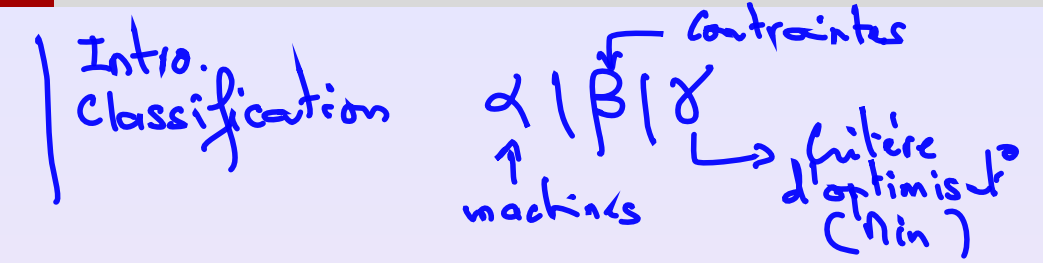


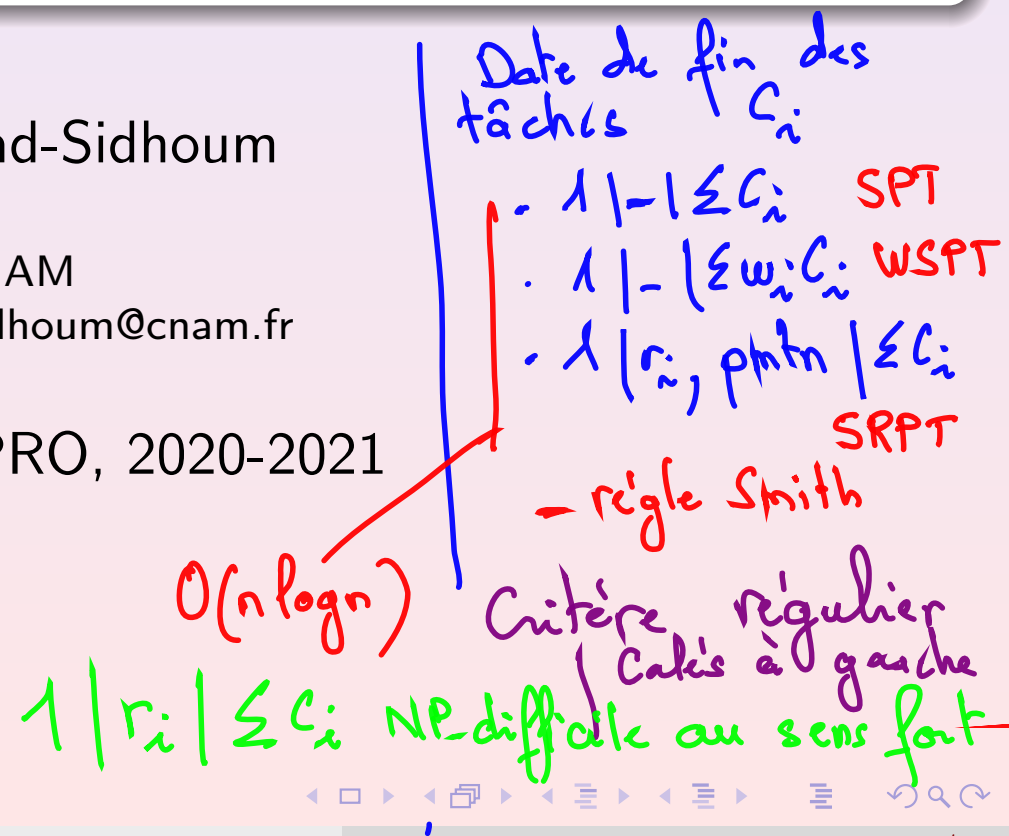
Bases de l'ordonnancement

Problèmes à une machine - Dates d'échéance

Safia Kedad-Sidhoum

CNAM
safia.kedad_sidhoum@cnam.fr

BOR - M2 MPRO, 2020-2021



Problèmes à une machine

Règle de Jackson

$1||-||L_{max}$ $\quad\quad$ $\max_i L_i$ / $L_i = C_i - d_i$

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, L_{max}
  Règle EDD. Preuve
- Définition Algorithme de liste.
- Autres problèmes : $1|r_i, p_i = 1|L_{max}$, $1|r_i, pmtn|L_{max}$
  Complexité ? Preuve

Remarque
$T_{max} = \max_i T_i$
$T_i = \max(0, C_i - d_i)$
Toute solution optimale pour $1||-|L_{max}$ sera optimale pour $1||-|T_{max}$

Problèmes à une machine

Règle de Jackson

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, $L_{\max}$
  Règle EDD. Preuve (Earliest Due Date)
- Définition Algorithme de liste.
- Autres problèmes : $1|r_i, p_i = 1|L_{\max}$, $1|r_i, pmtn|L_{\max}$
  Complexité ? Preuve

Problèmes à une machine

Règle de Jackson

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, $L_{\max}$

  Règle EDD. Preuve
- Définition Algorithme de liste.
- Autres problèmes : $1|r_i, p_i = 1|L_{\max}$, $1|r_i, pmtn|L_{\max}$

  Complexité ? Preuve

Problèmes à une machine

Règle de Jackson

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, $L_{\max}$

  Règle EDD. Preuve
- Définition Algorithme de liste.
- Autres problèmes : $1|r_i, p_i = 1|L_{\max}$, $1|r_i, pmtn|L_{\max}$

  Complexité ? Preuve

Problèmes à une machine

Règle de Jackson

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

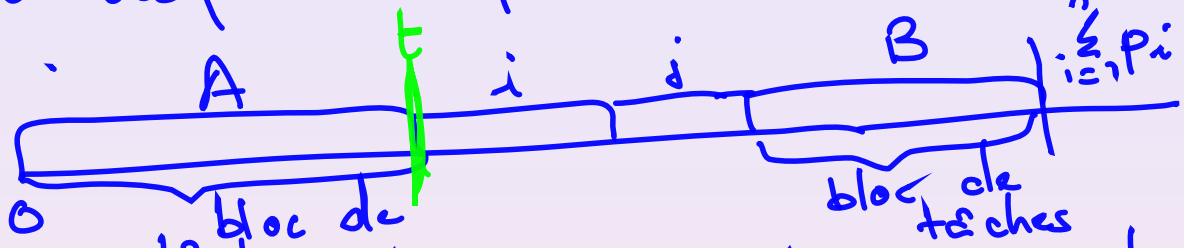
- Critère : minimiser le retard algébrique maximal, $L_{\max}$

  Règle EDD. Preuve
- Définition Algorithme de liste.
- Autres problèmes : $1|r_i, p_i = 1|L_{\max}$, $1|r_i, pmtn|L_{\max}$

  Complexité ? Preuve

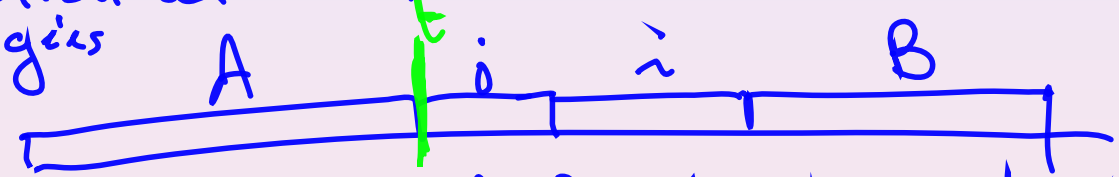
Propriété Ordre EDD construit 1 solution optimale pour $1||L_{max}$.

Preuve Soit 1 ordonnancement S (optimal) dans lequel 2 tâches consécutives i et j ne respectent pas l'ordre EDD, c.a.d $i < j$ et $d_i > d_j$.



$L_S \leftarrow S$ $\quad (\max_{i \in [1,n]} L_i^{(S)})$

On construit un ordonnancement S' / i et j sont échangées



$L_{S'} \leftarrow S'$ $\quad (\max_{i \in [1,n]} L_i^{(S')})$

On s'intéresse au calcul de $L_S - L_{S'}$

$L_S = \max\{L_A^{(S)}, L_B^{(S)}, L_i^{(S)}, L_j^{(S)}\}$

$L_{S'} = \max\{L_A^{(S')}, L_B^{(S')}, L_i^{(S')}, L_j^{(S')}\}$

On a $\begin{cases} L_A^{(S')} = L_A^{(S)} \\ \text{et } L_B^{(S')} = L_B^{(S)} \end{cases}$

$L_i^{(S)} = C_i^{(S)} - d_i = t + p_i - d_i$

$L_j^{(S)} = t + p_i + p_j - d_j$

$L_i^{(S')} = t + p_j + p_i - d_i$

$L_j^{(S')} = t + p_j - d_j$

On a $\begin{cases} L_i^{(S')} \leq L_i^{(S)} \\ L_j^{(S')} \leq L_j^{(S)} \end{cases}$ car $\boxed{d_i > d_j}$

$L_A^{(J)} = L_A^{(S')}$ et $L_B^{(S)} = L_B^{(S')}$

$\max \{ L_i^{(S)}, L_j^{(S)} \}$

$\max \{ L_i^{(J')}, L_j^{(S')} \} \leq L$

$L_{S'} \leq L_S$

Ordo. EDD est optimal.

Algorithme de Liste

Soit $L = (i_1, i_2, \dots, i_n)$ une liste de n tâches.
Soit l'ordonnancement S_L associé à L et obtenu en appliquant la règle :

Dès qu'1 machine est libre (au moins), exécuter les tâches dans l'ordre défini par L.

Propriété : L'ordonnancement de liste construit suivant la règle de Jackson est optimal pour $1 \mid r_i, p_i = 1 \mid L_{max}$ et pour $1 \mid r_i, pmtn \mid L_{max}$ entiers unitaires) $\Rightarrow$ Aucune tâche ne devient disponible durant l'exécution d'une autre tâche.

EDD s'applique dans ce cas (cf. preuve précédente)

dates de disponibilité $\rightarrow r_i$, p_i

$(S_i \geq r_i)$

S_i : date de début

Problèmes à une machine

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, $L_{\max}$

  Contraintes : Dates de disponibilité r_i

  Complexité ? Preuve

  Résolution exacte

Problèmes à une machine

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, $L_{\max}$

  Contraintes : Dates de disponibilité r_i

  Complexité ? Preuve

  Résolution exacte

Problèmes à une machine

On souhaite ordonnancer sur une machine, un ensemble de n tâches non préemptives de durée p_i et de dates d'échéance d_i.

- Critère : minimiser le retard algébrique maximal, $L_{\max}$

  Contraintes : Dates de disponibilité r_i

  Complexité ? Preuve

  Résolution exacte