Int. J. Production Economics 165 (2015) 12–18

Contents lists available at ScienceDirect

Int. J. Production Economics

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

A parallel neighborhood search for order acceptance and scheduling in flow shop environment

Deming Lei [a,*], Xiuping Guo [b]

[a] School of Automation, Wuhan University of Technology, Wuhan 430070, China
[b] School of Economic and Management, Southeast Jiaotong University, Chengdu, China

ARTICLE INFO

Article history:
Received 10 April 2014
Accepted 11 March 2015
Available online 24 March 2015

Keywords:
Parallel neighborhood search
Flow shop scheduling
Order acceptance
Multi-objective
Optimization

ABSTRACT

We consider the order acceptance and scheduling problem in a flow shop where the objective is to simultaneously minimize makespan and maximize total net revenue. We formulate the problem as a mixed integer linear programming model and develop an effective parallel neighborhood search algorithm. Two-string representation and three neighborhood structures are applied to generate new solutions. Parallelization is implemented by applying two independent searches and directly exchanging information between them. We assess the performance of the proposed method via computational experiments using an extensive set of instances. The experimental results show that the proposed method is highly effective and competitive when compared to other algorithms.

1. Introduction

In the traditional production scheduling models, it is often assumed that all jobs or orders must be processed through the production systems without rejection; however, in make-to-order (MTO) production systems, order acceptance and scheduling are often simultaneously considered. Some orders should be rejected if they cannot be managed effectively and delivered timely due to the limited capacities or resources. If some accepted orders cannot be scheduled effectively, then tardiness penalties will arise and the revenues will lost because of the customer dissatisfaction. The manufacturer should simultaneously determine which orders to be accepted and how to schedule them to use production capacity efficiently and optimize the performance indices such as total net revenue.

Over the past two decades, order acceptance and scheduling (OAS) problem has attracted considerable attention and diverse methods have been applied including mathematical programming, meta-heuristics, queuing theory, simulation, algorithm development and decision analysis (Slotnick, 2011). OAS has been considered in single machine (Slotnick and Morton, 1996, 2007; Rom and Slotnick, 2009; Oğuz et al., 2010; Talla Nobibon and Leus, 2011; Cesaret et al., 2012; Lin and Ying, 2013), parallel machines (Lu et al., 2008; Li and Yuan, 2010) and flow shop environments (Xiao et al., 2012; Wang et al., 2013a, 2013b). Slotnick and Morton (2007) study a branch-and-bound (BB) heuristic for small problems and a myopic heuristic based on relaxation for larger problems in single machine environments. Oğuz et al. (2010) give a mixed integer linear programming (MILP) formulation and three heuristics for OAS problem with release dates, due dates, deadlines, sequence dependent setup times and revenues in single machine environments. Talla Nobibon and Leus (2011) consider OAS in single machine where a pool consisting of firm planned orders and potential orders and present two MILP procedures and two BB algorithms. Cesaret et al. (2012) present a tabu search (TS) algorithm for OAS on a single machine with release dates and sequenced dependent setup times. Lin and Ying (2013) propose an artificial bee colony (ABC) algorithm for single machine OAS with release dates and sequence-dependent setup times for maximizing total net revenue.

Some studies have been done for OAS in multi-machine environments. Lu et al. (2008) define the problem of unbounded parallel batch machines with rejection and release dates and develop a pseudo-polynomial-time dynamical programming algorithm, a 2-approximation algorithm and an fully polynomial-time approximation schemes (FPTAS). Li and Yuan (2010) considered parallel machines scheduling problem where jobs can be rejected by paying penalties and propose two FPTAS and optimal dynamical programming algorithm for problems with different objectives. Xiao et al. (2012) study the permutation flow shop scheduling problem with order acceptance and weighted tardiness, formulate the problem as an integer programming model and present a simulated annealing based on partial optimization (SABPO). Wang et al., 2013a develop a modified ABC algorithm to make joint decisions on order acceptance and scheduling for maximizing total net revenue in a two-machine flow shop. Wang et al., 2013b

* Corresponding author. Tel.: +86 15327311013.
E-mail address: deminglei11@163.com (D. Lei).

http://dx.doi.org/10.1016/j.ijpe.2015.03.013
0925-5273/

formulate OAS problem in two-machine flow shop as a MILP models and develop a heuristic and BB algorithm based on some derived dominance rules and relaxation techniques. More details and extensive research results on OAS can be found in the recent survey papers (Slotnick (2011; Shabtay et al., 2013).

The first feature of the previous studies is that most of papers have been presented for OAS in single machine environment and OAS in flow shop is not considered fully. Only two-machine or permutation flow shop scheduling problems with order acceptance are investigated in recent years. OAS in flow shop with more than two machines is seldom dealt with. The second feature is that only one objective such as total net revenue is optimized in most of literature on OAS. To the best of our knowledge, OAS with multiple objectives is not studied. Thirdly, meta-heuristics have become the main approach to the classical flow shop scheduling; however, the application of meta-heuristics to OAS problem in flow shop is not done fully and only some meta-heuristics such as SABPO and ABC have been applied to solve the problem.

In this study OAS is considered in flow shop environment, which is composed of order selection sub-problem and scheduling sub-problem. An effective parallel neighborhood search (PNS) algorithm is applied to simultaneously minimize makespan and maximize total net revenue. An order permutation and a binary string are used to represent the solution of the problem and a method is applied to decide the number of the accepted orders. Two independent searches are used and allowed to exchange information between them if the condition is met. A novel principle is proposed to decide if the current solution can be replaced with the new one and a simple method is applied to update the set of non-dominated solutions produced by PNS. PNS is finally applied to the considered problem and computational results are analyzed.

The remainder of the paper is organized as follows. Problem under study and its mathematic programming model are described in Section 2. The proposed neighborhood search for the problem is shown in Section 3. Numerical test experiments on PNS are reported in Section 4 and the conclusions are summarized in the final section and some topics of the future research are provided.

2. Problem description and mathematical model

OAS problem in flow shop is commonly faced by the manufacturers that operate MTO production systems. The manufacturers have to make acceptance decisions for a number of candidate orders, which are associated with a due date, known revenue and deterministic processing times et al., and then schedule the accepted orders through a flow shop. Each accepted order should visit all machines according to a fix sequence. An accepted order can bring revenue; however, it also can incur a tardiness penalty if delivered later than its due date and there is no reward for early delivery.

The problem is formally described as follows: in a flow shop with m machines $M_1, M_2, \cdots, M_m$, a set of candidate orders $N=\{1,2,\cdots,n\}$ is available at time zero. Each candidate order i is identified with a due date d_i , a maximum revenue r_i , a weight (tardiness penalty) w_i and processing time p_{ik} on machine M_k. Each machine can only process one order at a time and each order can only be processed on a machine at a time. If the visiting sequence of all orders is $M_1, M_2, \cdots, M_m$, any order can begin its processing on M_k only after completing its processing on machine M_{k-1},$k \geq 2$. The goal of the problem is to decide which orders are to be accepted and then to schedule the accepted orders to simultaneously optimize makespan and total net revenue.

The related notations are first listed below.

S_a the set of the accepted orders
x_{ijk} a binary variable that indicates whether order i is processed on machine M_k

$$x_{ijk}=\begin{cases}1 & i\in S_a \text{ and the } j\text{th operation of order } i \text{ is processed on machine } M_k\\ 0 & \text{otherwise}\end{cases} \quad (1)$$

y_i a binary variable that indicates whether order i is accepted or rejected

$$y_i=\begin{cases}1 & \text{order } i \text{ is accepted}\\ 0 & \text{otherwise}\end{cases} \quad (2)$$

z_{ij} a binary variable that represents whether both orders i and j are accepted and order j is processed immediately after order i on the same machine

$$z_{ij}=\begin{cases}1 & i,j\in S_a, i \text{ precedes immediately } j \text{ on the same machine}\\ 0 & \text{otherwise}\end{cases} \quad (3)$$

C_{ik} the completion time of order i on machine M_k
θ an enough large positive number

The mathematical model of the problem is then shown as follows:

$$\text{Minimize } f_1=\max_{i\in S_a}\{C_{im}\} \quad (4)$$

$$\text{Maximize } f_2=\sum_{i=1}^{n} y_i \max\{0, e_i-w_i \max\{0, C_{im}-d_i\}\} \quad (5)$$

Subject to

$$\sum_{k=1}^{m} x_{ikk}=my_i \quad \forall i\in N \quad (6)$$

$$\sum_{j=1}^{m}\sum_{k=1}^{m} x_{ijk}\le 1 \quad \forall i\in N \quad (7)$$

$$z_{ij}\le y_i, z_{ij}\le y_j, \forall i,j\in N, i\ne j \quad (8)$$

$$C_{ik}+(z_{ij}-1)\theta+y_j p_{jk}\le C_{jk}, i,j\in N, i\ne j, k=1,2,\cdots,m \quad (9)$$

$$C_{i0}=0, \forall i\in N \quad (10)$$

$$C_{i(k-1)}+(x_{ikk}-1)\theta+y_i p_{ik}\le C_{ik}, \forall i\in N, k=1,2,\cdots,m \quad (11)$$

$$C_{ik}\ge 0, \forall i\in N, k=1,2,\cdots,m \quad (12)$$

$$x_{ijk}, y_i\in\{0,1\}, \forall i\in N, j,k=1,2,\cdots,m \quad (13)$$

The first objective (4) is maximum completion time of all orders. The second objective (5) is total net revenue. Constraints (6) and (7) indicate that each order visits all machines in the same sequence: $M_1, M_2, \cdots, M_m$. Constraint (8) represents that orders i and j must be accepted when $z_{ij}=1$. Constraint (9) indicates that if orders i and j are accepted and j is processed immediately after i on machine M_k, then the completion time of j on M_k is not less than $C_{ik}+p_{jk}$. Constraint (10) indicates that if order i is the first scheduled order on M_1, its beginning time is 0. Constraint (11) represents that if order i is accepted, then the completion of its kth operation is not less than $C_{i(k-1)}+p_{ik}$. Constraint (12) gives a lower bound on $C_{ik}$and constraint (13) represents the binary restriction of x_{ijk}, y_i.

Table 1 shows an illustrative instance with eight orders, 24 operations, processing times, due date and tardiness penalty.

For the problem with makespan (f_1) and total net revenue (f_2), the optimal result is not a single solution but a set of solutions; moreover, the optimal set cannot be obtained without comparing