Available online at www.sciencedirect.com

ScienceDirect

Computers & Industrial Engineering 52 (2007) 124–132

computers & industrial engineering

www.elsevier.com/locate/dsw

Heuristic algorithms for solving the maximum lateness scheduling problem with learning considerations

Chin-Chia Wu *, Wen-Chiung Lee, Tsung Chen

Department of Statistics, Feng Chia University, 100 Wenhua Road, Taichung, Taiwan

Received 10 February 2004; received in revised form 3 August 2006; accepted 19 October 2006
Available online 8 December 2006

Abstract

In many situations, a worker's ability improves as a result of repeating the same or similar task; this phenomenon is known as the "learning effect". In this paper, the learning effect is considered in a single-machine maximum lateness minimization problem. A branch-and-bound algorithm, incorporating several dominance properties, is provided to derive the optimal solution. In addition, two heuristic algorithms are proposed for this problem. The first one is based on the earliest due date (EDD) rule and a pairwise neighborhood search. The second one is based on the simulated annealing (SA) approach. Our computational results show that the SA algorithm is surprisingly accurate for a small to medium number of jobs. Moreover, the SA algorithm outperforms the traditional heuristic algorithm in terms of quality and execution time for a large number of jobs.

Keywords: Scheduling; Simulated annealing; Single machine; Maximum lateness; Learning effect

1. Introduction

Single-machine scheduling problems have received considerable attention for several reasons. First, there are obvious applications involving a single machine, e.g., serial job processing with a small computer. Then there are less obvious treatments of a large complex plant acting as if it were one machine, e.g., a paint manufacturing plant devoted to making one color of paint at a time. Finally, there are job shops with systems of many machines but containing one machine that acts as a "bottle-neck" (French, 1982).

Job processing times are assumed to be known and fixed in traditional research. However, in many realistic scheduling settings, the productivity of a production facility (a machine, a plant, a worker, etc.) improves continuously with time. As a result, the processing time of a given product is shorter if it is scheduled later in the production sequence (Mosheiov, 2001b). This phenomenon is known as the "learning effect," and the realism of its consideration in research has come to be recognized as both practical and important.

* Corresponding author.
E-mail address: cchwu@fcu.edu.tw (C.-C. Wu).

0360-8352/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2006.10.007

The impact of learning on productivity in manufacturing was first found by Wright (1936) in the aircraft industry and was subsequently discovered in many industries in both manufacturing and service sectors (Yelle, 1979). Numerous studies have been devoted to investigating the learning effect over a variety of industrial fields, but its research is still very limited in the scheduling field. Biskup (1999) considered a model that assumed the time needed to perform an operation decreases as a function of the number of repetitions and showed that three important single-machine problems remain polynomial solvable, namely, minimizing the flow time, minimizing the sum of job completion times, and minimizing the weighted sum of completion time deviations from a common due date. Mosheiov (2001a) gave several examples to demonstrate that the optimal schedule may be very different from that of the classical version of the problem when the learning effect is considered. In particular, he presented a counterexample to show that the EDD rule might not provide the optimal solution for the single-machine maximum lateness problem. However, the complexity of this problem remains open. Mosheiov and Sidney (2003) extended the study to the case of job-dependent learning curves, that is, when learning associated with some jobs is faster than that with others. They showed that the problems of makespan and total flow time minimization on a single machine, a due-date assignment problem, and total flow time minimization on unrelated parallel machines remain polynomial solvable. Apart from these articles, there are other approaches involving decreasing processing times that are similar to the concept of learning. (See Meilijson & Tamir, 1984; Dondeti & Mohanty, 1995, 1998; Gawiejnowicz, 1996; Cheng & Wang, 2000.)

The problem can be formulated as follows. There is a set of n jobs that are available at time zero. Each job i has a normal processing time t_i and due date d_i. The actual processing time of job i is $t_i r^a$ if it is scheduled in position r in a sequence, where a is the learning effect. For a given schedule S, the lateness of job i in S is defined as

$$L_i(S) = C_i(S) - d_i,$$

where $C_i(S)$ denotes the completion time of job i in S. Thus, the problem is to determine a schedule that minimizes the maximum lateness

$$L_{\max}(S) = \max_{1 \leqslant i \leqslant n}\{L_i(S)\}.$$

This paper is organized as follows: an exact solution procedure incorporating several dominance properties is presented in the next section. A heuristic algorithm based on EDD is derived in Section 3. A simulated annealing algorithm and the implementation details for the problem are described in Section 4. The simulation experiment and its results are provided in Section 5. Conclusions are presented in the last section.

2. An exact solution procedure

In order to evaluate the accuracy and efficiency of the proposed heuristic algorithms, a branch-and-bound technique is also proposed to search for the optimal solution. In this section, we develop non-adjacent and several dominance properties to reduce the searching scope.

Property 1. If $d_i \leqslant d_j$ and $t_i < t_j$, then job i must precede job j in an optimal schedule.

Proof. Let S_1 and S_2 be two job schedules, and the difference between S_1 and S_2 is a pairwise interchange of two jobs i and j scheduled in the r_1th and the r_2th positions. That is,

$$S_1 = (\pi_1 i \pi_2 j \pi_3)$$

and

$$S_2 = (\pi_1 j \pi_2 i \pi_3),$$

where π_1, π_2, and π_3 are partial sequences.

It is clear that the lateness of job k in S_1 and S_2 are equal if job k is in π_1 since it is processed in the same order in both sequences. Since $t_i < t_j$, we have

$$C_j(S_2) > C_i(S_1). \tag{1}$$