



**HAL**  
open science

# Méthode exacte pour le flowshop hybride avec machines à traitement par batches et compatibilité entre les tâches.

Adrien Bellanger, Ammar Oulamara

## ► To cite this version:

Adrien Bellanger, Ammar Oulamara. Méthode exacte pour le flowshop hybride avec machines à traitement par batches et compatibilité entre les tâches.. 9-ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision - ROADEF 2008, Feb 2008, Clermont-Ferrand, France. inria-00338745

**HAL Id: inria-00338745**

**<https://inria.hal.science/inria-00338745>**

Submitted on 14 Nov 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Méthode exacte pour le flowshop hybride avec machines à traitement par batches et compatibilité entre les tâches.

Bellanger, A. et Oulamara, A.

MACSI Projet LORIA-INRIA Lorraine, École des Mines de Nancy, Parc de Saurupt, 54042, Nancy  
adrien.bellanger@loria.fr ammar.oulamara@loria.fr

## 1 Introduction

Dans ce papier, nous nous intéressons au problème de minimisation de la durée total de l'ordonnancement de  $n$  tâches dans un flowshop hybride constitué de deux étages. Le premier étage est composé de  $m_1$  machines identiques en parallèle et le second étage contient  $m_2$  machines parallèles à traitement par batches. Les machines du premier étage ne peuvent exécuter qu'une seule tâche à la fois, alors que chaque machine du second étage peut exécuter jusqu'à  $k$  ( $k < n$ ) tâches simultanément, dans un même batch. Les tâches sont exécutées sur le premier étage puis sur le second étage. La durée opératoire d'une tâche  $T_j$  sur le premier étage est donnée par  $p_{1,j}$  ( $j = 1, \dots, n$ ), et sur le deuxième étage sa durée opératoire est donnée par un intervalle de temps  $[a_j, b_j]$ . Sur le second étage, les tâches peuvent être exécutées dans un même batch si elles sont compatibles entre elles. Autrement dit, si l'intersection de leurs intervalles de durée opératoire n'est pas vide (i.e.,  $[a_i, b_i] \cap [a_j, b_j] \neq \emptyset$ ). La durée d'exécution d'un batch sur le second étage correspond à la plus grande valeur initiale  $a_j$  des tâches compatibles.

Ce problème d'ordonnancement est issu du processus de fabrication de pneumatiques. En effet, les phases les plus importantes dans la production d'un pneu sont : l'assemblage et la cuisson. Ces deux phases peuvent être modélisées sous la forme d'un flowshop hybride. Au premier étage, les différents éléments (bande de roulement, flancs...) des pneus sont assemblés sur l'une des machines d'assemblage, pour obtenir un pneu dit *vert* (pneu avant vulcanisation). Le second étage représente la phase de cuisson, ou de vulcanisation. La vulcanisation s'effectue à haute température, et haute pression sur deux pneus à la fois (opération par batch). Ces deux pneus sont placés dans un moule à une température spécifique. La durée de vulcanisation de chaque type de pneu est comprise dans un intervalle. Deux pneus différents peuvent être cuits ensemble s'ils partagent une même durée de vulcanisation, car dès que la presse (machine contenant le moule) est fermée, il n'est plus possible de l'ouvrir avant la fin de la vulcanisation des deux pneus de la même presse.

Le problème de flowshop hybride avec des machines classiques à chaque étage est largement étudié dans la littérature de l'ordonnancement (voir Kis et al. [5] pour l'état-de-l'art). Or le cas où au moins un des deux étages est composé de machines à traitement par batches, à notre connaissance, n'est pas étudié dans la littérature. Pour plus de détails, se reporter à l'article long que nous avons présenté à la conférence FRANCORO V/ROADEF'07 [2], dans lequel nous avons abordé la complexité de ce problème et nous avons présenté quelques heuristiques.

## 2 Approche de Résolution

Nous proposons ici une méthode exacte de type séparation et évaluation (PSE) pour minimiser la durée totale de l'ordonnancement. D'après la notation standard en ordonnancement introduite par Graham et al. [1], le problème étudié ici est noté  $FH2(m_1, m_2)|p\text{-batch}(II), G_P = INT, k < n|C_{max}$ , où  $m_1$  représente le nombre de machines au premier étage, et  $m_2$  le nombre de machines au second étage. La notation  $p\text{-batch}(II)$  signifie que le second étage est composé de machines à traitement par batch,  $G_P = INT$  signifie que le graphe de compatibilité entre les tâches est un graphe d'intervalle qui concerne les durées opératoires des tâches, et  $k < n$  signifie que la capacité des machines à traitement par batches est une donnée du problème, enfin  $C_{max}$  indique que l'objectif est de minimiser la durée totale de l'ordonnancement (makespan).

Nous considérons ici le problème d'ordonnancement sur deux étages comme une succession de deux problèmes d'ordonnancement sur machines parallèles, où les dates de fin des tâches du premier problème représentent des dates d'arrivée pour le second problème. Ainsi, nous développons une

PSE pour le premier problème dans le but de générer des ordonnancements non dominés, puis en utilisant les résultats de ces ordonnancements nous proposons une deuxième PSE pour le second problème.

La première PSE consiste à générer un ensemble  $\Sigma$  d'ordonnancements contenant l'ordonnement optimal pour le problème de flowshop hybride. Pour ce faire, un arbre de recherche est utilisé afin de retourner, sous forme de listes, tous les ordonnancements non dominés du premier étage. Cette méthode est utilisée dans certaines PSE résolvant le problème de flowshop hybride, par exemple Haouari et Gharbi [4]. Ainsi un ensemble d'ordonnement  $\Sigma$  du premier étage est obtenu, chaque ordonnancement du premier étage fournit un ensemble de dates de disponibilité pour le second étage. La PSE du second étage, est différente non seulement du fait de la présence des batches, mais également de l'ordonnement du premier étage. Ainsi un arbre de recherche est construit pour chaque ordonnancement non dominé du premier étage. Chaque arbre correspond à la résolution d'un problème de minimisation de la date de fin du dernier batch (makespan) sur des machines parallèles à traitement par batch. À notre connaissance aucune PSE n'a été élaborée pour un tel problème. Les ordonnancement sont représentés par des listes de batches, ainsi les règles d'ordonnement de listes (*list scheduling*) sont utilisées. Le passage d'un noeud à un de ses fils est effectué en ajoutant un batch à la liste déjà créée, c'est pourquoi la création des fils se fait par énumération de tous les batches de tâches restantes, tout en respectant les compatibilités. Finalement nous conservons le meilleur ordonnancement de tous les arbres du second étage.

### 3 Réduction du nombre de noeuds

Afin de réduire le nombre de noeuds des règles de dominances adaptées au problème sont utilisée, mais l'efficacité de l'algorithme dépend également de la qualité des bornes inférieures et supérieures, notamment les bornes inférieures calculées pour chaque noeud. Enfin les tests de faisabilité (Feasibility and Adjustment Procedure FAP) utilisée par Gharbi et Haouari [3] pour le problème de machines parallèles classiques, sont adaptés à la résolution de notre problème. L'objectif de ces tests est de vérifier s'il est possible d'obtenir un ordonnancement de makespan strictement inférieur à la borne supérieure actuelle, généralement le meilleur ordonnancement obtenu jusqu'à présent.

### 4 Conclusion

Les premier tests que nous avons effectués nous ont permis de résoudre des problèmes contenant une dizaine de tâches et cinq machines à chaque étage en quelques secondes. Ces expériences seront approfondies pour la présentation finale du papier.

### Références

1. R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rimooy Kan. Optimization and approximation in deterministic sequencing and scheduling : a survey. *Annals of Discrete Mathematics*, 5 : 287-326 (1979).
2. A. Bellanger, et A. Oulamara. Flowshop hybride avec machines à traitement par batch et compatibilité entre les tâches. *FRANCORO V / ROADEF'07*, Livre des articles : 21-35 (2007)
3. A. Gharbi, M. Haouari. Minimizing makespan on parallel machines subject to release dates and delivery times. *J. Scheduling* 5 : 329-355 (2002)
4. M. Haouari, A.Gharbi. Optimal scheduling of a two-stage hybrid flow shop. *Math. Meth. Oper. Res.* 64 : 107-124 (2006)
5. T. Kis, et E. Pesch. A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. *European Journal of Operational Research* 164 : 592-608 (2005)