A genetic algorithm based heuristic for two machine no-wait flowshop scheduling problems with class setup times that minimizes maximum lateness

King-Wah Pang*

Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

ABSTRACT

Machine scheduling problem has been extensively studied by researchers for many decades in view of its numerous applications on solving practical problems. Due to the complexity of this class of scheduling problems, various approximation solution approaches have been presented in the literature. In this paper, we present a genetic algorithm (GA) based heuristic approach to solve the problem of two machine no-wait flowshop scheduling problems that the setup time on the machines is class dependent, and the objective is to minimize the maximum lateness of the jobs processed. This class of machine scheduling problems has many practical applications in manufacturing industry, such as metal refinery operations, food processing industry and chemical products production processes, in which no interruption between subsequent processes is allowed and the products can be grouped into families. Extensive computation experiments have been conducted to evaluate the effectiveness of the proposed algorithm. Results show the proposed methodology is suitable to be adopted for the development of an efficient scheduling plan for this class of problems in real life application.

1. Introduction

The problem of machine scheduling, with its extensive application and large impact on operating efficiency in both manufacturing and service industries, is one of the classical research topics that has been investigated since the 1950s. Variations of this class of scheduling problems include the number of machines in the system, number of stages the jobs need to be processed, job processing requirement, and the objective functions for optimization. In this paper, we study a variant of the classical machine scheduling problem, i.e., the two machine no-wait flowshop scheduling problem with class dependent setup time that minimizes the maximum lateness of the jobs. This class of machine scheduling problems has numerous applications in the steel processing industry where the metal slab must pass to the next operation without delay in order to maintain the temperature of the steel for further processing. Also, the dies for the forming process is suitable for a group of products that share the same physical characteristics, i.e., diameter of the pipe to be produced; setup is required only when a new product group is processed. Besides, the food processing industry also requires the materials are processed with no delay between processes, such as the canning operation must follow directly after the cooking process of the food in the canned food processing factory. Many other applications can be found that characterize this no-wait flowshop scheduling system, such as Just-in-time operations and chemical processing.

In recent years, a considerable amount of research study has been devoted to the machine scheduling problems that consider no-wait flowshop operations. That is, the job has to be processed in the succeeding operation right after it finishes its processing in the preceding operation. This processing requirement applies to many production systems such as metal processing (Callahan, 1971), food processing, chemical processing (Reklaitis, 1982) and computer system operations (Reddi & Ramamoorthy, 1972). Rajendran (1994) presented a heuristic for a no-wait flowshop scheduling problem to minimize the makespan. The author compared several heuristics on this class of scheduling problems that considers up to 25 stages in the flowshop. Comprehensive survey on this class of machine scheduling problem can be found in Hall and Sriskandarajah (1996). Ruiz and Allahverdi (2009) recently presented the dominance relation for the three machine no-wait flowshop scheduling problems. They also develop several heuristic approaches based on the dominance relation as part of their heuristic. For the general flowshop scheduling problems, Gupta and Stafford (2006) provided a review on the study of this topic over the last 50 years.

Besides the extensive research study on the no-wait flowshop scheduling problems, many researchers have also found interest

* Tel.: +852 2766 4126; fax: +852 2330 2704.
  E-mail address: lgtapang@polyu.edu.hk

in investigating the impact of the setup time on the machine before the jobs can be processed. And commonly, the setup time of the jobs is class dependent, and processing several jobs within the same product class requires the machine to be set up only once. This concept aligns with the product family grouping situation where similar products are grouped into a product family that requires the same setup on the machines to process this group of products. In such a situation, the scheduling of the jobs to be processed requires additional consideration on the setup time of the machines, where more products in the same class that are processed together can be beneficial to reduce the total setup time required, because this setup time is considered as non-productive time that should be minimized. However, we may consider to separate the processing of the products within the same class into several batches in sequence, especially when the delivery due date of the jobs is considered, in order to minimize the job lateness. One of the earlier studies on a two machine flowshop problem with separated setup time that considers job lateness is presented by Dileepan and Sen (1991). Later, Sidney et al. (2000) proposed a heuristic for solving a two machine no wait flow shops scheduling problem with anticipatory setup time on the second machine. The heuristic has a worst-case performance bound of 4/3 of the optimal solution on minimizing the makespan. A review on scheduling problem with setup consideration can be found in Allahverdi et al. (1999). Genoulaz (2000) presented several heuristics for solving the V-stage hybrid flow shop problems subject to job-precedence constraints, minimum time lags, setup and removal times and due dates, with the objective of minimizing the maximum lateness. In 2003, Lin and Liao (2003) presented a recursive based heuristic for solving the two-stage hybrid flow shop problem that considers sequence-dependent setup time and minimizes the weighted maximal tardiness. Al-Anzi and Allahverdi (2007) proposed a self-adaptive differential evolution heuristic for a two-stage assembly scheduling problem to minimize maximum lateness with setup times. They considered the case where there are multiple machines available at the first stage of the assembly operations, and also assumed the jobs can wait to be processed at the second stage. Later, they presented another paper to propose several heuristics for solving a two-stage assembly flowshop problem with a combined objective to minimize both the makespan and maximum lateness (Al-Anzi and Allahverdi, 2009). A survey on scheduling problems with batching or class based setup times can be found in Potts and Kovalyov (2000) and Allahverdi et al. (2008). Recently, Moslehi et al. (2009) presented a branch-and-bound based algorithm for solving the two machine flowshop scheduling problems that minimizes the sum of maximum earliness and tardiness.

According to the three-field $\alpha/\beta/\gamma$ classification scheme proposed by Graham et al. (1979), the proposed scheduling problem we study in this paper is defined as $F2/ST_{si,b}$, no-wait/L_{max}, which denotes a two-machine no-wait flowshop problem with sequence independent batch setup time that minimizes the maximum lateness of the jobs. There are only a few studies found in the literature that addresses this specific problem. Wang and Cheng (2006) proposed a heuristic approach for solving this problem that uses forward merge and backward merge policies, which is based on an earliest due date initial schedule. Besides this article, the authors are not aware of any other study on this specific problem. The remaining sections of this paper are organized as follows: The problem description is presented in Section 2. In Section 3, we introduce the genetic algorithm based heuristic and detail the operation flow of our proposed approach. Section 4 presents the computational experiments including the data generation and computation results. Finally, we draw some concluding remarks and possible future direction in Section 5.

2. Problem description

We first describe the model of the machine scheduling problem that we study in this paper, by defining the variables and parameters of the model, and then the objective function as the minimization of the maximum lateness of the jobs. According to the problem definition introduced in the last section, we are given a set of n jobs to be scheduled in a two-machine no-wait flowshop, these n jobs are grouped into f job families, where each job family f_j has a sequence-independent setup time on the two machines, i.e., s_j^m where $m = 1\,or\,2$ for the two machines. For simplicity, the n jobs are equally grouped into f job families where each family has $l = n/f$ jobs. We simply group Job 1 to Job l to family 1, Job $l+1$ to Job $2l$ into family 2, and so forth. We further define the following variables:

p_i^m is the processing time of job i on machine m, $m = 1\,or\,2$.
C_i^m is the completion time of job i on machine m, $m = 1\,or\,2$.
d_i is the due date of job i.

To determine the job completion time, we let the bracket denote the job in a given position of a job sequence. For example, $p_{[i]}^1$ denotes the processing time of the ith job on machine 1 in a particular job sequence. With this convention, assuming there is a job sequence σ, the completion time of ith job on machines 1 and 2 can be determined respectively, as follows:

If the ith job in the schedule σ belongs to the same family of the job $i-1^{th}$ preceding it:

Under the no-wait two machine flowshop system, the completion time of the ith job on machine 1:

$$C_{[i]}^1 = max(C_{[i-1]}^1 + p_{[i]}^1, C_{[i-1]}^2) \tag{1}$$

Completion time of the ith job on machine 2:

$$C_{[i]}^2 = C_{[i]}^1 + p_{[i]}^2 \tag{2}$$

If the ith job belongs to a different family k, while job $i-1^{th}$ preceding it belongs to family j:

Completion time of the ith job on machine 1:

$$C_{[i]}^1 = max(C_{[i-1]}^1 + s_k^1 + p_{[i]}^1, C_{[i-1]}^2 + s_k^2) \tag{3}$$

Completion time of the ith job on machine 2:

$$C_{[i]}^2 = C_{[i]}^1 + p_{[i]}^2 \tag{4}$$

where the 1st job in the schedule σ is the job i that belongs to family j: Completion time of the 1st job on machine 1:

$$C_{[i]}^1 = max(s_j^1 + p_{[i]}^1, s_j^2) \tag{5}$$

Completion time of the 1st job on machine 2:

$$C_{[i]}^2 = C_{[i]}^1 + p_{[i]}^2 \tag{6}$$

We further define the job lateness of ith job in a sequence as $L_{[i]} = C_{[i]}^2 - d_{[i]}$. And the objective is to minimize the maximum lateness of all jobs on the schedule.

$$L_{max} = \max_{i = 1,\dots,n} L_{[i]} \tag{7}$$

Fig. 1a–d illustrates the methods to determine the completion time of the ith job in a sequence. In a two machine no-wait flowshop environment, the job must be processed on the 2nd machine once it finishes the operation on the 1st machine. Also, if the 2nd machine is not available to process the job, the 1st machine operation of that job must be delayed in order to wait for the availability of the 2nd machine so that there is no interruption between the processing of the job on the two machines.