

## Problèmes d'ordonnancement - Exercices

I On considère 7 tâches devant passer sur une machine donnée.

On possède pour ces tâches les informations suivantes :

- elles sont toutes actuellement disponibles

- leur temps de traitement et la date à laquelle elles doivent être livrées sont donnés dans le tableau suivant :

Tâche	1	2	3	4	5	6	7
Durée en jours	4	5	3	6	7	2	1
Délai de livraison	8	10	15	10	17	20	14

a) Déterminer un ordonnancement minimisant la durée moyenne de séjour de ces tâches. Préciser l'ordre des tâches et la valeur de l'objectif à l'optimum.

b) Déterminer un ordonnancement minimisant le plus grand des retards. Préciser l'ordre des tâches et la valeur de l'objectif à l'optimum.

II Ordonnancement de tâches préemptives en attente devant des "machines" en parallèle et identiques ("machines" = n'importe quelle ressource susceptible d'effectuer une seule tâche à la fois).

On considère n tâches devant être exécutées sur des machines identiques travaillant en parallèle. La durée de la tâche i est  $p_i$ .

On dispose de m machines identiques travaillant en parallèle.

On suppose qu'il est possible d'interrompre une tâche commencée, la fin de la tâche pouvant être affectée à une autre machine. On dit que les **tâches sont préemptives**.

Les tâches ont des durées entières et le fractionnement d'une tâche se fait par durée entière.

On considère l'algorithme dont le principe est le suivant :

### Algorithme de Mac Naughton

On pose :

$$\max1 = \max(p_i) \text{ pour } i = 1, \dots, n \quad (\text{durée de la tâche la plus longue})$$

$$\max2 = \left( \sum_{i=1}^n p_i \right) / m$$

$$B = \text{le plus petit entier supérieur à } \max1 \text{ et à } \max2$$

Principe de l'algorithme :

Ordonner les tâches arbitrairement

Affecter les tâches successivement aux différentes machines

Lorsque le temps de traitement attribué à une machine dépasse B, effectuer un morcellement de la tâche, la partie restante étant attribuée à la machine suivante (dont elle deviendra la première tâche).

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

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

A quel ordonnancement des tâches sur les 4 machines conduit l'algorithme précédent ?

Quelle est la durée totale ? Pensez vous que cet ordonnancement soit optimal ?

b) Démontrer, dans le cas général, que cet algorithme fournit un ordonnancement optimal de n tâches préemptives en attente devant m machines en parallèle et identiques.

c) Dans cette question les tâches ne sont pas préemptives (*une fois commencées elles ne peuvent plus être interrompues*).

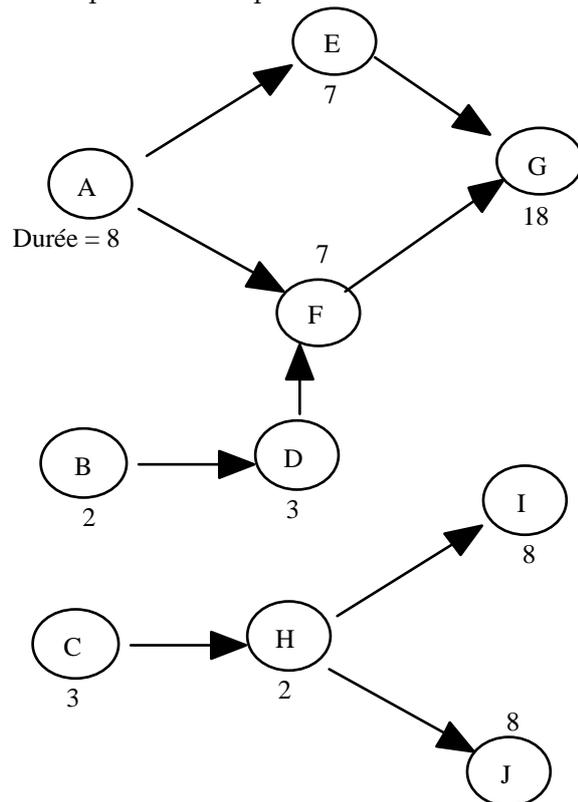
Donner une valeur approchée du nombre de machines nécessaires pour que toutes les tâches soient terminées dans un délai inférieur à 60. Indiquer la démarche utilisée.

Que pensez-vous de la solution trouvée?

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

On considère 10 tâches liées entre elles par les contraintes de succession décrites sur le diagramme ci-dessous, la durée de chaque tâche est indiquée sur le graphique : par exemple, G dont la durée est de 18 ne peut être exécutée avant la fin de E et de F.

On ne peut interrompre les tâches.



Pour réaliser ces tâches on dispose de plusieurs machines identiques travaillant en parallèle. Les machines sont numérotées.

La procédure de construction d'un ordonnancement est la suivante :

Dès qu'une machine est disponible on lui alloue une tâche exécutable.

- si plusieurs machines sont disponibles simultanément on commence par la machine de plus petit numéro
- si plusieurs tâches sont exécutables simultanément la règle de priorité retenue est l'ordre alphabétique.

Décrire le déroulement de l'exécution de ces tâches dans les cas suivants :

- on dispose de 2 machines, la durée des tâches est celle indiquée sur le graphique
- on dispose de 2 machines, les durées sont réduites d'une unité par tâche
- on dispose de 3 machines, la durée des tâches est celle indiquée sur le graphique
- on dispose de 3 machines, les durées sont réduites d'une unité par tâche