Résolution d'un problème de Job-Shop intégrant des contraintes de Ressources Humaines

ROADEF'09, 10 - 12 février 2009, Nancy (France)

O. Guyon[1,2], P. Lemaire[2], É. Pinson[1] et D. Rivreau[1]

[1] LISA - Institut de Mathématiques Appliquées
[2] IRCCyN, CNRS; École des Mines de Nantes

10 février 2009

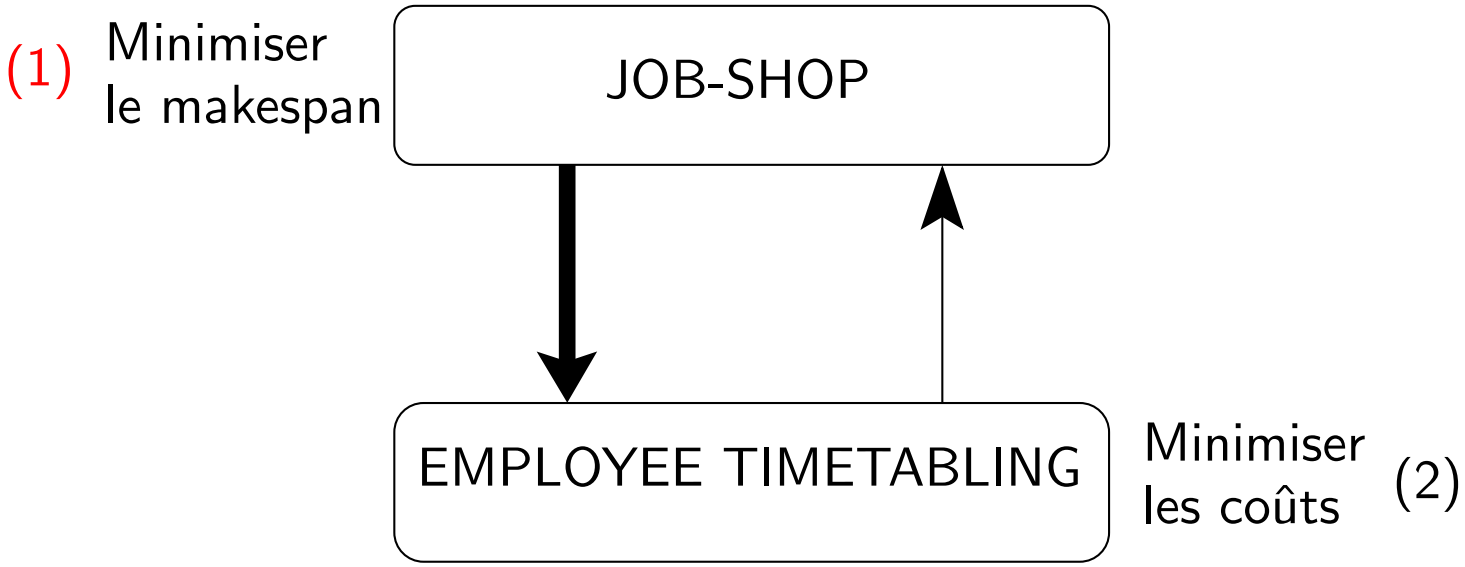
Sommaire

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Présentation du problème (1/3)



Présentation du problème (2/3)

Job-Shop : Produire un ensemble J de n jobs sur m machines

- $\forall j \in J$ $\{O_{ji}\}_{i=1..m}$ séquence d'opérations du job j
  - machine $m_{ji} \in \{1 \dots m\}$
  - durée p_{ji}
    $\hookrightarrow$ on notera ρ_{jk} la durée d'exécution de j sur la machine k
  - non interruptible
  - requiert un employé qualifié pour l'utilisation de m_{ji}

Planification d'emploi du temps d'un ensemble E de μ employés

- A_e Ensemble des machines que e maîtrise
- $\mathcal{T}_e$ Ensemble de tranches horaires (shifts) où e est disponible

Présentation du problème (3/3)

Temps

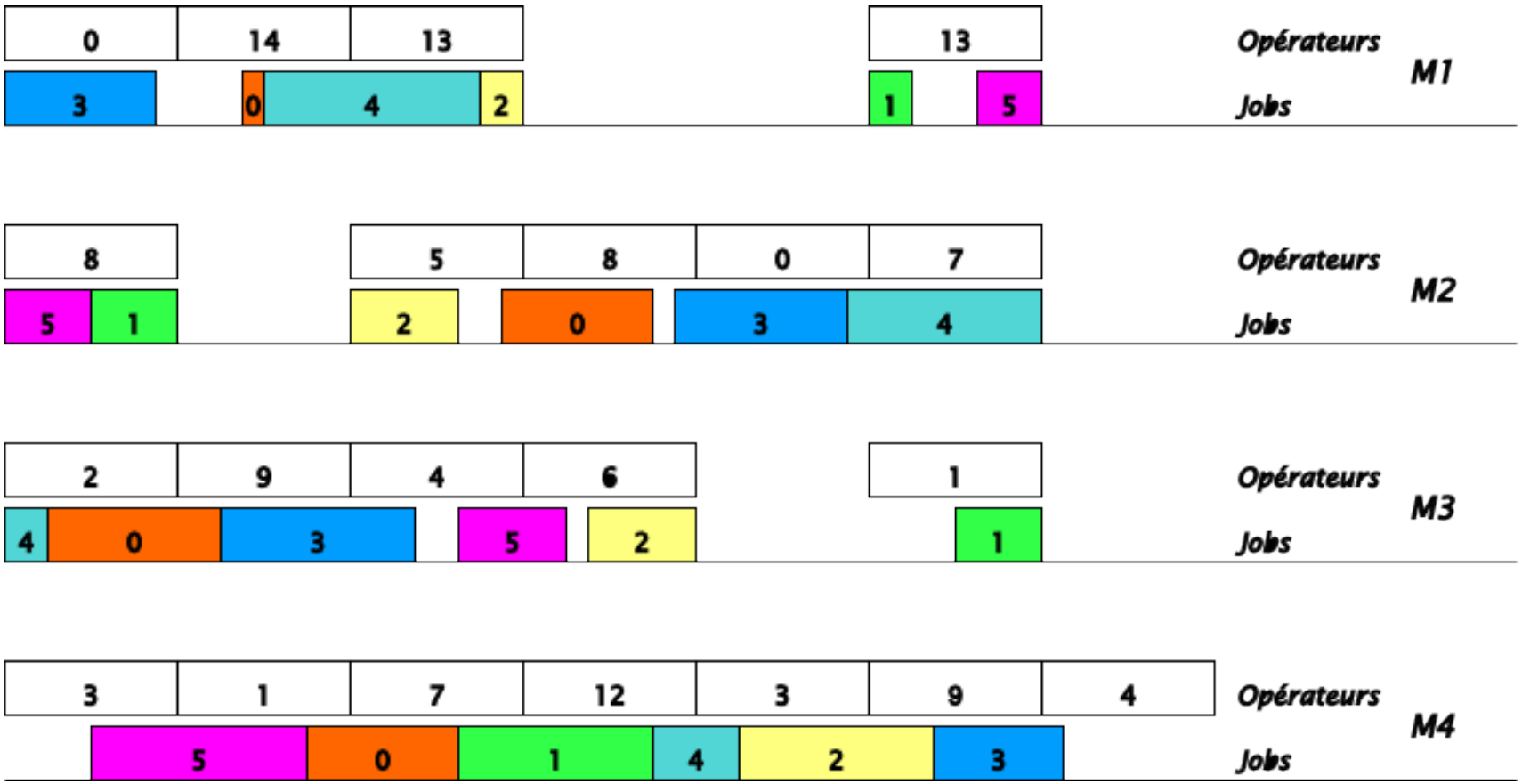
- Horizon de planification $H = \delta * \pi$ avec :
  - δ Nombre de tranches horaires (shift s)
  - π durée d'un shift

Nb : Formalisation permet la modélisation de travail en 3*8

Objectif

1. Minimiser le makespan *Cmax* du Job-Shop
2. Affecter au moindre coût un employé sur chaque couple (machine, shift) afin de couvrir les ressources du Job-Shop

Exemple (6 jobs - 4 machines - 15 employés)



Motivations (1/3) : Prolongement d'un travail

Prolongement d'un travail

- Guyon, Lemaire, Pinson et Rivreau.
  European Journal of Operational Research (à paraître)
- Couplage d'un problème de planification d'agents et d'ordonnancement de production
- *Méthode de résolution* :
  $\hookrightarrow$ Technique de décomposition et de génération de coupes
- $\rightsquigarrow$ Problème d'ordonnancement traité simple
- Motivation :
  $\hookrightarrow$ La décomposition et génération de coupes est-elle aussi efficace avec un problème d'ordonnancement + complexe (Job-Shop) ?

Motivations (2/3) : Confrontation à un cas industriel

Fabrication de tubes en zirconium

- Différents tubes passent, selon un ordre déterminé, sur différentes machines (*Job-Shop*) ...
- ... devant l'oeil expert d'un employé capable de détecter une anomalie (*Planification d'agents*)
- $\hookrightarrow$ Processus qualité de requalification des employés

Motivations (3/3) : Confrontation à la littérature existante

Etude d'un cas traité dans la littérature

- Artigues, Gendreau, Rousseau et Vergnaud.
  Computers and Operations Research (2008)
- Motivation : Tester méthode hybride RO-PPC sur des problèmes couplant job-shop - planification d'agents
- *Méthode de résolution* :
  $\hookrightarrow$ Résolution via un solveur de PPC avec une contrainte globale additive correspondant à la relaxation continue du problème de planification d'agents
- *Notre étude* : cas spécifique de mapping activités - machines
  $\hookrightarrow$ 8 instances de comparaison

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Variables de décision

Job-Shop : Affectation des jobs aux couples (machine, unité de temps)

$$\forall j = 1 \dots n, k = 1 \dots m, t = 0 \dots H, y_{jkt} = \begin{cases} 1 & \text{si } j \text{ débute sur } k \text{ en } t \\ 0 & \text{sinon} \end{cases}$$

Planification d'agents : Affectation des employés aux couples (machine, shift)

$$\forall e = 1 \dots \mu, k = 1 \dots m, s = 1 \dots \delta, x_{eks} = \begin{cases} 1 & \text{si } e \text{ travaille sur } k \text{ durant } s \\ 0 & \text{sinon} \end{cases}$$

Programme Linéaire

$$\min Lex\left(C_{max}, \sum_{e=1}^{\mu}\sum_{k\in A_e}\sum_{s\in\mathcal{T}_e} c_{eks}\cdot x_{eks}\right)$$

$$\sum_{t=0}^{H} t\cdot y_{ikt} + \rho_{ik} \leq C_{max} \qquad i=1,..,n \quad k=m_{im}$$

$$\sum_{t=0}^{H} y_{ikt} = 1 \qquad i=1,..,n \quad k=1,..,m$$

$$\sum_{u=\rho_{ik}}^{t} y_{ilu} - \sum_{u=0}^{t-\rho_{ik}} y_{iku} \leq 0 \qquad i=1,..,n \quad j=1,..,m-1$$

$$k=m_{ij} \quad l=m_{i(j+1)}$$

$$t=\rho_{ik},..,H-\rho_{il}$$

$$\sum_{i=1}^{n}\sum_{u=t-\rho_{ik}+1}^{t} y_{iku} \leq \min(1, \sum_{e=1}^{\mu} x_{eks}) \qquad k=1,..,m \quad t=0,..,H \quad s=\lfloor t/\pi \rfloor$$

$$\sum_{k\notin A_e}\sum_{s=0}^{\sigma} x_{eks} = 0 \qquad e=1,..,\mu$$

$$\sum_{k\in A_e}\sum_{s\notin\mathcal{T}_e} x_{eks} = 0 \qquad e=1,..,\mu$$

$$\sum_{k\in A_e}(x_{eks} + x_{ek(s+1)} + x_{ek(s+2)}) \leq 1 \qquad e=1,..,\mu \quad s=0,..,\sigma-2$$

$$y_{jkt} \in \{0,1\} \qquad j=1,..,n \quad k=1,..,m \quad t=0,..,H$$

$$x_{eks} \in \{0,1\} \qquad e=1,..,\mu \quad k=1,..,m \quad s=0,..,\delta$$

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Techniques de résolution (1/2)

Minimisation du makespan *Cmax*

- Observation - sur les instances testées - :
  Cmax sans ressources = *Cmax* avec ressources
- $\hookrightarrow$ Concentration sur le problème de minimisation des coûts
- À noter : Problème lexicographique peut se résoudre par recherche dichotomique sur *Cmax*

Techniques de résolution (2/2)

Minimisation des coûts salariaux

- MIP
- Décomposition et génération de coupes de réalisabilité
- Branch and Bound sur le nombre minimal de shifts travaillés par machine
- Branch and Bound 0-1 sur le travail (ou non) en chaque couple (machine, shift)

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Principe

Idées directrices

- Décomposition de $[P]$ (en relaxant les contraintes couplantes)
  - $[Job - Shop]$ (contraintes liées aux variables y_{jkt})
  - $[Employe]$ (contraintes liées aux variables x_{eks})
- Fixer une distribution de couples (machine, shift) travaillés et, par symétrie non travaillés
- Tester (à différents niveaux) sa réalisabilité sur
  - $[Job - Shop]$ via un solveur dédié efficace
  - $[Employe]$ via un solveur de PLNE
- Pour éviter exploration exhaustive :
  $\hookrightarrow$ Branch and Bound 0-1, avec génération de coupes
  - initiales
  - de réalisabilité (en cours d'exploration)

Plan

1 Introduction
- Problème
- Formalisation MIP indexée sur le temps
- Techniques de résolution

2 Branch and Bound 0-1
- Principe
- Coupes initiales
- Arborescence

3 Conclusion
- Expérimentations numériques
- Bilan

Coupes initiales de probing

Coupes de probing

Tester obligation de travail pour tout couple (machine k, shift s)

Comment ?

- Créer un job fictif j_f de durée π s'exécutant sur k durant s
- Résoudre [Job-Shop], via un solveur de Job-Shop dédié
- Si [Job-Shop] non réalisable ($\Leftrightarrow$ ne respecte pas $Cmax^*$) :
  - j_f ($\rightarrow$ absence de ressource sur (k, s)) non ordonnançable
  - Un employé travaillera sur (k, s) dans toute solution optimale

Résultat

- Sous-ensembe de couples (k, s) déjà arbitrés
- Coupes sur $[Employe] \Rightarrow \sum_{e=1}^{\mu} x_{eks} = 1 \ \forall \ (k, s)$ arbitré

Coupes initiales de capacité

Coupes de capacité sur machine k

Déterminer un nombre minimal de shifts b_k travaillés sur k

Comment ?

1. Somme des couples (k, s) arbitrés par probing sur k
2. $\left\lceil \frac{\sum_{j=1}^{n} \rho_{jk}}{\pi} \right\rceil$
3. Test (par probing) du nombre maximal valide de jobs fictifs

Résultat

Coupes sur $[Employe] \Rightarrow \sum_{e=1}^{\mu} \sum_{s \in \mathcal{T}_e} x_{eks} \geq b_k,\ k = 1, .., m$

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Caractéristiques (1/4)

Schéma de séparation binaire

- Fils gauche : couple(machine k, shift s) non travaillé
- Fils droit : couple(machine k, shift s) travaillé

Stratégie d'exploration

Profondeur d'abord

Choix de la variable de branchement

- $\bar{k}$: Machine dont l'écart entre le nombre de shifts imposés au travail et le nombre minimal de shifts à travailler est minimal
- $\bar{s}$: 1er shift depuis δ^* tel que $(\bar{k}, \bar{s})$ est non arbitré

Caractéristiques (2/4)

Définitions

- *Distribution relaxée* (moins contraignante)
  Distribution de couples (k, s) tels que chaque (k, s) arbitré (par arborescence ou probing) est fixé à sa valeur et tous les autres sont libres
- *Distribution stricte* (plus contraignante)
  Distribution de couples (k, s) tels que chaque (k, s) arbitré (par arborescence ou probing) est fixé à sa valeur et tous les autres sont fixés à 0

Evaluation

Coût engendré dans [*Employe*] par la *distribution relaxée*

Caractéristiques (3/4)

Implications

Si décision : $(\bar{k}, \bar{s}) = 0$, probing sur chaque couple (k, s) non arbitré

Règles d'élimination

On peut pruner le noeud en cours si l'une des 2 conditions suivantes est vérifiée :

1. [*Employe*] avec *distribution relaxée trop cher* ou non réalisable (évaluation)
2. *Distribution relaxée* irréalisable pour $[Job - Shop]$

Caractéristiques (4/4)

En cas de non élimination d'un noeud

Calcul de $(\bar{x}^*, \bar{z}^*)$ pour [*Employe*] respectant *distribution stricte*

- Si $\exists \bar{x}^*$, Test *distribution stricte* pour $[Job - Shop]$
  - Si *distribution stricte* valide, $UB = \bar{z}^*$
  - Génération d'une coupe d'irréalisabilité dans [*Employe*]

Coupe d'irréalisabilité dans [Employe]

- Interdire les solutions équivalentes à $\bar{x}^*$
- Forcer le travail sur un couple (k, s) non travaillé pour *distribution stricte*
- $$\sum_{(k,s)\text{non arbitré}} \sum_{e=1}^{\mu} x_{eks} \geq 1$$

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Données

8 instances d'Artigues

Instance	n	m	μ	μ_{extra}	δ	π
ejs	6	4	25	10	8	8
$ejs8*8$	8	8	40	20	8	8
$ejs10*10$	10	10	50	20	10	10

Outils

- Langage de développement : Java
- Solveur PLNE pour [*Employe*] : Ilog Cplex 9.1
- Solveur de Job-Shop dédié : Branch and Bound (*Carlier, Péridy, Pinson, Rivreau*)
- Processeur Intel Core 2 Quad Q6600 @ 2,40 GHz - 3 GB RAM
- Temps de résolution : 600 secondes

Résultats : de l'intérêt des coupes initiales ...

		MIP		MIP avec coupes initiales	
	Coût*	Coût	$\mathrm{Tps}_{(\#\mathrm{noeuds})}$	Coût	$\mathrm{Tps}_{(\#\mathrm{noeuds})}$
*ejs*4	23	23	$1.0_{(0)}$	23	$0.3_{(0)}$
*ejs*9	24	24	$32.8_{(1360)}$	24	$2.8_{(48)}$
*ejs*10	23	23	$1.4_{(0)}$	23	$0.5_{(0)}$
$ejs8 * 8_1$	78	X	$TL_{(2675)}$	87	$TL_{(8971)}$
$ejs8 * 8_2$	96	X	$TL_{(1798)}$	96	$42.8_{(5448)}$
$ejs8 * 8_3$	83	106	$TL_{(2724)}$	83	$221.4_{(3503)}$
$ejs10 * 10_1$	124	X	$TL_{(146)}$	130	$TL_{(497)}$
$ejs10 * 10_3$	95	X	$TL_{(99)}$	X	$TL_{(213)}$

Résultats : de l'intérêt du Branch and Bound ...

		MIP avec coupes initiales		Branch and Bound	
	Coût*	Coût	$\mathrm{Tps}_{(\#\mathrm{noeuds})}$	Coût	$\mathrm{Tps}_{(\#\mathrm{noeuds})}$
*ejs*4	23	23	$0.3_{(0)}$	23	$0.0_{(3)}$
*ejs*9	24	24	$2.8_{(48)}$	24	$1.2_{(89)}$
*ejs*10	23	23	$0.5_{(0)}$	23	$0.1_{(13)}$
$ejs8*8_1$	78	87	$TL_{(8971)}$	78	$52.9_{(103)}$
$ejs8*8_2$	96	96	$42.8_{(5448)}$	96	$41.9_{(211)}$
$ejs8*8_3$	83	83	$221.4_{(3503)}$	83	$3.2_{(33)}$
$ejs10*10_1$	124	130	$TL_{(497)}$	124	$30.0_{(21)}$
$ejs10*10_3$	95	X	$TL_{(213)}$	96	$TL_{(209)}$

Plan

1. Introduction
   - Problème
   - Formalisation MIP indexée sur le temps
   - Techniques de résolution
2. Branch and Bound 0-1
   - Principe
   - Coupes initiales
   - Arborescence
3. Conclusion
   - Expérimentations numériques
   - Bilan

Bilan

- Probing performant
- Technique de décomposition intéressante
- Approche par Branch and Bound très compétitive