Applied Mathematics and Computation 270 (2015) 312–333

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Non-permutation flow shop scheduling with order acceptance and weighted tardiness

Yiyong Xiao [a], Yingying Yuan [b], Ren-Qian Zhang [c,*], Abdullah Konak [d]

[a] School of Reliability and System Engineering, Beihang University, Beijing 100191, China
[b] CSR Zhuzhou Locomotive Research Institute Co., Ltd., Zhuzhou City, Hunan Province 412001, China
[c] School of Economics and Management, Beihang University, Beijing 100191, China
[d] Information Sciences and Technology, Penn State Berks, Tulpehocken Road, P.O. Box 7009, Reading, PA 19610-6009, United States

ARTICLE INFO

Keywords:
Flow shop scheduling
Order acceptance
Weighted tardiness
Genetic algorithm

ABSTRACT

This paper studies the non-permutation solution for the problem of flow shop scheduling with order acceptance and weighted tardiness (FSS-OAWT). We formulate the problem as a linear mixed integer programming (LMIP) model that can be optimally solved by AMPL/CPLEX for small-sized problems. In addition, a non-linear integer programming (NIP) model is presented to design heuristic algorithms. A two-phase genetic algorithm (TP-GA) is developed to solve the problem of medium and large sizes based on the NIP model. The properties of FSS-OAWT are investigated and several theorems for permutation and non-permutation optimum are provided. The performance of the TP-GA is studied through rigorous computational experiments using a large number of numeric instances. The LMIP model is used to demonstrate the differences between permutation and non-permutation solutions to the FSS-OAWT problem. The results show that a considerably large portion of the instances have only an optimal non-permutation schedule (e.g., 43.3% for small-sized), and the proposed TP-GA algorithms are effective in solving the FSS-OAWT problems of various scales (small, medium, and large) with both permutation and non-permutation solutions.

1. Introduction

The flow shop scheduling (FSS) problem and its variants have been studied for more than half a century since Johnson proposed the well-known SPT(1)-LPT(2) algorithm for $F_2||C_{\max}$ problem in the 1950s (Johnson [1]). The classical FSS problem typically aims to minimize the maximum completion time/makespan of n jobs that will be executed through m machines. If all machines process their jobs in the same sequence, a solution of the FSS problem is called a permutation schedule; otherwise, it is known as a non-permutation schedule (Lageweg et al. [2]; Potts et al. [3]). Since finding optimal non-permutation solutions is much more difficult than finding permutation ones even for small-sized FSS problems, the majority of the research in the literature has mainly focused on finding permutation solutions.

Palmer [4] proposed a heuristic based on *slope index* for permutation flow shop scheduling in minimizing total time ($F_m|prmt|C_{\max}$). Conway et al. [5] proved that when the number of machines is three or less, there must exist an optimal solution with a permutation processing sequence. Thus, one can consider only permutation schedules for $F_m||C_{\max}$ problems with three or less machines. Smith et al. [6] showed that if the first machine always takes the longest processing time among all

* Corresponding author. Tel.: +86 10 82338412.
E-mail address: zhangrenqian@buaa.edu.cn (R.-Q. Zhang).

http://dx.doi.org/10.1016/j.amc.2015.08.011
0096-3003/

machines, there must exist a permutation schedule that would result in an optimal makespan. Tandon et al. [7] compared the makespan values obtained by permutation and non-permutation schedules in flow shop problems, and they showed that the average percentage improvement of optimum non-permutation schedules over optimum permutation schedules is usually more than 1.5%. Ruiz and Maroto [8] presented a comprehensive review of many existing heuristics and metaheuristics for the permutation FSS problem. Liao et al. [9] compared the makespan, total tardiness, and total weighted tardiness obtained by permutation and non-permutation schedules. According to their computational results, the percentage improvement is quite small (less than 0.5%) with respect to the makespan, but it is significantly large (in the range of 1–5%) with respect to the total tardiness and the weighted total tardiness. Koulamas [10] proposed a simple construction heuristic (HFC) to produce non-permutation schedules for the flow shop makespan problem. Liao and Huang [11] and Mehravaran and Logendran [12,13] proposed meta-heuristic algorithms based on Tabu search to find non-permutation solutions for FSS problems with total tardiness.

In a majority of FSS problems, the processing times of different jobs on different machines are often assumed to be precisely known. In some cases, however, they may also be treated as fuzzy values (Chanas and Kasperski [14]; Niu et al. [15]) or as dependent on their starting times (Cheng et al. [16]; Wang et al. [17]; Cheng and Sun [18]; Yang and Wang [19]; Sun et al. [20]) or as dependent on their positions (Li et al. [21]). The learning-curve effect that decreases processing times in certain industries may also be considered (Wang and Guo [22]; Vahedi-Nouri et al. [23]).

An implicit underlying assumption in most FSS models is that all job orders must be accepted (i.e., job order rejections are not allowed at the scheduling stage). Only a few papers consider order rejection in the FSS problem (Pourbabai [24,25]; De et al. [26]; Stadje [27]; Shabtay and Gasper [28]; Xiao et al. [29]). In real-life, however, firms reject some jobs (or outsource them at a certain loss) in situations where accepting all jobs may result in harsh tardiness penalties because of limited production capacity. Pourbabai [24,25] first proposed a job selection model to determine how to accept (or reject) candidate orders and in what quantities such that the net operational profit including weighted tardiness penalties can be maximized. Gupta et al. [30] considered the problem of simultaneous selection of a subset from N projects and the determination of an optimal sequence for the selected projects to maximize the total net return. This work was later extended by De et al. [26] by considering job selection and scheduling on a single machine with random processing times and deadlines. Stadje [27] proposed a single-machine model for the problem of job selection and sequencing assuming that the machine is non-repairable and may breakdown with a predictable probability. Other single machine models include Cheng and Sun [18] who studied single machine scheduling problems in a deteriorating processing environment and Zhang et al. [31] who considered the release dates and order rejection. Shabtay and Gasper [28] studied a two-machine FSS problem considering order rejection cost. Xiao et al. [29] studied the multiple-machine FSS problem with order acceptance and weighted tardiness, and they developed a heuristic algorithm to find permutation solutions for the problem.

In recent decades, the importance of order acceptance and scheduling (OAS) has been recognized, and the benefits of cooperation between the sales and the planning functions have been increasingly studied in both academy and practice (Zijm [32]; Ebben et al. [33]; Slotnick and Morton [34]; Slotnick [35]). The OAS problem involves how to select candidate orders and schedule the accepted ones in a proper sequence to go through a processing system with the objective of maximizing the total net profit. Commonly, the accepted orders will bring in profits, but may also incur penalties if they are not delivered on time. Slotnick and Morton [36] first proposed an OAS model for a single machine system with weighted lateness penalties. Ghosh [37] proved that the Slotnick–Morton version of the OAS problem is NP-hard in the ordinary sense. Slotnick and Morton [34] extended their model to a weighted tardiness version that considers only delayed penalties, and Rom and Slotnick [38] developed a genetic algorithm to solve the OAS problem. Lewis and Slotnick [39] studied the OAS problem in a multi-period case and considered future order losses due to rejecting current orders. Ebben et al. [33] studied the OAS problem in a high-demand job shop environment and proposed a workload-based method. Yang and Geunes [40] assumed that the processing times of jobs are controllable and can be compressed at a specific cost. Oğuz et al. [41] incorporated sequence-dependent setup times and deadlines into the OAS model and developed meta-heuristic algorithms to solve large-sized problems (300 orders maximum). Cesaret et al. [42] developed a Tabu Search algorithm to solve the OAS problem on a single machine by considering release dates and sequence-dependent setup times. Wang et al. [43] recently proposed a two-machine OAS model and developed an exact B&B algorithm to solve small-sized problems. Xiao et al. [29] studied order acceptance combined with permutation FSS with a maximum of 20 machines. The planning and scheduling issues arising particularly in the batch process industries attracted significant interest (Raaymakers and Hoogeveen [44]; Raaymakers et al. [45]; Lewis and Slotnick [39]; Roundy et al. [46]; Ivănescu [47,48]). Batch process industries cover many industrial areas, such as food, chemistry, pharmaceutical, etc., and they are characterized as a no-wait job shop environment with overlapping operations in which different products may follow different routings.

In this paper, we study the problem of flow shop scheduling with order acceptance and weighted tardiness (FSS-OAWT), which is strongly NP-hard and the solution space for a non-permutation schedule is extremely large even for small-sized problems. We provide two mathematical models for solving the problem, one is a linear mixed integer programming (LMIP) model that can be optimally solved by commercial solvers such as CPLEX, Lingo, etc., and the other is a non-linear integer programming (NIP) model for heuristic algorithms to solve medium- and large-sized problems. We theoretically study the properties of permutation and non-permutation optimum solutions and develop several theorems for this problem. A two-phase genetic algorithm (TP-GA) is proposed to solve the problem with near-optimal solutions. The proposed algorithm is rigorously evaluated using a comprehensive set of problem instances of different scales (small, medium, large, and very large). The contributions of this paper are in four areas as follows: (i) a linear mathematical model is developed for the FSS-OAWT problem; (ii) a genetic algorithm is introduced to find both permutation and non-permutation solutions; (iii) the theoretical properties of the problem are studied, and the derived properties can be used to develop effective and efficient heuristics; (iv) through rigorous