Computers & Industrial Engineering 121 (2018) 189–194

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

A new model of parallel-machine scheduling with integral-based learning effect

Bartłomiej Przybylski

Adam Mickiewicz University in Poznań, Faculty of Mathematics and Computer Science, Umultowska 87, 61-614 Poznań, Poland

ARTICLE INFO

Keywords:
Parallel-machine scheduling
Variable processing times
Learning effect
Maximum completion time
Total completion time

ABSTRACT

We introduce a new model of parallel-machine scheduling with job processing times described by proper Riemann integrals of a given function. We also formulate and prove a few properties of that model. Based on presented results, we show that some problems of parallel-machine scheduling of jobs with integral-based learning effect can be solved using polynomial algorithms applied earlier to fixed job processing times.

1. Introduction

In many real-world scheduling problems job processing times are not fixed but variable. These processing times may change in reaction to some environmental factors, such as the amounts of resources available, the starting times of the jobs or their positions in a schedule. Recently, scheduling problems with *learning effect*—where the processing times of jobs decrease as their positions in a schedule increase—are gaining more and more attention. A dual group of scheduling problems with the so-called *ageing effect* is also widely considered. In this paper, we present some results concerning both mentioned groups, though we mainly focus on the learning effect in view of its greater popularity in scheduling applications.

Scheduling models with learning effect grew up from the observation that manufacturing experience may have a positive impact on the time needed to complete a job. As far back in 1930s, Wright (1936) made an observation that the processing times of aircraft industry production tasks decrease in conjunction with learning. Since then, many empirical studies have confirmed the relation between learning and the time needed to complete a job. Finding a good theoretical model describing this relation became the crucial point of decreasing costs and increasing speed in the areas of manufacturing, project management and software development—see e.g. Globerson and Seidmann (1988), Raccoon (1996), Anzanello and Fogliatto (2011), and Peteghem and Vanhoucke (2015).

Most of known models of learning and ageing effects are discrete and, to the best of our knowledge, no results related to parallel-machine scheduling of jobs with continuous learning effect have been published earlier. We introduce a new model of parallel-machine scheduling with variable processing times described by Riemann integrals. In our model, the processing time of a job is described by a Riemann integral of a given positive function. Riemann integrals, as generalized sums, can be used to describe processing times of jobs, eliminating most of the disadvantages of discrete learning and ageing models. This is caused by the fact that Riemann integrals can be used together with functions that approximate the continuous change of the worker's experience. Such an approach can be justified in practice. For example, let us consider a worker who screws on a complex element on a certain stage of car production, or paints a large surface. Such situations can be modeled by choosing appropriate integrable functions in our model, because a single job can be considered as a series of similar operations affecting the worker's overall experience. Moreover, as we show later, we can adapt our model to reflect similar cases for many known position-dependent models.

The paper is organized as follows. In Section 2, we review main models of scheduling with learning effects. In Section 3, we propose a new model of integral-based variability, where the processing time of a job is calculated as a Riemann integral of a given positive function on a specific interval. We also present some examples that illustrate the model itself and we prove a few of its properties. In Section 4, we analyse the relation between scheduling problems in the new model and some scheduling problems with fixed job processing times. We complete the paper by including conclusions and remarks on the future research in Section 5.

2. Literature review

The first model of position-dependent scheduling, where the actual processing time of a job depends on the number of jobs executed earlier, was proposed by Gawiejnowicz (1996). He considered a set of general

E-mail address: bap@amu.edu.pl.

https://doi.org/10.1016/j.cie.2018.05.035
Received 15 December 2017; Received in revised form 11 May 2018; Accepted 22 May 2018
Available online 25 May 2018
0360-8352/

position-dependent scheduling problems. A variation of this model, limited to the effect of learning, was introduced later by Biskup (1999). He assumed that the actual processing time of the jth job scheduled on the rth position, $p_{j,r}$, is the product of its basic processing time, $\overline{p}_j$, and the so-called learning factor, that is $p_{j,r} = \overline{p}_j r^a$, where $a < 0$ is a constant learning index. Mosheiov and Sidney (2003) extended this model to the case, where the learning indices are job-dependent, that is $p_{j,r} = \overline{p}_j r^{a_j}$. Both of these models have constituted a base for a rich research in the area of scheduling with learning effect. Below, we briefly review main models related to our model, especially those that are applied in manufacturing.

Kuo and Yang (2006) proposed a model, where the processing time of a job depends not on the number of jobs executed earlier, but on the sum of their basic processing times. In particular, they assumed that $p_{j,r} = \overline{p}_j (1 + \sum_{k=1}^{r-1} \overline{p}_{[k]})^a$, where $\overline{p}_{[k]}$ is the basic processing time of a job scheduled on the kth position. An exact dynamic programming algorithm for some parallel-machine scheduling problems within this class of sum-of-processing-times-based models, was presented by Rudek (2017). Independently of Kuo & Yang, Koulamas and Kyparisis (2007) considered a group of single-machine scheduling problems, where $p_{j,r} = \overline{p}_j (1-(\sum_{k=1}^{r-1} \overline{p}_{[k]})/(\sum_{k=1}^{n} \overline{p}_k))^a$. Later, Okołowski and Gawiejnowicz (2010) considered parallel-machine scheduling problems within the model of general DeJong's learning effect. In this model, $p_{j,r} = \overline{p}_j [M + (1-M) r^a]$, where $0 \le M < 1$ is the incompressibility factor. The same model was also analysed by Ji, Yao, Yang, and Cheng (2015) and Ji, Tang, Zhang, and Cheng (2016). The list of models describing learning and ageing effects is constantly expanding. For example, Zhang et al. (2018) proposed new models of such kind by mixing job deterioration with DeJong's learning effect. We refer the reader to monographs by Agnetis, Billaut, Gawiejnowicz, Pacciarelli, and Soukhal (2014), Strusevich and Rustogi (2017), and to the survey paper by Azzouz, Ennigrou, and Said (2017) for more details on scheduling models with learning effects. Scheduling models with job deterioration have been reviewed in detail by Gawiejnowicz (2008).

Existing models of scheduling with learning effects have a certain disadvantage. Namely, they are based on the assumption that the actual processing time of a job is proportional to the value of a function of either the number of jobs executed earlier, the sum of their basic processing times, or the position of a job in a schedule. In general, the actual processing time of a job is a product of its basic processing time and a value of a non-increasing function that does not depend on this time. It means that the execution of two unit jobs takes less time than execution of one job which is two units long. However, such an effect does not always correspond to reality, as in many real-life situations the process of learning occurs not only before, but also during the execution of a job. One of the ways to take the continuity of the learning process into account, and thus to deal with the above inconvenience, is to assume every job to be a chain of unsplittable unit jobs, each of which is susceptible to the process of learning independently. The general integral-based model presented in this paper implements such an assumption. Furthermore, as we will show later, a variety of discrete models presented at the beginning of this section can be easily transformed to their integral-based counterparts.

The observation that the process of learning takes place also while the job is being executed has been already reflected in literature—for example, Janiak and Rudek (2008) analysed a single-machine problem in which the actual processing time of a job is given by a discrete and stepwise function that depends not only on a sequence of jobs executed earlier, but also on the job being scheduled itself. The latter approach has been continued by Pakzad-Moghaddam, Mina, and Tavakkoli-Moghaddam (2014).

3. Job processing times described by integrals

In terms of scheduling theory, our new model of parallel-machine scheduling with integral-based variability effects can be formulated as follows. We are given m parallel machines $M_1, M_2, \ldots, M_m$, and n jobs $J_1, J_2, \ldots, J_n$ with or without precedence constraints. Every job J_j is described by its basic processing time $\overline{p}_j$ and may be additionally described by its weight w_j. We assume that both the basic processing times and the weights are positive integers. The actual processing time of any job is variable and depends on its position in a specific sequence of jobs assigned to a particular machine. By $p_{j,r,q}$ we denote the actual processing time of job J_j executed on the rth position on machine M_q. However, if the information about exact placement of the job can be omitted, we just write $p_{j,r}$ or p_j. Moreover, we assume that $p_{j,r,q} = 0$, if job J_j is *not* placed on the rth position on the qth machine. Let us notice that for any feasible schedule T there is only one pair of values r and q related to job J_j, such that $p_{j,r,q} > 0$. By $S_j(T)$ and $C_j(T)$ we denote the start time and the completion time of the jth job in schedule T, respectively. However, if it does not lead to a misunderstanding, we simply write S_j and C_j.

We will now describe how the actual processing time of a job is calculated in our model. Let $L_q(r)$ be a *load function* expressing the total load of machine M_q, i.e. the sum of basic processing times of jobs executed on machine M_q on positions up to $r-1$,

$$L_q(r) = \left\{ \sum \overline{p}_j : j = 1, \ldots, n \text{ and } 1 \le r' < r \text{ and } p_{j,r',q} > 0 \right\}.$$

The actual processing time of job J_j executed on the rth position on the qth machine equals

$$p_{j,r,q} = \int_{L_q(r)}^{L_q(r)+\overline{p}_j} \varphi(s)\, \mathrm{d}s, \tag{1}$$

where φ is a positive and Riemann integrable function. This function describes a continuous change in machine's capability while the machine executes jobs.

Let us notice the following property of this definition: the value of the L_q function depends iteratively on exact placements of jobs, and the actual processing time of a job depends on the value of the L_q function. However, it is easy to see that $L_q(1) = 0$ for any q, and that in order to determine the actual processing time of a given job we only need to know the basic processing times of already executed jobs. If the actual processing times of jobs are determined by this model, we will write in short that $p_{j,r} = \int \varphi$.

Though our integral-based model can be used together with different objective functions, in this paper we analyse the problem of minimizing the values of regular objective functions, especially the maximum completion time, $C_{\max} = \max\{C_j : j = 1, 2, \ldots, n\}$, and the total completion time, $\sum C_j = \sum_{j=1}^{n} C_j$. However, unless specified otherwise, every time we say that a schedule is optimal, we mean that it is optimal with respect to the $C_{\max}$ objective function. Finally, through this paper we assume that the φ function is non-increasing unless specified otherwise. In other words, we limit ourselves to the integral-based learning effect.

In order to illustrate Eq. (1), we will now show an example instance of a parallel-machine scheduling problem within the integral-based model. For simplicity, we will use an extended three-field scheduling notation described in Gawiejnowicz (2008) and Strusevich and Rustogi (2017).

Example 1. We are given an instance of the $P2|prec|C_{\max}$ problem with two parallel machines and five non-preemptable jobs with precedence constraints presented in Fig. 1. The basic processing times of jobs are equal to 1, 1, 2, 2 and 3 units, respectively.

If the actual processing times of jobs are fixed and equal to their basic processing times, then there are eight optimal schedules for such input data. Two of these schedules are presented in Fig. 2.

Now, consider the case in which the actual processing times are described by the integral-based model with the function $\varphi(s) = (\lfloor s \rfloor + 1)^{-1}$. To generate an optimal schedule, we assign jobs J_1, J_3 and J_5 to machine M_1, and jobs J_2 and J_4 to machine M_2. Job processing