European Journal of Operational Research 222 (2012) 393–407

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Invited Review

Simple matching vs linear assignment in scheduling models with positional effects: A critical review

Kabir Rustogi, Vitaly A. Strusevich *

School of Computing and Mathematical Sciences, University of Greenwich, Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, UK

ARTICLE INFO

Article history:
Received 11 November 2011
Accepted 25 April 2012
Available online 14 May 2012

Keywords:
Scheduling
Single machine
Parallel machines
Positional effects
Job deterioration
Assignment problem

ABSTRACT

This paper addresses scheduling models in which a contribution of an individual job to the objective function is represented by the product of its processing time and a certain positional weight. We review most of the known results in the area and demonstrate that a linear assignment algorithm as part of previously known solution procedures can be replaced by a faster matching algorithm that minimizes a linear form over permutations. Our approach reduces the running time of the resulting algorithms by up to two orders, and carries over to a wider range of models, with more general positional effects. Besides, the same approach works for the models with no prior history of study, e.g., parallel machine scheduling with deterioration and maintenance to minimize total flow time.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Recent scheduling literature includes numerous publications on models in which the contribution of an individual job to the objective function is represented by the product of its processing time and a certain positional weight. Among relevant scheduling models that exhibit such a property are models with positional effects such as positional learning or positional deterioration, or models in which the shape of the objective function imposes a positional weight to each job.

In this paper, we review most of the known results in this area of scheduling, for the single machine and the parallel machine environments. We present a classification of a variety of models, often more general than previously known, that are applicable to various practical situations and are, in particular, capable of handling variable processing times and machine maintenance periods. We also describe a unified approach to the design of fast algorithms for solving the relevant scheduling problems based on the well-known matching technique, which for some or another reason has been neglected in most of prior studies.

The models that we study are extensions of the following family of classical scheduling problems. In the most general setting, the jobs of set $N = \{1, 2, \dots, n\}$ have to processed on $m \geqslant 1$ *uniform* machines $M_1, M_2, \dots, M_m$ without preemption. The jobs are simultaneously available at time zero. A machine can handle only one job at a time and is permanently available from time zero. Each job $j \in N$ is associated with an integer p_j, which can be interpreted as its processing time under normal conditions. Each machine M_i has speed $s_i \geqslant 1$, such that the normal processing time of job j assigned to machine M_i becomes p_j/s_i.

As particular cases of the setup described above, we consider problems with *identical* parallel machines ($s_i = 1$, $1 \leqslant i \leqslant m$), and problems with a *single* machine ($m = 1$). The completion time of job j in some schedule is denoted by C_j. In this paper, we mainly concentrate on two objective functions: *the makespan* $C_{\max} = \max\{C_j | j \in N\}$ and *the sum of the completion time* or *the total flow time* $\sum_{j \in N} C_j$. For the single machine case, we also discuss other objective functions.

In the classical scheduling theory, it is usually assumed that the processing time of a job is fixed and has a constant value p_j. In many real-life situations, however, the processing conditions may vary and affect the actual durations of jobs. Such a phenomenon where the actual processing time of a job is variable, is traditionally attributed to one of the following causes: (i) deterioration, (ii) learning and (iii) resource allocation.

Informally, in scheduling with *deterioration* we assume that the later a job starts, the longer it takes to process. On the other hand, in scheduling with *learning* the actual processing time of a job gets shorter, provided that the job is scheduled later. Scheduling problems with these two effects have received considerable attention in the recent past; we refer to Cheng et al. (2004), Biskup (2008), Gawiejnowicz (2008) and Gordon et al. (2008) for recent state-of-the-art reviews in these areas, as well as for references to practical applications of these models. Scheduling with *resource*

* Corresponding author. Tel.: +44 2083318662; fax: +44 2083318665.
E-mail addresses: K.Rustogi@greenwich.ac.uk (K. Rustogi), V.Strusevich@greenwich.ac.uk (V.A. Strusevich).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.04.037

allocation allows the processing time of a job to be resource dependent, so that each job is allocated a certain amount of resource, and jobs with more allotted resources benefit from faster processing. We do not review this class of problems in this paper. For various aspects of models with resource-dependent processing times, we refer to the recent reviews by Shabtay and Steiner (2007), Błażewicz et al. (2010), Hartmann and Briskorn (2010), Leyvand et al. (2010), Węglarz et al. (2011) and Różycki and Węglarz (2012).

The effects of learning and deterioration are essentially antonymous to each other, and in practically all papers these two phenomena are discussed separately. No matter learning or deterioration, the corresponding effects most commonly found in the literature belong to one of the following three types:

- *Positional*: the actual processing time of job j depends on p_j and on the position of the job in the sequence;
- *Time-Dependent*: the actual processing time of job j depends on the start time of the job;
- *Cumulative*: the actual processing time of job j depends on p_j and on the sum of normal processing times of all jobs sequenced earlier.

Recently, there have been publications that consider enhanced models, that combine two of the above listed three effects. This gives rise to an additional wide range of problems, e.g., to models with time-dependent deterioration and positional learning (see Wang, 2006), or with positional deterioration and time-dependent learning (see Yang, 2010), or with cumulative deterioration and positional learning (see Wu and Lee, 2008), among other, often somewhat exotic models.

Gawiejnowicz (2008) gives a comprehensive exposition of time-dependent models. Gordon et al. (2008) discuss typical models in which each of the three effects listed above is applied individually. A recent review by Janiak et al. (2011) focuses on the variety of the models in the area, including those with combined effects.

In this paper, we focus mainly on *positional* effects. In the simplest case, if a schedule is determined by a sequence $\pi = (\pi(1), \ldots, \pi(n))$ of jobs, then in the case of a positional effect, the actual processing time of job $\pi(r)$ that occupies position r in the sequence π is given by $g(r)p_{\pi(r)}$. We refer to the given values of $g(1), g(2), \ldots, g(n)$ as *positional factors*. The function g is given in the form of an ordered array of numbers such that, in the case of deterioration

$$1 = g(1) \leqslant g(2) \leqslant \cdots \leqslant g(n) \tag{1}$$

and

$$1 = g(1) \geqslant g(2) \geqslant \cdots \geqslant g(n) \tag{2}$$

in the case of learning. Often in scheduling literature the factors $g(r)$ are defined as a known function, e.g., polynomial in r and exponential in r, i.e., $g(r) = r^a$ and $g(r) = \gamma^r$, respectively. In such cases, the values $g(1), g(2), \ldots, g(n)$ must be computed, and it is assumed that this can be done in constant time. Notice that in the past, apart from some recent papers, e.g., Gordon and Strusevich (2009), the positional factors have not been considered as given by a general function g; instead only specific functions have been analyzed.

Below we present several rather informal examples of positional effects. The most common rationales for deterioration, normally provided for a time-dependent effect and probably first stated by Gawiejnowicz (1996), are as follows: a machine is served by a human operator who gets tired or the machine loses the processing quality of its tools as more jobs are processed. To demonstrate that a deterioration effect can be *positional*, imagine that in a manufacturing shop there are several parts that need a hole of the same diameter to be punched through by a pneumatic punching unit. Ideally, the time that is required for such an operation depends on the thickness of the metal to be punched through; and this will determine the normal processing times for all parts. In reality however, there occurs an unavoidable gas leakage after each punch, due to which the punching unit loses pressure, so that the later a part is subject to punching the longer it takes to perform it, as compared to the duration under perfect conditions. Clearly, a positional deterioration effect is observed.

Deterioration is not a phenomenon that should be tolerated for a long time. For the example above, after a considerable drop of pressure, the punching unit can be subjected to maintenance, so that the cylinder is refilled and the unit is as good as new, or close to that state. The models that combine positional deterioration and machine maintenance activities are among the most general models addressed in this paper.

A learning effect can occur when what we call machines are in fact human operators that gain experience and improve their performance rate with each processed job. Being part of the academia, the authors have noticed that positional learning takes place when a teacher marks a number of coursework scripts based on the same question paper. It takes a reasonably long time to mark the first two or three scripts, then the teacher realizes the key factors to be checked, typical strong or weak points to be looked for, and the marking process goes faster and faster with each marked script.

In the literature on scheduling with positional effects, learning and deterioration are often studied separately, although similar methods can be employed in either case and some algorithmic ideas are either directly transferable from one effect to the other or at least can be adapted. In this paper, we argue that there is no need to separate the studies on learning from those on deterioration. In fact, we demonstrate that we may look at a general positional effect, since many algorithms discussed in the paper work for an arbitrary effect, given by a non-monotone sequence of the positional factors.

These general positional effects can be found in practice as well. Extending the coursework marking example above, after marking a certain number of scripts, the teacher might get tired or bored, her attention becomes less focused and each new script may even take longer to mark than the one before. We are sure our academic colleagues know this feeling, and they also know the remedy: take a break, have a cup of coffee – in other words, perform maintenance.

Apart from the positional effects, there can be other reasons due to which the contribution of a job to the objective function turns out to be equal to its processing time multiplied by a certain positional weight. This positional weight is not necessarily related to the positional factors alone, but could also arise due to the shape of the objective function. For example, for the problem of minimizing the sum of the completion times on a single machine, the processing time of the job in position r is multiplied by $(n - r + 1)$ to represent the contribution of that job. Such cases will be discussed in more detail in later sections.

Our main attention will be devoted to models with positional weights, in which the jobs have to be scheduled into one or more *groups*. This happens, e.g., for problems with several parallel machines, so that each group will consist of the jobs assigned to a particular machine. Besides, jobs are also split into groups in models with machine maintenance, and a group is the subset of jobs processed on a machine either before the first maintenance period or between two consecutive periods. In any case, each group can be associated with its own *group-dependent* weight. Notice that models with group-dependent weights essentially have no history of prior study, except Rustogi and Strusevich (2012).

Positional weights can be further classified as being either *job-dependent* or *job-independent*, based on whether the underlying positional factor is of the form $g_j(r)$ (job-dependent) or $g(r)$ (job-independent), respectively. A job-dependent positional effect implies that each job has a different (unique) effect on the state of

Download English Version:

https://daneshyari.com/en/article/6898188

Download Persian Version:

https://daneshyari.com/article/6898188

Daneshyari.com