

## Le retard algébrique maximum dans un *job shop*

Guillaume Pinot

LINA — UMR CNRS 6241  
Nantes, France  
guillaume.pinot@univ-nantes.fr

ROADEF 2010

## Table des matières

- 1 Transformation d'un  $J||L_{\max}$  en  $J||C_{\max}$
- 2 Expérimentations
- 3 Discussion

## Introduction

$J||C_{\max}$  très étudié : [Jain and Meeran, 1998]

$J||L_{\max}$  beaucoup moins :

[Demirkol et al., 1997] : adaptation du *shifting bottleneck* ;

[Su et al., 1998] : une heuristique et méthode exacte aux performances limitées.

Minimisation des retards pourtant très important en pratique.

Proposition d'une transformation très simple d'un problème

$J||L_{\max}$  en un problème  $J||C_{\max}$ .

## Intuition

Le *makespan* :

$$C_{\max} = \max_i C_i$$

Le retard algébrique maximum :

$$L_{\max} = \max_i L_i = \max_i (C_i - d_i)$$

Transformer un  $J||C_{\max}$  en  $J||L_{\max}$  :  $d_i = 0$ .

Transformer un  $J||L_{\max}$  en  $J||C_{\max}$  : modéliser le  $d_i$  avec une opération fictive.

# Transformation I

Soit  $P$  un problème  $J||L_{\max}$ .

Soit  $M = 1 + \max_{\forall j} d_j$ .

À partir du problème  $P$ , un problème  $P'$  de type  $J||C_{\max}$  est généré :

- Copie de  $P$  en  $P'$ .
- Pour chaque  $J_i$  de  $P$ , une machine fictive  $M_{J_i}$  est rajoutée à  $P'$ .
- Une opération fictive  $O_{i,k_i}^{P'}$  est rajoutés à  $P'$  à la fin de chaque job sur la machine fictive correspondante avec  $p_{i,k_i}^{P'} = M - d_i \geq 1$ .

Par construction,  $O_{i,k_i}^{P'}$  correspond à  $O_{i,k_i-1}^P$ .

# Transformation II

## Proposition

La résolution du problème  $P'$  est équivalente à la résolution du problème  $P$ . La solution du problème  $P$  correspond à la solution du problème  $P'$  privée des machines et opérations fictives.

## Corollaire

Le  $L_{\max}$  du problème  $P$  peut se calculer à partir du  $C_{\max}$  du problème  $P'$  de la manière suivante :  $L_{\max}^P = C_{\max}^{P'} - M$ .

# Exemple

TABLE:  $P$

$i$	$j$	$m_{i,j}$	$p_{i,j}$	$d_i$
1	1	$M_1$	1	
1	2	$M_2$	2	3
2	1	$M_2$	1	
2	2	$M_1$	2	6

$$M = 1 + \max_i d_i = 1 + \max(3, 6) = 7$$

TABLE:  $P'$ , la transformation de  $P$

$i$	$j$	$m_{i,j}$	$p_{i,j}$
1	1	$M_1$	1
1	2	$M_2$	2
1	3	$M_{J_1}$	4
2	1	$M_2$	1
2	2	$M_1$	2
2	3	$M_{J_2}$	1

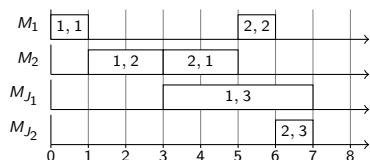
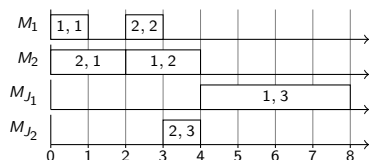


FIGURE: Ordonnancement actifs de  $P'$

# Preuve

Comme la machine de  $O_{i,k_i}^{P'}$  ne possède qu'une opération, cette opération commencera tout de suite après  $O_{i,k_i-1}^{P'}$ . Nous avons donc :

$$C_i^{P'} = C_{i,k_i}^{P'} = C_{i,k_i-1}^{P'} + p_{i,k_i} = C_{i,k_i}^P + M - d_i = C_i^P - d_i + M = L_i^P + M$$

Le makespan du problème  $P'$  est donc :

$$C_{\max}^{P'} = \max_{\forall J_i^{P'}} C_i^{P'} = \max_{\forall J_i^P} (L_i^P + M) = M + \max_{\forall J_i^P} L_i^P = M + L_{\max}^P$$

Comme  $M$  est une constante, optimiser  $C_{\max}^{P'}$  revient à optimiser  $L_{\max}^P$ . □

## Protocole

Instance de  $J||L_{\max}$  : petites instances (20 jobs, 15 machines) de [Demirkol et al., 1998]

Méthode exacte : [Brucker et al., 1994]

Temps de calcul : 1 heure maximum sur un Intel Core 2 Duo à 3 GHz (E8400)

## Récapitulatif

Cette transformation a permis :

- de résoudre optimalement des instances ouvertes avec des outils existants ;
- de trouver de meilleures solutions à des instances ouvertes ;
- de montrer que le  $J||L_{\max}$  n'est pas particulièrement plus difficile que le  $J||C_{\max}$ .

## Résultats

Instance	old LB	old UB	new LB	new UB	gap	time (s)
r_20_15_1_1_6	1027	1448	1027	<b>1429</b>	<b>4%</b>	3600
r_20_15_1_1_8	1127	<b>1552</b>	1127	1667	-27%	3600
r_20_15_1_1_4	1160	<b>1492</b>	1160	1557	-19%	3600
r_20_15_1_1_2	1140	1464	1140	<b>1383</b>	<b>25%</b>	3600
r_20_15_1_1_3	1182	<b>1501</b>	1182	1725	-70%	3600
r_20_15_1_2_1	1769	2090	<b>1817</b>	<b>1817</b>	<b>100%</b>	45
r_20_15_1_2_10	1775	2092	<b>1873</b>	<b>1873</b>	<b>100%</b>	0
r_20_15_1_2_9	1956	2246	<b>2020</b>	<b>2020</b>	<b>100%</b>	0
r_20_15_1_2_5	1925	2181	<b>1930</b>	<b>1930</b>	<b>100%</b>	105
r_20_15_1_2_8	1599	1785	<b>1636</b>	<b>1636</b>	<b>100%</b>	0
r_20_15_2_1_7	1575	<b>1957</b>	1575	2220	-68%	3600
r_20_15_2_1_3	1727	<b>2100</b>	1727	2203	-27%	3600
r_20_15_2_1_1	1785	2165	1785	<b>2083</b>	<b>21%</b>	3600
r_20_15_2_1_5	1521	<b>1839</b>	1521	2016	-55%	3600
r_20_15_2_1_9	1858	<b>2143</b>	1858	2182	-13%	3600
r_20_15_2_2_10	1282	<b>1682</b>	1282	1788	-26%	3600
r_20_15_2_2_2	<b>1688</b>	2174	1619	<b>2025</b>	<b>16%</b>	3600
r_20_15_2_2_3	1894	2381	1894	<b>2255</b>	<b>25%</b>	3600
r_20_15_2_2_7	1596	<b>1943</b>	1596	2012	-19%	3600
r_20_15_2_2_4	1663	2018	1663	<b>1769</b>	<b>70%</b>	3600

## Discussion




Cette transformation :

- permet de transformer simplement un  $J||L_{\max}$  en  $J||C_{\max}$  ;
- est proche de l'adaptation du *disjunctive graph* pour le  $J||L_{\max}$  par [Demirkol et al., 1997] ;
- conserve la structure du problème ;
- semble adaptée aux heuristiques.



# Merci

Merci pour votre attention.

# Bibliographie I

-  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.
-  Demirkol, E., Mehta, S., and Uzsoy, R. (1997).  
A computational study of shifting bottleneck procedures for shop scheduling problems.  
*Journal of Heuristics*, 3(2) :111–137.
-  Demirkol, E., Mehta, S., and Uzsoy, R. (1998).  
Benchmarks for shop scheduling problems.  
*European Journal of Operational Research*, 109(1) :137–141.

# Bibliographie II

-  Jain, A. S. and Meeran, S. (1998).  
Deterministic job-shop scheduling : Past, present and future.  
*European Journal of Operational Research*, 113 :390–434.
-  Su, L.-H., Chang, P.-C., and Lee, E. S. (1998).  
A heuristic for scheduling general job shops to minimize maximum lateness.  
*Mathematical and Computer Modelling*, 27(1) :1–15.