

## Heuristiques pour le meilleur des cas dans un ordonnancement de groupes

Guillaume Pinot Nasser Mebarki

IRCCyN — UMR CNRS 6597  
Nantes, France  
prenom.nom@ircryn.ec-nantes.fr

MOSIM 2008



## Table des Matières

- 1 Introduction
- 2 Ordonnement de groupes
- 3 Bornes inférieures
- 4 Heuristiques
- 5 Conclusion



## Introduction

L'ordonnement de groupes permet d'introduire une flexibilité séquentielle importante tout en garantissant une certaine qualité dans le pire des cas.

Une évaluation du meilleur des cas d'un ordonnancement de groupes pourrait également être utile :

- description plus complète de l'ordonnement de groupes dans sa globalité ;
- son utilisation dans un outil d'aide à la décision en temps réel basée sur l'ordonnement de groupes apporterait plus d'information au décideur.



## Ordonnement de groupes

L'ordonnement de groupes fut créé au LAAS-CNRS pour obtenir de la flexibilité séquentielle durant l'exécution de l'ordonnement tout en assurant une certaine qualité. Pour une description complète de la méthode : [Esswein, 2003, Esswein et al., 2004, Artigues et al., 2005]. Pour générer de la flexibilité séquentielle, cette méthode utilise des « groupes d'opérations permutable ».



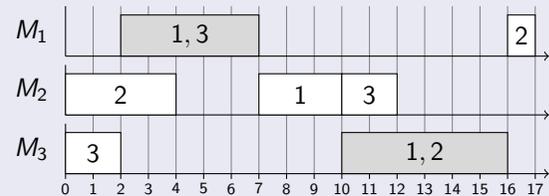
## Exemple : un problème de *job shop*

$i$  représente un travail,  $j$  une opération,  $M_{i,j}$  la machine requise par opération  $j$  du travail  $i$ , et  $p_{i,j}$  le temps requis par l'opération  $j$  du travail  $i$ .

### Problème

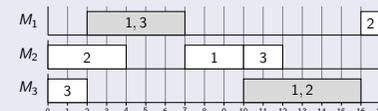
$i$	$j$	$M_{i,j}$	$p_{i,j}$
1	1	1	3
1	2	2	3
1	3	3	3
2	1	2	4
2	2	3	3
2	3	1	1
3	1	3	2
3	2	1	2
3	3	2	2

### Une Solution

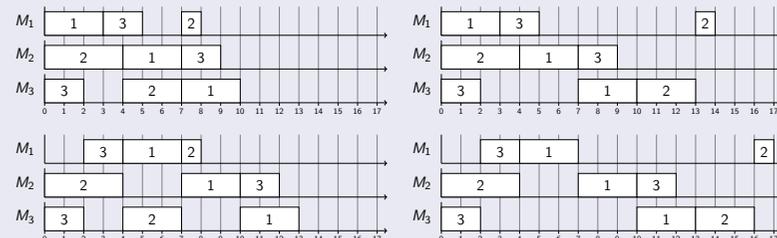


## Exécution de l'exemple

### L'Ordonnement de groupes



### Les Ordonnements semi-actifs correspondants



## Pourquoi l'ordonnement de groupes est-il intéressant ?

Pourquoi l'ordonnement de groupes est-il intéressant ?

- méthode prédictive réactive ;
- flexibilité sur les séquences ;
- méthode bien étudiée durant les 30 dernières années : [Erschler and Roubellat, 1989, Billaut and Roubellat, 1996, Wu et al., 1999, Artigues et al., 2005] ;
- les incertitudes ne doivent pas être modélisées ;
- méthode permettant de pallier certaines incertitudes : [Wu et al., 1999, Esswein, 2003, Pinot et al., 2007] ;
- évaluation de l'ordonnement dans le pire des cas en temps polynomial pour les objectifs de type *minmax* comme le  $C_{\max}$  et le  $L_{\max}$ .

Une évaluation du meilleur des cas d'un ordonnancement de groupes pourrait également être utile.



## Borne inférieure sur la date de fin des opérations

- $\theta_i$  Borne inférieure de la date de début d'une opération ;
- $\chi_i$  borne inférieure de la date de fin d'une opération ;
- $\gamma_k$  borne inférieure de la date de fin d'un groupe.

$$\begin{cases} \theta_i = \max(r_i, \gamma_{g^-(i)}, \max_{j \in \Gamma^-(i)} \chi_j) \\ \chi_i = \theta_i + p_i \\ \gamma_k = C_{\max} \text{ de } |r_i| C_{\max}, \forall O_i \in G_k, r_i = \theta_i \end{cases}$$

Utilisation directe pour des bornes inférieures :

$$LB(L_{\max}) = \max_{\forall O_i} L_i(\chi_i) = \max_{\forall O_i} (\chi_i - d_i)$$

$$LB(C_{\max}) = \max_{\forall G_k} \gamma_k$$



## Borne inférieure pour le *makespan*

Borne inférieure classique du *job-shop* : relaxation en *one-machine problem* [Carlier, 1982] sur chaque machine [Carlier and Pinson, 1989].

Pour faire cette relaxation, pour chaque opération, il nous faut :

- une borne inférieure de la date de début au plus tôt (*head*) :  $\theta_i$  est un bon candidat ;
- une borne inférieure de la durée de latence (*tail*) : un  $\theta_i$  « renversé » est un bon candidat (appelée  $\theta'_i$ ).

La relaxation se fait au niveau des groupes plutôt que des machines (plus de sous problème, mais de taille plus petite).  
Le *one-machine problem* est résolue par une méthode exacte appelée algorithme de Carlier [Carlier, 1982]



## Les Heuristiques

- Règle de priorité :
  - *most work remaining* (MWR) ;
  - *Square tail* (SQUTAIL) :  $\min_{\forall O_i} p_i - \theta_i'^2$  ;
- Règle de priorité couplée avec la borne inférieure : LB+MWR et LB+SQUTAIL ;
- Adaptation du *shifting bottleneck* : SB.



## Le *shifting bottleneck* I

Le *shifting bottleneck*  
[Adams et al., 1988]

- Commencer avec un problème de job shop.
- À chaque itération, la machine goulot est sélectionnée, l'ordre d'exécution des opérations sur cette machine est alors fixé.
- Les machines déjà séquencées sont ensuite réoptimisées.
- Cette procédure est répétée jusqu'à ce que toutes les machines soient séquencées.

Adaptation du *shifting bottleneck* pour l'ordonnement de groupes :

- Commencer avec un ordonnancement de groupes.
- Même procédure, mais au niveau des groupes plutôt que des machines.



## Le *shifting bottleneck* II

Pour le choix de la machine (ou du groupes) à séquencer, ainsi que pour son séquencement, la relaxation en *one-machine problem* est utilisée. L'algorithme de Carlier [Carlier, 1982] est utilisé pour séquencer les machines.

Remarques :

- Comme le nombre de groupes est plus important que le nombre de machines, le nombre de réoptimisations est plus important pour l'ordonnement de groupes.
- Le *shifting bottleneck* original peut engendrer des ordonnancements non réalisables, ce qui n'est pas le cas de notre adaptation.

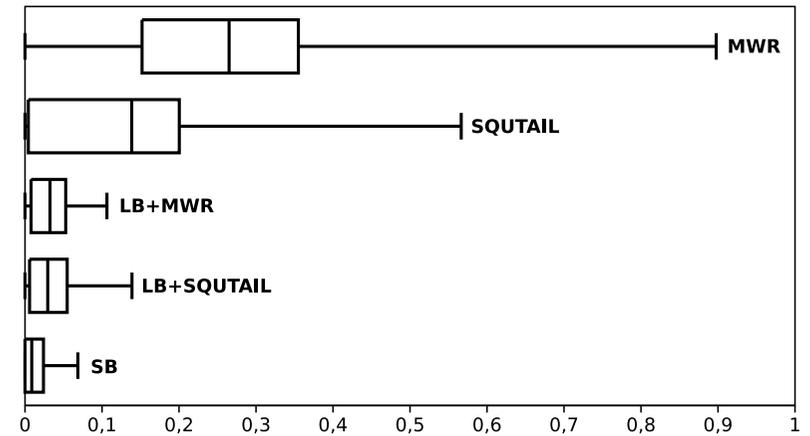


## Résultats I

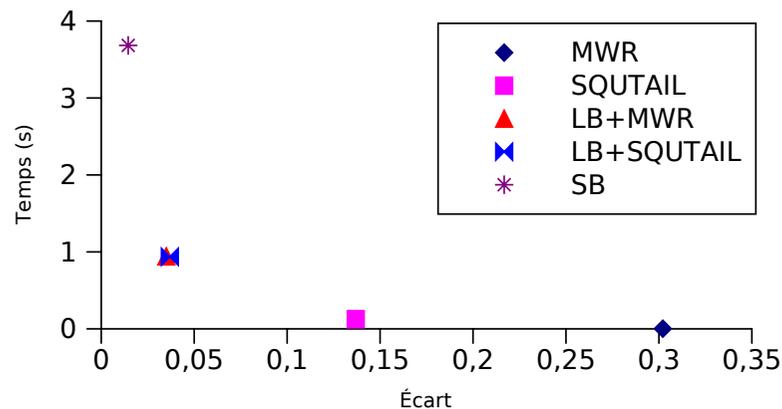
Instances : 1a01 à 1a40 de [Lawrence, 1984].  
 Pour chaque instance, nous générons un ordonnancement de groupes avec une qualité optimale connue (en utilisant l'algorithme décrit dans [Brucker et al., 1994]) et une très grande flexibilité (en utilisant EBJG décrit dans [Esswein, 2003]).



## Résultats II



## Résultats III



## Conclusion

Nous avons proposé des heuristiques pour résoudre le meilleur des cas dans un ordonnancement de groupes. Ces heuristiques apportent différents compromis entre la qualité et le temps de calcul.

Une méthode exacte permettrait de compléter la résolution du meilleur des cas dans un ordonnancement de groupes.



## Bibliographie I

-  Adams, J., Balas, E., and Zawack, D. (1988).  
The shifting bottleneck procedure for job shop scheduling.  
*Management Science*, 34(3) :391–401.
-  Artigues, C., Billaut, J.-C., and Esswein, C. (2005).  
Maximization of solution flexibility for robust shop scheduling.  
*European Journal of Operational Research*, 165(2) :314–328.
-  Billaut, J.-C. and Roubellat, F. (1996).  
A new method for workshop real-time scheduling.  
*International Journal of Production Research*,  
34(6) :1555–1579.



## Bibliographie II

-  Brucker, P., Jurisch, B., and Sievers, B. (1994).  
A branch and bound algorithm for the job-shop scheduling  
problem.  
*Discrete Applied Mathematics*, 49(1-3) :107–127.
-  Carlier, J. (1982).  
The one-machine sequencing problem.  
*European Journal of Operational Research*, 11(1) :42–47.
-  Carlier, J. and Pinson, E. (1989).  
An algorithm for solving the job-shop problem.  
*Management Science*, 35(2) :164–176.



## Bibliographie III

-  Erschler, J. and Roubellat, F. (1989).  
An approach for real time scheduling for activities with time  
and resource constraints.  
In Slowinski, R. and Weglarz, J., editors, *Advances in project  
scheduling*. Elsevier.
-  Esswein, C. (2003).  
*Un apport de flexibilité séquentielle pour l'ordonnancement  
robuste*.  
Thèse de doctorat, Université François Rabelais Tours.



## Bibliographie IV

-  Esswein, C., Billaut, J.-C., and Artigues, C. (2004).  
Ordonnancement de groupes : une approche multicritère pour  
un apport de flexibilité séquentielle.  
In Billaut, J.-C., Moukrim, A., and Sanlaville, E., editors,  
*Flexibilité et robustesse en ordonnancement*, *Traité IC2*, pages  
219–241. Hermes Science, Paris.
-  Lawrence, S. (1984).  
Resource constrained project scheduling : an experimental  
investigation of heuristic scheduling techniques (supplement).  
Technical report, Graduate School of Industrial Administration,  
Carnegie-Mellon University, Pittsburgh, Pennsylvania.



## Bibliographie V

-  Pinot, G., Cardin, O., and Mebarki, N. (2007).  
A study on the group sequencing method in regards with transportation in an industrial FMS.  
*In Proceedings of the IEEE SMC 2007 International Conference.*
-  Wu, S. D., Byeon, E.-S., and Storer, R. H. (1999).  
A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness.  
*Operations Research*, 47(1) :113–124.

