Computers & Operations Research 73 (2016) 132–140

Contents lists available at ScienceDirect

Computers & Operations Research

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

A beam search heuristic for scheduling a single machine with release dates and sequence dependent setup times to minimize the makespan

Mario C. Vélez-Gallego [a,*], Jairo Maya [a], Jairo R. Montoya-Torres [b]

[a] Departamento de Ingeniería de Producción, Universidad EAFIT, Carrera 49 #7 sur 50, Medellín, Colombia
[b] Escuela Internacional de Ciencias Económicas y Administrativas, Universidad de La Sabana, Km. 7 Autopista Norte de Bogotá D.C., Chía, Cundinamarca, Colombia

ARTICLE INFO

Article history:
Received 25 April 2013
Received in revised form
22 December 2015
Accepted 8 April 2016
Available online 9 April 2016

Keywords:
Single machine scheduling
Release dates
Sequence-dependent setup times
Beam search
Makespan

ABSTRACT

This paper considers the problem of scheduling a set of jobs subject to arbitrary release dates and sequence-dependent setup times on a single machine with the objective of minimizing the maximum completion of all the jobs, or makespan. This problem is often found in manufacturing processes such as painting and metalworking. A new mixed integer linear program (MILP) is firstly proposed. Because the problem is known to be NP-hard, a beam search heuristic is developed. Computational experiments are carried out using a well-known set of instances from the literature. Our results show that the proposed heuristic is effective in finding high quality solutions at low computational cost.

1. Introduction

Production scheduling is a decision-making process commonly found in both manufacturing and service industries [1]. Efficient production schedules can result in substantial improvements in productivity and cost reductions. Works in literature commonly simplify scheduling problems by assuming that the jobs are all available at the time of scheduling (i.e. the jobs are released to the system at time zero). On the other hand, in real practice in processes such as painting and metal working, the decision to manufacture multiple products on a common resource results in the need for setup activities, representing costly disruptions to the production processes. Setup reduction is therefore an important feature of continuous improvement programs [2]. In the literature, most of the works dealing with sequence-dependent setups assume that the setup times are symmetric, which is not always the case in practice [3].

This paper considers the problem of scheduling a set of jobs on a single machine under non-identical job release dates and sequence-dependent setup times. Formally, a set $J=\{j_1, j_2,\ldots j_n\}$ of n jobs need to be scheduled on a single machine with the objective of minimizing the makespan (i.e. $C_{max}=\max\{C_j\}$), where C_j is the completion time of job j in the sequence. The processing time and release date of job $j\in J$ are respectively denoted as p_j and r_j and are known in advance. A non-anticipatory setup time s_{ij} is incurred if job j is processed just after job i. If job j is the first job in the sequence, a setup time q_j is spent in order to get the machine prepared for processing such job. For simplicity a dummy job (i.e job j_0) with $p_0=0$ and $r_0=0$ is introduced to represent the setup of the first job in the sequence. We let $J'=J \cup \{j_0\}$ be the extended set of jobs so that $s_{0j}=q_j$ and $s_{j0}=0$ for all $j\in J$. Following the standard scheduling notation the problem under study can be denoted as $1|r_j, s_{ij}|C_{max}$. Without loss of generality we assume that all values are non-negative integers. The objective of this paper is to propose a new mixed integer linear program (MILP) formulation. Also, as the problem is known to be NP-hard [1], a beam search (BS) heuristic is proposed. The performance of the proposed BS heuristic is compared with the MILP when implemented in a commercial solver. The remainder of this paper is as follows. Section 2 presents a review of related literature. Section 3 is devoted to the MILP, while the BS heuristic is presented in Section 4. Section 5 presents in detail the results of the computational experiments carried out to assess the performance of the proposed approach. This paper ends in Section 6 by presenting some concluding remarks.

2. Literature review

Scientific work on sequence-dependent job scheduling has been published since the middle of the 1960s, with the works of Gilmore and Gomory [4], and Presby and Wolfson [5]. The former authors solved the problem using ideas from the classical traveling salesman problem, while the latter authors provided an

* Corresponding author.

http://dx.doi.org/10.1016/j.cor.2016.04.009
0305-0548/

optimization algorithm that is suitable only for small problem instances. Three major reviews on scheduling problems involving setup considerations have been published in the past. The first review we are aware of was presented by Allahverdi et al. [6]. In such review the authors classified the scientific literature in the area depending upon the nature of the setup times into two general groups: sequence independent and sequence dependent setup times. A setup is sequence independent if the time spent in setup activities depends only on the job being processed, regardless of the immediately preceding job. On the other hand, if the setup time depends on both the job being processed and the immediately preceding job, then the setup is said to be sequence dependent. A further classification depends on whether the problem being solved considers grouping jobs into batches or not. If jobs that are somehow similar are grouped into batches and all the jobs in a batch share a common setup, then the problem being solved is said to be a batch setup problem. If that is not the case and a setup is needed for every job being processed, then a no-batch setup problem is at hand. Following these two general classifications the authors presented a comprehensive review of the literature for the most common shop environments: single machine, parallel machines, flow shops and job shops. Later, Allahverdi et al. [7] and Allahverdi [8] updated the survey in [6] following the same classification scheme. In these reviews the authors introduced the distinction between anticipatory and non-anticipatory setups. A setup is said to be non-anticipatory if the setup operation cannot be performed before the job that is to be processed becomes available on the machine; otherwise the setup is said to be anticipatory. The review by Yang [9] classified the literature according to three different dimensions: (i) job vs. class, (ii) sequence-dependent vs. sequence-independent setup times, and (iii) separable vs. unseparable setups. By job vs. class the author presents what in the previously presented reviews was referred to as job batches, and by separable vs. unseparable setups the author refers to what in [7] was defined as anticipatory vs. non-anticipatory setups. Following this classification the author provided a thorough review of the literature in the field. We now describe the academic work found in the literature that is closer to the problem we address in this paper; this is, single machine scheduling problems under both sequence dependent setups and arbitrary release dates without any batch consideration. For the sake of completeness we include in this review works in this group that are also mentioned in the previously described reviews.

The first work found in the literature addressing a problem with these characteristics is the work of Farn and Muhlemann [10]. In their work the authors considered the dynamic version of the problem and compared six heuristics developed to minimize the total changeover time. A MILP and a heuristic algorithm using lower bounds and dominance criteria was developed by Bianco et al. [11] for the problem denoted as $1|r_j, s_{ij}|C_{max}$ with anticipatory setup times. Uzsoy et al. [12] proposed several heuristics for minimizing the maximum lateness and the number of tardy jobs. The maximum lateness problem was also considered in the work of Ovacikt and Uzsoy [13], in which the authors proposed a decomposition heuristic to solve it. Asano and Ohta [14] proposed a branch and bound algorithm to minimize the total earliness, whereas in [15] the same authors developed another branch-and-bound approach to minimize the maximum tardiness. The same objective (i.e. T_{max}) was also addressed by Shin et al. [16], who implemented a tabu search heuristic to solve the problem. The authors reported obtaining better results than those reported in [13] for the same set of test instances. For the more general objective of total weighted tardiness (i.e. $\sum_j w_j T_j$) a heuristic algorithm with complexity of $O(n^3)$ was proposed by Chang et al. [17]. Under the makespan objective function (i.e. C_{max}), heuristic approaches based on random generation of processing sequences were proposed by Montoya-Torres et al. in [2,18]. These works, however, do not consider a setup time for the first job in the sequence (i.e. it was assumed that the machine is always ready to process the first job of the sequence, which cannot clearly be known in advance). The objective of minimizing the total weighted completion time was considered by Chou et al. [19]. A MILP formulation, a lower bound, two constructive heuristics and a branch and bound algorithm were presented and tested on a set of randomly generated instances. Finally, the most recent work in the area is the work of Xi and Jang [20]. In such work the authors addressed the problem with the total weighted tardiness as the objective function and proposed two new ATC-based dispatching rules. Extensive computational experiments showed that the proposed rules yield near optimal solutions in most of the instances tested with up to ten jobs.

To the best of our knowledge, the works in [2,11,18] are the only works in the literature that address problems similar to that we study in this research. The main difference between the problems addressed in these works is in the nature of the setups: while anticipatory setups are considered in [11] we consider non-anticipatory setups. Also, as more than 20 years have passed since the publication of the cited reference, an up to date computational experiment is helpful in determining how modern solvers perform on such problems, and what instance sizes it is now possible to solve to optimality within a reasonable amount of time. As stated before, the difference between our work and that in references [2] and [18] is that these two works do not consider a setup time for the first job in the sequence. As per the proposed solution approach, the heuristic known as beam search has been successfully applied to solve various combinatorial optimization problems, such as the work of Akeb et al. in packaging [21], López-Ibáñez and Blum in routing [22], Blum and Miralles in assembly line balancing [23], and the works in scheduling of Valente and Alves [24], Valente [25], Valente and Alves [26], Esteve and Aubijoux [27], Ying [28], and Rakrouki et al. [29]. Some of these works consider single machine scheduling, with a few considering either job release dates or machine setup times.

3. Mathematical model

The model presented here is an extension to the model proposed in [11] that accounts for non-anticipatory setups. The following notation needs to be defined:

Sets

J original set of jobs

J' extended set of jobs

Parameters

p_j processing time of job j

r_j release date of job j

s_{jk} non-anticipatory setup time to change from job j to job k

π sum of the processing times of the jobs in J (i.e. $\pi = \sum_j p_j$).

MA large positive real number

Decision variables

x_{jk} binary variable that is equal to one if job k is processed just after job j in the sequence.

t_j time at which job j starts its non-anticipatory setup.

σ total machine idle time in the schedule.

The proposed MILP for the problem under study is the following:

$$\text{Minimize } C_{\max} = \pi + \sum_j \sum_k s_{jk} x_{jk} + \sigma \tag{1}$$

$$\text{Subject to: } \sum_{i \in J'} x_{ij} = 1 \qquad \forall\, j \in J' \tag{2}$$