Approche polyédrale pour les problèmes d'ordonnancement juste-à-temps à une machine

Anne-Elisabeth FALQ
encadrée par Pierre Fouillhoux et Safia Kedad-Sidhoum

LIP6- Laboratoire d'informatique de Paris 6

Avril-Août 2017

slides disponibles sur `http://perso.eleves.ens-rennes.fr/~afalq494/recherche-stage.html`

Un premier problème

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

Un ordonnancement solution =

- une famille de périodes d'exécution deux à deux disjointes

But : minimiser la somme pondérée des dates de fin

Illustration sur un exemple

Une instance : $J = \{1, 2, 3\}$

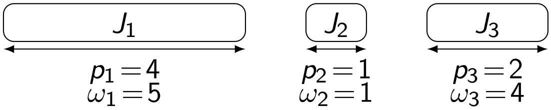
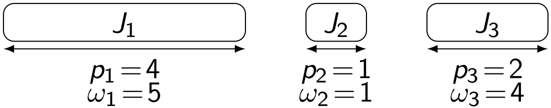


Illustration sur un exemple

Une instance : $J = \{1, 2, 3\}$



Un ordonnancement solution :

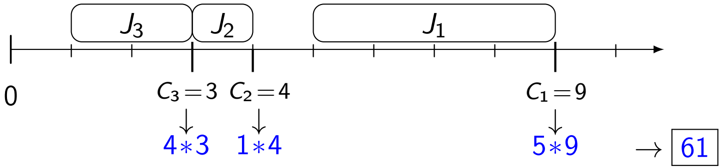
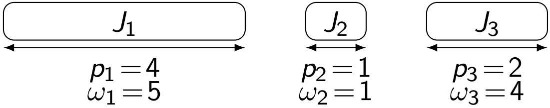
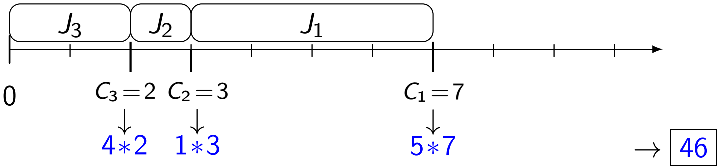


Illustration sur un exemple

Une instance : $J = \{1, 2, 3\}$



Un ordonnancement solution :



Notre problème juste-à-temps

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- une date d'échéance commune et non restrictive $d \geqslant \sum p_j$
- des coûts de retard pour chacune de ces tâches $(\alpha_j)_{j \in J}$
- des coûts d'avance pour chacune de ces tâches $(\beta_j)_{j \in J}$

Un ordonnancement solution =

- une famille de périodes d'exécution deux à deux disjointes

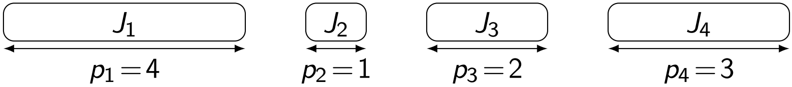
Illustration sur un exemple

Une instance : $J = \{1, 2, 3, 4\}$

J_1	J_2	J_3	J_4
$p_1 = 4$	$p_2 = 1$	$p_3 = 2$	$p_4 = 3$

Illustration sur un exemple

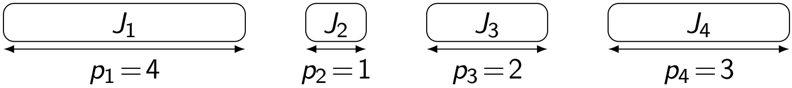
Une instance : $J=\{1,2,3,4\}$



poids tous égaux : $\forall j \in J$, $\alpha_j = \beta_j = 4$ et $d = 10$

Illustration sur un exemple

Une instance : $J = \{1, 2, 3, 4\}$



poids tous égaux : $\forall j \in J$, $\alpha_j = \beta_j = 4$ et $d = 10$

Un ordonnancement solution :

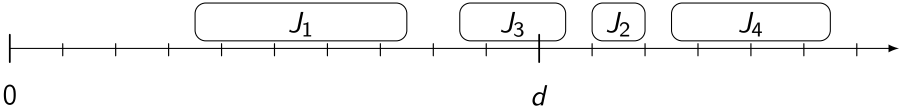
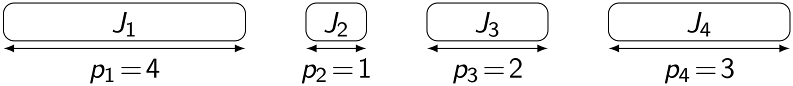


Illustration sur un exemple

Une instance : $J = \{1, 2, 3, 4\}$



poids tous égaux : $\forall j \in J$, $\alpha_j = \beta_j = 4$ et $d = 10$

Un ordonnancement solution :

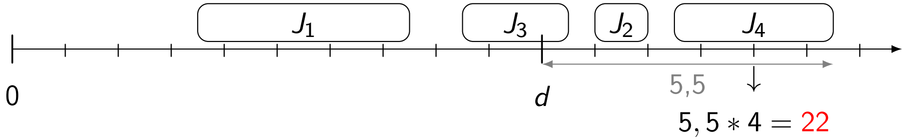
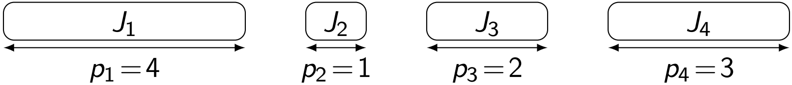


Illustration sur un exemple

Une instance : $J=\{1,2,3,4\}$



poids tous égaux : $\forall j \in J,\ \alpha_j = \beta_j = 4$ et $d = 10$

Un ordonnancement solution :

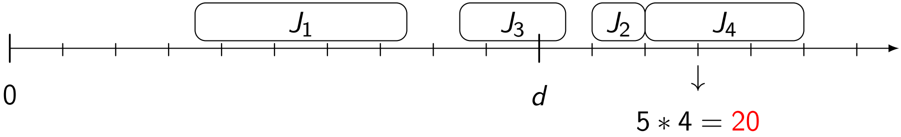
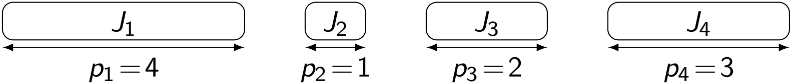


Illustration sur un exemple

Une instance : $J=\{1,2,3,4\}$



poids tous égaux : $\forall j \in J,\ \alpha_j = \beta_j = 4$ et $d = 10$

Un ordonnancement solution :

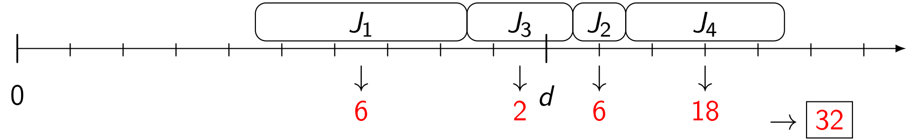
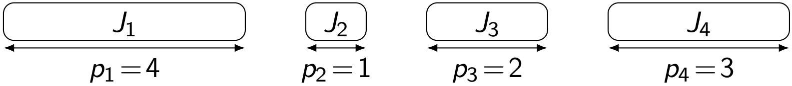


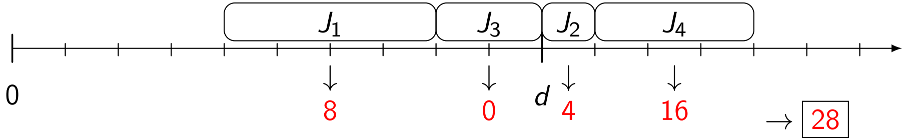
Illustration sur un exemple

Une instance : $J = \{1, 2, 3, 4\}$

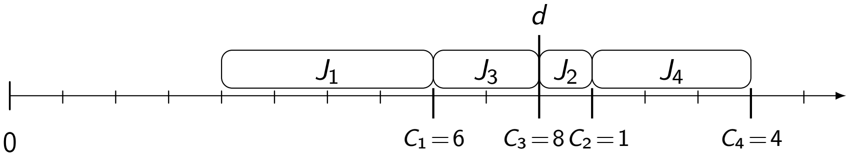


poids tous égaux : $\forall j \in J,\ \alpha_j = \beta_j = 4$ et $d = 10$

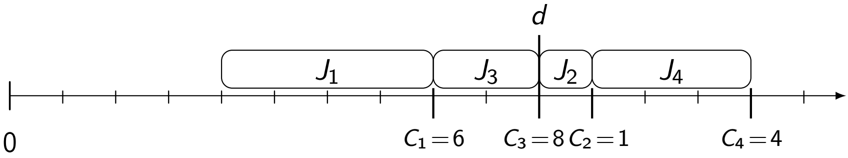
Un ordonnancement solution :



Codage par les $(C_j)_{j \in J}$

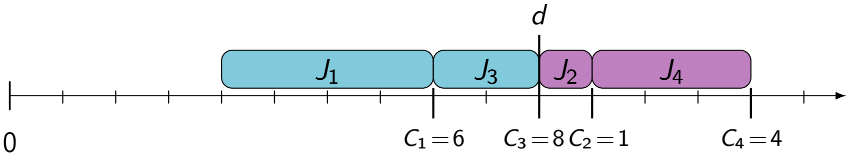


Codage par les $(C_j)_{j \in J}$



Le coût s'écrit alors $\boxed{\sum_{j \in J} \alpha_j * [C_j - d]^+ + \beta_j * [d - C_j]^+}$

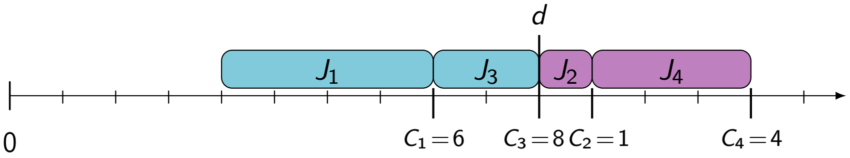
Codage par les $(C_j)_{j \in J}$



Le coût s'écrit alors $\boxed{\sum_{j \in J} \alpha_j * [C_j - d]^+ + \beta_j * [d - C_j]^+}$

En notant $E = \{j \in J \mid C_j \leqslant d_j\}$ les tâches en avance
$T = \{j \in J \mid C_j > d_j\}$ les tâches en retard

Codage par les $(C_j)_{j\in J}$

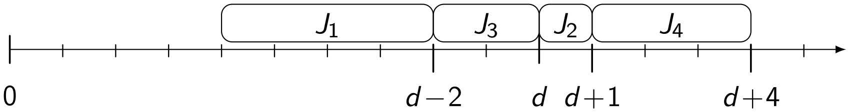


Le coût s'écrit alors $\boxed{\sum_{j\in J} \alpha_j * [C_j - d]^+ + \beta_j * [d - C_j]^+}$

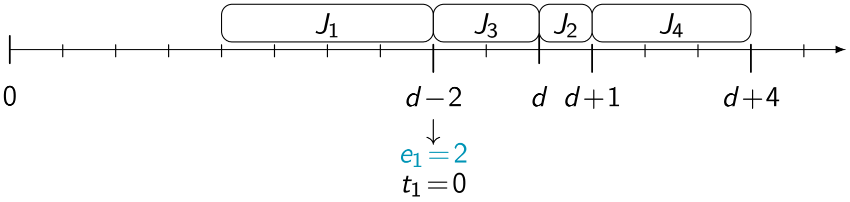
En notant $E = \{j \in J \mid C_j \leqslant d_j\}$ les tâches en avance
$T = \{j \in J \mid C_j > d_j\}$ les tâches en retard

le coût s'écrit aussi $\boxed{\sum_{j\in E} \alpha_j * (C_j - d) + \sum_{j\in T} \beta_j * (d - C_j)}$

Codage par les $(e_j)_{j\in J}$ et les $(t_j)_{j\in J}$

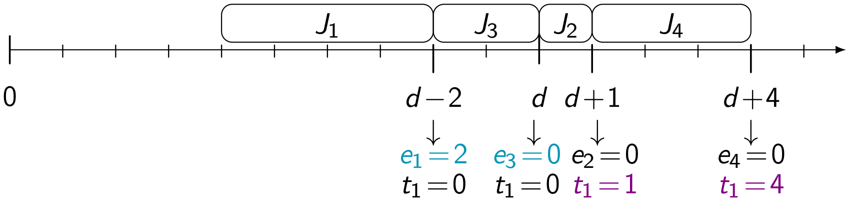


Codage par les $(e_j)_{j\in J}$ et les $(t_j)_{j\in J}$



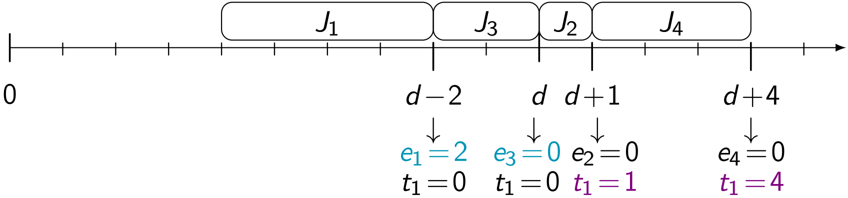
Codage par les $(e_j)_{j\in J}$ et les $(t_j)_{j\in J}$

On note $\forall j \in J$, $e_j = [d - C_j]^+$ et $t_j = [C_j - d]^+$



Codage par les $(e_j)_{j\in J}$ et les $(t_j)_{j\in J}$

On note $\forall j \in J,\ e_j = [d - C_j]^+$ et $t_j = [C_j - d]^+$



Le coût s'écrit alors $\boxed{\sum_{j\in J} \alpha_j * e_j + \beta_j * t_j}$ $\leftarrow$ linéaire en les variables e_j / t_j

Plan

Le problème

Hypothèses supplémentaires :

- coûts symétriques $\forall j \in J,\ \alpha_j = \beta_j$
- coûts indépendants de la tâches $\forall j \in J,\ \alpha_j = \alpha$ et $\forall j \in J,\ \beta_j = \beta$

$\hookrightarrow$ tous les coûts égaux
$\forall j \in J,\ \alpha_j = \beta_j =$ une constante qu'on suppose valoir 1

Rappel sur les dominances

Si **tous** les ordonnancements optimaux vérifient une propriété $\mathtt{p}$,
on dit qu'on a **dominance stricte** des ordonnancements vérifiant $\mathtt{p}$

$\boxed{\text{optimalité} \Rightarrow \mathtt{p}}$

Si **au moins un** ordonnancement optimal vérifie la propriété $\mathtt{p}$,
on dit alors qu'on a **dominance large** des ordonnancements vérifiant $\mathtt{p}$

optimalité $\not\Rightarrow \mathtt{p}$

Illustration de la dominance [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il est sans trou

Propriété

Pour le problème avec poids unitaires,
on a dominance stricte des ordonnancements

Illustration de la dominance 🕛 [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie 🕛 $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements 🕛

Illustration de la dominance [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements

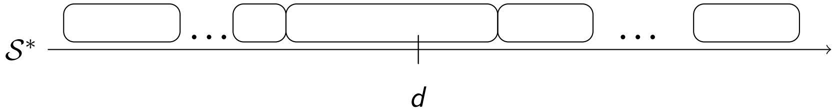


Illustration de la dominance 🕛 [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie 🕛 $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements 🕛

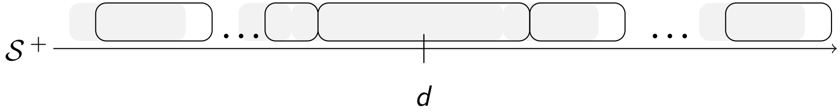
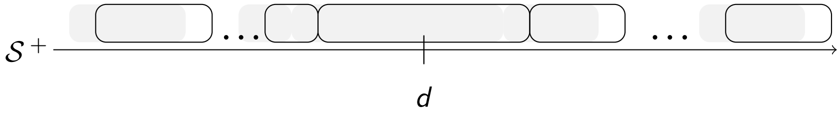


Illustration de la dominance [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements



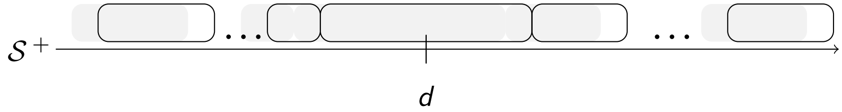
$$\text{coût}(\mathcal{S}^+) = \text{coût}(\mathcal{S}^*) - \varepsilon * \sum_{j \in E} \alpha_j + \varepsilon * \sum_{j \in T} \beta_j$$

Illustration de la dominance 🕛 [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie 🕛 $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements 🕛



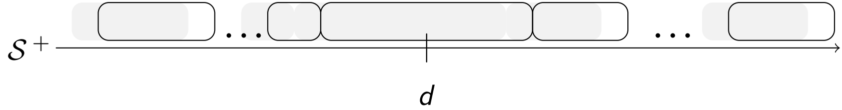
$$\text{coût}(\mathcal{S}^+) = \text{coût}(\mathcal{S}^*) - \varepsilon * \underbrace{\sum_{j \in E} \alpha_j}_{\alpha(E)} + \varepsilon * \underbrace{\sum_{j \in T} \beta_j}_{\beta(T)}$$

Illustration de la dominance [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements



$$\text{coût}(\mathcal{S}^+) = \text{coût}(\mathcal{S}^*) - \varepsilon * [\alpha(E) - \beta(T)] \Rightarrow \alpha(E) \geqslant \beta(T)$$

Illustration de la dominance [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements

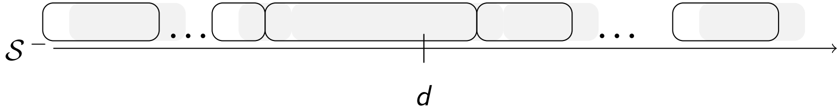
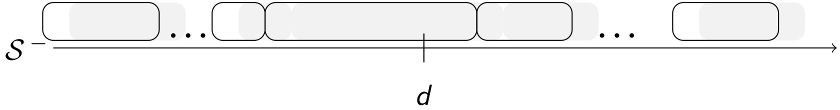


Illustration de la dominance [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements



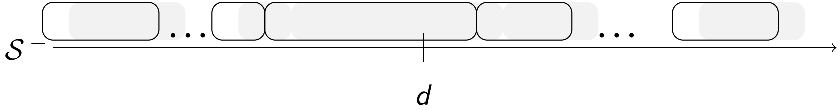
$$\text{coût}(\mathcal{S}^-) = \text{coût}(\mathcal{S}^*) - \varepsilon * [\beta(T) - \alpha(E)]$$

Illustration de la dominance 🕛 [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie 🕛 $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

Pour le problème avec poids unitaires,
on a dominance large des ordonnancements 🕛



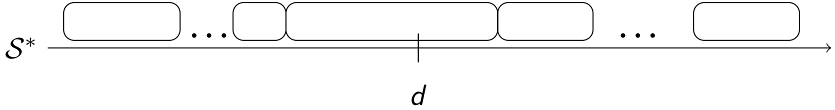
$$\text{coût}(\mathcal{S}^-) = \text{coût}(\mathcal{S}^*) - \varepsilon * [\beta(T) - \alpha(E)] \Rightarrow \alpha(E) \leqslant \beta(T)$$

Illustration de la dominance 🕛 [Kanet,81]

L'ordonnancement $\mathcal{S}$ vérifie 🕛 $\Leftrightarrow$ il admet une tâche qui finit à l'heure

Propriété

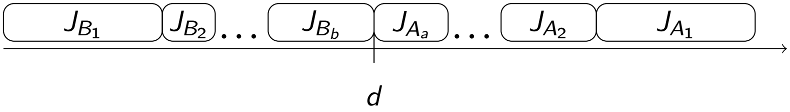
Pour le problème avec poids unitaires,
on a dominance large des ordonnancements 🕛



donc $\alpha(E) = \beta(T)$ et donc coût($\mathcal{S}^+$)=coût($\mathcal{S}^-$)=coût($\mathcal{S}^*$)

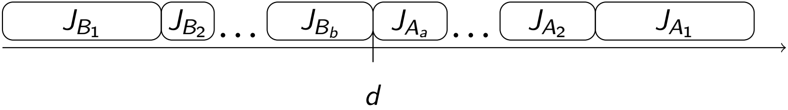
Utilisation des dominances [Kanet,81]

On cherche l'optimum parmi les ordonnancements et
donc de la forme :



Utilisation des dominances [Kanet,81]

On cherche l'optimum parmi les ordonnancements et
donc de la forme :

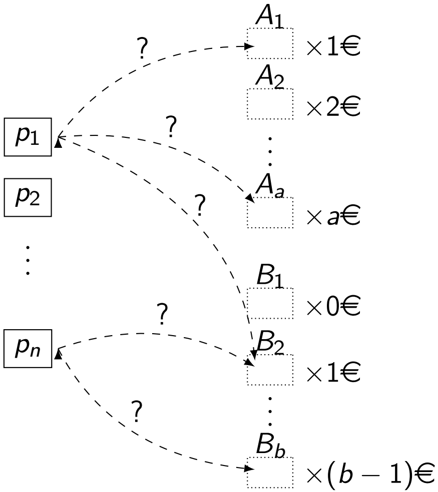


le coût s'écrit $\boxed{\sum_{k=1}^{a} k * p_{A_k} + \sum_{k=1}^{b} (k-1) * p_{B_k}}$

Se ramener à un problème d'affectation [Kanet,81]

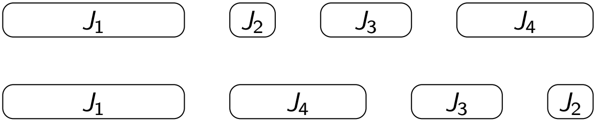
Quand a et b sont fixés
on est ramené à un simple
problème d'affectation

$$\boxed{\sum_{k=1}^{a} k * p_{A_k} + \sum_{k=1}^{b} (k-1) * p_{B_k}}$$

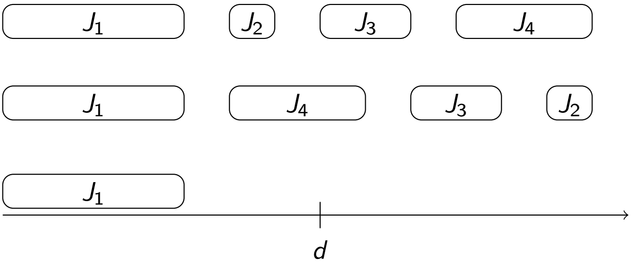


L'algorithme de Kanet sur l'exemple [Kanet,81]

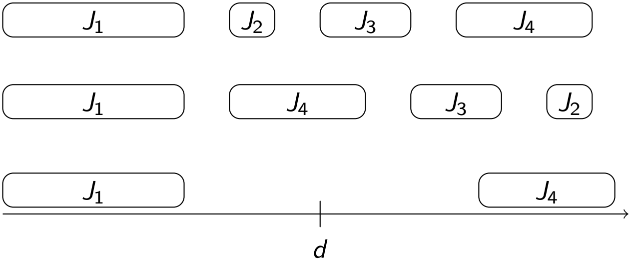
L'algorithme de Kanet sur l'exemple [Kanet,81]



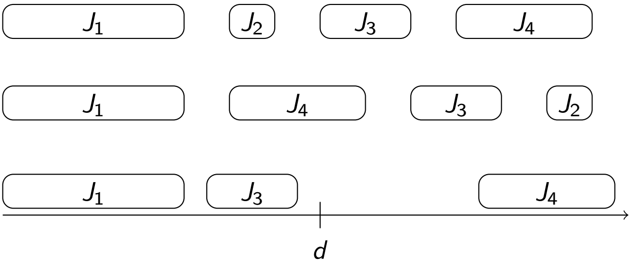
L'algorithme de Kanet sur l'exemple [Kanet,81]



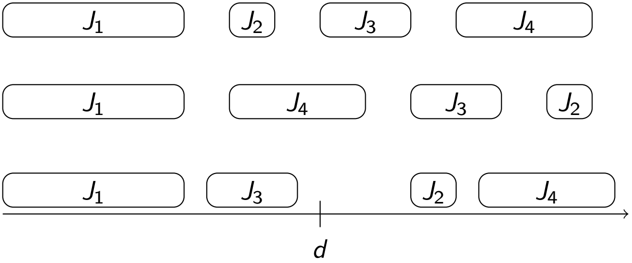
L'algorithme de Kanet sur l'exemple [Kanet,81]



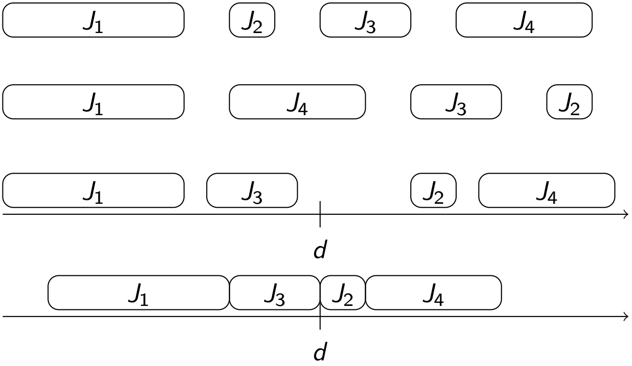
L'algorithme de Kanet sur l'exemple [Kanet,81]



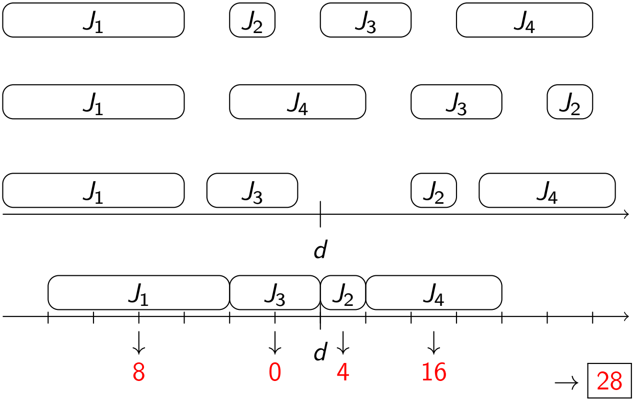
L'algorithme de Kanet sur l'exemple [Kanet,81]



L'algorithme de Kanet sur l'exemple [Kanet,81]



L'algorithme de Kanet sur l'exemple [Kanet,81]



Conclusion

Algorithme de Kanet en $O(n \log n)$
$\hookrightarrow$ le problème avec poids unitaires $\alpha_j = \beta_j = 1$ est dans P.

Le problème

Hypothèses supplémentaires :

- coûts symétriques $\forall j \in J,\ \alpha_j = \beta_j$

$\hookrightarrow$ on pose $\forall j \in J,\ \boldsymbol{\omega_j} := \alpha_j = \beta_j$

Ici le coût dépend de la tâche !

Dominances [Hall & Posner,91]

On pose $\forall j \in J$, $\boldsymbol{r_j} := \omega_j / p_j$

Dominances [Hall & Posner,91]

On pose $\forall j \in J$, $\boldsymbol{r_j} := \omega_j / p_j$

Propriété

Pour le problème avec coûts symétriques, on a :

- *dominance large des ordonnancements* 
- *dominance large des ordonnancements* 

Dominances [Hall & Posner,91]

On pose $\forall j \in J,\ \boldsymbol{r_j} := \omega_j / p_j$

Propriété

Pour le problème avec coûts symétriques, on a :

- *dominance large des ordonnancements*
- *dominance large des ordonnancements*
- *dominance stricte des ordonnancements* $\nearrow\searrow^r$

où $\mathcal{S}$ vérifie $\nearrow\searrow^r \Leftrightarrow \begin{cases} \text{les tâches en avance sont par } r \text{ croissant} \\ \text{les tâches en retard sont par } r \text{ décroissant} \end{cases}$

Complexité du problème [Hall & Posner,91]

- algorithme de programmation dynamique en $O(nP)$ où $P = \sum_{j \in J} p_j$

Complexité du problème [Hall & Posner,91]

- algorithme de programmation dynamique en $O(nP)$ où $P = \sum_{j \in J} p_j$

c'est une complexité pseudo-polynomiale

Complexité du problème [Hall & Posner,91]

- algorithme de programmation dynamique en $O(nP)$ où $P = \sum_{j \in J} p_j$

c'est une complexité pseudo-polynomiale

Propriété

Le problème avec poids symétriques est NP-difficile au sens faible

1. Introduction

2. État de l'art de l'ordonnancement juste-à-temps

3. État de l'art sur l'approche polyédrale du problème min $\sum \omega_i C_i$
   - Structure du polyèdre
   - Description du polyèdre par des inégalités
   - Une autre preuve, constructive

4. Étude polyédrale du problème juste-à-temps (coûts quelconques)

5. Étude expérimentale

6. Conclusion

Le problème

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

Un ordonnancement solution =

- une famille de périodes d'exécution deux à deux disjointes

But : minimiser $\sum \omega_i C_i$

Règle de Smith [Smith,56]

L'ordonnancement $\mathcal{S}$ vérifie  $\Leftrightarrow$ il est sans trou et tassé à gauche

L'ordonnancement $\mathcal{S}$ vérifie $\searrow^r$ $\Leftrightarrow$ les tâches sont par ratio décroissant où $\forall j \in J$, $\boldsymbol{r_j} := \omega_j / p_j$

Propriété

Pour le problème de minimisation de la somme des dates de fins, on a dominance stricte des ordonnancements  *et* $\searrow^r$

Règle de Smith [Smith,56]

L'ordonnancement $\mathcal{S}$ vérifie $\Leftrightarrow$ il est sans trou et tassé à gauche

L'ordonnancement $\mathcal{S}$ vérifie $\searrow^r \Leftrightarrow$ les tâches sont par ratio décroissant où $\forall j \in J$, $\boldsymbol{r_j} := \omega_j / p_j$

Propriété

Pour le problème de minimisation de la somme des dates de fins, on a dominance stricte des ordonnancements et $\searrow^r$

$\hookrightarrow$algorithme en $\mathrm{O}(n \log n)$

Règle de Smith [Smith,56]

L'ordonnancement $\mathcal{S}$ vérifie  $\Leftrightarrow$ il est sans trou et tassé à gauche

L'ordonnancement $\mathcal{S}$ vérifie $\searrow^r \Leftrightarrow$ les tâches sont par ratio décroissant où $\forall j \in J$, $\boldsymbol{r_j} := \omega_j / p_j$

Propriété

Pour le problème de minimisation de la somme des dates de fins, on a dominance stricte des ordonnancements  *et* $\searrow^r$

$\hookrightarrow$algorithme en $O(n \log n)$

$\hookrightarrow$ce problème $\in$ P

Ensemble des solutions [Queyranne,93]

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

Ensemble des solutions [Queyranne,93]

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

Ensemble des solutions [Queyranne,93]

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

$\boldsymbol{Q} :=$ l'ensemble des vecteurs codant des ordonnancements réalisables

Ensemble des solutions [Queyranne,93]

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

$$\boldsymbol{Q} := \left\{ C \in \mathbb{R}^J \;\middle|\; \begin{array}{l} \forall j \in J, \quad C_j \geqslant p_j \\ \forall (j,k) \in J^2, \quad C_k \geqslant C_j + p_k \text{ ou } C_j \geqslant C_k + p_j \end{array} \right\}$$

Ensemble des solutions [Queyranne,93]

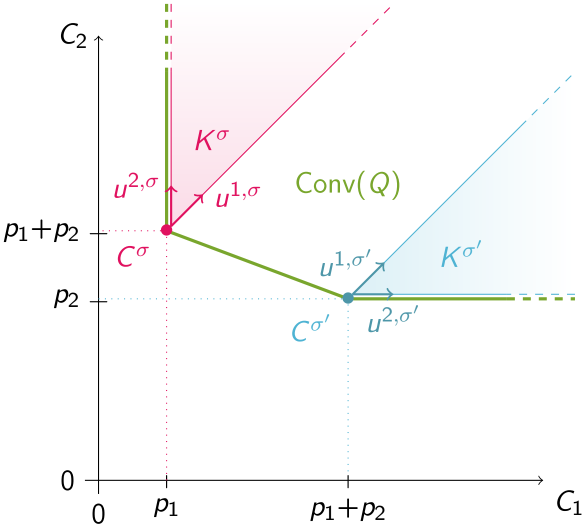
Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- des poids pour chacune de ces tâches $(\omega_j)_{j \in J}$

$$\boldsymbol{Q} := \left\{ C \in \mathbb{R}^J \;\middle|\; \begin{array}{l} \forall j \in J, \quad C_j \geqslant p_j \\ \forall (j,k) \in J^2, \quad C_k \geqslant C_j + p_k \text{ ou } C_j \geqslant C_k + p_j \end{array} \right\}$$

$\hookrightarrow$On va étudier Conv(Q)

Illustration pour $n = 2$ [Queyranne,93]

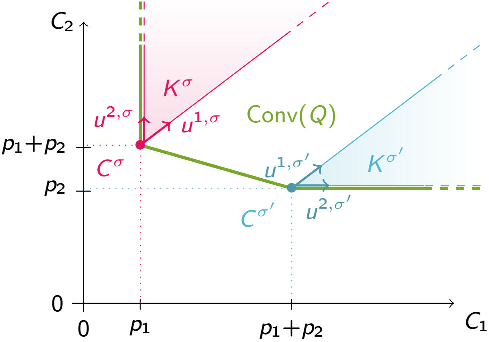


$\sigma = \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 2 \end{cases}$

$\sigma' = \begin{cases} 1 \mapsto 2 \\ 2 \mapsto 1 \end{cases}$

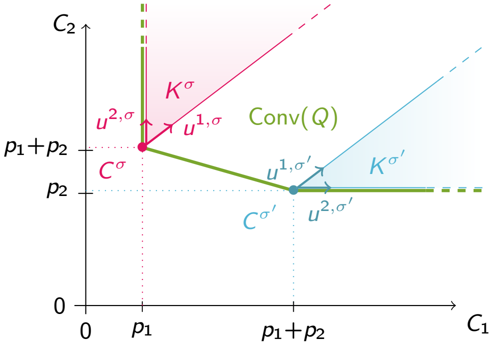
Description géométrique [Queyranne,93]

$$Q = \bigcup_{\sigma \in \mathfrak{S}} K^\sigma$$



Description géométrique [Queyranne,93]

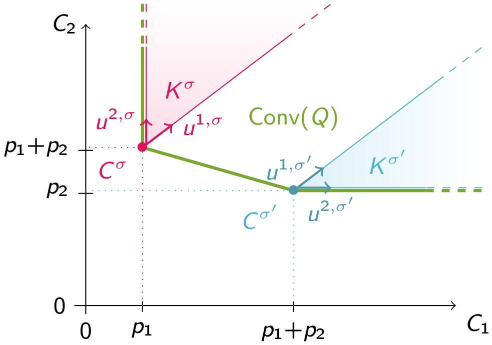
$$Q = \bigcup_{\sigma \in \mathfrak{S}} \mathrm{Conv}(\{C^\sigma\}) + \mathrm{CC}^\circ\left(\{u^{i,\sigma} \mid i \in [1..n]\}\right)$$



Description géométrique [Queyranne,93]

$$Q = \bigcup_{\sigma \in \mathfrak{S}} \mathrm{Conv}(\{C^\sigma\}) + \mathrm{CC}^\circ\left(\{u^{i,\sigma} \mid i \in [1..n]\}\right)$$

$$\hookrightarrow \overline{\mathrm{Conv}(Q)} = \mathrm{Conv}\{C^\sigma \mid \sigma \in \mathfrak{S}_n\} + \mathrm{CC}^\circ\left\{u^{i,\sigma} \mid {}^{i\in[1..n]}_{\sigma\in\mathfrak{S}_n}\right\}$$

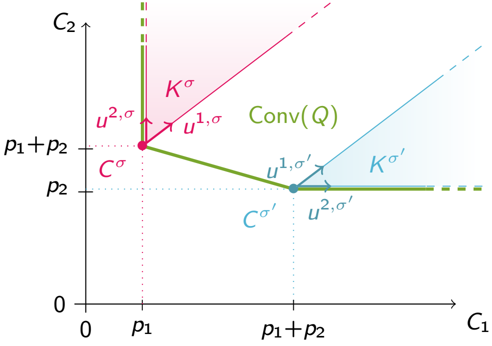


Description géométrique [Queyranne,93]

$$Q = \bigcup_{\sigma \in \mathfrak{S}} \mathrm{Conv}(\{C^\sigma\}) + \mathrm{CC}^\circ\left(\{u^{i,\sigma} \mid i \in [1..n]\}\right)$$

$$\hookrightarrow \overline{\mathrm{Conv}(Q)} = \mathrm{Conv}\{C^\sigma \mid \sigma \in \mathfrak{S}_n\} + \mathrm{CC}^\circ \left\{u^{i,\sigma} \mid {}^{i \in [1..n]}_{\sigma \in \mathfrak{S}_n}\right\}$$

$$\hookrightarrow \boxed{\mathrm{Conv}(Q) = \mathrm{Conv}\left(\{C^\sigma \mid \sigma \in \mathfrak{S}_n\}\right) + (\mathbb{R}^+)^J}$$



Inégalités de Queyranne [Queyranne,93]

Inégalité facette : $\langle \omega \mid C \rangle \geqslant c$ où $\omega \in (\mathbb{R}^+)^J$ et où $c = \min_{C \in Q} \langle \omega \mid C \rangle$

Inégalités de Queyranne [Queyranne,93]

Inégalité facette : $\langle \omega \mid C \rangle \geqslant c$ où $\omega \in (\mathbb{R}^+)^J$ et où $c = \min_{C \in Q} \langle \omega \mid C \rangle$

Poids particuliers : $\boldsymbol{\omega}^{\boldsymbol{S}} := \sum_{j \in S} p_j \mathbb{1}_j$ pour $S \subset J$

Inégalités de Queyranne [Queyranne,93]

Inégalité facette : $\langle \omega \mid C \rangle \geqslant c$ où $\omega \in (\mathbb{R}^+)^J$ et où $c = \min_{C \in Q} \langle \omega \mid C \rangle$

Poids particuliers : $\boldsymbol{\omega^S} := \sum_{j \in S} p_j \mathbb{1}_j$ pour $S \subset J$

$$\boldsymbol{g} := \begin{pmatrix} \mathcal{P}(J) \to \mathbb{R} \\ S \longmapsto \min_{C \in Q} \langle \omega^S \mid C \rangle \end{pmatrix}$$

Inégalités de Queyranne [Queyranne,93]

Inégalité facette : $\langle \omega \mid C \rangle \geqslant c$ où $\omega \in (\mathbb{R}^+)^J$ et où $c = \min_{C \in Q} \langle \omega \mid C \rangle$

Poids particuliers : $\boldsymbol{\omega^S} := \sum_{j \in S} p_j \mathbb{1}_j$ pour $S \subset J$

$$\boldsymbol{g} := \begin{pmatrix} \mathcal{P}(J) \to \mathbb{R} \\ S \longmapsto \min_{C \in Q} p * C(S) \end{pmatrix} \text{ où } p * C(S) = \sum_{i \in S} p_i * C_i$$

Inégalités de Queyranne [Queyranne,93]

Inégalité facette : $\langle \omega \mid C \rangle \geqslant c$ où $\omega \in (\mathbb{R}^+)^J$ et où $c = \min_{C \in Q} \langle \omega \mid C \rangle$

Poids particuliers : $\boldsymbol{\omega^S} := \sum_{j \in S} p_j \mathbb{1}_j$ pour $S \subset J$

$$\boldsymbol{g} := \begin{pmatrix} \mathcal{P}(J) \to \mathbb{R} \\ S \longmapsto \min_{C \in Q} p * C(S) \end{pmatrix} \text{ où } p*C(S) = \sum_{i \in S} p_i * C_i$$

Polyèdre de Queyranne : $\boldsymbol{P^Q} := \left\{ C \in \mathbb{R}^J \;\middle|\; \forall S \subset J,\;\; p*C(S) \geqslant g(S) \right\}$

Inégalités de Queyranne [Queyranne,93]

Inégalité facette : $\langle \omega \mid C \rangle \geqslant c$ où $\omega \in (\mathbb{R}^+)^J$ et où $c = \min_{C \in Q} \langle \omega \mid C \rangle$

Poids particuliers : $\boldsymbol{\omega^S} := \sum_{j \in S} p_j \mathbb{1}_j$ pour $S \subset J$

$$\boldsymbol{g} := \begin{pmatrix} \mathcal{P}(J) \to \mathbb{R} \\ S \longmapsto \min_{C \in Q} p * C(S) \end{pmatrix} \text{ où } p*C(S) = \sum_{i \in S} p_i * C_i$$

Polyèdre de Queyranne : $\boldsymbol{P^Q} := \left\{ C \in \mathbb{R}^J \;\middle|\; \forall S \subset J,\ \ p*C(S) \geqslant g(S) \right\}$

Résultat : $\boxed{P^Q = \mathrm{Conv}(Q)}$

Idée, par contraposée [AEF]

But point extrême $\Rightarrow$ sans-chevauchement

Idée, par contraposée [AEF]

But	point extrême $\Rightarrow$ sans-chevauchement
Preuve par contraposée	"chevauchement" $\Rightarrow$ non extrême

Idée, par contraposée [AEF]

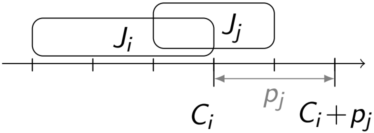
But	point extrême $\Rightarrow$ sans-chevauchement
Preuve par contraposée	"chevauchement" $\Rightarrow$ non extrême
Preuve plus précisément	"chevauchement" $\Rightarrow$ milieu de deux points

Idée, par contraposée [AEF]

But	point extrême $\Rightarrow$ sans-chevauchement
Preuve par contraposée	"chevauchement" $\Rightarrow$ non extrême
Preuve plus précisément	"chevauchement" $\Rightarrow$ milieu de deux points

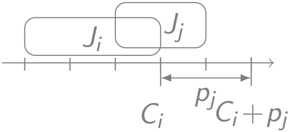
Soit $C \in P^Q$ avec chevauchement : il existe $(i,j) \in J^2$ tels que

$$\boxed{C_i \leqslant C_j < C_i + p_j}$$



On aimerait poser... [AEF]

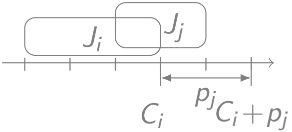
$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C^{+-}} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C^{-+}} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$



Choisir ε tel que $\forall S \subset J,\ \sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C}^{+-} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C}^{-+} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$

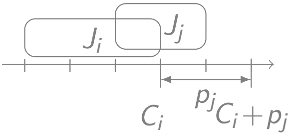


Choisir ε tel que $\forall S \subset J, \ \sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

$\rightarrow$ Si $i \in S$ et $j \in S$

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C}^{+-} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C}^{-+} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$

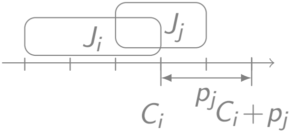


Choisir ε tel que $\forall S \subset J,\ \sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

$\rightarrow$ Si $i \in S$ et $j \in S$ ✓

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C}^{+-} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C}^{-+} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$



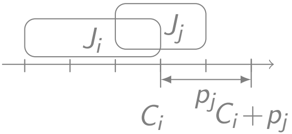
Choisir ε tel que $\forall S \subset J,\ \sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

→ Si $i \in S$ et $j \in S$ ✓

→ Si $i \notin S$ et $j \notin S$ ✓

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C^{+-}} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C^{-+}} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$

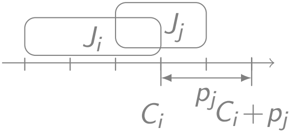


Choisir ε tel que $\forall S \subset J$, $\sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

- $\rightarrow$ Si $i \in S$ et $j \in S$ ✓
- $\rightarrow$ Si $i \notin S$ et $j \notin S$ ✓
- $\rightarrow$ Si $i \notin S$ et $j \in S$

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C^{+-}} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C^{-+}} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$

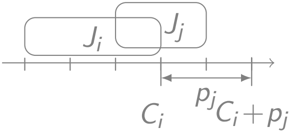


Choisir ε tel que $\forall S \subset J$, $\sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

- $\rightarrow$ Si $i \in S$ et $j \in S$ ✓
- $\rightarrow$ Si $i \notin S$ et $j \notin S$ ✓
- $\rightarrow$ Si $i \notin S$ et $j \in S$
  $\hookrightarrow$ $\boldsymbol{\varepsilon_1} := \min \left\{ p * C(S^1) - g(S^1) \mid S \in \mathcal{P}^*(J), i \notin S^1, j \in S^1 \right\}$

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \text{ où } \begin{array}{l} \boldsymbol{C}^{+-} := C + \frac{\varepsilon}{p_i}\mathbb{1}_i - \frac{\varepsilon}{p_j}\mathbb{1}_j \\ \boldsymbol{C}^{-+} := C - \frac{\varepsilon}{p_i}\mathbb{1}_i + \frac{\varepsilon}{p_j}\mathbb{1}_j \end{array}$$



Choisir ε tel que $\forall S \subset J$, $\sum_{k \in S} p_k C_k^{+-} \geqslant g(S)$ et $\sum_{k \in S} p_k C_k^{-+} \geqslant g(S)$

$\rightarrow$ Si $i \in S$ et $j \in S$ ✓

$\rightarrow$ Si $i \notin S$ et $j \notin S$ ✓

$\rightarrow$ Si $i \notin S$ et $j \in S$
$\hookrightarrow \boldsymbol{\varepsilon_1} := \min \left\{ p * C(S^1) - g(S^1) \mid S \in \mathcal{P}^*(J), i \notin S^1, j \in S^1 \right\}$

$\rightarrow$ Si $i \in S$ et $j \notin S$
$\hookrightarrow \boldsymbol{\varepsilon_2} := \min \left\{ p * C(S^2) - g(S^2) \mid S \in \mathcal{P}^*(J), i \in S^2, j \notin S^2 \right\}$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{array}{c} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p * C(S^1) > g(S^1)$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p * C(S^1) > g(S^1)$

Par l'absurde s'il existe S^1 avec $\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

On note $\tilde{S}^1 = S^1_{\setminus\{j\}}$. $\quad p * C(S^1) = p * C(\tilde{S}^1) + p_j C_j$

$$g(S^1) \;\; = g(\tilde{S}^1) + p_j p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

$$\textbf{\textit{Si}}\ C_i \leqslant C_j,\ \textbf{\textit{alors}}\ \forall S^1 \in \mathcal{P}(J),\ \left.\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p * C(S^1) > g(S^1)$$

Par l'absurde s'il existe S^1 avec $\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

On note $\tilde{S^1} = S^1_{\setminus\{j\}}$. $\quad p * C(S^1) = p * C(\tilde{S^1}) + p_j C_j$

$$\begin{array}{rl} p * C(S^1) & = p * C(\tilde{S^1}) + p_j C_j \\ \| & \\ g(S^1) & = g(\tilde{S^1}) + p_j p(S^1) \end{array}$$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p * C(S^1) > g(S^1)$

Par l'absurde s'il existe S^1 avec $\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

On note $\tilde{S}^1 = S^1_{\setminus\{j\}}$.

$$\begin{array}{ccc} p * C(S^1) & = & p * C(\tilde{S}^1) + p_j C_j \\ \shortparallel & & \rotatebox{-90}{$\geqslant$} \\ g(S^1) & = & g(\tilde{S}^1) + p_j p(S^1) \end{array}$$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J)$, $\left.\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p * C(S^1) > g(S^1)$

Par l'absurde s'il existe S^1 avec $\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

On note $\tilde{S}^1 = S^1_{\setminus\{j\}}$. $\quad \begin{array}{ccl} p * C(S^1) & = & p * C(\tilde{S}^1) + p_j C_j \\ \| & & \rotatebox{-90}{$\geqslant$} \\ g(S^1) & = & g(\tilde{S}^1) + p_j p(S^1) \end{array}$ donc $\boxed{C_j \leqslant p(S^1)}$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J), \left.\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p{*}C(S^1) > g(S^1)$

$p{*}C(S^1 \cup \{i\}) = p{*}C(S^1) + p_i C_i$

$\boxed{p{*}C(S^1) = g(S^1)}$

$\boxed{C_j \leqslant p(S^1)}$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{array}{r} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p{*}C(S^1) > g(S^1)$

$$\begin{aligned} p{*}C(S^1 \cup \{i\}) &= p{*}C(S^1) + p_i C_i \\ &= g(S^1) + p_i C_i \end{aligned}$$

$$\boxed{p{*}C(S^1) = g(S^1)}$$

$$\boxed{C_j \leqslant p(S^1)}$$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J), \left.\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}\right\} \Rightarrow p * C(S^1) > g(S^1)$

$$\begin{aligned} p * C(S^1 \cup \{i\}) &= p * C(S^1) + p_i C_i \\ &= g(S^1) + p_i C_i \\ &\leqslant g(S^1) + p_i C_j \end{aligned}$$

$\boxed{p * C(S^1) = g(S^1)}$

$\boxed{C_j \leqslant p(S^1)}$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J), \left.\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}\right\} \Rightarrow p * C(S^1) > g(S^1)$

$$\begin{aligned} p * C(S^1 \cup \{i\}) &= p * C(S^1) + p_i C_i \\ &= g(S^1) + p_i C_i \\ &\leqslant g(S^1) + p_i C_j \\ &\leqslant g(S^1) + p_i p(S^1) \end{aligned}$$

$$\boxed{p * C(S^1) = g(S^1)}$$

$$\boxed{C_j \leqslant p(S^1)}$$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}\right\} \Rightarrow p*C(S^1) > g(S^1)$

$$\begin{aligned} p*C(S^1 \cup \{i\}) &= p*C(S^1) + p_i C_i \\ &= g(S^1) + p_i C_i \\ &\leqslant g(S^1) + p_i C_j \\ &\leqslant g(S^1) + p_i p(S^1) \\ &< g(S^1) + p_i[p(S^1) + p_i] \end{aligned}$$

$\boxed{p*C(S^1) = g(S^1)}$

$\boxed{C_j \leqslant p(S^1)}$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}\right\} \Rightarrow p * C(S^1) > g(S^1)$

$$\begin{aligned} p * C(S^1 \cup \{i\}) &= p * C(S^1) + p_i C_i \\ &= g(S^1) + p_i C_i \\ &\leqslant g(S^1) + p_i C_j \\ &\leqslant g(S^1) + p_i p(S^1) \\ &< g(S^1) + p_i[p(S^1) + p_i] \\ &= g(S^1 \cup \{i\}) \end{aligned}$$

$\boxed{p * C(S^1) = g(S^1)}$

$\boxed{C_j \leqslant p(S^1)}$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leqslant C_j$, ***alors*** $\forall S^1 \in \mathcal{P}(J),\ \left.\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}\right\} \Rightarrow p{*}C(S^1) > g(S^1)$

$$\begin{aligned} p{*}C(S^1 \cup \{i\}) &= p{*}C(S^1) + p_i C_i \\ &= g(S^1) + p_i C_i \\ &\leqslant g(S^1) + p_i C_j \\ &\leqslant g(S^1) + p_i p(S^1) \\ &< g(S^1) + p_i[p(S^1) + p_i] \\ &= g(S^1 \cup \{i\}) \\ &\leqslant p{*}C(S^1 \cup \{i\}) \end{aligned}$$

$\boxed{p{*}C(S^1) = g(S^1)}$

$\boxed{C_j \leqslant p(S^1)}$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J), \left.\begin{matrix} i \in S^2 \\ j \notin S^2 \end{matrix}\right\} \Rightarrow p * C(S^2) > g(S^2)$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J),\ \left.\begin{matrix} i \in S^2 \\ j \notin S^2 \end{matrix}\right\} \Rightarrow p * C(S^2) > g(S^2)$

Par l'absurde s'il existe S^2 avec $\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

$$\textbf{\textit{Si}}\ C_j < C_i + p_j,\ \textbf{\textit{alors}}\ \forall S^2 \in \mathcal{P}(J),\ \left.\begin{matrix} i \in S^2 \\ j \notin S^2 \end{matrix}\right\} \Rightarrow p * C(S^2) > g(S^2)$$

Par l'absurde s'il existe S^2 avec $\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J), \left.\begin{array}{c} i \in S^2 \\ j \notin S^2 \end{array}\right\} \Rightarrow p * C(S^2) > g(S^2)$

Par l'absurde s'il existe S^2 avec $\begin{array}{c} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

$p * C(S^2 \cup \{j\}) = p * C(S^2) + p_j C_j$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

$$\textbf{\textit{Si}}\ C_j < C_i + p_j,\ \textbf{\textit{alors}}\ \forall S^2 \in \mathcal{P}(J),\ \left.\begin{matrix} i \in S^2 \\ j \notin S^2 \end{matrix}\right\} \Rightarrow p * C(S^2) > g(S^2)$$

Par l'absurde s'il existe S^2 avec $\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

$$\begin{aligned} p * C(S^2 \cup \{j\}) &= p * C(S^2) + p_j C_j \\ &= g(S^2) + p_j C_j \end{aligned}$$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J), \left.\begin{array}{l} i \in S^2 \\ j \notin S^2 \end{array}\right\} \Rightarrow p * C(S^2) > g(S^2)$

Par l'absurde s'il existe S^2 avec $\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

$$\begin{aligned} p * C(S^2 \cup \{j\}) &= p * C(S^2) + p_j C_j \\ &= g(S^2) + p_j C_j \\ &< g(S^2) + p_j [C_i + p_j] \end{aligned}$$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J),\ \left.\begin{array}{r} i \in S^2 \\ j \notin S^2 \end{array}\right\} \Rightarrow p * C(S^2) > g(S^2)$

Par l'absurde s'il existe S^2 avec $\begin{array}{c} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p * C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

$$\begin{aligned} p * C(S^2 \cup \{j\}) &= p * C(S^2) + p_j C_j \\ &= g(S^2) + p_j C_j \\ &< g(S^2) + p_j [C_i + p_j] \\ &\leqslant g(S^2) + p_j [p(S^2) + p_j] \end{aligned}$$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J),\ \left.\begin{array}{l} i \in S^2 \\ j \notin S^2 \end{array}\right\} \Rightarrow p*C(S^2) > g(S^2)$

Par l'absurde s'il existe S^2 avec $\begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array}$ tel que $\boxed{p*C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

$$\begin{aligned} p*C(S^2 \cup \{j\}) &= p*C(S^2) + p_j C_j \\ &= g(S^2) + p_j C_j \\ &< g(S^2) + p_j[C_i + p_j] \\ &\leqslant g(S^2) + p_j[p(S^2) + p_j] \\ &= g(S^2 \cup \{j\}) \end{aligned}$$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

Si $C_j < C_i + p_j$, ***alors*** $\forall S^2 \in \mathcal{P}(J),\ \left.\begin{matrix} i \in S^2 \\ j \notin S^2 \end{matrix}\right\} \Rightarrow p*C(S^2) > g(S^2)$

Par l'absurde s'il existe S^2 avec $\begin{matrix} i \notin S^1 \\ j \in S^1 \end{matrix}$ tel que $\boxed{p*C(S^1) = g(S^1)}$

De même on a $\boxed{C_i \leqslant p(S^2)}$

$$\begin{aligned} p*C(S^2 \cup \{j\}) &= p*C(S^2) + p_j C_j \\ &= g(S^2) + p_j C_j \\ &< g(S^2) + p_j[C_i + p_j] \\ &\leqslant g(S^2) + p_j[p(S^2) + p_j] \\ &= g(S^2 \cup \{j\}) \\ &\leqslant p*C(S^2 \cup \{j\}) \end{aligned}$$

Le problème

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- une date d'échéance commune et non restrictive $d \geqslant \sum p_j$
- des coûts de retard pour chacune de ces tâches $(\alpha_j)_{j \in J}$
- des coûts d'avance pour chacune de ces tâches $(\beta_j)_{j \in J}$

Le problème

Une instance =

- un ensemble de tâches J
- les durées de ces tâches $(p_j)_{j \in J}$
- une date d'échéance commune et non restrictive $d \geqslant \sum p_j$
- des coûts de retard pour chacune de ces tâches $(\alpha_j)_{j \in J}$
- des coûts d'avance pour chacune de ces tâches $(\beta_j)_{j \in J}$

Idée [AEF]

Utiliser les variables e et t pour pouvoir exprimer le coût linéairement.

Idée [AEF]

Utiliser les variables e et t pour pouvoir exprimer le coût linéairement.

MAIS il faut assurer :

[**cohérence**] Avec e_i et t_i on peut déterminer la période d'exécution de J_i

Idée [AEF]

Utiliser les variables e et t pour pouvoir exprimer le coût linéairement.

MAIS il faut assurer :

[**cohérence**] Avec e_i et t_i on peut déterminer la période d'exécution de J_i

[**non-chevauchement**] Les périodes d'exécution sont deux à deux disjointes.

Idée [AEF]

Utiliser les variables e et t pour pouvoir exprimer le coût linéairement.

MAIS il faut assurer :

[**cohérence**] Avec e_i et t_i on peut déterminer la période d'exécution de J_i

[**non-chevauchement**] Les périodes d'exécution sont deux à deux disjointes.

[**positivité**] Les périodes d'exécution ne commencent pas avant le temps 0.

Assurer la cohérence [AEF]

- Introduire des variables disjonctives $\rightarrow \delta_j$

$$\forall j \in J,\ e_j \geqslant 0 \quad (e.1)$$
$$e_j \leqslant M\delta_j \quad (e.2)$$

$$\forall j \in J,\ t_j \geqslant 0 \quad (t.1)$$
$$t_j \leqslant M(1-\delta_j) \quad (t.2)$$

Assurer la cohérence [AEF]

- Introduire des variables disjonctives $\rightarrow \delta_j$

$$\forall j \in J,\ e_j \geqslant 0 \quad (e.1) \qquad e_j \leqslant M\delta_j \quad (e.2)$$

$$\forall j \in J,\ t_j \geqslant 0 \quad (t.1) \qquad t_j \leqslant M(1-\delta_j) \quad (t.2)$$

- Pour M assez grand, ces contraintes assurent la **cohérence**

Assurer la cohérence [AEF]

- Introduire des variables disjonctives $\rightarrow \delta_j$

$$\forall j \in J,\ e_j \geqslant 0 \quad (e.1) \qquad e_j \leqslant M\delta_j \quad (e.2)$$

$$\forall j \in J,\ t_j \geqslant 0 \quad (t.1) \qquad t_j \leqslant M(1-\delta_j) \quad (t.2)$$

- Pour M assez grand, ces contraintes assurent la **cohérence**
- Avec $\boldsymbol{M} := \sum_{j \in J} p_j$ on a aussi la **positivité**

Idée pour assurer le non-chevauchement [AEF]

- Pour un ordonnancement 🕛 :

non-chevauchement =

Idée pour assurer le non-chevauchement [AEF]

- Pour un ordonnancement 🕛 :

non-chevauchement =	tâches en avance à gauche de d	et	tâches en retard à droite de d

Idée pour assurer le non-chevauchement [AEF]

- Pour un ordonnancement 🕛 :

non-chevauchement =	tâches en avance à gauche de *d*	et	tâches en retard à droite de *d*
	non-chevauchement côté avance	et	non-chevauchement côté retard

Idée pour assurer le non-chevauchement [AEF]

- Pour un ordonnancement :

non-chevauchement =	tâches en avance à gauche de d	et	tâches en retard à droite de d
	non-chevauchement côté avance	et	non-chevauchement côté retard

$\hookrightarrow$ Utiliser les inégalités de Queyranne, de part et d'autre de la due-date :

$$\begin{cases} \forall S \subset J,\ p * t(S) \geqslant g(S) \\ \end{cases}$$

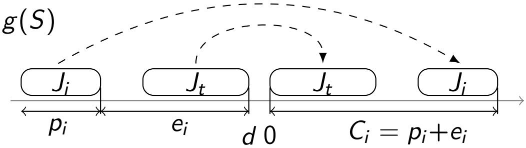
Idée pour assurer le non-chevauchement [AEF]

- Pour un ordonnancement :

non-chevauchement =	tâches en avance à gauche de d	et	tâches en retard à droite de d
	non-chevauchement côté avance	et	non-chevauchement côté retard

$\hookrightarrow$ Utiliser les inégalités de Queyranne, de part et d'autre de la due-date :

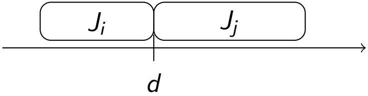
$$\begin{cases} \forall S \subset J,\ p * t(S) \geqslant g(S) \\ \forall S \subset J,\ p * [p+e](S) \geqslant g(S) \end{cases}$$



Ces inégalités sont-elles valides ? [AEF]

Pour $S = \{i, j\}$ où $t_i = 0$ l'inégalité s'écrit : $$\begin{aligned} p_j t_j &\geqslant \frac{1}{2}[(p_i + p_j)^2 + (p_i^2 + p_j^2)] \\ &= p_i^2 + p_j^2 + p_i p_j \\ &> p_j^2 \end{aligned}$$

Inégalité non valide pour



$\hookrightarrow$ On pose $\boldsymbol{E'} := \{ j \in J \mid \delta_j = 1 \}$ et $\boldsymbol{T} := \{ j \in J \mid \delta_j = 0 \}$ pour écrire :

$$\begin{cases} \forall S \subset J,\ p * [p + e](S \cap E') \geqslant g(S \cap E') & \text{(S1')} \\ \forall S \subset J,\ p * t(S \cap T) \geqslant g(S \cap T) & \text{(S2')} \end{cases}$$

Éliminer les intersections [AEF]

$$
\begin{aligned}
\text{(S1')} \Leftrightarrow \forall S \subset J,\ & \sum_{j\in J} p_j[p_j+e_j]\delta_j \geqslant \frac{1}{2}\left(\Big(\sum_{j\in J} p_j\delta_j\Big)^2 + \sum_{j\in J} p_j^2\delta_j\right)\\
\Leftrightarrow \forall S \subset J,\ & \sum_{j\in J} p_j^2\delta_j + \sum_{j\in J} p_j \underbrace{e_j\delta_j}_{=e_j} \geqslant \frac{1}{2}\sum_{(i,j)\in J^2} p_ip_j\delta_i\delta_j \ +\ \frac{1}{2}\sum_{j\in J} p_j^2\delta_j\\
\Leftrightarrow \forall S \subset J, & \cancel{\sum_{j\in J} p_j^2\delta_j} + \sum_{j\in J} p_je_j \geqslant \frac{1}{2}\sum_{\substack{(i,j)\in J^2\\ i\neq j}} p_ip_j\delta_i\delta_j + \cancel{\frac{1}{2}\sum_{j\in J} p_j^2 \underbrace{\delta_j^2}_{=\delta_j}} + \cancel{\frac{1}{2}\sum_{j\in J} p_j^2\delta_j}\\
\Leftrightarrow \forall S \subset J,\ & \sum_{j\in J} p_je_j \geqslant \sum_{\substack{(i,j)\in J^2\\ i<j}} p_ip_j\delta_i\delta_j
\end{aligned}
$$

Éliminer le terme quadratique [AEF]

On introduit des variables $l_{i,j}$ vérifiant

$$\begin{array}{ll} \forall (i,j) \in J^<,\ l_{i,j} \geqslant 0 & (l.1) \\ \quad l_{i,j} \leqslant \delta_i & (l.2) \\ \quad l_{i,j} \leqslant \delta_j & (l.3) \\ \quad l_{i,j} \geqslant \delta_i + \delta_j - 1 & (l.4) \end{array}$$

$$\forall (i,j) \in J^<,\ \delta_i \delta_j = 1 \Leftrightarrow \begin{cases} \delta_i = 1 \\ \delta_j = 1 \end{cases}$$

Éliminer le terme quadratique [AEF]

On introduit des variables $l_{i,j}$ vérifiant

$$\forall (i,j) \in J^<,\ \begin{array}{ll} l_{i,j} \geqslant 0 & (l.1) \\ l_{i,j} \leqslant \delta_i & (l.2) \\ l_{i,j} \leqslant \delta_j & (l.3) \\ l_{i,j} \geqslant \delta_i + \delta_j - 1 & (l.4) \end{array}$$

$$\forall (i,j) \in J^<,\ \delta_i \delta_j = 1 \Leftrightarrow \begin{cases} \delta_i = 1 \\ \delta_j = 1 \end{cases} \Rightarrow l_{i,j} = 1$$

Éliminer le terme quadratique [AEF]

On introduit des variables $l_{i,j}$ vérifiant

$$\forall (i,j) \in J^<,\ \begin{array}{ll} l_{i,j} \geqslant 0 & (l.1) \\ l_{i,j} \leqslant \delta_i & (l.2) \\ l_{i,j} \leqslant \delta_j & (l.3) \\ l_{i,j} \geqslant \delta_i + \delta_j - 1 & (l.4) \end{array}$$

$$\forall (i,j) \in J^<,\ \delta_i \delta_j = 1 \Leftrightarrow \begin{cases} \delta_i = 1 \\ \delta_j = 1 \end{cases} \Leftrightarrow l_{i,j} = 1$$

Éliminer le terme quadratique [AEF]

On introduit des variables $l_{i,j}$ vérifiant

$$\forall (i,j) \in J^<,\ \begin{array}{ll} l_{i,j} \geqslant 0 & (l.1) \\ l_{i,j} \leqslant \delta_i & (l.2) \\ l_{i,j} \leqslant \delta_j & (l.3) \\ l_{i,j} \geqslant \delta_i + \delta_j - 1 & (l.4) \end{array}$$

$$\forall (i,j) \in J^<,\ \delta_i \delta_j = 1 \Leftrightarrow \begin{cases} \delta_i = 1 \\ \delta_j = 1 \end{cases} \Leftrightarrow l_{i,j} = 1$$

On introduit des variables $r_{i,j}$ vérifiant

$$\forall (i,j) \in J^<,\ \begin{array}{ll} r_{i,j} \geqslant 0 & (r.1) \\ r_{i,j} \leqslant (1-\delta_i) & (r.2) \\ r_{i,j} \leqslant (1-\delta_j) & (r.3) \\ r_{i,j} \geqslant 1 - \delta_i - \delta_j & (r.4) \end{array}$$

$$\forall (i,j) \in J^<,\ (1-\delta_i)(1-\delta_j) = 1 \Leftrightarrow \begin{cases} \delta_i = 0 \\ \delta_j = 0 \end{cases} \Leftrightarrow r_{i,j} = 1$$

Formulation [AEF]

$$\boldsymbol{P}^{\text{"e,t,}\boldsymbol{\delta}\text{,l,r"}} := \Big\{(e, t, \delta, l, r) \in \mathbb{R}^J \times \mathbb{R}^J \times \mathbb{R}^J \times \mathbb{R}^{J^<} \times \mathbb{R}^{J^<} \ \Big|$$

$$\forall j \in J,\ \begin{aligned} e_j &\geqslant 0 && (e.1) \\ e_j &\leqslant M\delta_j && (e.2) \end{aligned} \qquad \forall j \in J,\ \begin{aligned} t_j &\geqslant 0 && (t.1) \\ t_j &\leqslant M(1-\delta_j) && (t.2) \end{aligned}$$

$$\forall j \in J, 0 \leqslant \delta_j \leqslant 1 \quad (\delta)$$

$$\forall (i,j) \in J^<,\ \begin{aligned} l_{i,j} &\geqslant 0 && (l.1) \\ l_{i,j} &\leqslant \delta_i && (l.2) \\ l_{i,j} &\leqslant \delta_j && (l.3) \\ l_{i,j} &\geqslant \delta_i + \delta_j - 1 && (l.4) \end{aligned} \qquad \forall (i,j) \in J^<,\ \begin{aligned} r_{i,j} &\geqslant 0 && (r.1) \\ r_{i,j} &\leqslant (1-\delta_i) && (r.2) \\ r_{i,j} &\leqslant (1-\delta_j) && (r.3) \\ r_{i,j} &\geqslant 1 - \delta_i - \delta_j && (r.4) \end{aligned}$$

$$\forall S \in \mathcal{P}^*(J),\ \begin{aligned} \sum_{i \in S} p_i e_i &\geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} && (\mathrm{S1}) \\ \sum_{i \in S} p_i t_i &\geqslant \sum_{(i,j) \in S^<} p_i p_j r_{i,j} + \sum_{i \in S} p_i^2 (1-\delta_i) && (\mathrm{S2}) \end{aligned}$$

$$\Big\}$$

Extr* [AEF]

Définition

$$\textbf{\textit{Extr}}^* := \left\{ x^* \in Extr(P^{\text{"}e,t,\delta,l,r\text{"}}) \;\middle|\; \begin{array}{l} x^* = (e,t,\delta,l,r) \text{ avec } (\delta,l,r) \in \{0,1\}^{n^2} \\ \text{il existe des coûts } (\alpha,\beta) \in (\mathbb{R}^{+*})^{2n} \text{ tq} \\ x^* \text{ minimise } \langle\alpha|e\rangle + \langle\beta|t\rangle \text{ sur } P^{\text{"}e,t,\delta,l,r\text{"}} \end{array} \right\}$$

Extr* [AEF]

Définition

$$\textbf{\textit{Extr}}^* := \left\{ x^* \in Extr\left(P^{"e,t,\delta,l,r"}\right) \;\middle|\; \begin{array}{l} x^* = (e,t,\delta,l,r) \text{ avec } (\delta,l,r) \in \{0,1\}^{n^2} \\ \text{il existe des coûts } (\alpha,\beta) \in (\mathbb{R}^{+*})^{2n} \text{ tq} \\ x^* \text{ minimise } \langle \alpha | e \rangle + \langle \beta | t \rangle \text{ sur } P^{"e,t,\delta,l,r"} \end{array} \right\}$$

Propriété

Si *$x \in Extr^*$* ***alors*** *x code un ordonnancement réalisable et .*

Extr* [AEF]

Définition

$$\textbf{\textit{Extr}}^* := \left\{ x^* \in Extr\left(P^{"e,t,\delta,l,r"}\right) \;\middle|\; \begin{array}{l} x^* = (e,t,\delta,l,r) \text{ avec } (\delta,l,r) \in \{0,1\}^{n^2} \\ \text{il existe des coûts } (\alpha,\beta) \in (\mathbb{R}^{+*})^{2n} \text{ tq} \\ x^* \text{ minimise } \langle \alpha | e \rangle + \langle \beta | t \rangle \text{ sur } P^{"e,t,\delta,l,r"} \end{array} \right\}$$

Propriété

Si $x \in Extr^*$ ***alors*** x *code un ordonnancement réalisable et .*

Propriété

Réciproquement *un ordonnancement réalisable et est codé par un vecteur de* $Extr^*$.

Inégalités à séparer

$$\forall S \in \mathcal{P}^*(J),\ \sum_{i\in S} p_i e_i \geqslant \sum_{(i,j)\in S^<} p_i p_j l_{i,j} \quad \text{(S1)}$$

$$\sum_{i\in S} p_i t_i \geqslant \sum_{(i,j)\in S^<} p_i p_j r_{i,j} + \sum_{i\in S} p_i^2 (1-\delta_i) \quad \text{(S2)}$$

deux familles d'inégalités en nombre exponentiel
$\hookrightarrow$ deux problèmes de séparation parallèles

Inégalités à séparer

$$\forall S \in \mathcal{P}^*(J),\ \sum_{i\in S} p_i e_i \geqslant \sum_{(i,j)\in S^<} p_i p_j l_{i,j} \qquad \text{(S1)}$$

$$\sum_{i\in S} p_i t_i \geqslant \sum_{(i,j)\in S^<} p_i p_j r_{i,j} + \sum_{i\in S} p_i^2(1-\delta_i) \qquad \text{(S2)}$$

deux familles d'inégalités en nombre exponentiel
$\hookrightarrow$ deux problèmes de séparation parallèles

$$\forall S \subset J, \sum_{i\in S} p_i e_i \geqslant \sum_{(i,j)\in S^<} p_i p_j l_{i,j} \Leftrightarrow \forall S \subset J, 0 \geqslant \sum_{(i,j)\in S^<} p_i p_j l_{i,j} - \sum_{i\in S} p_i e_i$$

Inégalités à séparer

$$\forall S \in \mathcal{P}^*(J),\ \sum_{i\in S} p_i e_i \geqslant \sum_{(i,j)\in S^<} p_i p_j l_{i,j} \qquad \text{(S1)}$$

$$\sum_{i\in S} p_i t_i \geqslant \sum_{(i,j)\in S^<} p_i p_j r_{i,j} + \sum_{i\in S} p_i^2(1-\delta_i) \qquad \text{(S2)}$$

deux familles d'inégalités en nombre exponentiel
$\hookrightarrow$ deux problèmes de séparation parallèles

$$\forall S \subset J, \sum_{i\in S} p_i e_i \geqslant \sum_{(i,j)\in S^<} p_i p_j l_{i,j} \Leftrightarrow \forall S \subset J, 0 \geqslant \underbrace{\sum_{(i,j)\in S^<} p_i p_j l_{i,j} - \sum_{i\in S} p_i e_i}_{:=\Gamma_1(S)}$$

Inégalités à séparer

$$\forall S \in \mathcal{P}^*(J),\ \sum_{i \in S} p_i e_i \geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} \qquad \text{(S1)}$$

$$\sum_{i \in S} p_i t_i \geqslant \sum_{(i,j) \in S^<} p_i p_j r_{i,j} + \sum_{i \in S} p_i^2 (1 - \delta_i) \qquad \text{(S2)}$$

deux familles d'inégalités en nombre exponentiel
$\hookrightarrow$ deux problèmes de séparation parallèles

$$\forall S \subset J, \sum_{i \in S} p_i e_i \geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} \Leftrightarrow \forall S \subset J, 0 \geqslant \underbrace{\sum_{(i,j) \in S^<} p_i p_j l_{i,j} - \sum_{i \in S} p_i e_i}_{:= \Gamma_1(S)}$$

$$\Leftrightarrow \max_{S \subset J} \Gamma_1(S) \leqslant 0$$

Inégalités à séparer

$$\forall S \in \mathcal{P}^*(J),\ \sum_{i \in S} p_i e_i \geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} \qquad \text{(S1)}$$

$$\sum_{i \in S} p_i t_i \geqslant \sum_{(i,j) \in S^<} p_i p_j r_{i,j} + \sum_{i \in S} p_i^2 (1 - \delta_i) \qquad \text{(S2)}$$

deux familles d'inégalités en nombre exponentiel
$\hookrightarrow$ deux problèmes de séparation parallèles

$$\forall S \subset J, \sum_{i \in S} p_i e_i \geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} \Leftrightarrow \forall S \subset J, 0 \geqslant \underbrace{\sum_{(i,j) \in S^<} p_i p_j l_{i,j} - \sum_{i \in S} p_i e_i}_{:= \Gamma_1(S)}$$

$$\Leftrightarrow \max_{S \subset J} \Gamma_1(S) \leqslant 0$$

$\hookrightarrow$ $\boxed{\text{séparation } 1 = \text{maximisation de } \Gamma_1}$

Inégalités à séparer

$$\forall S \in \mathcal{P}^*(J),\ \sum_{i \in S} p_i e_i \geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} \qquad \text{(S1)}$$

$$\sum_{i \in S} p_i t_i \geqslant \sum_{(i,j) \in S^<} p_i p_j r_{i,j} + \sum_{i \in S} p_i^2 (1 - \delta_i) \qquad \text{(S2)}$$

deux familles d'inégalités en nombre exponentiel
$\hookrightarrow$ deux problèmes de séparation parallèles

$$\forall S \subset J, \sum_{i \in S} p_i e_i \geqslant \sum_{(i,j) \in S^<} p_i p_j l_{i,j} \Leftrightarrow \forall S \subset J, 0 \geqslant \underbrace{\sum_{(i,j) \in S^<} p_i p_j l_{i,j} - \sum_{i \in S} p_i e_i}_{:= \Gamma_1(S)}$$

$$\Leftrightarrow \max_{S \subset J} \Gamma_1(S) \leqslant 0$$

séparation 2 = maximisation de Γ_2

$\hookrightarrow$ $\boxed{\text{séparation 1 = maximisation de } \Gamma_1}$

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

$\hookrightarrow$ algorithme de séparation polynomial

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

$\hookrightarrow$ algorithme de séparation polynomial

$\hookrightarrow$ algorithme de coupes polynomial

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

$\hookrightarrow$ algorithme de séparation polynomial

$\hookrightarrow$ algorithme de coupes polynomial

$\hookrightarrow$ difficulté du problème due aux δ entiers

De la théorie à la pratique

- Formulation mixte $\hookrightarrow$ algorithme de branchement

De la théorie à la pratique

- Formulation mixte $\hookrightarrow$ algorithme de branchement
- Notre formulation n'est pas compacte $\hookrightarrow$ algorithme de coupe

De la théorie à la pratique

- Formulation mixte $\hookrightarrow$ algorithme de branchement
- Notre formulation n'est pas compacte $\hookrightarrow$ algorithme de coupe
- En dur les inégalités (e,-), (t,-),(δ,-),(l,-),(r,-) et (S2) avec S singleton

De la théorie à la pratique

- Formulation mixte $\hookrightarrow$ algorithme de branchement
- Notre formulation n'est pas compacte $\hookrightarrow$ algorithme de coupe
- En dur les inégalités (e,-), (t,-),(δ,-),(l,-),(r,-) et (S2) avec S singleton
- À séparer les inégalités (S1) et (S2) restantes

De la théorie à la pratique

- Formulation mixte $\hookrightarrow$ algorithme de branchement
- Notre formulation n'est pas compacte $\hookrightarrow$ algorithme de coupe
- En dur les inégalités (e,-), (t,-),(δ,-),(l,-),(r,-) et (S2) avec S singleton
- À séparer les inégalités (S1) et (S2) restantes

- Algorithme global codé avec Cplex
- Problème de séparation ramené à MAX-CUT dans un graphe complet
  $\hookrightarrow$utilisation de Cplex à un autre niveau

Le benchmark [Biskup & Feldmann,01]

on fixe n
on fait varier k $\xrightarrow{\text{schéma de génération}}$ $\begin{pmatrix} p_j \\ \alpha_j \\ \beta_j \end{pmatrix}$ pour chaque k

$\hookrightarrow$ instances non-restrictives disponibles en ligne
http://people.brunel.ac.uk/~mastjjb/jeb/orlib/schinfo.html

Pour la due-date on pose $d = h * \sum_{j \in J} p_j$ pour divers $h \in]0, 1]$

Tests de leurs heuristiques.

Leurs résultats pour $n = 10$

Table 3
The upper bounds and optimal objective function values for the 10 job examples

10 jobs ($n = 10$)	$\sum_{i=1}^{10} p_i$	$h = 0.2$	$h = 0.4$	$h = 0.6$	$h = 0.8$
$k = 1$	116	2009[I] [1936]	1057[I/II] [1025]	841*[I/II]	818*[I/II]
$k = 2$	129	1125[II] [1042]	615*[I]	615*[I]	615*[I]
$k = 3$	125	1731[I/II] [1586]	931[I] [917]	793*[I/II]	793*[I/II]
$k = 4$	102	2392[I] [2139]	1251[II] [1230]	815*[I/II]	815[I/II] [803]
$k = 5$	94	1220[I] [1187]	661[I] [630]	521*[I/II]	521*[I/II]
$k = 6$	88	1623[I] [1521]	908*[II]	755*[I/II]	755*[I/II]
$k = 7$	103	2269[I] [2170]	1374*[I]	1102[I/II] [1101]	1083*[I/II]
$k = 8$	79	1774[II] [1720]	1104[II] [1020]	610*[I]	540*[I]
$k = 9$	92	1792[I] [1574]	876*[I]	582*[I]	554*[I/II]
$k = 10$	127	1934[II] [1869]	1173[I] [1136]	711[I/II] [710]	671*[I/II]

Nos résultats pour $n = 10$, avec séparation exacte

k	p(J)	valeur	temps en sec	tps sépa exacte	nb sépa exacte	nb inég ajoutées	nb sol entières	nb noeud
1	116	818	0,56	0,52	48(3)	24	3(0)	12
2	129	615	0,23	0,20	18(3)	12	1(0)	3
3	125	793	0,67	0,61	54(3)	25	5(0)	16
4	102	803	0,42	0,38	34(3)	18	4(0)	13
5	94	521	0,30	0,26	24(3)	15	3(0)	15
6	88	755	0,43	0,37	30(4)	16	3(0)	41
7	103	1083	0,84	0,78	68(3)	31	8(0)	46
8	79	540	0,37	0,33	28(3)	16	2(0)	13
9	92	554	0,52	0,47	44(4)	22	4(0)	28
10	127	671	0,63	0,59	52(3)	25	5(0)	9

Nos résultats pour $n = 10$, avec séparation heuristique

k	p(J)	valeur	temps en sec	sépa. heuristique tps	sépa. heuristique nb	sépa. exacte tps	sépa. exacte nb	nb inég.	nb sol	nb nd
1	116	818	0,54	104(14)	1,7E-04	42(0)	0,48	49	5(0)	13
2	129	615	0,17	26(3)	6,9E-05	12(0)	0,14	16	1(0)	5
3	125	793	0,29	50(6)	1,1E-04	20(0)	0,23	35	3(0)	47
4	102	803	0,39	72(2)	1,6E-04	28(0)	0,32	45	4(0)	47
5	94	521	0,40	60(3)	1,3E-04	30(0)	0,35	32	4(0)	26
6	88	755	0,33	66(2)	1,5E-04	22(0)	0,26	45	2(0)	56
7	103	1083	0,44	68(12)	1,5E-04	31(0)	0,36	48	4(0)	69
8	79	540	0,24	40(6)	8,7E-05	16(0)	0,20	29	1(0)	32
9	92	554	0,41	56(11)	1,1E-04	33(0)	0,36	22	5(0)	7
10	127	671	0,30	40(7)	9,7E-05	22(0)	0,26	22	5(0)	13

Nos résultats pour $n = 20$, avec séparation exacte

k	p(J)	valeur	temps en sec	sépa. tps	exacte nb	nb inég	nb sol	nb nd	prop. avance	valeurs article	écart relatif
1	217	2986	52,56	50,49	192(3)	133	8(0)	424	0,44	2986	-
2	237	2980	44,72	43,65	172(3)	107	9(0)	204	0,67	2980	-
3	233	3583	44,24	42,27	166(3)	109	8(0)	578	0,46	3600	0,5%
4	230	3040	36,77	35,48	148(3)	111	6(0)	337	0,67	3040	-
5	188	2173	30,75	29,54	114(3)	96	4(0)	334	0,52	2206	1,5%
6	207	3010	70,29	69,18	246(45)	111	15(0)	208	0,53	3016	0,2%
7	244	3878	61,18	60,00	248(43)	114	12(0)	208	0,67	3900	0,6%
8	202	1638	50,07	48,79	198(11)	129	10(0)	194	0,57	1638	-
9	139	1965	72,36	70,26	214(3)	132	16(0)	424	0,51	1992	1,4%
10	216	1995	27,68	26,95	104(3)	62	6(0)	149	0,67	1995	-

Nos résultats pour $n = 20$, avec séparation heuristique

k	p(J)	valeur	temps	sépa. Heurist.		sépa. Exacte		nb inég	nb	nb	proport°	valeurs	écart
			en sec	nombre	temps	nombre	temps	ajoutées	sol	nd	avance	article	relatif
1	217	2986	53,15	326(71)	1E−03	164(0)	50,11	232	14	603	0,44	2986	-
2	237	2980	25,57	216(46)	7E−04	116(1)	24,35	145	8	278	0,67	2980	-
3	233	3583	44,10	262(63)	1E−03	148(0)	41,53	176	13	859	0,52	3600	0,5%
4	230	3040	34,28	254(67)	9E−04	119(0)	32,46	201	5	325	0,67	3040	-
5	188	2173	40,73	334(73)	1E−03	137(0)	37,10	269	9	844	0,53	2206	1,5%
6	207	3010	52,86	376(63)	1E−03	202(0)	50,63	236	11	437	0,52	3016	0,2%
7	244	3878	54,92	314(59)	1E−03	184(0)	53,68	150	13	197	0,67	3900	0,6%
8	202	1638	35,85	236(55)	9E−04	119(0)	34,90	171	9	185	0,57	1638	-
9	139	1965	47,58	390(57)	1E−03	142(0)	43,27	304	10	1373	0,53	1992	1,4%
10	216	1995	37,11	264(70)	1E−03	130(0)	35,69	203	8	195	0,65	1995	-

En résumé

Le stage :

- un état de l'art à deux entrées

En résumé

Le stage :

- un état de l'art à deux entrées
- une nouvelle formulation pour le problème due-date commune
  non restrictive
  poids quelconques

En résumé

Le stage :

- un état de l'art à deux entrées
- une nouvelle formulation pour le problème due-date commune
  non restrictive
  poids quelconques
- une partie implémentation et expérimentation

Améliorations

La suite :

- améliorer l'algorithme de séparation

Améliorations

La suite :

- améliorer l'algorithme de séparation
- enrichir la formulation pour le cas restrictif

Améliorations

La suite :

- améliorer l'algorithme de séparation
- enrichir la formulation pour le cas restrictif
- retrouver la polynomialité du cas des poids unitaires

Améliorations

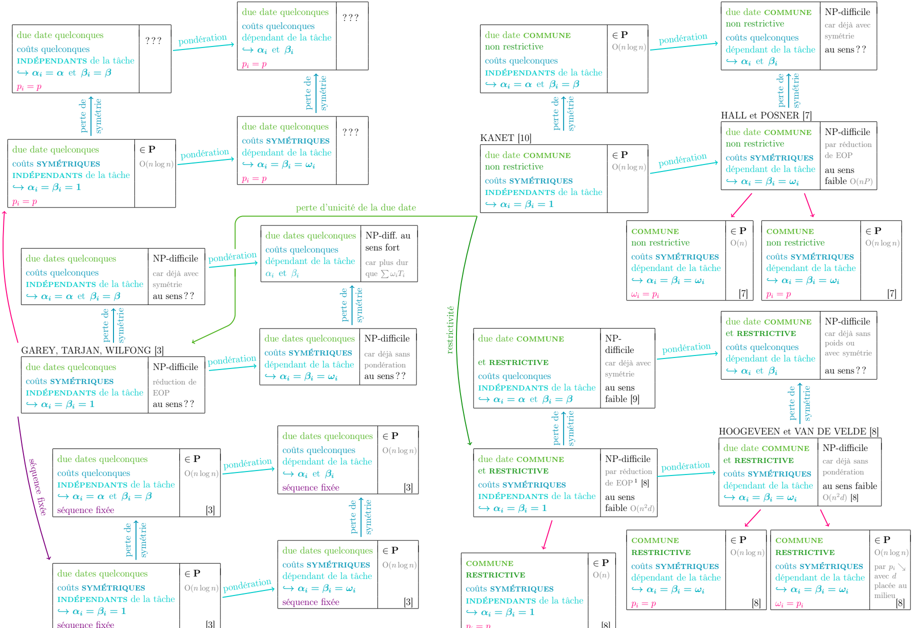
La suite :

- améliorer l'algorithme de séparation
- enrichir la formulation pour le cas restrictif
- retrouver la polynomialité du cas des poids unitaires
- comparer, envisager d'autres formulations

Améliorations

La suite :

- améliorer l'algorithme de séparation
- enrichir la formulation pour le cas restrictif
- retrouver la polynomialité du cas des poids unitaires
- comparer, envisager d'autres formulations
- caractériser complètement le polyèdre



1. EOP=EVEN-ODD PARTITION est défini en section 4

Bibliographie I

Walid BEN-AMEUR, A. Ridha MAHJOUB, and José NETO, *Le problème de coupe maximum*, pp. 17–60, Hermès, 2006.

M. Grötschel, L. Lovasz, and A. Scrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica **Vol 1** (1981), 169–197.

Michael R. Garey, Robert E. Tarjan, and Gordon T. Wilfong, *One-processor sceduling with symmectric earliness and tardiness penalties*, Mathematics of Operations Research **Vol 13** (1998), 330–348.

Nicholas G. Hall, Wieslaw Kubiak, and Suresh P.Sethi, *Earliness-tardiness sceduling problems, 2 : Deviation of completion times about a common due date*, Operations Research **Vol 39** (1991), 847–856.

Bibliographie II

Nicholas G. Hall and Marc E. Posner, *Earliness-tardiness sceduling problems, 1 : Weighted deviation of completion times about a common due date*, Operations Research **Vol 39** (1991), 836–846.

J. A. Hoogeveen and S.L. van de Velde, *Scheduling around a small common due date*, European Journal of Operational Research **Vol 55** (1991), 237–242.

Helmut G. Kahlbacher, *Termin- und ablaufplanung : ein analytischer zugang*, Ph.D. thesis, Kaiserslautern University of Technology, Germany, 1992.

John J. Kanet, *Minimizing the average deviation of job completion times about a common due date*, Naval research logistics quarterly **Vol 28** (1981), 643–651.

Bibliographie III

S. T. McCormick, *Submodular function minimization*, ch. 7, p. 321–391, Elsevier, 2006.

Maurice Queyranne, *Structure of a simple scheduling polyhedron*, Mathematical Programming **Vol 58** (1993), 263–285.

R. Tyrell Rockafellar, *Convex analysis*, Princetion University Press, 1970.

Wayne E. Smith, *Various optimizers for single-stage production*, Naval research logistics quarterly **Vol 3** (1956), 59–66.

Françis Sourd, *Ordonnancer juste-à-temps*, Mémoire d'habilitation à diriger des recherches, Université Pierre et Marie Curie, Avril 2008.