MOCA B2
ED 6
Ordonnancements

1 Tâches préemptives indépendantes - algorithme de McNaughton

On considère un ensemble de 7 tâches morcelables de durées $p_1 = 5, p_2 = 3, p_3 = 2, p_4 = 6, p_5 = 3, p_6 = 2, p_7 = 1$.

Question 1 *Déterminer, à l'aide de l'algorithme de McNaughton, un ordonnancement optimal de ces tâches sur 4 machines*

Question 2 *Déterminer, à l'aide de l'algorithme de McNaughton, un ordonnancement optimal de ces tâches sur 3 machines*

2 Flow-shop sur 2 machines - algorithme de Johnson

On considère un ensemble de 8 travaux constitués chacun de deux tâches a et b. La tâche a de chaque travail s'exécute sur une machine 1, la tâche b sur une machine 2. La tâche a de chaque travail précède la tâche b. On note a_i la durée d'exécution de la tâche a du travail i et b_i la durée d'exécution de la tâche b du travail i.

Les durées des tâches sont les suivantes : $a_1 = 3, b_1 = 4, a_2 = 2, b_2 = 4, a_3 = 2, b_3 = 2, a_4 = 4, b_4 = 3, a_5 = 5, b_5 = 1, a_6 = 3, b_6 = 3, a_7 = 1, b_7 = 4, a_8 = 2, b_8 = 3$.

Question 3 *Déterminer, à l'aide de l'algorithme de Johnson, un ordonnancement de durée minimale pour ces 8 travaux*

3 Tâches unitaires sur 1 machine

Soit $I = \{1, \ldots, n\}$ un ensemble de tâches unitaires devant s'exécuter sur 1 machine. A chaque tâche i sont associés une date d'échéance d_i et un poids w_i. Une tâche i est dite en retard si sa date de fin d'exécution est supérieure à sa date d'échéance ($t_i + 1 > d_i$). Si une tâche i est en retard, un coût w_i est à payer, si i est en avance ($t_i + 1 \leq d_i$) aucun coût n'est à payer. L'objectif est de déterminer un ordonnancement de coût minimal, c'est-à-dire minimiser la somme des coûts des tâches en retard.

Question 4 *Montrer qu'il existe un ordonnancement optimal tel que la machine n'a pas de période d'inactivité*

Nous considérons l'algorithme glouton suivant pour calculer un ordonnancement : les tâches sont d'abord triées de sorte que $w_1 \geq w_2 \geq \cdots \geq w_n$. Pour la tâche i, i variant de 1 à n, on détermine, si elle existe, la dernière période libre de la machine telque i ne soit pas en retard. Si cette période existe alors i est affectée a cette période, sinon i est affectée à la dernière période libre de la machine.