European Journal of Operational Research 000 (2015) 1–16

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

A real-time order acceptance and scheduling approach for permutation flow shop problems

Humyun Fuad Rahman*, Ruhul Sarker, Daryl Essam

School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia

ARTICLE INFO

Article history:
Received 4 June 2014
Accepted 8 June 2015
Available online xxx

Keywords:
Real-Time multiple-order scheduling
Flow shop scheduling
Random order arrival
Genetic algorithm
Memetic algorithm

ABSTRACT

The Permutation Flow Shop Scheduling Problem (PFSP) is a complex combinatorial optimization problem. PFSP has been widely studied as a static problem using heuristics and metaheuristics. In reality, PFSPs are not usually static, but are rather dynamic, as customer orders are placed at random time intervals. In the dynamic problem, two tasks must be considered: (i) should a new order be accepted? and (ii) if accepted, how can this schedule be ordered, when some orders may be already under process and or be in the queue for processing? For the first task, we propose a simple heuristic based decision process, and for the second task, we developed a Genetic Algorithm (GA) based approach that is applied repeatedly for re-optimization as each new order arrives. The usefulness of the proposed approach has been demonstrated by solving a set of test problems. In addition the proposed approach, along with a simulation model, has been tested for maximizing the revenue of a flow shop production business under different order arrival scenarios. Finally, a case study is presented to show the applicability of the proposed approach in practice.

1. Introduction

The Permutation Flow Shop Scheduling Problem (PFSP) is one of the challenging scheduling problems that occurs in the manufacturing industries. A conventional PFSP considers how to process n jobs on m machines. Each job has predefined tasks that are processed by a specific set of machines through a specific processing order. In solving PFSPs, makespan minimization is a common and popular measure of performance. The time difference between the start of the first job in the first machine, and the end of the last operation in the last machine, can be defined as makespan. In order acceptance and scheduling problems, the objective is to maximize the number of accepted orders, while minimizing the order completion times. PFSPs can be categorized as either single-order or multiple-order. In this research, we assume each order contains a certain number of jobs. The basic difference between a single-order and a multiple-order problem is that in a single order problem the decision maker has to determine an effective schedule for a single order (a given set of jobs in the order) on a set of machines with known sequence of operations and processing times. Whereas, in a multiple-order problem, the decision maker has to face a stream, or pool, of orders which are scheduled on a set of machines, where the scheduler has the option of accepting or rejecting the arriving orders (Slotnick, 2011). Besides, the single-order problem is static and the multiple-order problem is either static or dynamic. In a multiple-order static problem, the order arrival times and due dates are known well in advance. For static problems, either single or multiple-order, it is expected to solve a problem only once. However, in dynamic multiple-order problems, it is assumed that the order arrivals continue with time (on a real-time basis), and the problem is to select a set of orders that would be feasible for processing within the available shop capacity, and to determine an effective schedule for the jobs of those selected orders over a given period of time. In this case, the order selection must be done immediately after the arrival of any new orders, and the job schedule must be updated, if any order is accepted.

Single-order PFSPs have been widely studied in the literature. First, in 1954, Johnson (1954) introduced the flow shop problem as an interesting scheduling problem and proposed a simple algorithm that guaranties the optimal solution for a two machines static flow shop problem, and for a special case with a three machine problem, in polynomial time. For solving PFSPs with three or more machines, many researchers used exact techniques such as mixed integer programming (Selen & Hott, 1986), and Branch and Bound (B&B) algorithms (Ignall & Schrage, 1965). However, as the single-order PFSP is *NP Hard* (when the number of machines is three or more) (Garey, Johnson, & Sethi, 1976), researchers have focused on heuristic techniques (Ruiz & Maroto, 2005). Among recent heuristics, Nawaz et al.'s (Nawaz, Enscore, & Ham, 1983) NEH algorithm is regarded as one of the best constructive heuristics for solving static PFSPs. However, it still deviates by up to 7 percent from the known optimum for some

---

* Corresponding author. Tel.: +61 262688176
E-mail address: md.rahman4@student.adfa.edu.au (H.F. Rahman).

http://dx.doi.org/10.1016/j.ejor.2015.06.018

Please cite this article as: H.F. Rahman et al., A real-time order acceptance and scheduling approach for permutation flow shop problems, European Journal of Operational Research (2015), http://dx.doi.org/10.1016/j.ejor.2015.06.018

problems (Ruiz & Maroto, 2005; Taillard, 1990; Zobolas, Tarantilis, & Ioannou, 2009). This heuristic is based on the idea that longer jobs in the sequence should be processed as early as possible in the schedule. To improve the solution quality of PFSPs, researchers have switched their attention to metaheuristics, such as the Simulated Annealing (SA) algorithm (Ogbu & Smith, 1990; Osman and Potts, 1989), Cuckoo search algorithm (Dasgupta & Das, 2015), Genetic Algorithms (GAs), including Hybrid GAs (Murata, Ishibuchi, & Tanaka, 1996; Rahman, Sarker, & Essam, 2013; Ruiz, Maroto, & Alcaraz, 2006; Tseng & Lin, 2009; Zobolas et al., 2009), Ant colony algorithms (Rajendran & Ziegler, 2004), particle swarm optimizations (Tasgetiren, Liang, Sevkli, & Gencyilmaz, 2007), tabu search (Grabowski & Wodecki, 2004), and differential evolution (Onwubolu & Davendra, 2006). From the computational results provided in the literature, the hybrid metaheuristics, in general, show promising performance.

Many manufacturing firms receive a stream of orders (or a certain pool of orders) from which certain orders may be accepted and scheduled with respect to the available production capacity (Slotnick, 2011). If a firm either accepts a new order without checking its feasibility for on-time completion, or cannot schedule all the accepted orders for on-time completion, it will produce a poor production plan that would lead to reduced revenue (Guerrero & Kern, 1988). The order acceptance/rejection problem has been studied mainly in a single machine environment as a static problem, where the order arrival times are known well in advance (Lewis & Slotnick, 2002; Rom & Slotnick, 2009; Slotnick & Morton, 1996, 2007). A brief review of the single machine static acceptance/rejection problem is provided here. Slotnick and Morton (1996) proposed an integer programming algorithm with the objective of maximizing the profit. Lewis and Slotnick (2002) applied dynamic programming for a multi-period job selection process where job rejection involves future loss of customer. Slotnick and Morton (2007) later extended the problem of (Lewis & Slotnick, 2002; Slotnick & Morton, 1996) for limited capacity. In their work, a branch and bound algorithm was proposed to deal with order acceptance decisions and several heuristics were used to sequence the jobs to minimize weighted tardiness. Rom and Slotnick (2009) extended the previous approach of order acceptance and scheduling decision with lateness penalties (Lewis & Slotnick, 2002; Slotnick & Morton, 1996) and weighted tardiness (Slotnick & Morton, 2007). They proposed a GA which minimized weighted tardiness, and it performed well with respect to a previously proposed heuristic (Slotnick & Morton, 2007), even though it took more time in computation. Nobibon and Leus (2011) considered a problem where a company has to select orders from a pool of firms planned orders, as well as any other demanded orders. Wang, Zhu, and Cheng (2015) studied a subcontracting price scheme for the static order acceptance and scheduling problem in a single machine environment. The multi-order static problem has also been studied in a multiple machines environment (Chen, Mestry, Damodaran, & Wang, 2009; Pourbabai, 1989; Roundy et al., 2005; Wang, Huang, Hu, & Cheng, 2015; Wang, Xie, & Cheng, 2013a , 2013b). Wang et al. (2013a) developed a modified artificial bee colony algorithm for solving the order acceptance problem in two machine static multiple-order PFSPs. Wang et al. (2013b) proposed a B&B algorithm and a heuristic to solve the order acceptance problem in two machine static multiple-order PFSPs. Xiao, Zhang, Zhao, and Kaku. (2012) studied a static multiple-order PFSP with order acceptance and weighted tardiness problems. Lin and Ying (2015) proposed a multi-initiator SA for the same problem, and the experimental results showed that the proposed algorithm outperforms Xiao et al. (2012)'s approach. In both studies, each order contained a single job and at the beginning of the planning period, the firm received a pool of candidate orders with known arriving times, order compositions, and due dates. Wang, Huang et al. (2015) proposed a Lagrangian relaxation technique based exact algorithm, and two heuristics to solve the order acceptance problem in a static multiple-order two identical parallel machine problem. Chen et al. (2009) addressed static order arrival in a job shop environment by using a mixed integer programming approach for smaller problems, and a B&B algorithm with Lagrangian bounds and approximate branching features for larger problems. Pourbabai (1989) developed a model to identify potential orders, order splitting considering due dates, and job set up, and scheduled jobs using a dispatch rule based on order availability and due dates on a multiple machine environment where the machines are grouped into cells (group technology concept). Roundy et al. (2005) considered a job shop environment, in which an order is accepted, if it can in any way be inserted into the current schedule. They developed both a single machine heuristic, as well as meta-heuristics (tabu search, GA, SA), to solve the problem.

There are a few studies that have considered dynamic order arrival in a single machine environment. Wester, Wijngaard, and Zijm (1992) studied the relationship between three different order acceptance strategies: order acceptance based on the knowledge of previously accepted orders, order acceptance based on the total workload of all accepted orders, and order acceptance based on the aggregated load profile of accepted orders, and scheduling jobs to maximize the utilization of capacity. In this case, the authors found that using knowledge of the current production schedule when generating new schedules because of new order arrivals, was superior over the other two approaches. Duenyas and Hopp (1995) considered that order arrival and processing times were stochastic. In that study, an arriving order was only rejected if it was beyond the customer's tolerance limits. Later, Duenyas (1995) extended the work of Duenyas and Hopp (1995) to consider customer quoted due dates.

The next level of complexity is dynamic order arrival in a multiple machine environment, which is closer to the research presented in this paper. There is no doubt that this topic is much more complex and has more synergies with practical situations. Nandi and Rogers (2004) proposed simulation based order acceptance and scheduling decisions for two product types (regular and urgent), with profit maximization as the objective in a four stage hybrid flow shop environment. The order acceptance was done by pair look simulation (based on the total contribution in companies profit, if an order arrives and whether it is accepted or rejected), and scheduling was done by minimizing the amount of slack per operation remaining. Rogers and Nandi (2007) used a simulation tool to maximize the profit with a fixed capacity in a four stage hybrid flow shop. The scheduling was done by using dispatching rules (first come first serve rule, earliest due date, minimum slack per operation remaining). Moreira and Alves (2009) also used simulation to investigate multiple decision making (order acceptance, due date adjustment, order releasing and scheduling) in a job shop environment to improve lateness penalties and workload performance. Scheduling was generated by using both the earliest due date rules and also the first come first serve rules. Tang, Liu, and Liu (2005) proposed a neural network that integrated six priority rules, for the hybrid dynamic flow shop problem with the objective of minimizing the average flow time, average number of tardy jobs, and average tardy time. In that work, jobs arrival (each order containing a single job) was assumed to follow a Poisson distribution. Kang, Duffy, Shires, Smith, and Novels (2014) developed an integrated approach, based on the concept of advanced planning and scheduling, with a closed looped methodology for Lean-scheduling of practical dynamic semiconductor and cable manufacturing environments.

Pinedo (2012) criticized that most of the theoretical models for multiple machine scheduling made an assumption of scheduling an n set of jobs in an m set of machines, while in an actual manufacturing environment, the jobs of an order may be processed at any specific time, because orders may be placed by the customers at any point in time, i.e. new orders arrive in the system randomly. Machine and resource availabilities also change with time (Schmidt, 2000). The static single-order and static multiple-order PFSPs thus ignore the status of the manufacturing shop floor. In addition, each order has

Please cite this article as: H.F. Rahman et al., A real-time order acceptance and scheduling approach for permutation flow shop problems, European Journal of Operational Research (2015), http://dx.doi.org/10.1016/j.ejor.2015.06.018