MPRO - BOR
Bases de l'ordonnancement

Examen Janvier 2019

L'examen dure 2h30. Les notes de cours sont autorisées. Les livres ne sont pas autorisés. La clarté de rédaction sera prise en compte dans la notation. Le barème est indicatif.

Exercice 1 (8 points)

Ordonnancement et apprentissage

On considère un problème d'ordonnancement à une machine qui doit exécuter un ensemble de n tâches. On s'intéresse au cas particulier où la durée opératoire d'une tâche peut diminuer si on suppose que des tâches ont déjà été exécutées par la machine. Cette situation illustre un effet d'apprentissage pour lequel plus la tâche apparaît tard dans la séquence plus vite elle sera exécutée.

On suppose dans la suite de l'exercice qu'une tâche J_i possède une durée *nominale* p_i. Si la tâche J_i est la première à être exécutée, sa durée sera égale à p_i, si elle est exécutée à une position r sa durée sera égale à $p_{ir} = p_i \times r^a$ avec $a \leq 0$. Le paramètre a est un indicateur d'apprentissage.
Pour illustrer cette notion, si on suppose que l'indicateur d'apprentissage $a = -\frac{1}{2}$, une tâche J_i de durée nominale 100 qui serait exécutée en deuxième position dans un ordonnancement aurait une durée $p_{i2} = 100 \times 2^{-\frac{1}{2}} = 70,71$.

Question 1 (1/8) — Exprimer les valeurs p_{ir} pour $i = 1, \cdots, n$ et $r = 1, \cdots, n$ en fonction des durées nominales p_i et du paramètre a.

Question 2 (2, 5/8) — On considère le problème $1|p_{ir} = p_i r^a|C_{\max}$.

- a On suppose dans cette question que la contrainte d'apprentissage est relâchée. On considère donc le problème $1| - |C_{\max}$. Comment résoudre optimalement ce problème ? Justifiez votre réponse.
- b Est-ce que le même principe peut-être utilisé pour résoudre le problème $1|p_{ir} = p_i r^a|C_{\max}$? Justifiez votre réponse.
- c Montrer que l'ordonnancement obtenu en séquençant les tâches dans l'ordre des durées opératoires nominales croissantes est optimal pour $1|p_{ir} = p_i r^a|C_{\max}$. La preuve pourra être établie par argument d'échange.

Question 3 (2, 5/8) — On considère le problème $1|p_{ir} = p_i r^a| \sum_i C_i$.

- a On suppose dans cette question que la contrainte d'apprentissage est relâchée. On considère donc le problème $1| - | \sum_i C_i$. Rappelez la propriété vue en cours pour résoudre ce problème ?
- b Montrer que cette propriété permet de résoudre le problème $1|p_{ir} = p_i r^a| \sum_i C_i$. La preuve pourra être établie par argument d'échange.

Question 4 (2/8) — On considère le problème $1|p_{ir} = p_i r^a|L_{\max}$.

- a On suppose dans cette question que la contrainte d'apprentissage est relâchée. On considère donc le problème $1| - |L_{\max}$. Rappelez la propriété vue en cours pour résoudre ce problème ?
- b Cette propriété permet-elle de résoudre le problème $1|p_{ir} = p_i r^a|L_{\max}$. Si oui, le prouver, sinon fournir un contre-exemple.

Exercice 2 (7 points)
Ordonnancement Juste à Temps

Cet exercice porte sur un problème d'ordonnancement à une machine avec pénalités d'avance et de retard pour lequel on autorise des *préemptions* sur l'exécution des tâches.

Il s'agit de déterminer un ordonnancement optimal σ de n tâches $J_1, \cdots, J_n$ sur une machine. Chaque tâche J_i possède une durée opératoire p_i et doit être idéalement exécutée avant (ou au plus tard à) une date d'échéance d_i. On associe à chaque tâche J_i une pénalité unitaire d'avance α_i et une pénalité unitaire de retard β_i. Si l'on note C_i la date de fin d'exécution de la tâche J_i, l'avance (resp. le retard) de la tâche J_i est définie par $E_i = \max\{0, d_i - C_i\} = (d_i - C_i)^+$ (resp. $T_i = \max\{0, C_i - d_i\} = (C_i - d_i)^+$. La date de début d'exécution de la tâche J_i est notée S_i. On souhaite de plus pénaliser les interruptions des tâches, en effet, si une tâche est interrompue, son en-cours de production augmente. Le temps de séjour de la tâche J_i est défini par $C_i - S_i - p_i$. On introduit une pénalité d'en-cours γ_i pour chaque tâche J_i et on suppose que $\gamma_i \geq \alpha_i$. On s'intéresse à la minimisation de $c(\sigma) = \sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i + \gamma_i(C_i - S_i - p_i))$.

Question 1 (1, 5/7) — Calculer le coût de la solution définie par la séquence à 3 tâches décrite à la figure 1 pour $\alpha = (1, 1, 2)$, $\beta = (2, 3, 1)$, $\gamma = (5, 5, 4)$ et $d = (10, 8, 9)$.

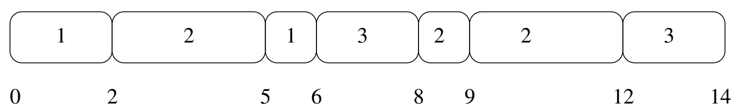

FIGURE 1 – Séquence à 3 tâches

Question 2 (1/7) — Que peut-on dire de la complexité du problème ?
Question 3 (2/7) — Montrer que si dans un ordonnancement σ, il existe un temps d'inactivité pour une tâche J_i entre S_i et C_i, alors on peut construire un ordonnancement σ' sans temps d'inactivité entre S'_i et C'_i avec $c(\sigma') \leq c(\sigma)$.
Question 4 (2/7) — Montrer que si dans un ordonnancement σ, il existe une paire de tâches (J_i, J_j) tel que $S_i \leq S_j \leq C_i \leq C_j$ alors on peut construire un ordonnancement σ' tel que $S_i \leq C_i \leq S_j \leq C_j$ et $c(\sigma') \leq c(\sigma)$.
Question 5 (0, 5/7) — Déduire de la question précédente, les différentes façons d'ordonnancer tout couple de tâches (J_i, J_j) dans un ordonnancement optimal.

Exercice 3 (5 points)
Ordonnancement avec communications

On considère un problème d'ordonnancement avec communications sous l'hypothèse petits délais de communication (i.e. $\max\{c_{ij}\} \leq \min\{p_i\}$) sur un nombre illimité de processeurs. Le critère à optimiser est la durée C_{max} de l'ordonnancement.

On note $G = (T, <)$ le graphe de précédence correspondant.

Un *pont* de G est un arc $(i, j) \in <$ tel que : $\Gamma^+(i) = \{j\}, \Gamma^-(j) = \{i\}$.

On note p_i la durée d'une tâche $i \in T$, c_{ij} la durée de communication associée à l'arc $(i, j) \in <$.

On notera t_i la date de début d'exécution de i dans un ordonnancement, π_i le processeur exécutant i.

Question 1 — Montrer qu'il existe un ordonnancement optimal de G tel que pour tout pont $(i, j) \in A$ on a $\pi_i = \pi_j$ et $t_j = t_i + p_i$.