Applied Mathematical Modelling 35 (2011) 4117–4126

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Two meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness

İnci Sarıçiçek *, Cenk Çelik

Eskişehir Osmangazi University, Department of Industrial Engineering, Turkey

ARTICLE INFO

Article history:
Received 24 September 2009
Received in revised form 4 February 2011
Accepted 17 February 2011
Available online 23 February 2011

Keywords:
Parallel machine scheduling
Total tardiness
Job splitting
Tabu search
Simulated annealing

ABSTRACT

Parallel machine scheduling is a popular research area due to its wide range of potential application areas. This paper focuses on the problem of scheduling n independent jobs to be processed on m identical parallel machines with the aim of minimizing the total tardiness of the jobs considering a job splitting property. It is assumed that a job can be split into sub-jobs and these sub-jobs can be processed independently on parallel machines. We present a mathematical model for this problem. The problem of total tardiness on identical parallel machines is NP-hard. Obtaining an optimal solution for this type of complex, large-sized problem in reasonable computational time by using an optimization solver is extremely difficult. We propose two meta-heuristics: Tabu search and simulated annealing. Computational results are compared on random generated problems with different sizes.

1. Introduction

Parallel machine scheduling (PMS) has been a popular research area due to its wide range potential application areas. This popularity has been increased considerably during recent years by the emergence of parallel processor computer technology [1]. A bank of machines in parallel is a setting that is important from both the theoretical and practical points of view. From the theoretical viewpoint, it is a generalization of the single machine, and a special case of the flexible flow shop. From the practical point of view, it is important because the occurrence of resources in parallel is common in the real-world. Furthermore, techniques for machines in parallel are often used in decomposition procedure for multistage systems [2].

One may consider parallel machine scheduling as a two-step process. First, one has to determine which jobs are to be allocated to which machines; second, one has to determine the sequence of the jobs allocated to each machine. With the makespan objective only the allocation process is important [2]. Although problems with due date-related performance measures were studied in many research articles, not much progress has been made for the objective of minimizing total tardiness in parallel machine scheduling problems except for special cases of identical parallel machines [3]. Furthermore, since it is not easy to obtain optimal solutions for parallel machine tardiness problems of a practical size, researchers have focused on the development of heuristic algorithms [4].

There are very few research results on the parallel machine scheduling problem with job-splitting properties. Serafini [5], studied cases in which each job can be split arbitrarily (into sub-jobs of continuous units, not discrete units) and processed independently on uniform or unrelated parallel machines and give heuristic algorithms for the objective of minimizing the maximum weighed tardiness. For an identical processor problem in which each job can be split arbitrarily, Xing and Zhang [6] propose a heuristic algorithm for the objective of minimizing makespan and analyze the worst case performance of the

* Corresponding author. Tel./fax: +90 222 239 3750.
E-mail address: incid@ogu.edu.tr (İ. Sarıçiçek).

0307-904X/$ - see front matter
doi:10.1016/j.apm.2011.02.035

algorithm. Tahar et al. [7] presented a method based on linear programming for identical parallel machine scheduling with job splitting and sequence-dependent setup times.

Kim et al. [4], studied cases in which each job can be split into a discrete number of sub-jobs and they are processed on a parallel machine independently. They suggested a two-phase heuristic algorithm for the parallel machine scheduling problem with a job splitting property for the objective of minimizing total tardiness. Shim and Kim [3] suggested a B&B algorithm for the identical machine scheduling problem with the objective of minimizing total tardiness considering the job splitting property described by Kim et al. [4]. They developed dominance properties and lower bounds and used them in their B&B algorithm. Results of their computational experiments showed that their suggested algorithm was able to find optimal solutions for problems with up to four machines and 12 jobs (and five machines and eight jobs) in a reasonable amount of CPU time. According to this paper, one can devise heuristic algorithms that give very good solutions for large problems within a reasonably short computation time.

Although job splitting reduces flow times or completion times of certain jobs, it increases overall set-up frequency and delays completion of other jobs. In other words, the more sub-jobs into which a job is split, the shorter the flow time of the job tends to become, since those sub-jobs can be processed simultaneously on different machines. However, more set-ups may be required if there are more sub-jobs. Hence, in the scheduling problem considered in this study, it is very important to find an appropriate number of sub-jobs to be split from each job [4].

In this paper, we focus on the problem of scheduling n independent jobs on m identical parallel machines with the objective of minimizing total tardiness of the jobs considering a job splitting property. It is assumed that a job can be split into sub-jobs and these sub-jobs can be processed independently on parallel machines. The problem is formulated as a mixed integer programming model. The results of tabu search are compared with simulated annealing from the point of objective function values and CPU computation times.

The remainder of this paper is organized as follows. In the next section, we introduce the identical parallel machine scheduling problem and the problem is formulated as a mixed integer programming model. The problem is solved by tabu search and simulated annealing algorithm in Section 3. The effectiveness of the algorithms is examined in Section 4 by several computational experiments. Finally, our results are summarized in Section 5.

2. Parallel machine scheduling problem

2.1. The definition of the problem

In this paper, we focused on the problem of scheduling n independent jobs on m identical parallel machines with the objective of minimizing total tardiness of the jobs considering a job splitting property. In certain manufacturing shops with parallel machines, each job can be regarded as a production order to process a product type in a specified quantity by a given due date. In scheduling problems for those shops, jobs can be split into a number of sub-jobs that can be processed independently on two or more parallel machines at the same time [3]. This type of problems is called as parallel machine scheduling problem with splitting jobs [3,5–7]. Indeed, a job can be split into an arbitrary number of parts and each part processed on a different machine. This problem arises, for example, in the textile industry where a job represents a batch (for example, a batch of 1500 socks) and the job splitting property allows decomposing each batch into sub-batches [8].

This research is motivated by a practical need at a printed circuit board (PCB) manufacturing system which was stated by Kim et al. [4]. In this system, each job can be considered as an order for a PCB product, which is specified by the product type, production quantity (order size) and due date. Therefore, there are multiple identical products to be processed to complete an order. At the drilling workstation in the PCB manufacturing system, a set of panels should be drilled holes for each order, and there are multiple parallel machines used to drill holes on the panels. At this workstation, each order (for a set of panels) can be split into groups of a (small) number of panels that are drilled together on a drilling machine. Such a group can be considered as a unit-job. Such panel groups can be processed on different machines simultaneously even if they are from the same lot or the same order. A batch of identical panel groups to be processed consecutively on a machine can be considered as a sub-job. Since the number of holes and their positions differ for different products, an NC program has to be downloaded on a drilling machine before a new product is started on the machine. In other words, set-up time is incurred when the product type of a sub-job to be processed on a machine is different from that of the sub-job just processed.

According to the standard three-field notation, the scheduling problem is denoted by $P|ST_{si}, split|\sum T_j$, where P is a system with machines in parallel. The ST_{si} represents the sequence independent setup time between jobs. The *split* imply that each job can be split into parts (sub-jobs) and processed independently on any of machines at the same time. The objective is to find the order in which the sub-jobs go through the system so that the sum of the tardiness of jobs is minimized.

The scheduling problems of identical parallel machine with splitting jobs that minimize total tardiness are considered to be in NP-hard class [6]. The sub-jobs of a total of n jobs on a machine can be sequenced in a total of $n!$. Considering the fact that the sequencing on a machine displays no dependence on any other machine, the number of unit-jobs constituting the sub-jobs can be assigned values as many as the number of unit-jobs of that job, a total of $(n!)^m(u+1)^{nm}$ possible schedules that can be formed together with those unsuitable or those not meet the constraints.