PROBLÈMES D'ORDONNANCEMENT À UNE MACHINE

Cadre On considère une famille de tâches $(J_i)_{i\in[\![1..n]\!]}$, associée des durées resp. de ces tâches $(p_i)_{i\in[\![1..n]\!]}$.
On s'intéresse à des problèmes d'ordonnancement de ces tâches lorsqu'on a une seule machine, c-à-d des problèmes $1|-|-$ dans la typologie $\alpha|\beta|\gamma$.

Rmq (Le mode non préemptif)
Pour un critère régulier on sait déduire d'une séquence optimale un ordonnancement optimal, en effet si on nous dit dans quel ordre faire les tâches il suffit de "les caler à gauche" c'est-à-dire de les commencer le plus tôt possible.
Or toutes les séquences possibles des tâches $(J_i)_{i\in[\![1..n]\!]}$ sont en bijection avec les permutations de n éléments par $\left(\begin{array}{ccc} \mathfrak{S}_n & \longrightarrow & \text{séquences} \\ \sigma & \longmapsto & (J_{\sigma(1)}, J_{\sigma(2)}, \dots J_{\sigma(n)}) \end{array}\right)$

A retenir [Dans la séquence associée à σ la i-ième tâche est $J_{\sigma(i)}$

Ccl Sachant cela, on travaillera à chercher une permutation optimale, c-à-d qui réalise une séquence optimale, elle-même fournissant un ordo. optimal.

1er problème

$1|-|\sum_{i=1}^{n} C_i$ — objectif à minimiser où C_i est la date de fin de J_i

- une seule machine
- pas de contraintes sur les tâches, elles ne sont pas "préemptables" ie non sécables.

$$\text{Ici coût}(\sigma) = \sum_{i=1}^{n} C_i(\sigma) = \sum_{i=1}^{n} C_{\sigma(i)}(\sigma)$$ car σ est une bijection de $[\![1..n]\!]$

et on sait exprimer $C_{\sigma(i)}$ la date de fin de la $i$ème tâche réalisée, il suffit de sommer les durées de la 1ère tâche, de la deuxième ... jusqu'à la $i$ème d'où $\boxed{C_{\sigma(i)} = \sum_{j=1}^{i} p_{\sigma(j)}}$ ★

Donc $\text{coût}(\sigma) = \sum_{i=1}^{n} \sum_{j=1}^{i} p_{\sigma(j)}$

$= \sum_{j=1}^{n} p_{\sigma(j)} \times \sum_{i=j}^{n} 1$ $\Big\} = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}_{j \le i}\, p_{\sigma(j)}$

$= \sum_{j=1}^{n} (n+1-j)\, p_{\sigma(j)}$

La question est maintenant de choisir σ minimisant cette f ; c'est à dire pour chaque j de choisir $p_{\sigma(j)}$

Quand j est petit, 1 par exemple, $p_{\sigma(j)}$ va beaucoup compter dans la somme (d'un coeff n ici) on cherche donc à prendre $p_{\sigma(j)}$ petit, par exemple $\sigma(1)$ l'indice de la tache la plus courte (ie $p_{\sigma(1)}$ le + petit des p_i).

↳ Cela donne l'intuition de prendre σ tq $p_{\sigma(1)} \nearrow$

MQ σ est optimale ssi $(p_{\sigma(i)})_{i \in [\![1..n]\!]} \nearrow$ = ie "σ vérifie la $\nearrow$"

⇒ Par contraposée. (ARGUMENT D'ÉCHANGE)

Si $(p_{\sigma(i)})_{i \in [\![1..n]\!]}$ n'est pas $\nearrow$, il existe $q \in [\![1..n[\![$ tq $p_{\sigma(q)} > p_{\sigma(q+1)}$

On pose alors $\tilde{\sigma} : i \to \begin{cases} \sigma(i) \text{ si } i \neq q, i \neq q+1 \\ \sigma(q+1) \text{ si } i = q \\ \sigma(q) \text{ si } i = q+1 \end{cases}$

(évidemment $\tilde{\sigma} \in \mathfrak{S}_n$ aussi !)

autrement dit on envisage d'échanger les q-ième et $q+1$ ème tâches.

Alors $\text{coût}(\sigma) - \text{coût}(\tilde{\sigma}) = \sum_{j=1}^{n} (n+1-j)\left(p_{\sigma(j)} - p_{\tilde{\sigma}(j)}\right)$

$= 0$ si $j \notin \{q, q+1\}$

$= (n+1-q)\left(p_{\sigma(q)} - p_{\tilde{\sigma}(q)=\sigma(q+1)}\right)$

$+ (n+1-(q+1))\left(p_{\sigma(q+1)} - p_{\tilde{\sigma}(q+1)=\sigma(q)}\right)$

$= \left(p_{\sigma(q)} - p_{\sigma(q+1)}\right)\left[n+1-q - (n+1-q-1)\right]$

> 0 ; $= 1$

Donc $\text{coût}(\tilde{\sigma}) < \text{coût}(\sigma)$, donc σ n'est pas optimale.

Par contraposée si σ est optimale, elle vérifie nécessairement $(p_{\sigma(i)})_{i \in [\![1..n]\!]} \nearrow$.

⇐ Il s'agit maintenant de MQ la croissance de $(p_{\sigma(i)})_{i \in [\![1..n]\!]}$ sachant que σ est optimale

CAS 1 Si les $(p_i)_{i \in [\![1..n]\!]}$ sont 2 à 2 $\neq$, une seule permutation vérifie la croissance, c'est nécessairement elle l'optimum.

CAS 2 S'il y a des égalités dans les $(p_i)_{i\in[1..n]}$ il nous faut montrer que toutes les permutation vérifiant la ↗ ont le même coût, et qu'elles sont alors toutes optimales.

Quitte à imaginer le graphe de ces permuta où on relie deux permutations si elles ne diffèrent que d'un échange de deux termes consécutifs, on peut par un argument de connexité se ramener à vérifier que le coût ne varie pas entre deux permuta voisines en ce sens.

Si $\begin{cases} \sigma = \tilde{\sigma} \text{ sur } [1..n]\setminus\{q,q+1\} \\ \sigma \text{ et } \sigma' \text{ vérifient la croissance.} \end{cases}$ avec $\begin{matrix} \sigma(q) = \tilde{\sigma}(q+1) \\ \sigma(q+1) = \tilde{\sigma}(q) \end{matrix}$

alors $\begin{matrix} p_{\sigma(q)} & \leq & p_{\sigma(q+1)} & \text{par } \nearrow \text{ de } \sigma \\ \| & & \| & \\ p_{\tilde{\sigma}(q+1)} & \geq & p_{\tilde{\sigma}(q)} & \text{par } \nearrow \text{ de } \tilde{\sigma} \end{matrix}$

donc nécessairement $p_{\sigma(q)} = p_{\sigma(q+1)}$

donc $\text{coût}(\sigma) = \sum_{j\in\{q,q+1\}} (n+1-j)\, p_{\sigma(j)} + p_{\sigma(q)} \times \left[(n+1-q)+(n+1-(q+1))\right]$

$= \sum_{j\in\{q,q+1\}} (n+1-j)\, p_{\tilde{\sigma}(j)} + p_{\tilde{\sigma}(q+1)} \times \left[(n+1-(q+1))+(n+1-q)\right]$

$= \text{coût}(\tilde{\sigma})$. CQFD.

CCL 1 $\boxed{1 \mid\mid -\sum_{i=1}^{n} C_i}$ se réduit à un problème de tri, il est donc en $O(n \log n)$ et $\in P$

L'ordre choisi ici, de la tâche la + courte à la plus longue, s'appelle l'ordre **SPT**

shortest processing time

Rq À $(d_i)_{i\in[1..n]}$ et $(r_i)_{i\in[1..n]}$ fixés, minimiser $\sum C_i - r_i$ ($\min_\sigma \sum C_i(\sigma) - r_i$) revient à minimiser $\sum C_i$. / $\sum C_i - d_i$ ($\min_\sigma \sum C_i(\sigma) - d_i$)

On a donc $\boxed{1 \mid r_i \mid \sum_{i=1}^{n} C_i - r_i} \in P$ via SPT

une seule machine toujours

f° objectif régulières

$\boxed{1 \mid d_i \mid \sum_{i=1}^{n} C_i - d_i} \in P$ via SPT

2ème problème

$1 \mid - \mid \sum_{i=1}^{n} w_i C_i$

- 1 : une seule machine
- $-$: pas de contraintes sur les tâches non préemptables
- $\sum_{i=1}^{n} w_i C_i$: fct objectif régulière où les w_i sont des coeff fixés, les C_i encore les dates de fin

Ici $\text{coût}(\sigma) = \sum_{i=1}^{n} w_i C_i(\sigma) = \sum_{i=1}^{n} w_{\sigma(i)} C_{\sigma(i)}(\sigma) = \sum_{i=1}^{n} w_{\sigma(i)} \times \sum_{j=1}^{i} p_{\sigma(j)}$

$= \sum_{j=1}^{n} \left(\sum_{i=j}^{n} w_{\sigma(i)} \right) p_{\sigma(j)}$.

↳ avec la même technique d'interversion de somme on n'est pas bien avancé car les coeff dépendent de σ, contrairement à $(n+1-j)$.

Mais reprenons un peu de la hauteur sur ce que l'on a fait pour le problème précédent : on les a classés par longueur croissante. Partant du "principe" que le tps c'est de l'argent on les a classés de la moins chère à la plus chère. Ici, en admettant que w_j représente l'importance, la valeur de la tâche J_j, il est naturel de comparer la durée de la tâche avec sa valeur, de faire le rapport de ce qu'elle coûte sur ce qu'elle apporte : on va donc plutôt trier selon $\frac{p}{w}$, ie de la "moins cher pour ce que c'est" à la "plus cher pour ce que c'est".

MQ σ est optimale ssi $\left(\frac{p_{\sigma(i)}}{w_{\sigma(i)}} \right)_{i \in [1..n]}$ → nouvelle hypothèse de croissance.

$\Rightarrow$ Par contraposée RE ARGUMENT D'ÉCHANGE

Si $\left(\frac{p_{\sigma(i)}}{w_{\sigma(i)}} \right)_{i \in [1..n]}$ n'est ~~pas~~ → il existe $q \in [1..n[$ tq $\frac{p_{\sigma(q)}}{w_{\sigma(q)}} > \frac{p_{\sigma(q+1)}}{w_{\sigma(q+1)}}$

On pose à nouveau $\tilde{\sigma} = \begin{cases} \sigma(i) & \text{si } i \notin \{q, q+1\} \\ \sigma(q+1) & \text{si } i = q \\ \sigma(q) & \text{si } i = q+1 \end{cases} = \sigma \circ (q+1, q)$

Si $i < q$: $C_{\tilde{\sigma}(i)}(\tilde{\sigma}) = \sum_{j=1}^{i} p_{\tilde{\sigma}(j)} = \sum_{j=1}^{i} p_{\sigma(j)} = C_{\sigma(i)}(\sigma)$

Si $i > q+1$: $C_{\tilde{\sigma}(i)}(\tilde{\sigma}) = \sum_{j=1}^{q-1} p_{\tilde{\sigma}(j)} + p_{\tilde{\sigma}(q)} + p_{\tilde{\sigma}(q+1)} + \sum_{j=q+1}^{i} p_{\tilde{\sigma}(j)}$

$= \sum_{j=1}^{q-1} p_{\sigma(j)} + p_{\sigma(q+1)} + p_{\sigma(q)} + \sum_{j=q+1}^{i} p_{\sigma(j)} = C_{\sigma(i)}(\sigma)$

Donc $\text{Coût}(\sigma) - \text{Coût}(\tilde{\sigma}) = \sum_{i \notin \{q, q+1\}} \underbrace{w_{\sigma(i)} C_{\sigma(i)}(\sigma) - w_{\tilde{\sigma}(i)} C_{\tilde{\sigma}(i)}(\tilde{\sigma})}_{=0} + w_{\sigma(q)} C_{\sigma(q)}(\sigma) + w_{\sigma(q+1)} C_{\sigma(q+1)}(\sigma) - w_{\tilde{\sigma}(q)} C_{\tilde{\sigma}(q)}(\sigma) - w_{\tilde{\sigma}(q+1)} C_{\tilde{\sigma}(q+1)}(\tilde{\sigma})$

$= w_{\sigma(q)} \left[C_{\sigma(q)}(\sigma) - C_{\tilde{\sigma}(q+1)}(\tilde{\sigma}) \right] + w_{\sigma(q+1)} \left[C_{\sigma(q+1)}(\sigma) - C_{\tilde{\sigma}(q)}(\tilde{\sigma}) \right]$

$= w_{\sigma(q)} \left[\sum_{i=1}^{q-1} p_{\sigma(i)} + p_{\sigma(q)} - \left(\sum_{i=1}^{q-1} p_{\tilde{\sigma}(i)} + p_{\tilde{\sigma}(q)} + p_{\tilde{\sigma}(q)} \right) \right]$

$+ w_{\sigma(q+1)} \left[\sum_{i=1}^{q-1} p_{\sigma(i)} + p_{\sigma(q)} + p_{\sigma(q+1)} - \left(\sum_{i=1}^{q-1} p_{\tilde{\sigma}(i)} + p_{\tilde{\sigma}(q)} \right) \right]$

Donc $\text{coût}(\sigma) - \text{coût}(\tilde{\sigma}) = -w_{\sigma(q)} p_{\sigma(q+1)} + w_{\sigma(q+1)} p_{\sigma(q)}$

Or puisque $\frac{p_{\sigma(q)}}{w_{\sigma(q)}} > \frac{p_{\sigma(q+1)}}{w_{\sigma(q+1)}}$ on a $p_{\sigma(q)} w_{\sigma(q+1)} > p_{\sigma(q+1)} w_{\sigma(q)}$

donc $\text{coût}(\sigma) - \text{coût}(\tilde{\sigma}) > 0$ soit $\text{coût}(\tilde{\sigma}) < \text{coût}(\sigma)$

donc σ n'est pas optimale.

⇐ Comme pour le 1er pb il nous suffit de RQ deux permutations vérifiant la croissance $\left(\frac{p_{\sigma(i)}}{w_{\sigma(i)}}\right)_i$ ↗ et ne différant que d'une transposition de la forme $(q, q+1)$, ont le même coût, qui sera alors nécessairement le coût optimal.

On considère donc σ et σ' vérifiant la ↗ et tq $\sigma' = \sigma \circ (q, q+1)$. △
Comme dans ⇒ on a alors

$\text{coût}(\sigma) - \text{coût}(\sigma') = -w_{\sigma(q)} p_{\sigma(q+1)} + w_{\sigma(q+1)} p_{\sigma(q)}$

On a $\frac{p_{\sigma(q)}}{w_{\sigma(q)}} \leq \frac{p_{\sigma(q+1)}}{w_{\sigma(q+1)}} = \frac{p_{\sigma'(q)}}{w_{\sigma'(q)}} \leq \frac{p_{\sigma'(q+1)}}{w_{\sigma'(q+1)}} = \frac{p_{\sigma(q)}}{w_{\sigma(q)}}$

(↗ de σ ; △ ; △ ; ↗ de σ')

donc $\frac{p_{\sigma(q)}}{w_{\sigma(q)}} = \frac{p_{\sigma(q+1)}}{w_{\sigma(q+1)}}$ et donc $\text{coût}(\sigma) = \text{coût}(\sigma')$ CQFD.

Cl 2

$1 \mid\mid -\mid \sum_{i=1}^{n} w_i C_i \in P$ via WSPT

weighted shortest processing time

puisqu'ici aussi on est ramené à un problème de tri.

3ème problème

$1 \mid r_i, pmtn \mid \sum C_i$

- 1 : une seule machine
- r_i : date de début au plus tôt des tâches
- pmtn : préemption des tâches autorisées
- $\sum C_i$: l'objectif