Ordonnancement.

Exercice : ordonnancement des tâches sur p processeurs identiques.

L'objectif de cet exercice est de déterminer la performance d'un ordonnancement de tâches sur p processeurs. Dans notre cas, il n'y a aucune dépendance entre les tâches entre elles. Voici la spécification du problème :

- un ensemble de tâches indépendantes : $E = \{1, 2, \ldots, n\}$.
- un ensemble de durée des tâches $w(1), w(2), \ldots, w(n)$ tel que la tâche i demande exactement $w(i)$ unités de temps pour s'exécuter. (On supposera que les tâches sont triées par ordre décroissant en fonction de la durée.)

On cherche à définir un ordonnancement σ qui attribue à une tâche i une machine $m(i)$ et une date de début d'exécution $t(i)$. Posons $D(\sigma) = max_{1 \leq i \leq n} t(i) + w(i)$. On note $D_{opt} = min\{D(\sigma')|\ \sigma'$un ordonnancement de $E\}$.

Question : En supposant que $D_{opt} < 3w(i)$ pour tout i, montrer que $n \leq 2p$, et donner un algorithme optimal.

Considérons l'algorithme de liste suivant : dès qu'une machine est libre, on lui affecte la tâche de durée maximale parmi les tâches non encore ordonnancées.

Question : Appliquer cet algorithme (3 processeurs) sur :

tâche	1	2	3	4	5	6	7
durée	6	4	4	4	3	3	1

Question : Posons $D(\sigma) = max_{1 \leq i \leq n} t(i) + w(i)$ avec σ l'ordonnancement de liste. Notons $D_{opt} = min\{D(\sigma')|\ \sigma'$un ordonnancement de $E\}$. Montrer que

$$D(\sigma) \leq D_{opt} + ((p-1)/p)d \text{ où } d \text{ durée d'une tâche finissant à } D(\sigma).$$

(indication : interressez-vous à la dernière tâche qui s'exécute et à sa date d'exécution.)

Question : Montrer que $D_{opt} \leq D(\sigma) \leq (\frac{4}{3} - \frac{1}{3p})D_{opt}$. *Indication : raisonner par l'absurde.*

Exercice : ordonnancement de tâches indépendantes sur p processeurs identiques.

L'objectif de cet exercice est de déterminer la performance d'un ordonnancement de tâches indépendantes sur p processeurs. Nous sommes dans la situation suivante. Il n'y a aucune dépendance entre les tâches entre elles. De plus, un contrôleur reçoit au fur et à mesure (une seule à la fois)[1] les tâches indépendantes : le contrôleur reçoit l'ensemble de tâches $E = \{1, 2, \ldots, n\}$ dans cet ordre. Selon un algorithme, un contrôleur affecte une tâche à un processeur et ce choix

[1] autrement dit, le contrôleur reçoit au plus une tâche par unité de temps.

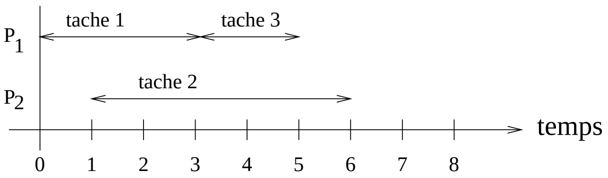
est irrévocable. Notons $d(i)$ la date d'arrivée de la tâche i. Remarquons que $d(i) < d(j)$ si $i < j$. Chaque tâche i demande exactement $w(i)$ unités de temps pour s'exécuter.

On définit un ordonnancement σ qui attribue à une tâche i une machine $m(i)$ et une date de début d'exécution $t(i)$. L'algorithme est le suivant : le contrôleur impose à une tâche i de s'exécuter à l'étape $t(i)$ sur la machine $m(i)$ si c'est la machine $m(i)$ qui est disponible à l'étape $t(i)$ le plus tôt possible parmi toutes les p machines.

Voici un exemple :

tâche	1	2	3	4	5	6	7	8
durée	3	5	2	3	3	3	1	2
date d'arrivée	0	1	2	3	7	8	9	10

On supposera qu'il y a deux processeurs. La tâche 1 arrive à l'étape 0. Comme les deux processeurs sont libres, la tâche 1 s'exécute à la date 0 sur le processeur P_1. A l'étape 1, la tâche 2 arrive et puisque le processeur P_2 est libre, la tâche 2 s'exécute à la date 1 sur ce processeur. A l'étape 2, la tâche 3 arrive. Comme le processeur P_1 sera libre le plus tôt à l'étape 3 (le processeur P_2 sera libre à l'étape 6), la tâche 3 s'exécute à la date 3 sur le processeur P_1. Et ainsi de suite...



Question 1 : Continuer l'exemple (Ordonnancer toutes les tâches de l'exemple).

On notera σ l'ordonnancement obtenu par l'algorithme précédent et $D(\sigma) = max\{t(i) + w(i) | i = 1, \ldots, n\}$ la date où l'ordonnancement σ se termine. Notons D_{opt}, la durée de l'ordonnancement optimal.

Question 2 : Donner la définition formelle de D_{opt}. En déduire un minorant de $D(\sigma)$.

Question 3 : En vous inspirant du cours, donner une borne inférieure de D_{opt} en fonction des durées des tâches.

Question 4 : Donner une borne inférieure de D_{opt} en fonction des dates d'arrivée des tâches.

Question 5 : Soit j une tâche de E se terminant en dernier.
Question 5.1 : Expliquer pourquoi la tâche j n'a pas été exécuté aux dates $d(j), \ldots, t(j) - 1$.
Question 5.2 : Que pouvez-vous en déduire sur $(t(j) - d(j))$ par rapport aux durées de toutes tâches.
Question 5.3 : Calculer la valeur β de l'inégalité suivante $D(\sigma) \leq \beta D_{opt}$. Que concluez-vous ?