

1 Minimisation du coût maximal

On se propose d'ordonnancer sur une machine un ensemble $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ de n tâches non préemptives de durées respectives p_1, p_2, \dots, p_n de manière à minimiser $\max\{g_i(C_i) : i = 1, 2, \dots, n\}$ où C_i est la date de fin de la tâche J_i et g_i une fonction croissante associée à la tâche J_i .

Question 1. Soient $P = \sum_{i=1}^n p_i$ et J_k la tâche telle que $g_k(P) = \min\{g_i(P) : i = 1, 2, \dots, n\}$. Montrer qu'il existe un ordonnancement optimal tel que J_k soit la dernière tâche exécutée.

Question 2. Pour l'instance définie dans le tableau ci-dessous. Identifier la tâche J_k .

i	1	2	3
p_i	2	3	5
$g_i(C_i)$	$1 + C_i$	$2C_i$	10

Question 3. Dans le cas général, donner le principe d'un algorithme calculant un ordonnancement optimal.

Question 4. Évaluer la complexité temporelle de votre algorithme.

Question 5. Appliquer votre algorithme à l'instance de la question 2.

Question 6. On suppose maintenant qu'à chaque tâche J_i est associée une date échue d_i et que $g_i(C_i)$ est le retard absolu T_i de la tâche J_i , où $T_i = \max(C_i - d_i, 0)$. Quel résultat vu en cours retrouvez-vous ?