Applied Soft Computing 30 (2015) 462–474

Contents lists available at ScienceDirect

Applied Soft Computing

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

Order acceptance and scheduling to maximize total net revenue in permutation flowshops with weighted tardiness

Shih-Wei Lin [a,b], Kuo-Ching Ying [c,*]

[a] Department of Information Management, Chang Gung University, Taoyuan, Taiwan
[b] Department of Medical Research and Development, Linko Chang Gung Memorial Hospital, Taiwan
[c] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 10 August 2014
Received in revised form 28 January 2015
Accepted 30 January 2015
Available online 12 February 2015

Keywords:
Order acceptance and scheduling problem
Multi-initiator simulated annealing
Permutation flowshop
Weighted tardiness

ABSTRACT

The order acceptance and scheduling (OAS) problem is important in make-to-order production systems in which production capacity is limited and order delivery requirements are applied. This study proposes a multi-initiator simulated annealing (MSA) algorithm to maximize the total net revenue for the permutation flowshop scheduling problem with order acceptance and weighted tardiness. To evaluate the performance of the proposed MSA algorithm, computational experiments are performed and compared for a benchmark problem set of test instances with up to 500 orders. Experimental results reveal that the proposed heuristic outperforms the state-of-the-art algorithm and obtains the best solutions in 140 out of 160 benchmark instances.

1. Introduction

The order acceptance and scheduling (OAS) problem, proposed by Guerrero and Kern [1], has attracted increasing attention both from researchers and practitioners. Order acceptance involves determining the orders to be accepted for processing, while order scheduling involves deciding the production sequence of the accepted orders [2]. A trade-off between revenue and cost is inevitably in the decision-making of order processing [1]. Trade-offs often occur in many make-to-order production systems that have limited production capacity and short delivery deadlines, such as in the printing [3], lamination [4], steel production [5] and laundry service [6] industries.

Various OAS problems with different characteristics and objectives have been studied over the last two decades. Many exact methods have been utilized to solve those OAS problems, such as integer programming [7], mixed integer programming [4,8–11], dynamic programming [12–16] and branch-and-bound algorithms [11,17–19]. Due to the complexity of the OAS problem, a global optimal solution can be difficult to find when the problem size is particularly large. Accordingly, researchers may seek efficient approximation heuristics in order to find a near-optimal solution within reasonable computation time. Many approximation heuristics are proven robust in delivering near-optimal solutions and resolving limitations encountered in exact methods [20,21]. Currently available approximation heuristics for solving OAS problems can be generally classified into two categories: constructive heuristics (CHs) and improvement heuristics (IHs).

CHs, such as those developed by Stern and Avivi [7], Kyparisis et al. [22], Lewis and Slotnick [16], Engels et al. [12], Yang and Geunes [23], Lee and Sung [17], Oğuz et al. [4], Cesaret et al. [24] and Xiao et al. [6], add orders one at a time and examine the effect of each addition on the objective function value. When an order has been accepted and sequenced, it is fixed and cannot be reversed. Among the available CHs, the weighted shortest processing time (WSPT) heuristic and the due date (DD) heuristic, proposed by Xiao et al. [6], are two of the better approaches. The WSPT heuristic makes order-acceptance decisions based on the increasing order of the weighted shortest processing time; the sequence order of the DD heuristic is made on the basis of increasing due dates, after which the same acceptance decision procedure as for WSPT is applied. However, a common feature of these CHs is the non-robustness of their solutions. Although these CHs yield solutions rapidly, there is not a specific CH which outperforms all other CHs for all problems under a specific performance criterion and manufacturing environment. Additionally, the quality of solutions obtained by the CHs is not always as good as expected, especially for large-scale problems [2].

On the other hand, IHs, such as those developed by Akkan [8], Lewis and Slotnick [16], Yang and Geunes [23], Charnsirisakskul et al. [10], Slotnick and Morton [25], Cesaret et al. [24] and Xiao et al. [6], begin with an initial solution, which is then

* Corresponding author. Tel.: +886 2 2771 2171; fax: +886 2 2731 7168.
E-mail address: kcying@ntut.edu.tw (K.-C. Ying).

http://dx.doi.org/10.1016/j.asoc.2015.01.069
1568-4946/

iteratively improved so as to yield a (near-) optimal solution. In recent decades, the development of metaheuristic-based IHs for solving OAS problems has attracted increasing interest from both industry and academia. The existing metaheuristic-based IHs for solving OAS problems include extremal optimization [26], genetic algorithm [5,26,27], simulated annealing [4,6,27], Tabu search [24], hybrid evolutionary algorithm [26], hybrid artificial bee colony algorithm [28,29] and partial optimization algorithm [6]. Experimental results have shown that these metaheuristic-based IHs provide satisfactory solutions for various OAS problems; however, most of the studies dealt with the single machine OAS problem [2,30]. The reader is referred to the excellent reviews of the literature by Keskinocak and Tayur [30] and Slotnick [2] for a detailed discussion and taxonomy of the application models and available heuristics for solving various OAS problems.

The permutation flowshop scheduling problem (PFSP) is one of the most extensively studied problems in industry and it continues to be of interest to researchers and practitioners [31–33]. Owing to the nature of various industrial processes, many variants of the basic PFSP have been formulated and studied. Recently, Xiao et al. [6] formulated the permutation flowshop scheduling problem with order acceptance and weighted tardiness (PFSP-OAWT) as an integer programming model. They proposed two versions of the partial optimization algorithm (POA) and a simulated annealing based on the partial optimization (SABPO) algorithm to solve it. The first version of POA, called POA_O, begins with the sequence of orders. The second version of POA, POA_Y, begins with the acceptance states of the orders. Their computational results showed that SABPO outperformed all existing algorithms. To the best of our knowledge, the SABPO algorithm is the best available algorithm for solving the PFSP-OAWT with respect to maximizing the total net revenue.

The simulated annealing (SA) algorithm, which is a type of single solution-based metaheuristic, can be used for solving hard optimization problems. Due to the theoretical challenges of solving the OAS problem, an innovative multi-initiator simulated annealing (MSA) algorithm is herein proposed as a step toward developing a more efficient algorithm for solving the PFSP-OAWT. Two problems associated with any optimization algorithm are related to its convergence and escape local optimality [34,35]. The MSA algorithm exhibits the advantages of the SA algorithm as it effectively achieves search convergence. The multi-initiator method incorporates a powerful form of diversification in the generation of initial solutions to help escape local optimality, without which the SA algorithm may become confined to a small region of the solution space, making it hard to discover a global optimum.

The remainder of this paper is organized as follows. The problem is formulated in Section 2; in Section 3 the proposed MSA algorithm is described; using an existing benchmark problem set, the effectiveness and efficiency of the proposed MSA algorithm is empirically evaluated in Section 4 by comparing its performance with that of the traditional SA and the state-of-the-art SABPO algorithm; and, finally, in Section 5, conclusions are drawn and recommendations for future research are made.

2. Problem definition

This section defines the PFSP-OAWT and formulates it as an integer programming (IP) model. Before this, the following notations are defined to simplify the exhibition of this formulation.

Parameters

n	number of orders
m	number of machines
i	index of orders, $i \in \{1, 2, ..., n\}$
j	index of machines, $j \in \{1, 2, ..., m\}$
k	index of positions, $k \in \{1, 2, ..., n\}$
$o_{[k]}$	identification (ID) of the order that ranks in kth position, $k \in \{1, 2, ..., n\}$
$p_{i,j}$	processing time of order i on machine j
d_i	due date of order i
Q_i	maximum revenue of order i
w_i	per unit-time delay penalty of order i

Decision variables

$C_{i,j}$	completion time of order i on machine j
$C_{[k],j}$	completion time of the order that ranks in kth position on machine j
x_i	a binary index, which equals 1 if order i is accepted and otherwise equals 0
s_i	an integer variable denoting the sequence position of order i

Based on the above notations, the PFSP-OAWT can be formally defined as follows. A collection of n incoming orders (jobs) are to be processed on m machines in an identical technological order given by the indexing of the machines; the sequence in which all accepted orders are processed is the same on all machines, with both the limited production capacity and the order delivery requirements determining the acceptance of orders.

Each incoming order i is identified with the following non-negative data: the processing time required for each incoming order i ($i = 1, ..., n$) on machine j ($j = 1, ..., m$) $p_{i,j}$; a preferred due date, d_i, after which a tardiness penalty is incurred; a maximum revenue, Q_i, gained by the manufacturer if order i is accepted and its tardiness is zero; and a weight, w_i, which is the penalty per unit-time delay beyond d_i in the delivery to the customer. Since tardiness penalties result in a loss of revenue, order acceptance and scheduling decisions are necessary.

Based on the above definitions, the objective is to determine the orders to be accepted for processing and the production sequence for the accepted orders, with the aim to maximize the total net revenue (TNR), which can be formulated as follows:

$$TNR = \sum_{i=1}^{n} x_i(Q_i - w_i T_i)$$

where x_i is an indicator that equals 1 if order i is accepted, and 0 otherwise; and $T_i = \max\{0, C_{i,m} - d_i\}$ is the tardiness of order i, in which $C_{i,m}$ is the completion time of order i. Obviously, TNR equals the revenues from all accepted orders minus the total weighted tardiness penalties.

In the PFSP-OAWT considered in this study, the following assumptions have been made:

- Each machine is initially idle at the beginning of the scheduling period and can execute at most one accepted order at a time.
- Each accepted order is processed no more than once on each machine and can be processed by only one machine at a time.
- No preemptive priority is assigned. When the processing of an accepted order on a machine has begun, it must be completed before another accepted order can be processed on that machine.
- Each order is independent of every other order and is released to the shop at the beginning of the scheduling period.
- The setup times of the orders on machines are negligible.
- The machines are always available for processing all accepted orders throughout the scheduling period and no interruptions occur.
- If the next machine in the sequence needed by an accepted order is not available, then the order can wait in a stocking area and joins the queue at that machine.