An exact method for solving the two-machine flow-shop problem with time delays

Mohamed Amine Mkadem, Aziz Moukrim, Mehdi Serairi

An exact method for solving the two-machine flow-shop problem with time delays

Mohamed Amine Mkadem (Speaker) * Aziz Moukrim * Mehdi Serairi *

1 Introduction

We address the flow-shop scheduling problem with two machines and time delays, which is denoted by $F2|l_j|C_{max}$. $F2|l_j|C_{max}$ can be described as follows. Given a set $J = \{1, 2, \ldots, n\}$ of n jobs and two machines (M_1, M_2), each job $j \in J$ has two operations $O_{1,j}$ and $O_{2,j}$. The operation $O_{1,j}$ (resp. $O_{2,j}$) must be executed without preemption during $p_{1,j}$ (resp. $p_{2,j}$) time units on M_1 (resp. M_2). In addition to the resource constraints of the machines and the non-preemption of jobs, a schedule S is considered feasible if, for each job $j \in J$, a time delay of at least l_j time units must elapse between the completion of $O_{1,j}$ and the start of $O_{2,j}$. The goal is to find a feasible schedule S that minimizes the makespan, i.e, the maximum job completion time.

The $F2|l_j|C_{max}$ problem is NP-hard in the strong sense even with unit-time operations [5]. Therefore, it has been the scope of a variety of investigations. As far we know, [2] and [4] proposed lower bound methods. Moreover, [2] investigated heuristic approaches where he introduced four constructive heuristics and a Tabu Search algorithm. It should be noted that [2] implemented an exact method based on the branch-and-bound method of [1], which is originally made for the job-shop problem. Another branch-and-bound method was proposed by [3] for the unit-time operations case.

The objective of this paper consists in introducing a branch-and-bound algorithm for $F2|l_j|C_{max}$. To our aim, we propose a set of lower bounds and an upper bound. In addition, a set of dominance rules is proposed in order to reduce the search space.

2 The Branch-and-Bound Algorithm

In this section, we present an exact method for $F2|l_j|C_{max}$ based on a branch-and-bound enumeration scheme. At first, let us introduce the following observation.

Observation 1 *Let σ be a fixed job sequence on M_1 of all jobs, then schedules in which the job sequence σ is fixed first on M_1 and the jobs are executed on M_2 according to the nondecreasing order of their arrival times are dominant.*

According to this observation, our branch-and-bound enumerates job sequences on M_1 as follows. At a given node N_{σ_1} of the search tree of the branch-and-bound, a partial

*{`mohamed-amine.mkadem, aziz.moukrim, mehdi.serairi`}`@hds.utc.fr`. Sorbonne universités, Université de technologie de Compiègne, CNRS, UMR 7253 Heudiasyc - CS 60 319 - 60 203 Compiègne cedex.

job sequence σ_1 of $|\sigma_1|$ jobs is fixed on M_1. In order to reduce the number of explored nodes, we invoke at each node N_{σ_1} the following features:

- A preprocessing procedure that aims to fix some precedence relationships between jobs on each machine.
- Several lower bounds of [2] and [4] that have been adapted to be used inside the branch-and-bound search tree.
- An upper bound based on a local search technique applied on the current subsequence.
- Three dominance rules that allow to discard nodes from additional expansions in order to reduce the computational time of the proposed branch-and-bound algorithm.

If these features fail to prune the current node N_{σ_1}, then for each unscheduled job j a *son node* is created from N_{σ_1}, in which j is fixed after σ_1. Note that we adopted the depth-first node selection strategy.

3 Computational results

A computer simulation of the branch-and-bound algorithm, which was carried out on a set of 360 instances of [2], shows that our branch-and-bound method outperforms the state of the art exact method. In particular, we manage to solve 358 instances among 360 possible ones.

References

[1] P. Brucker, B. Jurisch, and B. Sievers. A branch and bound algorithm for the job-shop scheduling problem. *Discrete Applied Mathematics*, 49(1):107 – 127, 1994.

[2] M. Dell'Amico. Shop problems with two machines and time lags. *Operations Research*, 44(5):777–787, 1996.

[3] A. Moukrim, D. Rebaine, and M. Serairi. A branch and bound algorithm for the two-machine flowshop problem with unit-time operations and time delays. *RAIRO - Operations Research*, 48(2):235–254, 2014.

[4] W. Yu. *The two-machine flow shop problem and the one-machine total tardiness problem.* PhD thesis, Eindhoven University of Technology, The Netherlands, 1996.

[5] W. Yu, H. Hoogeveen, and J.K. Lenstra. Minimizing makespan in a two-machine flow shop with delays and unit-time operations is np-hard. *Journal of Scheduling*, 7(5):333 – 348, 2004.