
20	2048	340	615	1	10000	1.5450672665308818	1.0001507448807794	4.016393601391588

	<b>k</b>	<b>m</b>	<b>n</b>	<b>dm in</b>	<b>dma x</b>	<b>Ratio Moyen LSA</b>	<b>Ratio Moyen LPT</b>	<b>Ratio Moyen RMA</b>
21	2048	340	615	9500	10000	1.126731662564909	1.103487952699612	3.740397902255462
22	2048	5	10	1	10000	1.2894106339181421	1.0805445907050888	1.9783753676518316
23	5	5	10	1	10000	1.2871581814780289	1.0848573230640366	1.7868776291050483
24	4096	340	615	100	999	1.629794610447015	1.0262893393034846	4.18854811068124
25	4096	600	615	100	999	1.0638964122556338	1.0	3.390111043975848
26	2048	500	1000	666	6666	1.5885865439001778	1.0198850096115903	4.185405400851517
27	2048	500	1000	200	1200	1.544337487353881	1.017304129915598	4.113229410362504
28	4096	500	1000	666	6666	1.5888905061358176	1.019631699803254	4.187642379476225
29	2	500	1000	666	6666	1.5937854892260335	1.0196319763865818	3.948647128557228
30	4096	500	1000	200	1200	1.5443727662964528	1.0169967742212753	4.118471786646331
31	1024	500	450	1	999	1.0	1.0	2.982235377013853
32	4096	500	499	1	999	1.0	1.0	3.130644129977919
33	4096	500	501	1	999	1.0000110130112054	1.0	3.144833435527941
34	4096	500	499	1000	99999	1.0	1.0	3.12807219023724
35	4096	500	499	1	9	1.0	1.0	3.403533935546875
36	4096	500	750	1	999	1.3630634223996958	1.0	3.806630799092901
37	4096	500	2000	1	999	1.3144298778255417	1.0079212390030188	3.2263354675682474
38	1024	500	2000	1	999	1.3149469835783683	1.0080397686489657	3.230302598371105
39	1024	500	2000	1	5	1.1790482954545507	1.0	2.9560458096590825
40	1024	500	3000	1	999	1.2088187467457991	1.0042226174926763	2.7534095861713883
41	4096	3000	2000	1	999	1.0	1.0	3.1884968477361784
42	4096	2000	150	1	999	1.0	1.0	1.4966219459347812
43	4096	2000	2000	1	999	1.0	1.0	3.6183212767019763

• Dans le cas où  $m > n$ :

- Le ratio moyen de LSA et LPT sont égaux à 1, quand à ratio moyen RMA, lui varie fortement en fonction de l'écart entre le nombre  $m$  de machines et  $n$  de tâches. Plus l'écart sera important et plus le ratio moyen RMA diminue, à l'inverse il augmente si l'écart est faible.

• Dans le cas où  $n > m$ :

- Les valeurs des ratios LSA et LPT augmentent jusqu'à tendre une valeur en fonction de l'écart entre  $m$  et  $n$ . Plus l'écart est grand, plus le ratio va augmenter jusqu'à une valeur.
- Avec les mêmes valeurs de  $m$  machines et  $n$  tâches, peu importe  $k$  instance et  $d_{min}$  et  $d_{max}$ , les ratios moyens tendent vers des valeurs différentes chacune où  $\text{RatioMoyenLPT} < \text{RatioMoyenLSA} < \text{RatioMoyenRMA}$

#### Résultats pour Semi-Fast Min Makespan :

	k	m	n	$d_{min}$	$d_{max}$	Ratio Moyen LSA	Ratio Moyen LPT	Ratio Moyen RMA
1	1	100	450	2	1000	2.203665987780041	1.5587628865979382	4.97538742023701
2	32	100	450	2	1000	2.094654003322275	1.5672686739196493	4.728597061283116
3	128	100	450	2	1000	2.107092818459856	1.5661983443618448	4.950595472023631
4	1024	100	450	2	1000	2.107902290625364	1.567962370183517	4.887607118251071
5	4096	100	450	2	1000	2.1089937308963687	1.567741084109582	4.89545177534141
6	8192	100	450	2	1000	2.110917318017794	1.5680877909073274	4.884340136129177
7	8192	100	450	1000	10000	2.061746493898134	1.5664724716394562	4.7421111810169165
8	8192	100	1000	1000	10000	PR	PR	PR
9	8192	10	100	1000	10000	1.6600126792252168	1.5226940498148538	2.8450609241917566
10	8192	10	100	2	10	1.6217093859626848	1.4970542142857348	2.8015344074299584
11	8192	10	100	2	20	1.6485656100256665	1.4992816225899854	2.8361350731556003

	k	m	n	dm in	dmax	Ratio Moyen LSA	Ratio Moyen LPT	Ratio Moyen RMA
12	32768	100	450	2	1000	PR	PR	PR
13	2048	3000	6150	1000	10000	PR	PR	PR
14	2048	300	615	100	1000	2.7984439977995583	1.7844634858262427	7.276919535080816
15	2048	2	615	100	1000	1.5018698780704958	1.5000863522991892	1.9981192383362434
16	2048	2	615	5000	10000	1.5016258589314988	1.5000332258660967	1.9996099085450245
17	2048	50	615	100	1000	1.6906767446261275	1.5089769531206318	3.3559829887633303
18	2048	340	615	100	1000	2.7313396035386566	1.5493885052023861	7.706490238908066
19	2048	340	615	5000	10000	2.6218598763437932	2.096133059192826	7.157802506997202
20	2048	340	615	2	10000	2.6126824034995746	1.4631164810822805	7.405848613849278
21	2048	340	615	9500	10000	2.428045823944017	2.337578416829488	7.003625041670982
22	2048	6	10	2	10000	1.9599230325192036	1.4779813840596316	3.271576723836248
23	5	6	10	2	10000	1.7552293874545715	1.3765778502659782	2.6030125166399856
24	2048	340	615	100	1000	2.6222865750719784	1.4684128608041451	7.448862621230097
25	2048	600	615	100	1000	1.9971247930666207	1.1203246072601714	6.21737880511502
26	2048	500	1000	666	6666	2.8520444257245816	1.7609945676433294	7.802023817940969
27	2048	500	1000	200	1200	2.7958497426341737	1.823274336098071	7.687410405583331
28	2048	500	1000	666	6666	2.854237849286966	1.7605060006566053	7.83532827821469
29	2	500	1000	666	6666	2.8221235078150473	1.776914440560664	7.55547239119109
30	1024	500	500	2	1000	1.9962148902251182	1.0196244354822603	5.807098256187059
31	1024	500	450	2	1000	1.994575150117874	1.0002210677354162	5.51744586904216
32	1024	500	499	2	1000	1.9963310135652563	1.0180367819639353	5.794972367443965
33	1024	500	501	2	1000	1.9962492606535596	1.0190253362461303	5.8368993528211455
34	1024	500	499	1000	100000	1.9963939830433879	1.022507617528923	5.783706369526361
35	1024	500	499	2	10	2.0	1.200000000000023	6.533593750000007
36	1024	500	750	2	1000	2.396025644341796	1.336546773542405	7.037087083516641
37	1024	500	1000	2	1000	2.769017441962748	1.5388962804597592	8.097884282001296
38	1024	500	1000	2	6	2.00048828125	1.5	6.8735351562500036
39	1024	500	1000	2	1000	2.7652725551960726	1.5405181158537646	8.097405470879918



	<b>k</b>	<b>m</b>	<b>n</b>	<b>d<sub>m</sub> in</b>	<b>d<sub>max</sub></b>	<b>Ratio Moyen LSA</b>	<b>Ratio Moyen LPT</b>	<b>Ratio Moyen RMA</b>
40	1024	3000	1000	2	1000	1.0	1.0	4.53340721804942
41	30	400	20	2	1000	1.0	1.0	1.9873640554286027
42	30	400	200	2	1000	1.0	1.0	4.17865780469491
43	30	400	201	2	1000	1.3000495739775866	1.0	4.1690687789247605
44	30	400	199	2	1000	1.0	1.0	3.954289235564433
45	1024	2000	150	2	1000	1.0	1.0	2.644096161224311

• Dans le cas où  $m > n$ :

- Contrairement au Min Makespan, les ratios LSA et LPT ne seront pas toujours égaux à 1, quand l'écart entre  $n$  et  $m$  est supérieur à 2 fois  $n$  alors le ratio moyen LSA devient égale à 1. Il faut encore augmenter l'écart pour avoir un ratio moyen LPT qui est égale à 1 également.

• Dans le cas où  $n > m$ :

- On constate que l'écart entre  $m$  et  $n$  influe sur les ratios moyens. Plus l'écart est grand et plus les ratios moyen baissent.  
De plus, tant que  $m$  et  $n$  ne change pas, changer  $k$ ,  $d_{min}$  et  $d_{max}$  ne fera pas changer significativement les valeurs des ratios moyen.