Computers & Operations Research 37 (2010) 83 – 90

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

A branch-and-bound algorithm for solving a two-machine flow shop problem with deteriorating jobs

C.T. Ng[a,*], J.-B. Wang[a,b], T.C.E. Cheng[a], L.L. Liu[a,c]

[a] *Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, People's Republic of China*
[b] *Department of Science, Shenyang Institute of Aeronautical Engineering, Shenyang 110034, People's Republic of China*
[c] *School of Science, Shanghai Second Polytechnic University, Shanghai 201209, People's Republic of China*

A R T I C L E I N F O

Available online 2 April 2009

Keywords:
Scheduling
Flow shop
Deteriorating jobs
Total completion time
Branch-and-bound algorithm

A B S T R A C T

In this paper we consider a two-machine flow shop scheduling problem with deteriorating jobs. By a deteriorating job we mean that the job's processing time is an increasing function of its starting time. We model job deterioration as a function that is proportional to a linear function of time. The objective is to find a sequence that minimizes the total completion time of the jobs. For the general case, we derive several dominance properties, some lower bounds, and an initial upper bound by using a heuristic algorithm, and apply them to speed up the elimination process of a branch-and-bound algorithm developed to solve the problem.

1. Introduction

In classical scheduling it is assumed that the processing times of the jobs are constant. This assumption, however, does not allow adequate modeling of realistic industrial processes. Hence, there is a growing interest in the literature to study scheduling problems involving *deteriorating jobs*, i.e., jobs whose processing times are increasing functions of their starting times. Job deterioration appears, e.g., in scheduling maintenance jobs or cleaning assignments, where any delay in processing a job is penalized by incurring additional time for accomplishing the job. Extensive surveys of different scheduling models and problems involving jobs with start time dependent processing times can be found in Alidaee and Womer [1], and Cheng et al. [2].

This paper addresses the problem of two-machine flow shop scheduling with deteriorating jobs. This model was proposed by Kononov and Gawiejnowicz [6]. They considered the makespan minimization problem. They showed that under linear deterioration the two-machine flow shop problem is strongly NP-hard, and the two-machine open shop problem is ordinary NP-hard. They also showed that in the three-machine flow shop with simple linear deterioration (i.e., the job's processing time is a simple linear function of its starting time), there does not exist a polynomial-time approximation algorithm with a worst-case ratio bounded by a constant. Finally, they proved that the three-machine open shop problem with simple linear deterioration is ordinary NP-hard. Mosheiov [8] considered the computational complexity of flow shop, open shop and job shop problems with simple linear deteriorating jobs to minimize the makespan. Mosheiov [8] introduced a polynomial-time algorithm for the two-machine flow shop and the two-machine open shop problems, respectively. He also proved that the three-machine flow shop, the three-machine open shop and the two-machine job shop problems are all NP-hard. Wang and Xia [11] considered the general, no-wait and no-idle flow shop scheduling problem with job processing times dependent on their starting times. They assumed that the job processing time is a linear function of its starting time, and that some dominating relationships between the processing times are satisfied. They showed that polynomial algorithms exist for the problems to minimize the makespan and the weighted sum of completion times. However, when the objective is to minimize the maximum lateness, the solutions of the classical version of the problems may not hold. Wang et al. [10] considered two-machine flow shop scheduling with simple linear deterioration. They implemented several dominance conditions and two lower bounds in a proposed branch-and-bound algorithm to search for the optimal solution. They also provided a heuristic algorithm to overcome the inefficiency of the branch-and-bound algorithm for large-sized problems.

The general problem under study can be described as follows. Each job in a set of n jobs is processed first on the first machine and then on the second machine. The jobs can only be processed by one machine at a time, and each machine can only process one job at a time. The jobs are processed without interruption or preemption.

* Corresponding author. Tel.: +852 2766 7364; fax: +852 2330 2704.
E-mail addresses: lgtctng@polyu.edu.hk (C.T. Ng), wangjibo75@yahoo.com.cn (J.-B. Wang), lgtcheng@polyu.edu.hk (T.C.E. Cheng).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2009.03.019

Both machines are always available and all the jobs are ready at time zero. The problem is to find an optimal schedule that minimizes the total completion time.

In this paper we consider two-machine flow shop scheduling to minimize the sum of completion times with job deterioration, which is modeled by a function that is proportional to a linear function of time. We call such a type of job deterioration "proportional deterioration". We believe that this is a very realistic setting, particularly in steel production. Consider the ingot preheating process in steel mills. Obviously, big ingots (i.e., ingots with long basic processing times) require longer preheating times (i.e., higher deterioration rates). It is well known that flow shop scheduling to minimize the total completion time is NP-hard even if there are no deteriorating jobs [3]. Therefore, the two-machine flow shop scheduling problem to minimize the sum of completion times with proportional deterioration is NP-hard.

The rest of this paper is organized as follows. In the next section we describe and formulate the problem. In Section 3 we propose several elimination rules, and apply them to enhance the search for the optimal solution for the general problem. In Section 4 we first develop some lower bounds to improve the branching procedure and an initial upper bound by using a heuristic algorithm, and then we propose a branch-and-bound algorithm to search for the optimal solution. In this section we also present the results of computational experiments to evaluate the proposed branch-and-bound algorithm. We conclude the paper in the last section.

2. Problem description

Let $N=\{J_1,J_2,\ldots,J_n\}$ be a set of jobs to be processed in a two-machine flow shop and $M=\{M_1,M_2\}$ be the set of two machines. Let p_{ij} be the actual processing time of job J_j ($j=1,2,\ldots,n$) on machine M_i ($i=1,2$) if it is started at time t in a sequence. The general job deterioration model is

$$p_{ij}=a_{ij}+b_{ij}t,$$

where a_{ij} is the basic processing time of job J_j on machine M_i, and b_{ij} is its deterioration rate. Wang et al. [10] considered the model where the processing times are $p_{ij}(t)=b_{ij}t$. As in Kononov and Gawiejnowicz [6], we consider the following general model:

$$p_{ij}=b_{ij}(a+bt), \tag{1}$$

where $a\geqslant 0, b\geqslant 0$. All the jobs are available for processing at time $t_0\geqslant 0$. The objective is to find a schedule that minimizes the total completion time. We assume unlimited intermediate storage between successive machines for the general flow shop scheduling problem.

Let $C_{i,j}(\pi)$ denote the completion time of job J_j on machine M_i under some schedule π. Let $C_{i,[j]}(\pi)$ denote the completion time of the jth job on machine M_i under schedule π. Thus, the completion time of job J_j is $C_j=C_{2,j}$. Using the three-field notation for scheduling problem classification, the problem under consideration can be denoted as $F2|p_{ij}=b_{ij}(a+bt)|\sum C_j$. For ease of exposition, we denote b_{1j} by α_j, and b_{2j} by $\beta_j, j=1,2,\ldots,n$. Since unlimited intermediate storage is assumed, it is evident that an optimal schedule exists with no idle time between consecutive jobs on machine M_1. Therefore, the completion time of the jth job on machine M_1 is given by (Kononov and Gawiejnowicz [6])

$$C_{1,[j]}=\left(t_0+\frac{a}{b}\right)\prod_{i=1}^{j}(1+b\alpha_{[i]})-\frac{a}{b}\quad \text{for } b>0,\quad j=1,2,\ldots,n. \tag{2}$$

3. Dominance properties

The problem under consideration is NP-hard since Garey et al. [3] showed that the problem is NP-hard even without linear deterioration. Therefore, a branch-and-bound algorithm is a viable approach to derive exact optimal solutions. In this paper we utilize the branch-and-bound technique to search for the optimal solution. In order to facilitate the search procedure, the domination properties of sequences are required for node elimination.

Given $S_1=(\pi,J_i,J_j,\pi')$, let $S_2=(\pi,J_j,J_i,\pi')$ be obtained from S_1 by only interchanging jobs J_i and J_j, where π and π' are partial sequences. Further, we assume that there are $r-1$ jobs in π. Thus jobs J_i and J_j are the rth (($r+1$)th) and the ($r+1$)th (rth)jobs in S_1 (S_2). To further simplify the notation, let A and B denote the completion times of the last job in π on M_1 and M_2, respectively. To show that S_2 dominates S_1, it suffices to show that $\sum_{j=1}^{n}C_{[j]}(S_2)\leqslant\sum_{j=1}^{n}C_{[j]}(S_1)$.

Proposition 1. *Suppose that two jobs J_i and J_j satisfy the following conditions*:

(i) *either* $(1+b\alpha_j)(1+b\beta_j)\leqslant(1+b\alpha_i)(1+b\beta_i)$ *or* $(A+(a/b))(1+b\alpha_j)(1+b\beta_j)\leqslant(B+(a/b))(1+b\beta_i)$;
(ii) *either* $(B+(a/b))(1+b\beta_j)\leqslant(A+(a/b))(1+b\alpha_i)(1+b\beta_i)$ *or* $\beta_j\leqslant\beta_i$;
(iii) *either* $\beta_i\leqslant\beta_j$ *or* $\alpha_j\leqslant\beta_j$ *or* $(A+(a/b))(1+b\alpha_j)(1+b\alpha_i)\leqslant(B+(a/b))(1+b\beta_j)$;
(iv) *either* $\beta_i\leqslant\alpha_i$ *or* $\alpha_j\leqslant\alpha_i$ *or* $(A+(a/b))(1+b\alpha_j)\leqslant(B+(a/b))$.

Then S_2 dominates S_1.

Proof. See Appendix A.

To further curtail the size of the branching tree, we develop two dominance properties for a pairwise interchange of two non-adjacent jobs. Let S_1 and S_2 be two job schedules in which the difference between S_1 and S_2 is a pairwise interchange of two non-adjacent jobs J_i and J_j, i.e., $S_1=(\pi,J_i,\pi',J_j,\pi'')$ and $S_2=(\pi,J_j,\pi',J_i,\pi'')$, where π, π' and π'' are partial sequences, and J_i and J_j are in positions r and τ in schedule S_1.

Theorem 1. *If jobs J_i and J_j satisfy $\alpha_i\geqslant\alpha_j$ and $\beta_i=\beta_j$, then S_2 dominates S_1.*

Proof. See Appendix B.

Theorem 2. *If jobs J_i and J_j satisfy $\alpha_i=\beta_i,\alpha_j=\beta_j$, $\alpha_i\geqslant\alpha_j$, then S_2 dominates S_1.*

Proof. See Appendix C.

Notice that the dominance relations given in Theorems 1 and 2 are global while the ones given in Proposition 1 is local. Since dominance relations (global or local), in general, do not yield optimal schedules, after the establishment of these relations, we use implicit enumeration such as the branch-and-bound technique to derive an optimal schedule.

4. Branch-and-bound algorithm

In this section we give four lower bounds to curtail the size of the branching tree, and an initial upper bound by using a heuristic algorithm. We also construct a branch-and-bound algorithm to solve small-sized problems.