1. On souhaites séquencer sur une machine un ensemble $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ de n tâches non préemptives de durées respectives $p_1, p_2, \cdots, p_n$.
   (a) Quel est l'ordre optimal d'exécution des tâches si l'on souhaite minimiser la somme des dates de fin des tâches (ou durée moyenne de réalisation) $\sum_{i=1}^{n} C_i$? Justifier.
   (b) On suppose maintenant qu'un poids positif ω_i est associé à chaque tâche et que l'on souhaite minimiser la somme pondérée des dates de fin des tâche (ou stock d'en-cours) $\sum_{i=1}^{n} \omega_i C_i$. Montrer que l'ordre optimal est obtenu en séquençant les tâches dans l'ordre des p_i/ω_i (Weighted Shortest Processing Times).
   (c) En déduire une solution optimale et son coût pour l'exemple donné dans le tableau ci-dessous.

i	1	2	3	4	5	6
p_i	8	6	3	7	4	8
ω_i	8	3	6	7	8	1

2. Montrer que l'ordre $SRPT$ est optimal pour $1|r_i,\ pmtn|\sum_{i=1}^{n} C_i$.

3. Soit C_j la valeur de la date de fin de la tâche J_i obtenue par l'ordonnancement réaliser suivant la règle de Smith. Prouver l'inégalité suivante :

$$C_j \leq r_j + \sum_{k \leq j} p_k$$

4. En déduire que la valeur de la solution obtenue en appliquant la règle de Smith ne peut excéder $2 \cdot W^*$, ou W^* désigne la valeur de la solution optimale de l'instance considérée.

5. Démontrer que l'ordre EDD (ordre Earliest Due Date) construit une solution optimale pour $1||L_{max}$.

6. Soit $K \subseteq \mathcal{J}$. On définit la fonction ensembliste $h(K)$ par :

$$h(K) = \min_{j \in K} r_j + \sum_{j \in K} p_j - \max_{j \in K} d_j$$

Montrer que $h(K)$ est une évaluation par défaut de la valeur d'un ordonnancement.

1. On souhaites séquencer sur une machine un ensemble $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ de n tâches non préemptives de durées respectives $p_1, p_2, \cdots, p_n$.
   (a) Quel est l'ordre optimal d'exécution des tâches si l'on souhaite minimiser la somme des dates de fin des tâches (ou durée moyenne de réalisation) $\sum_{i=1}^{n} C_i$? Justifier.

Solution: Soit $J_{i_1}, J_{i_2}, \cdots, J_{i_n}$ un ordre quelconque. On a :

$$C_{i_1} + C_{i_2} + \cdots + C_{i_n} = p_{i_1} + (p_{i_1} + p_{i_2}) + \cdots + (p_{i_1} + \cdots + p_{i_n})$$

Donc :

$$\sum_{i=1}^{n} C_i = np_{i_1} + (n-1)p_{i_2} + \cdots + p_{i_n}$$

On obtient par conséquent un ordre optimal en séquençant les taches dans l'ordre des p_i croissants. Cet ordre est appelé SPT (Shortest Processing Times).
On notera que cette règle permet également de minimiser le temps moyen de séjour dans l'atelier $\sum_{i=1}^{n}(C_i - r_i)$ ou encore le retard algébrique moyen de séjour lorsque des dates de début au plus tôt r_i ou des dates échues d_i sont associées aux taches.

   (b) On suppose maintenant qu'un poids positif ω_i est associé à chaque tâche et que l'on souhaite minimiser la somme pondérée des dates de fin des tâche (ou stock d'en-cours) $\sum_{i=1}^{n} \omega_i C_i$. Montrer que l'ordre optimal est obtenu en séquençant les tâches dans l'ordre des p_i/ω_i (Weighted Shortest Processing Times).

Solution: Nous allons démontrer cette affirmation par l'absurde.
Soient deux taches J_{i_q} et $J_{i_{q+1}}$ telles que $p_{i_q}/\omega_{i_q} > p_{i_{q+1}}/\omega_{i_{q+1}}$. Supposons qu'il existe une séquence S optimale dans laquelle l'ordre $WSPT$ n'est pas respecté pour ces deux taches (i.e. la tache J_{i_q} est ordonnancée avant $J_{i_{q+1}}$). Considérons la séquence S' obtenue à partir de S en inversant l'ordre d'exécution des deux taches J_{i_q} et $J_{i_{q+1}}$.
Comparons la valeur φ de la fonction objectif pour la séquence S avec la valeur φ' de cette même fonction pour la séquence S'. On a :

$$\varphi' - \varphi = \omega_{i_q}(C'_{i_q} - C_{i_q}) + \omega_{i_{q+1}}(C'_{i_{q+1}} - C_{i_{q+1}})$$

ou C_i est la date de fin de la tache J_i dans la séquence S et C'_i sa date de fin dans la séquence S'.
Ou encore :

$$\varphi' - \varphi = \omega_{i_q} p_{i_{q+1}} - \omega_{i_{q+1}} p_{i_q}$$

Or nous avons supposé que $p_{i_q}/\omega_{i_q} > p_{i_{q+1}}/\omega_{i_{q+1}}$, soit $p_{i_q}\omega_{i_{q+1}} > p_{i_{q+1}}\omega_{i_q}$. Nous en déduisons que $\varphi' - \varphi < 0$ et donc que la séquence S' est meilleure que S. Ce qui est contradictoire avec l'hypothèse que S est optimale.
L'ordre $WSPT$ est aussi appelé règle de Smith. Cet ordre permet également de minimiser le temps moyen pondéré de séjour $\sum_{i=1}^{n} \omega_i(C_i - r_i)$ ou le retard algébrique moyen pondéré $\sum_{i=1}^{n} \omega_i(C_i - d_i)$

(c) En déduire une solution optimale et son coût pour l'exemple donné dans le tableau ci-dessous.

i	1	2	3	4	5	6
p_i	8	6	3	7	4	8
ω_i	8	3	6	7	8	1

Solution: D'apres la partie (b), pour construire une solution optimale, il suffit de séquencer les taches par p_i/ω_i croissants. Il ya donc autant de solutions optimales que de sequences satisfaisant cet ordre. Dans cet exemple, nous avons : $p_1/\omega_1 = 1$, $p_2/\omega_2 = 2$, $p_3/\omega_3 = 0.5$, $p_4/\omega_4 = 1$, $p_5/\omega_5 = 0.5$ et $p_6/\omega_6 = 8$. Il y a donc quatre solution optimales (car $p_1/\omega_1 = p_4/\omega_4$ et $p_3/\omega_3 = p_5/\omega_5$). Dans cette solution, $\sum_{i=1}^{n} \omega_i C_i = 18+56+120+154+84+36$, soit 468.

2. Montrer que l'ordre $SRPT$ est optimal pour $1|r_i,\ pmtn|\sum_{i=1}^{n} C_i$.

Solution: Considérons un instant t au cours duquel deux taches J_i et J_j, de durées résiduelles strictement positives p_i^+ et p_j^+, sont disponibles. On suppose l'inégalité $p_i^+ < p_j^+$ vérifiée et que l'on ordonnance à l'instant t la tache J_j (en contradiction avec la critère SRPT). Soient C_i et C_j les dates de fin d'exécution des deux taches obtenues à la fin du processus d'ordonnancement. Sans perte de généralité, on peut toujours considérer que l'on a $C_j < C_i$ (sinon on peut toujours permuter à l'unité J_j séquence a l'instant t une unité de la tache J_i sans dégrader la valeur de l'ordonnancement). Sachant que la durée résiduelle d'exécution de J_j à l'instant t est strictement supérieure à celle de J_i, on peut clairement permuter les p_i^+ premiers unités de J_j rencontrées avec les unités de J_i ordonnancées au-delà de l'instant t. On observe alors que l'on obtient un nouvel ordonnancement dont les dates de fin d'exécution des taches notées C' ne sont pas modifiées, hormis pour C'_i et C'_j, pour lesquelles on a : $C'_j = C_i$ et $C'_i < C_j$. La valeur de ce nouvel ordonnancement est donc clairement préférable au précédent. Par induction, on déduit donc que l'on a toujours intérêt à ordonnancer la tache de durée résiduelle minimale.

3. Soit C_j la valeur de la date de fin de la tâche J_i obtenue par l'ordonnancement réaliser suivant la règle de Smith. Prouver l'inégalité suivante :

$$C_j \leq r_j + \sum_{k \leq j} p_k$$

Solution: Les taches étant supposées numérotées dans l'ordre correspondant à la règle de Smith, seules les taches d'indice inférieur à j sont prioritaires que J_j. Au pire, l'exécution de ces taches s'effectue intégralement entre la mise à disposition de J_j et sa date de fin d'exécution, ce qui nous conduit à l'inégalité :

$$C_j \leq r_j + \sum_{k \leq j} p_k$$

4. En déduire que la valeur de la solution obtenue en appliquant la règle de Smith ne peut excéder $2 \cdot W^*$, ou W^* désigne la valeur de la solution optimale de l'instance considérée.

Solution: D'après la question précédente, on a : $C_j \leq r_j + \sum_{k \leq j} p_k$. On en déduit :

$$\sum_j \omega_j C_j \leq \sum_j \omega_j r_j + \sum_j \left(\omega_j \sum_{k \leq j} p_k \right)$$

Étant donnée que la date de fin d'exécution d'une tache est toujours supérieure à sa date de mise à disposition, on a clairement : $\sum_j \omega_j r_j \leq W_*$. De même, $\sum_j \left(\omega_j \sum_{k \leq j} p_k \right)$ correspond à la valeur de l'ordonnancement optimal pour le problème dans lequel toutes les dates de mise à disposition ont été relaxées à la valeur 0. On a donc également $\sum_j \left(\omega_j \sum_{k \leq j} p_k \right) \leq W^*$, d'ou :

$$\sum_j \omega_j C_j \leq 2 \cdot W^*$$

5. Démontrer que l'ordre EDD (ordre Earliest Due Date) construit une solution optimale pour $1||L_{max}$.

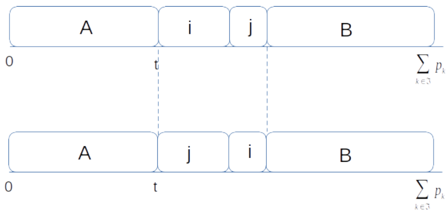
Solution: Considérons une solution S dans laquelle deux taches i et j séquencées consécutivement vérifient $d_i > d_j$ (c'est a dire ne respectant pas l'ordre EDD). Soit S' la solution obtenue en permutant les taches i et j. On se propose de montrer que la valeur de la solution S' est toujours inférieure ou égale à la valeur de la solution S.
Soient L_S, L_A, L_B, L_i et L_j le retard maximal des séquences S, A, B et des taches i et j pour la solution S (même notation pour S' en remplaçant L par L'). On a :

$$\begin{aligned} L_S &= max\{L_A, L_B, L_i, L_j\} = \max\{L_A, L_B, \max\{L_i, L_j\}\} \\ L'_{S'} &= max\{L'_A, L'_B, L'_i, L'_j\} = \max\{L'_A, L'_B, \max\{L'_i, L'_j\}\} \end{aligned}$$

Or, clairement, on observe : $L_A = L'_A$ et $L_B = L'_B$. En outre, on a :

$$\begin{array}{rcl} L_i &=& t + p_i - d_i \\ L_j &=& t + p_i + p_j - d_j \\ L'_i &=& t + p_i + p_j - d_i \quad \leq t + p_i + p_j - d_j = L_j \quad (d_i > d_j) \\ L'_j &=& t + p_j - d_j \quad \leq t + p_i + p_j - d_j = L_j \quad (p_i > 0) \end{array}$$

On en déduit $L'_{S'} \leq L_S$.



6. Soit $K \subseteq \mathcal{J}$. On définit la fonction ensembliste $h(K)$ par :

$$h(K) = \min_{j\in K} r_j + \sum_{j\in K} p_j - \max_{j\in K} d_j$$

Montrer que $h(K)$ est une évaluation par défaut de la valeur d'un ordonnancement.

Solution: Aucune tache de K ne peut démarrer avant l'instant $\min_{j\in K} r_j$. Par conséquent la dernière tache de K termine au plus tôt à l'instant $\min_{j\in K} r_j + \sum_{j\in K} p_j$. Puisque $\max_{j\in K} d_j$ représente la valeur la plus grande date échue d'une tache de K, on en déduit que $h(k)$ est une évaluation par défaut de la valeur de tout ordonnancement (et donc l'ordonnancement optimal).