Computers & Industrial Engineering 57 (2009) 1258–1267

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Modeling and scheduling a case of flexible flowshops: Total weighted tardiness minimization

B. Naderi [a], M. Zandieh [b,*], M.A.H.A. Shirazi [b]

[a] Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

[b] Department of Industrial Management, Management and Accounting Faculty, Shahid Beheshti University, GC, Tehran, Iran

ARTICLE INFO

Article history:
Received 4 November 2007
Received in revised form 4 June 2009
Accepted 10 June 2009
Available online 25 June 2009

Keywords:
Flexible flowshop
Total weighted tardiness
Anticipatory sequence-dependent setup times
Transportation times
Mixed integer linear program
Electromagnetism algorithm

ABSTRACT

Two of the most realistic assumptions in the field of scheduling are the consideration of setup and transportation times. In this paper, we study the flexible flowshop scheduling where setup times are anticipatory sequence-dependent and transportation times are job-independent. We also assume that there are several transporters to carry jobs. The objective is to minimize total weighted tardiness. We first formulate the problem as a mixed integer linear programming (MILP) model. With this, we solve small-sized instances to optimality. Since this problem is known to be NP-hard, we then propose an effective metaheuristic to tackle large-sized instances. This metaheuristic, called electromagnetism algorithm (EMA), originates from the attraction–repulsion mechanism of the electromagnetism theory. We conduct a series of experiments and complete statistical analyses to evaluate the performance of the proposed MILP model and EMA. On a set of instances, we first tune the parameters of EMA. Then, the efficiency of the model and general performance of the proposed EMA are assessed over a set of small-sized instances. To further evaluate EMA, we compare it against two high performing metaheuristics existing in the literature over a set of large-sized instances. The results demonstrate that the proposed MILP model and EMA are effective for this problem.

1. Introduction

Scheduling problems are the allocation of limited resources to perform a set of activities in a period of time (Pinedo, 2008). A flexible flowshop scheduling (FFSS) is one of the most distinguished scheduling environments that have numerous applications in real industrial settings (Ruiz & Maroto, 2006; Zandieh, Fatemi Ghomi, & Moattar Husseini, 2006). In FFSS, we have a set of n jobs need to be operated at a set of m production stages. Each stage i has a set of m_i identical machines in parallel where $m_i \geqslant 1$, $i = \{1, 2, \ldots, m\}$. Some stages may have only one machine, but for the plant to be qualified as a flexible flowshop, at least one stage must have more than one machine (i.e. $m_i > 1$). The aim of disposing machines in parallel at stages is to reduce the impact of bottleneck and to balance the work flow in shop floors. Each job j, $j = \{1, 2, \ldots, n\}$ is processed exactly by one machine at each stage. In FFSS, All n jobs need to be processed at all m stages in the same route, starting at stage 1 until finishing at stage m (Pinedo, 2008). The operation of job j at stage i is denoted by 0_{ji}. The following assumptions are usually characterized in FFSS: Jobs are independent (i.e. there is no precedence constraint between jobs) and are available at time 0. Each job can be performed by at most one machine at a time. Each job j requires a fixed and predetermined amount of processing time at each stage j, denoted by $P_{j,i}$. Machines are continuously available (i.e. there is no breakdown or machine failure). Each machine can process at most one job at a time. There is an unlimited buffer between every two consecutive stages.

There has been an upsurge of interest in considering setup times. The main reason why researchers have been motivated to utilize this assumption is to solve scheduling problems in a real manner and because of the tremendous savings when setup times are explicitly incorporated in scheduling decision (Allahverdi & Soroush, 2008). The setup times could be either sequence-independent or sequence-dependent. As general, sequence-independent setup times can be ignored or combined with the processing times. In sequence-dependent setup times (SDST), we consider that between the processing of two consecutive jobs on the same machine, some setup must be performed that depends on the ordering of these two jobs. For example, this may occur in a painting operation, where different initial paint colors require different levels of cleaning when being followed by other paint colors.

The sequence-dependent setup times can be either anticipatory or non-anticipatory (Allahverdi, Ng, Cheng, & Kovalyov, 2006). Fig. 1 shows the classification of setup times in the literature. If the setup is non-anticipatory (NSDST), the setup can begin only if both job and machine are available. On the other hand, if the setup

* Corresponding author.
E-mail address: M_zandieh@sbu.ac.ir (M. Zandieh).

0360-8352/$ - see front matter © 2009 Published by Elsevier Ltd.
doi:10.1016/j.cie.2009.06.005

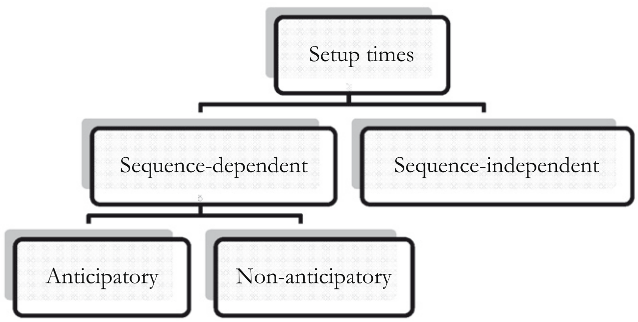


Fig. 1. Classification of setup times in scheduling problems.

is anticipatory (or ASDST), the setup can begin even if the job is not available to process as long as the machine is idle. Let us further clarify the difference between NSDST and ASDST by an example. Consider the operations of job j on two consecutive machines i and $i+1$ where $P_{j,i} = 20,\ P_{j,i+1} = 15$, the setup times of job j on machines i and $i+1$ are 10 and 5, respectively. In the case of ASDST, 0_{ji+1} can begin at 30 since its setup can start at 25. In the case of NSDST, 0_{ji+1}can begin at 35 since the earliest time that its setup can begin is at 30. The setup lasts for five time units. After finishing its setup at 35, the operation can be performed. Fig. 2 shows Gantt chart of the example for both ASDST and NSDST.

Another recently popular assumption is that it may be impossible to start 0_{ji} immediately after the completion of 0_{ji-1} because the job must first be transported from stage $i-1$ to stage i by a transporter. It picks job j from stage $i-1$ and delivers it to stage i, and then returns to stage $i-1$. The time to load and unload the transporter is included in the transportation time. When transporter reaches to stage i, it delivers the job to the machine if the setup time has been completed and machine is ready to receive the job. If not, it leaves the job at the buffer between two stages until the process of that job can begin i.e. the setup on one of the machines available at that stage is finished. After delivering the job, the transporter immediately starts its return to stage $i-1$. The transportation times could be either job-independent or job-dependent. In job-independent case, the magnitudes of transportation times only depend on the distance between two consecutive stages while in job-dependent, they are determined by the distances as well as the job to be carried. On another classification, there are two types of transportation systems: (1) Multi-transporter in which it is assumed there are several (unlimited) vehicles to do carry jobs; as a result, a job never has to wait for the transporter before its transportation. (2) Single-transporter in which it is assumed that all the transportations between two stages are done by a single-transporter; therefore, a job might wait for the transporter to return (Sule, 1996).

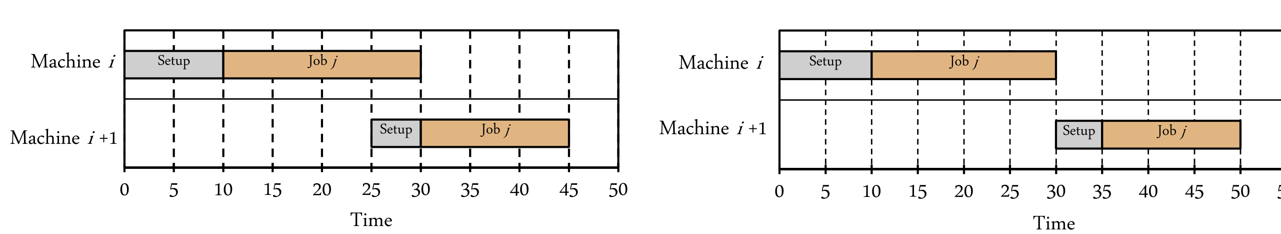
A variety of objectives has been focused by the researchers in production scheduling, ranging from minimizing makespan, maximum tardiness, total weighted flow time, and total weighted tardiness. Makespan is widely used in the literature of scheduling (Jin, Yang, & Ito, 2006; Logendran, Carson, & Hanson, 2005). The criteria based on due dates for delivery are more suitable for make-to-order environments, and tardiness objectives have recently become more important than those based on makespan (Vallada, Ruiz, & Minella, 2008). In this article, we consider total weighted tardiness (TWT) due to its importance to the real industrial setting.

FFSS is known to be an NP-hard problem (Zandieh et al., 2006). Recently many algorithms based on computational intelligence are proposed for FFSSs (Janiaka, Kozanb, Lichtensteina, & Oguzc, 2007; Jin et al., 2006; Logendran et al., 2005). This paper investigates FFSS where two realistic assumptions are considered: (1) The setup times are anticipatory and sequence-dependent. (2) The transportation are job-dependent and carried out in a multi-transporter system. The objective is the minimization of total weighted tardiness. A mathematical formulation, in the form of a mixed integer linear program, is developed. We then propose an electromagnetism algorithm (EMA) for the aforementioned problem. We evaluate the performance of the model and EMA on a series of the experiments.

The rest of paper is organized as follows: Section 2 goes over the literature of SDST FFSS. Section 3 provides an illustrative example of the problem. Section 4 formulates the problem as a mixed integer linear program. Section 5 describes the proposed EMA. In Section 6, the experimental design and comparison of the proposed EMA with the existing methods are presented. Finally Section 7 concludes the paper and introduces some directions for future studies.

2. Literature review

The area of production scheduling has been a very active field of research since Johnson (1954) proposed first systematic approach. He introduces a method to find the optimal solution of two-machine (and special case of three-machine) flowshops. Since then, numerous approaches have been proposed for the flowshop in many papers. Among all, we can point out to algorithms proposed by Campbell, Dudek, and Smith (1970), Nawaz, Enscore, and Ham (1983), Palmer (1965). In the case of the NSDST flexible flowshop, Kurz and Askin (2003) compare several dispatching rules in three different classes: (1) greedy heuristics, (2) The insertion heuristics and (3) Johnson's rule. The objective is makespan minimization. Moreover, Kurz and Askin (2004) consider the same problem as they do in (Kurz & Askin, 2003) and this time develop a random keys genetic algorithm (RKGA) for it. They evaluate the RKGA through comparing it against other heuristics they proposed aforetime. Zandieh et al. (2006) study the same problem and propose an immune algorithm. The objective is still makespan minimization. They show that this algorithm outperforms the RKGA of Kurz and Askin (2004). Ruiz and Maroto (2006) study the NSDST hybrid flowshop with unrelated machines to minimize makespan and propose a calibrated genetic algorithm. Naderi, Zandieh, Khaleghi Ghoshe Balagh, and Roshanaei (2008) study general hybrid



(a) Anticipatory setups (b) Non-anticipatory setups

Fig. 2. Difference between (a) anticipatory and (b) non-anticipatory setup times.