Computers & Operations Research 38 (2011) 1760–1765

Contents lists available at ScienceDirect

Computers & Operations Research

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

Single-machine scheduling with deteriorating jobs and setup times to minimize the maximum tardiness

T.C.E. Cheng [a], Chou-Jung Hsu [b], Yi-Chi Huang [c], Wen-Chiung Lee [c,*]

[a] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[b] Department of Industrial Engineering and Management, Nan Kai Institute of Technology, Taiwan
[c] Department of Statistics, Feng Chia University, Taichung, Taiwan

ARTICLE INFO

Available online 21 February 2011

Keywords:
Single-machine scheduling
Deteriorating jobs
Setup time
Maximum tardiness

ABSTRACT

In many realistic production situations, a job processed later consumes more time than the same job when it is processed earlier. Production scheduling in such an environment is known as scheduling with deteriorating jobs. However, research on scheduling problems with deteriorating jobs has rarely considered explicit (separable) setup time (cost). In this paper, we consider a single-machine scheduling problem with deteriorating jobs and setup times to minimize the maximum tardiness. We provide a branch-and-bound algorithm to solve this problem. Computational experiments show that the algorithm can solve instances up to 1000 jobs in reasonable time.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In many real-world scheduling environments, a job processed later consumes more time than the same job when it is processed earlier. For example, in steel production, iron ingots need to be reheated before rolling if their temperatures drop below a threshold level while waiting to enter the rolling machine. In fire-fighting, the time and effort required to control a fire increase if there is a delay in starting the fire-fighting effort. In health care, more extensive medical treatment is necessary for a patient that receives late treatment because his/her health condition worsens over time [1–3]. Recently, Rachaniotis and Pappis [4] propose several demand-covering models for the deployment of available fire-fighting resources so that a forest fire is attacked within a specified time limit. In these cases, accomplishing a task needs more time as time passes.

Melnikov and Shafransky [5], Gupta and Gupta [1], and Browne and Yechiali [6] were among the pioneers that introduced deteriorating jobs to scheduling problems. Since then, many scheduling models dealing with deteriorating jobs have been proposed from a variety of perspectives. Alidaee and Womer [7] and Cheng et al. [8] provide reviews of different models and problems concerning deteriorating jobs. Moreover, Gawiejnowicz [9] presents a comprehensive discussion of different aspects of time-dependent scheduling and its applications. Recently, Lee et al. [10] study the makespan problem in the two-machine flowshop. Low et al. [11] consider the single-machine makespan problem with an availability constraint where jobs undergo simple linear deterioration, while Lee and Wu [12] study the same problem in the multiple parallel-machine setting. Wang et al. [13] consider some single-machine scheduling problems with deteriorating jobs where the jobs are related by a series–parallel graph. They show that polynomial algorithms exist for the problem with general linear deteriorating jobs to minimize the makespan and for the problem with proportional linear deteriorating jobs to minimize the total weighted completion time. Toksari and Guner [14] consider a parallel-machine earliness/tardiness scheduling problem with different penalties under the effects of learning and deterioration. Lee et al. [15] study a two-machine flowshop problem with deteriorating jobs and blocking to minimize the makespan. Li et al. [16] investigate a single-machine scheduling problem with deteriorating jobs. They show that the optimal schedule to minimize the sum of absolute differences in completion times is V-shaped. Sun [17] and Wang et al. [18] study scheduling models in which deteriorating jobs and learning effect are both considered simultaneously. They provide the optimal schedule for several single-machine problems. Lee et al. [19] address a total completion time scheduling problem in the multi-machine permutation flowshop where each machine has its own deterioration rate. Gawiejnowicz and Kononov [20] consider the problem of scheduling a set of independent, resumable, and proportionally deteriorating jobs on a single machine with multiple periods of machine non-availability.

However, most studies assume the setup time is negligible or part of the processing time. While this assumption simplifies the analysis and/or reflects certain applications, it adversely affects

* Corresponding author.
E-mail address: wclee@fcu.edu.tw (W.-C. Lee).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.11.014

the solution quality in many applications that require an explicit treatment of the setup operation. There are many practical applications that call for a separate consideration of the setup tasks from the processing tasks. These applications can be found in various shop types and situations, e.g., computer systems, paper bag factories, and the textile, container manufacturing, bottling, chemical, pharmaceutical, and food processing industries. We refer the reader to the reviews by Allahverdi et al. [21], Yang and Liao [22], Cheng et al. [23], Potts and Kovalyov [24], and Allahverdi et al. [25]. To the best of our knowledge, research on scheduling problems with deteriorating jobs has rarely considered explicit (separable) setup time (cost), except the following studies. Wang et al. [26] show that the single-machine group scheduling problem to minimize the makespan or total completion time is polynomially solvable under the model $p_{ij}(t)=a_{ij}-b_{ij}t$ and $s_i(t)=s_i$, where a_{ij} and b_{ij} are the basic processing time and the deterioration rate of job j in group i, t is the starting time, and s_i is the setup time of group i. Wu and Lee [27] show that the single-machine group scheduling problem to minimize the makespan or total completion time is polynomially solvable under the model $p_{ij}(t)=a_{ij}+bt$ and $s_i(t)=\delta_i+gt$, where a_{ij} is the basic processing time of job j in group i, b is the common job deterioration rate, t is the starting time, δ_i is the basic setup time of group i, and g is the common deterioration rate of group setup time. Leung et al. [28] consider an identical parallel-machine scheduling problem where the jobs are processed in batches and the processing time of each job is a step function of its waiting time. They show that the problem to minimize the total completion time is NP-hard in the strong sense. Ji and Cheng [29] consider a batch scheduling problem where the processing time of each job is a simple linear function of its waiting time. They show that the problem to minimize the makespan is strongly NP-hard. Pappis and Rachaniotis [30] consider the fire suppression problem where the objective is to maximize the total value of the burnt area remaining. They propose a branch-and-bound algorithm and heuristic algorithms to tackle this problem. Moreover, Pappis and Rachaniotis [31] provide a real-time synchronous heuristic algorithm and test the efficiency of the heuristic using real data.

Wu et al. [32] point out that the late processing of a job may require a longer setup or preparation time in the food processing and health care industries because food quality deteriorates or a patient's condition worsens over time. In this paper, we consider a single-machine scheduling problem where the job processing times and setup times are simple linear functions of their starting times. The objective is to minimize the maximum tardiness. The remainder of this paper is organized into five sections. We introduce the notation and formulate the problem in Section 2. We provide a branch-and-bound algorithm to solve the problem in Section 3. We present the computational experiments to test the performance of the algorithm and discuss the results in Section 4. We conclude the paper and suggest topics for future research in the last section.

2. Problem formulation

There are n jobs to be processed on a single machine, each of which belongs to one of M families. All the jobs are available at time t_0, where $t_0>0$. For each job j, there is a processing time p_j, a due date d_j, and a family code f_j. When a job is processed first on the machine or immediately after a job of another family, a sequence-independent setup time is necessary. No setup is required between two jobs of the same family. During the setup time, the machine is not available for processing. We assume that the actual job processing time of job j is a simple linear function of its starting time t such that

$$p_j=\alpha_j t, \quad j=1,2,\ldots,n,$$

where $\alpha_j>0$ is the deterioration rate of job j's processing time. Moreover, we assume that the actual setup time of a job from family i is also a simple linear function of its starting time t such that

$$s_i=\theta_i t, \quad i=1,2,\ldots,M,$$

where $\theta_i>0$ is the deterioration rate of family i's setup time. Under a schedule S, let $C_j(S)$ be the completion time of job j and $T_j(S)=\max\{0,C_j(S)-d_j\}$ be the tardiness of job j. The objective is to find a schedule such that $T_{max}=\max_{1\le j\le n}\{T_j(S)\}$ is minimized.

Let $ST_{si,b}$ denote the sequence-independent setup time scheduling problem as in [25]. Using the traditional three-field notation for scheduling problems, we denote the problem under study as $1/ST_{si,b}$, $p_j=\alpha_j t$, $s_i=\theta_i t/T_{max}$.

3. A branch-and-bound algorithm

For an arbitrary number of families, Bruno and Downey [33] show that the classical single-machine scheduling problem with sequence-independent setup times to minimize the maximum lateness is NP-hard. Although the complexity of $1/ST_{si,b}$, $p_j=\alpha_j t$, $s_i=\theta_i t/T_{max}$ is unknown, it is likely to be NP-hard. Thus we provide a branch-and-bound algorithm to solve the problem. We first provide some dominance properties, followed by a lower bound to speed up the search process. We then present the details of the branch-and-bound algorithm.

3.1. Dominance properties

In this subsection we derive some dominance rules that are helpful in eliminating the dominated sequences.

Theorem 1. *If jobs i and j are from the same family, $\alpha_i<\alpha_j$, and $d_i\le d_j$, then job i precedes job j in an optimal sequence.*

Proof. Suppose that S and S' are two job schedules and the difference between S and S' is a pairwise interchange of two jobs i and j from family u. That is, $S=(\pi,i,\pi',j,\pi'')$ and $S'=(\pi,j,\pi',i,\pi'')$, where each of π, π', and π'' denotes a partial sequence. The tardiness of the jobs in the partial sequence π and π'' is the same in both sequences since jobs i and j are from the same family and they are processed in the same order in both sequences. Moreover, we have $C_k(S)<C_k(S')$ for job $k\in\pi'$ because $\alpha_i<\alpha_j$. Thus, to show S dominates S', it suffices to show that $\max\{T_i(S),T_j(S)\}\le\max\{T_j(S'),T_i(S')\}$. We see that $C_i(S)<C_i(S')$ because job i is processed in a later position in S' and $C_j(S)=C_i(S')$ because they are from the same family and processed in the same position. From $d_i\le d_j$, we have

$$\begin{aligned}\max\{T_i(S),T_j(S)\}&=\max\{0,C_i(S)-d_i,C_j(S)-d_j\}\\&\le\max\{0,C_i(S')-d_i\}\le\max\{T_j(S'),T_i(S')\}.\end{aligned}$$

Thus, S dominates S' and the proof is completed.

To further expedite the search process, we provide a proposition to determine the feasibility of a partial schedule. Assume that (π^c,π) is a sequence of jobs where π is the scheduled part and π^c is the unscheduled part. Moreover, let $S^*=(\pi^*,\pi)$ be a sequence in which the unscheduled jobs in π^c are arranged as follows: Jobs in the same family as the first job in π are scheduled last, if any, and they are arranged in the earliest due date (EDD) order if there is more than one job. For the other jobs, they are arranged family by family, where jobs in the same family are scheduled in the EDD order, and the families are arranged in the EDD order of the maximum due dates of the families.

Download English Version:

https://daneshyari.com/en/article/10347226

Download Persian Version:

https://daneshyari.com/article/10347226

Daneshyari.com