Accepted Manuscript

Research on no-idle permutation flowshop scheduling with time-dependent learning effect and deteriorating jobs

Yuan-Yuan Lu

PII: S0307-904X(15)00615-0
DOI: 10.1016/j.apm.2015.09.081
Reference: APM 10780

To appear in: *Applied Mathematical Modelling*

Received date: 7 September 2014
Revised date: 21 April 2015
Accepted date: 22 September 2015

Please cite this article as: Yuan-Yuan Lu, Research on no-idle permutation flowshop scheduling with time-dependent learning effect and deteriorating jobs, *Applied Mathematical Modelling* (2015), doi: 10.1016/j.apm.2015.09.081

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Research on no-idle permutation flowshop scheduling with time-dependent learning effect and deteriorating jobs

Yuan-Yuan Lu[1]

College of Mathematics, Jilin Normal University, Siping, Jilin 136000, China

Abstract–This note considers no-idle permutation flowshop scheduling problems with time-dependent learning effect and deteriorating jobs. The objective functions are to minimize the makespan and the total (weighted) completion time, respectively. Low and Lin [1] (C. Low, W.-Y. Lin, Some scheduling problems with time-dependent learning effect and deteriorating jobs, Applied Mathematical Modelling 37 (2013) 8865-8875) showed that an optimal sequence for these problems can be solved in polynomial time. We demonstrate these results to be incorrect by counter-examples for the no-idle permutation flowshop scheduling problems with an increasing series of dominating machines (idm), then introduce new exact solution algorithms that polynomially solve these problems.

Keywords: Scheduling; No-idle; Flowshop; Learning effect; Deteriorating jobs

1. Introduction

A recent paper by Low and Lin [1] addresses scheduling problems with time-dependent learning effect and deteriorating jobs. For some single machine and flowshop scheduling problems, they showed that these problems can be solved in polynomial time, respectively. As we observe, some results of the flowshop scheduling problems in Low and Lin [1] are incorrect. In this note we give counter-examples to show that their results are incorrect results, and corrected versions of related results are developed.

We follow the notations and terminologies given in Low and Lin [1]. There are n independent, non-preemptive jobs $J_1, J_2, \ldots, J_n$ available for processing at time 0 in an m-machine flowshop. Each job J_j consists of m operations $O_{1j}, O_{2j}, \ldots, O_{mj}$ $(j = 1, 2, \ldots, n)$. Let p_{ij} denote the normal processing time of operation O_{ij} (i.e., job J_j $(j = 1, 2, \ldots, n)$ on machine M_i $(i = 1, 2, \ldots, m)$). All the jobs are processed in the same order on each machine (i.e., a permutation schedule). The machines are continuously available from time zero onwards and each machine can handle at most one job at a time. As in Low and Lin [1], we assume that the actual processing time $p_{ijr}(t)$ of job J_j on machine M_i at the rth position in a sequence is

$$p_{ijr}(t) = p_{ij}\left(1 - \frac{\sum_{l=1}^{r-1} p_{i[l]}}{\sum_{l=1}^{n} p_{il}}\right)^a b^{r-1} + st = p_{ij}\left(\frac{\sum_{l=r}^{n} p_{i[l]}}{\sum_{l=1}^{n} p_{il}}\right)^a b^{r-1} + st, i = 1, 2, \ldots, m; r, j = 1, 2, \ldots, n, \quad (1)$$

where t is the starting time of job J_j on machine M_i, $a \geq 1$ and $0 < b < 1$ are learning effect indexes, $s \geq 0$ is the deterioration rate for all the jobs, and $[l]$ denotes the job scheduled lth position in a sequence. We also assume that the no-idle permutation scheduling problem (Adiri and Pohoryles [2], and Wang and Xia [3]) is formulated when each machine must process jobs without any interruption from the start of processing the first job to the completion of processing the last job.

For a given schedule $\pi = [J_1, J_2, \ldots, J_n]$, let $C_{ij} = C_{ij}(\pi)$ denote the completion time of job J_j on machine M_i (i.e., operation O_{ij}), $C_j = C_{mj}$ denote the completion time of job J_j, and w_j denote the weight of job J_j. The objective functions to be minimized are the makespan (i.e., $C_{\max} = \max\{C_j | j = 1, 2, \ldots, n\}$), the total

[1] Corresponding author. E-mail addresses: luyuanyuan_jilin@163.com

Download English Version:

https://daneshyari.com/en/article/10677574

Download Persian Version:

https://daneshyari.com/article/10677574

Daneshyari.com