Available online at www.sciencedirect.com

ScienceDirect

Applied Mathematics and Computation 180 (2006) 185–193

APPLIED MATHEMATICS AND COMPUTATION

www.elsevier.com/locate/amc

Minimizing total completion time in a two-machine flow shop with deteriorating jobs

Ji-Bo Wang [a,b,*], C.T. Daniel Ng [b], T.C.E. Cheng [b], Li-Li Liu [b]

[a] *Department of Science, Shenyang Institute of Aeronautical Engineering, Shenyang 110034, People's Republic of China*
[b] *Department of Logistics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, People's Republic of China*

Abstract

This paper considers a two-machine flow shop scheduling problem with deteriorating jobs. By a deteriorating job, we mean that the processing time of a job is an increasing function of its execution start time. A simple linear deterioration function is assumed. The objective is to find a sequence that minimizes total completion time. Optimal solutions are obtained for some special cases. For the general case, several dominance properties and two lower bounds are derived to speed up the elimination process of a branch-and-bound algorithm. A heuristic algorithm is also proposed to overcome the inefficiency of the branch-and-bound algorithm. Computational results show that the proposed heuristic algorithm performs effectively and efficiently.
© 2006 Published by Elsevier Inc.

Keywords: Scheduling; Flow shop; Simple linear deterioration; Total completion time; Branch-and-bound algorithm

1. Introduction

There is a growing interest in the literature to study scheduling problems of *deteriorating jobs*, i.e., jobs whose processing times are increasing (decreasing) functions of their starting times. Such deterioration appears, e.g., in scheduling maintenance jobs, modelling of fire fighting, cleaning assignments, etc. Most of these studies focus on single machine settings, see, e.g., [2,4,11,15,16,20,22]. Chen [3] and Mosheiov [17] considered scheduling deteriorating jobs in a *multi-machine* setting. They assumed linear deterioration and parallel identical machines. Chen [3] considered minimum flow time and Mosheiov [17] studied makespan minimization.

Mosheiov [18] considered makespan minimization problem in flow shop, open shop and job shop with simple linear deteriorating jobs. He introduced a polynomial-time algorithm for the two-machine flow shop and proved NP-hardness when an arbitrary number of machines (three and above) is assumed. Wang and Xia [23]

* Corresponding author. Address: Department of Science, Shenyang Institute of Aeronautical Engineering, Shenyang 110034, People's Republic of China.

E-mail addresses: wangjibo75@yahoo.com.cn (J.-B. Wang), lgtctng@polyu.edu.hk (C.T. Daniel Ng), lgtcheng@polyu.edu.hk (T.C.E. Cheng), llliu@263.net (L.-L. Liu).

0096-3003/$ - see front matter © 2006 Published by Elsevier Inc.
doi:10.1016/j.amc.2005.11.162

considered no-wait and no-idle flow shop scheduling problems with job processing times dependent on their starting times. In these problems the job processing time is a simple linear function of the job's starting time and some dominating relationships between machines are satisfied. They showed that for the problems to minimize makespan or minimize weighted sum of completion time, polynomial algorithms still exist. When the objective is to minimize maximum lateness, the solutions of a classical version may not hold. Extensive surveys of different models and problems concerning deteriorating jobs can be found in [1,5].

In this paper we consider the two-machine flow shop scheduling problem to minimize the sum of completion times with simple linear deterioration. This model was proposed by Mosheiov [18]. It is well known that the flow shop scheduling to minimize total completion time is NP-hard even if there are no deteriorating jobs [9]. Therefore, the problem of two-machine flow shop scheduling to minimize total completion time with simple linear deterioration is NP-hard. For the literature on flow shop scheduling to minimize completion time without deteriorating jobs, the reader is referred to papers [6–13,19,21,24].

The rest of the paper is organized as follows. In Section 2, we give the problem description. In Section 3 we consider some polynomially solvable special cases. In Section 4 we propose several elimination rules to enhance the efficiency of the search for the optimal solution. In Section 5 we first develop a heuristic algorithm to find near-optimal solutions, then we establish two lower bounds to improve the speed of branching procedures, and finally we propose a branch-and-bound algorithm to search for the optimal solution. In Section 6 we present computational experiments of the branch-and-bound algorithm and the heuristic algorithm. Conclusions are given in the last section.

2. Problem description

Let $N = \{J_1, J_2, \ldots, J_n\}$ be the set of jobs to be scheduled, and $M = \{M_1, M_2\}$ be the two machines. Each job in the set N is processed first on the first machine and then on the second machine. Jobs can only be processed by one machine at a time, and the machines can only process one job at a time. Jobs are processed without interruption or preemption. Both machines are available at all times. The processing time p_{ij} of job J_j $(j = 1, 2, \ldots, n)$ on machine M_i $(i = 1, 2)$ is given as a simple linear increasing function dependent on its execution start time t:

$$p_{ij}(t) = a_{ij}t, \tag{1}$$

where $a_{ij} \in (0, 1)$ denotes the deterioration rate of job J_j on machine M_i. All the jobs are available for processing at time $t_0 > 0$. The objective is to find a schedule that minimizes total completion time or mean flow time, a widely used performance measure in scheduling literature. We assume unlimited intermediate storage between successive machines for the general flow shop scheduling problem.

Let $C_{ij}(\pi)$ denote the completion time of job J_j on machine M_i under some schedule π, and $C_{i[j]}(\pi)$ denote the completion time of the jth job on machine M_i under schedule π. Thus, the completion time of job J_j is $C_j = C_{2j}$. Using the three-field natation for problem classification, the problem can be represented as $F2|a_{ij}t|\sum C_j$. For ease of exposition, we denote a_{1j} by a_j, and a_{2j} by b_j, $j = 1, 2, \ldots, n$. Since unlimited intermediate storage is assumed, clearly an optimal schedule exists with no idle times between consecutive jobs on machine M_1. Therefore, from Mosheiov [16], the completion time of the jth job on machine M_1 is given by

$$C_{1[j]} = t_0 \prod_{i=1}^{j} \left(1 + a_{[i]}\right), \quad j = 1, 2, \ldots, n. \tag{2}$$

3. Solvable cases

Lemma 1. *There exists an optimal schedule in which the job sequence is identical on both machines.*

Proof. Similar to the proof of Lemma 1 in Mosheiov [18]. □

The conclusion of Lemma 1 is that only permutation schedules need be considered for this problem. In what follows, it is shown that our problem is solvable for some special cases.

Download English Version:

https://daneshyari.com/en/article/4636823

Download Persian Version:

https://daneshyari.com/article/4636823

Daneshyari.com