Ordonnancer « Juste-à-Temps » à l'aide de plusieurs critères

Vincent T'kindt - Bertrand Estève

Laboratoire d'Informatique
DI/Polytech'Tours - FRANCE

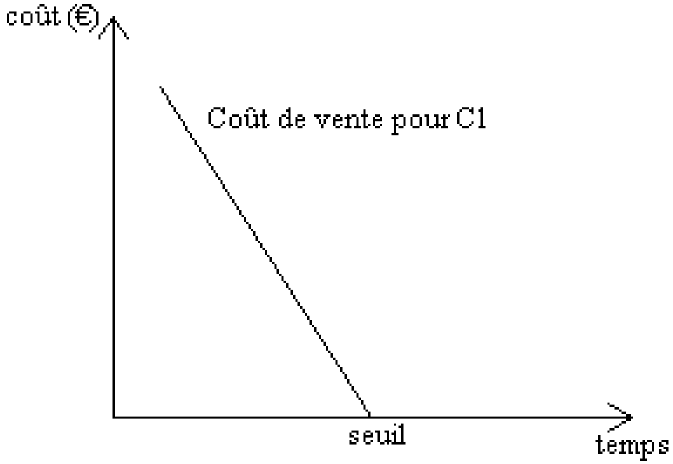
JFRO - 21 novembre 2003

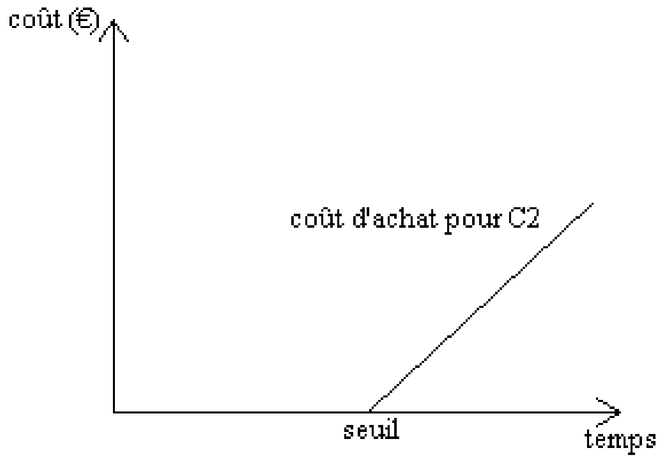
École polytechnique de l'Université de Tours
Département Informatique

Laboratoire d'Informatique

Exemple introductif

- Deux courtiers C1 et C2 doivent négocier entre eux,
- Le courtier C1 possède un portefeuille d'actions (fixe et connu) qu'il doit vendre à C2,
- Chaque action est définie par un coût d'achat dont la variation est supposée linéaire par morceaux,





Les coûts sont fonctions du cours de l'action :
C1 $\Rightarrow$ (Prix achat initial - cours actuel) / C2 $\Rightarrow$ (cours actuel - Prix achat moyen chez un autre courtier).

Exemple introductif

– Les deux courtiers, qui se connaissent bien, cherchent à minimiser leurs coûts d'achat $\Rightarrow$ Objectifs conflictuels,
– Chaque transaction doit être enregistrée par ordinateur. Le temps d'enregistrement est propre à chaque action et connu à l'avance,
– Règle du jeu (par tour) :

1. C2 choisit une action et provoque la transaction,
2. C2 décide du moment où la transaction débutera (datage),
3. C1 choisit une action et provoque la transaction,
4. C1 décide du moment où la transaction débutera (datage).

– Le but est de réaliser les ventes des actions tout en minimisant les coûts de chaque courtier.

$\Rightarrow$ Ce problème est bicritère.

Exemple introductif

- Il peut être formulé comme un problème d'ordonnancement de type "Juste-à-Temps",
  - Ordinateur = Ressource,
  - Action = Travail (seuil=d_i, temps d'enregistrement = p_i) ,
  - A chaque travail on attache une avance pondérée (coût de C1) et un retard pondéré (coût de C2),
  - Les deux critères sont l'avance totale pondérée $\bar{E}^{\alpha}$ et le retard total pondéré $\bar{T}^{\beta}$.

$\Rightarrow$ Pourtant dans la littérature, ce type de problème est rarement présenté comme étant bicritère (aggrégation des deux critères).

Plan de la présentation

1. L'ordonnancement en JaT : état des lieux

2. Ordonnancer en JaT : une approche au moins bicritère

3. Gestion de production et ordonnancement en JaT

L'ordonnancement en JaT : état des lieux

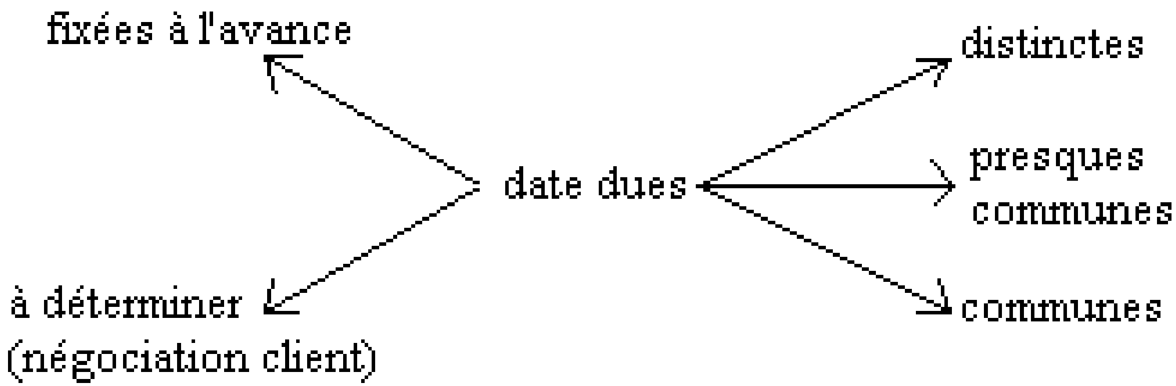
- Notations :
  - Considérons une seule machine,
  - n travaux (p_i, d_i, α_i, β_i),
  - $\bar{E}^\alpha = \sum_i \alpha_i E_i$ avec $E_i = \max(0, d_i - C_i)$,
  - $\bar{T}^\beta = \sum_i \beta_i T_i$ avec $T_i = \max(0, C_i - d_i)$.

- Préliminaire,
  - On utilise la notation $\alpha|\beta|\gamma$ ([Graham et al., 1979]),
  - Elle est étendue aux problèmes d'ordonnancement multicritères dans [T'kindt et Billaut, 2002].

- Première définition :

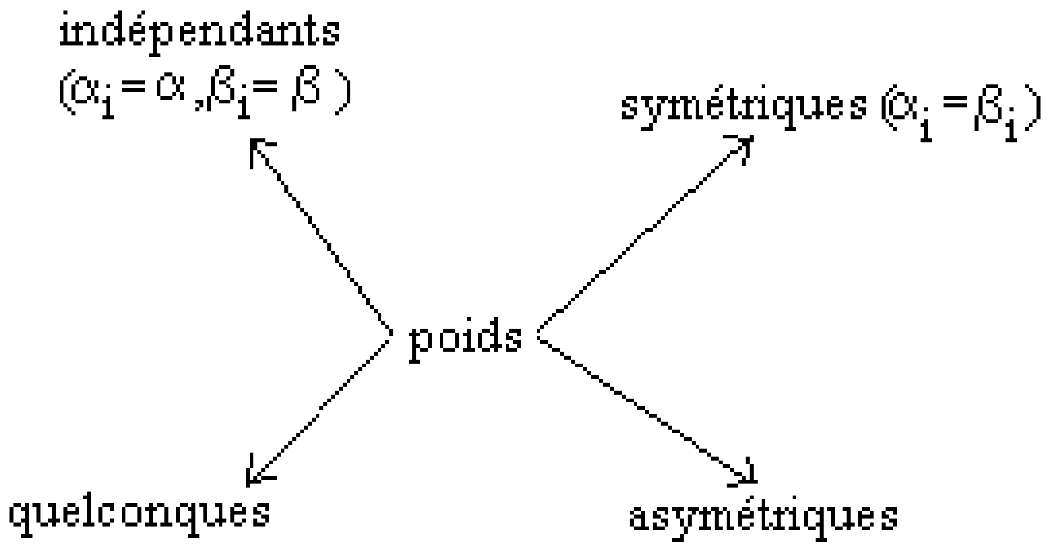
  Ordonnancer en Juste-à-Temps c'est programmer la fin d'exécution des travaux aussi proche que possible de leur date de fin souhaitée.

L'ordonnancement en JaT : état des lieux



Les dates dues communes peuvent être restrictives ou non restrictives.
Exemple : dates dues presques communes à déterminer ([Seidmann et al, 1981])
On connait une date due commune souhaitée, notée A et on souhaite minimiser comme critère $\bar{A} = \sum_i \max(0; d_i - A)$ en complément de l'avance totale et du retard total.

L'ordonnancement en JaT : état des lieux



– De nombreux modèles de fonctions de coût existent dans la littérature ([T'kindt et Billaut, 2002]),

L'ordonnancement en JaT : état des lieux

Fonction de coût
$\overline{E} + \overline{T}$
$\alpha\overline{E} + \beta\overline{T}$ (early/tardy)
$\overline{E}^{\alpha} + \overline{T}^{\beta}$
$(\overline{E} + \overline{T})^2$
$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \lvert C_i - C_j \rvert$
$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (C_i - C_j)^2$ ou $\sum_{i=1}^{n} \sum_{j=i+1}^{n} (C_i - \overline{C})^2$
$\alpha \sum_{i=1}^{n} E_i^2 + \beta \sum_{i=1}^{n} T_i^2$ ou $\sum_{i=1}^{n} \alpha_i E_i^2 + \sum_{i=1}^{n} \beta_i T_i^2$
$\max_{i=1,\ldots,n}(\alpha_i \lvert L_i \rvert)$
$\max(g(E_{max}), h(T_{max}))$ (increasing functions)
$\max_{i=1,\ldots,n}(g(E_i), h(T_i))$ (convex functions)
$\alpha\overline{E} + \beta\overline{T} + \gamma d$
$\alpha\overline{E} + \beta\overline{T} + \gamma\overline{C}$ (early/tardy+coût stockage partiel)
$\alpha\overline{E} + \beta\overline{T} + \gamma\max(0, d_i - A)$ with A a due date
$\overline{U}^{\alpha} + \overline{E}^{\beta}$

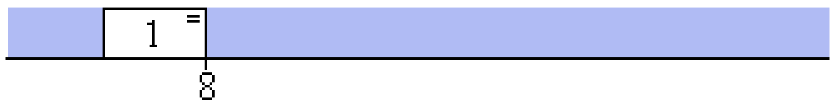
L'ordonnancement en JaT : état des lieux

- La fonction objectif ne constitue pas une fonction *régulière*,
- On est confronté à un double problème : séquencer puis calculer les dates de début optimales (*optimal timing*),

Illustration : le cas du problème $1|d_i, seq|F_\ell(\bar{T}, \bar{E})$ avec $F_\ell(\bar{T}, \bar{E}) = \bar{T} + \bar{E} = \sum_i |C_i - d_i|$ ([Garey et al, 1988])

i	1	2	3	4	5
p_i	2	4	3	5	2
d_i	8	10	14	18	19

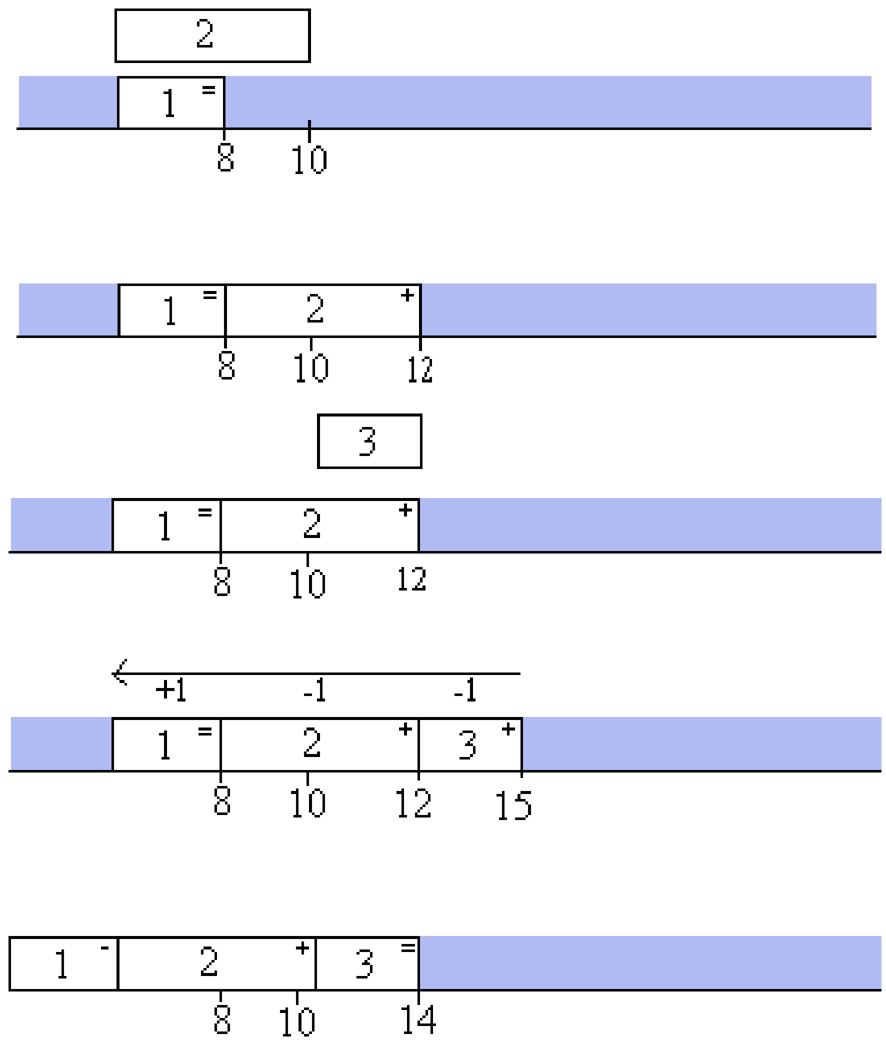
La séquence est $(1, 2, ..., n)$



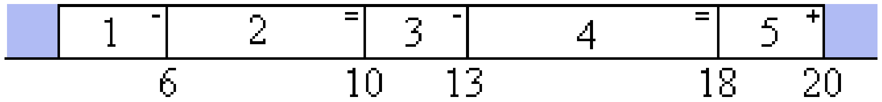
L'ordonnancement en JaT : état des lieux

i	1	2	3	4	5
p_i	2	4	3	5	2
d_i	8	10	14	18	19

La séquence est $(1, 2, ..., n)$



L'ordonnancement en JaT : état des lieux



$$\sum_i |C_i - d_i| = 4$$

- Algorithme en $O(n \log(n))$,
- Le problème $1|d_i|F_\ell(\bar{E}, \bar{T})$ avec $F_\ell(\bar{E}, \bar{T}) = \sum_i |C_i - d_i|$, est $\mathcal{NP}$-difficile,
- Le problème de calcul des dates de début a suscité de nombreux travaux,

L'ordonnancement en JaT : état des lieux

Problème	Complexité	Références
$1\|d_i, seq\|\bar{E} + \bar{T}$	Polynomial	[Garey et al., 1988]
$1\|d_i, seq\|\bar{E}^{\alpha} + \bar{T}^{\beta}$	Polynomial	[Davis et Kanet, 1993] [Szwarc et Mukhopadhyay, 1995]
$P\infty\|d_i, prec\|\bar{E}^{\alpha} + \bar{T}^{\beta}$	Polynomial	[Wennink, 1995] [Della Croce et Trubian, 2002] [Chrétienne et Sourd, 2003]
$P\infty\|d_i, prec\|\sum_i f_i(C_i)$	———	[Wennink, 1995] [Chrétienne et Sourd, 2003]

Note : Les approches proposées par [Wennink, 1995] et [Chrétienne et Sourd, 2003] peuvent être étendues pour prendre en compte d'autres données et/ou contraintes comme des r_i et $\tilde{d}_i$.

Note : Lorsque les fonctions de coûts f_i sont linéaires, le problème peut être résolu à l'aide de la Programmation Linéaire Simple.

L'ordonnancement en JaT : état des lieux

- Schéma classique de résolution d'un problème de type “Juste-à-Temps”,
  1. Calcule d'une séquence en temps polynomial,
  2. Résolution du problème d'*optimal timing*,
  3. Retour à l'étape 1 tant qu'une condition d'arrêt n'est pas satisfaite.

  Note : En règle générale, ne donne pas de bons résultats pour un faible nombre d'itérations ([Davis et Kanet, 1993]).

- Le cas du problème $1|d_i|F_\ell(\bar{E}^\alpha, \bar{T}^\beta)$,
  - d'un point de vue “heuristique” : par voisinage ([Fry et al., 1987]), Tabou ([James et Buchanan, 1997, 1998]),
  - d'un point de vue “exact” : PSE ([Sourd et Kedad-Sidhoum, 2003]/30 travaux).

    Note : pour le $1|d_i, nmit|F_\ell(\bar{E}^\alpha, \bar{T}^\beta)$ [Liaw, 1999] $\Rightarrow$ 30 travaux.

L'ordonnancement en JaT : état des lieux

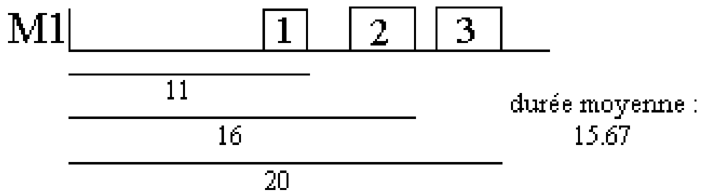
- *Et si laisser inactive une ressource coûtait TROP cher ?*
  $\Rightarrow$ Ajout de la contrainte *no machine idle time* (nmit),
  - Les problèmes du type $1|d_i, nmit|F_\ell(\bar{E}^\alpha, \bar{T}^\beta)$ ont fait l'objet de nombreuses études,
  - Ils ont de nombreuses propriétés (ordonnancements optimaux en V, faiblement en V, etc.).
    Note : Le problème d'"optimal timing" est complètement évacué.

L'ordonnancement en JaT : état des lieux

- *Stocker, ça coute !* ou *Produire, ça prend du temps !*

  ⇒ Généralement, prise en compte du critère $\bar{C}$,

  - Revient à minimiser le temps de séjour moyen,



  - Intuitivement, implique une réduction des coûts de stockage (en-cours et matières premières).

Problème	Référence
$1\|d_i = d\ inconnue, nmit\|\alpha\bar{E} + \beta\bar{T} + \gamma nd + \delta\bar{C}$	[Panwalker et al., 1982]
$1\|d_i, nmit\|(1-\alpha)\sum_i (E_i + T_i)^2 + \alpha\bar{C}$	[Dileepan et Sen, 1991]
$1\|d_i\|\alpha\bar{E} + \beta\bar{T} + \delta\bar{C}$	[Fry et al., 1987b]

L'ordonnancement en JaT : état des lieux

- Plus récemment, [Lauff et Werner, 2003] s'intéressent à des problèmes d'atelier simples avec coûts intermédiaires de stockage : $\{F2, O2\}|d_i = d, d\ non\ restrictive|\sum_i |C_i - d| + \sum_i (s_{i,[2]} - C_{i,[1]})$.

  - Montrent que ces deux problèmes sont $\mathcal{NP}$-difficiles,
  - Le terme $\sum_i (s_{i,[2]} - C_{i,[1]})$ est le temps d'attente entre les deux machines.

  Note : Dans le cas d'une machine unique, ce terme peut être assimilé à un temps d'attente entre une date de début souhaité et une date de début réel (*promptness*) $\Rightarrow$ réduction du coût de stockage des matières premières.

L’ordonnancement en JaT : état des lieux

Conclusions de cette première partie :

1. La notion de “Juste-à-Temps“ est prise en compte sous forme de différentes fonctions objectifs,
2. Ces fonctions sont des aggrégations de différentes mesures, souvent conflictuelles (avance/retard principalement),
3. Dans certaines études, une mesure des coûts de stockage est prise en compte dans la fonction objectif.

Quelques états de l’art : [Baker et Scudder, 1990], [Gordon et al., 2002a, 2002b] et [T’kindt et Billaut, 2002].

Ordonnancer en JaT : une approche bicritère

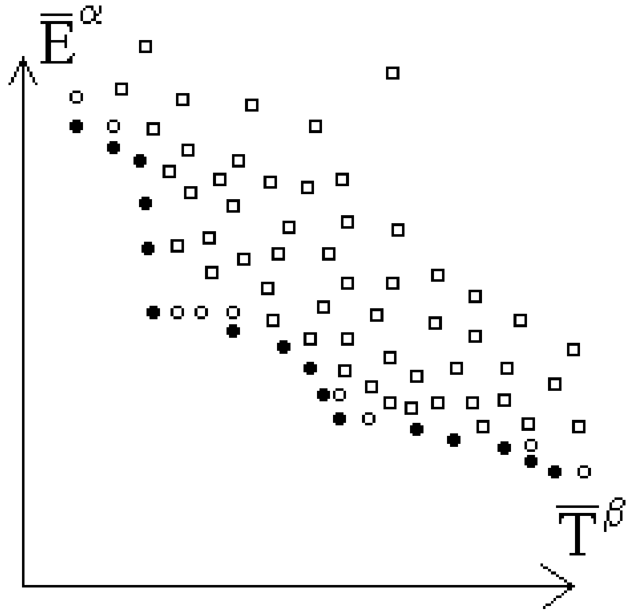
- Par exemple, on minimise les critères $\bar{E}^\alpha$ et $\bar{T}^\beta$.

- On arrive alors dans le “monde” du multicritère,

  Un ordonnancement s est un optimum de Pareto strict ssi $\nexists s'$ tel que $Z_i(s') \leq Z_i(s)$, $\forall i = 1, ..., n$, avec au moins une inégalité stricte.

  $\Rightarrow$ Optima de Pareto faibles,
  $\Rightarrow$ Vecteurs de critères non dominés.

  Un point critique est lié au choix d'une méthode de calcul de ces solutions ([T'kindt et Billaut, 2002]) : approche ϵ-contrainte, combinaison linéaire, ...



Ordonnancer en JaT : une approche bicritère

– Trés peu de travaux dans la littérature JaT,

Problème	Complexité	Référence	Apports
$1 \| d_i, s_i \in [d_i - p_i; d_i], nmit \| \epsilon(L_{max}/P_{max})$	$\mathcal{P}$	[Hoogeveen, 1996]	$O(n \log(n))$
$1 \| d_i, s_i \in [d_i - p_i; d_i] \| \epsilon(L_{max}/P_{max})$	$\mathcal{P}$	[Hoogeveen, 1996]	$O(n^2 \log(n))$
$1 \| d_i, nmit \| Lex(\bar{U}, E_{max})$	$\mathcal{NP}$-difficile	[Guner et al., 1998] [Azizoglu et al., 2003]	PSE (n=20) Heur, LB, Cond
$1 \| d_i, nmit \| \epsilon(\bar{U}/E_{max})$	$\mathcal{NP}$-difficile	[Azizoglu et al., 2003]	Heur, LB, Cond
$1 \| d_i \| \epsilon(\bar{C}/E_{max})$	Fort. $\mathcal{NP}$-dif	[Koktener et Koksalan, 2000] [Koksalan et Burak, 2003]	Heur Heur
$1 \| d_i \| \epsilon(\bar{E}^{\alpha} / \max_i \{\beta_i T_i\})$	$\mathcal{NP}$-difficile	[Tsiushuang et al., 1997]	Heur, PSE (n=30)
$1 \| d_i = d \| \epsilon(\bar{E} + \bar{T}/T_{max})$	Faib. $\mathcal{NP}$-dif.	[Cho et al., 2002]	Heur
$1 \| d_i = d \| \epsilon(\sum_i (E_i + T_i)^2 / T_{max})$	$\mathcal{NP}$-dif.	[Seo et al., 2001]	Res. théo. Exact
$F2 \| d_i, prmu \| \epsilon(C_{max}/E_{max})$	$\mathcal{NP}$-difficile	[Toktas et al., 2003]	Heur, PSE (n=10)

Ordonnancer en JaT : une approche bicritère

- Approche ϵ-contrainte : utilisée pour calculer des optima de Pareto,
  - $\epsilon(Z_1/Z_2) \Leftrightarrow \min Z_1$ sc $Z_2 \leq \epsilon$, où ϵ est fixé,
  - En faisant varier ϵ on peut obtenir un ensemble $W \subseteq WE$ tel que $E \subseteq W$.

- Si Z_2 est un critère régulier de la forme $\max_i(f_i(C_i))$ alors $Z_2 \leq \epsilon \Leftrightarrow \tilde{d}_i = f_i^{-1}(\epsilon), \forall i = 1, ..., n$,

- Si Z_2 est un critère non régulier de la forme $\max_i(f_i(C_i))$ alors $Z_2 \leq \epsilon \Leftrightarrow r_i = f_i^{-1}(\epsilon), \forall i = 1, ..., n$.

Ordonnancer en JaT : une approche bicritère

– Transformations,

Problème originel	Problème à résoudre
$1\|d_i, s_i \in [d_i - p_i; d_i],\ nmit\|\epsilon(L_{max}/\mathbf{P_{max}})$	$1\|d_i, \mathbf{r_i} \in [d_i - p_i - \epsilon; d_i - \epsilon],\ nmit\|L_{max}$
$1\|d_i, s_i \in [d_i - p_i; d_i]\ \|\epsilon(L_{max}/\mathbf{P_{max}})$	$1\|d_i, \mathbf{r_i} \in [d_i - p_i - \epsilon; d_i - \epsilon]\|L_{max}$
$1\|d_i, nmit\|\epsilon(\bar{U}/\mathbf{E_{max}})$	$1\|d_i, \mathbf{r_i} = d_i - \epsilon, nmit\|\bar{U}$
$1\|d_i\|\epsilon(\bar{C}/\mathbf{E_{max}})$	$1\|\mathbf{r_i} = d_i - \epsilon\|\bar{C}$
$1\|d_i\|\epsilon(\bar{E}^\alpha/\max_\mathbf{i}\{\beta_\mathbf{i}\mathbf{T_i}\})$	$1\|d_i, \tilde{\mathbf{d}}_\mathbf{i} = \frac{\epsilon}{\beta_i} + d_i\|\bar{E}^\alpha$
$1\|d_i = d\|\epsilon(\bar{E} + \bar{T}/\mathbf{T_{max}})$	$1\|d_i = d, \tilde{\mathbf{d}}_\mathbf{i} = \epsilon + d\|\bar{E} + \bar{T}$
$1\|d_i = d\|\epsilon(\sum_i (E_i + T_i)^2/\mathbf{T_{max}})$	$1\|d_i = d, \tilde{\mathbf{d}}_\mathbf{i} = \epsilon + d\|\sum_i (E_i + T_i)^2$
$F2\|d_i, prmu\|\epsilon(C_{max}/\mathbf{E_{max}})$	$F2\|\mathbf{r_{i,2}} = d_i - \epsilon - p_{i,2}, prmu\|C_{max}$

Ordonnancer en JaT : une approche bicritère

- Le cas du problème $1|d_i, s_i \in [d_i - p_i; d_i], nmit|\epsilon(L_{max}/P_{max})$ ([Hoogeveen, 1996])
  - Une machine est disponible,
  - n opérations à traiter (p_i, d_i, s_i),
  - "nmit" : pas de temps mort de la date 0 à la date $\sum_i p_i$,
  - Critères : $L_{max} = \max_i(C_i - d_i)$ et $P_{max} = \max_i(s_i - t_i)$,
  - Minimiser L_{max} sachant que $P_{max} \leq \epsilon$.

$\Rightarrow$ On se ramène au problème $1|r_i = s_i - \epsilon, nmit|L_{max}$ ($\mathcal{NP}$-difficile au sens fort).

$\Rightarrow$ Lorsqu'on considère que $s_i \in [d_i - p_i; d_i]$ alors le problème devient polynomial.

Ordonnancer en JaT : une approche bicritère

- Rappels :
  - Le problème $1|nmit|P_{max}$ est optimalement résoluble par la règle "*Minimum Target Start Time*" (MTST),
  - Le problème $1||L_{max}$ est optimalement résoluble par la règle "*Earliest Due Date*" (EDD).

  $\Rightarrow$ Le problème $1|s_i \in [d_i - p_i; d_i], nmit|\epsilon(L_{max}/P_{max})$ est optimalement résoluble en $O(n \log(n))$ par la règle MTST/EDD.

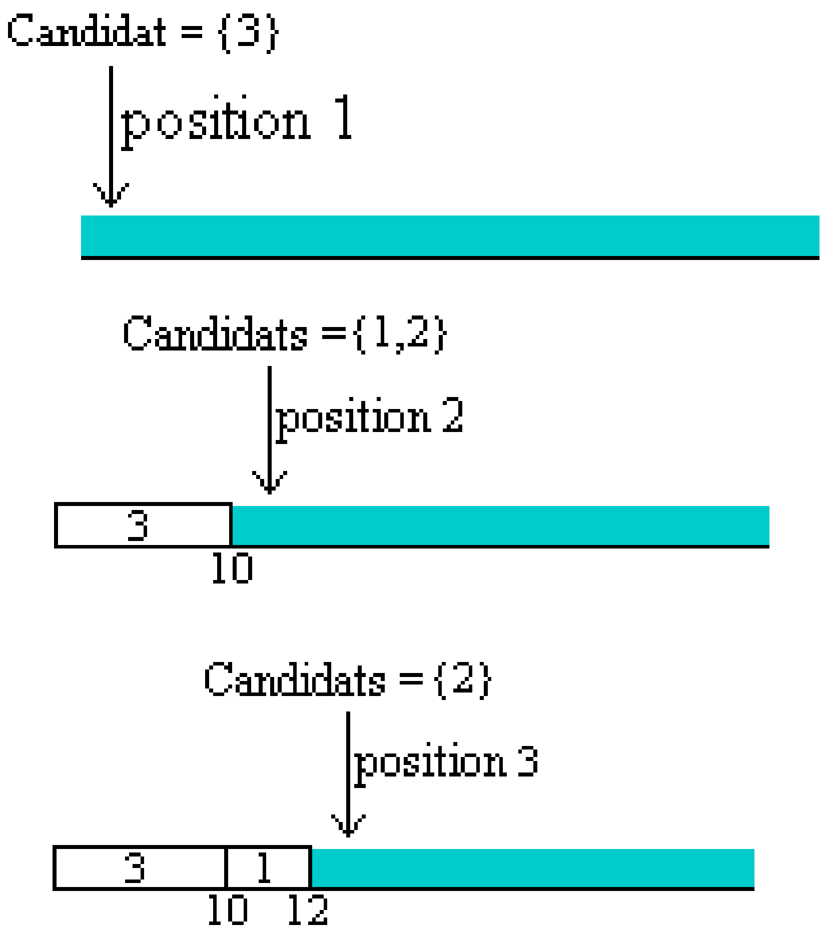
  $\Rightarrow$ Hoogeveen montre même que la règle MTST/EDD est optimale pour un problème un peu plus général :
  Soit $\epsilon_k \leq \epsilon_{k+1}, \forall k = 1, ..., n-1$,
  On pose $r_{[k]} = s_{[k]} - \epsilon_k, \forall k = 1, ..., n-1$.
  La règle MTST/EDD résout optimalement le problème $1|s_i \in [d_i - p_i; d_i], nmit, r_{[k]}|L_{max}$.

Ordonnancer en JaT : une approche bicritère

– Example,

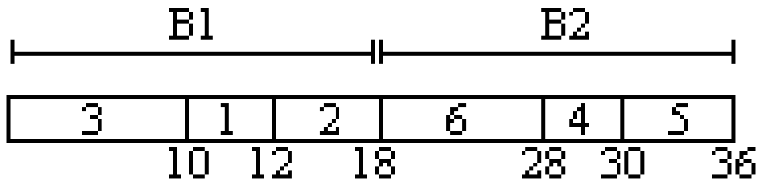
i	1	2	3	4	5	6
s_i	8	4	1	25	21	18
p_i	2	6	10	2	6	10
d_i	9	10	11	26	27	28

On considère $\epsilon = 2$.
$P_{max}(MTST) = 1$ (3,2,1,6,5,4)
$L_{max}(EDD) = 8$ (1,2,3,4,5,6)



Ordonnancer en JaT : une approche bicritère

i	1	2	3	4	5	6
s_i	8	4	1	25	21	18
p_i	2	6	10	2	6	10
d_i	9	10	11	26	27	28



- En partant de la solution du problème $Lex(P_{max}, L_{max})$ on ne réordonnance que les travaux dans chaque bloc (car toutes les "têtes" de bloc sont en EDD),
- Comme on a au plus n blocs (séquence EDD inversée), on a au plus n optima de Pareto stricts,
- Pour passer d'une valeur de ϵ à l'autre, il faut identifier le bloc B dans lequel on a i tq $L_i = L_{max}$ et calculer ϵ' tel que i soit ordonnancé plus tôt.

Gestion de production et ordonnancement JaT

- Ordonnancer en JaT a un sens lorsqu'on *produit en JaT*,

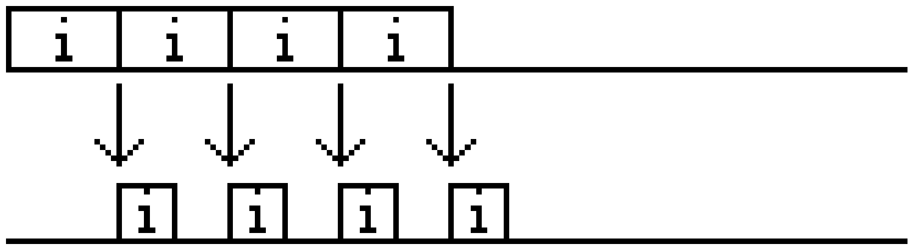
- La litterature en gestion de production est très fourni sur le sujet,

- La gestion de production en JaT fait l'objet de nombreuses descriptions plus générales que l'ordonnancement (dimensionnement de l'atelier, organisation de l'atelier, *etc.*),

- Un certain nombre d'éléments clefs sont à retenir pour l'ordonnancement :
  - Réduire les délais de production (retards de livraison, temps nécessaire pour produire),
  - Faire comme si les machines formaient "une chaine",
  - Réduire les coûts inutiles et réguler au strict nécessaire les en-cours (réduction des coûts de stockage).

Gestion de production et ordonnancement JaT

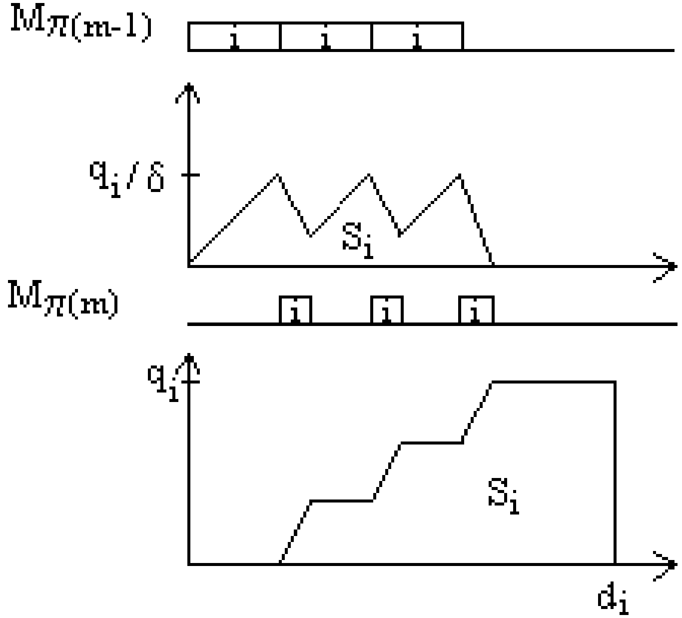
- Nous proposons une nouvelle approche dans le cadre des problèmes d'atelier,
  - m machines,
  - n travaux divisibles $(p_{i,j}, d_i, q_i)$,
  - Chaque travail possède m opérations et suit une gamme $(\pi_i(1), \pi_i(2), ..., \pi_i(m))$.
- Pour réduire le temps de production, on autorise le *lot streaming*,



- Par la suite, on suppose que toutes les opérations sont découpées en δ sous-lots (sous-lots de taille identique pour chaque opération),

Gestion de production et ordonnancement JaT

– Considérons pour un travail i les opérations sur $M_{\pi(m-1)}$ et $M_{\pi(m)}$,



– $S_i = \int_0^\infty f_i(t)dt$ où $f(i)(t)$ est le nombre d'élément de i en stock à l'instant t,
En discrétisant et en notant que $f_i(t) = f_i(t-1) + Prod_i(t) - Conso_i(t)$:
$S_i = \sum_0^\infty f_i^p(t) - \sum_0^\infty f_i^c(t)$,
avec
$f_i^p(t) = \sum_{k=0}^t Prod_i(k)$ et $f_i^c(t) = \sum_{k=0}^t Conso_i(k)$

Gestion de production et ordonnancement JaT

- Notations,
  $\gamma_i^{\pi_i(j)}$: coût unitaire de stockage entre $M_{\pi_i(j)}$ et $M_{\pi_i(j+1)}$,
  $f_{i,j}^p(t)$: nombre d'éléments de i produits dans le stock entre $M_{\pi_i(j)}$ et $M_{\pi_i(j+1)}$,
  κ_i : coût unitaire de stockage dans le stock de produit fini.

- Le coût total de stockage associé au travail i est alors donné par :
  $CS_i = \sum_{j=1}^{m-1} \left(\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}\right) \sum_{t=t_{i,\pi_i(j),1}}^{T} f_{i,\pi_i(j)}^p(t)$
  $-\gamma_i^{\pi_i(m-1)} \sum_{t=t_{i,\pi_i(m),1}}^{T} f_{i,\pi_i(m)}^p(t)$
  $+\kappa_i \sum_{t_{i,\pi_i(m),1}}^{T} f_{i,\pi_i(m)}^p(t)$,
  avec $\gamma_i^{\pi_i(0)} = 0$.

Gestion de production et ordonnancement JaT

– En remplaçant les fonctions $f_{i,j}^{p}$ par leur valeur et en manipulant l'équation de CS_i on obtient :
$CS_i = -\frac{q_i}{\delta}\sum_{j=1}^{m}\left(\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}\right)\sum_{k=1}^{\delta} t_{i,\pi_i(j),k}$
$-\frac{q_i}{2\delta}\sum_{j=1}^{m-1}\left(\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}\right)p_{i,\pi_i(j)} + \gamma_i^{\pi_i(m-1)}\frac{q_i p_{i,\pi_i(m)}}{2\delta}$
$+\kappa_i\frac{q_i p_{i,\pi_i(m)}}{2\delta} + \kappa_i q_i t_{i,\pi_i(m),\delta} + \kappa_i q_i E_i,$
avec $\gamma_i^{\pi_i(m)} = \kappa_i$ et $\gamma_i^{\pi_i(0)} = 0.$

Gestion de production et ordonnancement JaT

- Synthèse : quelle est à la "contribution" du travail i par rapport aux éléments clefs/objectifs du JaT ?

  Réduire les délais de production $\Rightarrow \beta_i T_i$ et lot-streaming $(\delta\lambda_i)$,

  $$\left.\begin{array}{l}\text{Faire comme si les machines formaient "une chaine"} \\ \text{Réduire les coûts inutiles/réguler en-cours}\end{array}\right\} \Rightarrow CS_i$$

- Pour un travail i on introduit Z_i,

  $Z_i = \delta\lambda_i + \beta_i T_i + \kappa_i q_i E_i - \frac{q_i}{\delta} \sum_{j=1}^{m} \left(\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}\right) \sum_{k=1}^{\delta} t_{i,\pi_i(j),k}$
  $-\frac{q_i}{2\delta} \sum_{j=1}^{m-1} \left(\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}\right) p_{i,\pi_i(j)} + \gamma_i^{\pi_i(m-1)} \frac{q_i p_{i,\pi_i(m)}}{2\delta}$
  $+\kappa_i \frac{q_i p_{i,\pi_i(m)}}{2\delta} + \kappa_i q_i t_{i,\pi_i(m),\delta}.$

  <u>Les variables sont :</u>

  $t_{i,j,k}$: dates de début de chaque sous-lot sur chaque machine,

  δ : nombre de sous-lots.

  Le coût Z_i n'est pas linéaire lorsque δ n'est pas fixé.

Gestion de production et ordonnancement JaT

– Le problème défini est multicritère car l'on considère que l'on a n coûts Z_i à minimiser (chaque travail doit être "Juste-à-Temps"),

– Comment calculer un optimum de Pareto ? ... approche paramétrique ([Soland, 1979]),

$$(P) \quad \left\{ \begin{array}{l} \text{Minimiser } \sum_i Z_i \\ \text{sous contrainte} \\ \quad Z_i \leq b_i, \forall i = 1, ..., n \end{array} \right.$$

Gestion de production et ordonnancement JaT

- Particularisation dans le cas 1 machine et produit indivisible ($\delta = 1$),
  $Z_i = \alpha_i E_i + \beta_i T_i$
  Le problème (P) se réduit alors au
  $1|r_i = d_i - p_i - \frac{b_i}{\alpha_i}, \tilde{d}_i = d_i - p_i + \frac{b_i}{\beta_i}| \sum_i \alpha_i E_i + \beta_i T_i$,

- Particularisation dans le cas $F2$ et produit indivisible ($\delta = 1$),
  $Z_i = \alpha_i E_i + \beta_i T_i + \gamma_i(t_{i,2} - t_{i,1})$
  Le problème (P) se réduit alors au
  $F2|\alpha_i E_i + \beta_i T_i + \gamma_i(t_{i,2} - t_{i,1}) \leq b_i| \sum_i \alpha_i E_i + \beta_i T_i + \gamma_i(t_{i,2} - t_{i,1})$,

- Particularisation dans le cas $F2$ et produit divisible,
  $Z_i = \alpha_i E_i + \beta_i T_i + \frac{\gamma_i q_i}{\delta} \sum_{k=1}^{\delta}(t_{i,2,k} - t_{i,1,k}) + \kappa_i q_i(t_{i,2,\delta} - \frac{1}{\delta} \sum_{k=1}^{\delta} t_{i,2,k}) + \delta\lambda_i$

Conclusions

– Ordonnancer en JaT implique bien plus que minimiser $\sum_i \alpha_i E_i + \beta_i T_i$,

– Ces problèmes commencent à être vus comme “au moins bicritère”,

– Nous avons proposé une approche générale pour les problèmes d’ateliers sans affectation,
  – Flowshop, Jobshop, Openshop,
  – Pour des produits divisibles ou indivisibles,
  – Avec ou sans prise en compte de coûts de stockages intermédiaires,
  – ... mais qui suppose que le nombre de sous-lots est identique pour toutes les opérations (et taille identique par opération) !

Bibliographie

[Azizoglu et al., 2003] M. Azizoglu, S. Kondakci et M. Koksalan (2003). Single machine scheduling with maximum earliness and number tardy. *Computers and Industrial Engineering*, 45 :257–268.

[Baker et Scudder, 1990] Baker, K. R. et Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties : a review. *Operations Research*, 38(1) :22–36.

[Chrétienne et Sourd, 2003] P. Chrétienne et F. Sourd (2003). PERT scheduling with convex cost functions, *Theoretical computer science*, 292 :145-164.

[Cho et al, 2002] S. A. Cho, C.B. Kim et D.H. Lee (2002). Single machine MAD/T_{max} problem with a common due date, *Computers and Operations Research*, 29 :1099-1113.

[Davis et Kanet, 1993] J. S. Davis and J. J. Kanet (1993). Single Machine Scheduling with Early and Tardy Completion Costs, *Naval Research Logistics*, 40 :85-101.

[Della Croce et Trubian, 2002] Della Croce, F. et Trubian, M. (2002). Optimal idle time insertion in early-tardy parallel machines scheduling with precedence constraints, *Production Planning and Control*, 13(2) :133-142.

[Dileepan et Sen, 1991] P. Dileepan and T. Sen (1991). Bicriterion jobshop scheduling with total flowtime and sum of squared lateness, *Engineering Costs and Production Economic*, 21 :295-299.

[Fry et al., 1987] T. D. Fry and R. D. Armstrong and R. H. Blackstone (1987). Minimizing Weighted Absolute Deviation in Single Machine Scheduling, *IIE Transactions*, 19(4) :445-450.

[Fry et al., 1987b] T. D. Fry and G. K. Leong and T. R. Rakes (1987). Single machine scheduling :

a comparison of two solution procedures, *Omega*, 15(4) :277-282.

[Garey et al., 1988] M. R. Garey, R. E. Tarjan et G. T. Wilfong (1988). One-processor scheduling with symmetric earliness and tardiness penalties, *Mathematics of Operations Research*, 13(2) :330-348.

[Gordon et al., 2002a] V. Gordon, JM. Proth et C. Chu (2002). A survey of the state-of-the-art of common due date assignment ans scheduling research, *European Journal of Operational Research*, 139(1) :1-25.

[Gordon et al., 2002b] V. Gordon, JM. Proth et C. Chu (2002). Due date assignment and scheduling : SLK, TWK and other due date assignment models, *Production Planning and Control*, 13(2) :117-132.

[James et Buchanan, 1997] R. J. W. James and J. T. Buchanan (1997). A neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem, *European Journal of Operational Research*, 102 :513-527.

[James et Buchanan, 1998] R. J. W. James and J. T. Buchanan (1998). Performance enhancements to tabu search for the early/tardy scheduling problem, *European Journal of Operational Research*, 106 :254-265.

[Koksalan et Burat Kehal, 2003] M. Koksalan et A. Burak Kehal (2003). Using genetic algorithms for single-machine bicriteria scheduling problems, *European Journal of Operational Research*, 45 : 543-556.

[Lauff et Werner, 2003] V. Lauff et F. Werner (2003). On the complexity and some properties of multi-stage scheduling problems with earliness and tardiness penalties, *Computers and Operations Research*, 31 :317-345.

[Liaw, 1999] C.-F. Liaw (1999). A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem, *Computers and Operations Research*, 26 :679-693.

[Panwalker et al., 1982] S. S. Panwalker and M. L. Smith and A. Seidmann (1982). Common due date assignment to minimize total penalty for the one machine scheduling problem, *Operations Research*, 30(2) :391-399.

[Seidmann et al., 1981] A. Seidmann and S. S. Panwalker and M. L. Smith (1981). Optimal assignment of due-dates for a single processor scheduling problem, *International Journal of Production Research*, 19(4) :393-399.

[Seo et al., 2001] J.H. Seo, C.B. Kim et D.H. Lee (2001). Minimizing mean squared deviation of completion times with maximum tardiness constraint, *European Journal of Operational Research*, 129 :95-104.

[Soland, 1979] R. M. Soland (1979). Multicriteria optimization : a general characterization of efficient solutions, *Decision Science*, 10 :27-38.

[Sourd et Kedad-Sidhoum, 2003] F. Sourd et S. Kedad-Sidhoum (2003). The one-machine problem with earliness and tardiness penalties, *Journal of Scheduling*, 6 :533-549.

[Szwarc et Mukhopadhyay, 1995] W. Szwarc and S. K. Mukhopadhyay(1995). Optimal Timing Schedules in Earliness-Tardiness Single Machine Sequencing, *Naval Research Logistics*, 42 :1109-1114.

[T'kindt et Billaut, 2002] V. T'kindt et J.-C. Billaut (2002). Multicriteria Scheduling : Theory, Models and Algorithms, *Springer-Verlag*, Heidelberg.

[Tsiushuang et al., 1997] C. Tsiushuang, Q. Xiangtong et T. Fengsheng (1997). Single machine scheduling to minimize weighted earliness subject to maximum tardiness, *Computers and Operations Research*, 24(2) :147-152.

[Wennink, 1995] M. Wennink (1995). Algorithmic Support for Automated Planning Boards, *Ph. D. thesis, Eindhoven University of Technology*, Pays-Bas.