

Problèmes d'ordonnement - Exercices - corrigé

I On considère 7 tâches devant passer sur un processeur donné...

a) La solution optimale de ce problème est obtenue en classant les tâches par temps de traitement croissant.

On obtient donc l'ordonnement : 7 6 3 1 2 4 5

Tâche	7	6	3	1	2	4	5
Durée en jours	1	2	3	4	5	6	7
Date d'achèvement	1	3	6	10	15	21	28

Durée moyenne de séjour = $(1 + 3 + 6 + 10 + 15 + 21 + 28) / 7 = 84 / 7 =$

b) Si le critère est la minimisation du plus grand des retards, l'ordonnement optimal est obtenu en classant les tâches par ordre de délai de livraison croissant.

On obtient donc l'ordonnement 1 2 4 7 3 5 6

Tâche	1	2	4	7	3	5	6
Durée en jours	4	5	6	1	3	7	2
Délai de livraison	8	10	10	14	15	17	20
Date d'achèvement	4	9	15	16	19	26	28
Retard	0	0	5	2	4	9	8

Max des retards = 9

Pour tout autre ordonnancement le plus grand retard sera au moins égal à 9.

II Ordonnement de tâches préemptives en attente devant des "machines" en parallèle et identiques

a) On considère 8 tâches en attente devant 4 machines identiques

<i>i</i>	1	2	3	4	5	6	7	8
<i>p_i</i>	13	18	22	30	20	35	16	25

Max1 = 35

Max2 = $179 / 4 = 44,75$

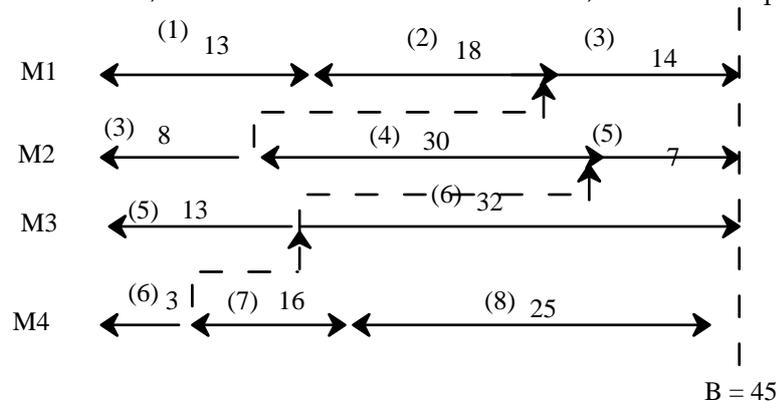
Si le temps n'est pas fractionnable au-delà de l'unité on a :

B = plus petit entier plus grand que 35 et que 44,75 soit B = 45

On place la tâche 1 de durée 13, puis la tâche 2 de durée 18, la tâche 3 doit être fractionnée, puisque la durée d'activité de M1 doit être limitée à 45. 14 unités de temps sont placées sur la machine 1. Les 8 autres sont affectées à la machine 2. La tâche 4 tient sur la machine 2 mais la tâche 5 est fractionnée

La durée de réalisation de l'ensemble des tâches est de 45.

Les tâches 3, 5 et 6 commencent sur une machine, sont interrompues, et terminent sur une autre.



L'ordonnement obtenu ainsi est clairement optimal.

b) Démontrer, dans le cas général,

Toutes les tâches devant être exécutées intégralement, le temps total est au moins égal à $\text{Max}(p_i)$. Le

temps total d'exécution est $\sum_{i=1}^n p_i$.

Donc s'il y a m machines, la durée d'exécution de toutes les tâches ne peut être inférieure à

$$\sum_{i=1}^n p_i / m$$

On en déduit donc que tout ordonnancement aura une durée au moins égale à

$$\text{Max} \left(\text{Max}(p_i), \sum_{i=1}^n p_i / m \right).$$

L'algorithme construit une affectation des tâches aux machines qui respecte cette borne.

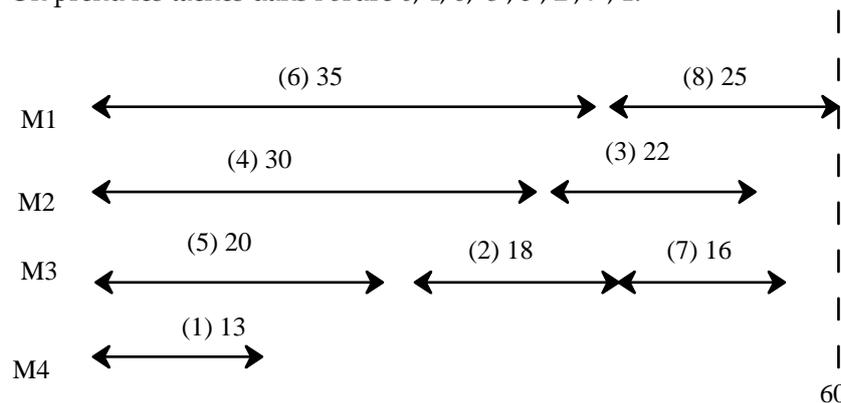
Il faut vérifier que c'est bien un ordonnancement c'est-à-dire que, lorsqu'il y a préemption pour une tâche, les 2 parties de la tâche ne sont pas faites simultanément ce qui est le cas puisque B est au moins égal à la durée de n'importe quelle tâche.

c) Dans cette question les tâches ne sont pas préemptives.

Il s'agit d'un problème de type bin packing.

On peut utiliser l'heuristique FFD pour avoir une valeur approchée du nombre de machines nécessaires pour réaliser l'ensemble des tâches dans un délai inférieur à 60. Elle consiste à classer les tâches par ordre de durée décroissante et à utiliser la machine de plus petit numéro susceptible de la réaliser sans préemption.

On prend les tâches dans l'ordre 6, 4, 8, 3, 5, 2, 7, 1.



On place la tâche 6, sur M1, la tâche 4 de durée 30 nécessite une autre machine, la tâche 8 de durée 25 peut être placée sur M1....

L'heuristique FFD donne une solution avec 4 machines.

On obtient aussi une solution à 4 machines si on prend les tâches dans l'ordre de leur numérotation.

Sur ce petit exemple, vous pourrez constater "à la main" que cette solution n'est pas optimale.

Il existe une solution avec 3 machines.

III Cet exercice reprend un exemple dû à R.Graham des Laboratoires Bell

On considère 10 tâches liées entre elles par les contraintes de

Cas 1 : on dispose de 2 machines, la durée des tâches est celle indiquée sur le graphique

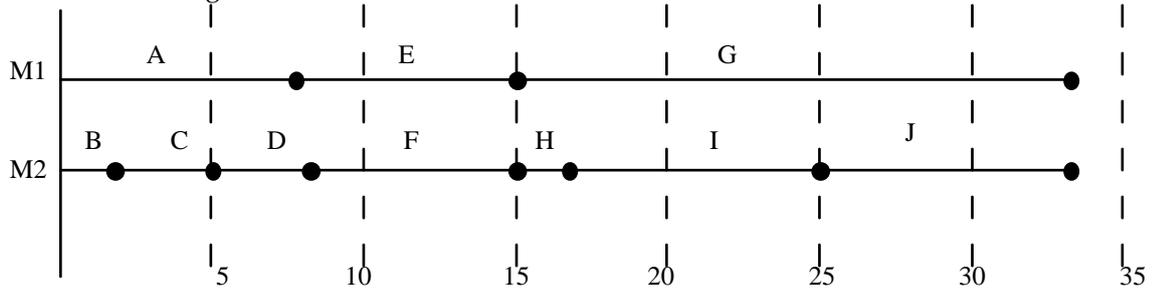
En t=0, les tâches A, B et C sont exécutables.

L'ordre de priorité défini et l'ordre des machines impliquent que A est exécutée sur M1 et B sur M2, C est en attente.

A la fin de C, C et D sont exécutables mais on place C sur M2

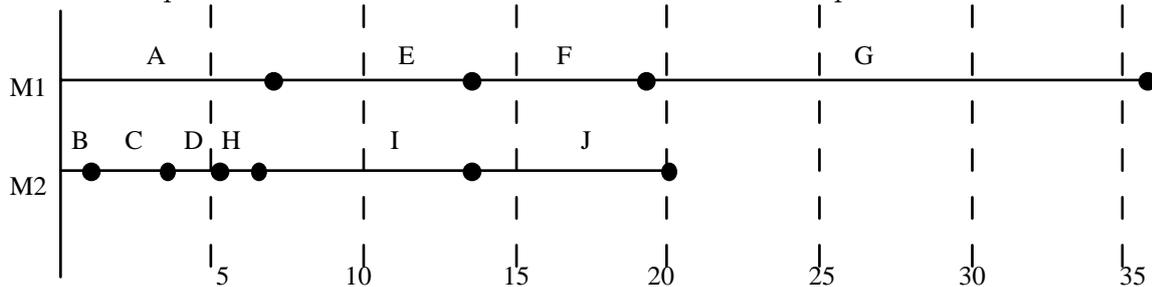
A la date 5, C est terminée, D et H sont exécutables : l'ordre de priorité retenu conduit à placer D sur M2, ainsi de suite.

On obtient le diagramme de Gantt suivant :



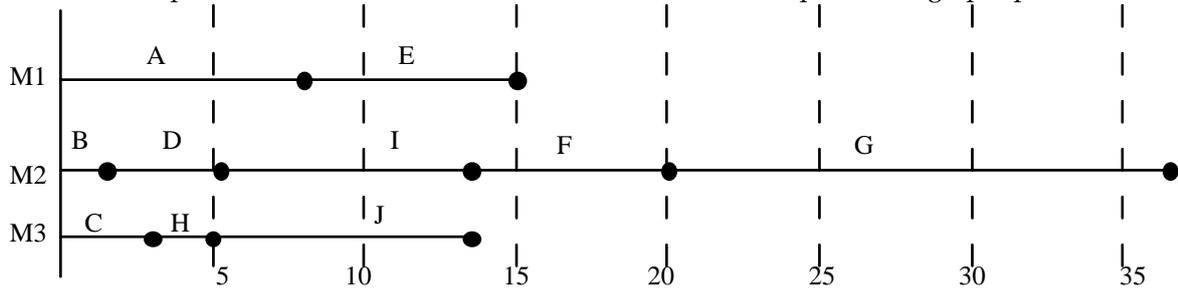
Il faut 33 unités de temps pour que l'ensemble des tâches soit exécuté.

Cas 2 : on dispose de 2 machines, les durées sont réduites d'une unité par tâche



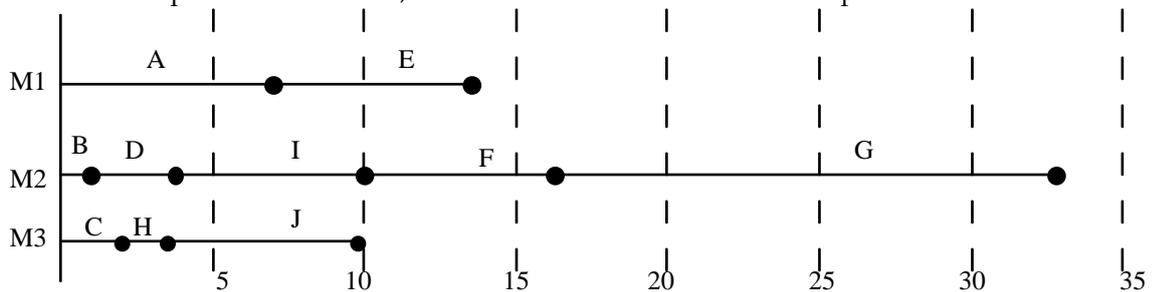
La solution obtenue est telle qu'il faut maintenant 36 unités de temps pour que l'ensemble des tâches soit exécuté, alors que la durée de chacune a été diminuée d'une unité.

Cas 3 : on dispose de 3 machines, la durée des tâches est celle indiquée sur le graphique



Durée = 38 !

Cas 4 : on dispose de 3 machines, les durées sont réduites d'une unité par tâche



La durée est de 33, comme dans le premier cas mais avec 3 machines et des durées réduites !