Computers & Operations Research 40 (2013) 2071–2082

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Makespan minimization flowshop with position dependent job processing times—computational complexity and solution algorithms

Agnieszka Rudek [a], Radosław Rudek [b,*]

[a] Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
[b] Wrocław University of Economics, Komandorska 118/120, 53-345 Wrocław, Poland

ARTICLE INFO

Available online 15 March 2013

Keywords:
Scheduling
Aging effect
Learning effect
Computational complexity
Heuristic

ABSTRACT

This paper considers flowshop scheduling problems where job processing times are described by position dependent functions, i.e., dependent on the number of processed jobs, that model learning or aging effects. We prove that the two-machine flowshop problem to minimize the maximum completion time (makespan) is NP-hard if job processing times are described by non-decreasing position dependent functions (aging effect) on at least one machine and strongly NP-hard if job processing times are varying on both machines. Furthermore, we construct fast NEH, tabu search with a fast neighborhood search and simulated annealing algorithms that solve the problem with processing times described by arbitrary position dependent functions that model both learning and aging effects. The efficiency of the proposed methods is numerically analyzed.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Although over 50 years have passed since Johnson [17] published the first paper on flowshop scheduling, it is still topical issue among researchers. Nevertheless, due to rigorous constraints, classical flowshop scheduling problems are perceived to be more interesting in a theoretical context than as a practical research (see [13]). It follows from observations that algorithms constructed on the basis of the classical models usually provide unsatisfactory (unstable) solutions for real-life flowshop problems, since these models do not take into consideration additional factors that are significant in practice, e.g., variable parameters [4,6,10,14,22,23,36,43]. Recently, a particular attention has been attracted by a group of problems, where job processing times change with the number of previously processed jobs (e.g., [24,26,41,46]).

If the time required to process a job decreases with the number of processed jobs, then such phenomenon is called the learning effect. It is typical for human activity environments or for automatized manufacturing, where a human support for machines is needed during activities such as operating, controlling, setup, cleaning, maintaining, failure removal, etc. Although learning can cease with time, it is often aroused by such factors as new inexperienced employees, extension of an assortment, new machines, more refined equipment, software update or general changes of the production environment [1]. However, highly automatized manufacturing systems may also benefit on the fact that if a machine does the same task repetitively, then the knowledge from the previous iterations can be used to improve the performance of a system when the task is processed the next time (e.g., [2]). Furthermore, machine learning methods, for instance reinforcement learning algorithms, that learn and operate on-line (see [39]), improve their efficiency on the basis of interactions with an environments (learning-by-doing). Thus, the performances of the systems optimized by such algorithms are improved with the number of their iterations (e.g., [3,16]). For a survey on scheduling problems with learning see [1,15].

On the other hand, if the times required to process jobs increase with the number of processed jobs, then it is called the aging or position dependent deteriorating effect. The problems with this phenomenon are present in many manufacturing systems, in which tiredness of human workers (e.g., [7]), tool wear of lathe machines (e.g., [32]) or decreasing activity of some chemical substances (e.g., [25]) affect the production output. More details concerning scheduling problems with the aging effect can be found in [9,21,44,45,47].

Although the makespan minimization flowshop problem with position dependent processing times (modelling learning or aging effects) was discussed (e.g., [23,38]), its computational complexity status for the two-machine case is still an open issue even for arbitrary aging characteristics. Therefore, the first main contribution of this paper is to prove that the two-machine flowshop problem to minimize makespan that is known to be polynomially solvable (see [17]) becomes NP-hard if job processing times are

* Corresponding author. Tel.: +48 71 368 0378; fax: +48 71 368 0376.
E-mail addresses: agnieszka.wielgus@pwr.wroc.pl (A. Rudek), radoslaw.rudek@ue.wroc.pl, rudek.radoslaw@gmail.com (R. Rudek).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.02.024

position dependent non-decreasing functions (aging effect) on one machine and strongly NP-hard if the aging effect concerns two machines.

Furthermore, there is a lack of efficient approximation algorithms for the makespan minimization flowshop problem with learning or aging effects (see [6,37,40,42]). Therefore, we contribute the field by providing fast heuristic and metaheuristic methods that solve the general version of the problem, i.e., with arbitrary position dependent job processing times that can describe both learning and aging effects. Namely, we construct the fast NEH algorithm that is significantly faster than the standard NEH (e.g., [6]). Moreover, we propose the fast neighborhood search that is much faster than the standard approach and can be used for tabu search and other metaheuristic algorithms solving similar problems.

To the best of our knowledge, the results presented in this paper have never been investigated in the scheduling domain.

This paper is organized as follows. Section 2 contains problem formulation and the computational status is established in Section 3. Subsequently, description of approximation algorithms is provided, followed by an experimental verification of their efficiency. The last section concludes the paper.

2. Problem formulation

In this paper, we investigate flowshop problems with the aging effect. These problems can be defined as follows. There are given a set $J=\{1,\ldots,n\}$ of n jobs and m machines, namely $M=\{M_1,\ldots,M_m\}$. Each job j consists of a set $O=\{O_{1,j},\ldots,O_{m,j}\}$ of m operations. Each operation $O_{z,j}$ has to be processed on machine M_z $(z=1,\ldots,m)$. Moreover, operation $O_{z+1,j}$ may start only if $O_{z,j}$ is completed. If it is assumed that machines have to process jobs in the same order, then the problem is called a permutation flowshop and such problems are considered in this paper. It is also assumed that each machine can process one operation at a time, and there are no precedence constraints between jobs. Operations are non-preemptive and available for processing at time 0 on M_1. Further, instead of operation $O_{z,j}$, we say job j on machine M_z.

The processing time $p_j^{(z)}(v)$ of job j that is scheduled in position v in a sequence on machine M_z $(z=1,\ldots,m)$ is a positive function dependent on its position v (i.e., the number of previously machined products, $v-1$). The function $p_j(v)$ models learning or aging effects if it is non-increasing or non-decreasing, respectively. Moreover, each job j is characterized by its normal processing time $a_j^{(z)}$ on machine M_z that is defined as the time required to process a job if the machine is not influenced by aging/learning, i.e., $a_j^{(z)} \triangleq p_j^{(z)}(1)$ for $z=1,\ldots,m$ and $j=1,\ldots,n$.

For the m-machine *permutation* flowshop problems the schedule of jobs on the machines can be unambiguously defined by their sequence (permutation) π. Thus, for each job $\pi(i)$, i.e., scheduled in the ith position in π, we can determine its completion time $C_{\pi(i)}^{(z)}$ on machine M_z

$$C_{\pi(i)}^{(z)} = \max\{C_{\pi(i)}^{(z-1)}, C_{\pi(i-1)}^{(z)}\} + p_{\pi(i)}^{(z)}(i), \tag{1}$$

where $C_{\pi(1)}^{(0)} = C_{\pi(0)}^{(z)} = 0$ for $z=1,\ldots,m$ and $C_{\pi(i)}^{(1)} = \sum_{l=1}^{i} p_{\pi(l)}^{(1)}(l)$ is the completion time of a job placed in position i in the permutation π on M_1.

The objective is to find such a schedule π of jobs on the machines that minimizes the maximum completion time (makespan) $C_{\max} \triangleq \max_{j\in J}\{C_j\}$ (in the considered problems $C_{\max} = C_{\pi(n)}^{(m)}$). The m-machine flowshop problem with non-decreasing (only aging), non-increasing (only learning) and arbitrary (learning/aging) processing times, according to the three-field notation scheme $X|Y|Z$, will be denoted as $Fm|AE|C_{\max}$, $Fm|LE|C_{\max}$ and $Fm|p_j^{(z)}(v)|C_{\max}$, respectively.

3. Computational complexity

In this section, we will prove the strong NP-hardness of the considered problem with the aging effect $F2|AE|C_{\max}$ and also NP-hardness of its special case. The proofs are based on similar observations as in [29].

3.1. NP-hardness

In this part, we will prove that the makespan minimization problem with the aging effect $F2|AE, p_j^{(1)}(v) = a_j^{(1)}|C_{\max}$ is at least NP-hard even if job processing times on the first machine are constant (i.e., it does not deteriorate).

Theorem 1. *The problem* $F2|AE, p_j^{(1)}(v) = a_j^{(1)}|C_{\max}$ *is NP-hard even if job processing times are constant on the first machine* (*i.e., this machine does not deteriorate*).

Proof. To prove this theorem, we transform Equal Cardinality Partition problem, that is known to be NP-complete, to the decision version of the considered scheduling problem with aging. At first the definition of Equal Cardinality Partition problem is given.

Equal Cardinality Partition (ECP) [8]: There are given positive integers t, B and $x_1,\ldots,x_{2t}$ of $2t$ positive integers satisfying $\sum_{q=1}^{2t} x_q = 2B$ for $q=1,\ldots,2t$. *Does there exist a partition of the set* $X=\{1,\ldots,2t\}$ *into two disjoint subsets* X_1 *and* X_2 *such that* $\sum_{q\in X_1} x_q = \sum_{q\in X_2} x_q = B$ *and* $|X_1| = |X_2| = t$?

The decision version of the problem $F2|AE, p_j^{(1)}(v) = a_j^{(1)}|C_{\max}$ (Decision problem with the Aging Effect on one machine, DAE1) is defined as follows: *Does there exist such a schedule* π *of jobs on the machines for which the criterion value* $C_{\max}$ *is not greater than the given value* y?

The transformation from ECP to DAE1 is constructed as follows. The instances of DAE1 contain $2t$ *partition* jobs $(j=1,\ldots,2t)$ constructed on the basis of the elements from the set X of ECP and one *enforcer* job e that is an auxiliary (additional) job, which forces the optimal schedules such that a sequence of jobs is important (i.e., jobs are scheduled before and after e). The parameters of the *partition* jobs $(j=1,\ldots,2t)$ are given as follows:

$$p_j^{(1)}(v) = a_j^{(1)} = 1 \quad \text{for } v=1,\ldots,2t+1,$$

$$p_j^{(2)}(v) = \begin{cases} tB + 2tx_j & \text{for } v=1,\ldots,t, \\ 3tB + tx_j & \text{for } v=t+1,\ldots,2t+1, \end{cases}$$

where $j=1,\ldots,2t$, and the parameters of the *enforcer* job e are

$$p_e^{(1)}(v) = a_e^{(1)} = 1 + tB(t+2) - t \quad \text{for } v=1,\ldots,2t+1,$$

$$p_e^{(2)}(v) = \begin{cases} tB & \text{for } v=1,\ldots,t+1, \\ 4tB(t+2) & \text{for } v=t+2,\ldots,2t+1, \end{cases}$$

and $y = 1 + 4tB(t+1)$ and $n = 2t+1$. Thus, we analyze step functions of job processing times. It is trivial to show that DAE1 belongs to the NP class and the given transformation is polynomial.

Now we will establish the position of the enforcer job e in an optimal schedule π for the constructed instances of DAE1. Observed that if the enforcer job e is scheduled in position $v_e > t+1$, then $p_e^{(2)}(v_e) = 4tB(t+2) > y$. On the other hand, if $v_e < t+1$, then

$$\begin{aligned} C_{\max} &> \sum_{i=1}^{v_e-1} p_{\pi(i)}^{(1)} + a_e^{(1)} + p_e^{(2)}(v_e) + \sum_{i=v_e+1}^{n} p_{\pi(i)}^{(2)}(i) \\ &> tB(t+2) + tB + (2t+1-v_e)3tB \\ &= tB(7t+6-3v_e) > tB(4t+6) > y. \end{aligned}$$

Download English Version:

https://daneshyari.com/en/article/10347945

Download Persian Version:

https://daneshyari.com/article/10347945

Daneshyari.com