Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 32 (2005) 1831–1847
computers & operations research
www.elsevier.com/locate/dsw

A computational study of the permutation flow shop problem based on a tight lower bound

Talel Ladhari[a,b], Mohamed Haouari[a,*]

[a] *Laboratory of Mathematical Engineering, Ecole Polytechnique de Tunisie, BP 743, La Marsa 2078, Tunisia*
[b] *Ecole Supérieure des Sciences Economiques et Commerciales, Tunis 1089, Tunisia*

Abstract

We consider the classical permutation flow shop problem which requires scheduling n jobs through m machines which are placed in series so as to minimize the makespan. This problem is known to be $\mathcal{NP}$-hard. We describe a branch-and-bound algorithm with a lower bounding procedure based on the so-called two-machine flow shop problem with time lags, ready times, and delivery times. We present extensive computational results on both random instances, with up to 8000 operations, and well-known benchmarks, with up to 2000 operations, which show that the proposed algorithm solves large-scale instances in moderate CPU time. In particular, we report proven optimal solutions for benchmark problems which have been open for some time.

Keywords: Scheduling; Permutation flow shop; Branch-and-bound algorithm

1. Introduction

In this paper, we address the *Permutation Flow Shop Problem* which can be defined as follows. Each of n jobs from the job set $J = \{1, 2, \ldots, n\}$ has to be processed nonpreemptively on m machines $M_1, M_2, \ldots, M_m$ in that order. The processing time of job j on machine M_i is p_{ij}. At any time, each machine can process at most one job and each job can be processed on at most one machine. The problem is to find a processing order of the n jobs, the same for each machine (i.e. passing is not allowed), such that the time $C_{\max}$ at which all the jobs are completed (makespan) is minimized. Using the notation specified in Pinedo [1], this problem is denoted $F \mid prmu \mid C_{\max}$.

It is noteworthy that if $m \geqslant 4$, then there might exist nonpermutation schedules (i.e. with a specific job sequence for each machine) which dominate permutation ones. For instance, Potts et al. [2]

* Corresponding author.

0305-0548/$ - see front matter
doi:10.1016/j.cor.2003.12.001

exhibit a family of instances for which the value of the optimal permutation schedule is worse than that of the optimal nonpermutation schedule by a factor of more than $\frac{1}{2}\sqrt{m}$. However, for the sake of simplicity, we will adhere to a well-established tradition in scheduling theory and restrict our attention solely to permutation schedules.

Since the publication of the seminal paper of Johnson [3], the flow shop problem has become one of the most intensively investigated topics in scheduling theory. This interest is not only motivated by its practical relevance, but also by its deceptive simplicity and challenging hardness. Though, the flow shop problem is still considered as a very hard nut to crack. Indeed, up to the mid of the 1990s, the best available branch-and-bound algorithms experience difficulty in solving instances with 15 jobs and 4 machines [4, p. 393]. It is interesting to observe that at about the same time, instances of the celebrated traveling salesman problem with few hundreds of vertices could be solved quite routinely.

It is well-known that the case of two machines ($m = 2$), could be easily solved using Johnson's rule which generates an optimal schedule in $O(n \log n)$ time. For $m \geqslant 3$, however, the problem is shown to be strongly $\mathcal{NP}$-hard [5]. Interestingly, one can note that the quest for optimization strategies for the $F \mid prmu \mid C_{\max}$ started about 40 years ago, shortly after the discovery by Land and Doig [6] of the branch-and-bound technique. Indeed, the first branch-and-bound algorithms for the $F \mid prmu \mid C_{\max}$ were developed simultaneously, but independently, by Ignall and Schrage [7] and Lomnicki [8]. Following this pioneering work, several additional branch-and-bound algorithms have been published. The most significant contributions include Brown and Lomnicki [9], McMahon and Burton [10], Ashour [11], Lageweg et al. [12], Potts [13], Grabowski [14], Carlier and Rebai [15], and Cheng et al. [16]. All these algorithms, except the latter, can solve only instances of very limited size.

In addition to optimization methods, the intractability of the $F \mid prmu \mid C_{\max}$ motivated several authors to focus on the development of heuristic solution strategies. These heuristics are traditionally divided into two broad classes: constructive methods and local search methods. A nonexhaustive list of constructive methods include Campbell et al. [17], Dannenbring [18], and Nawaz et al. [19], to quote just a few. These methods are simple and fast, but perform rather poorly. The only significant exception being the heuristic of Nawaz et al., which is currently considered as the champion among constructive heuristics. On the other hand, most local search methods are based on modern metaheuristics. The papers include the simulated annealing algorithm of Osman and Potts [20], the tabu search algorithm of Nowicki and Smutnicki [21], and the genetic algorithm of Reeves [22]. Recently, a new local search paradigm based on a truncated branch-and-bound strategy, and called *branch-and-bound-based local search*, has been implemented for the $F \mid prmu \mid C_{\max}$ by Haouari and Ladhari [23] and shown to yield approximate solutions of excellent quality.

Moreover, during the last decade, an increased effort has been devoted to the design of approximation algorithms with guaranteed worst case bounds for the $F \mid prmu \mid C_{\max}$. The best known approximation algorithm has performance guarantee $\lceil m/2 \rceil$. The reader is referred to Smutnicki [24] for a survey. Recently, Sviridenko [25] proposed a new approximation algorithm which delivers a permutation schedule with makespan at most $O\left(\sqrt{m \log m}\right)$ times of the optimal nonpermutation schedule.

In this paper, we develop an effective branch-and-bound algorithm for the $F \mid prmu \mid C_{\max}$. Many of the features of our algorithm, such as the upper bounding and branching strategies, were