

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

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

Avril-Août 2017

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

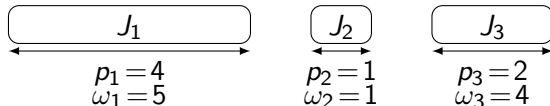
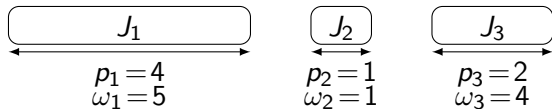


Illustration sur un exemple

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



Un ordonnancement solution :

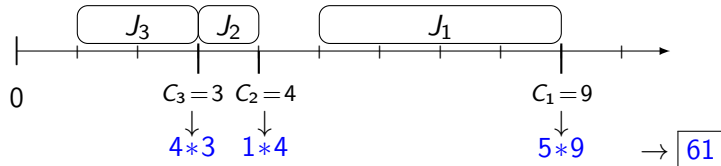
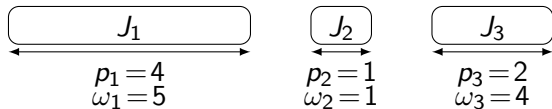
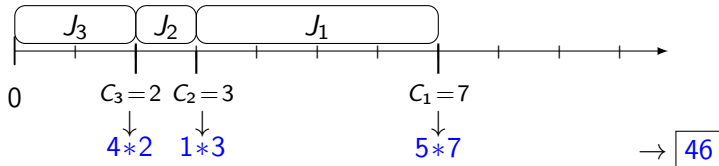


Illustration sur un exemple

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 $d \geq \sum p_j$
- 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 : $J = \{1, 2, 3, 4\}$

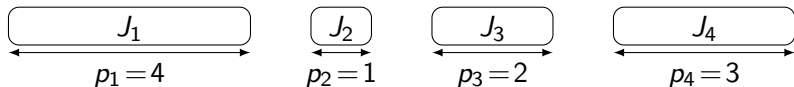
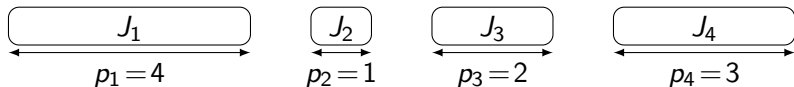


Illustration sur un exemple

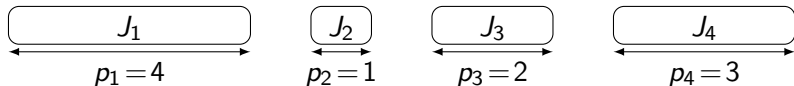
Une instance : $J = \{1, 2, 3, 4\}$



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 :

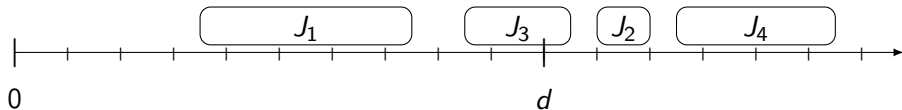
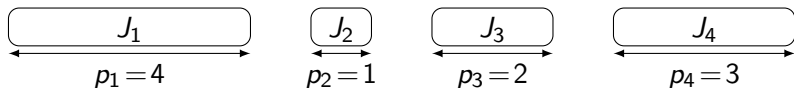


Illustration sur un exemple

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 :

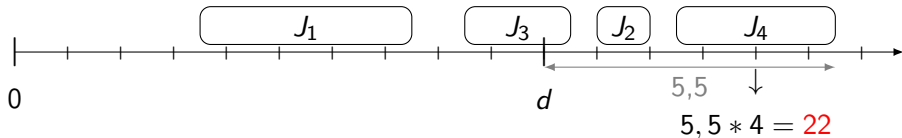
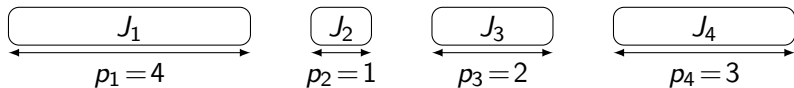


Illustration sur un exemple

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 :

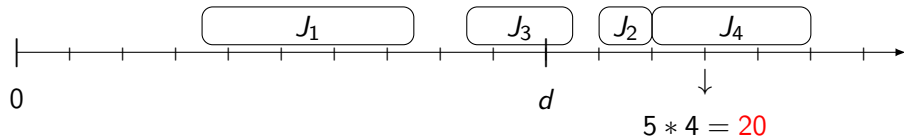
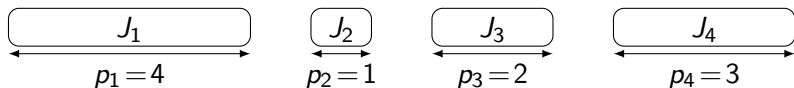


Illustration sur un exemple

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 :

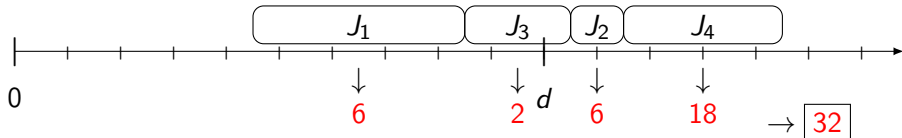
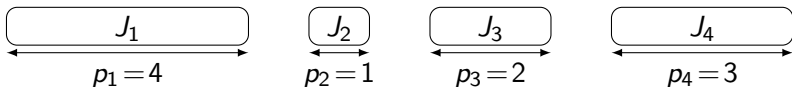


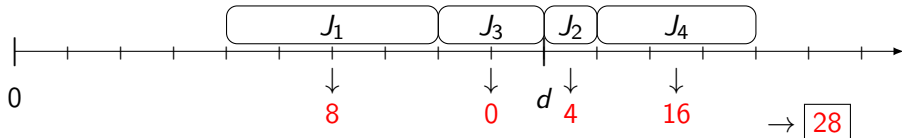
Illustration sur un exemple

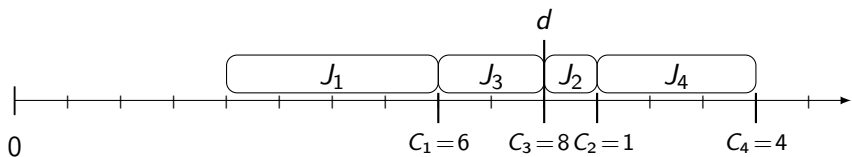
Une instance : $J = \{1, 2, 3, 4\}$

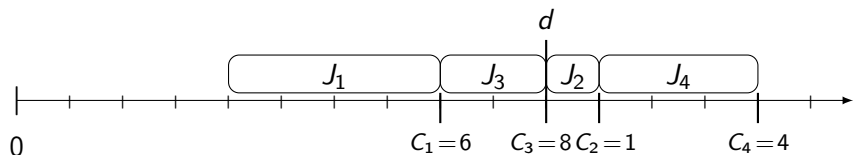


poids tous égaux : $\forall j \in J, \alpha_j = \beta_j = 4$ et $d = 10$

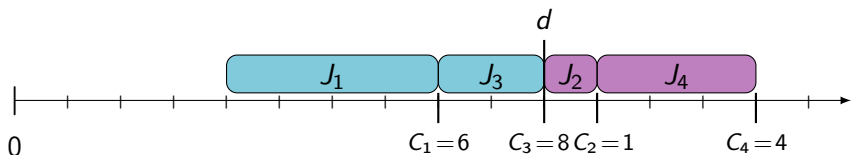
Un ordonnancement solution :



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

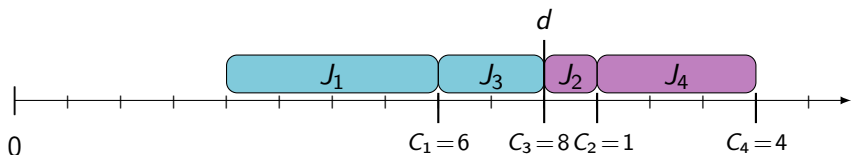
Codage par les $(C_j)_{j \in J}$ 

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

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

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

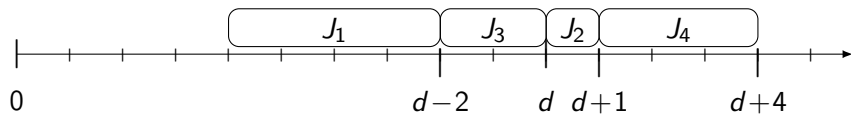
En notant $E = \{j \in J \mid C_j \leq d\}$ les tâches en avance
 $T = \{j \in J \mid C_j > d\}$ les tâches en retard

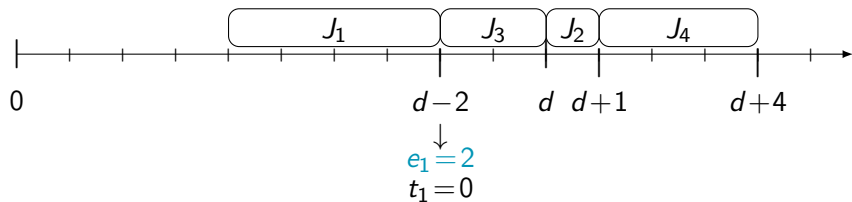
Codage par les $(C_j)_{j \in J}$ 

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

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

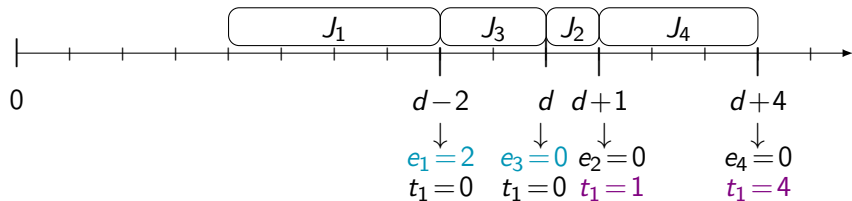
le coût s'écrit aussi
$$\sum_{j \in E} \alpha_j * (C_j - d) + \sum_{j \in T} \beta_j * (d - C_j)$$

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 $(t_j)_{j \in J}$ 

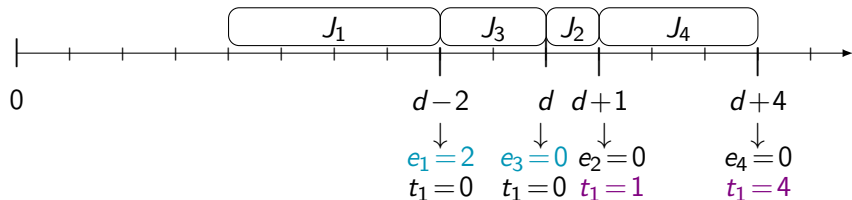
Codage par les $(e_j)_{j \in J}$ et les $(t_j)_{j \in J}$

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 $\sum_{j \in J} \alpha_j * e_j + \beta_j * t_j$ ← linéaire en les variables e_j / t_j

Plan

1. Introduction
2. État de l'art de l'ordonnancement juste-à-temps
3. État de l'art sur l'approche polyédrale du problème $\min \sum \omega_i C_i$
4. Étude polyédrale du problème juste-à-temps (coûts quelconques)
5. Étude expérimentale
6. Conclusion

1. Introduction

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

Le problème avec poids unitaires

Le problème avec poids symétriques

3. État de l'art sur l'approche polyédrale du problème $\min \sum \omega_j C_j$

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

5. Étude expérimentale

6. Conclusion

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 $\forall j \in J, \beta_j = \beta$

↔ 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 p ,
on dit qu'on a **dominance stricte** des ordonnancements vérifiant p

optimalité $\Rightarrow p$

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



optimalité $\nRightarrow p$


Illustration de la dominance $\square\square$ [Kanet,81]

L'ordonnement S vérifie $\square\square \Leftrightarrow$ il est sans trou

Propriété

*Pour le problème avec poids unitaires,
on a dominance stricte des ordonnancements $\square\square$*

Illustration de la dominance [Kanet,81]

L'ordonnement 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 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 🕒*

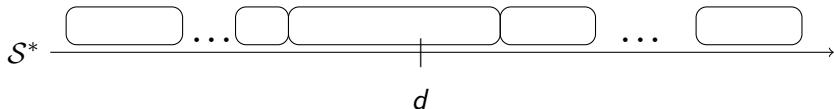


Illustration de la dominance 🕒 [Kanet,81]

L'ordonnancement 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 🕒*

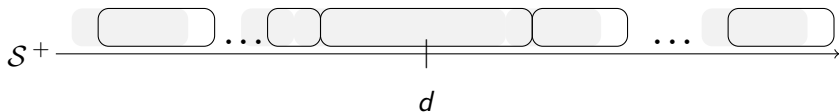


Illustration de la dominance 🕒 [Kanet,81]

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 🕒*



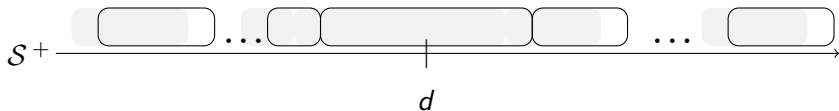
$$\text{coût}(\mathcal{S}^+) = \text{coût}(\mathcal{S}^*) - \varepsilon * \sum_{j \in E} \alpha_j + \varepsilon * \sum_{j \in T} \beta_j$$

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 🕒*



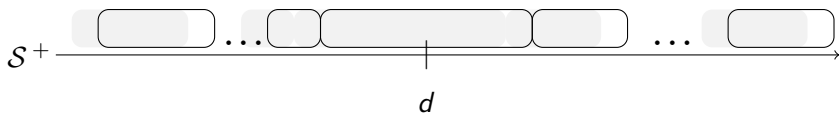
$$\text{coût}(\mathcal{S}^+) = \text{coût}(\mathcal{S}^*) - \varepsilon * \underbrace{\sum_{j \in E} \alpha_j}_{\alpha(E)} + \varepsilon * \underbrace{\sum_{j \in T} \beta_j}_{\beta(T)}$$

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 🕒



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

Illustration de la dominance 🕒 [Kanet,81]

L'ordonnancement 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 🕒*

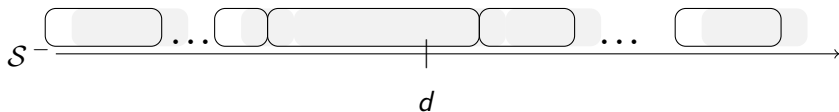
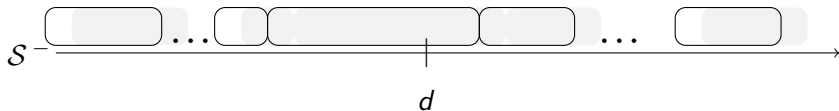


Illustration de la dominance 🕒 [Kanet,81]

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 🕒



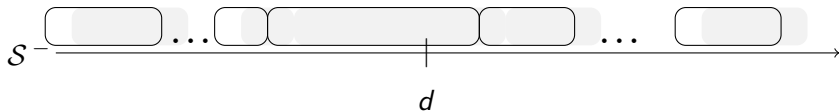
$$\text{coût}(\mathcal{S}^-) = \text{coût}(\mathcal{S}^*) - \varepsilon * [\beta(T) - \alpha(E)]$$

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 🕒



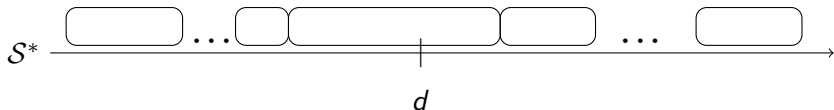
$$\text{coût}(\mathcal{S}^-) = \text{coût}(\mathcal{S}^*) - \varepsilon * [\beta(T) - \alpha(E)] \Rightarrow \alpha(E) \leq \beta(T)$$

Illustration de la dominance 🕒 [Kanet,81]

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


Propriété

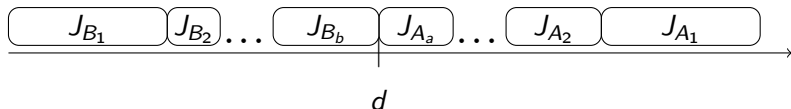
Pour le problème avec poids unitaires,
on a dominance large des ordonnancements 🕒



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

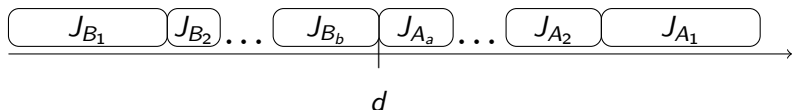
Utilisation des dominances [Kanet,81]

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



Utilisation des dominances [Kanet,81]

On cherche l'optimum parmi les ordonnancements $\square\square$ et 🕒
 donc de la forme :

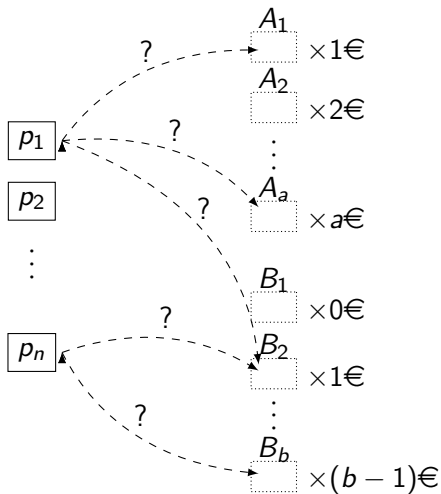


le coût s'écrit
$$\sum_{k=1}^a k * p_{A_k} + \sum_{k=1}^b (k-1) * p_{B_k}$$

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

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

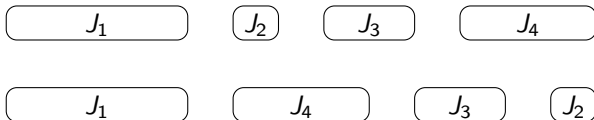
$$\sum_{k=1}^a k * p_{A_k} + \sum_{k=1}^b (k-1) * p_{B_k}$$



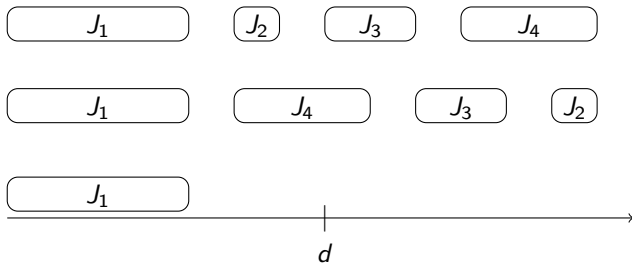
L'algorithme de Kanet sur l'exemple [Kanet,81]



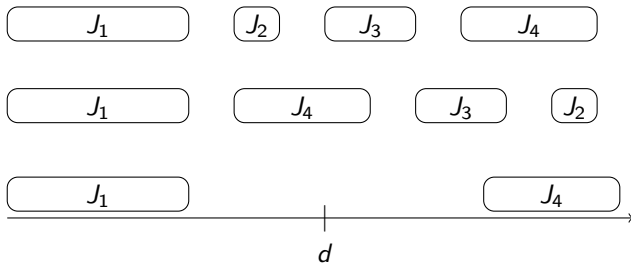
L'algorithme de Kanet sur l'exemple [Kanet,81]



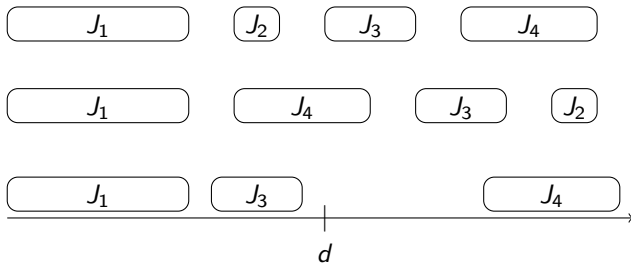
L'algorithme de Kanet sur l'exemple [Kanet,81]



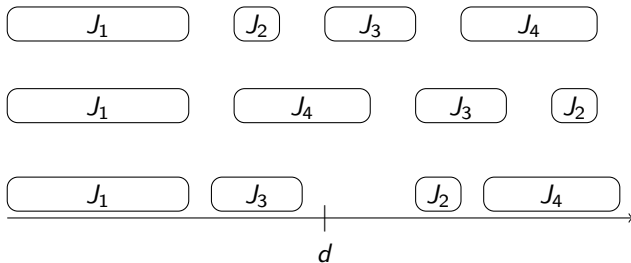
L'algorithme de Kanet sur l'exemple [Kanet,81]



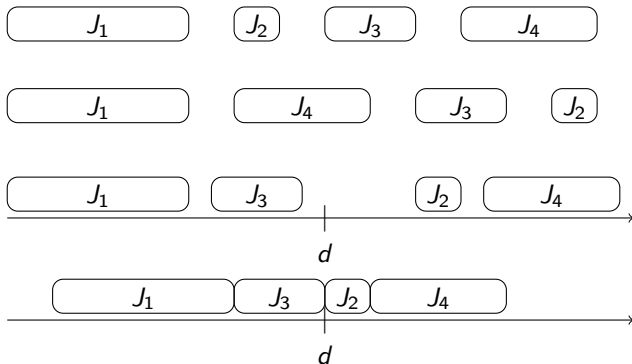
L'algorithme de Kanet sur l'exemple [Kanet,81]



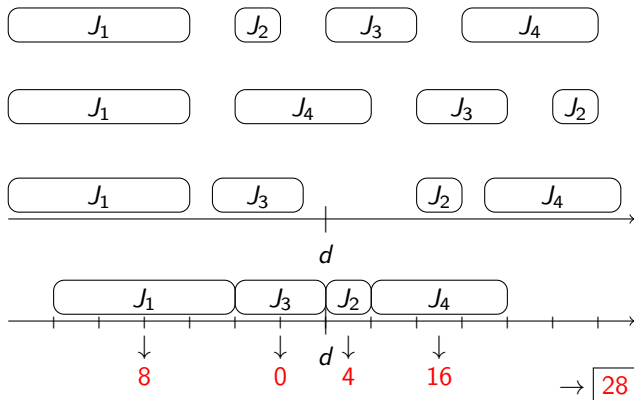
L'algorithme de Kanet sur l'exemple [Kanet,81]



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



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



Conclusion

Algorithme de Kanet en $O(n \log n)$

\Leftrightarrow le problème avec poids unitaires $\alpha_j = \beta_j = 1$ est dans P.

Le problème

Hypothèses supplémentaires :

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

↪ on pose $\forall j \in J, \omega_j := \alpha_j = \beta_j$



Ici le coût dépend de la tâche !

Dominances [Hall & Posner,91]



On pose $\forall j \in J, \mathbf{r}_j := \omega_j / p_j$

Dominances [Hall & Posner,91]

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

Propriété

Pour le problème avec coûts symétriques, on a :


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

Dominances [Hall & Posner,91]

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

Propriété

Pour le problème avec coûts symétriques, on a :

- dominance large des ordonnancements $\square\square$
- 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 $\mathbf{L} \boxplus \Leftrightarrow$ il est sans trou et tassé à gauche

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

Propriété

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

Règle de Smith [Smith,56]

L'ordonnement \mathcal{S} vérifie $\mathbb{1} \boxplus \boxminus \Leftrightarrow$ il est sans trou et tassé à gauche

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

Propriété

*Pour le problème de minimisation de la somme des dates de fins,
on a dominance stricte des ordonnancements $\mathbb{1} \boxplus \boxminus$ et \searrow^r*


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

Règle de Smith [Smith,56]

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

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

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 $(\omega_j)_{j \in J}$

$Q :=$ 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 $(\omega_j)_{j \in J}$

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

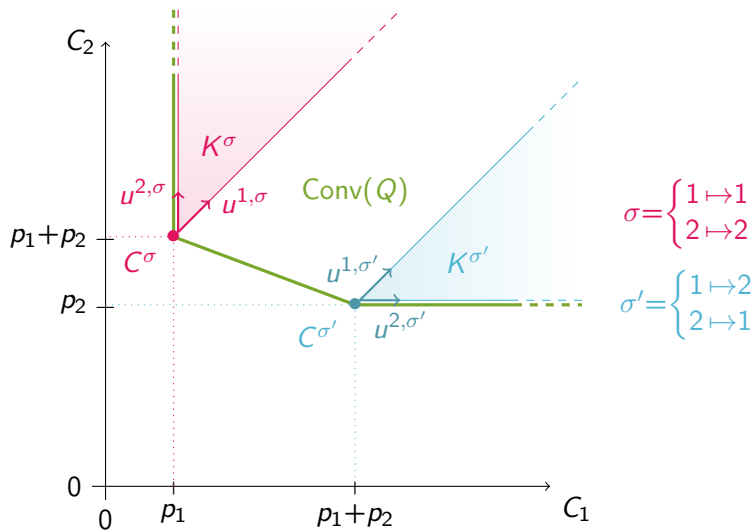
Ensemble des solutions [Queyranne,93]

Une instance =

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

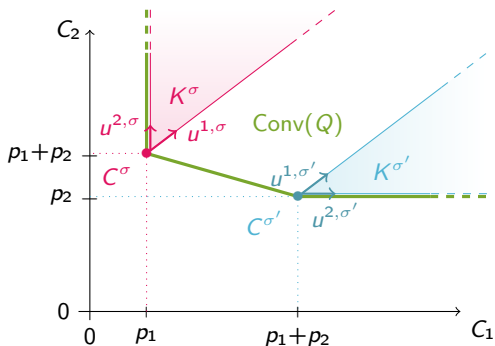
$$Q := \left\{ C \in \mathbb{R}^J \mid \begin{array}{l} \forall j \in J, C_j \geq p_j \\ \forall (j, k) \in J^2, C_k \geq C_j + p_k \text{ ou } C_j \geq C_k + p_j \end{array} \right\}$$

↔ On va étudier $\text{Conv}(Q)$

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

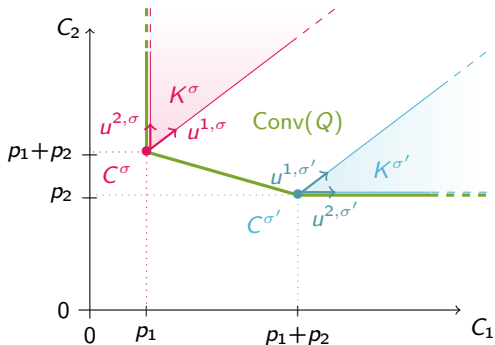
Description géométrique [Queyranne,93]

$$Q = \bigcup_{\sigma \in \mathcal{G}} K^\sigma$$



Description géométrique [Queyranne,93]

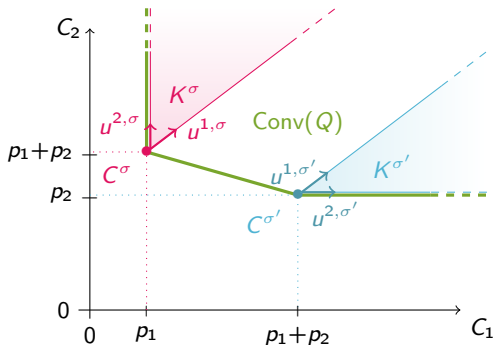
$$Q = \bigcup_{\sigma \in \mathcal{G}} \text{Conv}(\{C^\sigma\}) + \text{CC}^\circ(\{u^{i,\sigma} \mid i \in [1..n]\})$$



Description géométrique [Queyranne,93]

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

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

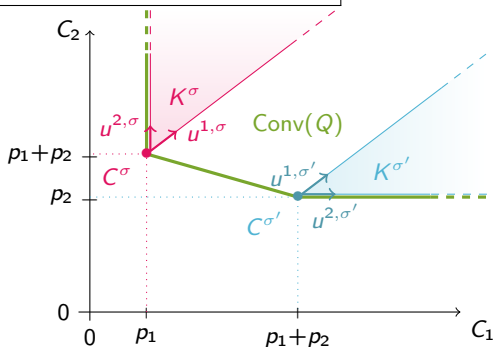


Description géométrique [Queyranne,93]

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

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

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



Inégalités de Queyranne [Queyranne,93]

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

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

Inégalités de Queyranne [Queyranne,93]

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

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

$$\mathbf{g} := \left(\begin{array}{l} \mathcal{P}(J) \rightarrow \mathbb{R} \\ S \mapsto \min_{C \in Q} \langle \omega^S \mid C \rangle \end{array} \right)$$

Inégalités de Queyranne [Queyranne,93]

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

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

$$\mathbf{g} := \left(\begin{array}{l} \mathcal{P}(J) \rightarrow \mathbb{R} \\ S \mapsto \min_{C \in Q} p * C(S) \end{array} \right) \text{ où } p * C(S) = \sum_{i \in S} p_i * C_i$$

Inégalités de Queyranne [Queyranne,93]

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

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

$$\mathbf{g} := \left(\begin{array}{l} \mathcal{P}(J) \rightarrow \mathbb{R} \\ S \mapsto \min_{C \in Q} p * C(S) \end{array} \right) \text{ où } p * C(S) = \sum_{i \in S} p_i * C_i$$

Polyèdre de Queyranne : $P^Q := \{ C \in \mathbb{R}^J \mid \forall S \subset J, p * C(S) \geq g(S) \}$

Inégalités de Queyranne [Queyranne,93]

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

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

$$\mathbf{g} := \left(\begin{array}{l} \mathcal{P}(J) \rightarrow \mathbb{R} \\ S \mapsto \min_{C \in Q} p^* C(S) \end{array} \right) \text{ où } p^* C(S) = \sum_{i \in S} p_i * C_i$$

Polyèdre de Queyranne : $P^Q := \{ C \in \mathbb{R}^J \mid \forall S \subset J, p^* C(S) \geq g(S) \}$

Résultat : $P^Q = \text{Conv}(Q)$

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]

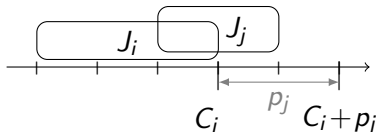
But point extrême \Rightarrow sans-chevauchement

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

$$C_i \leq C_j < C_i + p_j$$

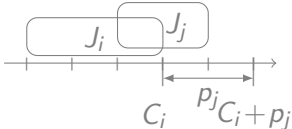


On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} C^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ C^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$

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

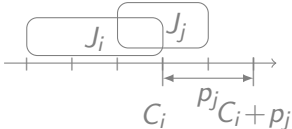
On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} C^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ C^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$


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

→ Si $i \in S$ et $j \in S$

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} C^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ C^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$


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

→ Si $i \in S$ et $j \in S$ ✓

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} C^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ C^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$

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

→ Si $i \in S$ et $j \in S$ ✓

→ Si $i \notin S$ et $j \notin S$ ✓

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} C^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ C^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$

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

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

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} C^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ C^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$

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

→ Si $i \in S$ et $j \in S$ ✓

→ Si $i \notin S$ et $j \notin S$ ✓

→ Si $i \notin S$ et $j \in S$

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

On aimerait poser... [AEF]

$$C = \frac{C^{+-} + C^{-+}}{2} \quad \text{où} \quad \begin{aligned} \mathbf{C}^{+-} &:= C + \frac{\varepsilon}{p_i} \mathbb{1}_i - \frac{\varepsilon}{p_j} \mathbb{1}_j \\ \mathbf{C}^{-+} &:= C - \frac{\varepsilon}{p_i} \mathbb{1}_i + \frac{\varepsilon}{p_j} \mathbb{1}_j \end{aligned}$$

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

→ Si $i \in S$ et $j \in S$ ✓

→ Si $i \notin S$ et $j \notin S$ ✓

→ Si $i \notin S$ et $j \in S$

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

→ Si $i \in S$ et $j \notin S$

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

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

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

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leq C_j$, **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 $\left. \begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array} \right\}$ tel que $\boxed{p^*C(S^1) = g(S^1)}$

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

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

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

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

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

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

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

Si $C_i \leq C_j$, **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 $\left. \begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array} \right\}$ tel que $\boxed{p^*C(S^1) = g(S^1)}$

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

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

Si $C_i \leq C_j$, **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 $\left. \begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array} \right\}$ tel que $p^* C(S^1) = g(S^1)$

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

$$\begin{array}{ccc} \parallel & \forall & \\ g(S^1) & = & g(\tilde{S}^1) + p_j p(S^1) \end{array}$$

Vérifier que $\varepsilon_1 > 0$ [AEF]

Lemme

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

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

$$p^* C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

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

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

$$p^* C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

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

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

$$p^* C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

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

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

$$p^* C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

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

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

$$p^*C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

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

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

$$p^*C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_1 > 0$ [AEF]Lemme

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

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

$$p^*C(S^1) = g(S^1)$$

$$C_j \leq p(S^1)$$

Vérifier que $\varepsilon_2 > 0$ [AEF]Lemme

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

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

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

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

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

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

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

De même on a $C_i \leq p(S^2)$

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

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

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

De même on a $C_i \leq p(S^2)$

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

Vérifier que $\varepsilon_2 > 0$ [AEF]Lemme

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

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

De même on a $C_i \leq p(S^2)$

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

Vérifier que $\varepsilon_2 > 0$ [AEF]Lemme

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

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 $\boxed{C_i \leq p(S^2)}$

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

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

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

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

De même on a $C_i \leq p(S^2)$

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

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 $\left. \begin{array}{l} i \notin S^1 \\ j \in S^1 \end{array} \right\}$ tel que $p^* C(S^1) = g(S^1)$

De même on a $C_i \leq p(S^2)$

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

Vérifier que $\varepsilon_2 > 0$ [AEF]

Lemme

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

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

De même on a $C_i \leq p(S^2)$

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

1. Introduction
2. État de l'art de l'ordonnancement juste-à-temps
3. État de l'art sur l'approche polyédrale du problème $\min \sum \omega_i C_i$
4. Étude polyédrale du problème juste-à-temps (coûts quelconques)
 - Formulation (e, t, δ, l, r)
 - Intérêt du polyèdre $P^{e,t,\delta,l,r}$
 - Problème de séparation associé
5. Étude expérimentale
6. Conclusion

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 $d \geq \sum p_j$
- 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 $d \geq \sum p_j$
- 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 $\rightarrow \delta_j$

$$\forall j \in J, e_j \geq 0 \quad (e.1)$$

$$e_j \leq M\delta_j \quad (e.2)$$

$$\forall j \in J, t_j \geq 0 \quad (t.1)$$

$$t_j \leq M(1 - \delta_j) \quad (t.2)$$

Assurer la cohérence [AEF]

- Introduire des variables disjonctives $\rightarrow \delta_j$

$$\forall j \in J, e_j \geq 0 \quad (e.1)$$

$$e_j \leq M\delta_j \quad (e.2)$$

$$\forall j \in J, t_j \geq 0 \quad (t.1)$$

$$t_j \leq M(1 - \delta_j) \quad (t.2)$$

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

Assurer la cohérence [AEF]

- Introduire des variables disjonctives $\rightarrow \delta_j$

$$\forall j \in J, e_j \geq 0 \quad (e.1)$$


$$e_j \leq M\delta_j \quad (e.2)$$

$$\forall j \in J, t_j \geq 0 \quad (t.1)$$

$$t_j \leq M(1 - \delta_j) \quad (t.2)$$


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

<p>non-chevauchement =</p>	<p>tâches en avance à gauche de d</p>	<p>et</p>	<p>tâches en retard à droite de d</p>
----------------------------	--	-----------	--

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

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

$$\left\{ \begin{array}{l} \forall S \subset J, p * t(S) \geq g(S) \\ \end{array} \right.$$

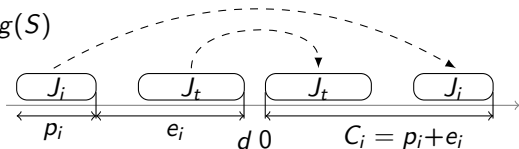
Idée pour assurer le non-chevauchement [AEF]

- 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

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

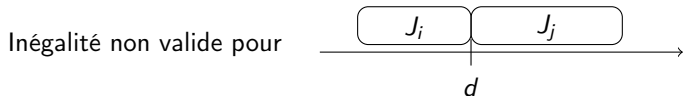
$$\begin{cases} \forall S \subset J, p * t(S) \geq g(S) \\ \forall S \subset J, p * [p + e](S) \geq g(S) \end{cases}$$



Ces inégalités sont-elles valides ? [AEF]

Pour $S = \{i, j\}$ où $t_i = 0$ l'inégalité s'écrit :

$$\begin{aligned} p_j t_j &\geq \frac{1}{2} [(p_i + p_j)^2 + (p_i^2 + p_j^2)] \\ &= p_i^2 + p_j^2 + p_i p_j \\ &> p_j^2 \end{aligned}$$



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

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

Éliminer les intersections [AEF]

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

Éliminer le terme quadratique [AEF]

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

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

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

Éliminer le terme quadratique [AEF]

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

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

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

Éliminer le terme quadratique [AEF]

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

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

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

Éliminer le terme quadratique [AEF]

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

$$\forall (i,j) \in J^<, l_{i,j} \geq 0 \quad (l.1)$$

$$l_{i,j} \leq \delta_i \quad (l.2)$$

$$l_{i,j} \leq \delta_j \quad (l.3)$$

$$l_{i,j} \geq \delta_i + \delta_j - 1 \quad (l.4)$$

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

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

$$\forall (i,j) \in J^<, r_{i,j} \geq 0 \quad (r.1)$$

$$r_{i,j} \leq (1 - \delta_i) \quad (r.2)$$

$$r_{i,j} \leq (1 - \delta_j) \quad (r.3)$$

$$r_{i,j} \geq 1 - \delta_i - \delta_j \quad (r.4)$$

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

Formulation [AEF]

$$P^{e,t,\delta,l,r} := \{(e, t, \delta, l, r) \in \mathbb{R}^J \times \mathbb{R}^J \times \mathbb{R}^J \times \mathbb{R}^{J^<} \times \mathbb{R}^{J^<} \mid$$

$$\forall j \in J, e_j \geq 0 \quad (e.1)$$

$$e_j \leq M\delta_j \quad (e.2)$$

$$\forall j \in J, t_j \geq 0 \quad (t.1)$$

$$t_j \leq M(1 - \delta_j) \quad (t.2)$$

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

$$\forall (i, j) \in J^<, l_{i,j} \geq 0 \quad (l.1)$$

$$l_{i,j} \leq \delta_i \quad (l.2)$$

$$l_{i,j} \leq \delta_j \quad (l.3)$$

$$l_{i,j} \geq \delta_i + \delta_j - 1 \quad (l.4)$$

$$\forall (i, j) \in J^<, r_{i,j} \geq 0 \quad (r.1)$$

$$r_{i,j} \leq (1 - \delta_i) \quad (r.2)$$

$$r_{i,j} \leq (1 - \delta_j) \quad (r.3)$$

$$r_{i,j} \geq 1 - \delta_i - \delta_j \quad (r.4)$$

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

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

}

Extr* [AEF]

Définition

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

Extr* [AEF]

Définition

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

Propriété

Si $x \in \text{Extr}^*$ **alors** x code un ordonnancement réalisable $\square\square$ et .

Extr* [AEF]

Définition

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

Propriété

Si $x \in \text{Extr}^*$ **alors** x code un ordonnancement réalisable $\square\square$ et 🕒 .

Propriété

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

Inégalités à séparer

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

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

deux familles d'inégalités en nombre exponentiel

↪ deux problèmes de séparation parallèles

Inégalités à séparer

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

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

deux familles d'inégalités en nombre exponentiel

↪ deux problèmes de séparation parallèles

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

Inégalités à séparer

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

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

deux familles d'inégalités en nombre exponentiel

↪ deux problèmes de séparation parallèles

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

Inégalités à séparer

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

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

deux familles d'inégalités en nombre exponentiel

↪ deux problèmes de séparation parallèles

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

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

Inégalités à séparer

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

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

deux familles d'inégalités en nombre exponentiel

↪ deux problèmes de séparation parallèles

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

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

↪ séparation 1 = maximisation de Γ_1

Inégalités à séparer

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

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

deux familles d'inégalités en nombre exponentiel

↪ deux problèmes de séparation parallèles

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

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

séparation 2 = maximisation de Γ_2
 ↪ séparation 1 = maximisation de Γ_1

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

↪ algorithme de séparation polynomial

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

↪ algorithme de séparation polynomial

↪ algorithme de coupes polynomial

Difficulté du problème

Γ_1 et Γ_2 sont super modulaires

↪ algorithme de séparation polynomial

↪ algorithme de coupes polynomial

↪ difficulté du problème due aux δ entiers

1. Introduction
2. État de l'art de l'ordonnancement juste-à-temps
3. État de l'art sur l'approche polyédrale du problème $\min \sum \omega_j C_j$
4. Étude polyédrale du problème juste-à-temps (coûts quelconques)
5. Étude expérimentale
 - Un algorithme de Branch-and-Cut
 - Résultats expérimentaux
6. Conclusion

De la théorie à la pratique

- Formulation mixte \leftrightarrow algorithme de branchement

De la théorie à la pratique

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

De la théorie à la pratique

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

De la théorie à la pratique

- Formulation mixte \leftrightarrow algorithme de branchement
- Notre formulation n'est pas compacte \leftrightarrow algorithme de coupe
- En dur les inégalités $(e,-)$, $(t,-)$, $(\delta,-)$, $(l,-)$, $(r,-)$ 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 $(e,-)$, $(t,-)$, $(\delta,-)$, $(l,-)$, $(r,-)$ 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

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

Nos résultats pour $n=10$

k	p(J)	valeur de l'objectif	propor° avance	borne sup	temps en sec	nbr de coupes	nbr de nœuds
1	116	818	0,66	818	<1	10	5
2	129	615	0,40	615	<1	12	0
3	125	793	0,54	793	<1	14	10
4	102	803	0,67	803	<1	13	0
5	94	521	0,54	521	<1	15	4
6	88	755	0,50	755	<1	13	0
7	103	1083	0,60	1083	1,4	18	4
8	79	540	0,77	540	<1	13	4
9	92	554	0,67	554	1,3	12	12
10	127	671	0,69	671	<1	11	0

Nos résultats pour $n=20$

k	p(J)	valeur objectif	propor° avance	borne sup	écart relatif	temps en sec	nbr de coupes	nbr de nœuds
1	217	2986	0,44	2986	0	73	56	269
2	237	2980	0,70	2980	0	76	54	107
3	233	3583	0,47	3600	0,47%	65	68	747
4	230	3040	0,70	3040	0	90	48	145
5	188	2173	0,53	2206	1,5%	83	77	354
6	207	3010	0,53	3016	0,19%	68	96	364
7	244	3878	0,67	3900	0,56%	71	53	250
8	202	1638	0,53	1638	0	161	52	144
9	139	1965	0,51	1992	1,3%	107	76	505
10	216	1972	0,62	1995	1,1%	71	33	100

Nos résultats pour $n=20$

k	p(J)	valeur objectif	propor ^o avance	borne sup	écart relatif	temps en sec	nbr de coupes	nbr de nœuds
1	217	2986	0,44	2986	0	73	56	269
2	237	2980	0,70	2980	0	76	54	107
3	233	3583	0,47	3600	0,47%	65	68	747
4	230	3040	0,70	3040	0	90	48	145
5	188	2173	0,53	2206	1,5%	83	77	354
6	207	3010	0,53	3016	0,19%	68	96	364
7	244	3878	0,67	3900	0,56%	71	53	250
8	202	1638	0,53	1638	0	161	52	144
9	139	1965	0,51	1992	1,3%	107	76	505
10	216	1972	0,62	1995	1,1%	71	33	100

1. Introduction
2. État de l'art de l'ordonnancement juste-à-temps
3. État de l'art sur l'approche polyédrale du problème $\min \sum \omega_i C_i$
4. Étude polyédrale du problème juste-à-temps (coûts quelconques)
5. Étude expérimentale
6. Conclusion

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, ajouter des heuristiques, des tests

Améliorations

La suite :

- améliorer l'algorithme de séparation, ajouter des heuristiques, des tests
- retrouver la polynomialité du cas des poids unitaires

Améliorations

La suite :

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

Améliorations

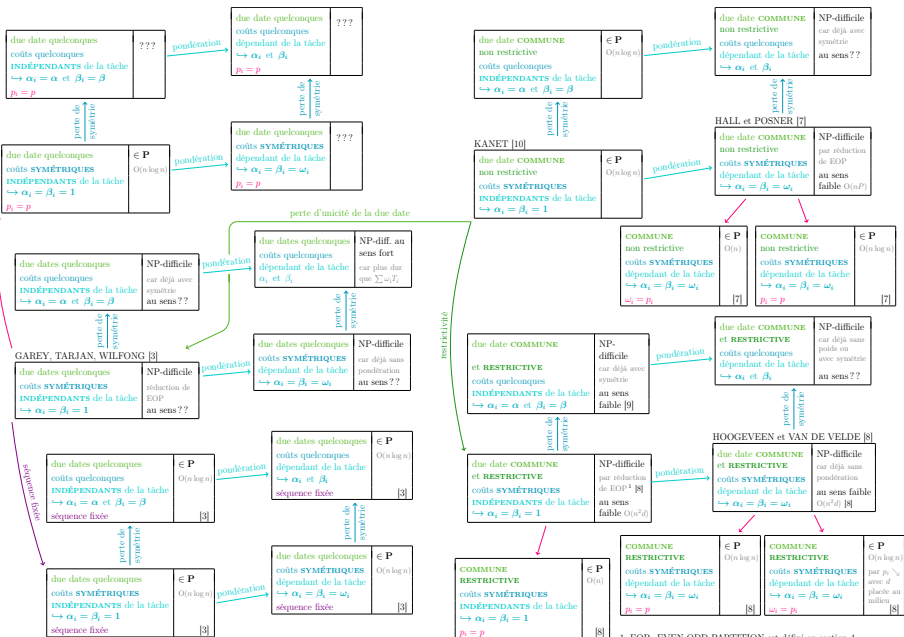
La suite :

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

Améliorations





La suite :

- améliorer l'algorithme de séparation, ajouter des heuristiques, des tests
- retrouver la polynomialité du cas des poids unitaires
- enrichir la formulation pour le cas restrictif
- 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 MAHJOUR, and José NETO, *Le problème de coupe maximum*, pp. 17–60, Hermès, 2006.
-  M. Grötschel, L. Lovasz, and A. Schrijver, *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 scheduling with symmetric earliness and tardiness penalties*, *Mathematics of Operations Research* **Vol 13** (1998), 330–348.
-  Nicholas G. Hall, Wieslaw Kubiak, and Suresh P. Sethi, *Earliness-tardiness scheduling 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 scheduling 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*, Princeton University Press, 1970.
-  Wayne E. Smith, *Various optimizers for single-stage production*, *Naval research logistics quarterly* **Vol 3** (1956), 59–66.
-  François Sourd, *Ordonnancer juste-à-temps*, Mémoire d'habilitation à diriger des recherches, Université Pierre et Marie Curie, Avril 2008.