European Journal of Operational Research 000 (2015) 1–13

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Order acceptance and scheduling problems in two-machine flow shops: New mixed integer programming formulations

Rasul Esmaeilbeigi, Parisa Charkhgard*, Hadi Charkhgard

School of Mathematical and Physical Sciences, The University of Newcastle, Australia

ARTICLE INFO

Article history:
Received 1 August 2014
Accepted 30 November 2015
Available online xxx

Keywords:
Order acceptance
Scheduling
Mixed integer programming
Preprocessing
Valid inequalities

ABSTRACT

We present two new mixed integer programming formulations for the order acceptance and scheduling problem in two machine flow shops. Solving this optimization problem is challenging because two types of decisions must be made simultaneously: which orders to be accepted for processing and how to schedule them. To speed up the solution procedure, we present several techniques such as preprocessing and valid inequalities. An extensive computational study, using different instances, demonstrates the efficacy of the new formulations in comparison to some previous ones found in the relevant literature.

1. Introduction

Many manufacturing companies use Make-To-Order (MTO) production systems. In MTO systems, planning for the manufacture of a product will begin only when a customer order is received. The main advantage of these systems is that they give rise to low finished goods inventories. However, these systems have a significant disadvantage in that the lead time for the fulfillment of orders may result in significant financial loss for companies because of the loss of business due to production limitations. As a consequence, to remain competitive, companies employing these systems must decrease their order delivery times. This can be achieved by employing an accurate production plan that determines which orders should be accepted and how they should be scheduled. The solution to the Order Acceptance and Scheduling Problem (OASP) is an important step in the development of such a plan.

OASPs have been studied extensively over the past 20 years and a number of different versions of these problems exist. We refer interested readers to the literature survey by Slotnick (2011) for details. Versions of OASPs in which the objective functions maximize the total net revenue, i.e. the difference between sum of revenues and total weighted tardiness or lateness, have been studied by many authors in a single-machine environment. Slotnick and Morton (1996) are believed to be the first researchers who addressed this problem under the assumption of static arrivals, meaning that all jobs are assumed to be available at time zero. They proposed two heuristic algorithms and a Branch and Bound (B&B) technique to solve the problem in this case.

Later, Ghosh (1997) proved that an OASP with lateness penalties is NP-hard. He also presented two pseudo-polynomial time dynamic programming algorithms, and a polynomial-time approximation scheme in order to solve the problem. Slotnick and Morton (2007) considered tardiness related penalties instead of lateness penalties. They developed a B&B algorithm and a number of heuristics to solve this problem exactly with at most 10 jobs in about 6000 seconds on average. As far as we know, the largest instances of OASP with tardiness related penalties in a single-machine environment were solved by Nobibon and Leus (2011). They proposed two Mixed Integer Linear Programming (MILP) formulations and could solve instances of the problem with at most 50 jobs to optimality within two hours using the IBM ILOG CPLEX Optimizer (see http://www-01.ibm.com/software/info/ilog). In order to compute high quality solutions for large size instances of the problem, Rom and Slotnick (2009) developed a genetic algorithm. They showed that while their proposed approach is slower than the available heuristics appearing in the relevant literature, it generates solutions of higher quality.

Oğuz, Salman, and Yalçın (2010) added more assumptions to the OASP with tardiness related penalties in a single machine environment. They considered release dates for each job and sequence dependent setup times. They gave a MILP formulation of the problem and could solve instances of the problem with at most 15 jobs to optimality. To compute high quality solutions for larger size instances of the problem, they also developed three heuristics. Later, Cesaret, Oğuz, and Salman (2012) developed a tabu search algorithm for this problem. They showed that their proposed algorithm is faster and can provide solutions with higher quality when compared with previous

* Corresponding author. Tel.: +61 2 4926 1326.
E-mail address: parisa.charkhgard@uon.edu.au (P. Charkhgard).

http://dx.doi.org/10.1016/j.ejor.2015.11.036
0377-2217/

heuristics. Lin and Ying (2013) introduced a new artificial bee colony based algorithm to solve this problem. Their experimental results indicated that their proposed heuristic is competitive with the algorithm by Cesaret et al. (2012).

There are also a few studies about the OASP with tardiness related penalties in an m-machine permutation flow shop environment. For example, Xiao, Zhang, Zhao, and Kaku (2012) developed a partial optimization based simulated annealing algorithm to solve instances of the problem. Later, Lin and Ying (2015) presented a multi-initiator simulated annealing algorithm, and showed that the new heuristic outperforms the algorithm by Xiao et al. (2012). Recently, Lei and Guo (2015) addressed the biobjective version of the problem where the objectives are minimization of the makespan and maximization of the total net revenue. To solve instances of the problem, they employed a parallel neighborhood search algorithm and compared it with a tabu search and a variable neighborhood search algorithm.

In this paper, we consider the OASP in a 2-Machine Flow shop environment (OASP-2MF) which is recently addressed by Wang, Xie, and Cheng (2013a) and Wang, Xie, and Cheng (2013b). In Wang et al. (2013b), the authors tried to generalize the work of Slotnick and Morton (2007). They introduced two MILP formulations which could solve instances of the problem with up to 13 jobs within a one hour time limit using CPLEX. In addition, they proposed a B&B algorithm which simultaneously took into account job selection and scheduling and benefited from the use of some dominance rules in the pruning procedure. They showed that their purpose-built solver can solve larger sized instances of the problem with up to 20 jobs within an hour. In Wang et al. (2013a), the authors developed a modified artificial bee colony algorithm to compute good solutions for even larger instances of the problem.

The main contribution of our research is the development of two new MILP formulations for the OASP-2MF. In addition, to speed up the solution procedure, we present several enhancements (cuts and preprocessing techniques) which can reduce the size of the problem significantly and make the formulations stronger. Our new formulations have the following three desirable characteristics:

- The number of variables and constraints in these formulations is quadratically bounded by the number of jobs. Some previous researchers have developed time-indexed formulations for single-machine or 2-machine flow shop versions of the OASP, but the size of these formulations can increase dramatically if processing times are large.
- They outperform previous formulations even before applying the enhancements.
- CPLEX can solve instances of the problem that are 5 times larger than those solved by the purpose-built solver which is developed in Wang et al. (2013b).

We compare our new formulations after applying the enhancements and show that one of them performs far better than the other. Using our best formulation, CPLEX can achieve an optimality gap of less than 2 percent, on average, within 1800 seconds, even for instances of OASP-2MF with as many as 100 jobs.

The rest of the paper is organized as follows. In Section 2, we review some preliminary notation and results. In Section 3, we introduce two new MILP formulations for OASP-2MF. In Section 4, we discuss enhancements to make the formulations stronger. In Section 5, we report the results of a comprehensive computational study. Finally, in Section 6, we give some concluding remarks.

2. Preliminaries

In an OASP-2MF two decisions must be made at the same time: which orders to be accepted for processing and how to schedule them. We assume that the set of orders (*jobs*) is known in advance. Due to the flow shop structure of the problem, each job can be processed on machine 2 at some time after its processing on machine 1 has been completed.

We denote the set of jobs by $N = \{1, 2, \ldots, n\}$. The revenue and processing times of each job $i \in N$ on machines 1 and 2 are denoted by $u_i \in \mathbb{Z}^+$, $p_i^1 \in \mathbb{Z}^+$ and $p_i^2 \in \mathbb{Z}^+$ (where we use $\mathbb{Z}^+$ to denote the set of positive integers), respectively. We sometimes sort the processing times on each machine from small to large. We use $p_{[i]}^1$ and $p_{[i]}^2$ to denote the processing times in the ith position of the sorted lists, for machines 1 and 2, respectively. We denote the completion time of job $i \in N$ by C_i. We assume that each job $i \in N$ has a due date, denoted by $d_i \in \mathbb{Z}^+$, and that there is a delay penalty, denoted by $w_i \in \mathbb{Z}^+$, for each unit of the completion time which exceeds d_i, i.e., $C_i - d_i$ (note that due dates are positive integers). Also, there is no reward or penalty for early delivery. We sometimes refer to the delay time of the job $i \in N$ as its *tardiness*, and denote it by T_i, i.e., $T_i = \max\{0, C_i - d_i\}$. The *net revenue*, i.e., the difference between the revenue and the delay cost, of each job $i \in N$ is defined by $\pi_i := u_i - w_i \cdot T_i$. Furthermore, we assume that the goal is to maximize the total net revenue in the OASP-2MF.

Observe that, in an optimal solution, if job $i \in N$ is accepted, then $\pi_i \geq 0$. Moreover, $\pi_i = u_i$ if job $i \in N$ is accepted and fulfilled before its due date, and $\pi_i < u_i$ if job $i \in N$ is accepted and fulfilled after its due date. We sometimes refer to $u_i - \pi_i$ (or equivalently $w_i \cdot T_i$) as the *tardiness penalty* for job $i \in N$, if it is accepted. The following propositions provide the basis for the development of the different MILP formulations and enhancements in this paper. Note that Propositions 1 and 3 are straight forward to prove, and they are known results in the literature of the classical 2-machine flow shop problem (see for instance Baker, 1974; Kim, 1993). Therefore, we have omitted their proofs. Moreover, it should be mentioned that Wang et al. (2013b) used Proposition 1 to validate their MILP formulations.

Proposition 1. *For accepted jobs, there is an optimal schedule in which each job is processed on both machines in the same sequence.*

Proposition 2. *In an optimal schedule, $C_i^U := \frac{u_i}{w_i} + d_i$ is an upper bound for the completion time of an accepted job $i \in N$.*

Proof. Suppose that the assertion is not true. Therefore, there must exist a job $i \in N$ in an optimal schedule whose completion time C_i is strictly larger than C_i^U. The net revenue of job i is

$$\pi_i = u_i - w_i \cdot \max\{0, (C_i - d_i)\}.$$

Because $C_i^U < C_i$,

$$\pi_i < u_i - w_i \cdot \max\{0, (C_i^U - d_i)\}$$
$$= u_i - w_i \cdot \max\left\{0, \left(\frac{u_i}{w_i} + d_i - d_i\right)\right\} = 0.$$

This contradicts our assumption that the schedule is optimal since we can improve the total net revenue by simply rejecting job i. □

Note that Proposition 2 is independent of machine environment and processing restrictions.

Proposition 3. *Let S be the list of accepted jobs for a schedule and suppose that the jobs will be processed in the same order as they appear in the list. Let $J_k \in S$ be the job which is allocated to position k in the list S where $1 \leq k \leq |S|$ and $|S|$ is the number of elements of S. Then*

$$C_{J_k}^L := \max\left\{\sum_{1\leq q\leq k} p_{J_q}^1 + p_{J_k}^2, \sum_{1\leq q\leq k} p_{J_q}^2 + p_{J_1}^1\right\}$$

is a lower bound for the completion time of job J_k.

3. New formulations

In this section, we describe two new MILP formulations for the OASP-2MF. We then compare them with two Previous Formulations