Min Makespan

Réponses aux exercices

1. **Indiquer ce que donne l’algorithme LPT sur l’exemple de l’exercice 3.2 de la feuille de TD3, sous la forme d’un dessin similaire à la Figure 1 de l’exercice 3.2.**

Pour rappel, les valeurs de l’instance de l’exercice 3.2, une fois triés dans l’ordre décroissant, sont :

T = 7 ; 6 ; 6 ; 5 ; 3 ; 3 ; 2 ; 2 ; 2 ; 1 et N = 11 et M = 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| M1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M3 |  |  |  |  |  |  |  |  |  |  |  |  |  |

On retrouve donc TLPT = 13

1. **Quel est le ratio d’approximation obtenu par LPT sur cet exemple ?**

On sait que Topt = 13 (Topt\_borne\_max = 7 et Topt\_borne\_moyenne = 39 / 3 = 13)

Donc le ratio .

1. **Montrer que pour toute instance I, , où est le numero de la tache qui termine en dernier.**

Voyons les cas au mieux est au pire de LPT :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Au mieux | | | | | |  | Au pire | | | | | | | | | | |
| M1 |  |  |  |  |  |  | M1 |  |  |  |  |  |  |  |  | D’j |  |
| M2 |  |  |  |  |  |  | M2 |  |  |  |  |  |  |  |  |  |  |
| … |  |  | … |  |  |  | … |  |  |  | … |  |  |  |  |  |  |
| Mm |  |  |  |  |  |  | Mm |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | (1) | | | | | | | (2) | | |

On a en effet le pire scenario dans lequel toutes les taches se terminent en même temps, sauf la dernière, qui commence à la terminaison des autres.

On a alors TLPT au pire qui est égal à (1) + (2). Or et , donc .

1. **Supposons pour commencer que . En déduire dans ce cas que LPT est toujours optimal.**

Si , une seule tache au maximum n’est affectée à chaque machine. Donc (TLPT est la durée maximale d’une tache).

Or on sait que , on peut alors en conclure que et donc que TLPT est optimal pour .

1. **Montrer que .**

On sait que pour . Avec , chaque machine a au moins une tache de durée . (Donc avec la durée cumulée des taches sur une machine).

À partir de la tache m+1, au moins une machine a au moins 2 taches de durée . Le résultat de LPT ne pouvant qu’augmenter, on en déduit que .

1. **Appelons le numéro de la tache qui se termine en dernier quand LPT est appliqué. Montrer que l’on est nécessairement dans un de ces deux cas suivants : (a) ou (b) .**

En effet, jusque-là on a déduit pour LPT :

Or est la dernière tache, donc :

Les 2 cas sont distincts par leur condition sur , on en conclue que .

1. **En vous appuyant sur les questions précédentes, montrer que pour toute instance I, où est une constante dont vous donnerez la valeur.**

Nous savons que pour , donc cela ne nous intéresse pas pour trouver r.

Pour , on sait que

1. **Conclure quant au ratio d’approximation de l’algorithme LPT.**

On en conclue alors que LPT est 2-approximation.

1. **Donner, en la justifiant, la valeur de . Illustrer votre réponse sur l’instance (donc m=5).**

On a et

On peut traduire par :

On a donc , donc

Par exemple, pour

1. **Donner, en la justifiant, la valeur de . Illustrer votre réponse sur l’instance .**

Si on suppose avoir uniquement 2*m* tâches (seulement 2 tâches de durée *m*), on retrouve obligatoirement un . En effet, chaque machine a exactement 2 tâches, et chaque tâche a un « opposé » : la dernière tâche sera avec la première, l’avant-dernière avec la seconde, etc… De ce fait, les machines ont toutes un temps d’exécution . Maintenant on rajoute la dernière tâche qui se lance sur M1 (car toutes les taches se sont finies en même temps), on trouve :

Par exemple, pour

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| M1 | 9 | | | | | | | | |  |  | 5 |  |  |  |  | 5 |  |  |  |
| M2 | 9 | | | | | | | | |  |  | 5 |  |  |  |  |  |  |  |  |
| M3 | 8 | | | | | | | |  |  | 6 |  |  |  |  |  |  |  |  |  |
| M4 | 8 | | | | | | | |  |  | 6 |  |  |  |  |  |  |  |  |  |
| M5 | 7 | | | | | | |  |  |  | 7 |  |  |  |  |  |  |  |  |  |

1. **Quelle est la valeur du ratio lorsque m devient grand ? En déduire une borne inférieure sur le ratio d’approximation de l’algorithme LPT.**

On a donc un ratio , et donc .

On peut considérer comme borne inférieur pour LPT.

1. **Donner, en la justifiant, la valeur de . Illustrer votre réponse sur l’instance .**

Les durées sont générées dans un ordre croissant, donc on affecte toutes les tâches de la plus courte à la plus longue, ce qui oblige le fait qu’une machine à laquelle on vient d’affecter une tâche est actuellement la machine la plus longue (et donc la dernière à recevoir une nouvelle tâche). Si on répète cette règle lors de la distribution de tâches, on peut affirmer qu’une machine Mi recevra obligatoirement la tâche , , , etc… Or on sait qu’il n’y a que tâches, donc la seule machine à recevoir 3 tâches (dont la dernière) est la machine M1. On en conclue que :

On peut calculer en sachant que la tâche vaut . On sait que . Soit :

Par exemple, pour

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| M1 | 5 | | | | | 7 | | | | | | | 9 | | | | | | | | |
| M2 | 5 | | | | | 7 | | | | | | |  |  |  |  |  |  |  |  |  |
| M3 | 5 | | | | | 8 | | | | | | | |  |  |  |  |  |  |  |  |
| M4 | 6 | | | | | | 8 | | | | | | | |  |  |  |  |  |  |  |
| M5 | 6 | | | | | | 9 | | | | | | | | |  |  |  |  |  |  |

1. **Quelle est la valeur du ratio lorsque m devient grand ?**

On a alors le ratio , et donc .